SISSA
Scuola
Internazionale
Superiore di
Studi Avanzati

DANIELI

Mathematics Area - PhD course in
Mathematical Analysis, Modelling, and Applications

TOWARDS REAL-TIME DIGITAL TWINS IN STEELMAKING: THE ROLE OF REDUCED ORDER METHODS IN CONTINUOUS CASTING TUNDISH PROCESS

Candidate:
Harshith GOWRACHARI

Advisor:
Prof. Gianluigi ROZZA
Co-advisor:
Prof. Giovanni STABILE

Industrial Advisors
from Danieli Research Center:
Mr. Gianfranco MARCONI
Mr. Gianluca BAZZARO
Mr. Mattia Giuseppe BARRA
Mr. Gabriele GUASTAFERRO

Academic Year 2024-25

MAI PER SEGUIR VIRTUTE E CANOSCENZA
SISSA

Grant Acknowledgment

I am grateful for the support provided by my industrial PhD grant, sponsored by Danieli and the PON Research and Innovation 2014-2020 program of the Italian Ministry of Research. This funding enabled me to present my research outcomes at international conferences and participate in advanced training schools and workshops across Europe.

I would also like to thank the ARIA (Accurate ROMs for Industrial Applications) project, funded by the H2020-MSCA-RISE-2019 program of the European Commission under Grant Agreement No. 872442, for sponsoring my three-month research visit, hosted by Inria Bordeaux (TEAM MEMPHIS) and VALOREM, Bordeaux, France.

Furthermore, I acknowledge the support of the CYPHER COST Action CA22151, which awarded me a travel grant to attend the training school on Machine Learning Methods in Reactive Flows and the workshop on Digital Twins for the Decarbonization of Hard-to-Abate Industries, held at CERTH in Thessaloniki, Greece.

Thanks to all the institutions and individuals who made these opportunities possible. These experiences advanced my research and allowed me to connect with like-minded researchers, experts, and professionals.

La borsa di dottorato è stata cofinanziata con risorse del
Programma Operativo Nazionale Ricerca e Innovazione 2014-2020 (CCI 2014IT16M2OP005),
Fondo Sociale Europeo, Azione I.1 "Dottorati Innovativi con caratterizzazione Industriale"













To my family and my perseverance.

Abstract

Continuous casting is the predominant method used in steel production, accounting for 96% of the world's steel production. Within this process, the continuous casting tundish plays a critical role as a metallurgical reactor in the continuous casting process, with its flow characteristics serving as a key parameter in the production of high-quality steel. During a typical production cycle, the tundish undergoes one steady-state operation and three transient operations, occurring during the initial filling, ladle changeover, and final emptying phase. This work particularly focuses on the phenomena associated with steady-state operation and the ladle changeover transient operations. The steady-state operation is the most prevalent stage during the production cycle, while the ladle changeover transient operation typically lasts no more than 10-15 minutes.

In steady-state operation, the key flow characteristics are typically evaluated using residence time distribution (RTD) curves. This study investigates fluid flow in a single-strand tundish through mathematical modelling, validated against water-model experiment results. Full-order steady simulations were carried out under both isothermal and non-isothermal conditions to assess the influence of thermal buoyancy on flow behavior. The results indicate that buoyancy effects under non-isothermal conditions have a negligible impact on the overall velocity field. The resulting steady-state flow fields were then used to perform transient tracer transport simulations to derive RTD curves and volume partitioning. Additionally, a sensitivity analysis is conducted to examine the effect of inlet velocity on RTD characteristics.

In the ladle changeover transient operation, the key quantity of interest is steel grade intermixing, particularly the new steel grade transition at the outlet of the tundish. Accurate prediction of grade intermixing during ladle changeover operations is critical for maintaining steel quality and minimizing material losses in the continuous casting process. Among various factors influencing grade intermixing, operating parameters play a significant role, in addition to tundish geometry and flow control devices. In this study, three-dimensional,

transient, multi-phase turbulent flow simulations are conducted to investigate the ladle changeover operation. During this process, the molten steel level in the tundish typically varies over time, significantly affecting the grade intermixing phenomena. The influence of ladle change time on intermixing time has been presented.

To address the high computational cost associated with the full-order industrial-scale simulations of both steady-state and transient ladle changeover operations of the tundish, reduced order methods are exploited. These methods enable efficient evaluation of key flow characteristics, such as RTD curves and steel grade intermixing time. For the steady-state operation RTD analysis, both projection-based model order reduction and data-driven reduced order model (ROM) strategies are explored. For parameter-time dependent RTD analysis, a projection-based ROM and data-driven model approach are employed to capture the flow characteristics within the tundish system. For the transient ladle changeover operation, a data-driven ROM approach is developed using reservoir computing (RC), specifically echo state networks (ESNs), in combination with proper orthogonal decomposition (POD). This POD–RC–ROM framework enables rapid prediction of new steel grade evolution and intermixing times, demonstrating strong potential for real-time process monitoring and optimization in continuous casting operations.

Advection-dominated problems are typically noticed in nature, engineering systems, and industrial processes. However, traditional linear model reduction methods, such as proper orthogonal decomposition and reduced basis methods, are ill-suited for these problems, due to slow Kolmogorov n-width decay, leading to inefficient and inaccurate ROMs. To address this, two advanced non-linear model reduction strategies are proposed in this work. The first approach proposes the complete neural network shift-proper orthogonal decomposition based reduced order model algorithm, consisting of offline-online stages. It utilizes a neural-networks shift augmented transformation technique for automatic shift detection. This method learn a parameter dependent mapping between the original solution manifold and a transformed manifold, where linear compression techniques are more effective. The second method employs cross-correlation based snapshot registration to align transported features to a reference frame, accelerating the Kolmogorov n-width decay and enabling the construction of efficient ROMs using linear methods.

Both methods offer complete offline-online stages for the development of non-intrusive or data-driven ROM. The effectiveness of these approaches is demonstrated on several benchmark problems, including the 1D linear advection equation, the 2D isentropic convective vortex, and a 2D two-phase flow scenario inspired by the flow physics within a tundish.

Contents

Abstract **iii**

Contents **vii**

1 Introduction and Motivation **1**
1.1 Continuous Casting 1
1.2 Tundish 2
1.3 Physical problem 3
1.4 Specific problem 5
1.4.1 Steady state operation 5
1.4.2 Ladle changeover operation 7
1.5 Objectives 9
1.6 Thesis contribution 10
1.7 Thesis outline 12

I Full Order Model **17**

2 Discretization method **19**
2.1 Introduction 19
2.2 Finite Volume Method 20
2.2.1 Convective flux interpolation 22
2.2.2 Gradient computation 26
2.2.3 Diffusive flux interpolation 27
2.3 Time discretization 29

3 Tundish steady state operation **31**
3.1 Introduction 31
3.2 Experiment 34
3.3 Full order model 37
3.3.1 Governing equations 37
3.3.2 Turbulence model 37

3.3.3 Boussinesq approximation 38
3.3.4 Species transport equation 38
3.4 Numerical setup 39
3.4.1 Computational domain 39
3.4.2 Boundary conditions 40
3.5 Numerical method 40
3.6 Results and discussion 42
3.7 Summary 46

4 Tundish ladle changeover operation **49**
4.1 Introduction 49
4.2 Full order model 51
4.2.1 Governing equations 51
4.2.2 Volume of Fluid method 52
4.2.3 Turbulence model 52
4.2.4 Species transport equation 53
4.3 Numerical setup 53
4.3.1 Computational domain 53
4.3.2 Tundish operating and boundary conditions 54
4.4 Numerical method 56
4.5 Results and discussion 58
4.5.1 Effects of ladle change time 58
4.6 Summary 61

II Reduced Order Model **63**

5 Preliminaries on model order reduction **65**
5.1 Introduction 65
5.2 Model reduction methods 66
5.2.1 Approximability of solution manifold 66
5.2.2 Linear dimensionality reduction 66
5.2.3 Non-linear dimensionality reduction 67
5.3 Curse of dimensionality 67
5.4 Overcoming slow Kolmogorov n-Width decay 68
5.5 Reduced order model construction: An overview 68

6 ROM using neural network shift augmented manifold transformation **71**
6.1 Introduction 71

6.2 Model problem . . . 74
6.3 Reduced order model . . . 75
6.3.1 Proper orthogonal decomposition . . . 75
6.3.2 Solution manifold approximation . . . 76
6.4 Neural network shift augmented transformation . . . 77
6.5 NNsPOD-ROM algorithm . . . 81
6.6 Numerical Experiments . . . 83
6.6.1 1D Linear advection equation . . . 83
6.6.2 2D Isentropic convective vortex . . . 89
6.6.3 2D Two-phase flow . . . 93
6.7 Conclusion . . . 102

7 ROM via cross-correlation based snapshots registration 105
7.1 Introduction . . . 105
7.2 Cross-correlation based snapshot registration . . . 106
7.3 ROM via cross-correlation based snapshot registration algorithm 108
7.4 Numerical experiments . . . 110
7.4.1 1D Travelling wave . . . 110
7.4.2 2D isentropic convective vortex . . . 113
7.5 Conclusion . . . 117

8 Model order reduction strategies for tundish RTD analysis 119
8.1 Introduction . . . 119
8.2 Projection based model order reduction . . . 123
8.2.1 Offline stage . . . 123
8.2.2 Online stage . . . 125
8.3 Results and discussion . . . 126
8.4 Conclusion . . . 131

9 ROM for grade intermixing in a continuous casting tundish 133
9.1 Introduction . . . 133
9.2 Reduced order model . . . 136
9.2.1 Proper orthogonal decomposition . . . 137
9.2.2 Reservoir computing . . . 138
Echo state network . . . 140
Evaluation of modal coefficients . . . 141
9.2.3 POD-RC-ROM algorithm . . . 142
9.3 Results and discussion . . . 143
9.4 Conclusion . . . 146

10 Conclusions and perspectives **147**

A Neural-networks hyperparameters **153**
A.1 NNsPOD-ROM 153
A.1.1 Scaling of data 153
A.1.2 Neural networks' Architecture 153
A.2 Reservoir computing hyperparameters 154

B FOMs and ROMs: Computational resources & performance **155**
B.1 Tundish steady-state operation 155
B.2 Tundish ladle changeover operation 157

Bibliography **159**

List of Figures

1.1 Schematic representation of Continuous Casting machine 1
1.2 Schematic representation of a single strand tundish 3
1.3 Summary of tundish operations and the associated phenomena 4
1.4 Schematics of the complete workflow of a continuous casting tundish steady-state operation RTD analysis. 6
1.5 Process diagram of the ladle changeover operation. 8
1.6 Thesis overview: The first part focuses on the Full Order Model (FOM) of tundish operations, and the second part focuses on non-linear model reduction and reduced order modelling strategies to efficiently obtain the key quantities of interest (QoIs) in tundish operations. 15

2.1 Showing the control volume of the element 21
2.2 Upwind differencing stencil 23
2.3 Face value computation 24
2.4 Structured vs. unstructured mesh alignment 25
2.5 Face value interpolation in unstructured meshes 26
2.6 Face gradient interpolation 27
2.7 Orthogonal and non orthogonal mesh 27
2.8 Non-orthogonality treatment in the over-relaxed approach. . . . 28

3.1 Schematic representation of a single-strand tundish with flow modifiers. 33
3.2 Test facility dedicated to water model experiments on tundish models. The test facility depicted is located at the Danieli Research Center in Buttrio, Province of Udine, Italy. 34
3.3 Concentration-time curves measured at the inlet and outlet of a single-strand tundish, highlighting the effect of the recirculation of the tracer inside the system. 36
3.4 Deconvolution process to obtain the actual RTD curve of the tundish 36

3.5 Computational domain of a 1:1 scaled single-strand tundish (isometric view); the inlet and outlet are indicated, and all other boundaries are treated as walls. 39
3.6 Discretized computational domain of a 1:1 scaled single-strand tundish. **Left**: Isometric view of the discretized domain; **Right**: 2D midplane showing mesh refinement at the inlet jet impingement region and outlet, and consists of six prism layers adjacent to the walls. 40
3.7 Imposed tracer concentration at the inlet boundary for the transient tracer simulation. 41
3.8 Locations of horizontal and vertical lines used for extracting streamwise velocity profiles within the tundish. 43
3.9 Streamwise velocity distributions along horizontal lines at different y-positions (shown in Figure 3.8), under isothermal and non-isothermal conditions. 43
3.10 Streamwise velocity distributions along vertical line at different x locations (shown in Figure 3.8), under isothermal and non-isothermal conditions. 43
3.11 RTD curve obtained from FOM CFD simulations, highlighting key flow characteristics. 44
3.12 Volume partitioning derived from RTD analysis based on FOM simulation results. 45
3.13 Comparison of residence time distribution (RTD) curves obtained from experiments and full-order simulations for a single-strand tundish. 45
3.14 RTD curves obtained from full-order simulation, illustrating the variation in tracer concentration over time for each parameter, as listed in Table 3.3. 46

4.1 Computational domain of a 0.5 scaled single strand tundish. **Left**: Isometric view highlighting the inlet, outlet, and symmetry plane, with all other boundaries treated as walls. **Right**: 2D left-side view overlooking the symmetry plane. 54
4.2 Discretized computational domain of a 0.5-scale single-strand tundish. **Left**: Isometric view of discretized domain; **Top right**: 2D symmetry plane showing the mesh refinement strategies to capture accurately the interface development and turbulence mixing phenomena; **Bottom right**: Inner jet impingement zone with an impact pot to enhance turbulence mixing. 55

4.3 New steel grade transition contours corresponding to a ladle change duration of 64s and a filling time of 40.40s. 59
4.4 Intermixing time corresponding to a ladle change duration of 64s and a filling time of 40.40s. 59
4.5 Effect of ladle change time on melt bath height and residual volume fraction. 61
4.6 Effect of ladle change time on intermixing time 61

6.1 Schematic representation of the complete workflow for constructing NNsPOD-ROMs 84
6.2 Snapshots of the 1D linear advection equation collected at 21 uniformly sampled time instances, $t \in [0, 10]$, over 256 equidistant spatial nodes $x \in [0, 12]$. 85
6.3 For 1D linear advection equation, **Left**: Showing trained InterpNet, able to reconstruct the field values of reference snapshot corresponding to time instance, $t_{\text{ref}} = 4$ and **Right**: The original snapshot at time t = 10 (labelled as physical), the same transported to the reference frame utilising the shifted space (labelled as shifted), using optimal shift obtained by shift operator $\mathcal{T}_{\text{shift}}$ and the reference frame is also shown for comparison. 86
6.4 For 1D linear advection equation, **Left**: showing training set snapshots in physical space x and **Right**: the shifted snapshots at the reference frame in physical space after neural network shift augmented transformation. 87
6.5 For 1D linear advection equation, **Left**: Comparison of the first Neural Network Shifted-Proper Orthogonal Decomposition (NNsPOD) mode, obtained via neural-network shifted augmented manifold transformation, with all POD modes of the full-order model solutions. and **Right**: Singular value decay comparison, showing Proper Orthogonal Decomposition (POD) (without transformations) and NNsPOD (with the neural-network shifted augmented manifold transformation) 87
6.6 Prediction of NNsPOD-RBF-ROM (considered first NNsPOD mode with 99.99% cumulative energy) for test parameters, labelled as Predict (solid lines), compared with the full-order solutions of 1D linear advection equation, labelled as Truth (dashed lines). . 88

6.7 Relative L^2 reconstruction error of predictions obtained by NNsPOD-RBF-ROM for the 1D linear advection equation, shown for training set parameters (left) and test set parameters (right). The red line on the left indicates the mean training error, while the red line on the right represents the mean test error, demonstrating that the resulting Reduced Order Model (ROM) exhibits excellent generalization capabilities. 89
6.8 Mean L^2 relative reconstruction error comparison of POD-RBF-ROM and NNsPOD-RBF-ROM predictions for test set parameters of 1D linear advection equation. 89
6.9 FOM solution of isentropic convective vortex at time, $t \in \{0.625, 25, 62.5\}$ and for FOM pre-processing, the original snapshot at t = 0.625 (left) is considered as reference configuration. 91
6.10 Shifted snapshots (top) of isentropic convective vortex after neural-network shift augmented transformations and the respective original snapshot (bottom) at time, $t \in \{0.625, 25, 62.5\}$ from the training set. 91
6.11 **Left**: Comparison of singular values decay of isentropic convective vortex without the transformation, is labelled as POD and after the neural-network shifted augmented manifold transformation, is labelled as NNsPOD, and **Right**: Showing first two modes of FOM manifold without the transformations (first row - POD modes) and after the neural-network shift augmented manifold transformation (second row - NNsPOD modes). . . . 91
6.12 Predictions of snapshots by NNsPOD-RBF-ROM of isentropic convective vortex test set parameters, showing for time instances $t = \{9.375, 25.625, 58.75\}$s. 92
6.13 Relative L^2 reconstruction error of predictions by NNsPOD-RBF-ROM for Isentropic convective vortex, for both training set parameters (left) and test set parameters (right). 93
6.14 Predicted snapshots by the POD-RBF-ROM for test set parameters of the isentropic convective vortex, considering 10 POD modes, shown for time instances $t = \{4.375, 24.375, 61.75\}$ s. . . 94
6.15 Mean L^2 relative reconstruction error comparison of POD-RBF-ROM and NNsPOD-RBF-ROM predictions for test set parameters of 2D isentropic convective vortex. 94

6.16 Discretized domain used for the two-phase flow test case. The inlet and outlet boundaries are highlighted in green and red, respectively. 96
6.17 Velocity magnitude contour of FOM solutions for the two-phase flow at times t ∈ {10, 20, 40}. We highlight that fluid enters only through the inlet, while outflow through the outlet is restricted. 97
6.18 FOM solution of two-phase flow at the time, t ∈ $\{0.05, 20, 40\}$ and for pre-processing of the solution manifold, the original snapshot at t = 40 (right figure) is considered as reference configuration. 97
6.19 Shifted snapshots (top) of two-phase flow, after neural network shift augmented manifold transformations and the respective original snapshot (bottom) at a time, t = $\{0.05, 21.3, 40\}$ from the training set. 97
6.20 Comparison of singular values decay of two-phase flow FOM manifold without pre-processing, is labelled as POD, and after the neural-network shifted augmented manifold transformation, is labelled as NNsPOD 98
6.21 Transformed snapshots (top) of two-phase flow, after transforming all the snapshots to the reference snapshot using interpolation to accelerate KnW decay and the respective original snapshot (bottom) from the training set at times, t = $\{0.05, 21.55, 40\}$ are shown. 98
6.22 **Left**: Comparison of singular values decay of two-phase flow case FOM manifold without transformations, is labelled as POD and after transforming all the snapshots to reference snapshot using interpolation, after learning shift operator, is labelled as TransformedPOD, and **Right**: First TransformedPOD mode obtained after this manifold transformation. 99

6.23 Predictions of snapshots for test set parameters obtained by TransformedPOD-RBF-ROM of the two-phase flow test case, showing predictions for parameters (time) t = $\{9.8, 20, 39.95\}$s. To construct this ROM, we use the transformed linear subspace obtained by transforming all the snapshots in the training set to the reference snapshot using interpolation, to accelerate KnW decay, and before that, we learn the shift operator. In this case, except for this transformation step in Algorithm 1, the rest of Algorithm 1 and Algorithm 2 remain the same for the construction of efficient and accurate ROMs. 100
6.24 Relative L^2 reconstruction error of predictions by TransformedPOD-RBF-ROM of the two-phase flow test case, for both training set parameters (left) and test set parameters (right). 100
6.25 Predicted snapshots for test set parameters by the POD-RBF-ROM for the two-phase flow test case, considering 10 POD modes, showing predictions for parameters (time) t = $\{3.4, 22.2, 39.9\}$s. 101
6.26 Mean L^2 relative reconstruction error of predictions by TransformedPOD-RBF-ROM of the two-phase flow test case, for both training set parameters (left) and test set parameters (right). 101

7.1 2D contour of the original snapshots of the 1D travelling waves. Here, the white line separates the first 50% dataset as the training set and the rest as the test set. 111
7.2 1D travelling waves training set registered at the reference configuration. 111
7.3 **Left:** Comparison of the singular values of registered and unregistered 1D travelling waves. **Right:** The first 10 POD modes obtained from the unregistered snapshots, alongside the 1st POD mode obtained via cross-correlation based snapshot registration. 112
7.4 Comparison of the predictions of the resulting ROM of the 1D travelling waves with the original snapshots as ground truth, here the white line separates the first 50% dataset as the training set and the remaining as the test set, and also shows the absolute difference between the prediction and the ground truth. 113
7.5 Relative L_2 predictions error by the ROM of the 1D travelling wave via cross-correlation based snapshots registration, for both training set and test set parameters. 113

7.6 Comparison of mean relative L_2 prediction error for 1D travelling wave test set parameters using POD-RBF-ROM, presented for the unregistered case (without registration) and the registered case (with registration via cross-correlation). 114
7.7 Comparison of the isentropic convective vortex registered snapshots with the reference and the corresponding original snapshot. 114
7.8 **Left**: Comparison of singular values decay of registered snapshots matrix, labelled as Registered and of the unregistered snapshots matrix as Unregistered. **Right**: Showing the first two modes of the registered snapshots (first row - Registered) and showing the first two modes of the unregistered snapshots matrix (second row - Unregistered). 115
7.9 Predicted snapshots using POD-RBF-ROM with cross-correlation based registration for the isentropic convective vortex test set parameters at time instances $t \in \{25, 45, 58.75\}$s. 115
7.10 Relative L_2 predictions error of the isentropic convective vortex ROM via cross-correlation based registrations, for both training and test set parameters. 116
7.11 Comparison of mean relative L_2 prediction error for 2D isentropic convective vortex test set parameters using POD-RBF-ROM, presented for the unregistered case (without registration) and the registered case (with registration via cross-correlation). 116

8.1 Comparison of RTD curves from the experiment, FOM CFD simulation, and projection-based ROM, illustrating the good agreement between the simulations and experimental data. 127
8.2 RTD curves for test parameter μ_5: comparison between the parameter-time dependent projection based ROM and full-order simulation results, showing excellent agreement within the interpolatory regime. 128
8.3 RTD curves for test parameter μ_6: comparison between the parameter-time dependent projection based ROM and full-order simulation results, showing excellent agreement within the interpolatory regime. 128
8.4 RTD curves for test parameter μ_{11} in the extrapolatory regime: comparison between the parameter-time dependent projection based ROM and full-order simulation results, ROM predictions accurately capturing the overall trend of the RTD curve, with a slight discrepancy in the maximum concentration prediction. . . 129

8.5 RTD curves for test parameter μ_5: comparison between the data-driven POD-RBF-ROM and full-order simulation, showing excellent agreement within the interpolatory regime 130
8.6 RTD curves for test parameter μ_6: comparison between the data-driven POD-RBF-ROM and full-order simulation, showing excellent agreement within the interpolatory regime 130
8.7 RTD curves for test parameter μ_{11} in the extrapolatory regime: comparison between the data-driven POD-RBF-ROM and full-order simulation, ROM predictions showing fluctuations in the initial phase but accurately capturing the overall trend of the RTD curve, with a slight discrepancy in the concentration prediction. 131

9.1 Schematic representation of an echo state network, a type of reservoir computing 139
9.2 Singular values and cumulative energy distribution of the training dataset used in developing the POD-RC-ROM for predicting intermixing time. 143
9.3 Prediction of intermixing time by the POD-RC-ROM for test set parameter in the extrapolation regime. 144
9.4 Prediction relative L_2 error of POD-RC-ROM for both train and test set parameters, again the red vertical line separates the training and test sets, with the left region representing the training set and the right region representing the test set. 145
9.5 Comparison of intermixing time obtained from the FOM and the POD-RC based ROM. The red vertical line separates the training and test sets, with the left region representing the training set and the right region representing the test set. 145

List of Tables

3.1 Input parameters and boundary conditions used for the isothermal and non-isothermal simulations. 41
3.2 Discretization schemes employed in the tundish steady-state operation study. 42
3.3 Inlet velocities for the parametric study of Residence Time Distribution (RTD) curves in a single-strand tundish. 46

4.1 Model coefficients of $k - \varepsilon$ turbulence model 53
4.2 Operating conditions of the tundish 55
4.3 Fluid properties used in the ladle changeover operation simulations. 55
4.4 Initial and boundary conditions considered for the three different stages of ladle changeover operations of a single-strand tundish. 56
4.5 Discretization schemes employed in the interFoam-based multiphase species transport solver for ladle changeover operation transient simulation. 57
4.6 Numerical experiment parameters considered to investigate the effects of ladle change time, ladle change time with corresponding filling time to reach steady-state operation. 60

6.1 Model coefficients of $k - \epsilon$ turbulence model. 95

A.1 ANNs hyperparameters for 1D travelling wave test case. 154
A.2 ANNs hyperparameters for 2D Isentropic Convective Vortex test case. 154
A.3 ANNs hyperparameters for 2D two-phase flow test case. 154
A.4 RC hyperparameters considered to construct predictive ROM for steel grade intermixing. 154

B.1 Summary of computational resources used for the FOM simulations to obtain the tundish steady-state operation QoIs on the Linux cluster. The wall-clock time refers to the elapsed real time, while the CPU time denotes the cumulative usage across all allocated cores (core-hours). These results correspond to the steady RANS + transient tracer simulations discussed in Table 3.3 of chapter 3. 155
B.2 Computational resources used for projection-based ROM construction and prediction to obtain the tundish steady-state operation quantity of interest (QoI). The offline phase includes computation of POD modes (whole domain is considered), assembling reduced operators, and solving the reduced-order Ordinary Differential Equation (ODE) to obtain the modal coefficients. The online phase refers to the prediction of the desired QoI at the tundish outlet. Here, online evaluation is performed only at the outlet probe location, where the QoI is recorded. Evaluating over the entire domain takes approximately 46 seconds, primarily due to the cost of the dot product operation. 156
B.3 Computational resources used for the parametric time-dependent data-driven POD-RBF-ROM construction and prediction to obtain the tundish steady-state operation QoIs. The offline phase includes computation of POD modes (whole domain is considered), evaluation of modal coefficients, and Radial Basis Function (RBF) regression map construction. The online phase corresponds to the prediction of quantities of interest, the Residence Time Distribution (RTD) curve at the tundish outlet. Again here, the online evaluation is performed only at the outlet probe location, where the QoI is recorded. Evaluating over the entire domain takes approximately 90 seconds, primarily due to the cost of the dot product operation. 156
B.4 Summary of computational resources used for the ladle changeover FOM transient simulations on the Linux cluster. Wall-clock time refers to the elapsed real time, and CPU time represents the cumulative time across all allocated cores (core-hours). These experiments correspond to those reported in Table 4.6 of chapter 4. 157

B.5 Computational resources used for the ladle changeover operation ROM construction and prediction. The offline phase includes the computation of POD modes (only the outlet region of the domain is considered), evaluation of modal coefficients, and the training of the reservoir computing model to construct a regression map. The online phase corresponds to the prediction of the quantities of interest at the tundish outlet. 157

List of Algorithms

1 Neural network shift augmented transformation 81
2 NNsPOD-ROM 83
3 Customized framework for the two-phase flow case. 99

4 Registration via cross-correlation 108
5 ROM via cross-correlation based snapshot registration 109

6 POD-Reservoir Computing based ROM 142

Acronyms

CC	Continuous Casting
SEN	Submerged Entry Nozzle
RTD	Residence Time Distribution
PDE	Partial Differential Equation
CFD	Computational Fluid Dynamics
FDM	Finite Difference Method
FVM	Finite Volume Method
FEM	Finite Element Method
DGM	Discontinuous Galerkin Method
SEM	Spectral Element Method
CV	Control Volume
NSE	Navier-Stokes equations
TVD	Total Variation Diminishing
FOM	Full Order Model
VOF	Volume of Fluid
ROM	Reduced Order Model
KnW	Kolmogorov n-Width
POD	Proper Orthogonal Decomposition
RB	Reduced Basis
SVD	Singular Value Decomposition
ODE	Ordinary Differential Equation
sPOD	Shifted-Proper Orthogonal Decomposition
NNsPOD	Neural Network Shifted-Proper Orthogonal Decomposition
MOR	Model Order Reduction
PMOR	Projection based Model Order Reduction
RBF	Radial Basis Function

ANNs	Artificial Neural Networks
RNNs	Recurrent Neural Networks
RC	Reservoir Computing
ESN	Echo State Network

Chapter 1

Introduction and Motivation

1.1 Continuous Casting

The Continuous Casting (CC) process is the predominant method used in steel production, accounting for 96% of the world's steel production, as per the World Steel Association 2024 report [209]. The first attempts at CC usage date back to the middle of the 19th century and significantly gained popularity in the 1950s [83].

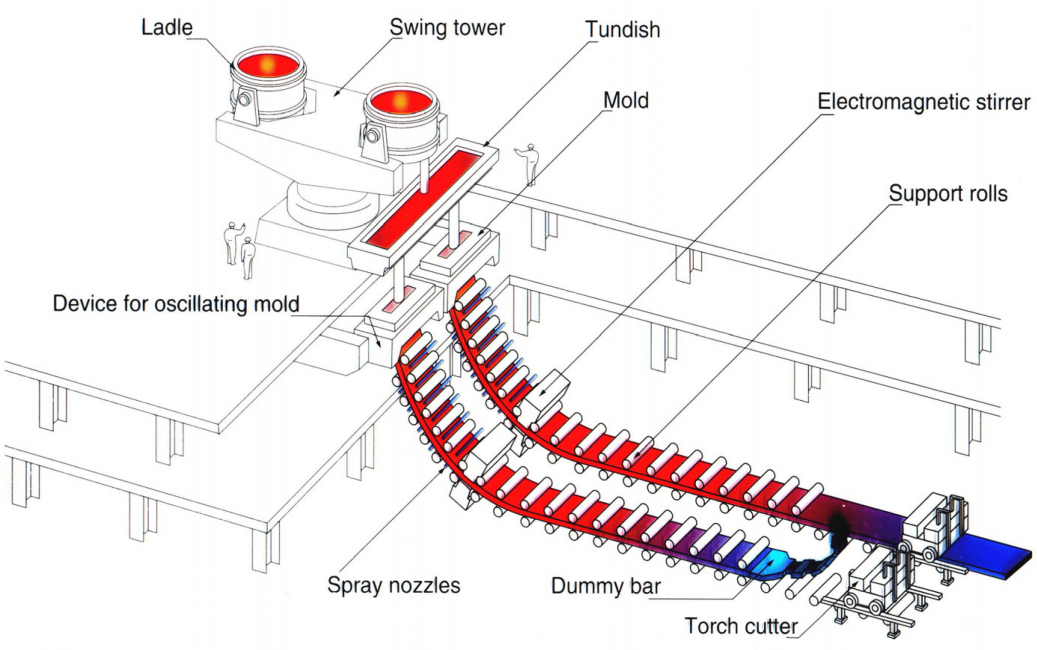


Figure 1.1: Schematic representation of continuous casting machine (reproduced from [67]), showing the vital components: Ladle, Tundish, and Mould. The ladle delivers the molten metal to the tundish, and then the tundish delivers molten metal to the mould with the desired throughput rate and temperature, through the Submerged Entry Nozzle. In the mould, the metal starts to solidify, and in the secondary cooling region, complete solidification of the metal is achieved.

The CC machine (shown in Figure 1.1) is now a mature technological process, that has gone through significant developments over decades, driven by the experience of the commercial operators, by the water model experiment setup for the fluid flow behaviour study and recently, by advancement in numerical simulations [190]. At present, it is the most efficient method for manufacturing semi-finished products like billets, slabs, and blooms. It is not just used to manufacture steel products, but also for aluminium, copper, non-ferrous metals like zinc and nickel, and special metals, such as titanium and magnesium.

As shown in Figure 1.1, the CC machine consists of several vital components: the ladle, tundish, mould, spray cooling nozzles, and rollers to facilitate cooling. At first, the molten metal is obtained using furnaces and is transferred to the ladle. The metal is drained from the ladle into the tundish, the intermediate vessel between the ladle and the mould. Then, the metal at the desired temperature and throughput rate is delivered to the mould through the Submerged Entry Nozzle (SEN). The metal starts to solidify in the mould, developing a solid exterior skin. The inner region of the slab still consists of molten metal; the complete solidification of the metal is achieved in the secondary cooling region, due to the effect of cooled rollers and direct spraying of water over the product. The casting is completed at the end of the tertiary cooling region and is followed by procedures like flattening, rolling, or extruding of the metal to have a desired shape.

1.2 Tundish

The tundish plays a critical role in the continuous casting process, and this component is the main focus of this work. In the CC machine, the tundish is an intermediate vessel that transfers molten metal evenly from the ladle to the moulds, with a desired throughput rate and temperature without limited inclusion contamination. It acts as a reservoir during the ladle change period and continues to supply molten metal to the moulds when the inflow of the melt is stopped, making it possible to perform sequential casting based on the availability of the ladles. The tundish plays a significant role in producing clean steel. The single strand tundish in Figure 1.2 shows the schematic of the flow physics inside the tundish, dealing with multiphase flows.

A tundish is a batch reactor, made of a refractory-lined channel (rectangular or square cross-section), consisting of one inlet, one or multiple outlets, and sometimes with flow modifiers like an impact pot as a turbulent inhibitor, dams,

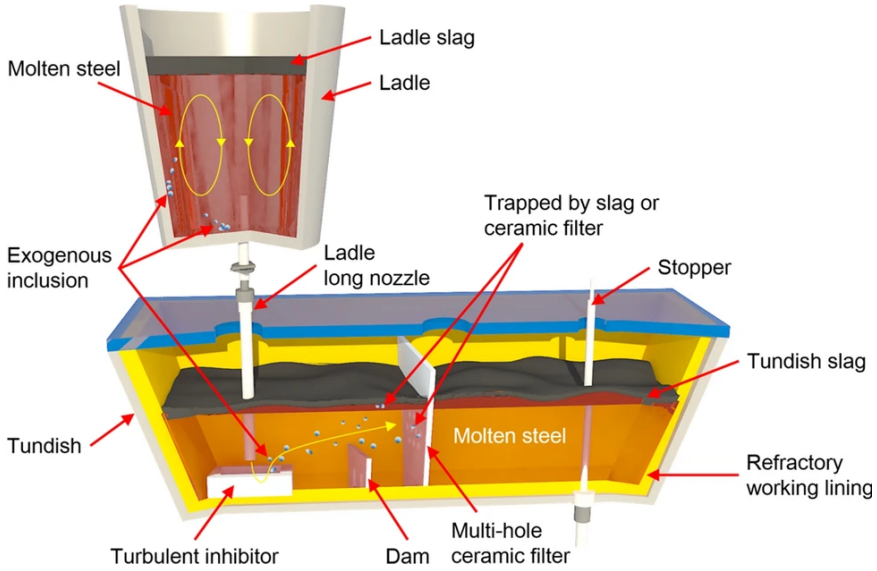


Figure 1.2: Schematic representation of single strand tundish (reproduced from [201]), showing the detailed flow physics and components of tundish, consisting of the fluid molten steel with the presence of slag and the inclusion flotation, dealing with multiphase flows. The flow modifiers (impact pod as a turbulent inhibitor and dam) that improve steel quality and ensure better castability and other components are labelled.

and ceramic filters placed along the cross-sectional length. It may consist of a refractory-lined lid and outlet ports assembled with stoppers or slide gates to facilitate the melt into the mould. These components of the tundish can be depicted in Figure 1.2. The proper design and operation of the tundish play a vital role in producing the desired steel composition and quality, resulting in clean steel production. In the last three decades of the 19th century, significant advancements in tundish technology have occurred for clean steel production. However, the advances in tundish technology have slowed down in recent years, as a result of becoming a mature technology. It is still of great concern for industries, because of significant interest in developing tundish metallurgy and strict demands regarding steel quality. Also, the concerns of cost, energy, and environmental impacts are becoming significantly important.

1.3 Physical problem

During the CC production cycle, the tundish undergoes one steady-state operation and three transient-state operations, which occur during the initial filling, ladle-change over and final emptying phase. It is mainly operated in steady-state conditions and the transient period could be a small fraction

of the steady-state duration, mainly depending on the ladle size and the sequence length. During both periods of operations, various phenomena of interest (chart shown in Figure 1.3) are of keen importance for steel producers and equipment manufacturers, as these phenomena have significant effects on steel quality.

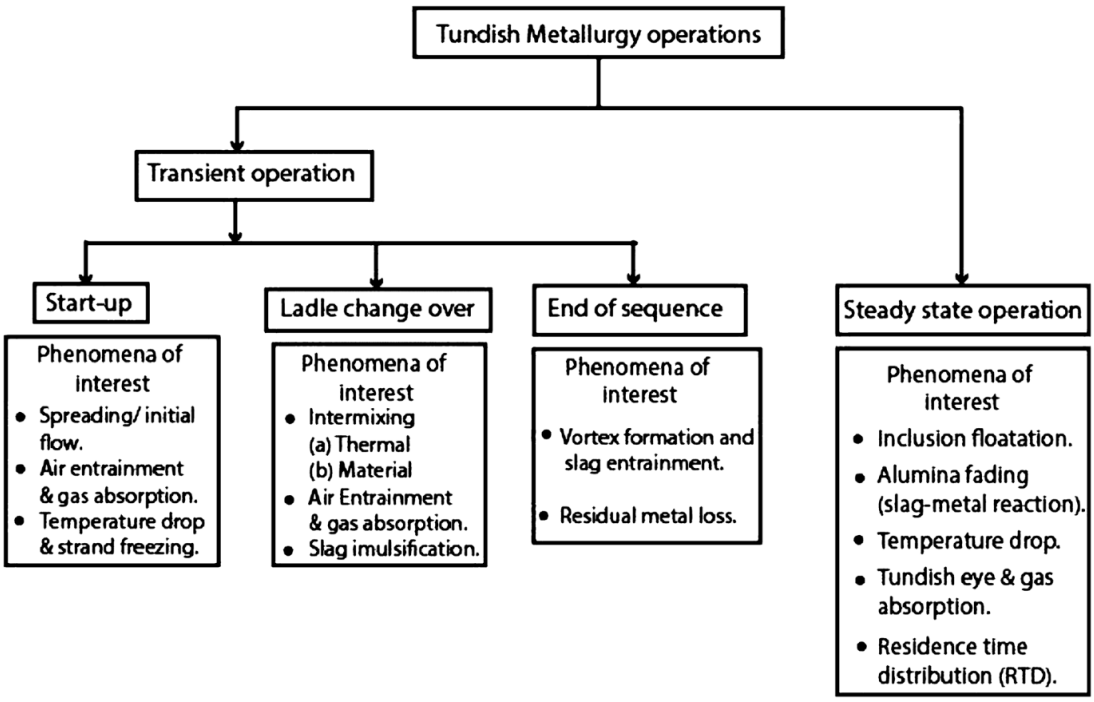


Figure 1.3: Summary of tundish operations, with the associated phenomena of each operation [121].

During start-up, a tundish preheated (approximately at 1250 °**C**) is used to transfer the molten steel from a ladle to a mould. Initially, the molten steel from the ladle is fed into the tundish smoothly at a greater rate to minimize the mixing of gas-liquid-slag, air entrainment, and gas absorption in the tundish. At this stage, the tundish outlet nozzle (SEN) is completely closed such that a sufficient amount of melt accumulates rapidly in the tundish before the beginning of casting. After reaching a sufficient threshold of molten steel level, the tundish stopper rods are lifted gradually, allowing the molten steel to enter the moulds. Soon, the inflow rate is modified to maintain a constant volume of molten steel (or working level) in the tundish.

Once the pre-determined melt working level is achieved following the start-up (initial filling), the tundish is mostly operated in a steady state by maintaining constant operational parameters such as inflow rate, throughput rate, liquid steel volume, etc. The phenomena of interest during steady-state operation are

inclusion flotation, temperature drop, slag-metal reactions, tundish-eye formation, and mainly the Residence Time Distribution (RTD).

Again, the transient operations occur during ladle changeover and end of sequence. Once the molten steel or the melt in the ladle is exhausted, to maintain sequential casting, it is replaced with a new ladle. During the ladle changeover, the inflow is stopped for a period, and as a result, the melt level drops down, as there is a constant throughput rate to continue casting. As the new ladle delivers melt into a tundish, it can lead to severe thermal and material mixing between melt-slag and the ambient air, and also can lead to air entrainment and slag emulsification. Residual metal losses and slag entrainment become important issues during the ladle changeover and end-of-sequence operations.

The aforementioned discussion explains that tundish metallurgical operations are characterized by multiphase flow, heat and mass transfer, turbulence, and chemical reactions between different phases (molten metal, slag, refractory material, and ambient air). The accurate modelling of these operations is generally difficult and has to deal with several unwarranted complications, such as nozzle clogging, refraction and gunning material erosion, and operations fluctuations and disturbances.

1.4 Specific problem

1.4.1 Steady state operation

As mentioned earlier, the tundish typically operates under steady-state conditions. Serving as the last metallurgical vessel in the continuous casting process, the tundish plays a crucial role in modifying the content of non-metallic inclusions in the molten steel [213]. Therefore, analyzing and optimizing the fluid flow within the tundish is essential for maintaining and improving product quality.

A common technique for quantitatively evaluating the flow behaviour of molten steel within a tundish is the stimulus–response method [204]. This method involves introducing a tracer into the incoming steel stream as the stimulus or input. Among the different types of tracer inputs, step and pulse inputs are the most frequently used. The resulting output is the tracer concentration over time at the tundish exit, known as the Residence Time Distribution (RTD) curve [168].

As outlined earlier, the main aim of this study is to determine the RTD curve for a continuous casting tundish and utilize it for analyzing volume partitioning. The entire RTD analysis workflow is illustrated in Figure 1.4.

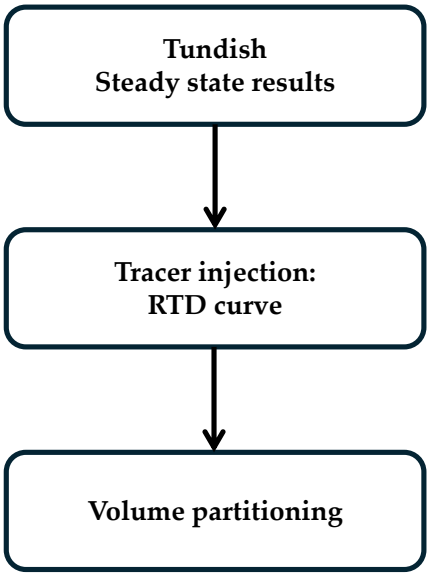


Figure 1.4: Schematics of the complete workflow of a continuous casting tundish steady-state operation RTD analysis.

First, the steady-state flow of the fluid (molten steel or water) is attained for the given initial and boundary conditions. Depending on the fluid considered, the flow may be isothermal (in the case of water) or non-isothermal (for molten steel). A tracer is then injected into the tundish, and its concentration is monitored at the outlet. As reported earlier, the RTD curve is derived from the tracer concentration-time profile recorded at the outlet of the tundish. Next to perform RTD analysis for volume partitioning, the widely used model for the single strand tundish is a combined model proposed by Sahai and Emi [167]. According to the theory, the tundish can be divided into three distinct regions: plug flow, fully mixed flow, and dead flow. The combined plug flow and fully mixed flow region is referred to as the active zone. A tundish can be considered as a combination of different idealized reactors. Sahai and Ahuja [166] revised their model, presenting an enhanced combined model, which is based on a concentration-time curve resulting from an impulsive tracer injection at the inlet, which is as follows:

$$\frac{V_{\text{dead}}}{V} = 1 - \frac{T_{\text{avg}}}{T_{\text{th}}}, \tag{1.1}$$

$$\frac{V_{\text{plug}}}{V} = \frac{\frac{1}{2}(T_{\text{min}} + T_{\text{max}})}{T_{\text{th}}} = \frac{T_{\text{plug}}}{T_{\text{th}}}, \tag{1.2}$$

$$\frac{V_{\text{mix}}}{V} = 1 - \frac{V_{\text{dead}}}{V} - \frac{V_{\text{plug}}}{V}, \tag{1.3}$$

$$T_{\text{th}} = \frac{V}{Q}. \tag{1.4}$$

Where T_{avg} is the average residence time, defined as the time required for half of the tracer to exit the tundish; T_{th}, the theoretical residence time (1.4) is defined as the ratio of the tundish volume V and the volumetric flow rate Q; T_{min} is the time at which the tracer concentration first reaches 1×10^{-4} at the outlet; and T_{max} is the time at which highest concentration is observed.

The dead volume fraction (1.1) is the region of the tundish in which molten steel remains for an extended duration, typically exceeding twice the theoretical residence time. The plug volume fraction (1.2) is the ratio between the average time delay to reach the maximum concentration at the outlet of the tundish and the theoretical residence time, and it's the fraction of active volume where the molten steel flows in a stable manner. The mixing volume fraction (1.3) is the fraction of the active tundish volume where steel is homogenized.

In this study, we focus on mathematical modelling and water-model experiments of a single-strand tundish to analyze RTD characteristics. The numerical investigation involves simulating three-dimensional, steady-state, turbulent flow under both isothermal (water as working fluid) and non-isothermal (molten steel as working fluid) conditions. Subsequently, the transient evolution of tracer concentration is modelled to obtain the key quantity of interest (QoI), the RTD curve. Further, we perform RTD analysis for volume partitioning, which provides insights into the flow behaviour within the tundish.

1.4.2 Ladle changeover operation

In this work, we are particularly focusing on transient operations occurring during the ladle changeover. Once the molten steel or the melt in the ladle is exhausted, it is replaced with a new ladle to maintain sequential casting. During this transition, the inflow ceases for a period, and as a result, the melt level drops as there is a constant throughput rate required for continuous casting. As the new ladle delivers melt into a tundish, it can lead to severe thermal and

material mixing between melt, slag, and the ambient air, which can lead to air entrainment and slag emulsification.

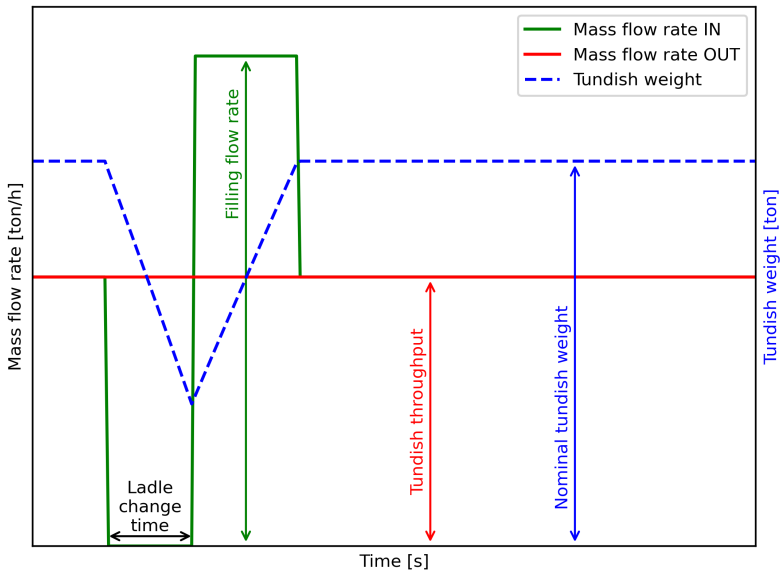


Figure 1.5: Process diagram of the ladle changeover operation.

Figure 1.5 illustrates the different stages of the ladle change operation. Throughout this operation, the volume inside the tundish changes continuously, and at any instant of time, the working volume of a tundish is occupied by two or more phases in variable proportions, changing with time. The ladle changeover and the resulting grade intermixing can be categorized into three distinct yet interconnected stages of tundish operation:

Stage 1: Steady-state operation before ladle changeover.

Stage 2: Transient emptying phase with outflow but no inflow.

Stage 3: Transient refilling phase with a new ladle containing a new steel grade.

The duration of Stage 1 varies based on the ladle size and casting rate and can extend for an hour or longer. In contrast, Stages 2 and 3 together typically last no more than 10–15 minutes before the process stabilizes and returns to steady-state operation.

Tundish metallurgical operations primarily involve multiphase flow, heat and mass transfer, turbulence, and chemical interactions between various phases, including molten metal, slag, refractory material, and ambient air. Accurately modelling these processes is complex and presents several challenges.

In this study, we focus on modelling a single-strand tundish. For the numerical investigation, we simulate three-dimensional, transient, two-phase, isothermal, and turbulent flow dynamics. The key quantity of interest (QoI) in this process is steel grade intermixing, particularly the new steel grade transition at the outlet of the tundish. To simulate these phenomena sequentially, the following assumptions have been made:

- The tundish is modelled as a two-phase flow system, considering molten steel and slag as the two immiscible fluids with fixed physical properties, while the presence of air is neglected.
- The fluid flow is assumed to be isothermal.
- Flow phenomena within the ladle shroud are assumed to be homogeneous.
- There are no chemical interactions between the phases within the molten bath.
- Only half of the tundish is modelled, taking advantage of its geometric symmetry.

1.5 Objectives

Over the past few decades, numerous studies have focused on the physical and mathematical modelling of tundish operations [122, 31, 121]. This area remains of significant interest to the steel industry due to the continuous advancement of tundish metallurgy and increasingly stringent steel quality requirements.

A tundish typically undergoes one steady-state operation and three transient-state operations, with steady-state operation being the most prevalent during casting. The primary objective of this work is the physical and mathematical modelling of tundish operations, particularly the steady-state operation and transient phenomena associated with the ladle changeover operations.

Despite significant research, reliable methods for obtaining key Quantities of Interest (QoIs) in real-time remain limited. Therefore, the goal of the present study is to develop mathematical tools capable of accurately predicting these QoIs for both steady-state and transient tundish operations, with potential applications in real-time decision-making processes.

To achieve this goal, the following key objectives are set: physical and mathematical modelling of tundish flow behaviour of both the steady-state and the

ladle changeover transient operations; development of mathematical frameworks for the reduction of computational cost, enabling near real-time applicability; and validation of the developed mathematical framework, with experimental data when available.

1.6 Thesis contribution

- To obtain the QoIs of the steady-state operations, both projection-based model order reduction and data-driven ROM strategies have been explored. Projection based ROM developed to evaluate the RTD curve is validated against full-order simulations and experimental results, demonstrating its effectiveness as an efficient surrogate model for industrial-scale applications. However, its practical application depends on access to the CFD solver's source code, an important limitation in industrial settings where commercial software is commonly used. To address this, a data-driven ROM is proposed as an alternative compatible with commercial solvers, yielding excellent accuracy across parametric variations, making it well-suited for industrial applications.
- Industrial-scale simulations are often constrained by the high computational cost of full-order models and with very long time integrations, restricting efficient parameter exploration and design optimization. To overcome this challenge, a POD–Reservoir Computing (POD-RC) based ROM is developed to predict grade intermixing and evaluate intermixing time during ladle changeover operations. By integrating Reservoir Computing into the POD framework, the model captures parameter-dependent temporal dynamics with limited training data. The POD-RC-ROM achieves competitive accuracy with significantly fewer training samples and lower computational demands compared to conventional neural networks. With only minor deviations from the full order model results, the POD-RC-ROM demonstrates a reliable, cost-effective, time-efficient, practical solution well-suited for industrial-scale applications.
- Advection-dominated flows are prevalent in engineering systems and industrial processes, posing significant challenges for reduced order modelling due to the slow Kolmogorov n-Width (KnW) decay. To address this, two registration-inspired methodologies are proposed to develop

efficient and accurate ROMs using linear methods. First, the NNsPOD-ROM framework, utilizing neural-network shift augmented manifold transformation techniques that employ automatic shift detection. Second, a cross-correlation based snapshots registration method. Both frameworks offer complete offline-online stages and do not require a priori knowledge of the underlying advection mechanisms. Their effectiveness is demonstrated on several benchmark problems, highlighting their potential as robust, accurate ROM solutions for advection-dominated systems.

Publications and conferences

The work carried out during this thesis has led to the following article, currently available on arXiv and submitted for journal publication:

- Non-intrusive model reduction of advection-dominated hyperbolic problems using neural network shift augmented manifold transformation. H. Gowrachari, N. Demo, G. Stabile, and G. Rozza. [63] (*In review, Computers & Fluids*)
- Model reduction for transport-dominated problems via cross-correlation based snapshot registration. H. Gowrachari, G. Stabile, and G. Rozza. [62] (*Accepted, to appear in SMLET24 special volume in the SEMA-SIMAI series*)
- Reservoir computing based predictive reduced order model for steel grade intermixing in a continuous casting tundish. H. Gowrachari, M. Barra, G. Stabile, G. Bazzaro, and G. Rozza. (*In preparation*)
- Projection based model order reduction for residence time distribution analysis of continuous casting tundish. H. Gowrachari, M. Barra, M. Khamlich, G. Stabile, G. Bazzaro, and G. Rozza. (*In preparation*)

The results of this work were presented at the following international conferences, workshops, and seminars:

- 3rd IACM Digital Twins in Engineering Conference & 1st AICOMAS 2025, Paris, France, Feb 17–21, 2025
  - Talk title: *Digital twin technologies for steelmaking continuous casting tundish.*

- Cyber-physical Systems and Digital Twins for the Energy-Intensive Industries Workshop[1], organized by CYPHER COST Action CA22151 at CERTH, Thessaloniki, Greece, Sept 9–13, 2024
  - Poster: *Reduced order method for multi-phase system of continuous casting tundish.*
- 9th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2024), Lisboa, Portugal, June 3–7, 2024
  - Talk title: *ROM for interface-capturing of two-phase flow using neural-network shift augmented manifold transformation.*
- Analysis Junior Seminars (AJS) 2023–2024, SISSA, Trieste, Italy, April 12, 2024
  - Talk title: *Non-linear transformation techniques for model reduction of advection-dominated problems.*
- SIAM Conference on Uncertainty Quantification (UQ24), Trieste, Italy, Feb 27 – Mar 1, 2024
  - Poster at UQ24 Satellite Event *Power of Diversity in UQ*:
    Data-driven reduced order model for advection-dominated problems using neural network shifted proper orthogonal decomposition.
- MORTech 2023 – 6th International Workshop on Model Reduction Techniques, ENS Paris-Saclay, France, Nov 22–24, 2023
  - Poster: *Data-driven reduced order model for advection-dominated problems using neural network shifted proper orthogonal decomposition.*

1.7 Thesis outline

This thesis is divided into two main parts: Full Order Model (FOM) and Reduced Order Model (ROM).

Part I focuses on FOMs for the simulation of tundish metallurgical flows. These models are crucial for capturing the key flow phenomena associated with steady-state and ladle changeover operations in continuous casting (CC) tundishes, and they serve as the foundation for subsequent reduced order modelling approaches.

[1]HG acknowledges COST Action STSMs travel grant

Chapter 2 – Discretization method: This chapter establishes the numerical foundation for the full-order simulations. It provides a brief introduction to the Finite Volume Method (FVM) used for spatial discretization, covering the interpolation of convective and diffusive fluxes, gradient computation, and time integration schemes. The emphasis is on ensuring numerical accuracy and stability in capturing complex flow and transport phenomena, especially in the context of turbulent and multiphase flows relevant to tundish operations.

Chapter 3 – Tundish steady-state operation: This chapter applies the FOM framework to simulate steady-state tundish operation. It begins with a description of the experimental setup used for validation, followed by the formulation of the FOM, including governing equations, turbulence modelling, the Boussinesq approximation, and species transport. The computational setup, boundary conditions, and numerical methods are then detailed. Results are analyzed in terms of flow patterns and RTD distribution, offering key insights into tundish performance during regular casting.

Chapter 4 – Tundish ladle changeover operation: This chapter extends the FOM to capture the transient dynamics during ladle changeover, a critical event that affects steel quality due to grade intermixing. The model incorporates free surface tracking using the Volume of Fluid (VOF) method, along with turbulence models and scalar transport equations. It details the operational boundary conditions that reflect the real-world casting process. The results illustrate the effects of ladle change timing on mixing behaviour and provide valuable data for process optimization. These high-fidelity simulations also serve as the basis for validating reduced order models developed in the second part of the thesis.

Part II is devoted to the development and application of ROM methodologies for complex dynamical systems. As high-fidelity simulations of such systems are often computationally expensive, ROMs provide an efficient alternative by approximating the full-order models while preserving essential dynamical features. This part begins with a comprehensive introduction to the theoretical foundations of model order reduction, followed by a series of novel methodologies that leverage both classical and data-driven approaches.

Chapter 5 - Preliminaries on model order reduction: This lays the groundwork by reviewing the fundamental concepts of model order reduction. It explores the approximability of solution manifolds and distinguishes between linear and non-linear dimensionality reduction techniques. It also addresses

the challenges posed by the curse of dimensionality and slow KnW, establishing the motivation for more advanced methods. The chapter concludes with an overview of the general pipeline for constructing ROMs.

Chapter 6 – ROM Using neural network shift-augmented manifold transformation: This chapter presents a neural network shift-augmented manifold transformation approach for efficient ROM development for advection-dominated problems. By integrating POD with shift-augmented transformations learned by neural networks, the method aims to overcome the slow decay of the KnW in advection-dominated hyperbolic problems. Its performance is evaluated on several benchmark problems, including 1D advection, 2D isentropic convective vortex, and two-phase flow scenarios inspired by tundish flow physics.

Chapter 7 - ROM via cross-correlation based snapshots registration: This chapter introduces a reduced order modelling strategy based on cross-correlation based snapshot registration. The technique offers a robust approach to aligning solution snapshots to overcome the slow decay of KnW. The method's capability to capture travelling phenomena efficiently is demonstrated through applications to both 1D travelling wave and 2D isentropic convective vortex problems.

Chapter 8 - Model order reduction strategies for tundish RTD analysis: This chapter investigates both projection-based model order reduction and data-driven ROM techniques applied to RTD analysis in tundish systems. For parameter-time dependent scenarios, the standard Galerkin projection-based model reduction method is utilized to efficiently compute RTD while maintaining the interpretability of the reduced dynamics. In scenarios where solver code is inaccessible, commonly due to the use of commercial software in industries, a data-driven ROM approach is adopted to overcome these limitations.

Chapter 9 - ROM for grade intermixing in a continuous casting tundish: This chapter proposes a data-driven ROM approach using reservoir computing, specifically echo state networks, combined with POD. The resulting POD-RC-ROM framework is applied to the prediction of steel grade intermixing time during ladle changeover operations, demonstrating strong potential for time-series forecasting in industrial process applications.

Chapter 10 – Conclusion: This chapter concludes the thesis by summarizing the key findings and contributions from both parts. Part I focused on FOM

of tundish operations, providing high-fidelity simulations to analyze steady-state and transient ladle changeover phenomena. Part II explored various ROM strategies, evaluating their trade-offs, strengths, and limitations across multiple flow scenarios. The chapter also reflects on the broader implications of combining FOM and ROM approaches for industrial process optimization and outlines potential directions for future work.

Figure 1.6 illustrates the structure and key contributions of the thesis, and provides a schematic overview of the modelling approaches and their interconnections.

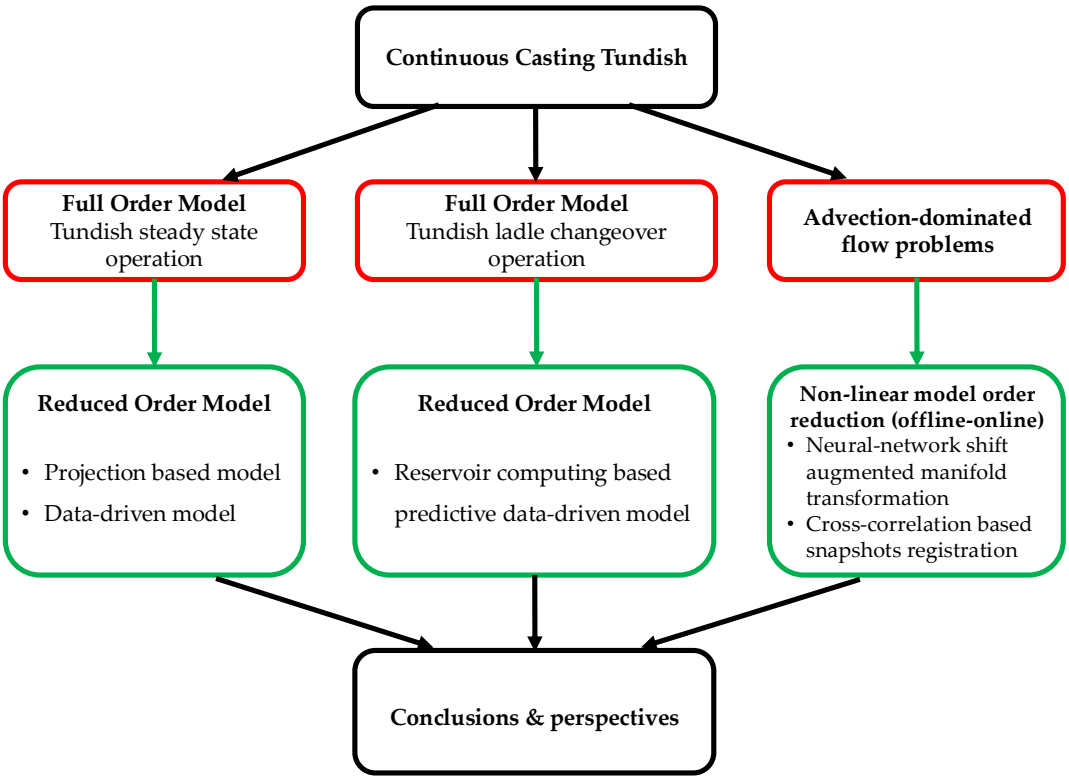


Figure 1.6: Thesis overview: The first part focuses on the Full Order Model (FOM) of tundish operations, and the second part focuses on non-linear model reduction and reduced order modelling strategies to efficiently obtain the key quantities of interest (QoIs) in tundish operations.

Part I

Full Order Model

Chapter 2

Discretization method

2.1 Introduction

Tundish metallurgical operations are governed by complex physical phenomena, including multiphase flow, species transport, heat transfer, buoyancy-driven flow, and turbulence. Mathematical modelling of these operations requires the coupling of various physical models to accurately represent the underlying processes. Once an appropriate mathematical model is established, consisting of a system of Partial Differential Equations (PDEs) that govern the tundish flow physics, the next step is to choose a suitable discretization method. This method approximates the Partial Differential Equations (PDEs) by converting it into a system of algebraic equations, which can be solved for the variables at discrete spatial and temporal points. In Computational Fluid Dynamics (CFD), there are various discretization methods, the most important and popular ones are:

- Finite Difference Method $\rightarrow$ FDM
- Finite Volume Method $\rightarrow$ FVM
- Finite Element Method $\rightarrow$ FEM

The Finite Volume Method (FVM) has gained significant popularity in Computational Fluid Dynamics (CFD) due to its flexibility as a discretization technique [54, 196, 17, 126]. Although, the Finite Difference Method (FDM) [38] and Finite Element Method (FEM) [39, 113, 132] were developed and utilized earlier, the FVM has become more favoured because of its adaptability. Most commercial CFD codes, as well as many open-source platforms like OpenFOAM, are based on the FVM.

The method's popularity can be attributed to its versatility with arbitrary Control Volumes (CVs), ease of implementation, and strong conservation properties, which ensure that conservation is maintained in each mesh cell and throughout the entire computational domain. The FVM became especially prominent for simulating fluid flow and related transport phenomena due to the pioneering efforts of the CFD research group at Imperial College in the early 1970s, led by Spalding [105]. Key contributors to this development included Patankar [184], Gasman [57], Wolfshtein [123], and Runchal [164], among others [163, 162].

2.2 Finite Volume Method

In FVM, the discretization is carried out directly in the physical space. The solution domain is divided into a finite number of control volumes, and the governing equations are applied to each CV. The FVM utilizes the integral form of the governing equations as the initial starting point, the governing equations are integrated over the finite volumes that constitute the discretized domain. Subsequently, the Gauss theorem is applied to reformulate the volume integrals of the convection and diffusion terms into equivalent surface integrals. Thereafter, the surface and volume integrals are converted into discrete forms and integrated numerically using integration points. To illustrate this approach, we consider the conservation equation for a general scalar variable ϕ [126], expressed as:

$$\underbrace{\frac{\partial \rho\phi}{\partial t}}_{\text{transient term}} + \underbrace{\nabla\cdot(\rho\mathbf{u}\phi)}_{\text{convective term}} = \underbrace{\nabla\cdot\left(\Gamma_\phi\nabla\phi\right)}_{\text{diffusive term}} + \underbrace{S_\phi(\phi)}_{\text{source term}} . \tag{2.1}$$

It should be noted that by considering the correct values or variables for ϕ, Γ^ϕ and S_ϕ in (2.1), it can represent any conservation equations [126]. For example, the Navier-Stokes equations (NSE) can be expressed from (2.1). Now let us write the above equation by integrating over the element P shown in Figure 2.1, transforming the (2.1) to integral form shown in (2.2), which is the starting point of FVM.

$$\int_{V_P}\frac{\partial \rho\phi}{\partial t}dV + \int_{V_P}\nabla\cdot(\rho\mathbf{u}\phi)dV = \int_{V_P}\nabla\cdot\left(\Gamma_\phi\nabla\phi\right)dV + \int_{V_P}S_\phi(\phi)dV. \tag{2.2}$$

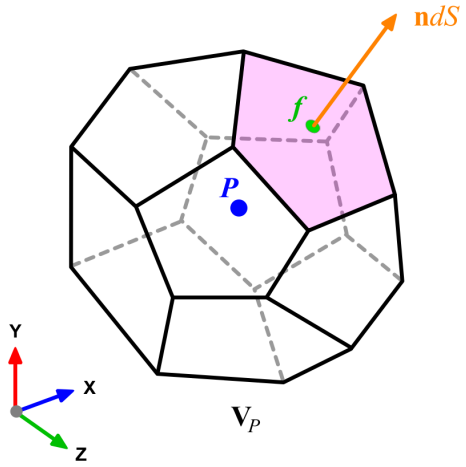


Figure 2.1: Showing the Control Volume (CV) V_P of the element P, where ∂V_P is the closed surface bounding the control volume V_P, dS represents an infinitesimal surface element with its associated outward-pointing normal vector $\mathbf{n}$ and the product $\mathbf{n}dS$ is represented as $d\mathbf{S}$ (reproduced from [66]).

Before proceeding to the next steps, let us recall the Divergence theorem, also known as the Gauss theorem or Ostrogradsky theorem.

Divergence theorem

Let us consider Control Volume (CV) V_P in a three-dimensional space shown in Figure 2.1, with dS representing the surface element with its associated normal vector $\mathbf{n}$ pointing outwards. For any vector field v defined on V_P, the divergence theorem states that

$$\int_{V_P} \nabla \cdot v dV = \oint d\mathbf{S} \cdot v. \tag{2.3}$$

The divergence theorem implies that the total flux of a vector field across a closed surface is equivalent to the volume integral of the field's divergence over the region enclosed by the surface. This is a significant theorem for fluid dynamics and a fundamental theorem utilized in FVM to convert volume integrals into surface integrals.

Now, applying this divergence theorem on (2.2), replacing the volume integrals of the convective and diffusive terms with surface integrals, gives us

$$\frac{\partial}{\partial t}\int_{V_P} \rho\phi dV + \oint_{\partial V_P} \underbrace{d\mathbf{S} \cdot (\rho \mathbf{u}\phi)}_{\text{convective flux}} = \oint_{\partial V_P} \underbrace{d\mathbf{S} \cdot (\Gamma_\phi \nabla\phi)}_{\text{diffusive flux}} + \int_{V_P} S_\phi(\phi) dV. \tag{2.4}$$

To further approximate each term, the integral over a closed surface is expressed as the sum of the integrals over its faces. Each integral is then approximated using the midpoint rule, which is second-order accurate. All major commercial codes and OpenFOAM use the midpoint rule to approximate integrals [54]. As a result, the convective and diffusive fluxes become

$$\oint_{\partial V_P} \underbrace{d\mathbf{S} \cdot (\rho \mathbf{u} \phi)}_{\text{convective flux}} = \sum_f \int_f d\mathbf{S} \cdot (\rho \mathbf{u} \phi)_f \approx \sum_f \mathbf{S}_f \cdot (\overline{\rho \mathbf{u} \phi})_f = \sum_f \mathbf{S}_f \cdot (\rho \mathbf{u} \phi)_f. \quad (2.5)$$

$$\oint_{\partial V_P} \underbrace{d\mathbf{S} \cdot (\Gamma_\phi \nabla \phi)}_{\text{diffusive flux}} = \sum_f \int_f d\mathbf{S} \cdot (\Gamma_\phi \nabla \phi)_f \approx \sum_f \mathbf{S}_f \cdot (\overline{\Gamma_\phi \nabla \phi})_f = \sum_f \mathbf{S}_f \cdot (\Gamma_\phi \nabla \phi)_f. \quad (2.6)$$

This source term S_ϕ approximation is exact when S_ϕ remains constant or varies linearly within the control volume. Otherwise, it is second-order accurate.

$$\int_{V_P} S_\phi(\phi) dV = S_c V_P + S_p V_P \phi_P, \quad (2.7)$$

Here, S_c represents the constant component of the source term, while S_p corresponds to the non-linear part. s We derive the following semi-discrete equation by applying the previous equations to the general transport equation across all control volumes.

$$\int_{V_P} \underbrace{\frac{\partial \rho \phi}{\partial t} dV}_{\text{Time derivative}} + \sum_f \underbrace{\mathbf{S}_f \cdot (\rho \mathbf{u} \phi)_f}_{\text{Convective flux}} = \sum_f \underbrace{\mathbf{S}_f \cdot (\Gamma_\phi \nabla \phi)_f}_{\text{Diffusive flux}} + \underbrace{(S_c V_P + S_p V_P \phi_P)}_{\text{Source term}}, \quad (2.8)$$

where, the convective flux $F^C = \mathbf{S} \cdot (\rho \mathbf{u} \phi)$ and the diffusive flux $F^D = \mathbf{S} \cdot (\Gamma_\phi \nabla \phi)$. All variables are computed and stored at the centroid of the control volumes, and the face values in the convective and diffusive fluxes are obtained through interpolation from the centroid values of the adjacent control volumes.

2.2.1 Convective flux interpolation

The convective differencing schemes are used to evaluate ϕ values at the face from the cell centres. There are various schemes available for determining ϕ and its gradient at the cell face. A review of the performance of numerous schemes in the context of FVM is presented in [206]. Here, we discuss some of the most commonly used schemes available in commercial CFD codes and open-source code OpenFOAM.

Upwind differencing

In upwind differencing scheme, the face values are computed according to the direction of the mass flow:

$$\phi_f = \begin{cases} \phi_f = \phi_P & \text{if} \quad F \geq 0 \quad \text{mass flow going out of the cell,} \\ \phi_f = \phi_N & \text{if} \quad F < 0 \quad \text{mass flow into the cell.} \end{cases} \tag{2.9}$$

This is first-order accurate, and this scheme unconditionally satisfies bound-

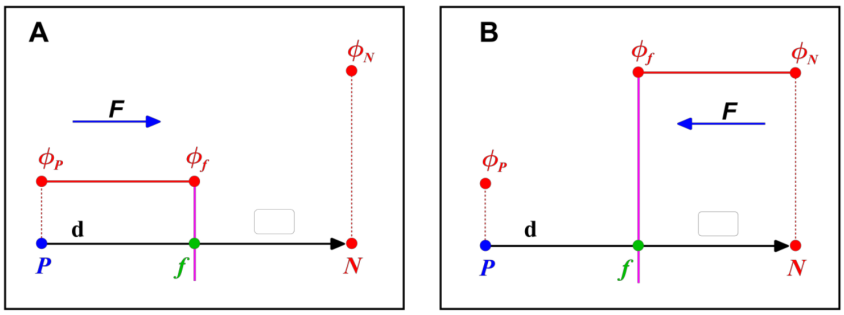


Figure 2.2: Upwind differencing stencil (reproduced from [66]).

edness, yielding non-oscillatory solutions. However, this is achieved at the cost of numerical diffusion.

Linear/central differencing

Another straightforward approach to evaluating value at the CV face center is linear interpolation between the two nearest cell centroids/nodes (P & N). Assuming the linear variation of ϕ between P and N (Figure 2.3), the face value is computed as follows:

$$\phi_f = \lambda_f \phi_p + (1 - \lambda_f)\phi_N. \tag{2.10}$$

Where the interpolation factor λ_f is defined as:

$$\lambda_f = \frac{fN}{PN} = \frac{|x_f - x_N|}{|x_N - x_P|}. \tag{2.11}$$

This differencing scheme is called central differencing scheme. It is a second-order accurate, but unstable scheme. This may generate oscillatory solutions (unbounded solutions).

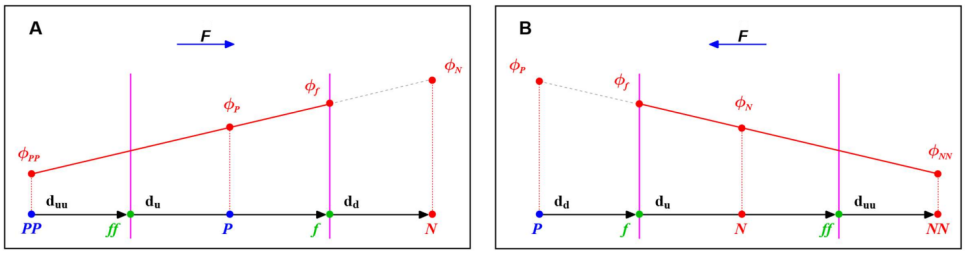


Figure 2.3: Illustration of face interpolation used to compute face values for the convective flux in the finite volume method (taken from [66]).

Linear upwind differencing

In linear upwind differencing scheme, the face values appearing in the convective flux term is computed as follows:

$$\phi_f = \begin{cases} \phi_P + \frac{1}{2}(\phi_P - \phi_{PP}) & \text{if} \quad F \geq 0, \\ \phi_N + \frac{1}{2}(\phi_N - \phi_{NN}) & \text{if} \quad F < 0. \end{cases} \tag{2.12}$$

This interpolation scheme is also known as second order upwind differencing or Beam-Warming scheme and is second order accurate. This scheme yields oscillatory (unbounded) solutions for highly convective flows or in the presence of a strong gradient.

TVD scheme

To prevent oscillations in the linear upwind differencing scheme, a limiter function $\psi(r)$, also referred to as flux or gradient limiter is incorporated.

$$\phi_f = \begin{cases} \phi_P + \frac{1}{2}\psi_f^-(\phi_P - \phi_{PP}) & \text{if} \quad F \geq 0, \\ \phi_N + \frac{1}{2}\psi_f^+(\phi_N - \phi_{NN}) & \text{if} \quad F < 0. \end{cases} \tag{2.13}$$

When the limiter detects strong gradients, it switches locally to low resolution (upwind). The concept of the limiter function is based on monitoring the ratio of successive gradients.

$$r_f^- = \frac{\phi_N - \phi_P}{\phi_P - \phi_{PP}} \quad \text{and} \quad r_f^+ = \frac{\phi_P - \phi_N}{\phi_N - \phi_{NN}}. \tag{2.14}$$

By incorporating a well-designed limiting function $\psi(r)$ results in a high-resolution (second order accurate) and bounded scheme. This is named as Total Variation Diminishing (TVD) scheme. The selection of the limiter functions determines

both the order of the accuracy and its boundedness. Limiters inherently reduce the scheme's accuracy to first order (low resolution) in regions with sharp gradients, opposing slopes, or zero gradients. However, this compromise is necessary to prevent oscillations. Different limiters have distinct switching behaviours and are chosen based on the specific problem and solution schemes. No single limiter performs optimally for all cases, so the selection is typically made through a trial-and-error approach.

All the high-order (HO) and high-resolution (HR) schemes discussed so far assume line structure, meaning they are designed for structured, orthogonal meshes, as shown in Figure 2.4 **A** (Left). In such meshes, the cell centers *PP*, *P*, and *N* are collinear, making the construction of numerical stencils relatively simple. However, in unstructured meshes or cases where the cell centers are not collinear, applying these schemes becomes more complex, as the cell center *PP* is not aligned with the vector connecting cells *P* and *N*, as shown in Figure 2.4 **B** (Right).

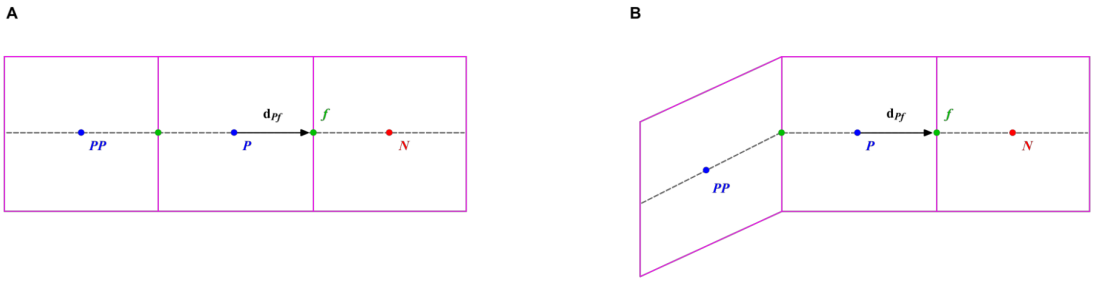


Figure 2.4: Comparison of mesh alignment in higher-order schemes: (A) structured mesh with aligned cell centers PP, P, and N; (B) unstructured mesh where PP is not aligned with the vector between P and N, complicating the use of standard interpolation schemes (reproduced from [66]).

One way to address this problem is by redefining higher-order schemes using the gradients at the control volume *P*. For unstructured mesh, the face values can be computed using upwind-biased formulations by utilizing the gradients at the cell center *P* and the face center *f*.

Upwind $\rightarrow \quad \phi_f = \phi_P$,
Linear/central difference $\rightarrow \phi_f = \phi_P + \nabla\phi_f \cdot \mathbf{d}_{Pf}$,
Linear upwind differencing $\rightarrow \phi_f = \phi_P + \left(2\nabla\phi_P - \nabla\phi_f\right) \cdot \mathbf{d}_{Pf}$. In this revised formulation, the cell *PP* is no longer present. The problem now shifts to accurately computing the gradients at both cell and face centers.

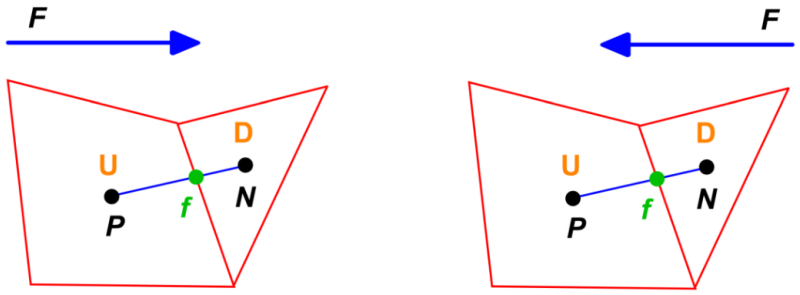


Figure 2.5: Face value interpolation in unstructured meshes (reproduced from [66]).

In unstructured meshes (Figure 2.5), face centered values are determined using the following flux-limited scheme,

$$\phi_f = \phi_U + \frac{1}{2}\psi(r_f^{\pm})\,(\phi_D - \phi_U)\,. \tag{2.15}$$

Notably, the formulation of this scheme remains the same as when cell centers are collinear. The key difference lies in how the ratio of successive gradients is computed. One approach to calculating this ratio in unstructured meshes is based on the formulation presented in [46].

$$r_f = \frac{(2\nabla\phi_P \cdot \mathbf{d}_{PN})}{\phi_N - \phi_P} - 1. \tag{2.16}$$

The limiter function $\psi(r)$ remains the same for a structured mesh.

2.2.2 Gradient computation

There are various methods available for computing cell-centered gradients, such as least squares, Gauss cell-based, and Gauss node-based approaches. Using the Gauss cell-based method, the cell-centered gradients are evaluated as follows:

$$(\nabla\phi)_P = \frac{1}{V_P}\sum_f (\mathbf{S}_f\phi_f)\,. \tag{2.17}$$

This method is second-order accurate and is known as Gauss cell-based. In OpenFOAM, gradients are computed using either the Gauss or least-squares method.

Reconstructing gradients at face centers: A common approach to reconstructing face gradients is to use weighted interpolation of the cell-centered values. However, mesh non-orthogonality and skewness can introduce errors in the

face gradient approximation, requiring corrections. This process is iterative, where progressively more accurate gradient approximations are computed, starting from an initial approximation.

$$\phi_f = \lambda_f \phi_p + (1 - \lambda_f)\phi_N \quad \text{where} \quad \lambda_f = fN/PN. \tag{2.18}$$

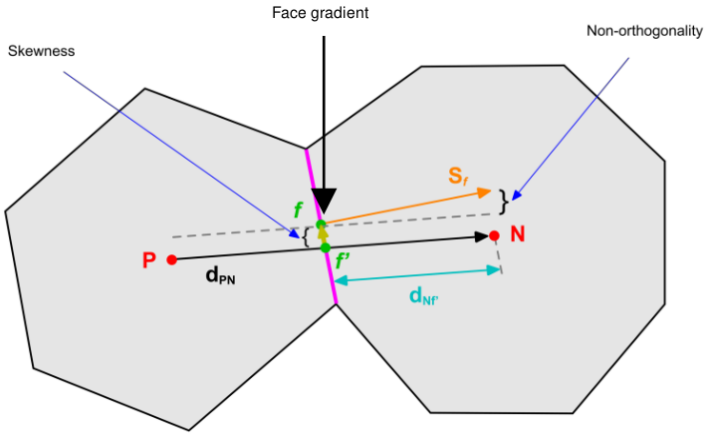


Figure 2.6: Face gradient interpolation (reproduced from [66]).

2.2.3 Diffusive flux interpolation

In an orthogonal mesh (as shown on the Left of Figure 2.7), the vectors **d** and **S** are parallel, and the following expression is used:

$$\mathbf{S} \cdot (\nabla\phi)_f = |\mathbf{S}| \frac{\phi_N - \phi_P}{|\mathbf{d}|}. \tag{2.19}$$

The face gradient can be computed using the two values surrounding the face. This is a central difference approximation for the first-order derivative. Alternatively, follow the previous subsection 2.2.2 steps for gradient computations. Both these methods are second-order accurate.

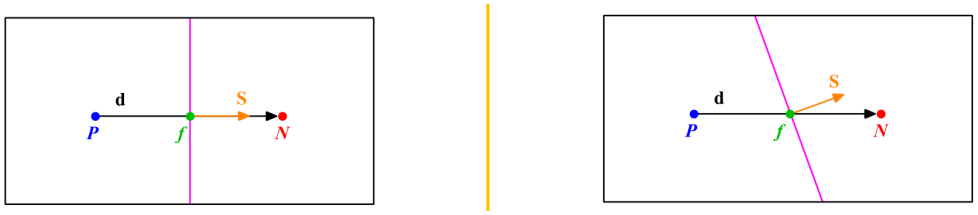


Figure 2.7: orthogonal and non orthogonal mesh (reproduced from [66]).

For non-orthogonal mesh (as shown on the Right of Figure 2.7), the product $\mathbf{S} \cdot (\nabla\phi)_f$ is divided into two terms:

$$\mathbf{S} \cdot (\nabla\phi)_f = \underbrace{|\Delta_\perp| \cdot (\nabla\phi)_f}_{\text{orthogonal contribution}} + \underbrace{\mathbf{k} \cdot (\nabla\phi)_f}_{\text{non-orthogonal contribution}} . \tag{2.20}$$

The two vectors introduced above, $\Delta_\perp$ and $\mathbf{k}$, must satisfy the following condition:

$$\mathbf{S} = \Delta_\perp + \mathbf{k}. \tag{2.21}$$

The vector Δ is selected to be parallel to $\mathbf{d}$, enabling the use of Equation 2.19 for the orthogonal contribution, while the less accurate method is applied solely to the non-orthogonal part. There is extensive literature on the correction of diffusive fluxes in non-orthogonal meshes. Jasak PhD thesis [89] offers a geometric interpretation of various correction approaches found in the literature, categorizing them into three main types:

- Minimum correction approach,
- Orthogonal correction approach,
- Over-relaxed approach.

For a review of various methods and a derivation of the algebra behind the previously mentioned geometric interpretations, the interested reader can refer to [49]. Among these approaches, the over-relaxed method is the most commonly used.

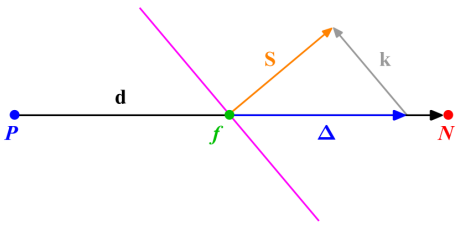


Figure 2.8: Non-orthogonality treatment in the over-relaxed approach (reproduced from [66]).

In over-relaxed approach, the diffusive flux is corrected as

$$\Delta_\perp = \frac{\mathbf{d}}{\mathrm{d} \cdot \mathrm{S}} |\mathbf{S}|^2, \tag{2.22}$$

$$\mathbf{S} \cdot (\nabla \phi)_f = \underbrace{|\Delta_\perp| \frac{\phi_N - \phi_P}{|\mathbf{d}|}}_{\text{orthogonal contribution}} + \underbrace{\mathbf{k} \cdot (\nabla \phi)_f}_{\text{non-orthogonal contribution}} . \tag{2.23}$$

The secondary face gradients, representing the non-orthogonal contribution in the discretization of the diffusive flux on non-orthogonal meshes, must be reconstructed from the cell center to the face center. This can be carried out following the gradient computation procedure mentioned in subsection 2.2.2.

2.3 Time discretization

The previous section introduced the discretization of spatial terms, which can be divided into two components: the conversion of surface and volume integrals into discrete sums and the formulation of expressions that determine the face values of the variable based on cell values. Now, we proceed with the time evaluation of the semi-discrete general transport equation.

$$\begin{aligned} \int_t^{t+\Delta t} \left[\left(\frac{\partial \rho \phi}{\partial t} \right)_P V_P + \sum_f \mathbf{S}_f \cdot (\rho \mathbf{u} \phi)_f - \sum_f \mathbf{S}_f \cdot \left(\Gamma_\phi \nabla \phi \right)_f \right] dt \\ = \int_t^{t+\Delta t} \left(S_c V_P + S_p V_P \phi_P \right) dt. \end{aligned} \tag{2.24}$$

At this point, any time discretization scheme, such as Crank-Nicolson, Euler implicit, forward Euler, or backward differencing, can be employed. It is important to note that the order of discretization for the transient term does not have to be the same as that for the spatial terms. Each term can be discretized separately, allowing for different levels of accuracy. The overall solution will exhibit second-order accuracy if each term is at least second-order accurate.

Chapter 3

Tundish steady state operation

3.1 Introduction

The tundish plays a critical role in the continuous casting process. It is a refractory-lined metallurgical vessel through which molten steel passes before entering the mold for solidification. Initially, the tundish was primarily viewed as a simple buffer reservoir between the ladle and the mold, employed to bridge the discontinuous process of secondary metallurgy (e.g., degassing, de-oxidation, alloying, temperature control, and cleanliness adjustment) with the continuous casting process, and to distribute steel uniformly to multiple outlets [161]. Over time, the paradigm has shifted, and the tundish is now regarded as a true metallurgical reactor, playing a very crucial role in ensuring the quality of the cast product.

The growing demand for high-quality steel has heightened awareness among steel makers regarding product cleanliness requirements [212]. Steel cleanliness is primarily determined by the concentration of non-metallic inclusion particles that can be either endogenous or exogenous in origin. If not effectively removed during processing, these inclusions can significantly degrade the quality of the final steel product [191]. The composition, size, shape, quantity, and distribution of non-metallic inclusions significantly influence steel performance, affecting properties such as strength, toughness, fatigue resistance, and corrosion resistance [157]. The methods for minimizing non-metallic inclusions primarily involve controlling the composition of steel, slag, and refractory materials, as well as regulating fluid flow patterns within each vessel during refining and continuous casting of molten steel [213].

Since the tundish is the final stage in the continuous casting process, where the content of non-metallic inclusions in the steel melt can be substantially

modified [213], investigating and optimizing the fluid flow within this vessel is critical to ensuring product quality.

A widely used method for quantitatively analyzing molten steel flow in a tundish is the stimulus–response technique [204]. In this approach, the stimulus or input involves introducing a tracer material into the steel stream entering the tundish. Although various tracer input types may be applied, step and pulse inputs are the most commonly employed. The response, or output, is the concentration–time profile measured at the tundish outlet [168], commonly referred to as the Residence Time Distribution (RTD) curve.

The stimulus–response technique is suitable for analyzing both single-strand (single outlet) and multi-strand (multiple outlets) tundishes. In the case of multi-strand tundishes, the overall RTD curve is determined by calculating the weighted average of the individual RTD curves obtained at each outlet. The weighting factors correspond to the flow rates through each outlet, which may differ due to variations in flow distribution.

The RTD approach provides valuable insights into the flow characteristics of molten steel within the tundish. One of the key indicators it reveals is the presence or absence of flow shortcuts, the preferential paths through which steel reaches the outlet with minimal residence time. Steel that passes through preferential paths retains a high concentration of non-metallic inclusions, as the low residence time prevents sufficient flotation and removal of impurities. The presence of such flow patterns adversely affects steel cleanliness by promoting the discharge of inclusions from the tundish. In RTD curves, these shortcuts are typically indicated by multiple early peaks in concentration or by a dominant peak occurring at very low residence times.

Another key parameter derived from RTD analysis is the dead volume fraction within the tundish. Dead volume refers to regions where the steel residence time exceeds twice the theoretical residence time, which is defined as the ratio of tundish volume to the volumetric flow rate at the inlet. These zones are characterized by low velocity, often exhibiting recirculating flow patterns. A high dead volume fraction can lead to several operational issues, including steel freezing (partial solidification) due to prolonged residence time, and reduced inclusion removal efficiency, as inclusions tend to become trapped within these stagnant regions.

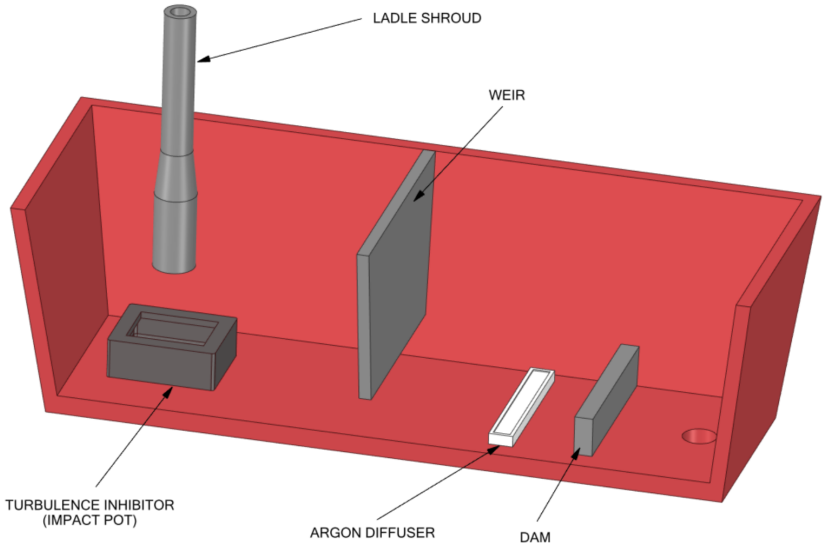


Figure 3.1: Schematic representation of a single-strand tundish with flow modifiers.

A detailed analysis of the RTD curve provides valuable insights into the effectiveness of tundish design. If necessary, it also offers guidance on modifying the tundish design to improve the steel flow pattern. Various flow control devices, such as turbulence inhibitors (impact pots), weirs, dams, and argon diffusers, are commonly located within the tundish volume to enhance tundish performance. Figure 3.1 presents a simplified schematic of a single-strand tundish, illustrating various flow modifiers that could be placed inside the tundish volume.

RTD curves for tundishes can be obtained through both numerical and physical simulations. Physical experiments are typically carried out using full-scale or scaled-down water models, where molten steel flow is simulated with water, and a conductive NaCl solution is commonly used as a tracer [168, 52]. The physical properties of the tracer, along with the injection method, can significantly influence the experimental results [112, 43, 15]. Factors such as tracer concentration, density, quantity, and type all affect the accuracy and reliability of the RTD measurements. While lower tracer concentrations and smaller quantities are generally preferred to minimize flow disturbance, the sensitivity of the concentration measurement devices imposes a practical lower limit on the tracer dosage [112, 15, 33].

Accurate simulation results, including the RTD curve, are contingent upon the proper selection of the turbulence model used to characterize the liquid metal flow. Model verification is typically conducted by comparing simulation results with data obtained from water model experiments [112]. The test facility

used for the water model experiment, located at the Danieli Research Center, is discussed in the later section 3.2.

3.2 Experiment

The experimental setup used to measure the RTD curve of the tundish is illustrated in Figure 3.2. The experimental setup consists of a large storage tank that supplies water to the system. Two variable-speed pumps draw water from the tank and deliver it to the tundish inlet at the desired flow rate. The outlet flow is gravity-driven and regulated by two control valves installed along the outlet pipeline, which returns the water to the storage tank. The test facility is equipped with a control system that enables precise regulation of the water flow rate and maintenance of a constant water level in the tundish model. It also facilitates the controlled injection of tracer material into the water flow. The tracer used in the experiments consists of a salt solution prepared by dis-

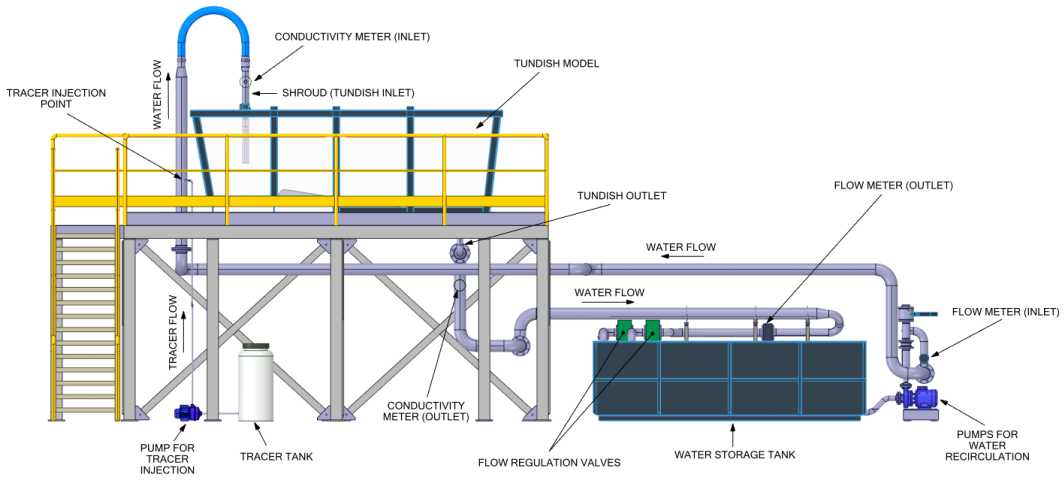


Figure 3.2: Test facility dedicated to water model experiments on tundish models. The test facility depicted is located at the Danieli Research Center in Buttrio, Province of Udine, Italy.

solving 10 kg of NaCl in 200 liters of industrial water. This solution is stored in a dedicated tank positioned beneath the tundish structure. A dedicated pump ensures the injection of the tracer into the system. The stimulus–response technique is employed in the experiment. The tracer is injected through the shroud, and its concentration at the inlet and outlet is continuously monitored by two conductivity meters.

The experiments are carried out on a full-scale Plexiglas model of the actual tundish. By employing a unitary scaling factor, both Froude and Reynolds number similarities are preserved. Although the Reynolds number does not match exactly, due to a slight difference (within 10%) [168] between the kinematic viscosities of water at room temperature and molten steel. However, this deviation is considered negligible for practical purposes. As a result, the Reynolds similarity criterion is reasonably satisfied. The use of a full-scale model ensures both geometric and dynamic similarity with the actual tundish, allowing the volumetric flow rates in the water model to correspond directly to those in the real plant casting process with molten steel.

Once the operational parameters (water flow rate and water level within the tundish) are set and the pumps are activated, a waiting period is required before commencing the test. This allows the system to reach steady-state conditions and permits the dissipation of transient effects arising from the tundish filling process. Following this stabilization period, tracer injection is initiated for 30 seconds. During the test, the data acquisition system continuously monitors the tracer concentration at both the inlet and outlet. The test is considered complete when the conductivity meter readings at both locations stabilize and approximate a steady state, typically occurring 40 to 50 minutes after tracer injection.

To obtain the real RTD curve of the tundish, which represents the system's impulse response, the tracer must be circulated only once, as opposed to continuously, as is done in the closed-loop water circuit of the facility. Figure 3.3 shows an example of the concentration-time curves measured at both the inlet and outlet of a single-strand tundish tested in the water model facility, highlighting the effect of tracer recirculation, which leads to an increase in tracer concentration at the inlet. Starting with the RTD curve measured at the water model facility, the true RTD curve of the tundish can be reconstructed through a deconvolution process. For a linear time-invariant system (LTI), the system's response $y(t)$ to a generic input $x(t)$ is represented by the convolution integral of the impulse response $g(t)$ and the input $x(t)$:

$$y(t) = \int_0^t g(\tau)x(t-\tau)\,d\tau. \tag{3.1}$$

The discretized form of the convolution integral is given by:

$$y(t) \cong \sum_0^n g(\tau_i)x(t-\tau_i)\,\Delta\tau. \tag{3.2}$$

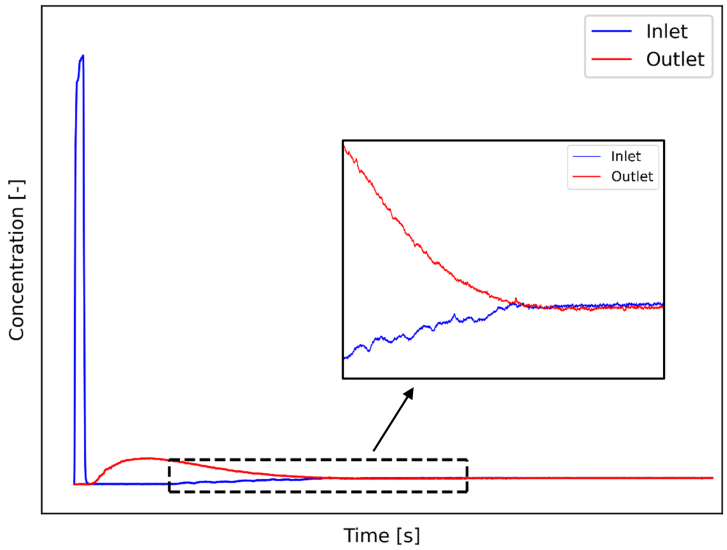


Figure 3.3: Concentration-time curves measured at the inlet and outlet of a single-strand tundish, highlighting the effect of the recirculation of the tracer inside the system.

In this context, $y(t)$ represents the signal from the sensor measuring tracer concentration at the tundish outlet, while $x(t)$ represents the signal from the sensor at the inlet. The function $g(t)$ denotes the impulse response of the system, which corresponds to the true RTD curve of the tundish. By numerically solving (3.2) for $g(t)$ (via the deconvolution process), the impulse response of the tundish can be reconstructed, thereby obtaining its RTD curve. Figure 3.4 illustrates an example of the application of the deconvolution process to reconstruct the actual RTD curve of the tundish.

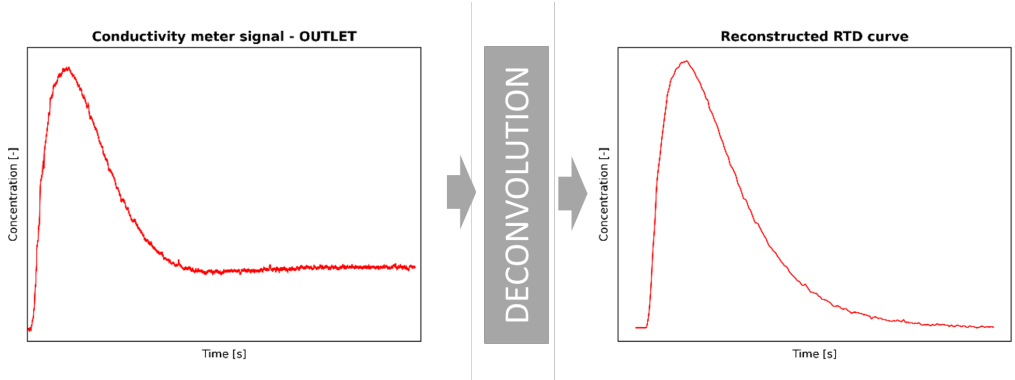


Figure 3.4: Deconvolution process to obtain the actual RTD curve of the tundish

3.3 Full order model

3.3.1 Governing equations

The isothermal fluid flow within the tundish is modelled by numerically solving the continuity and momentum equations under turbulent flow conditions. The governing equations, expressed in Cartesian coordinates, are as follows:

Continuity equation:

$$\frac{\partial(\rho u_i)}{\partial x_i} = 0, \tag{3.3}$$

Momentum equation:

$$\rho \frac{\partial u_i u_j}{\partial x_j} = \frac{\partial}{\partial x_i}\left[\mu_{\text{eff}}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] - \frac{\partial P}{\partial x_i} + f, \tag{3.4}$$

$$\mu_{\text{eff}} = \mu + \mu_{\text{t}} = \mu + \rho c_\mu \frac{k^2}{\varepsilon}, \tag{3.5}$$

Here, the body force is represented by f, the effective viscosity coefficient, μ_{eff}, primarily depends on the turbulence parameters k and ε, which are determined by solving the following two transport equations turbulence model.

3.3.2 Turbulence model

The eddy viscosity based, the Standard $k-\varepsilon$ two transport equations turbulence model [104] is considered:

$$\rho u_i \frac{\partial k}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu_{\text{eff}}}{\sigma_k}\frac{\partial k}{\partial x_i}\right) + \mu_{\text{t}}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \rho\varepsilon, \tag{3.6}$$

$$\rho u_j \frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\mu_{\text{eff}}}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_j}\right) + C_1\mu_{\text{t}}\frac{\varepsilon}{K}\frac{\partial u_j}{\partial x_i}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - C_2\frac{\varepsilon}{K}\rho\varepsilon. \tag{3.7}$$

In the equations above, the subscripts i and j represent the three spatial coordinate directions (x, y, and z), respectively. The values of the empirical constants used in the turbulence model are: $\sigma_k = 1.00$, $\sigma_\varepsilon = 1.30$, $C_1 = 1.44$, $C_2 = 1.92$, $C_\mu = 0.09$, and $\text{Pr}_t = 0.90$.

3.3.3 Boussinesq approximation

To model a non-isothermal fluid flow in the tundish, the following momentum equation is considered:

$$\rho\left(\frac{\partial u_i u_j}{\partial x_j}\right) = \frac{\partial}{\partial x_i}\left[\mu_{\text{eff}}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] - \frac{\partial P}{\partial x_i} + \rho_{\text{ref}}\beta\Delta T g_i, \tag{3.8}$$

where the last term accounts for the density difference between the existing fluids in the tundish. The last term is derived using the Boussinesq approximation. This approach is commonly employed in natural convection flows to achieve faster convergence. Instead of modelling fluid density as a temperature-dependent variable throughout the domain, the Boussinesq model assumes constant density in all governing equations, except in the buoyancy term of the momentum equation, where density variation with temperature is retained.

$$(\rho - \rho_{\text{ref}})g_i = -\rho_{\text{ref}}\beta(\Delta T)g_i. \tag{3.9}$$

In this context, ρ_{ref} denotes the reference fluid density, and β represents the thermal expansion coefficient. The Boussinesq approximation is applicable when temperature variations are small, specifically when the product $\beta\Delta T \ll 1$. For water or molten steel, β is typically on the order of 10^{-4}, and even for temperature differences of 15-20 °C, $\beta\Delta T$ remains on the order of 10^{-3}, which satisfies the approximation's validity condition. The transient temperature field within the tundish is subsequently evaluated by solving the energy conservation equation.

$$\frac{\partial\left(\rho C_p T\right)}{\partial t} + \frac{\partial\left(\rho C_p T u_j\right)}{\partial x_j} = \frac{\partial}{\partial x_j}\left(k_{eff}\frac{\partial T}{\partial x_j}\right) \tag{3.10}$$

The temperature field is coupled with the velocity field via the buoyancy term in (3.8). In non-isothermal simulations, the tundish walls and the free surface were considered adiabatic, implying that no heat transfer occurred through these boundaries.

3.3.4 Species transport equation

Using the previously computed flow field, the transient evolution of tracer concentration (expressed in terms of mass fraction) is modelled by solving the

species transport equation.

$$\frac{\partial\left(\rho c_i\right)}{\partial t}+\frac{\partial\left(\rho u_i c_i\right)}{\partial x_i}=\frac{\partial}{\partial x_i}\left(\rho D_{\text{eff}}\frac{\partial c_i}{\partial x_i}\right), \tag{3.11}$$

$$D_{\text{eff}}=D_m+D_t=D_m+\frac{\mu_{\text{t}}}{\rho Pr_t}. \tag{3.12}$$

Here, the effective diffusivity D_{eff} is defined as the sum of molecular diffusivity D_m and turbulent diffusivity D_t and Pr_t is the turbulent Prandtl number.

3.4 Numerical setup

3.4.1 Computational domain

A three-dimensional computational domain was constructed based on a 1:1 scaled industrial tundish, as illustrated in Figure 3.5. The inlet, outlet, and remaining surfaces, all treated as walls, are highlighted.

The computational mesh was generated using `snappyHexMesh` in OpenFOAM. The final mesh consists of approximately 3.6 million cells, comprising primarily hexahedra (96.8%), with an additional 6 prism layers, pyramids, tetrahedra, and polyhedra near complex geometrical features, shown in Figure 3.6.

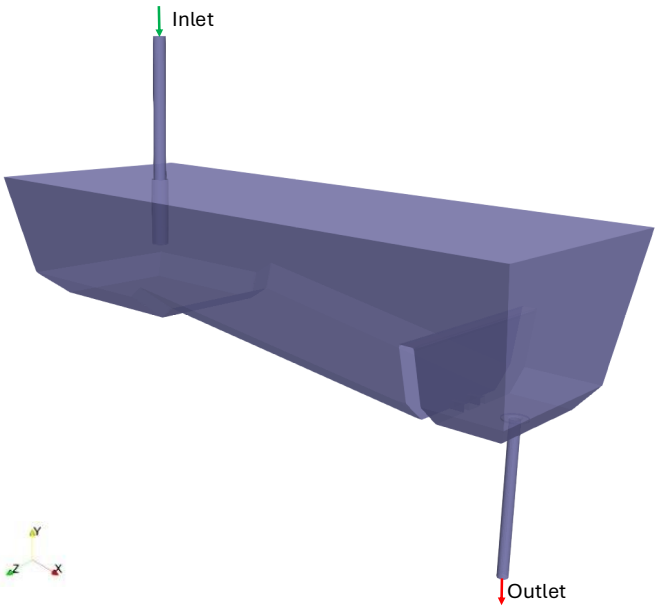


Figure 3.5: Computational domain of a 1:1 scaled single-strand tundish (isometric view); the inlet and outlet are indicated, and all other boundaries are treated as walls.

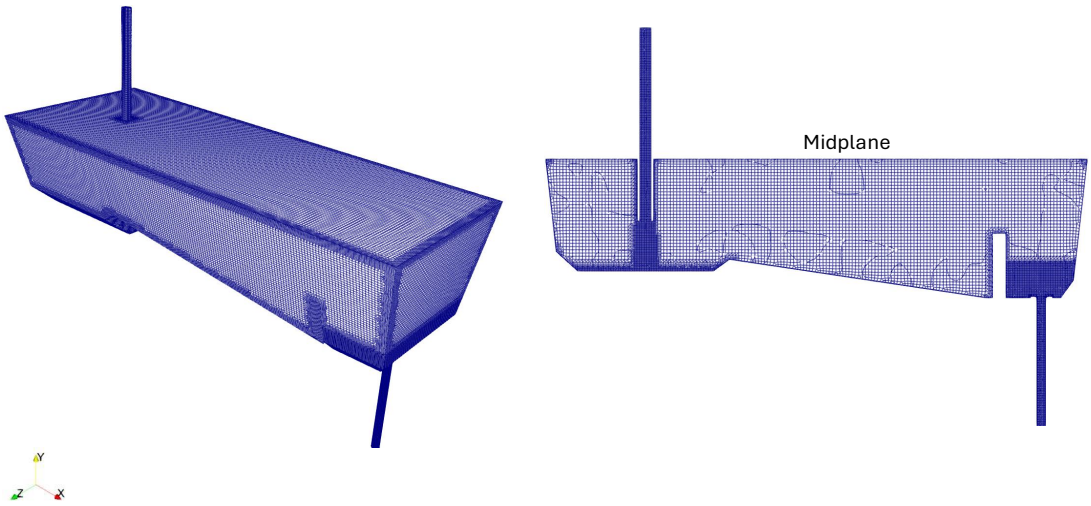


Figure 3.6: Discretized computational domain of a 1:1 scaled single-strand tundish. **Left**: Isometric view of the discretized domain; **Right**: 2D midplane showing mesh refinement at the inlet jet impingement region and outlet, and consists of six prism layers adjacent to the walls.

3.4.2 Boundary conditions

A constant volumetric flow rate is imposed at the inlet, an outflow boundary condition is applied at the outlet, and no-slip boundary condition is imposed on all walls; the near-wall region is modelled using standard wall functions. For non-isothermal simulation, where the working fluid is molten steel, the heat losses through the lateral and bottom walls are set to 6 kW m^{-2}. A summary of the input parameters and boundary conditions is provided in Table 3.1.

For tracer injection, the tracer concentration (mass fraction) at the inlet is set to unity for $0 \leq t \leq 2$ s, and to zero for $t > 2$ s, as shown in Figure 3.7. The tracer concentration at the outlet is monitored over $0 \leq t \leq 3000$ s to construct the RTD curves.

3.5 Numerical method

In this study, the OpenFOAM C++ library is utilized for the finite-volume discretization of partial differential equations. The object-oriented structure of C++ allows the code to closely mirror the mathematical formulation, while providing flexibility for further development and modification [207]. To model

Parameter	Water (Isothermal)	Molten steel (Non-isothermal)
Density	1000 kg/m^3	6900 kg/m^3
Kinematic viscosity	1×10^{-6} m^2/s	0.8×10^{-6} m^2/s
Reference pressure	101,325 Pa	101,325 Pa
Heat capacity	–	800 J/kg · K
Thermal conductivity	–	35 W/m · K
Thermal expansion coefficient	–	0.000127 K^{-1}
Inlet volumetric flow rate	0.0102657 m^3/s	0.0102657 m^3/s
Inlet velocity	1.6 m/s	1.6 m/s
Inlet temperature	–	1823.15 K
Wall (flow)	No-slip	No-slip
Wall (heat loss)	–	6 kW/m^2

Table 3.1: Input parameters and boundary conditions used for the isothermal and non-isothermal simulations.

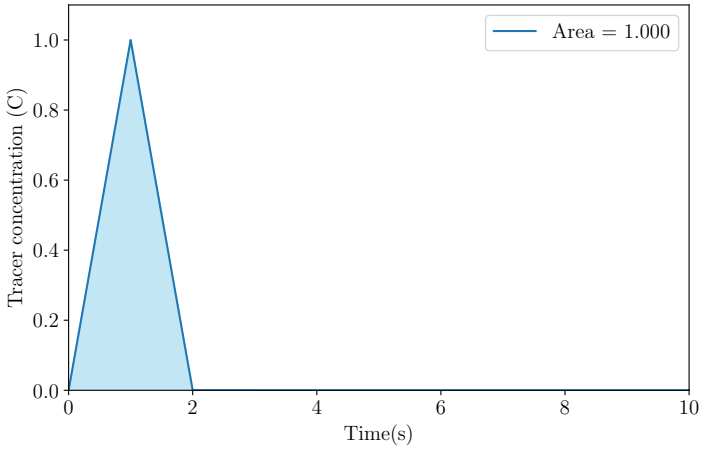


Figure 3.7: Imposed tracer concentration at the inlet boundary for the transient tracer simulation.

isothermal flow, the steady-state solver for incompressible, turbulent flows, `simpleFoam`, is employed. For non-isothermal flow, the steady-state solver for buoyant, turbulent flow of incompressible fluids, `buoyantBoussinesqSimpleFoam`, is used. To model the transient evolution of the tracer using the converged steady-state fields, a modified version of `scalarTransportFoam` is used, incorporating an additional turbulent diffusivity term in the scalar transport equation. This solver is referred to as `scalarTurbulentTransportFoam`.

The solution procedure for both isothermal and non-isothermal turbulent flow is based on the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm, as implemented in OpenFOAM using the solvers `simpleFoam` and

`buoyantBoussinesqSimpleFoam`, respectively. This algorithm adopts an iterative procedure to solve the governing equations for momentum, pressure, and, in the case of non-isothermal flow, temperature, thereby ensuring the conservation of mass, momentum, and energy.

Term	Scheme
Time derivative	steadyState
Convective term (momentum)	linearUpwindV
Convective term (energy)	linearUpwind
Convective term (turbulence: k, ϵ)	upwind
Convective term (species)	linearUpwind
Diffusive term (e.g., viscous, thermal)	linear
Gradient term	cellMDLimited linear

Table 3.2: Discretization schemes employed in the tundish steady-state operation study.

Table 3.2 summarizes the discretization schemes employed for solving the governing equations. The linear upwind scheme is used for the convective terms in the energy and species transport equations, while a linear upwind scheme based on velocity direction (linearUpwindV) is applied to the momentum equation. For turbulence quantities (k, ϵ), a more robust upwind scheme is employed to ensure numerical stability. Diffusive terms, including viscous and thermal diffusion, are discretized using a linear scheme, and the cellMDLimited linear method is used for gradient computations to maintain boundedness near steep gradients. However, for transient tracer simulations, a backward time discretization scheme is employed. Appropriate under-relaxation factors are used to enhance convergence, with values selected through trial-and-error.

3.6 Results and discussion

In this work, we perform three-dimensional steady-state turbulent flow simulations under both isothermal (where water is the working fluid) and non-isothermal (where molten steel is the working fluid) conditions. The streamwise velocity component U_x is extracted along horizontal and vertical lines (shown in Figure 3.8) within the tundish to assess the lateral and vertical variations in the flow field. Figure 3.9 presents a comparison of the streamwise velocity distributions under isothermal and non-isothermal conditions along

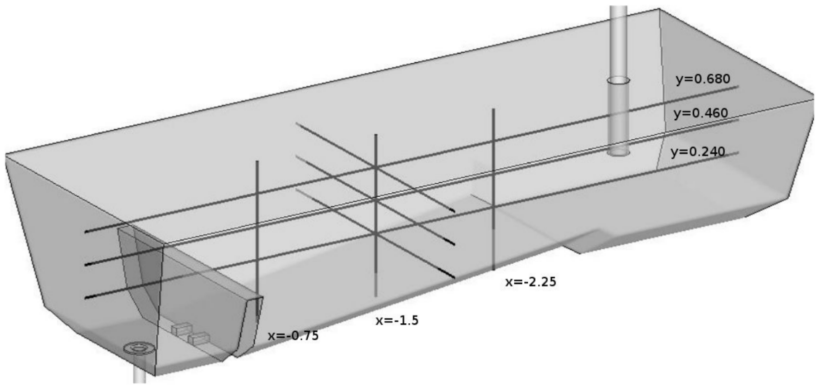


Figure 3.8: Locations of horizontal and vertical lines used for extracting streamwise velocity profiles within the tundish.

horizontal lines located at various y-positions. Figure 3.10 shows the vertical variation of the streamwise velocity along different x-positions.

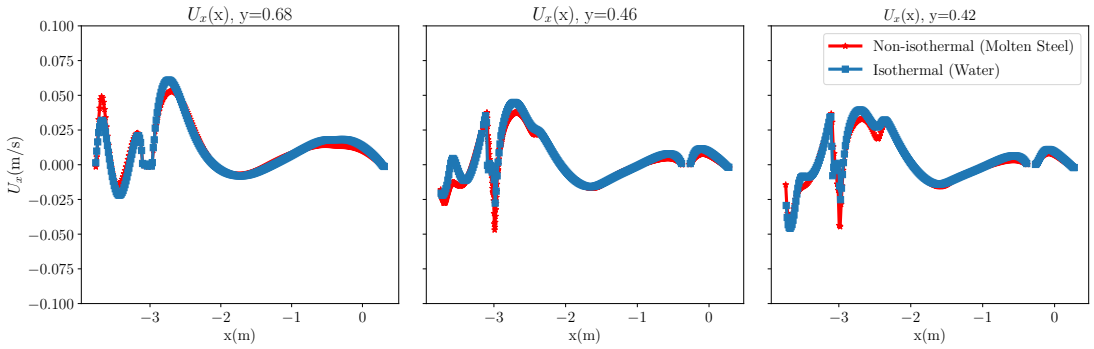


Figure 3.9: Streamwise velocity distributions along horizontal lines at different y-positions (shown in Figure 3.8), under isothermal and non-isothermal conditions.

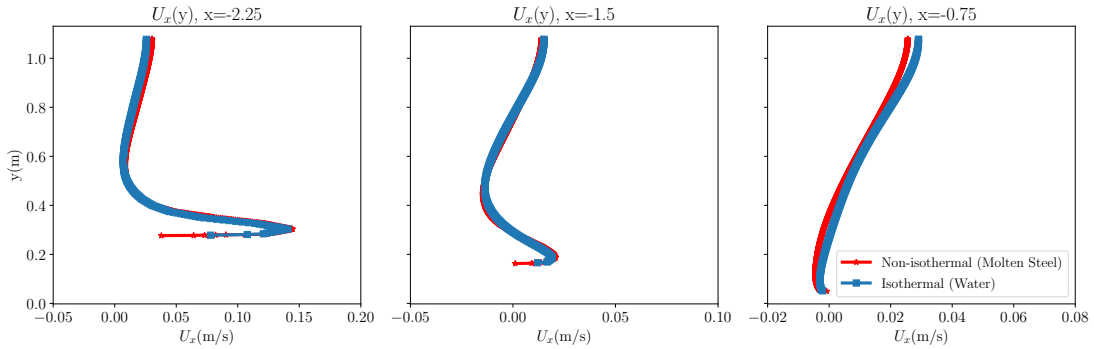


Figure 3.10: Streamwise velocity distributions along vertical line at different x locations (shown in Figure 3.8), under isothermal and non-isothermal conditions.

It is observed that the effect of buoyancy under non-isothermal conditions on the velocity field is negligible, as the velocity distributions under isothermal

and non-isothermal conditions match closely. These velocity distributions, together with the tracer studies, provide the basis for the subsequent RTD analysis and volume partitioning, enabling a comprehensive assessment of the flow behaviour and mixing efficiency within the tundish.

The transient simulation of tracer injection is employed to characterize fluid flow behaviour within the tundish. The resulting tracer concentration-time profile, referred to as the RTD curve, forms the basis for RTD analysis and volume partitioning. Figure 3.11 presents the RTD curve obtained from the FOM CFD simulations, with the key flow characteristics such as T_{min}, T_{max}, T_{avg}, T_{th} of the RTD curve being highlighted. The resulting plug volume V_{plug}, mixing volume V_{mix}, and dead volume V_{dead} are presented in Figure 3.12.

Figure 3.13 presents a comparison between the RTD curves obtained from full-order simulations and experimental measurements for the single-strand tundish. The simulation captures the overall trend and shape of the RTD curve well, including the onset, spread, and tailing behavior. A slight mismatch is observed at the peak concentration, likely due to idealizations in the numerical model or experimental variability. Nevertheless, the good agreement in key features such as the curve's width and decay profile supports the model's capability to represent the dominant transport dynamics.

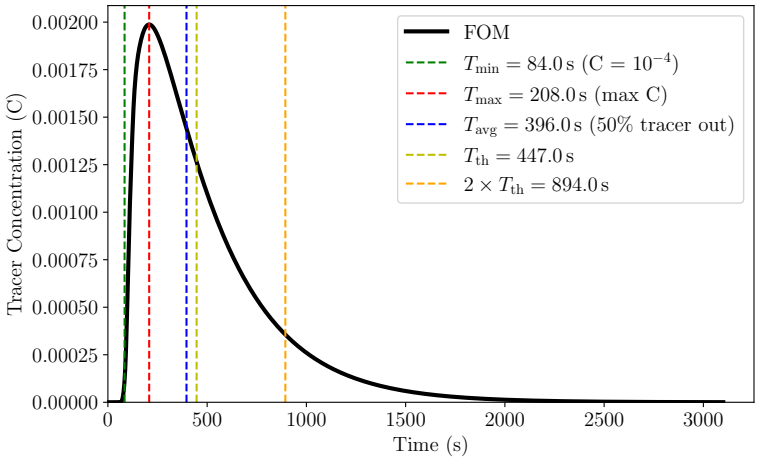


Figure 3.11: RTD curve obtained from FOM CFD simulations, highlighting key flow characteristics.

Table 3.3 presents the inlet velocities and flow rates used in the parametric study of the RTD curves for the single-strand tundish. These values are considered to perform a sensitivity analysis on the RTD characteristics of the system.

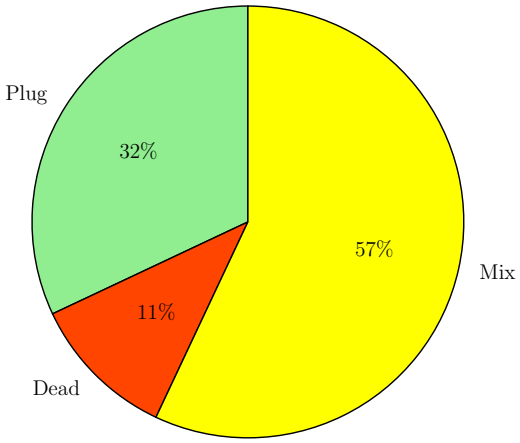


Figure 3.12: Volume partitioning derived from RTD analysis based on FOM simulation results.

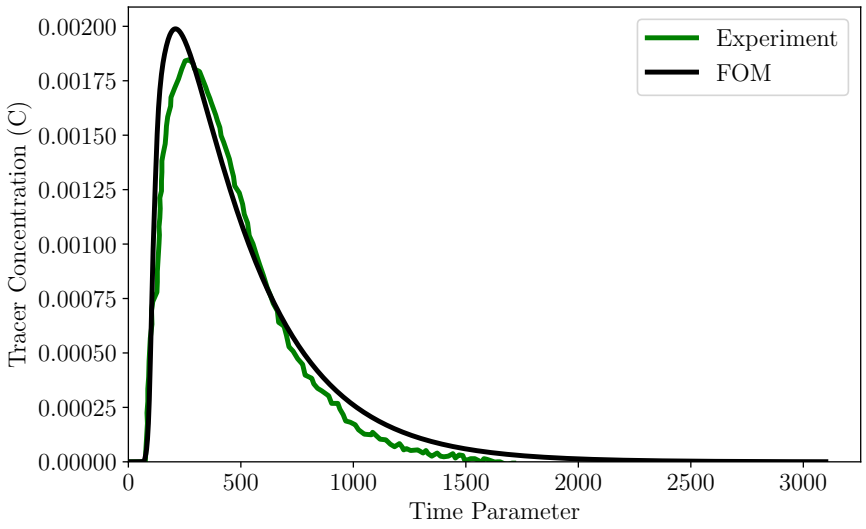


Figure 3.13: Comparison of residence time distribution (RTD) curves obtained from experiments and full-order simulations for a single-strand tundish.

For the parametric study of the RTD curves, we first obtain the convective velocity field by performing isothermal steady-state simulations for each parameter listed in Table 3.3. Next, transient scalar transport simulations are carried out to derive the corresponding RTD curves, which are presented in Figure 3.14. As the inlet velocity decreases, the RTD curves broaden, indicating an increase in fluid residence time within the tundish. The peak concentration of the RTD curve also decreases, and the distribution becomes more spread out. This occurs because lower velocities result in slower fluid movement, leading to less efficient mixing and a wider range of residence times to exit the system. Consequently, the RTD curves at lower velocities are less sharp and more

Parameter No.	Flow rate (m^3/hr)	Velocity (m/s)
1	36.9365	1.6301
2	34.52985	1.5239
3	32.1232	1.4177
4	29.71655	1.3115
5	27.3099	1.2053
6	24.90325	1.0991
7	22.4966	0.9929
8	20.08995	0.8866
9	17.6833	0.7804
10	15.27665	0.6742
11	12.87	0.5680

Table 3.3: Inlet velocities for the parametric study of RTD curves in a single-strand tundish.

spread out, highlighting the impact of velocity on residence time distribution and system efficiency. These changes highlight the influence of velocity on RTD characteristics and system performance.

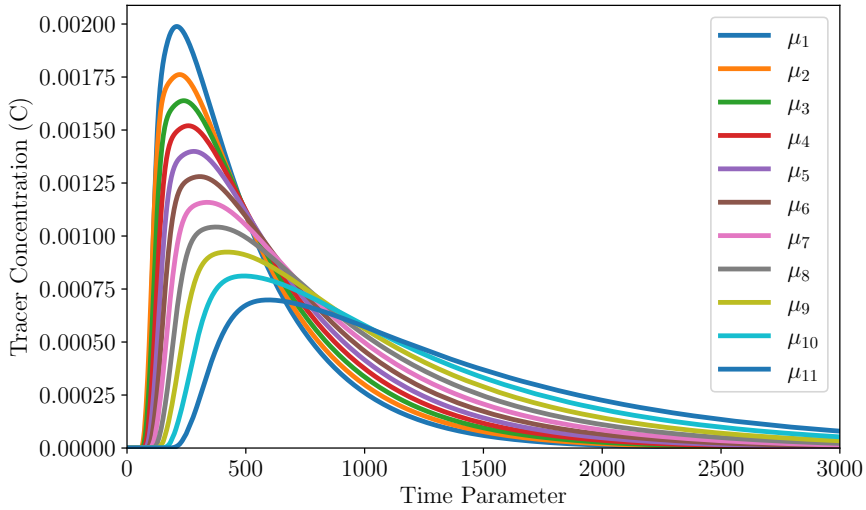


Figure 3.14: RTD curves obtained from full-order simulation, illustrating the variation in tracer concentration over time for each parameter, as listed in Table 3.3.

3.7 Summary

Over the past two decades, mathematical modelling has become an essential tool for investigating flow phenomena within steelmaking tundishes. Due to

the challenging nature of experimental studies in such high-temperature environments, physical investigations remain limited in scope. Advances in computational techniques have enabled significant progress, with numerical simulations now playing a pivotal role in optimising tundish performance.

In this study, full-order simulations were performed under both isothermal and non-isothermal conditions. The results indicate that buoyancy effects under non-isothermal conditions have minimal influence on the velocity field, as the velocity distributions in both cases are nearly identical. These velocity fields, combined with tracer transport simulations, serve as the foundation for the subsequent analysis of RTD and volume partitioning. Together, these results provide a comprehensive assessment of the flow behaviour and mixing efficiency within the tundish.

The parametric sensitivity study of the RTD curves highlights the significant influence of inlet velocity on the residence time distribution within the tundish. As the inlet velocity decreases, the RTD curves broaden, indicating longer fluid residence times and reduced mixing efficiency. These changes highlight the sensitivity of the tundish system to flow variations and underscore the importance of optimizing inlet velocity for maintaining optimal system performance and flow behavior.

Chapter 4

Tundish ladle changeover operation

4.1 Introduction

In sequential continuous casting of steel, when the teeming ladle is exhausted, it is replaced with a new one, allowing casting to continue without interruption for several hours. However, when the new ladle contains a different steel grade, the mixing of the two grades occurs in the tundish. To prevent grade intermixing, various industrial techniques, such as flying tundish changes and grade separator plates, have been developed for techno-economic reasons. However, these methods often result in significant revenue losses, increased processing time, and the need for handling multiple tundishes for maintenance and preheating, making them less favourable. With the growing demand for a diverse range of steel products in smaller batches, it has become increasingly important to address these challenges efficiently. The industry has found it more practical to implement a straightforward ladle changeover process, allowing continuous casting of different steel grades using the same tundish. However, this approach leads to the mixing of two distinct steel grades within the tundish, forming an intermediate grade with a composition between the two original grades.

Grade intermixing is majorly influenced by tundish operational parameters, such as inflow rate, residual volume, and outflow (throughput) rate have a more pronounced effect on the mixing of grades during sequential casting [13, 173, 174]. These parameters influence the melt flow structure inside the tundish. Understanding and quantifying the mixing process inside the tundish is crucial for steel makers. It has been observed that the melt bath height in the tundish changes during transient operations [128, 171]. A detailed investigation of the mixing phenomenon inside the tundish was conducted by Odenthal et al.[133], examining both steady and transient casting conditions, including

the transient filling sequence of the tundish and ladle change. They also numerically modelled the tundish filling process using the volume of fluid (VOF) method. Research to date has primarily focused on the steady-state operating bath height of the melt, both through numerical and experimental analysis [91, 80, 189, 193, 151]. A novel mixing model to quantify the composition distribution in solidified steel during grade mixing is proposed by Cho and Kim [35]. In their consecutive work [36], they demonstrated that the model could predict the intermixed zone for various casting operations, including bloom, slab, and thin slab casting. Alizadeh et al. [3] developed a numerical model to calculate grade mixing in the tundish and found that the extent of mixed-grade steel depends on the tundish Richardson number. Pieprzyca [148] studied the impact of mixing on the quality of solidified steel, noting that improper flow and mixing methods in the tundish can affect the chemical and thermal homogeneity of the steel, ultimately influencing cast quality. A detailed systematic study of mixing in the tundish was carried out by Muralikrishna et al. [128], who conducted an extensive physical investigation into grade intermixing. They found that residual volume significantly affects mixing and discussed how the internal geometry of the tundish influences grade mixing, developing a correlation to quantify mixed-grade steel formation. Siddiqui and Jha reported [172] that the inflow rate plays a crucial role in grade mixing during sequential castings. Cwudzinski [41] conducted a numerical study of flow control devices in the tundish to alter the hydrodynamic patterns, reporting that an active recycle zone could mitigate stagnant volumes, and that alloy addition mixing time depends on location and hydrodynamic patterns. Further explored hydrodynamic patterns by developing a numerical model to understand alloy dispersion in the melt, studying residence time distribution (RTD) and alloy dispersion in a single strand tundish. A more comprehensive multiphase study on mixing was conducted by Al-Harbi et al. [73]. They developed a 3D multiphase numerical model to quantify grade mixing in the tundish, validated by chemical analysis of solidified slabs. They concluded that 80% of the old steel grade could be removed from the tundish within the first 5-6 minutes of mixed-grade slab casting. Cwudzinski [42] also conducted physical and numerical experiments to understand the formation of the mixing zone during continuous casting, using the buoyancy number to visualize the hydrodynamic phenomena in the tundish. A recent study by Krashnavtar and Mazumdar [100] reported the influence of operating variables on grade intermixing time. More recently, Siddiqui and Kim [175] conducted numerical simulations to investigate the impact of residual volume and outflow (throughput)

rate on the mixing phenomenon inside the tundish.

The intermixing of steel grades and the subsequent production of mixed grades occur over a relatively short period, requiring a fully transient formulation for the hydrodynamic modelling of this process. Additionally, a multiphase approach is necessary, as the tundish working volume typically contains two or more phases, which change over time during ladle changeover operations. Mathematical modelling of grade intermixing requires substantial computational resources. Consequently, the studies conducted so far have been largely simplistic and rarely captured the key process features of ladle changeover and intermixing. As a result, despite numerous studies, a physically realistic model of ladle changeover and the associated grade intermixing has not yet been developed [100, 176].

Intermixed products are typically downgraded, leading to significant yield and revenue losses [100, 175]. As a consequence, the industry has a strong interest in minimizing intermixing within the tundish to reduce the production of mixed-grade slabs or blooms [100, 175]. The issue of grade intermixing in steelmaking tundishes has garnered significant attention over the past two decades [122, 31, 190, 120, 176]. Efforts have generally focused on predicting intermixing time through both mathematical and physical modelling, both differential and macroscopic models to address this challenge [100, 176].

4.2 Full order model

Multiphase modelling in tundish metallurgical operations enables the investigation of various phenomena, such as the molten steel-slag interface, ladle changeover and impurity dispersion in the melt, etc. The VOF method is capable of accurately predicting the variation in tundish melt height during the ladle changeover operation due to sequential casting.

4.2.1 Governing equations

The continuity (4.1) and momentum (4.2) equations, governing the flow of two immiscible fluids, are utilized to model fluid flow and steel grade intermixing in the tundish, assuming isothermal conditions.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho(\mathbf{U}) = 0, \tag{4.1}$$

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = -\nabla p + \nabla \cdot [\mu_{eff}(\nabla \mathbf{U} + \nabla^T \mathbf{U})] + \mathbf{F}. \tag{4.2}$$

where ρ is the density, $\mathbf{U}$ is the velocity, p is the pressure and $\mathbf{F}$ denotes gravitational and external body forces.

4.2.2 Volume of Fluid method

To capture the interface between two immiscible fluids, we use the one-fluid model approach VOF method. In the one-fluid model approach, two immiscible fluids are considered as one effective fluid, in which $\mathbf{U}$ acts as the shared velocity of the two fluids $\mathbf{U} = \mathbf{U}_1 = \mathbf{U}_2$. The physical properties of the fluids are averaged over the domain, which leads to constituent relations of ρ and μ weighted values given by:

$$\begin{aligned} \rho &= \alpha_1 \rho_1 + (1 - \alpha_1)\, \rho_2, \\ \mu &= \alpha_1 \mu_1 + (1 - \alpha_1)\, \mu_2. \end{aligned} \tag{4.3}$$

In the VOF method, the modified volume fraction advection equation (4.4) is solved simultaneously with continuity and momentum equations. Where $\alpha \in [0, 1]$ is phase fraction which takes values $\alpha = 0$ for volume is filled only with phase 2, $\alpha = 1$ for control volume is filled only with phase 1 and at the interface, it ranges between $0 < \alpha < 1$, and $\mathbf{U}_r$ is the relative velocity shown in (4.5) and is perpendicular to the interface.

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{U}) + \nabla \cdot [\mathbf{U}_r \alpha\, (1 - \alpha)] = 0, \tag{4.4}$$

$$\mathbf{U}_r = C_\alpha |\mathbf{U}| \frac{\nabla \alpha}{|\nabla \alpha|}, \qquad \text{with } C_\alpha \in [0, 1]. \tag{4.5}$$

4.2.3 Turbulence model

For turbulence modelling in the tundish, we consider the $k - \varepsilon$ turbulence model based on [106], which is a two-equation linear eddy viscosity turbulence closure model consisting of k - turbulent energy and ε - turbulent kinetic energy dissipation rate equations:

$$\frac{\partial(\rho k)}{\partial t} = \nabla \cdot (\rho D_k \nabla k) + P - \rho \varepsilon, \tag{4.6}$$

$$\frac{\partial(\rho\varepsilon)}{\partial t} = \nabla \cdot (\rho D_\varepsilon \nabla \varepsilon) + \frac{C_1 \varepsilon}{k}\left(P + C_3 \frac{2}{3} k \nabla \cdot \mathbf{U}\right) - C_2 \rho \frac{\varepsilon^2}{k}. \tag{4.7}$$

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{4.8}$$

where, D_k - effective diffusivity for k, P - turbulent kinetic energy production rate, D_ε - effective diffusivity for ε, C_1, C_2, C_3 are model coefficients. After obtaining k, ε, values, the turbulent (eddy) viscosity, μ_t is given by (4.8), here C_μ is the model coefficient for turbulent viscosity. We consider the standard model coefficient values shown in Table 4.1.

C_μ	C_1	C_2	C_3	σ
0.09	1.44	1.92	0	1

Table 4.1: Model coefficients of $k - \varepsilon$ turbulence model

4.2.4 Species transport equation

The evolution of the new steel grade is modelled using the species transport equation (4.9), which is a convection-diffusion equation:

$$\frac{\partial(\rho C)}{\partial t} + \nabla \cdot (\rho \mathbf{U} C) = \nabla \cdot (\mu_{eff} \nabla C), \tag{4.9}$$

$$\mu_{eff} = \mu + \mu_t. \tag{4.10}$$

where, C denotes the concentration of new steel grade, μ_{eff} represents the effective diffusivity (4.10), defined as the sum of molecular diffusivity - μ and turbulent diffusivity - μ_t.

4.3 Numerical setup

4.3.1 Computational domain

To reduce computational cost while preserving the essential flow characteristics, a three-dimensional computational domain is developed based on a 0.5 scaled industrial tundish, as shown in Figure 4.1. The decision to employ a scaled model is motivated by the significant reduction in computational resources required for transient simulations, particularly when evaluating multiple operating conditions. This scaling approach, commonly used in physical

modeling such as water model experiments, is here extended to numerical simulations. To further enhance computational efficiency, only half of the scaled tundish is simulated, taking advantage of the geometric symmetry.

The computational mesh consists of 2,181,611 cells, consisting of 1,496,381 hexahedral and 685,230 polyhedral elements, as shown in Figure 4.2. To enhance near-wall resolution, three prism layers are incorporated. The mesh is refined in the region of interface movement to accurately capture phase interactions. Additionally, further refinement is applied in the inlet jet impingement zone, as illustrated in the bottom right of Figure 4.2, to accurately capture the turbulent mixing phenomena.

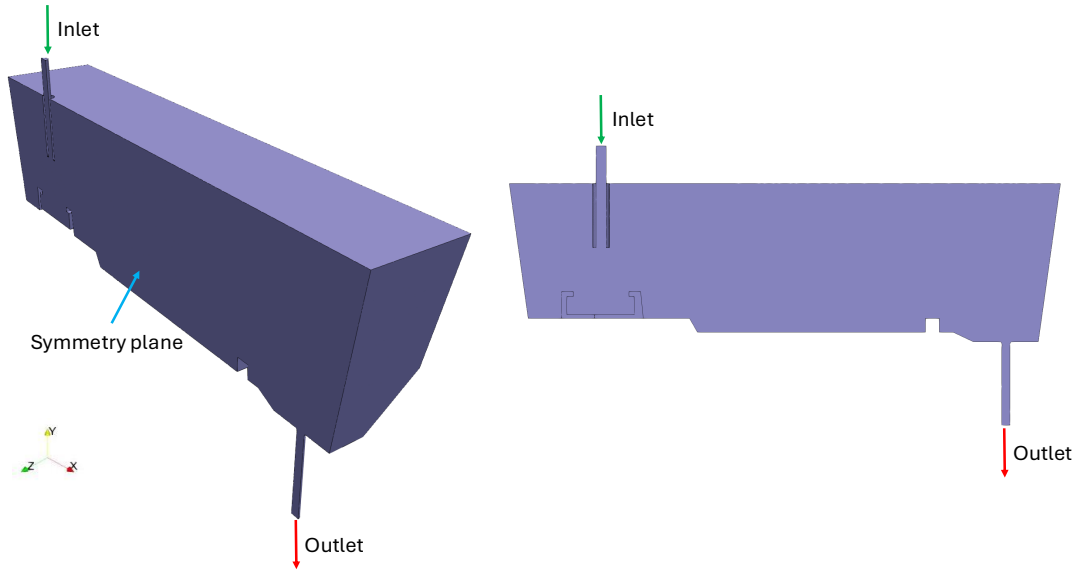


Figure 4.1: Computational domain of a 0.5 scaled single strand tundish. **Left**: Isometric view highlighting the inlet, outlet, and symmetry plane, with all other boundaries treated as walls. **Right**: 2D left-side view overlooking the symmetry plane.

4.3.2 Tundish operating and boundary conditions

Table 4.2 lists the operating parameters of the tundish, whereas Table 4.3 presents the fluid properties used in the FOM simulations. Table 4.4 summarizes the boundary conditions applied during the three significant stages of the ladle changeover operation of the tundish.

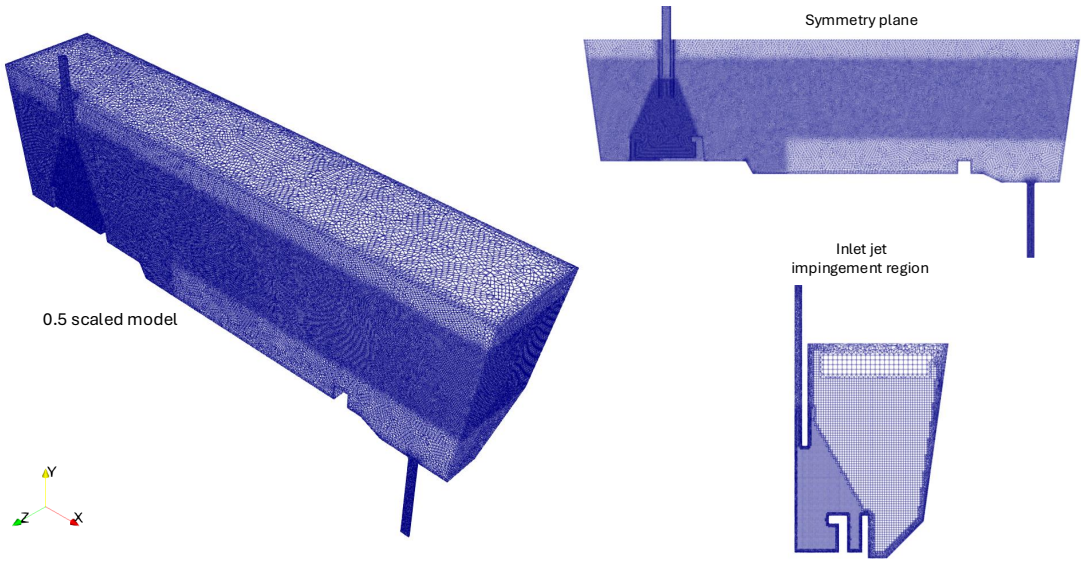


Figure 4.2: Discretized computational domain of a 0.5-scale single-strand tundish. **Left**: Isometric view of discretized domain; **Top right**: 2D symmetry plane showing the mesh refinement strategies to capture accurately the interface development and turbulence mixing phenomena; **Bottom right**: Inner jet impingement zone with an impact pot to enhance turbulence mixing.

Tundish operating parameters	Value
Height of molten steel (mm)	458.5
Inlet nozzle diameter (mm)	55
Outlet nozzle diameter (mm)	45
Tundish throughput (m^3/hr)	9.77
Inlet throughput velocity (m/s)	1.1424
Outlet throughput velocity (m/s)	1.7069
Filling flow rate (m^3/hr)	25.2538
Filling period inlet velocity (m/s)	2.9563

Table 4.2: Operating conditions of the tundish

Fluid properties	Value
Density of molten steel (kg/m^3)	7000
Kinematic viscosity of molten steel (m^2/s)	0.8×10^{-6}
Density of slag (kg/m^3)	2560
Kinematic viscosity of slag (m^2/s)	2.34×10^{-4}

Table 4.3: Fluid properties used in the ladle changeover operation simulations.

Stage 1: Steady-state operation (RANS steady and VOF - two phase flow model)	
Inlet	Constant velocity at inlet, volume fraction of old grade throughout = 1.0
	velocity-based turbulence kinetic energy and dissipation rate
Outlet	flowRateOutletVelocity condition to maintain a constant outflow rate.
Walls	No slip with standard wall function.
Stage 2: Tundish emptying operation (URANS, VOF - two phase flow model)	
Inlet	No inflow.
Outlet	flowRateOutletVelocity condition to maintain a constant outflow rate.
Walls	No slip with standard wall function.
Stage 3: Tundish refilling operation (URANS, VOF - two phase flow model + scalar transport equation)	
Inlet	Fixed or varying desired inlet velocity with volume fraction of new grade = 1.0.
	velocity-based turbulence kinetic energy and dissipation rate
Outlet	flowRateOutletVelocity condition to maintain a constant outflow rate.
Walls	No slip with standard wall function;

Table 4.4: Initial and boundary conditions considered for the three different stages of ladle changeover operations of a single-strand tundish.

4.4 Numerical method

Among the numerical methods used for multiphase flows, the open-source VOF solver `interFoam` has gained significant attention and widespread adoption [51]. Initially developed by Ubbink [195] within the FOAM framework [90], `interFoam` has undergone continuous modifications and improvements. It is now part of the OpenFOAM suite of C++ libraries, designed for the finite-volume discretisation of partial differential equations, particularly related to continuum mechanics. The object-oriented structure of C++ enables the code to closely mirror its mathematical formulation, while also making the high-level syntax flexible for further development and modification [207].

In this study, we use the `interFoam` solver from OpenFOAM, a two-phase incompressible, isothermal and immiscible fluid solver that employs a VOF phase fraction based approach for interface capturing. It also supports optional mesh motion and topology modifications, including adaptive re-meshing, and we highlight that these strategies were not employed in this study. For a detailed description of the solver, the reader is referred to [51]. In this work, the momentum equation is solved using the PIMPLE algorithm, which combines the PISO and SIMPLE methods to improve stability and convergence.

The solution procedure involves computing an initial velocity field, followed by multiple pressure correction loops to ensure accuracy.

Term	Scheme
Time derivative	Euler
Convective term (momentum)	linearUpwind
Convective term (volume fraction α)	MULES (vanLeer + interface compression)
Convective term (species: C)	upwind
Convective term (turbulence: k, ε)	upwind
Diffusive term (viscous)	linear
Gradient term	cellMDLimited Gauss

Table 4.5: Discretization schemes employed in the interFoam-based multiphase species transport solver for ladle changeover operation transient simulation.

Table 4.5 summarizes the discretization schemes employed for solving the governing equations. The volume fraction α is solved using the MULES scheme, which ensures boundedness and sharp interface capturing in the VOF formulation. The vanLeer scheme is applied for convective fluxes of α, with additional interface compression to maintain phase fraction limits. A linear upwind scheme is used for convective terms in the momentum equation, and the upwind scheme is applied to turbulence quantities (k, ε), temperature, and species transport equations to enhance numerical robustness. Diffusive terms, including viscous and thermal contributions, are treated with a linear scheme, and gradient terms are computed using the cellMDLimited Gauss method. For transient flow simulations, time discretization is performed using the Euler method. Appropriate under-relaxation was applied to ensure numerical stability and improve the convergence rate of the iterative solution process, with optimal parameters determined through trial-and-error. Additionally, the time step was considered such that the maximum Courant number of 2 is maintained to ensure stability across all the simulations.

Considering these tundish operating parameters, the specified initial and boundary conditions, the following methodology is employed to model the three distinct stages of ladle changeover and grade intermixing in the tundish:

Stage 1: **Steady-state operation** – A steady-state simulation is first conducted to obtain the converged velocity and turbulence fields, which serve as initial conditions for the subsequent VOF simulation with a tundish filled with molten steel and slag. The pre-ladle changeover phase is

then modelled using the VOF method, treating it primarily as a turbulent, homogeneous flow problem. The computed flow and turbulence fields within the active tundish volume are stored to initialize the subsequent transient simulation.

Stage 2: **Transient emptying operation** – The tundish emptying phase, occurring without inflow, is modelled using a transient, two-phase VOF approach, considering molten steel and slag as immiscible fluids. The flow and turbulence fields from Stage 1 serve as initial conditions, while the mass inflow is set to zero with constant outflow. Within this framework, the evolution of mixture velocity, turbulence, and phase volume fraction in the active tundish volume is computed over time.

Stage 3: **Transient refilling operation** – The refilling and grade intermixing phase is simulated using a numerical procedure similar to that described in Stage 1. In this stage, the velocity and turbulence fields, along with the volume fraction distribution obtained from Stage 2, are used as initial conditions for the simulation.

During all three stages, the tundish outflow was maintained at a constant predetermined rate to replicate the steady casting conditions characteristic of industrial continuous casting processes.

4.5 Results and discussion

For the initial parameter study, we consider the tundish throughput rate 9.77 m^3/hr with a ladle change time of 64s and corresponding filling duration of 40.40 s, with a filling flow rate of 25.25 m^3/hr for the newly replaced ladle. During the filling period, the volume of steel emptied during the ladle change process is replaced by the new steel grade. Figure 4.3 shows the contours of the new steel grade evolution. A probe is located at the tundish outlet to monitor the new steel grade evolution. The new steel concentration recorded by the probe positioned at the outlet is shown in Figure 4.4. For this initial parameter, ladle change time of 64s, the intermixing time determined by the FOM is 1092.24 s.

4.5.1 Effects of ladle change time

In this study, we examine the impact of ladle change time on the intermixing behaviour of a new steel grade in a single-strand tundish. Five different ladle

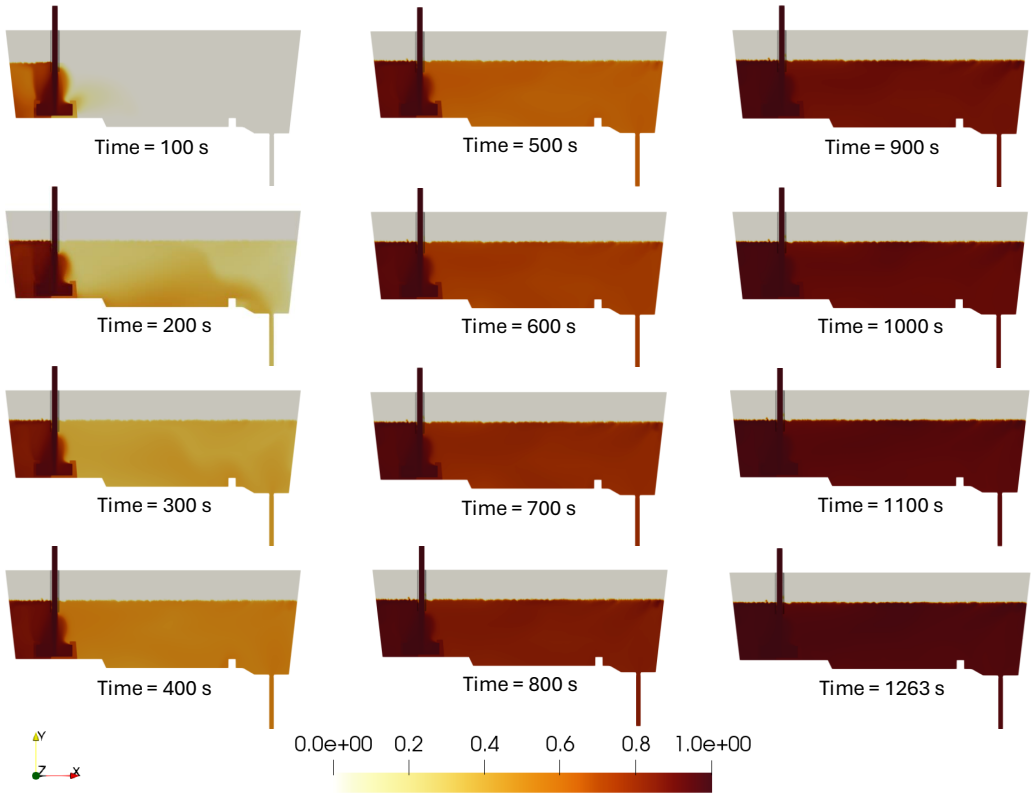


Figure 4.3: New steel grade transition contours corresponding to a ladle change duration of 64s and a filling time of 40.40s.

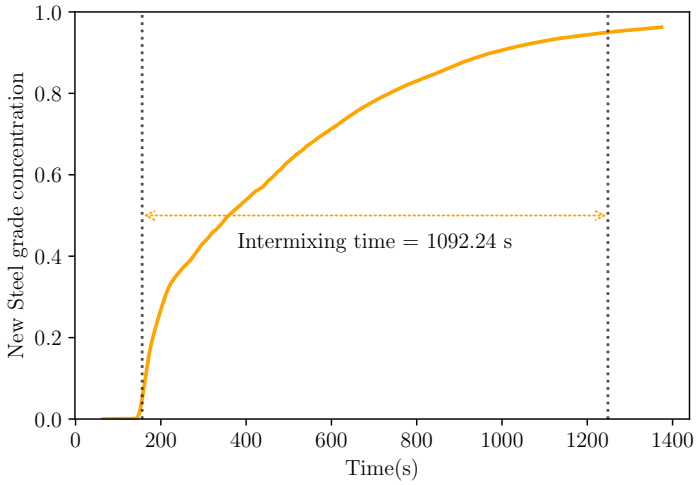


Figure 4.4: Intermixing time corresponding to a ladle change duration of 64s and a filling time of 40.40s.

change times and their corresponding filling durations, required to restore the melt height to steady-state levels, are considered. Table 4.6 summarizes the parameters used in the numerical experiments. Here, the primary quantity of interest (QoI) is the intermixing time, defined as the duration required for the

concentration of new grade steel to increase from 5% to 95% at the tundish outlet.

During the ladle change operation cycle, the tundish melt bath height remains low when a new ladle is introduced to supply molten metal. To rapidly restore steady-state melt height, the inflow rate from the ladle shroud is typically increased. At the start of pouring the new steel grade, a substantial volume of residual steel from the previous grade remains in the tundish, referred to as the residual steel volume (RVF). As the ladle change time increases, the filling duration also extends to maintain the melt height. Since ladle change time directly influences RVF, it has a significant impact on the intermixing of the two steel grades.

Expt. No	Ladle change time (s)	Filling time (s)
1	64	40.40
2	125	78.92
3	185	116.79
4	245	154.67
5	305	192.55

Table 4.6: Numerical experiment parameters considered to investigate the effects of ladle change time, ladle change time with corresponding filling time to reach steady-state operation.

The effect of ladle change time on melt bath height and residual steel volume (RVF) is illustrated in Figure 4.5. The results indicate that as ladle change time increases, the melt bath height decreases, and similarly, the RVF decreases. The mixing phenomena of two different miscible steel grades are influenced by variations in viscosity and density. While macroscopic flow dynamics primarily govern the mixing behaviour, microscopic interactions also contribute to the process. The extent of mixing is largely determined by turbulent kinetic energy, which facilitates the exchange of kinetic energy between the mean flow and fluctuating layers of fluids [172, 175]. Figure 4.6 shows the effect of ladle change time on the intermixing time. As illustrated, the intermixing time decreases with increasing ladle change time. It reveals that the intermixing time was shortest for the lowest residual volume and longest for the highest residual volume in the tundish. This trend can be attributed to the reduction in RVF, which leads to more efficient mixing of the new and old steel grades in the initial stages. It indicates that longer ladle change times promote faster intermixing of the two steel grades, thereby accelerating the homogenization of the new steel grade within the tundish. This study is essential for optimizing the steel production process.

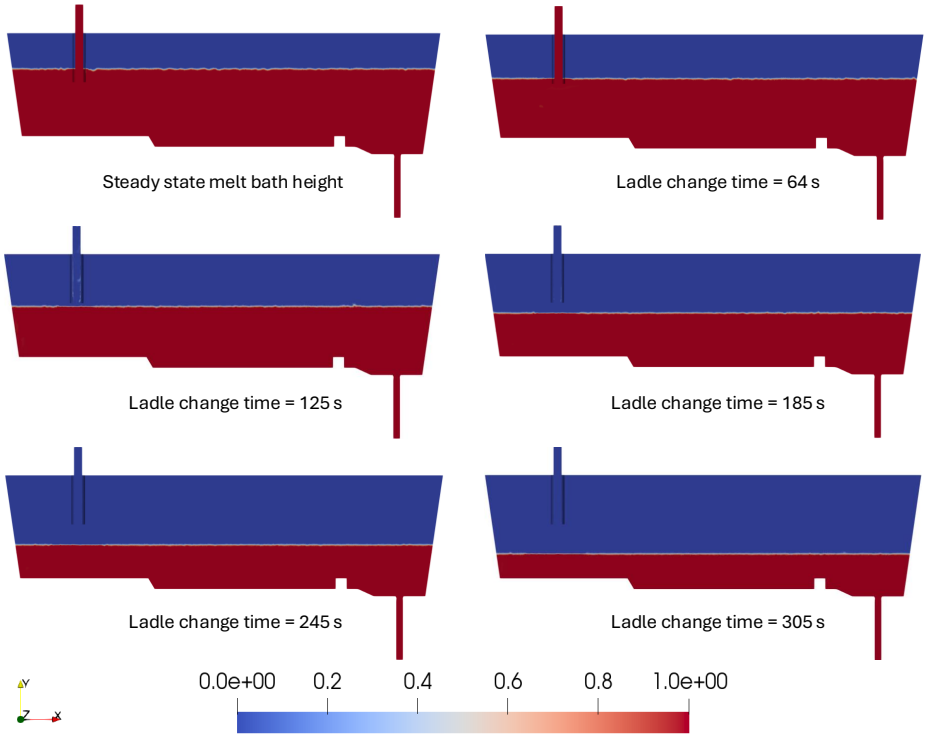


Figure 4.5: Effect of ladle change time on melt bath height and residual volume fraction.

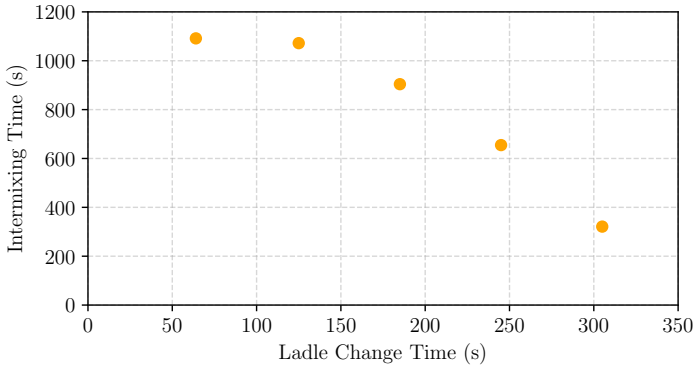


Figure 4.6: Effect of ladle change time on intermixing time

4.6 Summary

To ensure continuous casting and enhance productivity, different steel grades are cast sequentially through a ladle changeover operation in the continuous casting process. The key objective is to minimize grade intermixing, as the resulting intermixed region is typically regarded as inferior quality and often downgraded. A three-dimensional, transient, two-phase turbulent flow has been carried out to investigate ladle changeover operation and the associated

grade intermixing within the tundish. The effect of ladle change time on intermixing time is studied, revealing that ladle change time significantly influences the melt height and residual volume fraction (RVF). Moreover, RVF plays a crucial role in grade intermixing within the tundish. A lower RVF facilitates faster intermixing of grades, while a higher RVF tends to prolong the intermixing duration. Generally, for industrial-scale systems, obtaining full-order high-fidelity solutions of these physical processes is computationally expensive and requires long-time integration to simulate the entire operations cycle. This makes it prohibitive to perform extensive parameter exploration, which is crucial for the development of ROM.

Part II

Reduced Order Model

Chapter 5

Preliminaries on model order reduction

5.1 Introduction

Reduced order modelling is a rapidly advancing field in computational science and engineering. Industries are keenly interested in obtaining Reduced Order Models (ROMs) for engineering systems, particularly in applications involving control, optimization, and uncertainty quantification. ROMs offer significant computational efficiency in many query scenarios and are suitable for real-time computations. These models achieve computational efficiency by approximating high-dimensional parametric Partial Differential Equations (PDEs) solutions with low-dimensional representations, substantially reducing the computational cost compared to Full Order Models (FOMs).

To solve parametric PDEs that describe real-world scenarios, various discretization methods are used, such as the Finite Difference Method (FDM), the Finite Element Method (FEM), the Finite Volume Method (FVM), the Discontinuous Galerkin Method (DGM), and the Spectral Element Method (SEM). Although these methods are accurate, they are also computationally intensive. The ROM framework addresses this by dividing the process into two stages:

- Offline stage (training): This expensive phase involves computing a set of solutions (snapshots) and constructing the ROM based on a low-dimensional approximation subspace of the FOM solution manifold.
- Online stage (testing and predictive): This cost-effective phase uses the compressed information from the offline stage to predict reduced solutions for new unseen parameters, significantly reducing the computation costs.

5.2 Model reduction methods

Let the FOM solution manifold $\mathcal{M} \subset \mathcal{V}$ of $\dim(\mathcal{M}) = n$, consists of $s_i(x,t;\mu)$ snapshots for given parameters $\mu \in \mathcal{P} \subset \mathbb{R}^p$ and in spatial domain $\Omega \subset \mathbb{R}^d$, $d = \{1,2,3\}$, with varying time t in interval $[0, T_{\boldsymbol{\mu}}]$:

$$\mathcal{M} = \{s(\mu,t) \mid \mu \in \mathcal{P}, t \in [0, T_\mu]\} \subset \mathcal{V}. \tag{5.1}$$

5.2.1 Approximability of solution manifold

In model-order reduction literature, most of the work employs linear trial subspaces or manifolds [160]. The Kolmogorov n-Width (KnW) provides a rigorous measure of the reducibility of the solution manifold by linear approximation subspaces, obtained using POD and Reduced Basis (RB) methods using greedy approaches. For the solution manifold $\mathcal{M}$ consisting of f elements, embedded in some normed linear space $(X_\mathcal{N}, \|\cdot\|_{X_\mathcal{N}})$,

Definition 5.1 (Kolmogorov n-Width)**.** The definition of the KnW $d_n(\mathcal{M})$ is given by:

$$d_n(\mathcal{M}) := \inf_{E_n \subset X_N} \sup_{f \in \mathcal{M}_N} \inf_{g \in E_n} \|f - g\|_{X_\mathcal{N}}, \tag{5.2}$$

where E_n is a linear subspace of $X_\mathcal{N}$ of dimension n. The KnW $d_n(\mathcal{M})$ represents the worst error incurred when approximating any element $f \in \mathcal{M}$ by an element $g \in E_n$.

Furthermore, the faster the KnW $d_n(\mathcal{M})$ decays as n increases, the more effective a low-dimensional linear subspace of dimension n is at approximating the solution manifold $\mathcal{M}$.

5.2.2 Linear dimensionality reduction

In classical model reduction, the strong assumption considered that the solution manifold $\mathcal{M}$ can be expressed by a linear combination of global basis functions and modal coefficients as shown in (6.5), where the approximated snapshots $\tilde{s} \approx s$ are obtained by linear combinations of few global basis functions $\varphi_k \in \mathbb{R}^n$ and modal coefficients $c_i(t;\boldsymbol{\mu})$ (also coined as low-dimensional generalized coordinates), and here, s_{ref} is a parameterized reference state. The linear trail manifold $E_n = \{\varphi_1, ..., \varphi_n\}$ is fixed and independent of the $s_i(x,t;\boldsymbol{\mu})$ to be approximated. Here, the linear trial manifold is obtained by performing

POD using Singular Value Decomposition (SVD) [183] or the method of snapshots [150].

$$\tilde{s}_i(x,t;\boldsymbol{\mu}) = s_{\text{ref}}(\boldsymbol{\mu}) + \sum_{k=1}^{r} \varphi_k(x) c_i(t;\boldsymbol{\mu}) \quad r \ll n. \tag{5.3}$$

There are various model reduction methods to obtain a low-dimensional approximation subspace [150, 76, 34, 160]. Linear compression methods, such as POD and RB Methods, are the most popular and promising techniques for various fluid dynamics and solid mechanics applications. Although linear compression methods are broadly effective across numerous applications, many problems of interest (e.g., advection-dominated problems) exhibit slow Kolmogorov n-width decay, which limits the reduction attainable via linear approximation subspace [149].

5.2.3 Non-linear dimensionality reduction

As mentioned earlier, the solution manifolds of advection-dominated problems often exhibit slow KnW, implying there is no low-dimensional linear subspace that can approximate the solution space efficiently. Consequently, resulting in inefficient and inaccurate linear projection based ROMs for such problems. To overcome this barrier, a continuously differentiable non-linear trial manifold is considered, where approximations are sought in the following form:

$$\tilde{s}_i(x,t;\boldsymbol{\mu}) = s_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{g}(c_i(t;\boldsymbol{\mu})), \tag{5.4}$$

where, $\tilde{s} \approx s$ is approximated solutions, $E_n := \{\boldsymbol{g}(\hat{\boldsymbol{\xi}}) \mid \boldsymbol{\xi} \in \mathbb{R}^p\}$ is the non-linear trail manifold, $s_{\text{ref}}(\boldsymbol{\mu})$ is parameterized reference state and $\boldsymbol{g} : \boldsymbol{\xi} \mapsto \boldsymbol{g}(\hat{\boldsymbol{\xi}})$, referred to as decoder which employs non-linear mappings from low-dimensional generalized co-ordinates to the high-dimensional state approximation. To obtain this non-linear trail manifold, convolutional autoencoders can be employed [107, 159], and other non-linear manifold approaches are quadratic approximation manifolds [88, 12, 60].

5.3 Curse of dimensionality

For several elliptic and parabolic problems, the KnW decay is exponential, i.e., $d_n(\mathcal{M}) \leq Ce^{-\beta n}$, where constants $C < \infty$ and $\beta > 0$ [23, 134], which explains the effectiveness of model reduction methods for this class of problems. For hyperbolic problems, the n-width decay is at most $d_n(\mathcal{M}) > n^{-1/2}$

[64, 134], which indicates that advection-dominated problems cannot be effectively approximated using low-dimensional linear subspaces. Even the solution manifold of a simple linear advection problem exhibits slow KnW decay [142], meaning there is no low-dimensional linear approximation subspace that can effectively approximate the FOM solutions, and even if we consider the best-possible subspace, a large dimensional subspace should be considered to achieve a small error. This poses a considerable challenge for effective model reduction, as computational efficiency depends on the reduced linear subspace. Consequently, linear compression methods such as POD and RB are ill-suited for this class of problems, resulting in inefficient and inaccurate linear projection-based ROMs.

5.4 Overcoming slow Kolmogorov n-Width decay

In recent years there has been significant development in model-reduction methods to overcome the barrier of the slow KnW decay, which are leading towards non-linear approaches [142], particularly by performing model-reduction on non-linear manifolds, by obtaining non-linear trial subspaces using convolutional auto-encoders (CAEs) [93, 74, 40, 77, 107, 55, 95, 159] and by employing quadratic manifold [88, 165, 12, 60]; by incorporating adaptive local-in-parameter and/or local-in-space enrichment strategies [68, 205, 28, 145, 143, 20] to derive efficient ROMs; by relying on multiple-linear subspaces (known as the dictionary approach) for constructing local approximation subspace, instead of single global approximation subspace [5, 146, 44, 59]; and by transforming the linear subspace to improve the its approximation properties for advection-dominated problems, such methods incorporate freezing [135], shifting the POD basis [156] and manifold calibration/transformation [124, 81, 26, 158, 186, 192, 131, 130]. These approaches substantially rely on the prior knowledge of the underlying problem and physical properties, such as advection phenomena that guide the basis shifting.

5.5 Reduced order model construction: An overview

In reduced order modelling, once a low-dimensional approximation subspace of the FOM solution manifold is obtained, it becomes necessary to compute the evolution of the system dynamics within this subspace. This is achieved by solving for modal coefficients or latent coordinates. Two primary classes of approaches exist for this task: intrusive and non-intrusive.

In the intrusive approach, the discretized PDEs (governing equations) are projected onto a low-dimensional subspace to obtain a system of reduced order Ordinary Differential Equations (ODEs). This projection based method, commonly referred to as Galerkin or Petrov–Galerkin ROM, explicitly utilizes the governing equations and is inherently physics-based [14, 181, 182, 107, 159]. However, this approach requires full access to the underlying equations and the internal data structures of the high-fidelity solver, including the discretized operators. This dependency often makes intrusive methods impractical in industrial settings, where solvers are proprietary or inaccessible. Moreover, for non-linear or non-affine problems, the intrusive approach does not yield a significant speed-up [159].

In contrast, non-intrusive approaches are entirely data-driven and do not require access to the governing equations. These methods have demonstrated substantial speed-ups even for non-linear and non-affine problems [188]. Due to their flexibility and minimal implementation constraints, non-intrusive techniques are particularly attractive in real-world applications.

This work considers both intrusive and non-intrusive approaches, and the associated methodologies are elaborated in the subsequent chapters.

Chapter 6

ROM using neural network shift augmented manifold transformation

6.1 Introduction

Advection-dominated problems are predominantly noticed in engineering systems and industrial processes. For example, the multiphase turbulent flows within a tundish are inherently non-linear and often advection-dominated. These problems pose a significant challenge for traditional model reduction methods, such as POD and RB methods, due to slow KnW decay. As a result, an accurate approximation requires a high-dimensional linear subspace, which hinders the development of efficient ROM construction using linear methods. To address these limitations, this work utilizes the neural network shift augmented transformation technique employed within the NNsPOD framework [136] to overcome slow KnW associated with advection-dominated problems. This framework incorporates automatic shift detection to obtain a transformed linear subspace and does not require substantial prior knowledge of the underlying problem, such as governing equations or advection phenomena. This feature overcomes the limitation of the Shifted-Proper Orthogonal Decomposition (sPOD) approach [136, 156], which relies on prior knowledge of the problem, like several other transformation approaches mentioned earlier. For a wide range of problems, knowing the shifts a priori is either challenging or impossible, where automatic shift detection plays a crucial role. The neural network shift augmented transformation technique seeks optimal mapping of numerous snapshots to a reference frame by leveraging neural networks for

automatic-shift detection, aiming to obtain a low-dimensional transformed linear approximation subspace. Similarly, this type of transformation is achieved by solving an optimal transport problem as outlined in [81, 82], using a registration method [186, 97] and through an implicit tracking method [2]. In all these approaches, the mapping is built via an optimization procedure that aligns numerous snapshots with a reference state. Additionally, in [84], the method of characteristics and cross-correlation is utilised to circumvent the slow decay of KnW.

Further, in works [135, 185, 186], incorporating the mappings to obtain transformed linear approximation subspace for the construction of efficient ROMs, the construction of mapping is separated from the construction of solution coefficients and in [124] the mappings and solution coefficients are built simultaneously. Learning mapping and solution coefficients simultaneously leads to non-linear and non-convex optimization problems, even for linear PDEs [186]. Furthermore, applying parametric model reduction techniques to problems incorporating mappings is standard [186]. Conversely, when dealing with non-linear manifolds/trial subspaces, developing rapid and reliable procedures for online computation might be difficult and require several involved strategies [107, 159].

The previous work of NNsPOD [136] primarily focuses on the non-linear model reduction of the hyperbolic equations. It solely reports the transformation of the solution manifold by exploiting a deep learning framework to incorporate automatic shift detection, for seeking optimal mapping of numerous snapshots to the reference frame, followed by POD, to derive a low-dimensional transformed linear subspace. However, it is important to emphasize that it addresses only a portion of the offline stage within the ROM framework—specifically, the non-linear model reduction using NNsPOD. The complete algorithm comprising both the offline and online stages for the construction of an efficient ROM is not reported. In this work, we construct a ROM on the transformed linear approximation subspace obtained by NNsPOD, and further, we employ automatic shift detection in the online phase to map/transport back the predictions in a correct physical frame. Thereby, we propose a complete NNsPOD-ROM algorithm comprising offline and online stages for the construction of accurate and efficient ROMs.

The main contribution of this chapter are as follows:

- We construct reduced order models on the low-dimensional transformed linear approximation subspace, obtained by performing non-linear model

reduction using NNsPOD. This employs automatic shift detection by exploiting neural networks to derive optimal mapping of the numerous snapshots to the reference frame and aims to accelerate the KnW decay for advection-dominated problems [136].

- We propose the complete NNsPOD-ROM algorithm comprising both offline and online stages, where we integrate the general framework of non-intrusive parametric ROMs within the NNsPOD approach to construct accurate and efficient ROMs.
- We employ automatic shift detection in the online stage, to transport/map back the predicted field values at the reference frame (where ROM is constructed) to the physical frame/location in the physical space.
- We evaluate the proposed methodology through various numerical experiments, by assessing its performance on a 1D linear advection equation, the popular higher order benchmark test case - 2D isentropic convective vortex [179, 203] and the 2D two-phase flow test case with a focus on capturing the interface, whose physics is inspired by an industrial scenario. Both the 2D problems are governed by hyperbolic conservation laws. Further, we highlight that we focus on time-dependent problems where there is slow decay of KnW due to the advection nature of the problem.

We came across that the works reported in [98, 24] propose a workflow for the development of ROM using sPOD comprising offline-online stages, in which neural networks are employed for mapping parameters to modal coefficients and also for the mapping of shifts, particularly for the predictions in the online stage. In this work, we utilise neural networks for automatic shift detection, enabling the transformation of linear subspace by transporting all the snapshots to the reference frame by utilising automatic shift detection. This approach yields a low-dimensional transformed linear approximation subspace, facilitating the construction of efficient and accurate ROMs using linear methods. The framework proposed in [217] is inspired by NNsPOD, it solely addresses model reduction applied to the manufactured test cases and the complete workflow comprising offline-online stages for the construction of ROMs is not reported. Contrary, we propose a complete NNsPOD-ROM algorithm comprising offline-online stages for the construction of ROMs and we test our methodology on a higher-order benchmark test case - an isentropic convective vortex and a two-phase flow test case inspired by an industrial scenario, which is governed by hyperbolic conservation laws.

6.2 Model problem

Let us consider the abstract problem defined by the following general time-dependent PDE for an unknown space-time dependent scalar field $\mathbf{u}(\mathbf{x}, t)$:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathcal{L}(\mathbf{u}) = \mathcal{S}(\mathbf{u}, t), \quad \mathbf{x} \in \Omega, \quad t \in T, \tag{6.1}$$

where $\frac{\partial \mathbf{u}}{\partial t}$ represents the rate of change of the scalar field with respect to time, $\mathcal{S}(\mathbf{u}, t)$ is a source term (e.g., external forcing or a sink term), and $\mathcal{L}(\mathbf{u})$ is a differential operator that governs the evolution of $\mathbf{u}$ due to various processes such as advection, diffusion and/or other dynamical interactions. The operator $\mathcal{L}(\mathbf{u})$ governs the dynamics of the problem and can be either linear or non-linear. In many physical systems, $\mathcal{L}(\mathbf{u})$ combines both an advection term, $\mathbf{v} \cdot \nabla \mathbf{u}$ and a diffusion term, $\mathcal{D}\nabla^2 \mathbf{u}$:

$$\mathcal{L}(\mathbf{u}) = \mathbf{v} \cdot \nabla \mathbf{u} + \mathcal{D}\nabla^2 \mathbf{u}. \tag{6.2}$$

Here, $\mathbf{v}(\mathbf{x}, t)$ represents the velocity field and $\mathcal{D}$ is the diffusivity. In advection-dominated problems, the dynamics are primarily governed by the advection term $\mathbf{v} \cdot \nabla \mathbf{u}$, which describes the transport of the scalar field $\mathbf{u}$ by the velocity field $\mathbf{v}$, rather than by diffusion or other processes. In such cases, the diffusion term is negligible compared to the advection term.

To solve such time-dependent PDE (6.1) with given initial and boundary conditions, one can use any discretization method mentioned earlier (like FDM, FVM) to obtain semi-discretized equations or ordinary differential equations (ODEs) and further, the choice of time integration schemes, like Runge-Kutta schemes. However, it is important to note that solving such problems is often computationally expensive, and the solution manifold of advection-dominated problems (where diffusion is negligible) exhibits slow KnW decay.

Let $\mathcal{M}$ represent the solution manifold of the advection-dominated problem (6.1), consisting of all the solutions $\mathbf{u}(\mathbf{x}, t)$, where $\mathbf{x}$ lies in the spatial domain Ω and t spans the temporal domain T. The solution manifold is defined as:

$$\mathcal{M} := \{\mathbf{u}(\mathbf{x}, t) \mid \mathbf{x} \in \Omega, t \in T\}. \tag{6.3}$$

As we are dealing with time-dependent problems, the FOM solutions database is represented as a collection of time-snapshot pairs, denoted by $\{t_i, \mathbf{u}_i\}_{i=1}^{N_T}$, where $t_i \in T \subset \mathbb{R}^{N_T}$ corresponds to N_T discrete time instances, and $\mathbf{u}(t_i) \in$

$\mathbb{R}^{N_h}$ represents the solution state with N_h degrees of freedom at each time instance t_i. These snapshots are organized column-wise to form the snapshot matrix $X \in \mathbb{R}^{N_h \times N_T}$, which can be expressed as:

$$X = \begin{bmatrix} | & | & & | \\ \mathbf{u}(t_1) & \mathbf{u}(t_2) & \dots & \mathbf{u}(t_{N_T}) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{N_h \times N_T}. \tag{6.4}$$

6.3 Reduced order model

In this section, we present the general framework for non-intrusive data-driven ROMs based on POD [188], a popular linear compression method that has been extensively used in the ROM community over recent decades, especially for applications in computational fluid dynamics [160].

6.3.1 Proper orthogonal decomposition

We obtain the linear approximation subspace by performing linear dimensionality reduction using POD. The snapshots $\mathbf{u}_i$ can be expressed by a linear combination of global basis functions as shown in (6.5), approximating the snapshots by few global basis functions (spatial modes), $\varphi_k(\mathbf{x}) \in \mathbb{R}^{N_h}$ and few solution coefficients (temporal coefficients), $c_i^k(t)$. The linear approximation subspace $\mathcal{V}_{N_T} = \{\varphi_1, \dots, \varphi_{N_T}\} \in \mathbb{R}^{N_h \times N_T}$ is fixed, independent of the $\mathbf{u}_i$ to be approximated. To obtain the spatial modes $\varphi_k(\mathbf{x})$, we compute POD modes using singular value decomposition (SVD) [183] as shown in (6.6).

$$\mathbf{u}_i = \sum_{k=1}^{N_T} \varphi_k(\mathbf{x}) c_i^k(t) \approx \sum_{k=1}^{r} \varphi_k(\mathbf{x}) c_i^k(t) \quad \forall i \in [1, \dots, N_T], \quad r \ll N_T, \tag{6.5}$$

$$\underbrace{X}_{N_h \times N_T} = \underbrace{\mathbf{U}}_{N_h \times N_h} \underbrace{\mathbf{\Sigma}}_{N_h \times N_T} \underbrace{\mathbf{V}^\top}_{N_T \times N_T} \quad \Rightarrow \quad \underbrace{X}_{N_h \times N_T} \approx \underbrace{\hat{\mathbf{U}}}_{N_h \times r} \underbrace{\hat{\mathbf{\Sigma}}}_{r \times r} \underbrace{\hat{\mathbf{V}}^\top}_{r \times N_T}, \tag{6.6}$$

where, $\mathbf{U}$ and $\mathbf{V}$ are the two orthonormal matrices and hence, unitary matrices, $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbb{I}$, where $\mathbb{I}$ is the identity matrix. Columns of $\mathbf{U}$ are the left singular vectors, also called POD modes and columns of $\mathbf{V}$ are the right singular vectors of the transformed snapshot matrix $\mathcal{X}$ respectively. $\mathbf{\Sigma}$ is the diagonal matrix with L non-zero real singular values, arranged in descending order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_L > 0$, which indicates the energy contribution of the corresponding modes. As stated in the Schmidt–Eckart–Young–Mirsky theorem [53], our goal is to approximate the columns of $\mathcal{X}$ by identifying a

low-dimensional subspace spanned by $r < L$ orthonormal left singular vectors:

$$\mathcal{V}_r = \{\varphi_1, \ldots, \varphi_r\} \in \mathbb{R}^{N_h \times r}. \tag{6.7}$$

This is achieved by selecting the first r singular values and their corresponding left singular vectors, also known as POD modes. The resulting POD modes satisfy the following minimization problem [150]:

$$\min_{\mathcal{V}_r} \left\| \mathcal{X} - \mathbf{U}_r \mathbf{U}_r^T \mathcal{X} \right\|_F, \tag{6.8}$$

where $\|\cdot\|_F$ denotes a frobenius norm, $\mathcal{V}_r$ is the linear subspace of reduced dimension r, $\mathcal{X}$ is the transformed snapshots matrix and $\mathbf{U}_r$ consists of the first r columns of $\mathbf{U}$ (POD modes). Thus, the reduced linear subspace is the set of vectors that minimize the distance between the snapshots and their projection onto the subspace spanned by the POD modes.

The energy retained by the first r POD modes is given by (6.9), where r is chosen such that the $E(r)$ exceeds a specified threshold.

$$E(r) = \frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{L} \sigma_i^2}. \tag{6.9}$$

Once we have the r most energetic POD modes (spatial modes), we compute the modal coefficients (temporal coefficients) as shown in (6.10), by projecting the snapshots matrix onto the POD modes. The initial snapshot set can only be reconstructed using these coefficients by following the procedure outlined in (6.5). To predict accurate snapshots for new, unseen parameters, we need to build parameters to the modal coefficients map, which is discussed in the next subsection. Then, just by simple matrix multiplication, we can obtain the transformed snapshots of interest.

$$c(t_i) = \mathbf{U}_r^T \, \mathbf{u}(t_i). \tag{6.10}$$

6.3.2 Solution manifold approximation

There are several choices available for solution manifold reconstruction described by the modal coefficients corresponding to the initial database, in our case, the snapshots matrix X. Here, the main aim is to construct a regression $s : T \mapsto \mathbb{R}^r$ which approximates the map f as in (6.11) given by a set of N_{train} input-output pairs $\{t_i, c_i\}_{i=1}^{N_{train}}$, where t_i is the parameter value associated with

the i^{th} snapshot and c_i are the corresponding temporal coefficients obtained by (6.10).

$$f : t \in T \mapsto c \in \mathbb{R}^r. \tag{6.11}$$

Then this constructed regression model is used to predict the snapshot $\tilde{\mathbf{u}}^*$ for a new unseen input parameter t^* by computing,

$$\mathbf{u}^* = \mathbf{U}_r \, s(t^*). \tag{6.12}$$

To device this regression s for the parameters to coefficients mapping, there are several techniques [188], namely linear interpolation, non-linear Gaussian process regression (GPR), radial basis function (RBF) interpolation [210, 186], multi-fidelity methods, inverse distance weighting (IDW) and artificial neural networks (ANNs) [75], to name few. In this work, we mainly use radial basis function (RBF) interpolation to build parameters to the modal coefficients map.

The solution manifold of advection-dominated problems exhibits slow KnW decay, meaning it requires a high-dimensional linear approximation subspace to accurately represent the solutions. This creates a significant challenge for efficient model reduction, as the computational efficiency of ROMs depends on the reduced dimensionality of the linear approximation subspace. To circumvent this issue, we employ a manifold transformation technique that accelerates the KnW decay and enables the construction of efficient and accurate ROMs using linear methods. Further, we highlight that this transformation technique doesn't require any a priori knowledge of the physical properties (i.e., characteristic advection velocities) and the underlying equations of the given problem.

6.4 Neural network shift augmented transformation

In this section, we introduce the neural network shift augmented transformation technique, which leverages deep learning architectures to employ automatic shift detection by learning a shift operator. We exploit this operator to align numerous snapshots to a reference frame, thereby accelerating the KnW decay and obtaining a low-dimensional transformed linear subspace that effectively approximates the solutions of the advection-dominated problems. In section 6.3, we discuss the construction of efficient and accurate non-intrusive reduced-order models (ROMs) utilizing the transformed linear subspace.

The neural network shift augmented transformation technique introduced within the NNsPOD framework [136] is inspired by the sPOD approach [156]. In sPOD, snapshots are spatially transported to a reference frame using a shifting operator $\mathcal{T}_{\mathbf{b}}$, which acts on the space-time dependent field $\mathbf{u}(\mathbf{x}, t)$ as shown in (6.13). The resulting shifted snapshot, $\tilde{\mathbf{u}}$, corresponds to a transformed spatial distribution obtained by applying the shifting quantity that is proportional to the advection velocity, $\mathbf{b}$.

$$\tilde{\mathbf{u}} = \mathcal{T}_{\mathbf{b}}\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x} - \mathbf{b}t, t), \quad \mathbf{x} \in \Omega, \quad t \in T. \tag{6.13}$$

The application of sPOD relies on comprehensive knowledge of the advection velocities of a given problem, being the significant limitation of this approach [136]. Although the authors of [156] propose different methods to deduce the shift operator when the advection velocities are not known explicitly, these methods inevitably involve complexities and limitations, which are thoroughly discussed in section 2.4 of [136]. The NNsPOD framework [136, 160] overcomes this limitation by employing a semi-supervised learning paradigm for automatic shift detection. This approach automatically finds the optimal shifts for numerous snapshots to the reference frame by leveraging deep learning frameworks. The capability for automatic shift detection is particularly critical when shifts are not known a priori, as determining them in advance is often impractical or even infeasible for a wide range of problems.

In NNsPOD, the neural-network shift augmented manifold transformation is followed by linear compression POD, for non-linear reduction of hyperbolic equations. In this approach, the deep-learning framework [136] is utilized to derive a mapping between the original manifold $\mathcal{M}$ and the shifted manifold $\tilde{\mathcal{M}}$ as shown in (6.14). This method does not require any underlying equations (physical models or governing equations) and a priori knowledge of the given problem's physical properties (i.e., advection velocities), making it purely data-driven. Here, we learn the shift operator $\mathcal{T}_{\text{shift}}$ using neural networks, which acts on the field $\mathbf{u}(\mathbf{x}, t)$ as shown in (6.15), where $\mathcal{T}_{\text{shift}} : \Omega \times T \to \Omega$ is the shifting quantity that is not a fixed value linearly depending on time as in (6.13), but instead it depends on space and time [136, 160].

$$\mathcal{C}^1 \ni \mathcal{T}_{\text{shift}} : \mathcal{M} \mapsto \tilde{\mathcal{M}}, \tag{6.14}$$

$$\tilde{\mathcal{M}} := \{\mathbf{u}(\mathbf{x} - \mathcal{T}_{\text{shift}}(\mathbf{x}, t), t), \quad \mathbf{x} \in \Omega, \quad t \in T\}. \tag{6.15}$$

In this work, we employ an artificial neural network (ANN), specifically a

multi-layer perceptron (MLP) referred to as ShiftNet, to learn the shift operator for the given problem. In principle, one can opt to choose different architectures depending on the complexity of the problems at hand. The learning process is built on the set of snapshots as a discrete field, as defined in (6.16), where $\mathbb{V}$ is the discretized domain, and $t_i \in T$ with N_T number of time instances at which snapshots are collected. The primary objective is to transform the snapshots effectively by minimizing the Euclidean distance between them and the selected reference configuration, $\mathbf{u}_{\text{ref}}$. The choice of the reference configuration is arbitrary, as it is selected from the set of snapshots. For this task, we define the loss function $\mathcal{L}_{\text{ShiftNet}}$, as shown in (6.18).

After the shift detection, in a discrete context, the transformation $\mathbf{x} - \mathcal{T}_{\text{shift}}(\mathbf{x}, t)$ may not lie within the discrete space $\mathbb{V}$. This necessitates an additional step to calculate the mean difference between the snapshots and the reference configuration. Consequently, another crucial step in this transformation technique is reconstructing the transformed snapshots' field values within the same discrete space $\mathbb{V}$, also referred to as physical space [136, 156]. To facilitate the reconstruction of field values in the same discrete space, a neural network termed InterpNet is integrated into the algorithm. InterpNet is specifically devised to learn an operator that serves as an interpolator during the training phase of this transformation technique. The loss function $\mathcal{L}_{\text{InterpNet}}$ considered for InterpNet is the mean squared error (MSE) function, as shown in (6.17), where $\mathbf{u}_{\text{ref}}$ represents the field values of the reference snapshot, while $\mathbf{x} \in \mathbb{V}$ denotes the coordinates of the cell centroids or nodes in the discretized domain $\mathbb{V}$ and N_h is the number of degrees of freedom. The reference snapshot is chosen arbitrarily from the snapshot set.

$$\{\mathbb{V} \ni \mathbf{u}(t_i)\}_{i=1}^{N_T} = \{\mathbf{u}(\mathbb{V}, t_i)\}_{i=1}^{N_T}, \tag{6.16}$$

$$\mathcal{L}_{\text{InterpNet}} = \frac{1}{N_h} \sum_{j=1}^{N_h} \left\| \mathbf{u}_{\text{pred}}(\mathbf{x}_j) - \mathbf{u}_{\text{ref}}(\mathbf{x}_j) \right\|_2^2, \quad \mathbf{x} \in \mathbb{V}, \tag{6.17}$$

$$\mathcal{L}_{\text{ShiftNet}} = \frac{1}{N_T} \sum_{i=1}^{N_T} \| \mathbf{u}_{\text{ref}} - \mathbf{u}(\mathbf{x} - \mathcal{T}_{\text{shift}}(\mathbf{x}, t_i), t_i) \|_2^2, \quad \mathbf{x} \in \mathbb{V}. \tag{6.18}$$

In Algorithm 1, we report the neural network shift augmented transformation algorithm. In this transformation technique, the automatic shift detection and the reconstruction of field values workloads are divided between two separate neural networks:

- **ShiftNet** learns the optimal shift operator $\mathcal{T}_{\text{shift}}(\mathbf{x}, t)$ for a given problem. This operator precisely quantifies the optimal shifts, which results in shifted space that transports/maps numerous snapshots to the reference frame. To learn this shift operator we minimize the loss function $\mathcal{L}_{\text{ShiftNet}}$ given in (6.18).
- **InterpNet** learns the reference configuration based on the grid distribution $\mathbf{x} \in \mathbb{V}$ (physical space). To achieve this, we define the loss function $\mathcal{L}_{\text{InterpNet}}$ shown in (6.17). Once trained, InterpNet serves as an interpolator in the ShiftNet training process, to compute transformed snapshot field values in each shifted space.

Training neural networks: InterpNet and ShiftNet are trained sequentially, with InterpNet being trained first, followed by ShiftNet. InterpNet learns to reconstruct the field values of the reference configuration, and its forward map is subsequently used in ShiftNet training to reconstruct field values in each shifted space. Therefore, effective optimization of the InterpNet loss function, $\mathcal{L}_{\text{InterpNet}}$, is crucial.

The input to InterpNet is $\mathbf{x}$, the spatial coordinates of cell centroids or nodes in the discretized domain $\mathbb{V}$ of the selected reference configuration, while the output is the corresponding field values $\mathbf{u}_{\text{ref}}$. As mentioned earlier, the reference configuration is chosen arbitrarily from the training set. Upon achieving the required criterion by minimizing the loss function $\mathcal{L}_{\text{InterpNet}}$, the operator $\mathcal{T}_{\text{InterpNet}}$ is obtained, which functions as an interpolator in ShiftNet training. It is important to note that a single shift operator is trained using the complete training dataset.

For ShiftNet, the inputs are $(\mathbf{x}, t_i)$, where t_i represents the time instances at which snapshots are collected, and the output is the spatial coordinates $\mathbf{x}$ (physical space). During training, for each shifted space, the field values are computed using the $\mathcal{T}_{\text{InterpNet}}$ operator. The loss function $\mathcal{L}_{\text{ShiftNet}}$ is minimized to determine the optimal shift operator $\mathcal{T}_{\text{Shift}}$.

Manifold transformation: After training neural networks, the shift operator $\mathcal{T}_{\text{Shift}}$ is used to compute the optimal shifted space $\tilde{\mathbf{x}}_i^{\text{ref}}$, aligning multiple snapshots to the reference frame. The optimal shifted space is obtained by performing the operation $\tilde{\mathbf{x}}_i^{ref} = \mathbf{x} - \mathcal{T}_{\text{shift}}(\mathbf{x}, t_i)$, a procedure referred to as *automatic shift detection*. After transporting/mapping all snapshots to the reference frame utilising the optimal shifted space $\tilde{\mathbf{x}}_i^{\text{ref}}$, the shifted/transformed snapshots field values must be reconstructed in physical space $\mathbf{x}$ (where $\mathbf{x} = \mathbf{x}_{\text{ref}}$), in the same

grid distribution. This step is crucial in achieving sharp KnW decay and is performed using a linear interpolator $\mathcal{I}$. After this step, all the transformed snapshots are at the reference frame and in the same grid distribution $\mathbf{x} \in \mathbb{V}$.

Algorithm 1: Neural network shift augmented transformation

Data: Inputs $(\mathbf{x}, t_i)$ and corresponding snapshots $\mathbf{u}_i \in \mathbf{X}$, where i = 0,...,$N_{\text{train}}$; reference configuration index ; InterpNet and ShiftNet architecture and respective criterion $\epsilon_{\text{interp}}, \epsilon_{\text{shift}}$ for loss functions.

Result: Transformed snapshots at reference frame and $\mathcal{T}_{\text{shift}}$ optimal shift operator.

while $\mathcal{L}_{InterpNet} > \epsilon_{interp}$ **do**
  InterpNet.forward ;
  compute $\mathcal{L}_{\text{InterpNet}}$;
  InterpNet.backward ;
end
$\mathcal{T}_{\text{InterpNet}}$ = **InterpNet**.forward; ▷ reconstructs field values
for $\mathbf{u}(t_i) \in \mathbf{X}$, *where i = 0* **to** N_{train} **do**
  while $\mathcal{L}_{ShiftNet} > \epsilon_{shift}$ **do**
    $\tilde{\mathbf{x}}_i$ = $\mathbf{x}_i$ - **ShiftNet**.forward ; ▷ x is physical space (x = $\mathtt{x_{ref}}$)
    $\tilde{\mathbf{u}}_i = \mathcal{T}_{\text{interp}}(\tilde{\mathbf{x}}_i)$; ▷ reconstruct field values at each shifted space
    compute $\mathcal{L}_{ShiftNet}$;
    ShiftNet.backward ;
  end
  $\mathcal{T}_{\text{shift}}$=**ShiftNet**.forward; ▷ optimal shift operator
  $\tilde{\mathbf{x}}_i^{ref}$ = $\mathbf{x}$ - $\mathcal{T}_{\text{shift}}$; ▷ shifted space that transport snapshot to reference frame
  $\mathcal{L} = \mathcal{I}(\tilde{\mathbf{x}}_i^{ref}, \tilde{\mathbf{u}}_i)$; ▷ interpolator
  $\tilde{\mathbf{u}}_i^{ref} = \mathcal{L}(\mathbf{x})$; ▷ transformed snapshots in physical space
end

6.5 NNsPOD-ROM algorithm

In this section, we present the implementation of the complete NNsPOD-ROM algorithm, which includes both the offline and online stages for constructing efficient and accurate ROMs for advection-dominated problems. As outlined in Algorithm 2, the NNsPOD-ROM algorithm is divided into two stages: 1) FOM Pre-processing (Offline Stage): This stage involves the neural network shift-augmented transformation (which includes the training of neural networks and manifold transformation) and the construction of the ROM on the transformed linear subspace, and 2) FOM Post-processing (Online Stage): This stage focuses on the testing and prediction phase.

FOM Pre-processing (offline stage): At first, we collect input parameters $(\mathbf{x}, t_i)$ and their corresponding snapshots $\mathbf{u}_i$, which are required for training. We then select a reference configuration, specify the neural network architectures for InterpNet and ShiftNet, and define their stopping criteria. Initially, we obtain the transformed snapshots matrix $\mathcal{X}$ and the shift operator $\mathcal{T}_{\text{shift}}$ by following Algorithm 1, which reports neural network shift augmented transformation. After this step, all the transformed snapshots are at the reference frame and in the same grid distribution $\mathbf{x} \in \mathbb{V}$. Finally, we perform linear dimensionality reduction on the transformed snapshots matrix, to achieve sharp KnW decay and obtain a low-dimensional transformed linear approximation subspace. We obtain the r number of most energetic POD modes to construct ROM and follow the framework outlined in section 6.3 to construct non-intrusive or data-driven ROMs.

FOM Post-processing (online Stage): Initially, the predicted snapshot field values from the constructed ROM are in the reference frame (where the ROM is built). Therefore, shifting or transporting these field values back to the correct frame in physical space is necessary. This is accomplished by computing the shifted space $\tilde{\mathbf{x}}_i^{shift} = \mathbf{x} + \mathcal{T}_{\text{shift}}(\mathbf{x}, t_i)$, which transports/maps the predicted snapshot field values to the correct frame. Here, we highlight that we are employing *automatic shift detection* for predictions in the online phase. The predicted snapshot field values are transported to the correct frame utilising the shifted space $\tilde{\mathbf{x}}_i^{shift}$. Subsequently, the transported field values are then reconstructed in physical space $\mathbf{x}$ using a linear interpolator $\mathcal{I}$. The predicted snapshot field values intensity variations in the correct frame/location in the physical space $\mathbf{x} \in \mathbb{V}$.

To assess the accuracy of the predictions, we compute the relative L_2 error of the predicted snapshots $\mathbf{u}^*$, defined as in (6.19). Here, $\mathbf{u}_i$ represents the original or true field value, and $\mathbf{u}_i^*$ is the ROM-predicted field value at the i^{th} cell.

$$\mathcal{L}_2 = \frac{\left\|\mathbf{u} - \mathbf{u}_i^*\right\|}{\|\mathbf{u}\|} = \frac{\sqrt{\sum_{i=0}^{N_h} \left(\mathbf{u}_i - \mathbf{u}_i^*\right)^2}}{\sqrt{\sum_{i=0}^{N_h} \left(\mathbf{u}_i^2\right)}}. \tag{6.19}$$

The implementation of both Algorithm 1 and Algorithm 2 is carried out using the `EZyRB` package [50], a `Python` library for non-intrusive data-driven model order reduction of parameterized problems, and here, the deep neural networks are implemented using the `PyTorch` package [139]. A schematic representation of the complete workflow of the NNsPOD-based ROM algorithm is

Algorithm 2: NNsPOD-ROM

Data: Inputs $(\mathbf{x}, t_i)$ and corresponding snapshots $\mathbf{u}_i \in \mathbf{X}$, where $i = 0, \ldots, N_{\text{train}}$;
r number of POD modes; reference configuration index;
InterpNet and ShiftNet architecture and respective criteria $\epsilon_{\text{interp}}, \epsilon_{\text{shift}}$ for loss functions.
Result: NNsPOD-ROM prediction of snapshots for given parameter t_i
FOM pre-processing: offline stage
Obtain transformed snapshots matrix $\mathcal{X}$ and shift operator $\mathcal{T}_{\text{shift}}$ using Algorithm 1;
POD = POD(rank = r); ▷ returns r POD modes
s = POD.singular_values; ▷ get singular values
$\tilde{\mathbf{u}}_i^{ref}$ = ROM($\mathcal{X}$, POD, •); ▷ construct ROM, • = Linear, GPR, RBF, or ANN
FOM post-processing: online stage
for t_i, *where* $i = 0$ **to** N_{test} **do**
  $\tilde{\mathbf{u}}_i^{ref} = \text{ROM.predict}(t_i)$; ▷ predictions in reference frame
  $\tilde{\mathbf{x}}_i^{shift} = \mathbf{x} + \mathcal{T}_{\text{shift}}$; ▷ shifted space transports back to physical frame
  $\mathcal{L} = \mathcal{I}(\tilde{\mathbf{x}}_i^{shift}, \tilde{\mathbf{u}}_i^{ref})$; ▷ interpolator
  $\mathbf{u}_i^* = \mathcal{L}(\mathbf{x})$; ▷ predicted field values in physical frame
end
Get $\mathbf{u}_i^*$ in physical frame and compute relative L_2-norm error;
$\mathbf{u}_{\text{new}} = \text{ROM.predict}(t_{\text{new}})$; ▷ predictions for unseen new parameters

shown in Figure 6.1.

6.6 Numerical Experiments

In this section, we evaluate the effectiveness of the proposed NNsPOD-ROM methodology by applying it to the construction of efficient and accurate ROMs for advection-dominated problems. The methodology is demonstrated by considering three test cases: a 1D linear advection equation, a popular higher-order methods benchmark case—a 2D Isentropic convective vortex, and a 2D Two-phase flow, focusing on capturing the interface.

6.6.1 1D Linear advection equation

We consider the 1D linear advection equation (6.20), where $u(x, t)$ represents the scalar quantity (i.e, concentration, temperature, etc.) being transported as a function of space $x \in \Omega$ and time $t \in T$. The constant c is the advection velocity

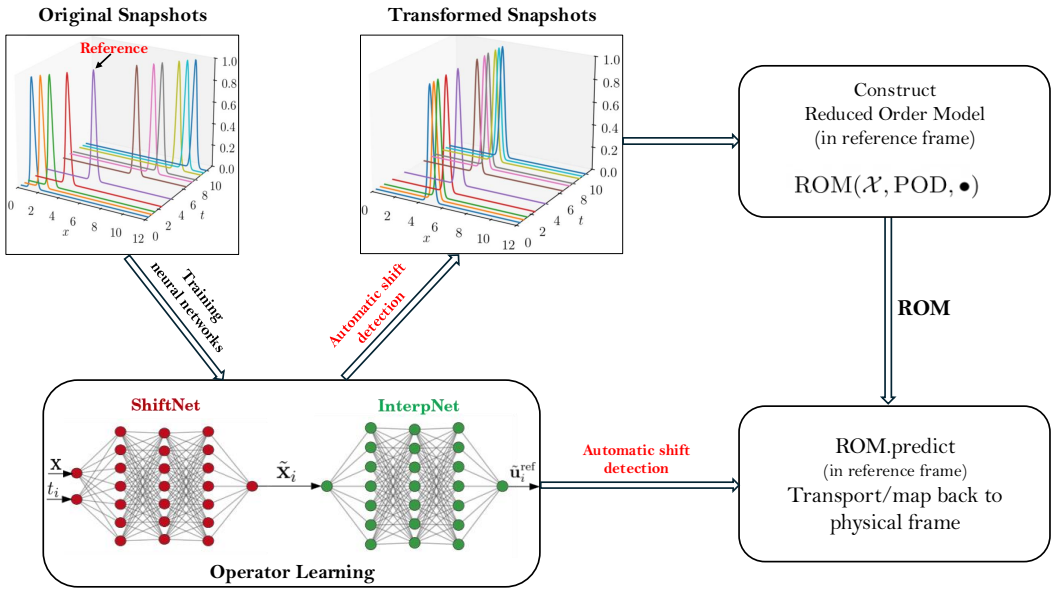


Figure 6.1: Schematic representation of the complete workflow for constructing NNsPOD-ROMs. The offline stage consists of training neural networks, manifold transformation, and construction of non-intrusive ROM. In the online stage, automatic shift detection is employed to transport back the predicted snapshot field values to the correct frame in physical space. To illustrate the workflow, we consider 1D travelling waves, referred to as the Original Snapshots. During the training phase, we first minimize the $\mathcal{L}_{\text{InterpNet}}$ (6.17) of InterpNet to learn the $\mathcal{T}_{\text{Interp}}$ operator, which functions as an interpolator to reconstruct the reference configuration field values. Subsequently, we train ShiftNet, where the $\mathcal{T}_{\text{Interp}}$ operator is used to reconstruct field values in the shifted space (new grid distribution). The loss function $\mathcal{L}_{\text{ShiftNet}}$, as defined in (6.18), is minimized to learn the optimal shift operator $\mathcal{T}_{\text{ShiftNet}}$. Further to perform manifold transformation, the optimal shift operator $\mathcal{T}_{\text{ShiftNet}}$ (automatic shift detection) is employed to transport/map numerous snapshots to the reference frame. Then the transformed snapshots' field values are reconstructed in discrete physical space $\mathbf{x}$ (referred to as Transformed snapshots) using a linear interpolator. This process accelerates KnW decay and results in a low-dimensional transformed linear approximation subspace. The non-intrusive ROM is built on this low-dimensional transformed linear subspace. In the online stage, the predicted snapshot field values are initially at the reference frame (where ROM is constructed), and therefore, automatic shift detection is employed to transport/map back the field values to the physical frame. The transported field values are reconstructed in physical space using a linear interpolator. At this stage, the predicted snapshot field values intensity variations are in the physical frame.

representing the rate at which the quantity is transported in the x direction.

$$\frac{\partial u(x,t)}{\partial t} + c\frac{\partial u(x,t)}{\partial x} = 0, \quad x \in \Omega. \tag{6.20}$$

For a given initial condition $u(x,0) = f(x)$ at $t = 0$, the solution to the linear advection equation is $u(x,t) = f(x-ct)$. A commonly used initial condition in transport problems is a Gaussian function (6.21) [55, 60], parameterized by its mean μ and standard deviation σ. In this study, we utilize the linear advection equation to describe the evolution of a Gaussian wave. The initial condition is defined as:

$$u(x,0) = f(x) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{6.21}$$

where mean μ represents the initial position of the wave, while standard deviation σ determines the width of the wave. In this case, both values are kept constant, $\mu = 0.75$ and $\sigma = 2$. The solution to the linear advection equation with this initial condition is obtained in closed form. As we are dealing with time-dependent problems, we parametrize time within the range $t \in [0, 10]$. At each time instance t, the quantity $u(x,t)$ simply shifts along the x-axis at a constant rate $c = 1$ without changing its shape. We considered this test case to illustrate the complete workflow of the proposed algorithm. We generate a database consisting of 21 snapshots at uniformly sampled time instances $t \in [0, 10]$, over 256 equidistant nodes $x \in [0, 12]$, as shown in Figure 6.2. We divide 50% of the database as a training set (Figure 6.4, Left) and the rest as a testing set.

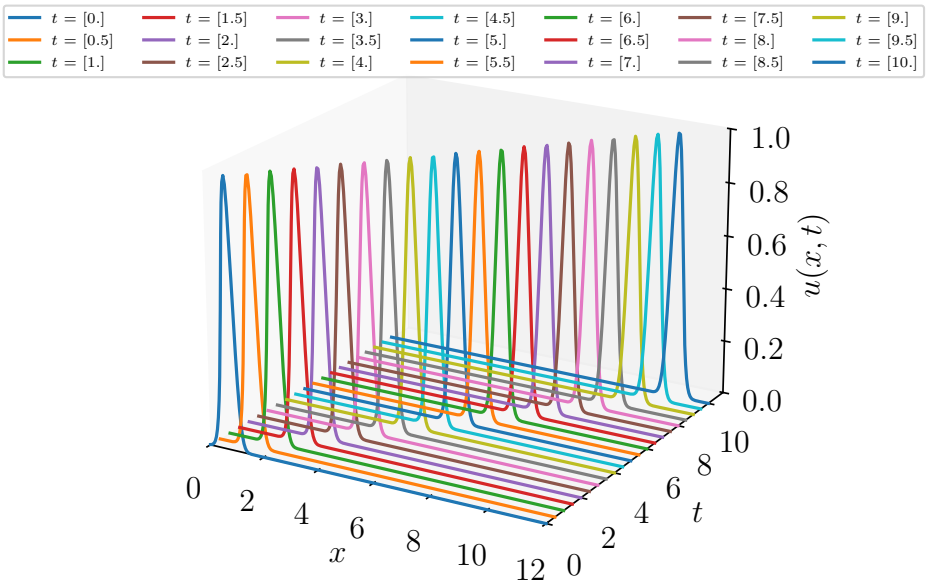


Figure 6.2: Snapshots of the 1D linear advection equation collected at 21 uniformly sampled time instances, $t \in [0, 10]$, over 256 equidistant spatial nodes $x \in [0, 12]$.

For training neural networks, the snapshot corresponding to the reference parameter t_{ref} = 4 from the training set is selected as the reference configuration. The settings for the two neural network architectures used in this test case are shown in Table A.1. InterpNet learns to reconstruct the field values of the reference snapshot as shown in Figure 6.3 (left). This is utilised in the ShiftNet training to learn the optimal shift operator, which yields the shifted space that transports/maps the original snapshot field values to the reference frame/configuration. In Figure 6.3 (right), we show the original snapshot field values mapped to the reference frame utilizing the respective shifted space for the parameter t = 10, is labelled as Shifted, the original snapshot of the same parameter t = 10, is labelled as physical and the reference configuration is labelled as reference. It can be noticed in Figure 6.3 (right) that the shifted snapshot field values don't lie in the same discretized domain $x \in [0, 12]$ or physical space. As mentioned in the previous section, the shifted/transformed snapshot field values must be reconstructed in physical space using interpolation to achieve sharp KnW decay. In Figure 6.4, the comparison between the original snapshots of the training set (left) and the reconstructed shifted snapshots at the reference frame in physical space is shown.

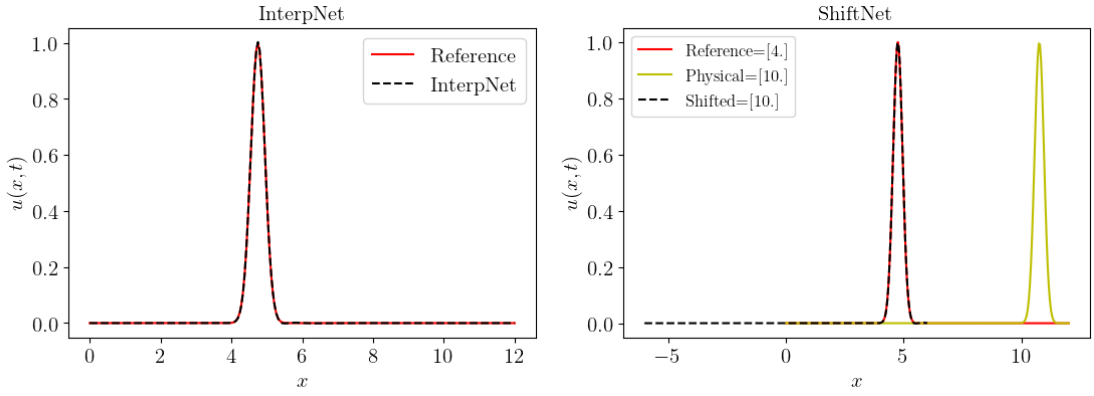


Figure 6.3: For 1D linear advection equation, **Left**: Showing trained InterpNet, able to reconstruct the field values of reference snapshot corresponding to time instance, t_{ref} = 4 and **Right**: The original snapshot at time t = 10 (labelled as physical), the same transported to the reference frame utilising the shifted space (labelled as shifted), using optimal shift obtained by shift operator $\mathcal{T}_{\text{shift}}$ and the reference frame is also shown for comparison.

After the neural-network shift augmented transformations of the training set database, all the snapshots are transported/mapped to the reference frame, as shown in Figure 6.4. The low-dimensional linear approximation subspace is then obtained by performing singular value decomposition (SVD) of the transformed snapshot matrix $\mathcal{X}$. Figure 6.5 demonstrates the effectiveness of

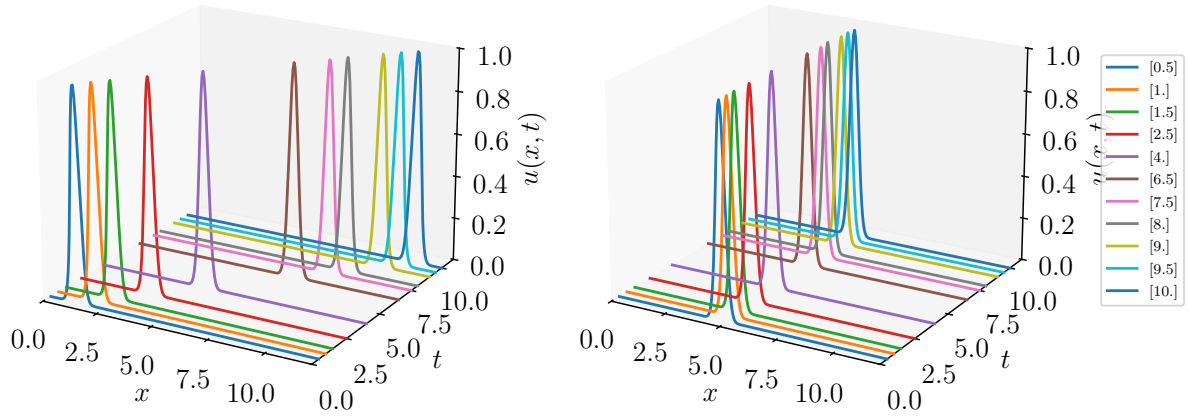


Figure 6.4: For 1D linear advection equation, **Left**: showing training set snapshots in physical space x and **Right**: the shifted snapshots at the reference frame in physical space after neural network shift augmented transformation.

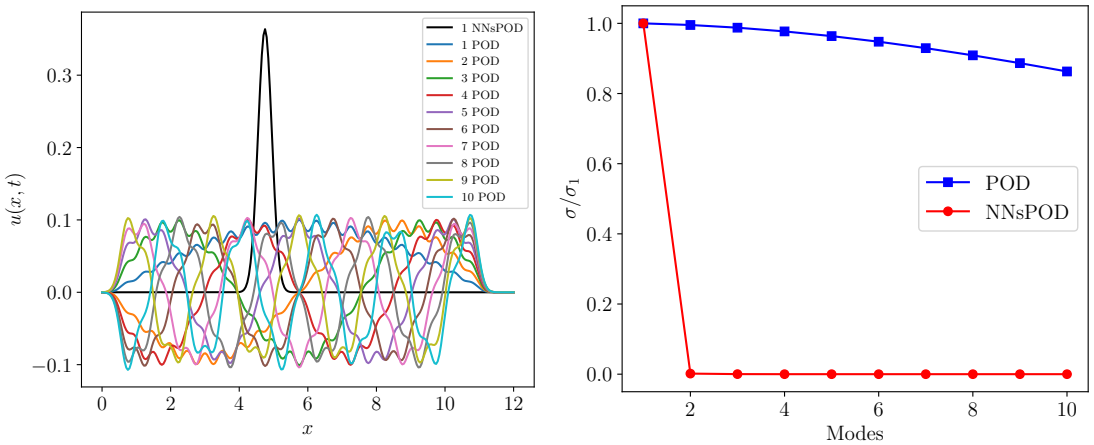


Figure 6.5: For 1D linear advection equation, **Left**: Comparison of the first NNsPOD mode, obtained via neural-network shifted augmented manifold transformation, with all POD modes of the full-order model solutions. and **Right**: Singular value decay comparison, showing POD (without transformations) and NNsPOD (with the neural-network shifted augmented manifold transformation)

NNsPOD over transitional POD, showing the sharp singular values decay of transformed matrix $\mathcal{X}$ and the most energetic first NNsPOD mode, capturing 99.99% of the cumulative energy. In contrast, achieving the same energy requires all the POD modes. Furthermore, the POD modes fail to capture the shape of the Gaussian wave, whereas NNsPOD captures the Gaussian wave accurately. To construct efficient ROMs, the modal coefficients are computed by projecting the transformed snapshot matrix $\mathcal{X}$ onto the NNsPOD subspace.

To obtain fast and accurate predictions for given unseen parameters, we build parameters to modal coefficients map using radial basis function (RBF) interpolation. Then, we obtain the snapshot for unseen parameters by simple matrix multiplication. Initially, the predicted snapshots are in the reference frame. As discussed in the previous section, we employ automatic shift detection to transport/map back the predicted snapshot field values to the physical frame.

In Figure 6.6, the predicted snapshot field values (solid line) for test set parameters in the physical frame are compared with full-order linear advection equation solutions (dotted line). The reconstruction errors for the training and test set parameters are shown in Figure 6.7, with the error magnitudes exhibiting similar variations. This demonstrates that NNsPOD-RBF-ROM predicts reliable accurate snapshots for the new unseen parameter. In Figure 6.8, we compare the mean relative L^2 reconstruction error of standard POD-RBF-ROM and NNsPOD-RBF-ROM against the modes rank for test set parameters of the linear advection equation. By considering the first NNsPOD mode with a cumulative energy of 99.99%, the NNsPOD-RBF-ROM provides accurate predictions, outperforming standard POD-RBF-ROM. Even when all POD modes are considered, the performance of the standard POD-RBF-ROM remains significantly inadequate, being ill-suited for advection-dominated problems.

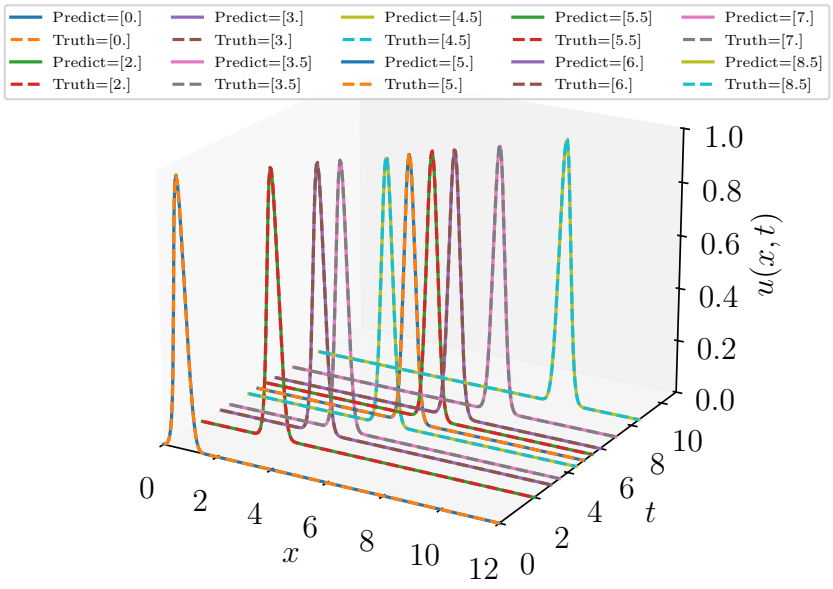


Figure 6.6: Prediction of NNsPOD-RBF-ROM (considered first NNsPOD mode with 99.99% cumulative energy) for test parameters, labelled as Predict (solid lines), compared with the full-order solutions of 1D linear advection equation, labelled as Truth (dashed lines).

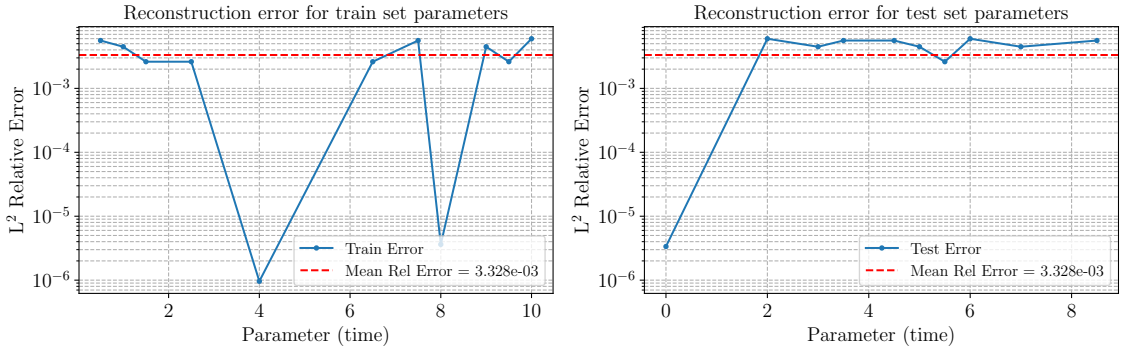


Figure 6.7: Relative L^2 reconstruction error of predictions obtained by NNsPOD-RBF-ROM for the 1D linear advection equation, shown for training set parameters (left) and test set parameters (right). The red line on the left indicates the mean training error, while the red line on the right represents the mean test error, demonstrating that the resulting ROM exhibits excellent generalization capabilities.

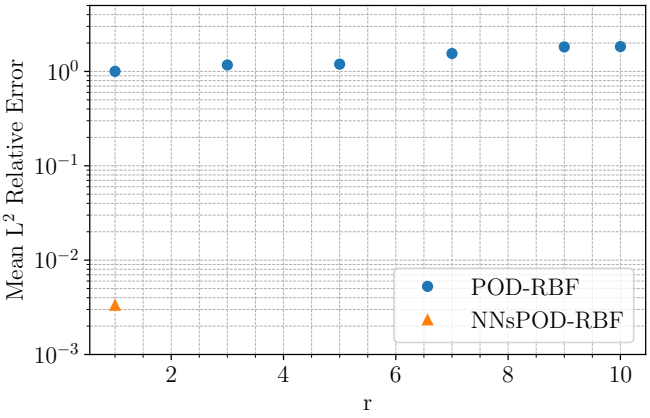


Figure 6.8: Mean L^2 relative reconstruction error comparison of POD-RBF-ROM and NNsPOD-RBF-ROM predictions for test set parameters of 1D linear advection equation.

6.6.2 2D Isentropic convective vortex

We consider the popular higher order methods benchmark case [179, 203], isentropic convective vortex to test the proposed NNsPOD-ROM methodology. The governing equations considered are the 2D Unsteady Euler equations (6.22), to describe the motion of the inviscid, compressible flow of our benchmark case.

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{bmatrix}=0. \tag{6.22}$$

where ρ is density, (u, v) are velocity x and y component, p is pressure and E is total energy. The thermodynamic closure is obtained by the total energy equation (6.23) and in our case we consider constant specific heat ratio, $\gamma = 1.4$ and gas constant, $R_{gas} = 287.15$ J/Kg.K.

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2). \tag{6.23}$$

The initial condition of the freestream flow is given by $\rho_\infty = 1, u_\infty = 0.1, v_\infty = 0, p_\infty = 1$ and the vortex is imposed in the domain by considering the analytical solution as in (6.24), where vortex strength $b = 0.5$, the radius of the vortex, $r = \left[(x - x_c)^2 + (y - y_c)^2\right]^{1/2} = 0.5$ and vortex centre $(x_c, y_c) = (5, 10)$. There are different versions of the vortex problem available in the community [179, 170], we consider the version implemented in the package HyPar 1.0 - Finite Difference Hyperbolic-Parabolic PDE Solver on Cartesian Grids [48] and the results of full-order solutions are obtained using this package.

$$\begin{cases} \rho = \left[1 - \frac{(\gamma-1)b^2}{8\gamma\pi^2} e^{1-r^2}\right]^{\frac{1}{\gamma-1}}, \\ p = \rho^\gamma, \\ u = u_\infty - \frac{b}{2\pi} e^{\frac{1}{2}(1-r^2)} (y - y_c), \\ v = v_\infty + \frac{b}{2\pi} e^{\frac{1}{2}(1-r^2)} (x - x_c). \end{cases} \tag{6.24}$$

We consider a fifth-order hybrid-compact weighted essentially non-oscillatory (WENO) scheme to approximate the spatial derivatives of the governing equation, WENO scheme belongs to the class of higher-order accurate numerical methods for hyperbolic PDEs and convection-dominated problems [214]. We consider the third-order strong-stability preserving runge-kutta (SSPRK3) method [61], for time integration.

We consider a domain $\Omega = [0, 40] \times [0, 20]$ discretized into 28800 Cartesian grids, the time interval considered for the simulation is $t \in [0, 62.5]$ with a time step, $\Delta t = 0.00625$. We collect snapshots every 100th time-step, yielding $N_T = 100$ number of snapshots and leading to the solution manifold definition $\mathcal{M} := \{\boldsymbol{\rho}(t_i) \in \Omega, i = 1, \ldots, 100\}$, consisting of density field snapshots for 100 instances.

For this numerical experiment, we generate a database consisting of 100 snapshots for each time instance, from which we randomly divide 80% of the database into the training set and the remaining 20% into the test set. Before training, we scale the database to range between 0 and 1 using the Min-Max Scaler,

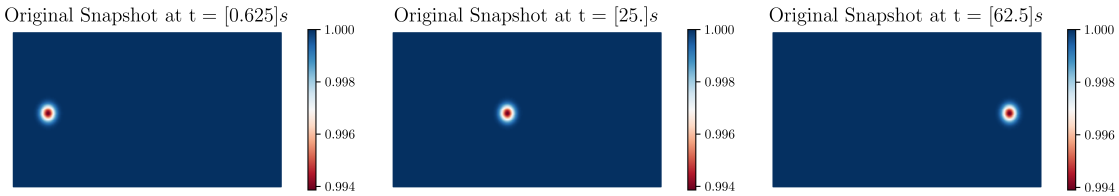


Figure 6.9: FOM solution of isentropic convective vortex at time, t $\in$ $\{0.625, 25, 62.5\}$ and for FOM pre-processing, the original snapshot at t = 0.625 (left) is considered as reference configuration.

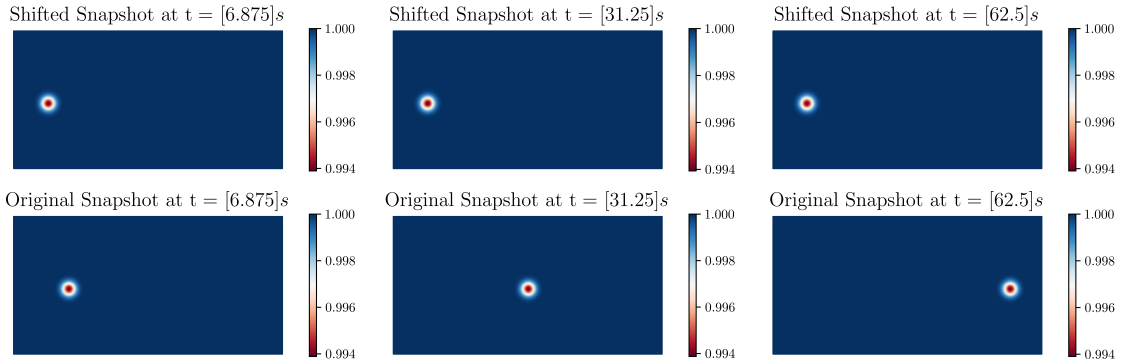


Figure 6.10: Shifted snapshots (top) of isentropic convective vortex after neural-network shift augmented transformations and the respective original snapshot (bottom) at time, t $\in$ $\{0.625, 25, 62.5\}$ from the training set.

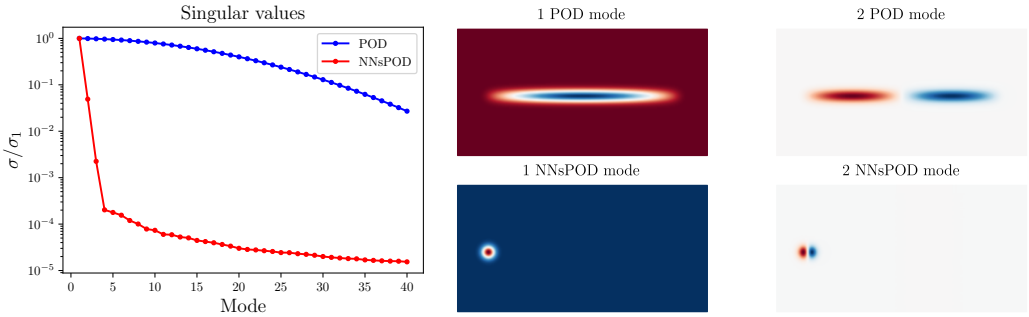


Figure 6.11: **Left**: Comparison of singular values decay of isentropic convective vortex without the transformation, is labelled as POD and after the neural-network shifted augmented manifold transformation, is labelled as NNsPOD, and **Right**: Showing first two modes of FOM manifold without the transformations (first row - POD modes) and after the neural-network shift augmented manifold transformation (second row - NNsPOD modes).

reported in subsection A.1.1. For training neural networks, we select the reference configuration parameter, t = 0.625, the corresponding snapshot is shown in Figure 6.9 (left). In this case, only the x-ordinates are shifted, as the vortex is convected in the x-direction. The settings considered for two neural network architectures for this test case are as given in Table A.2. InterpNet learns

to reconstruct the reference snapshot field values and is utilised in ShiftNet training to learn the optimal shift operator that yields optimal shifted space, which maps the original snapshot field values to the reference frame. After the neural network shift augmented manifold transformation, we obtain shifted snapshots that are transported/mapped to the reference frame as shown in the Figure 6.10.

We then perform linear dimensionality reduction on this transformed manifold, to obtain sharp singular values decay as shown in Figure 6.11 (left), compared with singular values obtained without the manifold transformation. In Figure 6.11(right), the first two modes of the FOM manifold without transformations and after the neural network shift augmented transformations are shown. It can be observed that NNsPOD accurately captures the vortex structure, whereas the standard POD fails to do so. Furthermore, the first 4 NNsPOD modes account for 99.99% of the cumulative energy, whereas 40 POD modes are required to constitute the same energy. We consider the first 4 NNsPOD modes to build efficient NNsPOD-RBF-ROM, and its predictions (rescaled to the original range) for unseen test set parameters are shown in Figure 6.12, with the original snapshots (Truth) and the corresponding absolute error. The reconstruction error for the training and test set parameters is

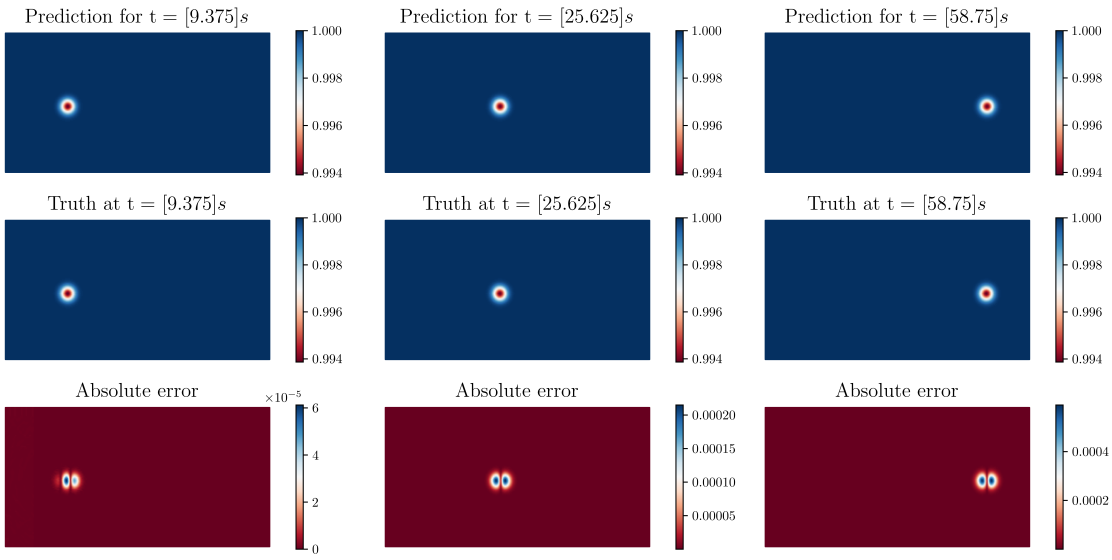


Figure 6.12: Predictions of snapshots by NNsPOD-RBF-ROM of isentropic convective vortex test set parameters, showing for time instances t = {9.375, 25.625, 58.75}s.

shown in Figure 6.13. We plot the relative L^2 error of the predicted snapshots for each respective parameter and the mean relative L^2 error. The error varies

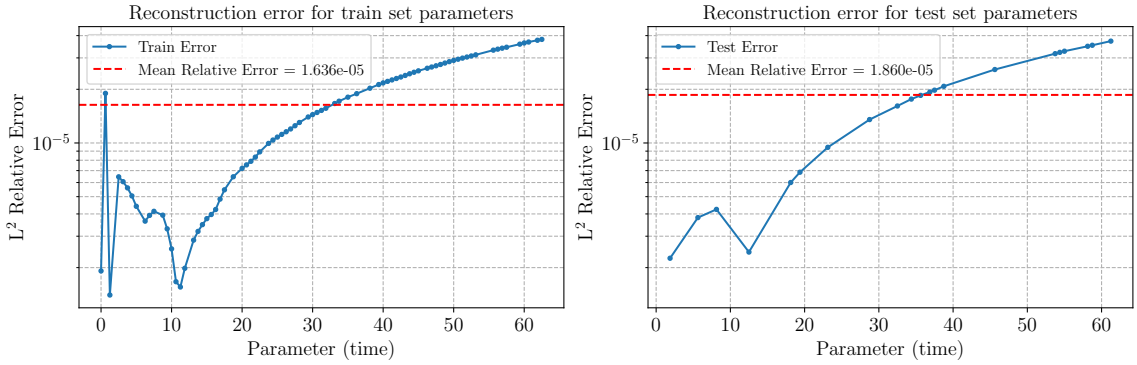


Figure 6.13: Relative L^2 reconstruction error of predictions by NNsPOD-RBF-ROM for Isentropic convective vortex, for both training set parameters (left) and test set parameters (right).

in the same order of magnitude for both train and test set parameters predictions, with the error trend increasing for the parameters further from the chosen reference configuration parameter. In Figure 6.15, we compare the mean relative L^2 reconstruction error of the standard POD-RBF-ROM and NNsPOD-RBF-ROM to modes rank, for test set parameters of 2D isentropic convective vortex. The NNsPOD-RBF-ROM obtained by considering the first 4 NNsPOD modes that account for 99.99% of the cumulative energy, provides accurate predictions and effectively captures the vortex structure. The standard POD-RBF-ROM fails to adequately represent the vortex structure, even when 10 POD modes are considered, as shown in Figure 6.14. Overall, NNsPOD-RBF-ROM outperforms standard POD-RBF-ROM qualitatively and quantitatively, providing exceptional reconstruction accuracy and a more precise representation of the vortex structure via a low-dimensional transformed linear approximation subspace.

6.6.3 2D Two-phase flow

Two-phase flows are governed by the unsteady Navier-Stokes equation (6.25), with continuity and momentum equation for incompressible fluids, where $\mathbf{U}$ is velocity field, ρ is density, p is pressure and $\mathbf{F}_{ST}$ is the volumetric surface tension.

$$\begin{cases} \partial_t(\rho) + \nabla \cdot (\rho \mathbf{U}) = 0, \\ \partial_t(\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = -\nabla p + \nabla \cdot [\mu(\nabla \mathbf{U} + \nabla^T \mathbf{U})] + \mathbf{F}_{ST}. \end{cases} \tag{6.25}$$

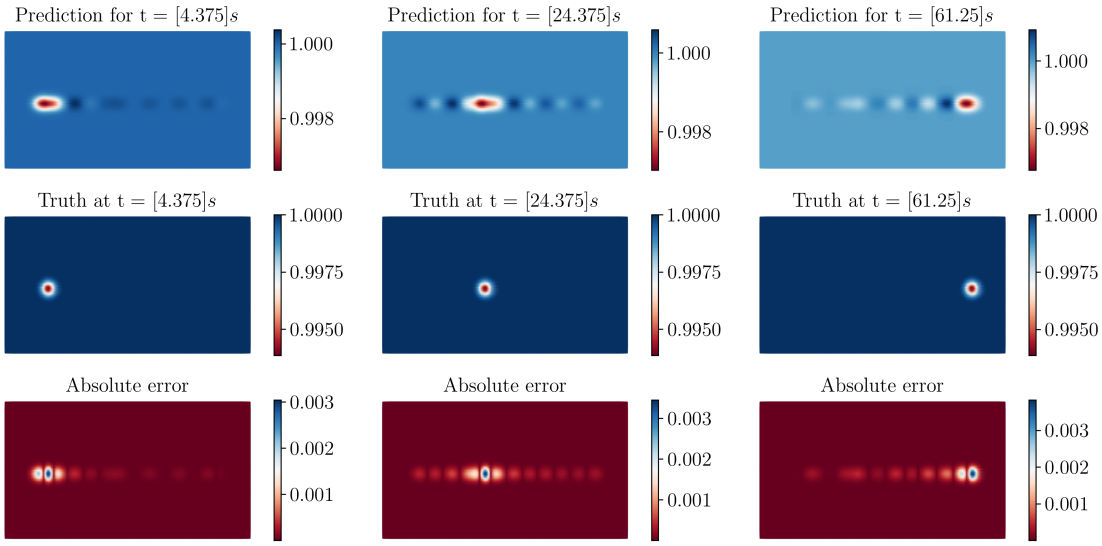


Figure 6.14: Predicted snapshots by the POD-RBF-ROM for test set parameters of the isentropic convective vortex, considering 10 POD modes, shown for time instances t = $\{4.375, 24.375, 61.75\}$ s.

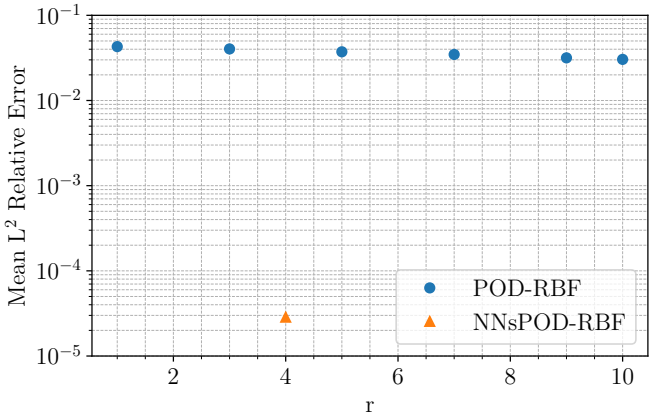


Figure 6.15: Mean L^2 relative reconstruction error comparison of POD-RBF-ROM and NNsPOD-RBF-ROM predictions for test set parameters of 2D isentropic convective vortex.

For capturing the interface between two-phase flows we consider the one-fluid models approach, which are volume-of-fluids (VOF), level-set (LS) and phase-field (PF) methods. In the one-fluid model approach, two immiscible fluids are considered as one effective fluid, in which $\mathbf{U}$ acts as the shared velocity of the two fluids $\mathbf{U} = \mathbf{U}_1 = \mathbf{U}_2$. The physical properties of the fluids are averaged over the domain, which leads to constituent relations of ρ and μ weighted values given by:

$$\begin{aligned} \rho &= \alpha_1 \rho_1 + (1 - \alpha_1)\, \rho_2, \\ \mu &= \alpha_1 \mu_1 + (1 - \alpha_1)\, \mu_2. \end{aligned} \tag{6.26}$$

Here, we use the VOF method, in which the modified volume fraction advection equation (6.27) is solved simultaneously with continuity and momentum equations. Where $\alpha \in [0,1]$ is phase fraction which takes values $\alpha = 0$ for gas, $\alpha = 1$ for liquid and at the interface, it ranges between $0 < \alpha < 1$, and $\mathbf{U}_r$ is the relative velocity shown in (6.28), which is perpendicular to the interface. To solve this advection equation, we use a semi-implicit MULES algorithm.

$$\partial_t \alpha + \nabla \cdot (\alpha \mathbf{U}) + \nabla \cdot [\mathbf{U}_r \alpha (1-\alpha)] = 0, \tag{6.27}$$

$$\mathbf{U}_r = C_\alpha |\mathbf{U}| \frac{\nabla \alpha}{|\nabla \alpha|}, \qquad \text{with } C_\alpha \in [0,1]. \tag{6.28}$$

To model the turbulence, we consider $k-\epsilon$ turbulence model based on [106], which is a two-equation linear eddy viscosity turbulence closure model consisting of k - turbulent energy and ϵ - turbulent kinetic energy dissipation rate equations:

$$\begin{cases} \partial_t(\rho k) = \nabla \cdot (\rho D_k \nabla k) + P - \rho\epsilon, \\ \partial_t(\rho \epsilon) = \nabla \cdot (\rho D_\epsilon \nabla \epsilon) + \frac{C_1 \epsilon}{k} \left(P + C_3 \frac{2}{3} k \nabla \cdot \mathbf{U}\right) - C_2 \rho \frac{\epsilon^2}{k}. \end{cases} \tag{6.29}$$

$$\nu_t = C_\mu \frac{k^2}{\epsilon}. \tag{6.30}$$

where, D_k - effective diffusivity for k, P - turbulent kinetic energy production rate, D_ϵ - effective diffusivity for ϵ, C_1, C_2, C_3 are model coefficients. After obtaining k, ϵ, values, the turbulent viscosity, ν_t is given by (6.30), here C_μ is the model coefficient for turbulent viscosity. We consider the standard model coefficient values shown in Table 6.1. We consider water ($\rho_1 = 10^3, \nu_1 = 10^{-6}$) and

C_μ	C_1	C_2	C_3	σ
0.09	1.44	1.92	0	1

Table 6.1: Model coefficients of $k-\epsilon$ turbulence model.

air ($\rho_2 = 1, \nu_2 = 1.48 \times 10^{-5}$) as two fluids for this two-phase flow test case, and consider constant inflow of water into the domain with inlet velocity **U**(x, 0, 0) = (0.5, 0, 0). We consider a domain $\Omega = [-0.25, 2] \times [-0.25, 1]$ discretized into 2856 hexahedra cells, the time interval considered for the simulation is $t \in [0, 40]$ with a time step $\Delta t = 0.05$, yielding $N_s = 800$ number of snapshots and leading to the solution manifold definition:

$$\mathcal{M}_h := \{\boldsymbol{\alpha}_h(t_i) \in \mathcal{V}_h, i = 1, \ldots, 800\}.$$

We construct a database consisting of 800 snapshots, each corresponding to a specific time instance, where time is parameterized. The discretized domain used for the two-phase flow test case is shown in Figure 6.16. The inlet and outlet are highlighted in green and red, respectively. We emphasize that fluid enters only through the inlet, while the outflow at the outlet is restricted. The full-order solutions of the two-phase flow case are obtained using the interFoam solver [51] of OpenFOAM [207], which is based on the FVM. In Figure 6.17, we show the velocity magnitude contours at times $t \in \{10, 20, 40\}$. These plots illustrate the flow complexity and help characterize the convective field. The original snapshots at a few time instances are shown in Figure 6.18, and here, the inlet region at the bottom left and outlet region at the bottom right of the domain can be depicted. We are interested in capturing the interface of two-phase flow, and hence, we only consider the region of a domain where the interface evolves. For this reason, we neglect inlet and outlet regions, as depicted in Figure 6.19. Then, we randomly divide 80% of the database as a training set and the rest 20% as a test set. For training neural networks, we select the last snapshot (completely filled configuration) at t = 40s as the reference configuration, and the settings considered for the two neural network architectures for this test case are as reported in Table A.3. InterpNet learns to reconstruct the reference configuration, shown in Figure 6.19 (bottom right) at $t = 40$. In this case, since the interface primarily evolves in the y-direction, we incorporate shifts only in the y-coordinates of the spatial domain. ShiftNet learns the shift operator, which finds the optimal shift for a given parameter and is used to transport/map the original snapshot to the reference frame as shown in Figure 6.19.

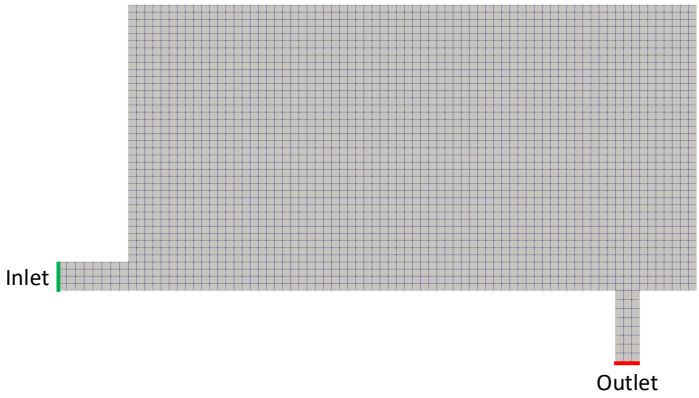


Figure 6.16: Discretized domain used for the two-phase flow test case. The inlet and outlet boundaries are highlighted in green and red, respectively.

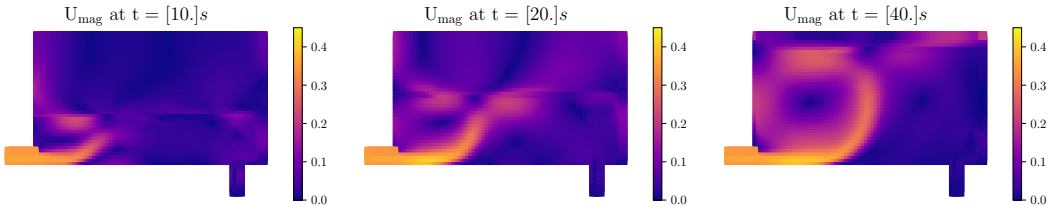


Figure 6.17: Velocity magnitude contour of FOM solutions for the two-phase flow at times t ∈ {10, 20, 40}. We highlight that fluid enters only through the inlet, while outflow through the outlet is restricted.

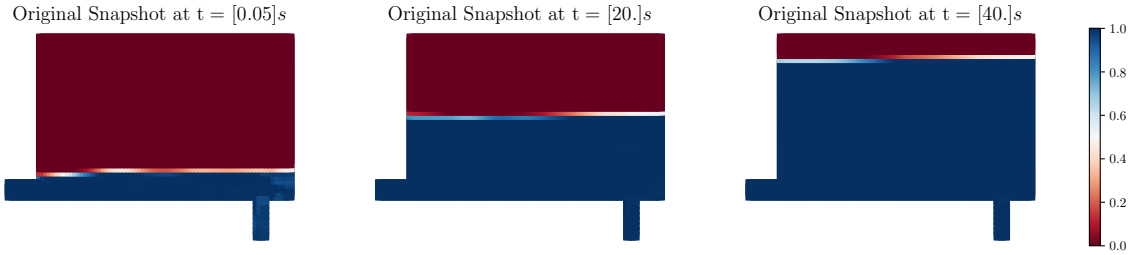


Figure 6.18: FOM solution of two-phase flow at the time, t ∈ $\{0.05, 20, 40\}$ and for pre-processing of the solution manifold, the original snapshot at t = 40 (right figure) is considered as reference configuration.

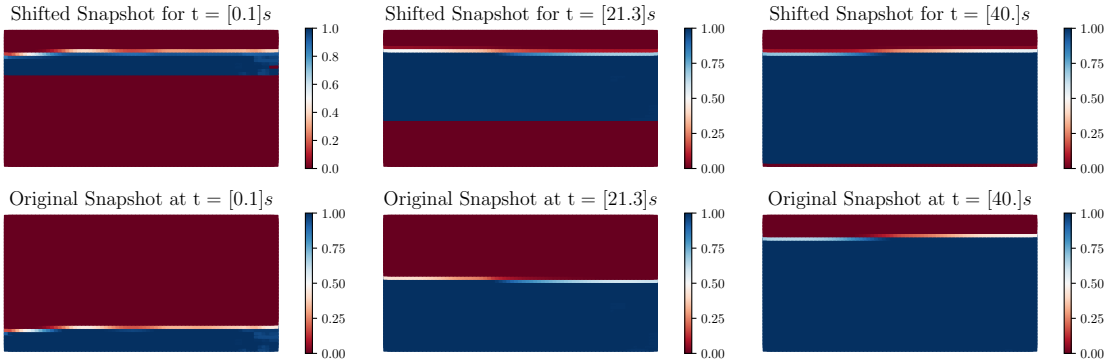


Figure 6.19: Shifted snapshots (top) of two-phase flow, after neural network shift augmented manifold transformations and the respective original snapshot (bottom) at a time, t = $\{0.05, 21.3, 40\}$ from the training set.

After neural-network shifted augmented manifold transformations, and performing model reduction using SVD yields singular values as shown in Figure 6.20. In this case, using neural network shift augmented transformations is not effective, as we don't achieve significant improvement in accelerating KnW decay, and therefore it doesn't yield a suitable low-dimensional linear approximations subspace for efficient ROM construction. To address this, we adopt a different approach: First, we learn the shift operator and transform all the snapshots in the training set to the reference snapshot using interpolation. The customized framework for two-phase flow is presented in Algorithm 3.

Apart from this transformation step, the rest of Algorithm 1 and Algorithm 2 remain unchanged for the construction of efficient ROMs.

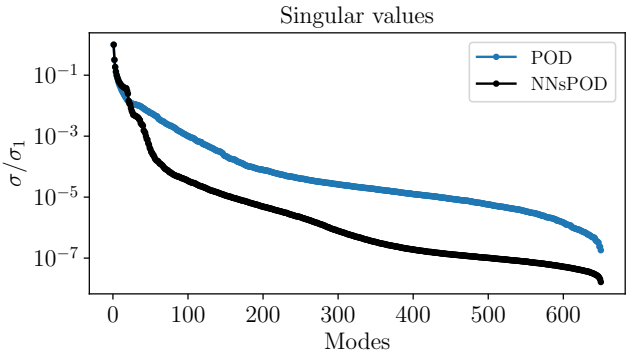


Figure 6.20: Comparison of singular values decay of two-phase flow FOM manifold without pre-processing, is labelled as POD, and after the neural-network shifted augmented manifold transformation, is labelled as NNsPOD

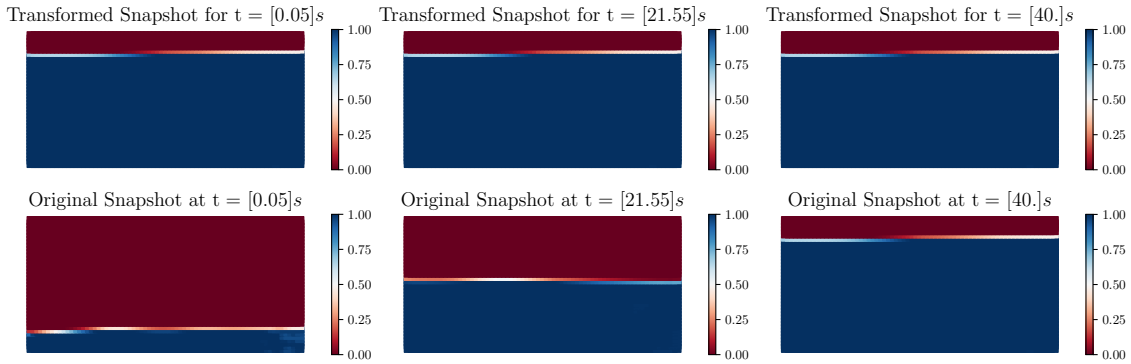


Figure 6.21: Transformed snapshots (top) of two-phase flow, after transforming all the snapshots to the reference snapshot using interpolation to accelerate KnW decay and the respective original snapshot (bottom) from the training set at times, t $= \{0.05, 21.55, 40\}$ are shown.

Algorithm 3: Customized framework for the two-phase flow case.

Data: Inputs $(\mathbf{x}, t_i)$ and corresponding snapshots $\mathbf{u}_i \in \mathbf{X}$, where i = 0,...,$N_{\text{train}}$; reference configuration index ; InterpNet and ShiftNet architecture and respective criterion $\epsilon_{\text{interp}}, \epsilon_{\text{shift}}$ for loss functions.

Result: Transformed snapshots at reference frame and $\mathcal{T}_{\text{shift}}$ optimal shift operator.

while $\mathcal{L}_{InterpNet} > \epsilon_{interp}$ **do**
  InterpNet.forward ;
  compute $\mathcal{L}_{\text{InterpNet}}$;
  InterpNet.backward ;
end
$\mathcal{T}_{\text{InterpNet}}$ = **InterpNet**.forward; ▷ `reconstructs field values`
for $\mathbf{u}(t_i) \in \mathbf{X}$, *where* $i = 0$ **to** N_{train} **do**
  while $\mathcal{L}_{ShiftNet} > \epsilon_{shift}$ **do**
    $\tilde{\mathbf{x}}_i$ = $\mathbf{x}_i$ - **ShiftNet**.forward ; ▷ `x is physical space (x =` $\mathbf{x}_{\text{ref}}$`)`
    $\tilde{\mathbf{u}}_i = \mathcal{T}_{\text{interp}}(\tilde{\mathbf{x}}_i)$; ▷ `reconstruct field values at each shifted space`
    compute $\mathcal{L}_{ShiftNet}$;
    ShiftNet.backward ;
  end
  $\mathcal{T}_{\text{shift}}$=**ShiftNet**.forward; ▷ `optimal shift operator`
  $\mathcal{L} = \mathcal{I}(\mathbf{x}_i, \mathbf{u}_{\text{ref}})$; ▷ `interpolator`
  $\tilde{\mathbf{u}}_i^{ref} = \mathcal{L}(\mathbf{x})$; ▷ `transformed snapshots`
end

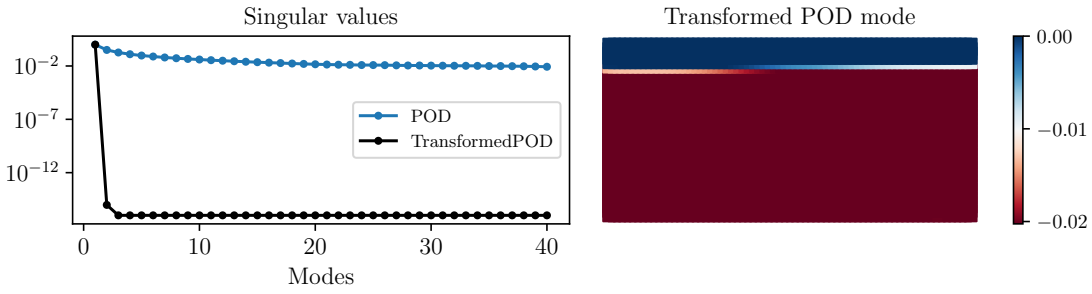


Figure 6.22: **Left**: Comparison of singular values decay of two-phase flow case FOM manifold without transformations, is labelled as POD and after transforming all the snapshots to reference snapshot using interpolation, after learning shift operator, is labelled as TransformedPOD, and **Right**: First TransformedPOD mode obtained after this manifold transformation.

After transforming all the snapshots to reference snapshot, perform reduction using SVD, the singular values, and the first TransformedPOD mode with a

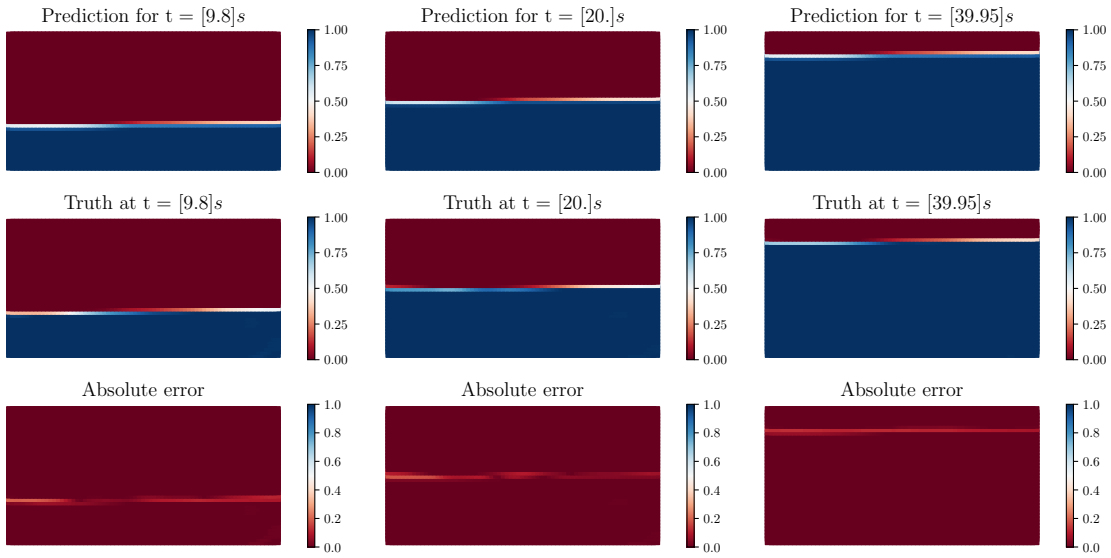


Figure 6.23: Predictions of snapshots for test set parameters obtained by TransformedPOD-RBF-ROM of the two-phase flow test case, showing predictions for parameters (time) t = $\{9.8, 20, 39.95\}$s. To construct this ROM, we use the transformed linear subspace obtained by transforming all the snapshots in the training set to the reference snapshot using interpolation, to accelerate KnW decay, and before that, we learn the shift operator. In this case, except for this transformation step in Algorithm 1, the rest of Algorithm 1 and Algorithm 2 remain the same for the construction of efficient and accurate ROMs.

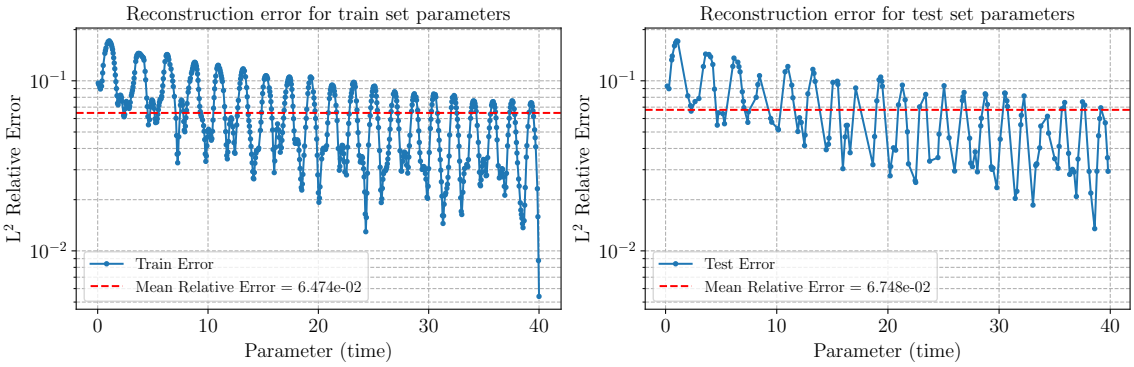


Figure 6.24: Relative L^2 reconstruction error of predictions by TransformedPOD-RBF-ROM of the two-phase flow test case, for both training set parameters (left) and test set parameters (right).

cumulative energy of 99.99 % is as shown in the Figure 6.22, we attain the same energy by considering 70 POD modes. For the construction of ROM, we utilize the first TransformedPOD mode, obtain the temporal coefficient, and use RBF interpolation to build parameters to modal coefficient map. The predictions of TransformedPOD-RBF-ROM for unseen test set parameters are presented in Figure 6.23, the corresponding original snapshot (Truth), and the absolute error. It can be depicted that the TransformedPOD-RBF-ROM predicts

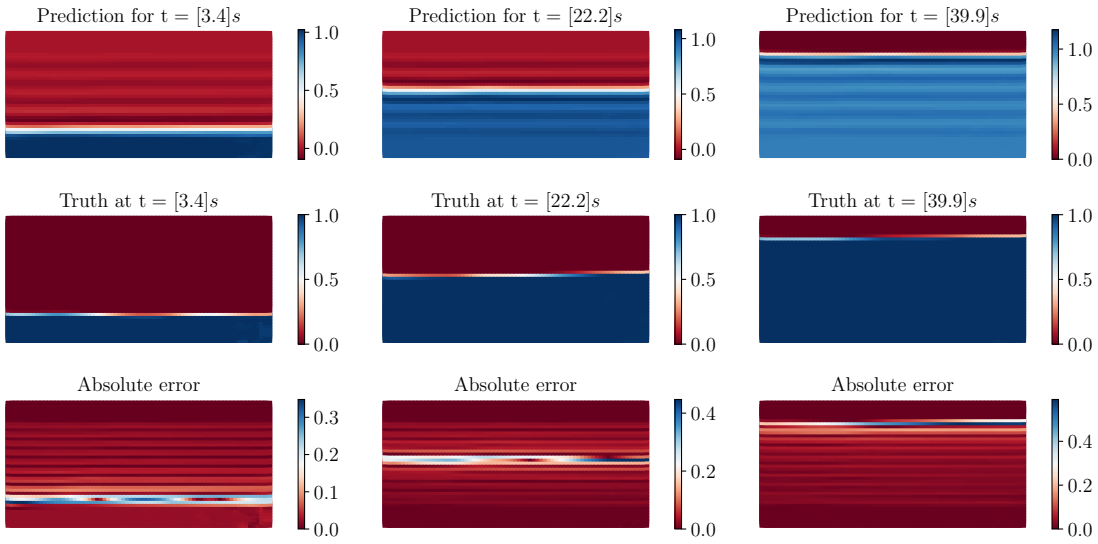


Figure 6.25: Predicted snapshots for test set parameters by the POD-RBF-ROM for the two-phase flow test case, considering 10 POD modes, showing predictions for parameters (time) t = $\{3.4, 22.2, 39.9\}$s.

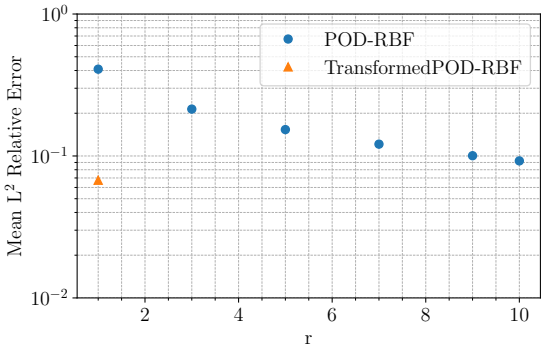


Figure 6.26: Mean L^2 relative reconstruction error of predictions by TransformedPOD-RBF-ROM of the two-phase flow test case, for both training set parameters (left) and test set parameters (right).

the interface accurately and with negligible difference in the field values at the interface region. The reconstruction error for both train set and test set parameters is as shown in the Figure 6.24. The error varies in the same order of magnitude for training and test set parameters. As the interface evolves over the time interval, the interface region wobbles from the left wall to the right wall of the domain (vice-versa), contributing to the observed error trend.

In Figure 6.25, we illustrate the predictions for the test set parameters obtained from the POD-RBF-ROM, constructed using the first 10 POD modes. These results are qualitatively suboptimal. As discussed earlier, our objective is to develop efficient and accurate ROMs using a low-dimensional linear approximation subspace. To achieve this, we employ the first TransformedPOD mode,

enabling the construction of an efficient and accurate TransformedPOD-RBF-ROM. In Figure 6.26, we compare the performance of POD-RBF-ROM and TransformedPOD-RBF-ROM within the low-dimensional linear approximation subspace. While the quantitative results of the POD-RBF-ROM with 10 POD modes are comparable to those of the TransformedPOD-RBF-ROM, the latter demonstrates superior qualitative performance, highlighting its advantage over standard POD-RBF-ROMs. We highlight that the application of this method is limited to a set of problems characterized by coherent, advection-dominated features. It's not suitable for problems with significant diffusion, interface breaking, or chaotic interface dynamics.

We demonstrated the proposed methodology through a series of test cases with increasing complexity. In the first two cases, although the advection velocities were known, this knowledge was deliberately not utilized to perform the manifold transformation. In the two-phase flow case, where the advection velocity governing interface evolution is unknown, the proposed methodology becomes indispensable. As discussed earlier, this approach is particularly valuable for problems where determining advection velocities a priori is either impractical or impossible. It enables the derivation of a low-dimensional transformed linear approximation subspace of the transformed manifolds, efficiently approximating the FOM solutions and facilitating the construction of accurate and efficient ROMs.

6.7 Conclusion

The main focus of this work is the development of a complete NNsPOD-ROM algorithm for advection-dominated problems, comprising both offline-online stages. To extend the applicability of the automatic shift detection procedure of the NNsPOD, which introduces neural-network shift augmented manifold transformations, this approach eliminates the need for comprehensive knowledge of the underlying equations and advection velocities of the given problem. It aims to derive a better linear approximation subspace of the transformed solution manifold, enabling the construction of efficient ROMs for advection-dominated problems on the transformed linear subspace. The key contribution of this work is the construction of ROM on the transformed linear subspace yielded by NNsPOD, the employment of automatic shift detection in the online stage, and we propose the complete NNsPOD-ROM algorithm for

the construction of efficient ROMs. Furthermore, we tested our proposed algorithm on advection-dominated problems of increasing complexity: 1D transport equation, higher order benchmark test case - 2D isentropic convective vortex, and 2D two-phase flow test cases. For 1D transport equation and 2D isentropic convective vortex test cases, the NNsPOD-ROM algorithm results in efficient and accurate ROMs. For the 2D two-phase flow test case, except for the minor adjustment in the transformation step that was required due to the specific configuration of the test case, the rest of the algorithm remains the same resulting in efficient ROM, predicts accurately the interface location, with negligible small differences in field values at the interface region. In the two-phase numerical experiment, the inlet and outlet boundaries are removed because the learned shift operator maps discrete solution points to a reference frame via spatial translation. This mapping is not a diffeomorphism and therefore does not preserve the geometry of complex domains. As a result, geometric inconsistencies may arise if such features are retained. Addressing this limitation by developing a diffeomorphic mapping, as proposed in [215], is an important direction for future work.

Based on the numerical experiments we performed, the proposed approach yields efficient and accurate ROMs for advection-dominated flow problems. It demonstrates excellent generalization, with the training and test set errors exhibiting similar distributions across all cases. While our current experiments focus on unidirectional, the method remains effective even when the solution exhibits shifts in multiple directions. The framework is general and can be extended to handle multidirectional solutions (for example, it is demonstrated in [99]). The flexibility of the approach allows for different neural network architectures to be employed depending on the specific characteristics and complexity of the problem. This flexibility could be leveraged to address such multidirectional shift scenarios, which we plan to explore in future work. Also, extending the proposed method to parametric-time dependent reduced order models is conceptually straightforward. For example, one could parameterize the inlet velocity in the two-phase flow setting. While this is not explored in the current work, we consider it a promising direction for future work. Nevertheless, the application of this approach is limited to specific advection-dominated problems, where the advected features/structure shape should be the same, and this approach is not suitable for advection-diffusion problems. The application of this approach can be extended to developing ROM for particle tracking, for which different neural-network architectures like long short-term

memory (LSTM) or transformers (well suited for time series data) can be employed to learn the optimal shift operator. We postpone this for our future work.

Chapter 7

ROM via cross-correlation based snapshots registration

7.1 Introduction

The previous chapter 6 introduced a fully non-intrusive, data-driven ROM approach based on a neural-network shift augmented manifold transformation framework. This method belongs to the broader class of registration-based techniques, enabling of construction of efficient ROMs using linear methods.

Registration methods are widely used in fields such as medical imaging [47], where aligning complex spatial structures is essential for accurate representation and analysis. Image registration plays a central role in this context, facilitating the alignment of data to a common reference frame. This process typically involves the selection of transformation models, the use of similarity measures, and the application of optimization techniques [110]. The general workflow includes feature detection, correspondence matching, transformation model estimation, and the subsequent resampling and transformation of the input data.

Inspired by these registration strategies, this chapter proposes a simpler, interpretable, and training-free alternative: a cross-correlation-based snapshot registration method. Unlike learning-based approaches that require extensive training data and parameter tuning, the proposed method leverages cross-correlation to determine optimal spatial shifts between snapshots and a reference configuration. This transformation aligns dominant transport structures, accelerating the Kolmogorov n-Width decay, and enables the construction of low-dimensional linear approximation subspaces using classical model reduction techniques such as POD.

A key advantage of this approach is that it does not rely on any a priori knowledge of the underlying transport velocity field, and it does not require optimization or learning algorithms. This makes it particularly appealing for transport-dominated problems in fluid mechanics, where explicit transport information may be unavailable or difficult to extract. It is purely data-driven and operates directly on the snapshot ensemble, making it broadly applicable across a range of transport-dominated scenarios.

This chapter introduces a cross-correlation-based snapshot registration technique for the model reduction of transport-dominated problems. The proposed approach aligns with classical, unimodal image registration methods [47], relying on straightforward, interpretable transformations rather than complex machine learning architectures.

7.2 Cross-correlation based snapshot registration

Cross-correlation measures the similarity between two signals as a function of the displacement or lag between them. It plays a crucial role in signal processing, pattern recognition, time series analysis, and filtering. Cross-correlation aids in identifying or quantifying the relationships between signals and is used to detect signals that are embedded in noise. Image registration through cross-correlation is an essential and widely adopted technique in the field of medical imaging [16, 32]. In this work, we exploit cross-correlation to register the snapshots of transport-dominated problems and aim to accelerate the KnW decay.

Definition 7.1 (Continuous Cross-Correlation)**.** Let us consider two continuous functions $f(t), g(t) \in \mathbb{C}$. The continuous cross-correlation between f and g is defined as:

$$(f \star g)(\tau) \triangleq \int_{-\infty}^{\infty} \overline{f(t)} g(t+\tau)\, dt, \tag{7.1}$$

where $\overline{f(t)}$ denotes the complex conjugate of $f(t)$, and τ is the time lag or displacement. The cross-correlation measures the similarity between $f(t)$ and a time-shifted version of $g(t)$, integrating the product of complex conjugate $\overline{f(t)}$ and the shifted $g(t+\tau)$ over all time.

Definition 7.2 (Discrete Cross-Correlation)**.** Let us consider two discrete signals $f[n], g[n] \in \mathbb{C}^{N_h}$, where N_h is the number of elements in the signals. The

discrete cross-correlation between f and g is defined as:

$$(f \star g)[k] \triangleq \sum_{n=0}^{N_h-1} \overline{f[n]} g[n+k], \tag{7.2}$$

where $\overline{f[n]}$ is the complex conjugate of the discrete signal $f[n]$, n denotes the index of the signal elements, and k is the displacement or lag. The discrete cross-correlation is computed by summing the element-wise product between $\overline{f[n]}$ and the shifted sequence $g[n+k]$. Depending on the context, $g[n+k]$ may be assumed to be zero outside its defined range (zero-padding) or periodically extended (circular correlation).

Cross-correlation offers a robust framework for analyzing the relationship between time-shifted signals. In this work, we leverage cross-correlation snapshot registration, as outlined in Algorithm 4. In this framework, snapshots are treated as discrete signals, and the cross-correlation between each snapshot s_i and the reference snapshot s^{ref} is used to compute the shifts required for registration. Here, we employ the registration of snapshots via cross-correlation to circumvent KnW decay for transport-dominated problems. This method aims to determine the optimal shift for each snapshot, aligning them with the reference based on the maximization of their cross-correlation.

The process of cross-correlations based registration is as follows: First, we select a set of training snapshots $\{s_1, s_2, \ldots, s_{N_{train}}\}$ corresponding to time instances $\{t_1, t_2, \ldots, t_{N_{train}}\}$, and choose one snapshot as the reference s^{ref}. For each training snapshot s_i, we compute the cross-correlation[1] between the reference s^{ref} and the snapshot s_i at various time shifts (lags). Cross-correlation measures the similarity between two signals as a function of their relative time shifts. The optimal shift Δ_i for each snapshot s_i is determined by identifying the shift that maximizes the cross-correlation, i.e., the shift that yields the highest similarity between the reference and the respective snapshot s_i. Once the optimal shift Δ_i is determined for each snapshot s_i, these shifts are employed to align the snapshots with the reference s^{ref}. Then, we store the registered snapshots s_i^{ref} and their corresponding optimal shifts for subsequent steps in developing the ROM.

[1] We use `scipy` python library to compute correlation

Algorithm 4: Registration via cross-correlation

1: **Input:** Set of training snapshots $\{s_1, s_2, \ldots, s_{N_{train}}\}$ for corresponding time instances $\{t_1, t_2, \ldots, t_{N_{train}}\}$, reference snapshot s^{ref}

2: **Output:** Set of Registered snapshots $\{s_1^{\text{ref}}, s_2^{\text{ref}}, \ldots, s_{N_{train}}^{\text{ref}}\}$, corresponding shifts $\Delta = \{\Delta_1, \Delta_2, \ldots, \Delta_{N_{train}}\}$

3: **for** each snapshot s_i in set of training snapshots **do**

4: Compute discrete cross-correlation between the reference snapshot s^{ref} and the snapshot s_i:

$$\text{correlation}(s^{\text{ref}}, s_i) = \text{correlate}(s^{\text{ref}}, s_i)$$

5: Obtain optimal shift Δ_i:

$$\Delta_i = \text{argmax}\left(\text{correlation}(s^{\text{ref}}, s_i)\right) - (N_h - 1)$$

6: To register snapshots: Apply the shift Δ_i to the snapshot s_i using circular shift:

$$s_i^{\text{ref}} = \text{shift}(s_i, \Delta_i)$$

7: Store the Registered snapshot s_i^{ref} and the shift Δ_i

8: **end for**

9: **return** the set of Registered snapshots $S^{\text{ref}} = \{s_1^{\text{ref}}, s_2^{\text{ref}}, \ldots, s_{N_{train}}^{\text{ref}}\}$ and shifts Δ_i.

7.3 ROM via cross-correlation based snapshot registration algorithm

In this section, we present the complete workflow for constructing ROMs using the cross-correlation based snapshot registration framework. As outlined in Algorithm 5, the workflow consists of two stages: (1) the offline stage, where snapshots are registered and the ROM is constructed, and (2) the online stage, the ROM predictions which are at the reference are reverted to the correct physical frame.

In the offline phase, the snapshots are registered using Algorithm 4, which

exploits cross-correlation to employ registration and yields a registered snapshots matrix S^{ref} and a corresponding shifts matrix Δ. Now, all the snapshots are in the reference frame, accelerating the KnW and yielding a low-dimensional linear approximation subspace. We construct ROM[2] utilizing the registered snapshots matrix S^{ref}, following the steps mentioned in previous chapter section 6.3, with radial basis function (RBF) interpolation used for regression. To obtain shifts for new time instances in the online phase, we construct a map α using a regression on the training pairs $\{t_i, \Delta_i\}$. In the online phase, the ROM predictions are initially at the reference frame, and we exploit the corresponding optimal shift obtained by the map α to revert the snapshots to the correct physical frame.

Algorithm 5: ROM via cross-correlation based snapshot registration

1: **Input:** Set of training snapshots $\{s_1, s_2, \ldots, s_{N_{train}}\}$ for corresponding time instances $\{t_1, t_2, \ldots, t_{N_{train}}\}$, reference snapshot s^{ref}, N_r number of POD modes.
2: **Output:** Prediction of snapshots via cross-correlation based snapshots registration.
Offline stage:
3: Registration of snapshots via cross-correlation using Algorithm 4: Obtain the Registered snapshots matrix S^{ref} and the corresponding optimal shifts matrix $\boldsymbol{\Delta}$
4: Construct ROM on the registered snapshots matrix S^{ref} following the steps in section 6.3.

$$\text{rom} = \text{ROM}(S^{\text{ref}}, \text{POD}, \text{RBF})$$

5: Construct map α using regression for $\{t_i, \Delta_i\}$ pairs: To predict shifts for new time instances in the online phase.
Online stage:
6: **for** each new time instance t_i in test set: **do**
7: s_i^{ref} = rom.predict(t_i)
8: Obtain the optimal shift Δ_i^* using map α .
9: Transport back the predicted snapshots to the correct physical frame by reverting them with the optimal shift:

$$s_i = \text{shift}(s_i^{\text{ref}}, -\Delta_i^*)$$

10: **end for**
11: **return** the set of predicted snapshots $\{s_1, s_2, \ldots, s_{N_{test}}\}$ in correct physical frame.

[2]using `EZyRB` package [50], a Python library for data-driven ROMs.

After working on this approach, we came across the work [84], which closely aligns with the framework we propose. In their approach, cross-correlation based snapshot registration, a widely used technique in medical imaging [16, 32], is integrated with the operator inference framework [144] for the construction of ROMs. This framework is referred to as shift operator inference [84], where shifts are obtained via cross-correlation. In this work, we integrate cross-correlation based snapshot registration within the non-intrusive data-driven ROM framework [188].

7.4 Numerical experiments

In this section, we evaluate the proposed approach to model reduction using cross-correlation based snapshot registration to construct ROMs for transport-dominated problems. We consider two numerical experiments: the 1D travelling waves and a higher-order benchmark test case - the 2D isentropic convective vortex.

7.4.1 1D Travelling wave

We consider a parameterized 1D Gaussian function (7.3) to mimic the solutions of the linear transport equation considered in [55, 60, 63], with $\beta = 1$, $x \in [0, 10.25]$ is the spatial coordinate, constant σ (standard deviation), and $t \in [0, 10.25]$ is the mean, interpreted here as time to treat this as a parameterized time-dependent problem. We generate a database consisting of 100 time instances and their corresponding snapshots over the 256 equidistant spatial nodes in $x \in [0, 10.25]$. We divide 50% of the database as a training set and the rest as a testing set, as shown in the Figure (7.1)

$$f(t) = \beta e^{-(x-t)^2/\left(2\sigma^2\right)}. \tag{7.3}$$

In the offline phase, we select the reference snapshot corresponding to the parameter $t = 3.03$. Using the procedure outlined in Algorithm 4, we perform snapshot registration, aligning multiple snapshots to the reference by leveraging cross-correlation based registration, as shown in Figure 7.2. From Figure 7.3, it is evident that the registration significantly accelerates the KnW decay, leading us to seek a low-dimensional linear approximation subspace for efficient ROM construction. Figure 7.3b presents the first POD mode obtained by employing registration, which captures the structure of the Gaussian travelling waves, alongside the first 10 POD modes of the unregistered snapshots. In the

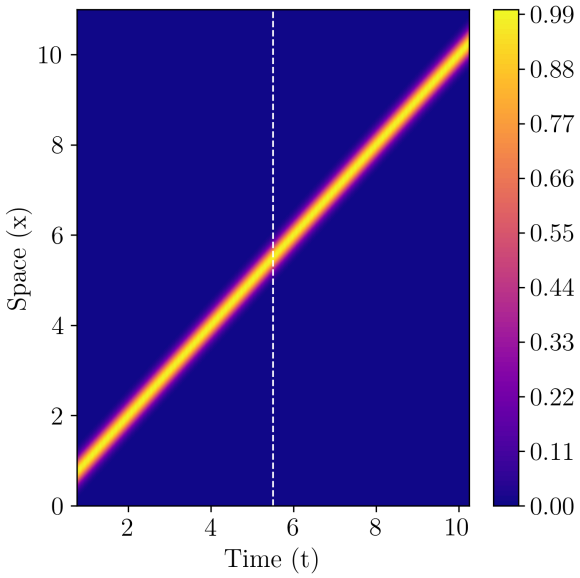
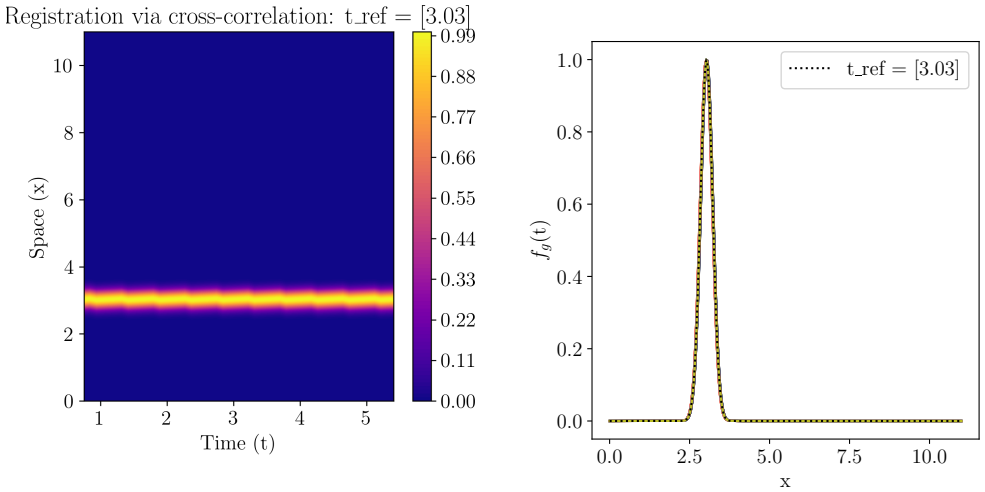


Figure 7.1: 2D contour of the original snapshots of the 1D travelling waves. Here, the white line separates the first 50% dataset as the training set and the rest as the test set.

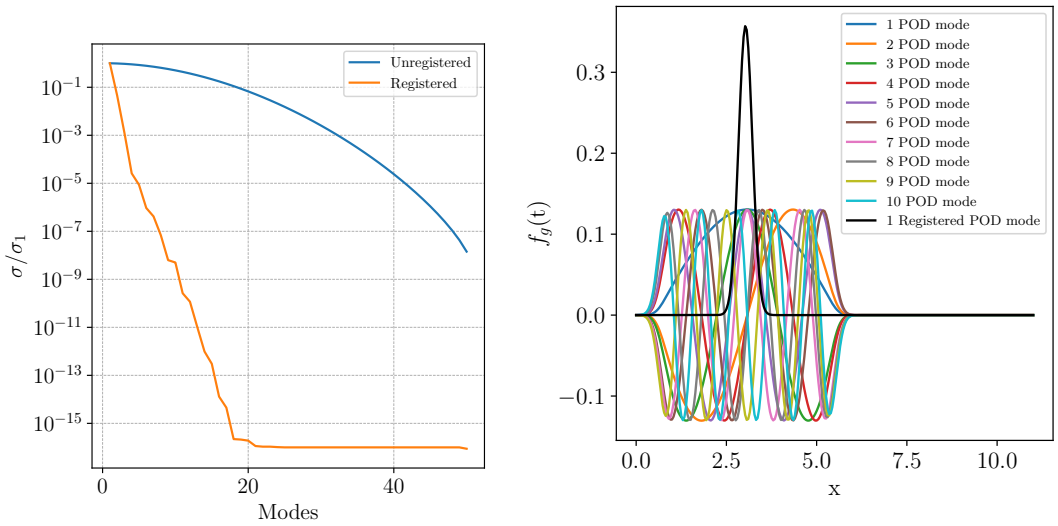
case of unregistered snapshots, we have to consider numerous POD modes to construct ROM, which leads to inefficient ROMs. We construct ROM by considering the first POD mode (with 99.99 % of cumulative energy) obtained via registration, and we exploit the POD-R strategy discussed in section 6.3. Here, we utilize the radial basis function (RBF) interpolation for the regression task.



(a) 2D contour of the registered train set.

(b) Snapshots aligned with reference.

Figure 7.2: 1D travelling waves training set registered at the reference configuration.



(a) Singular values decay comparison

(b) 1D travelling wave registered POD mode

Figure 7.3: **Left:** Comparison of the singular values of registered and unregistered 1D travelling waves. **Right:** The first 10 POD modes obtained from the unregistered snapshots, alongside the 1st POD mode obtained via cross-correlation based snapshot registration.

After constructing the ROM, we incorporate shifts in the online phase by mapping parameter t_i to the optimal shift Δ_i (denoted as α in Algorithm 5) using linear interpolation. To predict snapshots at new time instances, we employ the regression α to transform the ROM-predicted snapshots (in the reference frame) back to the correct physical frame. Figure 7.4 presents the predicted snapshots in the correct physical frame for both train and test set parameters, alongside the original snapshots (ground truth) and the absolute differences between predictions and ground truth. The relative prediction error, shown in Figure 7.5, reveals that test errors are an order of magnitude larger than training errors, with the mean relative error highlighted.

In Figure 7.6, we present the mean relative L_2 prediction error for test set parameters, with varying POD ranks, for the construction of POD-RBF-ROM, both unregistered and registered snapshots via cross-correlation. It is evident that in the unregistered case, even after considering 25 modes (with 99.99 % cumulative energy), it leads to inefficient and inaccurate ROMs even for simple linear transport problems, with errors two orders of magnitude larger than in the registered case. In contrast, for the registered case, the first mode (retaining 99.99% of the cumulative energy) results in an efficient and accurate ROM.

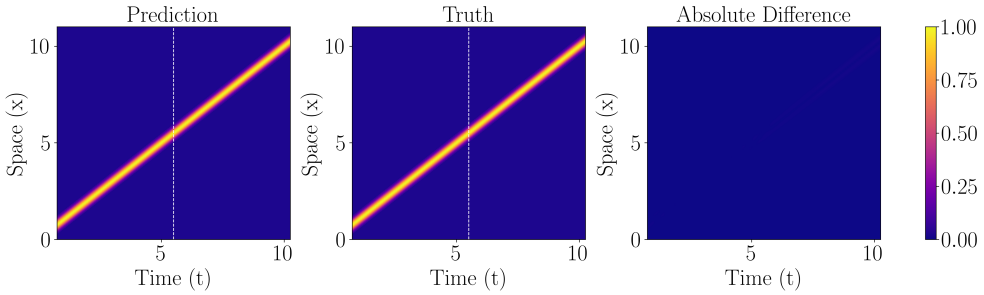


Figure 7.4: Comparison of the predictions of the resulting ROM of the 1D travelling waves with the original snapshots as ground truth, here the white line separates the first 50% dataset as the training set and the remaining as the test set, and also shows the absolute difference between the prediction and the ground truth.

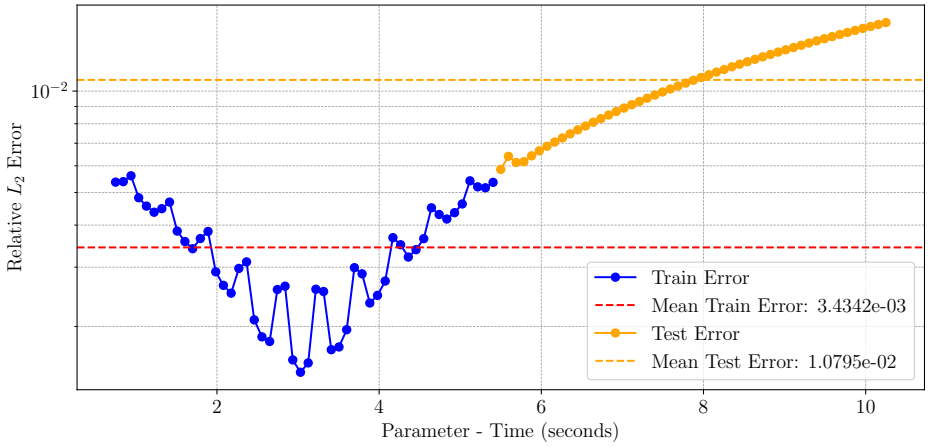


Figure 7.5: Relative L_2 predictions error by the ROM of the 1D travelling wave via cross-correlation based snapshots registration, for both training set and test set parameters.

7.4.2 2D isentropic convective vortex

In this numerical experiment, we consider the dataset obtained in section 6.6.2. We consider the first 30% dataset as a training set and the remaining 70% as the test set. We perform registration on the training set. Figure 7.7 presents the registered snapshots at time instances $t \in \{0.625, 25, 62.5\}$, comparing with the reference and the corresponding original snapshot. We perform reduction using POD on both the registered snapshots matrix and unregistered snapshots matrix, the singular values decay is shown in Figure 7.8a with sharp decay for registered snapshots. Furthermore, Figure 7.8b illustrates the POD modes obtained from the registered snapshots, which accurately capture the vortex structure (first row of Figure 7.8b). In contrast, the POD modes from the unregistered snapshots fail to capture the vortex structure effectively (second row

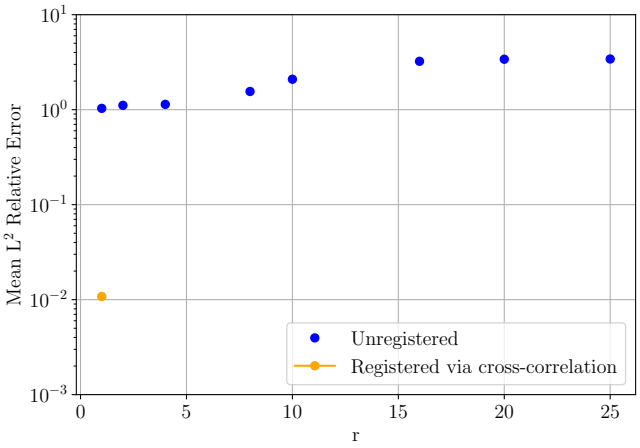


Figure 7.6: Comparison of mean relative L_2 prediction error for 1D travelling wave test set parameters using POD-RBF-ROM, presented for the unregistered case (without registration) and the registered case (with registration via cross-correlation).

of Figure 7.8b).

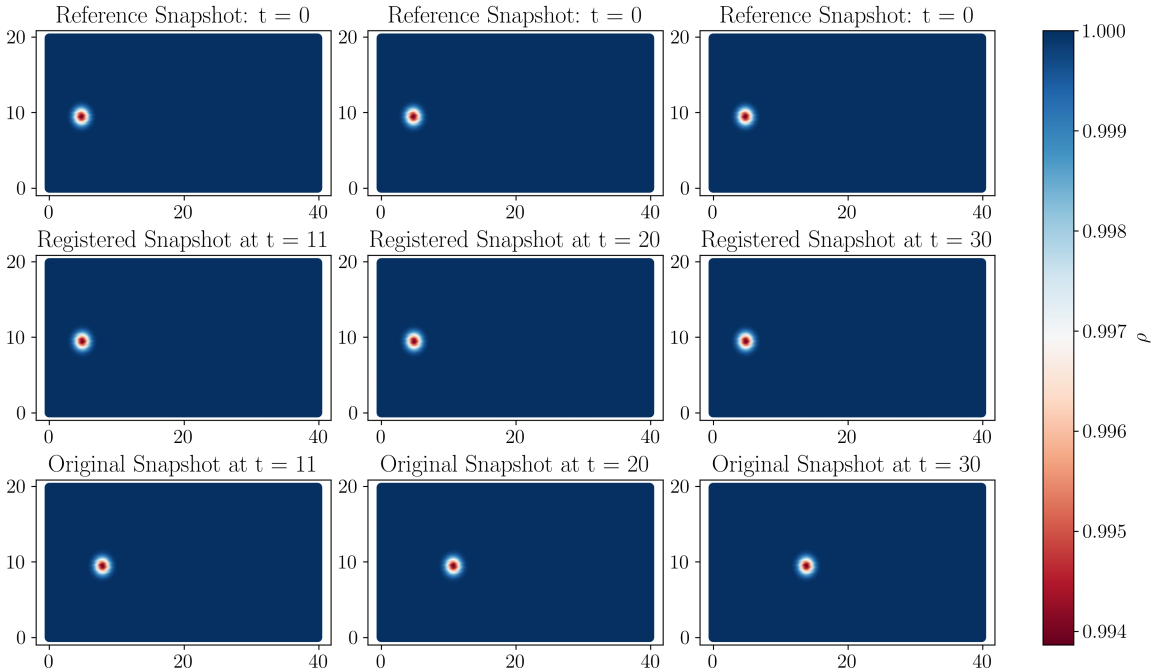
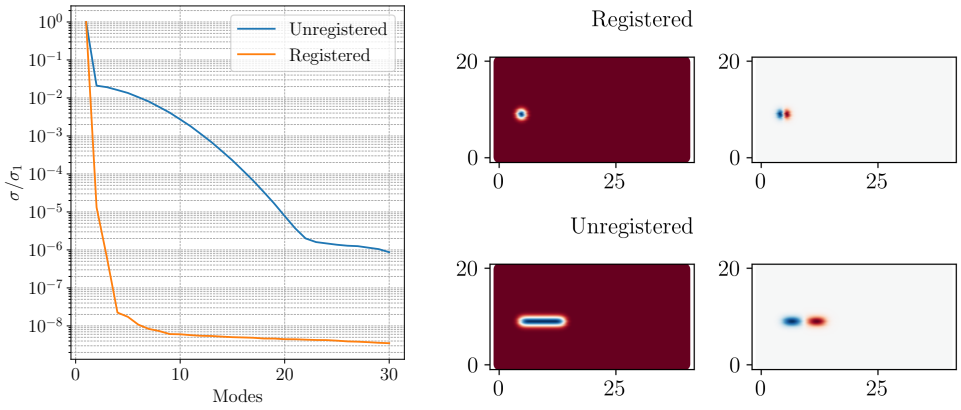


Figure 7.7: Comparison of the isentropic convective vortex registered snapshots with the reference and the corresponding original snapshot.

We utilize the first POD mode obtained via registration and as previously, we employ the POD-R strategy to construct ROM for this numerical experiment. To employ shifts in the online phase, we again use linear interpolations to build parameter t_i to optimal shift Δ_i mappings. As specified in the previous case in the online phase, we utilize the predicted optimal shifts alongside the ROM



(a) Singular values decay (b) Comparison of POD modes

Figure 7.8: **Left**: Comparison of singular values decay of registered snapshots matrix, labelled as Registered and of the unregistered snapshots matrix as Unregistered. **Right**: Showing the first two modes of the registered snapshots (first row - Registered) and showing the first two modes of the unregistered snapshots matrix (second row - Unregistered).

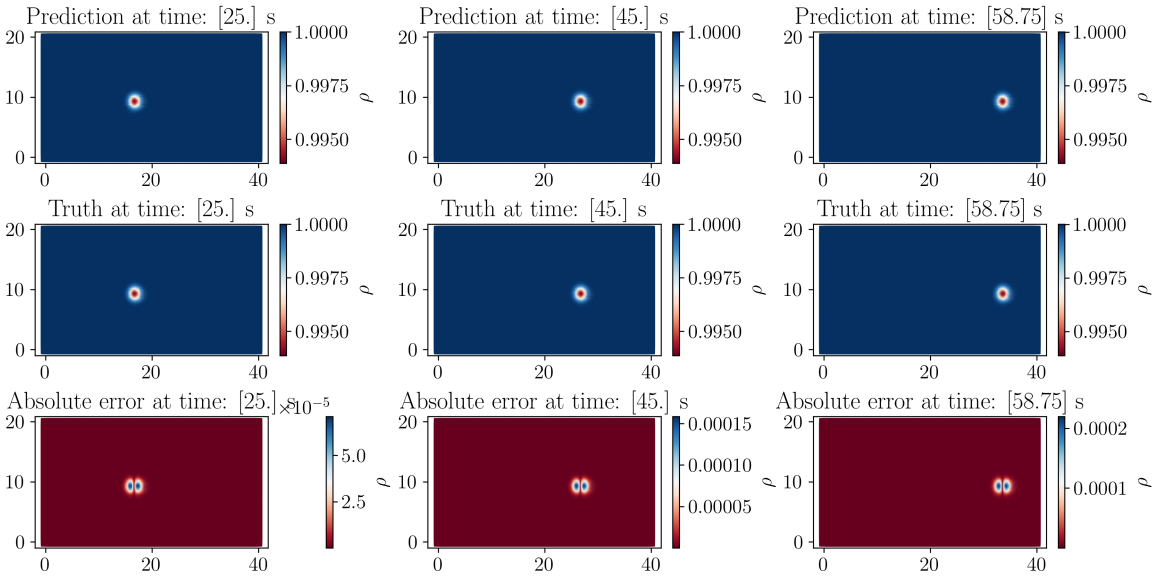


Figure 7.9: Predicted snapshots using POD-RBF-ROM with cross-correlation based registration for the isentropic convective vortex test set parameters at time instances $t \in \{25, 45, 58.75\}$s.

prediction to revert the predicted snapshots to the correct physical frame. The predicted snapshots in the correct physical frame for three-time instances $t \in \{25, 45, 58.75\}$ from the test set are shown in Figure 7.9, with the corresponding original snapshot as ground truth and the absolute difference. The prediction error for both the training and test set parameters is presented in 7.10. The relative L_2 error of the predicted snapshots is plotted for each parameter, along

with the mean relative L_2 error.

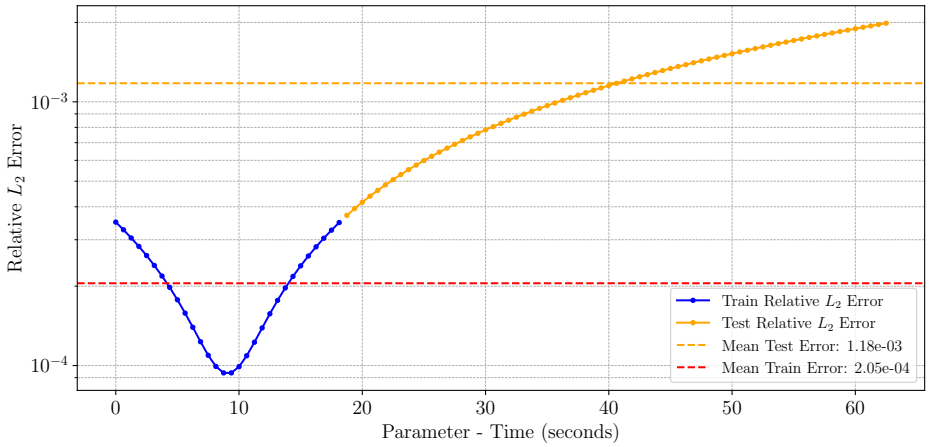


Figure 7.10: Relative L_2 predictions error of the isentropic convective vortex ROM via cross-correlation based registrations, for both training and test set parameters.

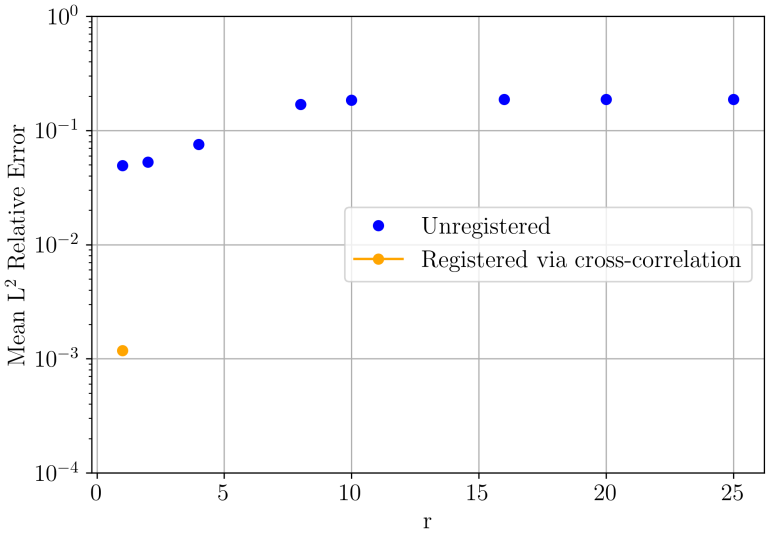


Figure 7.11: Comparison of mean relative L_2 prediction error for 2D isentropic convective vortex test set parameters using POD-RBF-ROM, presented for the unregistered case (without registration) and the registered case (with registration via cross-correlation).

In Figure 7.11, we illustrate the mean relative L_2 prediction error for test set parameters across different POD ranks in the construction of POD-RBF-ROM, using both unregistered and registered snapshots via cross-correlation. The results show that in the unregistered case, even with 25 modes, the ROM remains inefficient and inaccurate, yielding errors that are two orders of magnitude higher than in the registered case. Conversely, in the registered case,

the first POD mode retaining 99.99 % of the cumulative energy is sufficient to yield efficient and accurate ROM.

7.5 Conclusion

In this work, we employ cross-correlation based snapshot registration for model reduction of transport-dominated problems. This cross-correlation based registration accelerates the KnW decay, facilitating the construction of a low dimensional linear approximation subspace for efficient ROM development. We propose a complete framework comprising offline-online stages for the construction of ROMs using the cross-correlation based snapshots registration. We tested our proposed workflow on transport-dominated problems: 1D travelling waves, and 2D isentropic vortex. In both numerical experiments, the proposed approach results in efficient and accurate ROMs. However, the applicability of this approach is confined to certain transport-dominated problems where the structures of the transported features remain consistent, meaning not diffusive. This approach is not suitable for addressing advection-diffusion problems.

Snapshot registration has seen significant advancements in medical imaging, largely driven by the progress of deep learning frameworks [218]. Methods such as VoxelMorph [7], LapIRN [125], and FAIM (FAST Image Registration) [101] have demonstrated both speed and accuracy. These methods can be integrated within model reduction frameworks to address the challenges posed by slow KnW decay.

Chapter 8

Model order reduction strategies for tundish RTD analysis

8.1 Introduction

The chapter 3 of Part I presents the Full Order Model (FOM) framework used to simulate the steady-state operation of an industrial single-strand tundish, which comprises one steady-state simulation and later the tracer transient simulation, initialized with the converged steady-state fields. The resulting high-fidelity simulations capture key flow characteristics and species transport behavior, yielding Residence Time Distribution (RTD) under regular casting conditions. The FOMs yields detailed and accurate predictions; its high computational cost and complex simulation pipeline pose significant limitations for tasks requiring repeated evaluations, such as design optimization, parametric studies, or real-time process monitoring.

To address these challenges, this chapter investigates Model Order Reduction (MOR) strategies tailored to RTD analysis in continuous casting tundish systems. Here, the objective is to develop ROMs that substantially reduces computational effort while preserving the essential flow and transport dynamics relevant to RTD. Both intrusive, projection-based MOR and non-intrusive, data-driven approaches [188] are explored to obtain the RTD curve efficiently, enabling faster and more scalable tundish simulations.

Projection-based model order reduction methods offer a systematic framework for reducing the computational cost associated with solving high-dimensional discretized PDEs. These methods aim to construct ROMs that capture the essential dynamics of the FOMs while operating in a low-dimensional subspace.

Among the most prominent MOR strategies, POD has gained significant traction for its capability to identify and extract the most energetic modes from full-order high-fidelity simulations.

In the POD-Galerkin method, the governing equations are projected onto the reduced subspace (spanned by POD basis), which is a widely adopted methodology. Hall et al. [71] employed the snapshot POD method to model unsteady aerodynamic flows in both transonic and subsonic regimes, examining configurations such as an isolated airfoil and a cascade of flat plate airfoils. Their work demonstrated the ability of POD-based ROMs to deliver accurate predictions, highlighting their potential for use in active flow control. In particular, POD-Galerkin ROMs has found widespread application across various domains, including optimal control, design optimization, data assimilation, and real-time query systems. Ravindran [155] developed a POD-Galerkin ROM for optimal fluid flow control in channel problems, highlighting both the model's accuracy in short-time prediction and its ability to deliver significant computational speed-ups. Such characteristics are essential for time-sensitive applications like feedback control. Similarly, Bourguet and Braza [19] investigated compressible, unsteady transonic flows over a NACA0012 airfoil using a POD-Galerkin ROM. Their model captured key flow instabilities—such as von Kármán vortex shedding and buffeting—with excellent agreement to full-order simulations, although the evaluation of non-linear terms remained computationally demanding.

Further advancements in ROM development include notable efforts using finite volume FOMs of the NSE, as demonstrated in the foundational studies [114, 181, 182, 69]. Ballarin et al. [8] proposed a monolithic POD-Galerkin approach for fluid-structure interaction problems. Also in [10] presented a stable POD-Galerkin ROM tailored for the unsteady, incompressible, and parameter-dependent NSE. Another contribution by Ballarin et al. [9], the use of POD-Galerkin ROMs in the context of haemodynamics, further highlights the versatility of model reduction techniques for parameterized PDEs.

Several efforts have been made to extend POD-Galerkin ROMs to non-isothermal problems. An early attempt is reported in [1], where the authors developed a POD-Galerkin ROM for modelling the temperature field within a rapid thermal processing chamber, focusing on a two-dimensional steady-state case. In [4], Alonso et al. combined POD with a genetic algorithm to construct a ROM

for heat transfer in a backward-facing step flow. A one-dimensional conduction problem was addressed in [152] using a POD-Galerkin approach, and further studies on the heat conduction equation were carried out in [202]and [72]. Natural circulation was investigated in [109] by developing an ROM from a FOM based on the coupled Navier-Stokes and energy equations. However, the resulting reduced model only allowed variations in the two-dimensional temperature field, keeping the flow field fixed, thus limiting its applicability to small perturbation scenarios in temperature control. In [138], a POD-Galerkin ROM was proposed for groundwater flow problems subject to spatially distributed stochastic inputs. The authors generated snapshots in the probability space, leading to a reduced order Monte Carlo framework (ROMC). An alternative ROM approach is explored in [127] for uncertainty propagation in porous media, where the Karhunen-Loève decomposition (equivalent to POD) and polynomial chaos expansion with Smolyak sparse grids were applied. The POD method was also employed in [108] for two-dimensional solute transport problems. [25] presented a POD-Galerkin ROM for weakly coupled Navier-Stokes and heat transport equations, using a hybrid finite element–finite volume discretization. Finally, Georgaka et al. [178] developed a POD-Galerkin reduced-order model for the three-dimensional unsteady Navier–Stokes equations minimally coupled with the heat transport equation, parametrizing both the boundary condition and physical parameter variations.

In metallurgical applications such as RTD analysis in tundish systems, Projection based Model Order Reduction (PMOR) enables efficient simulation of unsteady transport phenomena. RTD analysis is essential for assessing flow behavior and mixing characteristics within the tundish, both of which directly affect steel cleanliness and product quality. High-fidelity simulations of RTD based on the Navier–Stokes and scalar transport equations offer accurate representations of these transport phenomena. However, these models are computationally expensive, especially for parametric studies or real-time applications. By projecting the governing equations onto a reduced basis subspace, ROMs yields fast and accurate approximations of scalar concentration fields. This approach facilitates detailed RTD characterization across varying flow configurations and operational conditions, making it particularly valuable for design optimization, process monitoring, and digital twin development in continuous casting. In this study, we adopt the framework proposed by Khamlich et al. [94] to construct a projection-based ROM tailored to RTD analysis in a tundish configuration. Further, to obtain convective fields from steady state

simulations, PMOR can be applied for isothermal and non-isothermal conditions by following the procedure of [181, 182] and [178], where the treatment of non-linear terms poses additional computational challenges, often necessitating specialized stabilization techniques or hyper-reduction strategies. Here, we'll explore both projection based ROM and data-driven ROM approaches.

In this study, we adopt the framework proposed by Khamlich et al. [94] to construct a projection-based reduced order model (ROM) tailored for residence time distribution (RTD) analysis of an industrial-scale continuous casting tundish. To compute the convective flow fields from steady-state simulations, projection-based model order reduction can be applied under both isothermal and non-isothermal conditions, following the procedures outlined in [181, 182, 178]. In the non-isothermal case, the treatment of non-linear terms poses additional computational challenges, often requiring stabilization techniques or hyper-reduction strategies.

In this work, we propose an intrusive projection-based ROM framework for Residence Time Distribution (RTD) analysis in an industrial-scale continuous casting tundish, building on the methodology presented in [94]. Specifically to enable rapid online evaluation of the reduced operators, we incorporate an operator approximation strategy based on regression models. In industrial scenarios, where commercial CFD codes are commonly used and access to source code is restricted, hindering the development of intrusive ROMs, we address this limitation by developing a data-driven ROM.

The main contributions of this chapter are as follows:

- We developed both intrusive projection-based and non-intrusive data-driven reduced order model for residence time distribution analysis in an industrial-scale, three-dimensional continuous casting tundish.
- We validate the proposed ROM against experimental data and high-fidelity full order model (FOM) simulation results, demonstrating its accuracy.
- We show that the ROM provides reliable predictions for unseen parameters, with good agreement with FOM results. Additionally, we assess the effectiveness of the operator approximation strategy employed in projection based ROM for rapid online evaluation of reduced operators in an industrial setting.
- The proposed physics-based and data-driven ROM framework is efficient and scalable, enabling real-time process monitoring, optimization,

and design-space exploration for continuous casting operations, thus offering a practical tool to enhance steelmaking processes.

8.2 Projection based model order reduction

8.2.1 Offline stage

The FOM described in section 3.3 is typically solved for each parameter value $\mu_k \in \mathcal{K} = \{\mu_1, \ldots, \mu_{N_\mu}\} \subset \mathcal{P}$, where $\mathcal{K}$ is a finite-dimensional training set derived from the parameter space $\mathcal{P}$. Since the problem is both parameter and time dependent, the collection of snapshots for constructing reduced basis spaces must account for both variations. Consequently, discrete time instances $t_k \in \{t_1, \ldots, t_{N_t}\} \subset [0, T]$ are also treated as part of the training set, forming a finite set of time samples within the simulation time interval. The total number of snapshots is thus $N_s = N_\mu \cdot N_t$. The snapshot matrix $\mathcal{S}$ is assembled using these N_s full-order solutions of the concentration field, evaluated at N_h discrete points in the computational domain:

$$\mathcal{S} = [c(\mu^1, t^1), \ldots, c(\mu^{N_\mu}, t^{N_t}] \in \mathbb{R}^{N_h \times N_s}. \tag{8.1}$$

To construct a reduced basis subspace for the projection of the governing equations, there are several established techniques [150, 76, 160], such as Proper Orthogonal Decomposition (POD), Proper Generalized Decomposition (PGD), and the Reduced Basis (RB) method based on greedy sampling strategy. In the present work, POD is employed to obtain a reduced basis subspace. Particularly, the nested POD method, where POD is first applied to the temporal domain and subsequently in the parametric space. Let $c(\mu, t)$ be a scalar or vector-valued function, with a set of realizations $\{c_1, c_2, \ldots, c_{N_s}\}$. The POD problem seeks, for each dimension of POD space $N_{\text{POD}} = 1, \ldots, N_s$, a set of basis functions $\{\varphi_1, \ldots, \varphi_{N_{\text{POD}}}\}$ and coefficients $\{a_i^j\}$ for $i = 1, \ldots, N_s$, $j = 1, \ldots, N_{\text{POD}}$, that minimize the functional:

$$\min \sum_{i=1}^{N_s} \left\| c_i - \sum_{j=1}^{N_{\text{POD}}} a_i^j \varphi_j \right\|_{L_2(\Omega)}^2 \quad \forall\, N_{\text{POD}} = 1, \ldots, N_s. \tag{8.2}$$

with the orthonormality constraint:

$$\langle \varphi_i, \varphi_j \rangle_{L_2(\Omega)} = \delta_{ij}, \quad \forall i, j = 1, \ldots, N_s. \tag{8.3}$$

As demonstrated in [150], the minimization problem presented in (8.2) is mathematically equivalent to solving the associated eigenvalue problem:

$$\boldsymbol{\mathcal{C}}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{\lambda}, \tag{8.4}$$

$$\boldsymbol{\mathcal{C}}_{ij} = \left\langle c_i, c_j \right\rangle_{L_2(\Omega)} \quad \forall\, i,j = 1,\ldots,N_s, \tag{8.5}$$

where $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N_s \times N_s}$ is the correlation matrix constructed from the snapshots matrix $\mathcal{S}$, $\boldsymbol{Q}$ is the square matrix containing the eigenvectors, and $\boldsymbol{\lambda}$ is the diagonal matrix consisting of corresponding eigenvalues. The basis functions are subsequently derived as:

$$\varphi_i = \frac{1}{N_s \lambda_i} \sum_{j=1}^{N_s} c_j \boldsymbol{Q}_{ij}. \tag{8.6}$$

The POD basis functions obtained using the aforementioned methodology yield the following reduced subspace:

$$L = \{\varphi_1, \ldots, \varphi_{N_{rb}}\} \in \mathbb{R}^{N_h \times N_{rb}}, \tag{8.7}$$

where the cardinality $N_{rb} < N_s$ is chosen based on the eigenvalue decay of the associated problem.

Once the reduced subspace is constructed, the reduced tracer concentration fields $c(\mu, t)$ are approximated with:

$$c(\mu, t) \approx \sum_{i=1}^{N_{rb}} a_i(\mu, t)\varphi(x), \tag{8.8}$$

where $a_i(\mu, t)$ are the parameter-time dependent coefficients, and $\varphi_i(x)$ are the parameter independent basis functions.

The unknown modal coefficients are obtained via Galerkin projection [114, 181] of the governing species transport equation (3.11) [94] onto a reduced subspace spanned by the POD basis functions. This results in a parameter-dependent reduced-order system of ordinary differential equations (ODEs):

$$\boldsymbol{M_r}(\mu)\dot{a}(t;\mu) + \boldsymbol{C_r}(\mu)a(t;\mu) = \boldsymbol{B_r}(\mu)a(t;\mu), \tag{8.9}$$

where $a(t;\mu) \in \mathbb{R}^{N_{\text{rb}}}$ denotes the vector of reduced modal coefficients for a given parameter $\mu \in \mathcal{P}$, and each operator in Eq. (8.9) is parameter-dependent.

These reduced operators are obtained through Galerkin projection of the full-order operators onto the reduced POD subspace:

$$\begin{cases} (\boldsymbol{M_r}(\mu))_{ij} = \langle \varphi_i, \varphi_j \rangle_{L^2(\Omega)}, \\ (\boldsymbol{B_r}(\mu))_{ij} = \langle \varphi_i D_{\text{eff}}(\mu), \Delta \varphi_j \rangle_{L^2(\Omega)}, \\ (\boldsymbol{C_r}(\mu))_{ij} = \langle \varphi_i, \nabla \cdot (u(\mu)\varphi_j) \rangle_{L^2(\Omega)}, \end{cases} \tag{8.10}$$

where $\{\varphi_i\}_{i=1}^{N_{\text{rb}}}$ denote the orthonormal POD basis functions. Note that the effective diffusivity $D_{\text{eff}}(\mu)$ and the velocity field $u(\mu)$ depend on the parameter μ, and thus induce parametric dependence in the reduced operators.

The POD basis is computed using the method of snapshots, originally introduced by Sirovich [177], and implemented using the ITHACA-FV framework[1] [180]. ITHACA-FV is an OpenFOAM-based platform providing reduced-order modeling capabilities for parametrized problems. The same framework is used to compute the reduced operators defined in Eq. (8.10). Time integration of the reduced system is performed using the implicit backward Euler method. Finally, the reduced-order solution is reconstructed via the linear modal expansion described in Eq. (8.8).

8.2.2 Online stage

In the online stage of the parametric reduced order modeling framework, the goal is to rapidly evaluate the reduced system for a new, unseen parameter value $\mu^\star \in \mathcal{P}$, without solving the full-order problem. To this end, the reduced operators $\boldsymbol{B_r}(\mu^\star)$ and $\boldsymbol{C_r}(\mu^\star)$ must be estimated, as they are parameter-dependent and govern the system dynamics through the reduced order ODE:

$$\boldsymbol{M_r}(\mu^\star)\dot{a}(t;\mu^\star) + \boldsymbol{C_r}(\mu^\star)a(t;\mu^\star) = \boldsymbol{B_r}(\mu^\star)a(t;\mu^\star). \tag{8.11}$$

In practice, the reduced mass matrix $\boldsymbol{M_r}(\mu)$ is often independent of the parameter μ. Therefore, we assume $\boldsymbol{M_r}(\mu) \approx \boldsymbol{M_r}$ and pre-compute $\boldsymbol{M_r}$ during the offline stage.

To enable operator prediction at new $\mu^\star$, we construct regression models to approximate the mappings:

$$\mu \in \mathcal{P} \mapsto \boldsymbol{B_r}(\mu), \quad \mu \in \mathcal{P} \mapsto \boldsymbol{C_r}(\mu), \tag{8.12}$$

[1] The ITHACA-FV source code is available at `https://github.com/ITHACA-FV/ITHACA-FV`.

we use regression models trained on the operator values computed at a finite set of training parameters $\{\mu_i\}_{i=1}^{N_{\text{train}}}$ during the offline stage. Several interpolation methods can be employed for operator approximation [160], including inverse distance weighting, Gaussian process regression, or neural networks [94].

In this work, we use Radial Basis Function (RBF) interpolation to approximate the operators at $\mu^\star$. The interpolated operators $\boldsymbol{B_r}(\mu^\star)$ and $\boldsymbol{C_r}(\mu^\star)$ are then used to assemble and solve Eq. (8.11) using the backward Euler method. The solution $a(t;\mu^\star)$ is finally used to reconstruct the full state via the linear modal expansion:

$$c^{\text{ROM}}(x,t;\mu^\star) = \sum_{i=1}^{N_{\text{rb}}} a_i(t;\mu^\star)\varphi_i(x). \tag{8.13}$$

Although standard RBF interpolation does not guarantee the preservation of structural properties such as symmetry or anti-symmetry of the reduced operators, in our numerical experiments, we observed that the interpolated operators still yield reduced order solutions that closely match the FOM results across the tested parameter space. This suggests that, in our case, the effect of structure violation is negligible regarding overall accuracy and stability. Nevertheless, this remains a known theoretical limitation, and future extensions could incorporate structure-preserving interpolation strategies to ensure consistency with operator properties. This is further supported by the numerical experiments in section 8.3, where the RBF-interpolated operators' ROM predictions reproduce key dynamics of the full-order system.

8.3 Results and discussion

For the RTD parametric study, we consider 11 different inlet velocities, we consider eleven distinct inlet velocity values to systematically investigate the influence of flow rate on transport and mixing behavior within the tundish. These inlet velocities span a representative range of operational conditions, allowing us to assess the sensitivity of the scalar transport field and the resulting RTD curves to variations in the inflow boundary condition. This parametric approach is essential for understanding process dynamics and optimizing tundish design and operation.

In this study, the projection-based ROM is constructed following the procedure described in section 8.2. To construct the parametric projection based ROM for

RTD analysis, we consider parameter values corresponding to distinct simulation cases with varying inlet velocity conditions. Specifically, we select eight parameter instances — indexed as μ_i, where, $i = \{1, 2, 3, 4, 7, 8, 9, 10\}$, form the training set (from Table 3.3). POD is performed on the training set, the first 21 modes capturing 99.99% of the cumulative energy, are retained to construct the reduced basis subspace. Subsequently, the governing equations for tracer transport are projected onto the reduced basis subspace using Galerkin projection. Following the projection, a reduced order system of ODEs is obtained, governing the temporal evolution of the modal coefficients. These ODEs are integrated in time using an implicit backward Euler method. The reduced tracer concentration fields are then reconstructed by a linear combination of the time-dependent coefficients and the spatial POD modes. To obtain the parameter-dependent reduced operator in the online phase, we build the parameter-to-reduced operator mapping using RBF interpolation and employ it in the online phase for fast evaluation of reduced operators $\boldsymbol{B_r}(\mu)$ and $\boldsymbol{C_r}(\mu)$. The remaining parameter instances, μ_i, where, $i = \{5, 6, 11\}$, are for testing. These test parameters are not used during the training phase and serve to evaluate the predictive capabilities and generalization of the ROM, especially for interpolation and mild extrapolation regimes.

Figure 8.1 compares the RTD curves for parameter μ_1 obtained from the experiment, the full-order model (FOM) CFD simulation, and the projection based ROM. Both the FOM and ROM predictions exhibit good agreement with the experimental data, with only minor discrepancies observed.

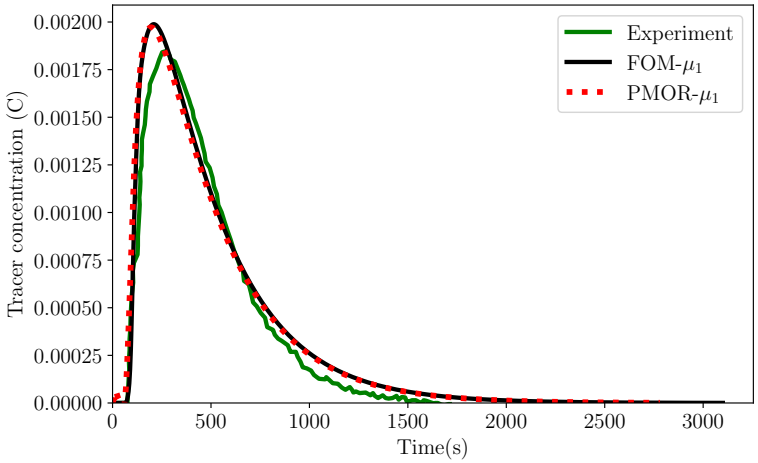


Figure 8.1: Comparison of RTD curves from the experiment, FOM CFD simulation, and projection-based ROM, illustrating the good agreement between the simulations and experimental data.

In the interpolatory regime, μ_i, where, $i = \{5, 6\}$, the parametric projection based ROM shows excellent agreement with the full-order simulation, accurately predicting the RTD curves for test parameters within the range used for training. The model captures the dynamics of the system effectively, providing reliable results for these parameters.

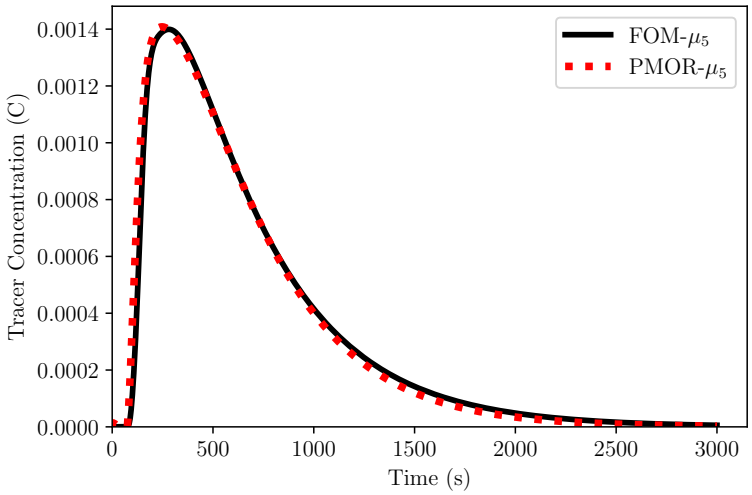


Figure 8.2: RTD curves for test parameter μ_5: comparison between the parameter-time dependent projection based ROM and full-order simulation results, showing excellent agreement within the interpolatory regime.

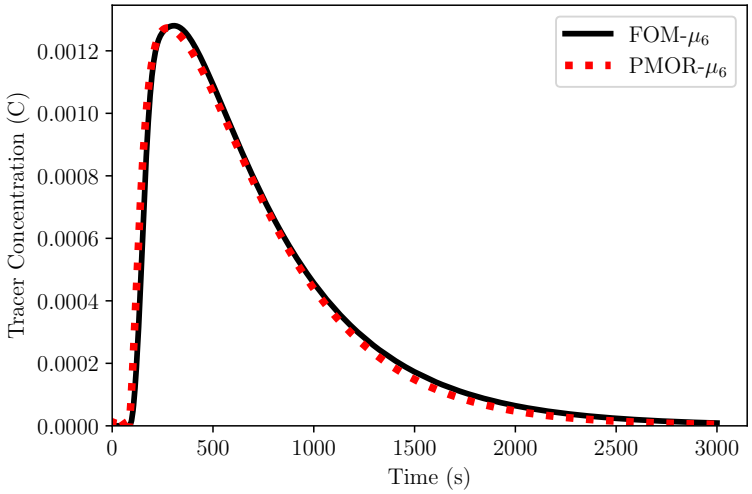


Figure 8.3: RTD curves for test parameter μ_6: comparison between the parameter-time dependent projection based ROM and full-order simulation results, showing excellent agreement within the interpolatory regime.

In the extrapolatory regime, where the test parameter μ_{11} is outside the training range, the ROM still captures the overall trend of the RTD curve but shows a slight discrepancy in maximum concentration prediction. This demonstrates

the ROM's ability to extrapolate beyond the trained parameters, though with some limitations in precision.

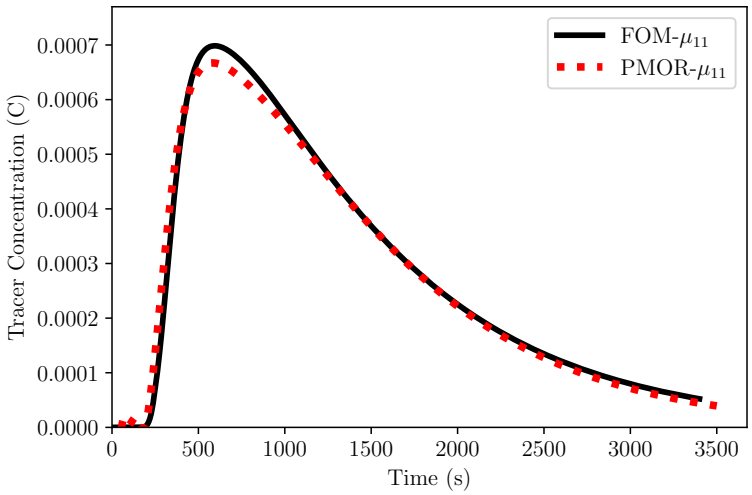


Figure 8.4: RTD curves for test parameter μ_{11} in the extrapolatory regime: comparison between the parameter-time dependent projection based ROM and full-order simulation results, ROM predictions accurately capturing the overall trend of the RTD curve, with a slight discrepancy in the maximum concentration prediction.

In industry, the most common practice is to use commercial CFD solvers, which limits access to the source code and makes the development of intrusive ROMs impractical. With this constraint in mind, we construct a data-driven ROM to enable efficient parametric analysis of RTD.

To construct the parametric data-driven ROM (see 6.3) for RTD analysis, we consider parameter values that correspond to distinct simulation cases with varying inlet velocity conditions. Again, we select eight parameter instances — indexed as μ_i, where, $i = \{1, 2, 3, 4, 7, 8, 9, 10\}$, form the training set (from Table 3.3). The full-order simulation snapshots of these are assembled to build the reduced basis subspace via POD, and to build the mapping from the parameter-time space to the reduced coefficients, RBF interpolation is employed. The remaining parameter instances, μ_i, where, $i = \{5, 6, 11\}$, are for testing. These test parameters are not used during the training phase and serve to evaluate the predictive capabilities and generalization of the ROM, especially for interpolation and mild extrapolation regimes.

In the interpolatory regime, μ_i, where, $i = \{5, 6\}$, the data-driven POD-RBF-ROM shows excellent agreement with the full-order simulation, accurately

predicting the RTD curves for test parameters within the range used for training. The model captures the dynamics of the system effectively, providing reliable results for these parameters.

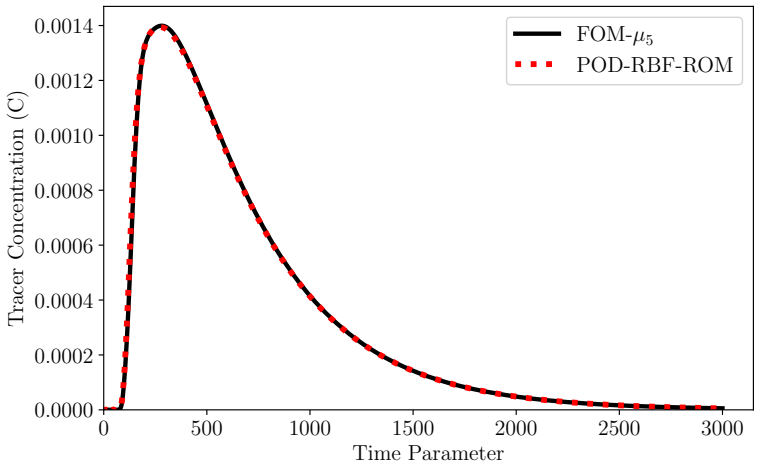


Figure 8.5: RTD curves for test parameter μ_5: comparison between the data-driven POD-RBF-ROM and full-order simulation, showing excellent agreement within the interpolatory regime

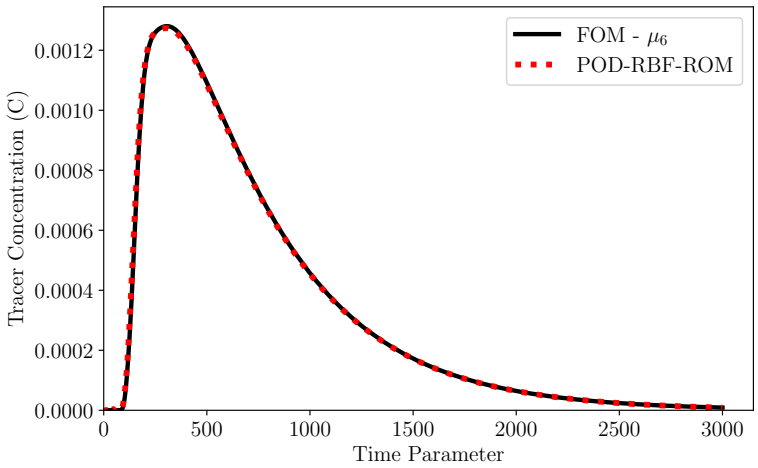


Figure 8.6: RTD curves for test parameter μ_6: comparison between the data-driven POD-RBF-ROM and full-order simulation, showing excellent agreement within the interpolatory regime

In the extrapolatory regime, where the test parameter μ_{11} is outside the training range, the ROM still captures the overall trend of the RTD curve but shows fluctuations in the initial phase and a slight discrepancy in concentration prediction. This demonstrates the ROM's ability to extrapolate beyond the trained parameters, though with some limitations in precision.

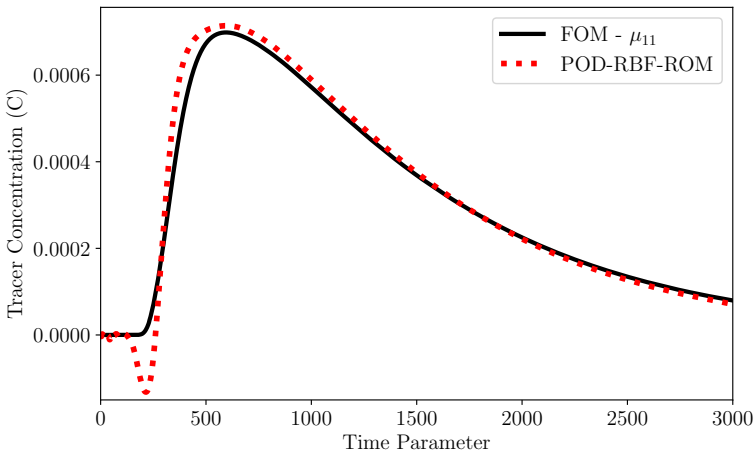


Figure 8.7: RTD curves for test parameter μ_{11} in the extrapolatory regime: comparison between the data-driven POD-RBF-ROM and full-order simulation, ROM predictions showing fluctuations in the initial phase but accurately capturing the overall trend of the RTD curve, with a slight discrepancy in the concentration prediction.

8.4 Conclusion

In this chapter, a projection-based ROM was developed for RTD analysis of the continuous casting tundish. The resulting ROM was validated against both experimental data and FOM simulations. The comparison of RTD curves shows good agreement among the experiment, full-order simulation, and the projection-based ROM results, with only minor discrepancies. These results validate the accuracy of the projection-based ROM and highlight its potential as an efficient surrogate for full-scale simulations in analysing flow behaviour within the tundish, provided that access to the source code of the CFD solver is available. This is an important consideration, as industrial practice typically relies on commercial CFD software, where such access is often restricted.

In such cases, data-driven ROMs offer a viable alternative, enabling efficient parametric analysis using only full-order simulation data, without requiring modifications to the underlying solver. For the parametric RTD analysis, the data-driven POD-RBF-ROM is employed. The model demonstrates excellent accuracy in the interpolatory regime, with predictions closely matching full-order simulations. However, in the extrapolatory regime, slight discrepancies are observed, particularly in the initial phase, where the model captures the overall trend of the RTD curve but experiences fluctuations and minor concentration prediction errors. This indicates that the data-driven ROM is reliable in the interpolatory regime; further refinement is required in the extrapolatory regime, potentially through increasing the training data or via advanced

interpolation techniques.

Overall, the integration of ROM techniques significantly reduces computational costs while retaining essential flow characteristics, demonstrating their potential for real-time analysis, design optimization, and digital twin application in metallurgical processes such as continuous casting.

Chapter 9

ROM for grade intermixing in a continuous casting tundish

9.1 Introduction

Ladle changeover transient operation results in the intermixing of different steel grades within the tundish. Accurately predicting the evolution of this mixing process is critical for minimizing yield loss and maintaining product quality. The chapter 4 presents the FOM framework considered to simulate the ladle changeover operations. As reported in Table 4.4, the operation can be broadly categorized into three stages, namely steady-state, emptying, and filling, and each requires tailored modelling FOM strategies. While FOMs provides comprehensive insights into the complex non-linear multiphase flow physics involved within the tundish, their high computational cost and long simulation times make them impractical/prohibitive for tasks such as real-time decision-making, parametric studies, or process optimization.

This chapter explores the development and application of a ROM for predicting steel grade intermixing during ladle changeover. ROMs have been successfully applied across various fields, including fluid dynamics [160], structural dynamics [88], and electromagnetic systems [208]. They serve as an effective tool for approximating FOMs, significantly reducing the computational cost required to obtain numerical solutions in parametric settings [150, 76, 160]. Here, we introduce a non-intrusive, data-driven ROM tailored to the tundish grade mixing problem. Specifically, the proper orthogonal decomposition with regression (POD-R) [188] strategy is utilized, where POD is used to obtain the reduced linear approximation subspace [76, 160], and neural-network is employed for the regression task to evaluate the modal coefficients [75, 169, 147].

Artificial Neural Networks (ANNs) have become a fundamental and powerful tool in machine learning, particularly well suited for supervised learning in data-driven science and engineering [160, 74, 153, 107, 95, 20]. They have been widely applied to forecasting problems in complex dynamic systems [21, 56, 74, 79, 129, 103, 111]. However, conventional feedforward ANNs trained via backpropagation can be computationally expensive to optimize, even with advancements such as stochastic gradient descent and hardware innovations like GPU-based processing. Recurrent neural networks (RNNs) are particularly effective for handling temporal data in dynamical systems [56, 96, 6, 11, 154, 37], as they inherently capture sequential dependencies. Among these, long short-term memory (LSTM) networks have demonstrated high accuracy and reliability [30, 37, 78, 200, 198, 211, 119, 70], but they tend to be data-hungry, requiring long observation times and substantial computational resources for full training [137, 58].

The Reservoir Computing (RC) [87, 197, 117], along with its closely related models, the Echo State Network (ESN) [86, 116] and liquid state machine (LSM) [118, 65], represent the specialized variants of Recurrent Neural Networks (RNNs) in which only the output layer is trained, while the internal network weights remain fixed. This approach significantly reduces the computational complexity, as it requires only a simple and efficient least-squares computation rather than the costly non-linear optimization typically needed for a fully trained RNNs. A particularly striking outcome is that, despite this substantial simplification, RC can still achieve competitive forecasting performance, even for chaotic or spatio-temporally complex systems [27, 29, 45, 115, 141, 140, 216]. RC is especially effective when a full-state observation is available. In contrast, more sophisticated RNNs variants, such as LSTMs, generally outperform it when only a reduced set of variables is accessible [200, 198]. Nonetheless, RC remains widely adopted, not only due to its computational efficiency and ease of training but also because of its surprising accuracy [18].

Full-order simulations of steel grade intermixing during ladle changeover operations, which involve multiphase flows, are computationally expensive and require long-time integration, as these operations typically last for 10-15 minutes (physical time). This high computational cost hinders parametric studies and limits the availability of training data for surrogate modelling. To overcome these challenges, we develop a proper Orthogonal decomposition and reservoir computing based reduced order modelling framework.

In this study, the POD-RC-ROM approach is used to predict the intermixing time during ladle changeover operations. The framework integrates the POD and RC model, where POD is utilized to obtain a low-dimensional representation of the tundish system dynamics, and RC is employed as an efficient non-linear regression technique to approximate the temporal evolution of modal coefficients. Integrating reservoir computing architectures into ROM frameworks presents multiple advantages. A primary benefit lies in the significantly lower training cost, as only the output weights are trained, eliminating the need for backpropagation. RC is also well-suited for capturing temporal dynamics, leveraging the reservoir to project inputs into a high-dimensional space through the reservoir. Among RC approaches, ESNs are particularly notable for their echo state property, which ensures long-term stability and efficiency in modelling complex dynamical systems.

Compared to conventional deep learning models such as RNNs, LSTM networks [137] and Deep Neural Networks (DNNs) [102], RC methods offer comparable accuracy while requiring significantly fewer training data and computational resources [58]. As reported in [18], RC models can achieve remarkable predictive performance with reduced training requirements. By leveraging the computational efficiency and accuracy of the POD-RC-ROM framework, this study aims to enable fast and reliable predictions for real-time monitoring and optimization of ladle changeover operations.

The main contributions of this chapter are as follows:

- We develop a non-intrusive reduced order modelling framework based on proper orthogonal decomposition and reservoir computing (POD-RC-ROM) to predict grade intermixing time in continuous casting processes. Specifically, we leverage Echo State Networks (ESNs) to model the temporal dynamics of the modal coefficients, enabling efficient and accurate forecasting without the need for backpropagation. This leads to significant simplification in training while maintaining high predictive accuracy, as demonstrated in comparison with traditional machine learning architectures [18].
- We demonstrate that the proposed POD-RC-ROM achieves accurate predictions with limited training data, significantly reducing both the computational cost and training time. The framework's efficiency and scalability make it well-suited for real-time industrial applications involving complex multiphysics phenomena.

- We evaluate the POD-RC-ROM in the extrapolation regime by comparing its predictions against high-fidelity, three-dimensional, transient, multiphase flow simulations under industrially relevant conditions. The reduced order model accurately captures the grade intermixing occurring during ladle changeover operations, demonstrating its robustness and generalization capability beyond the training regime even with limited training data.
- We present a computationally efficient and scalable modelling framework that enables real-time process monitoring, optimization, and design-space exploration in continuous casting. This offers a practical tool to support more sustainable and intelligent steelmaking operations.

9.2 Reduced order model

In this work, we integrate POD with reservoir computing to develop a data-driven reduced order model [188]. The framework consists of two stages:

- **Offline stage**: First, full-order high-fidelity solutions are obtained by solving the FOM for parameters of interest. Following this, a reduced basis subspace is obtained by performing POD of a set of high-fidelity snapshots. Once the reduced basis subspace is obtained, the original snapshots are projected onto this reduced subspace to get the corresponding parameter-dependent modal coefficients. Subsequently, reservoir computing is then employed to approximate the mapping between parameters and modal coefficients. This stage is computationally expensive but is performed only once.
- **Online stage**: Given a new parameter, the trained reservoir computing model evaluates the corresponding modal coefficients. The reduced order solution is then reconstructed as a linear combination of the POD basis functions weighted by these modal coefficients. This enables efficient exploration of the parameter space at a significantly lower computational cost.

In the context of a time-dependent problem with parameters, we define the solution manifold $\mathcal{M}$ as the set of all possible solutions of the FOM system of equations (mentioned in the previous section) for varying time and parameters, expressed as:

$$\mathcal{M} = \{C(t;\mu) \mid (t,\mu) \in \mathcal{T} \times \mathcal{P}\}.$$

Its discrete counterpart, denoted as M_h, consists of the corresponding numerical solutions and is given by

$$\mathcal{M}_h = \{C_{N_h}(t;\mu) \mid (t,\mu) \in \mathcal{T} \times \mathcal{P}\} \subset \mathbb{R}^{N_h}.$$

Let us consider a set of parameters $\mathcal{K} = \{\mu_1, \ldots, \mu_{N_p}\} \subset \mathcal{P}$, a finite training set sampled from the parameter domain $\mathcal{P}$. For each parameter $\mu_k \in \mathcal{K}$, we denote a set of time instances $t_j^{(k)} \in \{t_1^{(k)}, \ldots, t_{N_t^{(k)}}^{(k)}\} \in (0, T] \subset \mathcal{T}$, where the number of time instances $N_t^{(k)}$ may vary for each parameter, drawn from the time domain $\mathcal{T}$. The FOM is evaluated for each μ_k, producing a different number of snapshots per parameter. Consequently, the total number of snapshots is given by $N_s = \sum_{k=1}^{N_p} N_t^{(k)}$. Once the full-order solutions are computed, they are assembled column-wise in a snapshots matrix $\mathcal{S} \in \mathbb{R}^{N_h \times N_s}$, as follows:

$$\mathcal{S} = \begin{bmatrix} C_1(t_1;\boldsymbol{\mu}_1) & \cdots & C_1\left(t_{N_t^{(1)}};\boldsymbol{\mu}_1\right) & C_1(t_1;\boldsymbol{\mu}_2) & \cdots & C_1\left(t_{N_t^{(N_k)}};\boldsymbol{\mu}_{N_k}\right) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{N_h}(t_1;\boldsymbol{\mu}_1) & \cdots & C_{N_h}\left(t_{N_t^{(1)}};\boldsymbol{\mu}_1\right) & C_{N_h}(t_1;\boldsymbol{\mu}_2) & \cdots & C_{N_h}\left(t_{N_t^{(N_k)}};\boldsymbol{\mu}_{N_k}\right) \end{bmatrix} \tag{9.1}$$

9.2.1 Proper orthogonal decomposition

Proper Orthogonal Decomposition (POD) is a widely used linear approximation method in reduced order modelling (ROM), particularly in computational fluid dynamics applications over the past decades [160]. In this work, POD is employed to derive an optimal orthonormal basis and construct a reduced basis subspace that effectively approximates the full-order solutions. The orthonormal basis is obtained by computing POD modes using singular value decomposition (SVD) [183], as follows:

$$\underbrace{\mathcal{S}}_{N_h \times N_s} = \underbrace{\mathbf{U}}_{N_h \times N_h} \underbrace{\mathbf{\Sigma}}_{N_h \times N_s} \underbrace{\mathbf{V}^T}_{N_s \times N_s} \Rightarrow \underbrace{\mathcal{S}}_{N_h \times N_T} \approx \underbrace{\hat{\mathbf{U}}}_{N_h \times r} \underbrace{\hat{\mathbf{\Sigma}}}_{r \times N_s} \underbrace{\hat{\mathbf{V}}^T}_{N_s \times N_s}, \tag{9.2}$$

where, $\mathbf{U} = \{\varphi_1 | \ldots | \varphi_{N_h}\} \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{V} = \{\mathbf{w}_1 | \ldots | \mathbf{w}_{N_s}\} \in \mathbb{R}^{N_s \times N_s}$ are orthonormal matrices and hence, unitary matrices, $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbb{I}$, where $\mathbb{I}$ is the identity matrix. The Columns of $\mathbf{U}$ are the left singular vectors, also known as POD modes, and the columns of $\mathbf{V}$ are the right singular vectors of the snapshot matrix $\mathcal{S}$. The diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{N_h \times N_s}$ contains L non-zero

real singular values arranged in descending order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_L > 0$, indicating the energy contribution of the corresponding modes. According to the Schmidt–Eckart–Young–Mirsky theorem [53], the objective is to approximate the column of $\mathcal{S}$ using a reduced basis subspace spanned by the first $r < L$ dominant left singular vectors, thus identifying a low-dimensional representation of the original data.

$$\mathcal{V} = \{\varphi_1, \ldots, \varphi_r\} \in \mathbb{R}^{N_h \times r}. \tag{9.3}$$

The reduced basis subspace is obtained by selecting the first r singular values and their corresponding left singular vectors, also referred to as POD modes. The resulting POD modes satisfy the following minimization problem:

$$\underset{\mathcal{V} \in \mathbb{R}^{N_h \times r}}{\operatorname{argmin}} \left\| \mathcal{S} - \mathcal{V}\mathcal{V}^T\mathcal{S} \right\|_F, \quad \text{where} \quad \mathcal{V}^T\mathcal{V} = \mathcal{I}, \tag{9.4}$$

where $\|\cdot\|_F$ represents the Frobenius norm, $\mathcal{V}$ is the reduced basis subspace of dimension r and $\mathcal{S}$ is the snapshots matrix. Thus, the reduced basis subspace is the set of vectors that minimize the distance between the original snapshots and their projection onto the subspace spanned by the POD modes.

The energy retained by the first r POD modes is determined by the expression in (9.5), where r is chosen such that the cumulative energy $E(r)$ exceeds a specified threshold.

$$E(r) = \frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{L} \sigma_i^2}. \tag{9.5}$$

Once the reduced basis subspace has been constructed, the reduced order solution $C_{rb}(t_k, \boldsymbol{\mu}_k)$ is obtained, which serves as an approximation to the high fidelity full order solution $C_h(t_k, \boldsymbol{\mu}_k)$:

$$C_h(t_k; \boldsymbol{\mu}_k) \approx C_{rb}(t_k; \boldsymbol{\mu}_k) = \sum_{j=1}^{r} [\mathcal{V}^T C_h(t_k; \boldsymbol{\mu}_k)]_j \varphi_j, \tag{9.6}$$

where $(\mathcal{V}^T C(t_k, \boldsymbol{\mu}_k))_j$ is the modal coefficient corresponding to the j-th mode.

9.2.2 Reservoir computing

Reservoir computing (RC) is a machine learning framework designed to efficiently capture temporal dependencies in dynamical systems. It focuses primarily on input-output relationships and is a branch of neuromorphic computing, which mimics the brain's efficient information processing at low energy

costs [87, 187]. From a machine learning perspective, RC models fall under the category of RNNs and are particularly effective for modelling dynamical systems. RC excels in tackling even the most difficult problems, including those involving chaotic [141] or complex spatio-temporal behaviours [140], which an optimized RC can process with ease.

The RC technique has been developed through three distinct methods: liquid state machine, echo state network, and backpropagation decorrelation learning rule. Unlike traditional recurrent neural networks, which often require high computational cost and complex training procedures when applied to non-linear dynamical systems. RC offers an effective alternative. It leverages a fixed, randomly initialized dynamical system as a reservoir, with the training occurring only on the output layer via a regularized linear least-squares optimization procedure. The typical RC consists of three components:

- Input layer: Maps input data into the reservoir, often applying a simple transformation or projection.
- Reservoir layer: A high-dimensional, fixed, recurrent dynamical system that transforms the input data into complex, temporal patterns.
- Output layer: A trainable layer that maps the updated reservoir state to the desired output, typically using linear regression or other techniques.

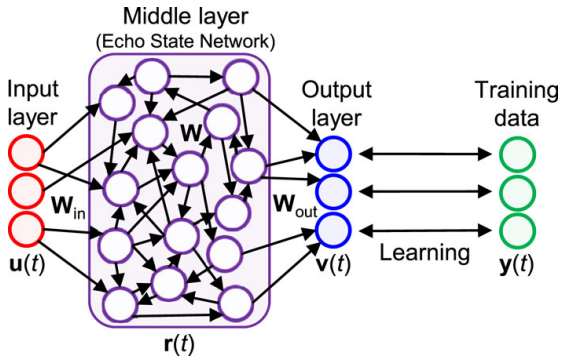


Figure 9.1: Schematic representation of an echo state network, a type of reservoir computing (RC), as defined in (9.7). The network consists of a randomly initialized $N_\mathrm{r} \times N_\mathrm{in}$ input weight matrix W_in that maps the $N_\mathrm{in} \times 1$ input state vector $\mathbf{u}(t)$ into the reservoir. The reservoir dynamics are governed by a randomly selected $N_\mathrm{r} \times N_\mathrm{r}$ recurrent weight matrix $\mathbf{W}$, which updates the internal reservoir state vector $\mathbf{r}$ of size $N_\mathrm{r} \times 1$. Finally, the trained readout weight matrix $\mathbf{W}_\mathrm{out}$, of dimensions $N_\mathrm{out} \times N_\mathrm{r}$, is used to produce the final output.

Echo state network

In this work, we consider the echo state recurrent neural networks, the schematic is presented in Figure 9.1 [85]. An echo state network (ESN) is a random recurrent neural network. Given a predefined input $\mathbf{u}(t) \in \mathbb{R}^{N_{\text{in}}}$ and target output $\mathbf{y}(t) \in \mathbb{R}^{N_{\text{out}}}$, the node state $\mathbf{r}(t+1) \in \mathbb{R}^{N_{\text{r}}}$ and the corresponding output $\mathbf{v}(t+1) \in \mathbb{R}^{N_{\text{out}}}$ are computed as follows:

$$\mathbf{r}(t+1) = \underbrace{(1-\gamma)\mathbf{r}(t)}_{\text{Linear Memory}} + \underbrace{\gamma f\left[\beta\left(\varepsilon \mathbf{W}_{\text{in}}\,\mathbf{u}(t) + \mathbf{W}\mathbf{r}(t) + \eta\mathbf{b}\right)\right]}_{\text{Nonlinear Activation}}, \tag{9.7}$$

$$\mathbf{v}(t+1) = \mathbf{W}_{\text{out}}\mathbf{r}(t+1), \tag{9.8}$$

where $f : \mathbb{R} \mapsto \mathbb{R}$ is the non-linear activation function, drawing from terminology in the machine learning and neural network literature to replicate the behaviour of a biological network that activates when a certain voltage threshold is reached. The most common choice of f would be $\tanh(\cdot)$, which applies the scalar hyperbolic tangent function element-wise, and other activation functions, such as alternative sigmoidal functions and the ReLU function, are widely used in neural network theory, though ReLU is less commonly employed in RC [22]. $\mathbf{b} \in \mathbb{R}^N$ is a constant bias vector, initialized as a random vector following a standard normal distribution. The parameters $\beta \in \mathbb{R}$, $\varepsilon \in \mathbb{R}$, and $\eta \in \mathbb{R}$ are the hyperparameters governing the dynamics of the RC model. Specifically, σ is a scaling factor that adjusts the amplitude of the input to the activation function, ε is a scaling factor applied to the input-to-reservoir weights, controlling the influence of the input signal on the reservoir state, and η scales the contribution of the bias vector $\mathbf{b}$ to the reservoir dynamics. The dimensions N_{in}, N_{r}, and N_{out} denote the number of input features, reservoir node states, and output features, respectively. The reservoir maps input $u(t)$ into N_{r} dimensional space, where a sufficiently large N_{r} ensures accurate computation of $\mathbf{v}(t)$ by ESN. The weight matrices connecting the input to the reservoir, $\mathbf{W}_{\text{in}} \in \mathbb{R}^{N_{\text{r}} \times N_{\text{in}}}$, and those within the reservoir, $\mathbf{W} \in \mathbb{R}^{N_{\text{r}} \times N_{\text{r}}}$, are initialized randomly, typically following a standard normal distribution and remain fixed during training and testing phase. As highlighted in (9.7), the right-hand side consists of two components: a linear memory term and a nonlinear activation term, which are combined via a leakage or decay rate $0 \leq \gamma \leq 1$. $\mathbf{W}$ should be designed with specific properties, such as spectral radius for convergence [29, 92], sparsity [115, 140, 199], or other considerations related to the echo state property [22]. In RC, the only trainable parameter is the weight matrix $\mathbf{W}_{\text{out}} \in \mathbb{R}^{N_{\text{out}} \times N_{\text{r}}}$ between the reservoir and the output layer.

The output layer represent the RC output $\mathbf{v}(t+1) = y(t+1)$ as a linear transformation of $\mathbf{W}_{\text{out}}$ and the reservoir state $\mathbf{r}(t+1)$, as shown in (9.8). The RC model is trained using supervised learning through regularized least-squares regression. The optimal $\mathbf{W}_{\text{out}}$ is determined by solving the following optimization problem:

$$\underset{\mathbf{W}_{\text{out}}}{\arg\min} \sum_{t=1}^{n} \|\mathbf{y}(t) - \mathbf{W}_{\text{out}}\, \mathbf{r}(t)\|_2^2 + \lambda \, \|\mathbf{W}_{\text{out}}\|_F^2 , \tag{9.9}$$

where λ is a regularization parameter that prevents over fitting to the training data. The solution to (9.9) is obtained in the least square sense using Tikhonov regularization, as follows:

$$\mathbf{W}_{\text{out}} = \sum_{t=1}^{n} \mathbf{y}(t)\mathbf{r}(t)^{\top} \left(\lambda \mathbf{I} + \sum_{t=1}^{n} \mathbf{r}(t)\mathbf{r}(t)^{\top} \right)^{-1} \tag{9.10}$$

This approach significantly reduces the computational cost of training compared to conventional neural network architectures, as only the output weights are optimized by a simple and efficient least squares computation.

Evaluation of modal coefficients

The modal coefficients are obtained via Galerkin projection $\mathcal{V}^T C_{N_h}(t_k; \boldsymbol{\mu}_k)$. As mentioned earlier, we employ the RC model to obtain a reliable approximation, $\boldsymbol{\Theta}$, for the following input-output relation:

$$\boldsymbol{\Theta} : (t_k; \boldsymbol{\mu}_k) \in \mathbb{R}^{N_s} \mapsto \left[\mathcal{V}^T C_{N_h}(t_k; \boldsymbol{\mu}_k) \right]_{j=1}^{r} \in \mathbb{R}^r. \tag{9.11}$$

During the online phase, the reduced solution for any new time instant t_{new} and new parameter $\boldsymbol{\mu}_{new}$ can be computed efficiently as follows:

$$C_{rb}(t^*; \boldsymbol{\mu}^*) = \sum_{j=1}^{r} \boldsymbol{\Theta}_j(t^*; \boldsymbol{\mu}^*)\, \varphi_j, \tag{9.12}$$

here, $\boldsymbol{\Theta}_j(t^*; \boldsymbol{\mu}^*)$ represents the modal coefficients at the new time instant and parameter, with each φ_j corresponding to the j-th basis function or mode.

9.2.3 POD-RC-ROM algorithm

Algorithm 6 outlines the complete framework for the construction of POD-RC based ROMs. The offline stage involves computing full-order solutions for parameters of interest and the construction of ROM, including the training of the RC model to build a non-linear regression of the map $\boldsymbol{\Theta}$. By leveraging the inherent separation between the training and evaluation phases of RC models, we effectively satisfy the prerequisite for ensuring the decoupling of offline-online stages and computational efficiency during the online stage. Once trained, the online stage rapidly predicts reduced order solutions for new input parameters with minimal computational cost. This framework enables accurate and efficient approximations, making real-time inference of the QoI feasible.

Algorithm 6: POD-Reservoir Computing based ROM

1: **Offline Stage:**
2: Collect the parameter set $\mathcal{K} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{N_p}\}$;
3: Compute full-order solutions: $\{C_{N_h}(t_j^{(1)}; \boldsymbol{\mu}_1), \ldots, C_{N_h}(t_j^{(N_p)}; \boldsymbol{\mu}_{N_p})\}$;
4: Extract POD basis functions $\{\varphi_1, \ldots, \varphi_r\}$ via Singular Value Decomposition (SVD);
5: Construct the reduced basis subspace: $\mathcal{V} = \{\varphi_1, \ldots, \varphi_r\}$;
6: Perform Galerkin projection to evaluate modal coefficients;
7: Train the reservoir computing (RC) model to obtain the approximation $\boldsymbol{\Theta}(\cdot;\cdot)$;
8: **Online Stage:**
9: Evaluate the output $\boldsymbol{\Theta}(t^*; \boldsymbol{\mu}^*)$ of the RC model for a new time-parameter pair $(t^*, \boldsymbol{\mu}^*)$;
10: Compute the reduced-order solution: $C_{rb}(t^*; \boldsymbol{\mu}^*) = \mathcal{V}\boldsymbol{\Theta}(t^*; \boldsymbol{\mu}^*)$;

Incorporating RC architectures into ROM offers several advantages. One key benefit is the reduction in training costs, as only the output layer is optimized, eliminating the need for backpropagation. Additionally, RC effectively captures temporal features by mapping input data into a high-dimensional space via the reservoir. Specifically, ESNs, a widely used approach within RC, ensure long-term stability due to the echo state property, making them highly efficient for modelling complex dynamical systems. Furthermore, RC can achieve the same level of learning accuracy with less training data and computational resources compared to conventional deep learning architectures, such as RNNs,

long short-term memory (LSTM) networks, and deep feedforward neural networks (DNNs). In this work, a reservoir computing architectures library `reservoirPy` (https://github.com/reservoirpy/reservoirpy) [194] is employed to implement RC within the ROM framework.

9.3 Results and discussion

Database creation: Prior to applying the model order reduction strategies, it is necessary to prepare a database consisting of full-order solutions. Since the primary focus is on the outlet to obtain the QoI, the snapshot database is constructed by considering only the outlet region. We start collecting snapshots where the concentration of the new steel grade is 1% at the outlet. A database is then generated, consisting of five numerical experiment snapshots of the new steel grade, $C(t_k, \mu_k)$, obtained from the FOM. We consider the first four numerical experiments as a training set and the last one as a test set.

Further, we perform model reduction using POD on the training set snapshots to obtain the reduced basis subspace. Figure 9.2 shows both singular values and cumulative energy corresponding to the rank of POD modes. We notice a sharp decay in singular values, with the first mode alone capturing 99.99 % of the cumulative energy. This rapid decay enables the construction of an efficient ROM using a low-dimensional linear approximation subspace.

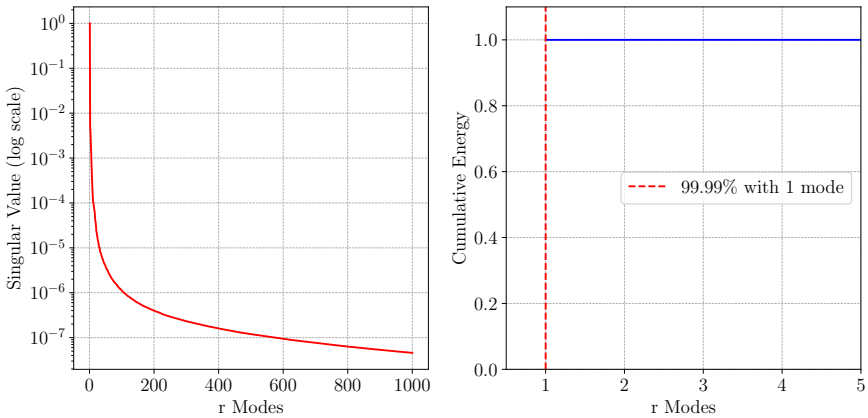


Figure 9.2: Singular values and cumulative energy distribution of the training dataset used in developing the POD-RC-ROM for predicting intermixing time.

We utilize the first POD mode to construct a reduced basis subspace and then follow the procedure outlined in section 9.2.1 to obtain modal coefficients via Galerkin projection. To enable rapid online computations, reservoir computing is employed to establish a mapping between the parameters and modal

coefficients. This approach facilitates the development of a data-driven POD-RC based ROM for monitoring steel grade intermixing time.

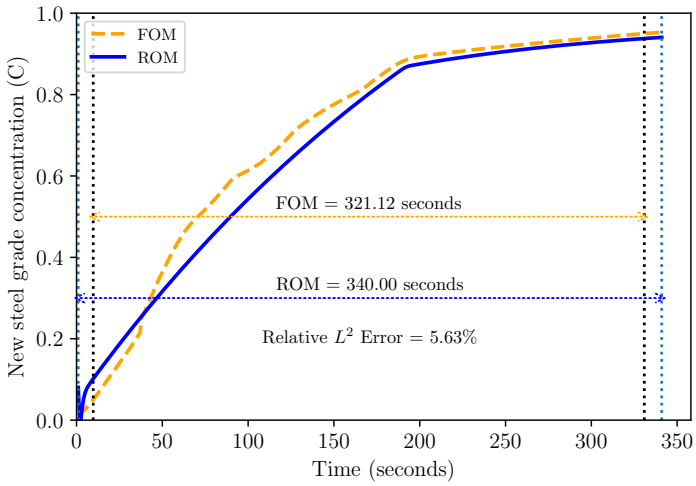


Figure 9.3: Prediction of intermixing time by the POD-RC-ROM for test set parameter in the extrapolation regime.

Figure 9.3 presents the prediction obtained from the POD-RC-ROM for the test parameter, compared against the FOM results, and additionally, it reports the relative L_2 error of the prediction, which is 5.63%. The results demonstrate that the ROM predictions are both qualitatively and quantitatively accurate, closely following the trend of the FOM results.

The relative L_2 error of ROM predictions for both the training and test set parameters is shown in Figure 9.4. The observed decrease in error for the training cases suggests that the ROM is progressively learning the system dynamics and improving its approximation capability. The reduction in error indicates that the reservoir computing framework successfully captures the dynamics of the system, leading to enhanced predictive accuracy. For the test case, even though there's a slight increase in relative L_2 error, the ROM provide reliable results as shown in Figure 9.3.

The decreasing trend in training errors confirms the effectiveness of the reservoir computing-based ROM in learning the reduced order dynamics. The marginal increase in the test set error suggests that further refinements, such as incorporating additional training samples or adjusting hyper-parameters could enhance the generalization capability of the model. We use the resulting POD-RC-ROM to predict intermixing time. Figure 9.5 compares the intermixing time obtained from the FOM and ROM. The results show that the ROM predictions closely align with the FOM values for the train and test parameters, with a negligible difference compared to the FOM.

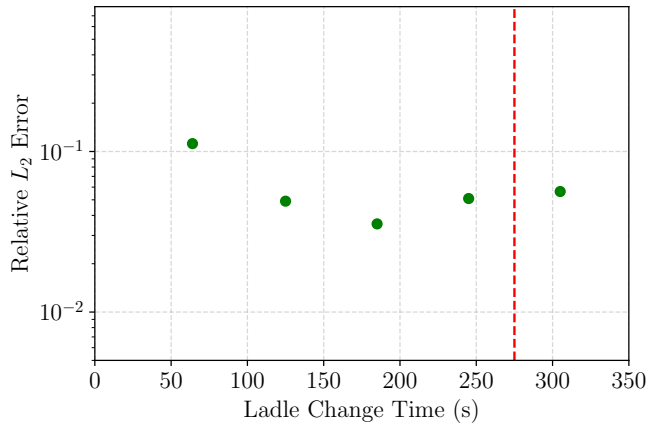


Figure 9.4: Prediction relative L_2 error of POD-RC-ROM for both train and test set parameters, again the red vertical line separates the training and test sets, with the left region representing the training set and the right region representing the test set.

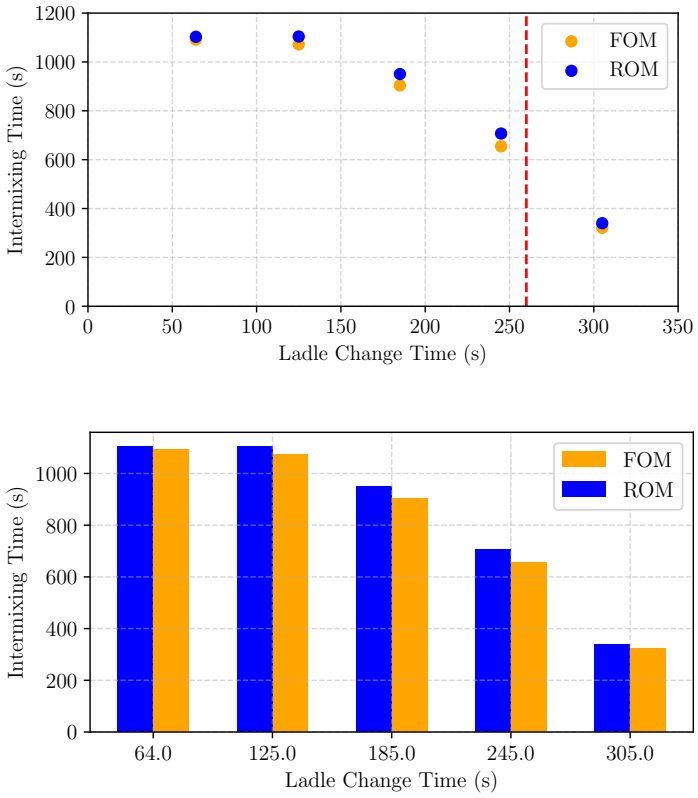


Figure 9.5: Comparison of intermixing time obtained from the FOM and the POD-RC based ROM. The red vertical line separates the training and test sets, with the left region representing the training set and the right region representing the test set.

9.4 Conclusion

Generally, for industrial-scale systems, obtaining full-order high-fidelity solutions of these physical processes is computationally expensive and requires long-time integration to simulate the entire operations cycle. This makes it prohibitive to perform extensive parameter exploration, which is crucial for the development of ROM.

To address this issue, a POD-Reservoir Computing based ROM is employed to evaluate the intermixing time, the key QoI during ladle changeover operations. RC is integrated within the POD-R ROM strategy to accurately capture the evolution of parameter-dependent temporal coefficients, particularly in scenarios with limited training data. In such cases, RC achieves competitive predictive accuracy with fewer training samples, reduced computational resources, and shorter training times compared to conventional neural network architectures. The POD-RC-ROM developed in this study demonstrates that the ROM predictions closely align with the full-order solution results, with slight overestimation of the intermixing time. The POD-RC-ROM has several advantages for industrial-scale operations, offering cost-effective, time-effective, and accurate solutions for predicting complex phenomena. Furthermore, it can be effectively leveraged in design optimization processes, enabling more efficient decision-making and enhancing process control.

Chapter 10

Conclusions and perspectives

This thesis presents a comprehensive study of tundish metallurgical flow through the development and application of Full Order Models (FOMs) and Reduced Order Models (ROMs). Motivated by the industrial need for accurate and efficient prediction of key flow characteristics in continuous casting tundishes. This work addresses both the physical understanding of flow behaviour within the tundish and the development of computational strategies to overcome the challenges associated with high-fidelity full-order simulations.

Full-order simulations of steady-state operations, performed under both isothermal and non-isothermal conditions, demonstrate that buoyancy has a negligible effect on the overall velocity field, thereby justifying the use of isothermal models for certain analyses. These simulations serve as the foundation for accurate RTD analysis and volume partitioning, offering critical insights into flow behaviour and mixing efficiency. Additionally, transient scalar transport simulations are conducted to obtain the RTD curve, which is validated against the water-model experiment results. Further, a sensitivity study on the influence of inlet velocities on the RTD curve is presented. To investigate ladle changeover operations, three-dimensional transient multiphase turbulent full-order simulations are performed, highlighting the significant impact of ladle change timing on steel grade intermixing and melt stratification.

To address the high computational cost of full-order simulations, several model order reduction strategies are explored. A projection-based ROM for RTD analysis in continuous casting tundish was developed and validated against both experimental data and full-order simulation results. The projection based ROM accurately reproduced RTD curves with minor discrepancies, demonstrating its potential as an efficient surrogate for full-scale simulations. However, its practical application depends on access to the CFD solver's source code, an important limitation in industrial settings where commercial software

is commonly used. Data-driven ROMs, such as the POD-RBF-ROM, offer an efficient alternative for parametric analysis using full-order simulation data, without modifying the underlying solver. The model shows excellent accuracy in the interpolation regime, closely matching full-order simulations, but exhibits minor discrepancies in the extrapolation regime, particularly in the initial phase, with slight fluctuations and differences in concentration prediction.

For industrial-scale systems, obtaining full-order high-fidelity solutions is computationally expensive and time-consuming, limiting parameter exploration. To address this, a POD-Reservoir Computing (POD-RC) based ROM is employed to predict grade intermixing and evaluate intermixing time during ladle changeover operations. By integrating Reservoir Computing within the POD-R ROM, the model effectively captures parameter-dependent temporal coefficients with limited training data. The POD-RC-ROM demonstrates competitive accuracy with fewer training samples and reduced computational resources compared to conventional neural networks. The POD-RC-ROM predictions closely match full-order solutions, with slight overestimation of intermixing time. This approach offers cost-effective, time-efficient, and reliable predictions, making it suitable for industrial-scale operations and design optimization. The computational efficiency achieved by the resulting ROMs for both steady-state and ladle changeover operations is reported in Appendix B.

Beyond conventional ROM approaches, this thesis introduces two advanced non-linear model reduction techniques tailored for advection-dominated problems. The NNsPOD-ROM framework leverages manifold transformations to construct efficient ROMs without requiring explicit knowledge of the underlying advection dynamics. Further, the cross-correlation based snapshot registration technique is presented, both approaches accelerate the Kolmogorov n-width decay and enable the construction of efficient ROM using linear methods. Both methods are validated on benchmark problems, demonstrating strong generalization and high accuracy under suitable conditions. While the NNsPOD-ROM is particularly effective for complex, non-linear dynamics where data-driven manifold learning provides added flexibility, the cross-correlation approach offers a computationally robust and lightweight alternative better suited for problems with coherent transport features and limited training data.

In summary, this work advances the modelling and simulation of tundish operations through a combination of physical insight and methodological innovation. The integration of classical approaches with machine learning driven

ROMs paves the way for computationally efficient, real-time tools that support optimization, control, and continued advancements in industrial steelmaking.

Future directions

While the ROM strategies developed in this work demonstrate strong performance across a range of advection-dominated problems, several limitations and opportunities remain for future enhancement. One promising avenue for future research is the integration of advanced machine learning architectures, such as LSTMs or transformers, into the ROM framework, particularly for tasks like particle tracking and non-linear transport phenomena.

Another important direction involves improving snapshot registration techniques, especially for cases where transported structures exhibit complex deformations or advection-diffusion behavior. Recent advances in the medical imaging domain, particularly deep learning-based methods such as VoxelMorph [7], LapIRN [125], and FAIM [101], offer promising solutions to the limitations of conventional registration in ROM contexts. These frameworks have demonstrated high accuracy and computational efficiency and could be adapted to enhance model reduction workflows, especially in scenarios characterized by slow Kolmogorov n-width decay. Leveraging these techniques could significantly expand the applicability of ROMs to more diverse and challenging physical systems, including those with non-linear transport structures or evolving feature geometries.

As sustainability becomes an increasingly important goal in industrial processes, the development of reduced order methods can play a pivotal role in making steelmaking more environmentally friendly. Future research can focus on using ROMs to optimize key factors such as energy consumption, material usage, and emissions during tundish operations. By enabling more precise control of the casting process, ROMs can help minimize energy waste and reduce the need for excessive energy input. Moreover, by optimizing flow behavior and improving grade intermixing efficiency, ROMs can reduce material losses and enhance both product quality and conserve resources.

Acknowledgements

I would like to express my sincere gratitude to my advisors, Giovanni Stabile and Gianluigi Rozza, for their guidance and supervision throughout my PhD journey. I am especially thankful for the autonomy they granted me, allowing me to explore ideas at my own pace, and for the trust they placed in my work.

I thank Giovanni in particular for his precise and insightful responses on both core theoretical topics and software-related matters, including aspects of software development. His constructive criticism has been immensely helpful across many areas of my research.

I thank Gianluigi for his humility and kindness, as well as for managing all the project administrative arrangements. I am especially grateful for his patience in listening to students' concerns and his prompt efforts to resolve certain hurdles we encountered.

I am grateful for the many fruitful discussions and interactions with my industrial collaborators, Mattia Giusseppe Barra, Gianluca Bazzaro, Gabriele Guestaferro, and Gianfranco Marconi, from the Danieli Research Centre, Buttrio, Italy. I would also like to thank the mathLab collaborators and group members for their valuable insights, support, and expertise in their respective fields.

I would like to thank Tommaso Taddia and Angelo Iollo from TEAM MEMPHIS at Inria and IMB, Bordeaux, France, and also, Simon Brillet and Coralie Hay from VALOREM, for hosting my research visit. A special thank you to Anne Laure for managing all the administrative work and helping me settle down in Bordeaux. I also want to thank the acquaintance for introducing me to the vélo pour les étudiants plan, which made it easy for me to get around the city.

I heartly thank student secretaries, Evy, Marco, and Riccardo, for handling all the administrative work, but also for their constant kindness and warmth, which made every visit to the office feel truly welcoming.

Finally, I would like to thank the referees, Karen Veroy-Grepl, Department of Mathematics and Computer Science, Eindhoven University of Technology – TU/e, and Annalisa Quaini, Department of Mathematics, University of Houston, for reviewing the thesis. I also extend my sincere thanks to the members of the defense committee: external members Tommaso Taddei (Inria Bordeaux), Andrea Mola (IMT School for Advanced Studies Lucca), Michele Girfoglio (University of Palermo), and Davide Torlo (University of Rome, La Sapienza); and internal members Giovanni Noselli and Gianluigi Rozza.

Trieste, the hidden gem of a city in Italy, aside the Adriatic Sea. It has been a pleasant host. I am especially grateful for the stunning view of Trieste from my office at SISSA, which sort of motivated me to show up regularly. The campus, surrounded by greenery, offered one of the most beautiful workplace views imaginable. I am thankful for countless sunsets and peaceful evening walks, which helped me unwind after long days of work.

This journey of 3.5 years has been long and demanding, and I could not have made it through without the unwavering support and companionship of the incredible friends I was fortunate to meet along the way. I do not wish to mention names, but I would like to acknowledge the quality time spent with friends throughout this journey. Their presence, support, and shared moments made this experience all the more enjoyable and memorable. I want to take a moment to acknowledge them here, so they know these memories remain close to my heart.

First, I would like to thank the 111 officemates for their warm welcome during my initial days in Trieste and for making me feel comfortable. Introducing me to Italian culture, the uniqueness of Trieste, and helping me settle down in Trieste. I cannot imagine my stay in Trieste without some beautiful souls with whom I spent most of my leisure time. I cherish every coffee break, lunches (kudos to early lunch people at SISSA), ping pong, aperitivo, Barcola evening, osmizas, movie nights, dinner, oggi, and the long evenings spent together.

Appendix A

Neural-networks hyperparameters

A.1 NNsPOD-ROM

A.1.1 Scaling of data

Before starting the training of neural networks, we scale the data ranging between the interval [0,1] to improve the prediction accuracy of neural networks. We use Min-Max scaling for the scaling of training data between the interval [0, 1]. The Min-Max scaling for the given matrix U is:

$$U_{ij}^{\text{scaled}} = \frac{U_{ij} - \min\limits_{j=1,\ldots,N_s}(U_{ij})}{\max\limits_{j=1,\ldots,N_s}(U_{ij}) - \min\limits_{j=1,\ldots,N_s}(U_{ij})}, \tag{A.1}$$

where $i = 1, 2, ..., N_h$, is the index of cell centroids or nodes in the discretized domain Ω. The minimum and maximum values are stored and used to rescale the predicted results back to the original scale.

A.1.2 Neural networks' Architecture

In our numerical experiments, we utilize Artificial Neural Networks (ANNs), specifically the multi-layer perceptron (MLP), to implement both InterpNet and ShiftNet. The architecture and hyperparameters used for each test case are summarized below.

Neural network parameters	InterpNet	ShiftNet
Layers	[10, 10]	—
Activation function	SoftPlus	LeakyReLU
Learning rate	0.03	0.0023
Accuracy threshold	10^{-6}	10^{-2}
Optimizer	Adam	Adam

Table A.1: ANNs hyperparameters for 1D travelling wave test case.

Neural network parameters	InterpNet	ShiftNet
Layers	[40, 40]	[10, 4]
Activation function	Sigmoid	LeakyReLU
Learning rate	0.0023	0.03
Accuracy threshold	1.5×10^{-6}	0.0024
Optimizer	Adam	Adam

Table A.2: ANNs hyperparameters for 2D Isentropic Convective Vortex test case.

Neural network parameters	InterpNet	ShiftNet
Layers	[30, 30]	[10, 4]
Activation function	LeakyReLU	LeakyReLU
Learning rate	0.0023	0.03
Accuracy threshold	5×10^{-7}	0.0005
Optimizer	Adam	Adam

Table A.3: ANNs hyperparameters for 2D two-phase flow test case.

A.2 Reservoir computing hyperparameters

RC hyperparameter	**Value**
Number of neurons	300
Activation function	Sigmoid
Leaky rate	0.0023
Spectral radius	0.01
Regularization tolerance	5×10^{-9}

Table A.4: RC hyperparameters considered to construct predictive ROM for steel grade intermixing.

Appendix B

FOMs and ROMs: Computational resources & performance

B.1 Tundish steady-state operation

Expt No.	Wall-clock time	CPU time (core-hours)	Cores Allocated
1	2.81 + 5.38 hours	177.05 + 360 hours	64
2	4.43 + 9 hours	177.2 + 360 hours	40
3	7.26 + 9 hours	283.08 + 360 hours	40
4	6.67 + 9 hours	260.92 + 360 hours	40
5	5.16 + 9 hours	201.55 + 360 hours	40
6	4.24 + 9 hours	165 + 360 hours	40
7	14.17 + 9 hours	555.24 + 360 hours	40
8	3.75 + 9 hours	145.53 + 360 hours	40
9	6.10 + 9 hours	237.70 + 360 hours	40
10	6.54 + 9 hours	256.27 + 360 hours	40
11	10.48 + 9 hours	410.90 + 360 hours	40

Table B.1: Summary of computational resources used for the FOM simulations to obtain the tundish steady-state operation QoIs on the Linux cluster. The wall-clock time refers to the elapsed real time, while the CPU time denotes the cumulative usage across all allocated cores (core-hours). These results correspond to the steady RANS + transient tracer simulations discussed in Table 3.3 of chapter 3.

Intrusive ROM	Offline Wall-clock Time	Offline CPU Time (core-hours)	Online Time (Wall-clock)	Cores Used (Offline / Online)
PMOR	40 hours	80 core-hours	0.04 seconds	2 / 1

Table B.2: Computational resources used for projection-based ROM construction and prediction to obtain the tundish steady-state operation quantity of interest (QoI). The offline phase includes computation of POD modes (whole domain is considered), assembling reduced operators, and solving the reduced-order ODE to obtain the modal coefficients. The online phase refers to the prediction of the desired QoI at the tundish outlet. Here, online evaluation is performed only at the outlet probe location, where the QoI is recorded. Evaluating over the entire domain takes approximately 46 seconds, primarily due to the cost of the dot product operation.

Data-driven ROM	Offline Wall-clock Time	Offline CPU Time (core-hours)	Online Time (Wall-clock)	Cores Used (Offline / Online)
POD-RBF-ROM	40 hours	80 core-hours	0.33 seconds	2 / 1

Table B.3: Computational resources used for the parametric time-dependent data-driven POD-RBF-ROM construction and prediction to obtain the tundish steady-state operation QoIs. The offline phase includes computation of POD modes (whole domain is considered), evaluation of modal coefficients, and RBF regression map construction. The online phase corresponds to the prediction of quantities of interest, the Residence Time Distribution (RTD) curve at the tundish outlet. Again here, the online evaluation is performed only at the outlet probe location, where the QoI is recorded. Evaluating over the entire domain takes approximately 90 seconds, primarily due to the cost of the dot product operation.

B.2 Tundish ladle changeover operation

Expt No.	Wall-clock time	CPU time (core-hours)	Cores Allocated
1	351.6 hours (14.7 days)	78,810 hours	200
2	527.6 hours (22.0 days)	105,318 hours	200
3	396.5 hours (16.5 days)	73,535 hours	192
4	335.7 hours (14.0 days)	64,176 hours	192
5	161.83 hours (6.74 days)	38,203 hours	256

Table B.4: Summary of computational resources used for the ladle changeover FOM transient simulations on the Linux cluster. Wall-clock time refers to the elapsed real time, and CPU time represents the cumulative time across all allocated cores (core-hours). These experiments correspond to those reported in Table 4.6 of chapter 4.

Data-driven ROM	Offline Time (Wall-clock)	Online Time (Wall-clock)	Cores Used (Offline / Online)
POD-RC-ROM	75 seconds	0.42 seconds	1 / 1

Table B.5: Computational resources used for the ladle changeover operation ROM construction and prediction. The offline phase includes the computation of POD modes (only the outlet region of the domain is considered), evaluation of modal coefficients, and the training of the reservoir computing model to construct a regression map. The online phase corresponds to the prediction of the quantities of interest at the tundish outlet.

Bibliography

[1] H. Aling et al. “Nonlinear model reduction for simulation and control of rapid thermal processing”. In: *Proceedings of the American Control Conference*. Vol. 4. Proceedings of the 1997 American Control Conference, Part 3 of 6, June 4–6, 1997. IEEE, 1997, pp. 2233–2238.

[2] Marzieh Alireza Mirhoseini and Matthew J. Zahr. “Model reduction of convection-dominated partial differential equations via optimization-based implicit feature tracking”. In: *Journal of Computational Physics* 473 (Jan. 2023), p. 111739. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2022.111739. URL: http://dx.doi.org/10.1016/j.jcp.2022.111739.

[3] Mehdi Alizadeh, Hossein Edris, and Ahmad Reza Pishevar. “Behavior of Mixed Grade during the Grade Transition for Different Conditions in the Slab Continuous Casting”. In: *ISIJ International* 48.1 (2008), pp. 28–37. ISSN: 1347-5460. DOI: 10.2355/isijinternational.48.28. URL: http://dx.doi.org/10.2355/isijinternational.48.28.

[4] D. Alonso, A. Velazquez, and J.M. Vega. “Robust reduced order modeling of heat transfer in a back step flow”. In: *International Journal of Heat and Mass Transfer* 52.5–6 (Feb. 2009), pp. 1149–1157. ISSN: 0017-9310. DOI: 10.1016/j.ijheatmasstransfer.2008.09.011. URL: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2008.09.011.

[5] David Amsallem, Matthew J. Zahr, and Charbel Farhat. “Nonlinear model order reduction based on local reduced-order bases”. In: *International Journal for Numerical Methods in Engineering* 92.10 (June 2012), pp. 891–916. ISSN: 1097-0207. DOI: 10.1002/nme.4371. URL: http://dx.doi.org/10.1002/nme.4371.

[6] Coryn A L Bailer-Jones, David J C MacKay, and Philip J Withers. “A recurrent neural network for modelling dynamical systems”. In: *Network: Computation in Neural Systems* 9.4 (Jan. 1998), pp. 531–547. ISSN: 1361-6536. DOI: 10.1088/0954-898x_9_4_008. URL: http://dx.doi.org/10.1088/0954-898X_9_4_008.

[7] Guha Balakrishnan et al. "VoxelMorph: A Learning Framework for Deformable Medical Image Registration". In: *IEEE Transactions on Medical Imaging* 38.8 (Aug. 2019), pp. 1788–1800. ISSN: 1558-254X. DOI: 10.1109/tmi.2019.2897538. URL: http://dx.doi.org/10.1109/TMI.2019.2897538.

[8] Francesco Ballarin and Gianluigi Rozza. "POD–Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems". In: *International Journal for Numerical Methods in Fluids* 82.12 (June 2016), pp. 1010–1034. ISSN: 1097-0363. DOI: 10.1002/fld.4252. URL: http://dx.doi.org/10.1002/fld.4252.

[9] Francesco Ballarin et al. "Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization". In: *Journal of Computational Physics* 315 (June 2016), pp. 609–628. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.03.065. URL: http://dx.doi.org/10.1016/j.jcp.2016.03.065.

[10] Francesco Ballarin et al. "Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations". In: *International Journal for Numerical Methods in Engineering* 102.5 (Nov. 2014), pp. 1136–1161. ISSN: 1097-0207. DOI: 10.1002/nme.4772. URL: http://dx.doi.org/10.1002/nme.4772.

[11] T.G. Barbounis et al. "Long-Term Wind Speed and Power Forecasting Using Local Recurrent Neural Network Models". In: *IEEE Transactions on Energy Conversion* 21.1 (Mar. 2006), pp. 273–284. ISSN: 0885-8969. DOI: 10.1109/tec.2005.847954. URL: http://dx.doi.org/10.1109/TEC.2005.847954.

[12] Joshua Barnett and Charbel Farhat. "Quadratic approximation manifold for mitigating the Kolmogorov barrier in nonlinear projection-based model order reduction". In: *Journal of Computational Physics* 464 (Sept. 2022), p. 111348. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2022.111348. URL: http://dx.doi.org/10.1016/j.jcp.2022.111348.

[13] Valerio Battaglia et al. "Steel Thermo-Fluid-Dynamics at Tundish Drainage and Quality Features". In: *steel research international* 84.3 (Oct. 2012), pp. 237–245. ISSN: 1869-344X. DOI: 10.1002/srin.201200123. URL: http://dx.doi.org/10.1002/srin.201200123.

[14] Peter Benner, Serkan Gugercin, and Karen Willcox. "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems".

In: *SIAM Review* 57.4 (Jan. 2015), pp. 483–531. ISSN: 1095-7200. DOI: 10.1137/130932715. URL: http://dx.doi.org/10.1137/130932715.

[15] Ariane Bérard, Bruno Blais, and Gregory S. Patience. "Experimental methods in chemical engineering: Residence time distribution—RTD". In: *The Canadian Journal of Chemical Engineering* 98.4 (Mar. 2020), pp. 848–867. ISSN: 1939-019X. DOI: 10.1002/cjce.23711. URL: http://dx.doi.org/10.1002/cjce.23711.

[16] Kostas Berberidis and Irene Karybali. "A new efficient cross-correlation based image registration technique with improved performance". In: *2002 11th European Signal Processing Conference*. IEEE. 2002, pp. 1–4.

[17] Jiri Blazek. *Computational fluid dynamics: Principles and applications*. Elsevier Science & Technology, May 2015.

[18] Erik Bollt. "On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.1 (Jan. 2021). ISSN: 1089-7682. DOI: 10.1063/5.0024890. URL: http://dx.doi.org/10.1063/5.0024890.

[19] Rémi Bourguet, Marianna Braza, and Alain Dervieux. "Reduced-order modeling for unsteady transonic flows around an airfoil". In: *Physics of Fluids* 19.11 (Nov. 2007). ISSN: 1089-7666. DOI: 10.1063/1.2800042. URL: http://dx.doi.org/10.1063/1.2800042.

[20] Joan Bruna, Benjamin Peherstorfer, and Eric Vanden-Eijnden. "Neural Galerkin schemes with active learning for high-dimensional evolution equations". In: *Journal of Computational Physics* 496 (Jan. 2024), p. 112588. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2023.112588. URL: http://dx.doi.org/10.1016/j.jcp.2023.112588.

[21] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, Jan. 2019. ISBN: 9781108422093. DOI: 10.1017/9781108380690. URL: http://dx.doi.org/10.1017/9781108380690.

[22] M. Buehner and P. Young. "A tighter bound for the echo state property". In: *IEEE Transactions on Neural Networks* 17.3 (May 2006), pp. 820–824. ISSN: 1941-0093. DOI: 10.1109/tnn.2006.872357. URL: http://dx.doi.org/10.1109/TNN.2006.872357.

[23] Annalisa Buffa et al. "A prioriconvergence of the Greedy algorithm for the parametrized reduced basis method". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 46.3 (Jan. 2012), pp. 595–603. ISSN: 1290-3841. DOI: 10.1051/m2an/2011056. URL: http://dx.doi.org/10.1051/m2an/2011056.

[24] Shubhaditya Burela, Philipp Krah, and Julius Reiss. "Parametric model order reduction for a wildland fire model via the shifted POD based deep learning method". In: *arXiv preprint* (Apr. 2023). DOI: 10.48550/arxiv.2304.14872. URL: https://doi.org/10.48550/arxiv.2304.14872.

[25] S. Busto et al. "POD–Galerkin reduced order methods for combined Navier–Stokes transport equations based on a hybrid FV-FE solver". In: *Computers &; Mathematics with Applications* 79.2 (Jan. 2020), pp. 256–273. ISSN: 0898-1221. DOI: 10.1016/j.camwa.2019.06.026. URL: http://dx.doi.org/10.1016/j.camwa.2019.06.026.

[26] N Cagniart et al. "Model Order Reduction for Hyperbolic Problems: a new framework". In: *HAL archive* (2017). URL: https://hal.science/hal-01583224.

[27] Daniel Canaday, Aaron Griffith, and Daniel J. Gauthier. "Rapid time series prediction with a hardware-based reservoir computer". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.12 (Dec. 2018). ISSN: 1089-7682. DOI: 10.1063/1.5048199. URL: http://dx.doi.org/10.1063/1.5048199.

[28] Kevin Carlberg. "Adaptive h-refinement for reduced-order models". In: *International Journal for Numerical Methods in Engineering* 102.5 (Nov. 2014), pp. 1192–1210. ISSN: 1097-0207. DOI: 10.1002/nme.4800. URL: http://dx.doi.org/10.1002/nme.4800.

[29] T. L. Carroll and L. M. Pecora. "Network structure effects in reservoir computers". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.8 (Aug. 2019). ISSN: 1089-7682. DOI: 10.1063/1.5097686. URL: http://dx.doi.org/10.1063/1.5097686.

[30] Ashesh Chattopadhyay, Pedram Hassanzadeh, and Devika Subramanian. "Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network". In: *Nonlinear Processes in Geophysics* 27.3 (July 2020), pp. 373–389. ISSN: 1607-7946. DOI: 10.5194/npg-27-373-2020. URL: http://dx.doi.org/10.5194/npg-27-373-2020.

[31] Kinnor Chattopadhyay, Mihaiela Isac, and Roderick I. L. Guthrie. "Physical and Mathematical Modelling of Steelmaking Tundish Operations: A Review of the Last Decade (1999–2009)". In: *ISIJ International* 50.3 (2010), pp. 331–348. ISSN: 1347-5460. DOI: 10.2355/isijinternational.50.331. URL: http://dx.doi.org/10.2355/isijinternational.50.331.

[32] S Chelbi and A Mekhmoukh. "Features based image registration using cross correlation and Radon transform". In: *Alexandria engineering journal* 57.4 (2018), pp. 2313–2318.

[33] Chao Chen et al. "Effects of Salt Tracer Amount, Concentration and Kind on the Fluid Flow Behavior in a Hydrodynamic Model of Continuous Casting Tundish". In: *steel research international* 83.12 (July 2012), pp. 1141–1151. ISSN: 1869-344X. DOI: 10.1002/srin.201200086. URL: http://dx.doi.org/10.1002/srin.201200086.

[34] Francisco Chinesta et al. "Model Reduction Methods". In: *Encyclopedia of Computational Mechanics Second Edition* (Dec. 2017), pp. 1–36. DOI: 10.1002/9781119176817.ecm2110. URL: http://dx.doi.org/10.1002/9781119176817.ecm2110.

[35] Myung Jong Cho and In Cheol Kim. "Simple Tundish Mixing Model of Continuous Casting during a Grade Transition". In: *ISIJ International* 46.10 (2006), pp. 1416–1420. ISSN: 1347-5460. DOI: 10.2355/isijinternational.46.1416. URL: http://dx.doi.org/10.2355/isijinternational.46.1416.

[36] Myung Jong Cho and Sang Joon Kim. "A Practical Model for Predicting Intermixed Zone During Grade Transition". In: *ISIJ International* 50.8 (2010), pp. 1175–1179. ISSN: 1347-5460. DOI: 10.2355/isijinternational.50.1175. URL: http://dx.doi.org/10.2355/isijinternational.50.1175.

[37] Edward Choi et al. "Using recurrent neural network models for early detection of heart failure onset". In: *Journal of the American Medical Informatics Association* 24.2 (Aug. 2016), pp. 361–370. ISSN: 1527-974X. DOI: 10.1093/jamia/ocw112. URL: http://dx.doi.org/10.1093/jamia/ocw112.

[38] R. Courant, K. Friedrichs, and H. Lewy. "Über die partiellen Differenzengleichungen der mathematischen Physik". In: *Mathematische Annalen* 100.1 (Dec. 1928), pp. 32–74. ISSN: 1432-1807, English translation, with commentaries by Lax, P.B., Widlund, O.B., Parter, S.V., in IBM J. Res.

Develop. 11 (1967). DOI: 10.1007/bf01448839. URL: http://dx.doi.org/10.1007/BF01448839.

[39] J. Crank and P. Nicolson. "A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 43.1 (Jan. 1947), pp. 50–67. ISSN: 1469-8064. DOI: 10.1017/s0305004100023197. URL: http://dx.doi.org/10.1017/S0305004100023197.

[40] R. Crisovan et al. "Model order reduction for parametrized nonlinear hyperbolic problems as an application to uncertainty quantification". In: *Journal of Computational and Applied Mathematics* 348 (Mar. 2019), pp. 466–489. ISSN: 0377-0427. DOI: 10.1016/j.cam.2018.09.018. URL: http://dx.doi.org/10.1016/j.cam.2018.09.018.

[41] A. Cwudziński. "Influence of Hydrodynamic Structure on Mixing Time of Alloy Additions with Liquid Steel in One Strand Tundish". In: *Archives of Metallurgy and Materials* 61.1 (Mar. 2016), pp. 295–300. ISSN: 2300-1909. DOI: 10.1515/amm-2016-0056. URL: http://dx.doi.org/10.1515/amm-2016-0056.

[42] Adam Cwudziński. "Physical and mathematical simulation of liquid steel mixing zone in one strand continuous casting tundish". In: *International Journal of Cast Metals Research* 30.1 (Sept. 2016), pp. 50–60. ISSN: 1743-1336. DOI: 10.1080/13640461.2016.1234223. URL: http://dx.doi.org/10.1080/13640461.2016.1234223.

[43] Chandrashekhar Damle and Yogeshwar Sahai. "The Effect of Tracer Density on Melt Flow Characterization in Continuous Casting Tundishes. A Modeling Study." In: *ISIJ International* 35.2 (1995), pp. 163–169. ISSN: 0915-1559. DOI: 10.2355/isijinternational.35.163. URL: http://dx.doi.org/10.2355/isijinternational.35.163.

[44] Thomas Daniel et al. "Model order reduction assisted by deep neural networks (ROM-net)". In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1 (Apr. 2020). ISSN: 2213-7467. DOI: 10.1186/s40323-020-00153-6. URL: http://dx.doi.org/10.1186/s40323-020-00153-6.

[45] Daniel J. Gauthier. *Reservoir Computing: Harnessing a Universal Dynamical System, SIAM News*. 2023. URL: https://www.siam.org/publications/siam-news/articles/reservoir-computing-harnessing-a-universal-dynamical-system.

[46] M.S. Darwish and F. Moukalled. "TVD schemes for unstructured grids". In: *International Journal of Heat and Mass Transfer* 46.4 (Feb. 2003), pp. 599–

611. ISSN: 0017-9310. DOI: 10.1016/s0017-9310(02)00330-7. URL: http://dx.doi.org/10.1016/S0017-9310(02)00330-7.

[47] Fatemehzahra Darzi and Thomas Bocklitz. "A Review of Medical Image Registration for Different Modalities". In: *Bioengineering* 11.8 (Aug. 2024), p. 786. ISSN: 2306-5354. DOI: 10.3390/bioengineering11080786. URL: http://dx.doi.org/10.3390/bioengineering11080786.

[48] Ghosh Debojyoti, Loffeld John, and Youngdae Kim. *HyPar 1.0 - Finite-Difference Hyperbolic-Parabolic PDE Solver on Cartesian Grids*. https://github.com/debog/hypar. 2015. URL: https://hypar.github.io/index.html.

[49] I. Demirdžić. "On the Discretization of the Diffusion Term in Finite-Volume Continuum Mechanics". In: *Numerical Heat Transfer, Part B: Fundamentals* 68.1 (Apr. 2015), pp. 1–10. ISSN: 1521-0626. DOI: 10.1080/10407790.2014.985992. URL: http://dx.doi.org/10.1080/10407790.2014.985992.

[50] Nicola Demo, Marco Tezzele, and Gianluigi Rozza. *EZyRB: Easy Reduced Basis method*. "https://github.com/mathLab/EZyRB". Apr. 2018. DOI: 10.21105/joss.00661. URL: http://dx.doi.org/10.21105/joss.00661.

[51] Suraj S Deshpande, Lakshman Anumolu, and Mario F Trujillo. "Evaluating the performance of the two-phase flow solver interFoam". In: *Computational Science & Discovery* 5.1 (Nov. 2012), p. 014016. ISSN: 1749-4699. DOI: 10.1088/1749-4699/5/1/014016. URL: http://dx.doi.org/10.1088/1749-4699/5/1/014016.

[52] Soumitra Kumar Dinda et al. "Continuous Casting Tundish Dead Volume Study by Physical Modeling and Computational Investigation". In: *steel research international* 95.11 (Sept. 2024). ISSN: 1869-344X. DOI: 10.1002/srin.202400125. URL: http://dx.doi.org/10.1002/srin.202400125.

[53] Carl Eckart and Gale Young. "The approximation of one matrix by another of lower rank". In: *Psychometrika* 1.3 (Sept. 1936), pp. 211–218. ISSN: 1860-0980. DOI: 10.1007/bf02288367. URL: http://dx.doi.org/10.1007/BF02288367.

[54] Joel H. Ferziger, Milovan Perić, and Robert L. Street. *Computational Methods for Fluid Dynamics*. Springer International Publishing, 2020. ISBN: 9783319996936. DOI: 10.1007/978-3-319-99693-6. URL: http://dx.doi.org/10.1007/978-3-319-99693-6.

[55] Stefania Fresca, Luca Dede', and Andrea Manzoni. "A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs". In: *Journal of Scientific Computing* 87.2 (Apr. 2021). ISSN: 1573-7691. DOI: 10.1007/s10915-021-01462-7. URL: http://dx.doi.org/10.1007/s10915-021-01462-7.

[56] Ken-ichi Funahashi and Yuichi Nakamura. "Approximation of dynamical systems by continuous time recurrent neural networks". In: *Neural Networks* 6.6 (Jan. 1993), pp. 801–806. ISSN: 0893-6080. DOI: 10.1016/s0893-6080(05)80125-x. URL: http://dx.doi.org/10.1016/S0893-6080(05)80125-X.

[57] A. D. Gasman et al. "Heat and Mass Transfer in Recirculating Flows." In: *Academic Press London* (Jan. 1969), pp. 60–60. ISSN: 2059-6464.

[58] Daniel J. Gauthier et al. "Next generation reservoir computing". In: *Nature Communications* 12.1 (Sept. 2021). ISSN: 2041-1723. DOI: 10.1038/s41467-021-25801-2. URL: http://dx.doi.org/10.1038/s41467-021-25801-2.

[59] Rudy Geelen and Karen Willcox. "Localized non-intrusive reduced-order modelling in the operator inference framework". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 380.2229 (June 2022). ISSN: 1471-2962. DOI: 10.1098/rsta.2021.0206. URL: http://dx.doi.org/10.1098/rsta.2021.0206.

[60] Rudy Geelen, Stephen Wright, and Karen Willcox. "Operator inference for non-intrusive model reduction with quadratic manifolds". In: *Computer Methods in Applied Mechanics and Engineering* 403 (Jan. 2023), p. 115717. ISSN: 0045-7825. DOI: 10.1016/j.cma.2022.115717. URL: http://dx.doi.org/10.1016/j.cma.2022.115717.

[61] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. "Strong Stability-Preserving High-Order Time Discretization Methods". In: *SIAM Review* 43.1 (Jan. 2001), pp. 89–112. ISSN: 1095-7200. DOI: 10.1137/s003614450036757x. URL: http://dx.doi.org/10.1137/S003614450036757X.

[62] Harshith Gowrachari, Giovanni Stabile, and Gianluigi Rozza. *Model Reduction for Transport-Dominated Problems via Cross-Correlation Based Snapshot Registration*. 2025. DOI: 10.48550/ARXIV.2501.01299. URL: https://arxiv.org/abs/2501.01299.

[63] Harshith Gowrachari et al. *Non-intrusive model reduction of advection-dominated hyperbolic problems using neural network shift augmented manifold transformations*. 2024. DOI: 10.48550/ARXIV.2407.18419. URL: https://arxiv.org/abs/2407.18419.

[64] Constantin Greif and Karsten Urban. "Decay of the Kolmogorov N-width for wave problems". In: *Applied Mathematics Letters* 96 (Oct. 2019), pp. 216–222. ISSN: 0893-9659. DOI: 10.1016/j.aml.2019.05.013. URL: http://dx.doi.org/10.1016/j.aml.2019.05.013.

[65] B. J. Grzyb et al. "Which Model to Use for the Liquid State Machine?" In: *2009 International Joint Conference on Neural Networks*. IEEE, 2009, pp. 1018–1024.

[66] Joel Guerrero. *OpenFOAM advanced training. Introduction to the FVM method. Standard practices in general CFD with applications to OpenFOAM*. 2024. DOI: 10.6084/M9.FIGSHARE.19308740.V3.

[67] Roderick I. L. Guthrie and Mihaiela M. Isac. "Continuous Casting Practices for Steel: Past, Present and Future". In: *Metals* 12.5 (May 2022), p. 862. ISSN: 2075-4701. DOI: 10.3390/met12050862. URL: http://dx.doi.org/10.3390/met12050862.

[68] Bernard Haasdonk and Mario Ohlberger. "Adaptive basis enrichment for the reduced basis method applied to finite volume schemes". In: *Proceedings of the 5th International Symposium on Finite Volumes for Complex Applications*. 2008, pp. 471–478.

[69] Bernard Haasdonk and Mario Ohlberger. "Reduced basis method for finite volume approximations of parametrized linear evolution equations". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 42.2 (Mar. 2008), pp. 277–302. ISSN: 1290-3841. DOI: 10.1051/m2an:2008001. URL: http://dx.doi.org/10.1051/m2an:2008001.

[70] Arash Hajisharifi et al. "An LSTM-enhanced surrogate model to simulate the dynamics of particle-laden fluid systems". In: *Computers & Fluids* 280 (Aug. 2024), p. 106361. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2024.106361. URL: http://dx.doi.org/10.1016/j.compfluid.2024.106361.

[71] Kenneth C. Hall, Jeffrey P. Thomas, and Earl H. Dowell. "Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows". In: *AIAA Journal* 38.10 (Oct. 2000), pp. 1853–1862. ISSN: 1533-385X. DOI: 10.2514/2.867. URL: http://dx.doi.org/10.2514/2.867.

[72] Dongxu Han, Bo Yu, and Xinyu Zhang. "Study on a BFC-Based POD-Galerkin Reduced-Order Model for the Unsteady-State Variable-Property Heat Transfer Problem". In: *Numerical Heat Transfer, Part B: Fundamentals* 65.3 (Feb. 2014), pp. 256–281. ISSN: 1521-0626. DOI: 10.1080/10407790.

2013.849989. URL: http://dx.doi.org/10.1080/10407790.2013.849989.

[73] M.N. Al-Harbi et al. "Mathematical modeling approach toward the optimization of the tundish mix grades process". In: *Proceedings of the 5th International Conference on ThermoMechanical Processing*. Milan, Italy, Oct. 26–28, 2016.

[74] David Hartman and Lalit K. Mestha. "A deep learning framework for model reduction of dynamical systems". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, Aug. 2017. DOI: 10.1109/ccta.2017.8062736. URL: http://dx.doi.org/10.1109/CCTA.2017.8062736.

[75] J.S. Hesthaven and S. Ubbiali. "Non-intrusive reduced order modeling of nonlinear problems using neural networks". In: *Journal of Computational Physics* 363 (June 2018), pp. 55–78. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.02.037. URL: http://dx.doi.org/10.1016/j.jcp.2018.02.037.

[76] Jan S Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer International Publishing, 2016. ISBN: 9783319224701. DOI: 10.1007/978-3-319-22470-1. URL: http://dx.doi.org/10.1007/978-3-319-22470-1.

[77] Chi Hoang et al. "Projection-based model reduction of dynamical systems using space–time subspace and machine learning". In: *Computer Methods in Applied Mechanics and Engineering* 389 (Feb. 2022), p. 114341. ISSN: 0045-7825. DOI: 10.1016/j.cma.2021.114341. URL: http://dx.doi.org/10.1016/j.cma.2021.114341.

[78] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 1530-888X. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[79] Jin-Quan Huang and F.L. Lewis. "Neural-network predictive control for nonlinear dynamic systems with time-delay". In: *IEEE Transactions on Neural Networks* 14.2 (Mar. 2003), pp. 377–389. ISSN: 1045-9227. DOI: 10.1109/tnn.2003.809424. URL: http://dx.doi.org/10.1109/TNN.2003.809424.

[80] Xiaoqing Huang and Brian G. Thomas. "Intermixing model of continuous casting during a grade transition". In: *Metallurgical and Materials Transactions B* 27.4 (Aug. 1996), pp. 617–632. ISSN: 1543-1916. DOI: 10.1007/bf02915660. URL: http://dx.doi.org/10.1007/BF02915660.

[81] Angelo Iollo and Damiano Lombardi. "Advection modes by optimal mass transfer". In: *Physical Review E* 89.2 (Feb. 2014). ISSN: 1550-2376. DOI: 10.1103/physreve.89.022923. URL: http://dx.doi.org/10.1103/PhysRevE.89.022923.

[82] Angelo Iollo and Tommaso Taddei. "Mapping of coherent structures in parameterized flows by learning optimal transportation with Gaussian models". In: *Journal of Computational Physics* 471 (Dec. 2022), p. 111671. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2022.111671. URL: http://dx.doi.org/10.1016/j.jcp.2022.111671.

[83] W. R. Irving. *Continuous Casting of Steel*. The Institute of Materials(UK), 1993.

[84] Opal Issan and Boris Kramer. "Predicting solar wind streams from the inner-heliosphere to Earth via shifted operator inference". In: *Journal of Computational Physics* 473 (Jan. 2023), p. 111689. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2022.111689. URL: http://dx.doi.org/10.1016/j.jcp.2022.111689.

[85] Yuto Iwasaki et al. "Reservoir computing reduced-order model based on particle image velocimetry data of post-stall flow". In: *AIP Advances* 13.6 (June 2023). ISSN: 2158-3226. DOI: 10.1063/5.0150947. URL: http://dx.doi.org/10.1063/5.0150947.

[86] H. Jaeger. *The "Echo State" Approach to Analysing and Training Recurrent Neural Networks - With an Erratum Note*. GMD Technical Report 148. Bonn, Germany: German National Research Centre for Information Technology, 2001, p. 13.

[87] Herbert Jaeger and Harald Haas. "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication". In: *Science* 304.5667 (Apr. 2004), pp. 78–80. ISSN: 1095-9203. DOI: 10.1126/science.1091277. URL: http://dx.doi.org/10.1126/science.1091277.

[88] Shobhit Jain et al. "A quadratic manifold for model order reduction of nonlinear structural dynamics". In: *Computers & Structures* 188 (Aug. 2017), pp. 80–94. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2017.04.005. URL: http://dx.doi.org/10.1016/j.compstruc.2017.04.005.

[89] Hrvoje Jasak. "Error Analysis and Estimation for the Finite Volume Method With Applications to Fluid Flows". PhD thesis. Imperial College, London University, London, UK, 1996.

[90] Hrvoje Jasak. "OpenFOAM: Open source CFD in research and industry". In: *International Journal of Naval Architecture and Ocean Engineering* 1.2 (Dec. 2009), pp. 89–94. ISSN: 2092-6782. DOI: 10.2478/ijnaoe-2013-0011. URL: http://dx.doi.org/10.2478/IJNAOE-2013-0011.

[91] Pradeep K. Jha et al. "Mixing in a tundish and a choice of turbulence model for its prediction". In: *International Journal of Numerical Methods for Heat & Fluid Flow* 13.8 (Dec. 2003), pp. 964–996. ISSN: 0961-5539. DOI: 10.1108/09615530310501920. URL: http://dx.doi.org/10.1108/09615530310501920.

[92] Junjie Jiang and Ying-Cheng Lai. "Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius". In: *Physical Review Research* 1.3 (Oct. 2019). ISSN: 2643-1564. DOI: 10.1103/physrevresearch.1.033056. URL: http://dx.doi.org/10.1103/PhysRevResearch.1.033056.

[93] Kenji Kashima. "Nonlinear model reduction by deep autoencoder of noise response data". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, Dec. 2016. DOI: 10.1109/cdc.2016.7799153. URL: http://dx.doi.org/10.1109/CDC.2016.7799153.

[94] Moaad Khamlich et al. "A physics-based reduced order model for urban air pollution prediction". In: *Computer Methods in Applied Mechanics and Engineering* 417 (Dec. 2023), p. 116416. ISSN: 0045-7825. DOI: 10.1016/j.cma.2023.116416. URL: http://dx.doi.org/10.1016/j.cma.2023.116416.

[95] Youngkyu Kim et al. "A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder". In: *Journal of Computational Physics* 451 (Feb. 2022), p. 110841. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2021.110841. URL: http://dx.doi.org/10.1016/j.jcp.2021.110841.

[96] M. Kimura and R. Nakano. "Learning dynamical systems by recurrent neural networks from orbits". In: *Neural Networks* 11.9 (Dec. 1998), pp. 1589–1599. ISSN: 0893-6080. DOI: 10.1016/s0893-6080(98)00098-7. URL: http://dx.doi.org/10.1016/S0893-6080(98)00098-7.

[97] Hendrik Kleikamp, Mario Ohlberger, and Stephan Rave. *Nonlinear Model Order Reduction using Diffeomorphic Transformations of a Space-Time Domain*. 2022. DOI: 10.48550/ARXIV.2203.05833. URL: https://arxiv.org/abs/2203.05833.

[98] Anna Kovárnová et al. "Shifted Proper Orthogonal Decomposition and Artificial Neural Networks for Time-Continuous Reduced Order Models of Transport-Dominated Systems". In: *Topical Problems of Fluid Mechanics 2022*. TPFM. Institute of Thermomechanics of the Czech Academy of Sciences, 2022. DOI: 10.14311/tpfm.2022.016. URL: http://dx.doi.org/10.14311/TPFM.2022.016.

[99] Philipp Krah et al. "A Robust Shifted Proper Orthogonal Decomposition: Proximal Methods for Decomposing Flows with Multiple Transports". In: *SIAM Journal on Scientific Computing* 47.2 (Mar. 2025), A633–A656. ISSN: 1095-7197. DOI: 10.1137/24m164392x. URL: http://dx.doi.org/10.1137/24M164392X.

[100] Krashnavtar and D. Mazumdar. "Transient, Multiphase Simulation of Grade Intermixing in a Tundish Under Constant Casting Rate and Validation Against Physical Modeling". In: *JOM* 70.10 (July 2018), pp. 2139–2147. ISSN: 1543-1851. DOI: 10.1007/s11837-018-2988-8. URL: http://dx.doi.org/10.1007/s11837-018-2988-8.

[101] Dongyang Kuang and Tanya Schmah. "FAIM – A ConvNet Method for Unsupervised 3D Medical Image Registration". In: *Machine Learning in Medical Imaging*. Springer International Publishing, 2019, pp. 646–654. ISBN: 9783030326920. DOI: 10.1007/978-3-030-32692-0_74. URL: http://dx.doi.org/10.1007/978-3-030-32692-0_74.

[102] J. Nathan Kutz. "Deep learning in fluid dynamics". In: *Journal of Fluid Mechanics* 814 (Jan. 2017), pp. 1–4. ISSN: 1469-7645. DOI: 10.1017/jfm.2016.803. URL: http://dx.doi.org/10.1017/jfm.2016.803.

[103] Martin Längkvist, Lars Karlsson, and Amy Loutfi. "A review of unsupervised feature learning and deep learning for time-series modeling". In: *Pattern Recognition Letters* 42 (June 2014), pp. 11–24. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2014.01.008. URL: http://dx.doi.org/10.1016/j.patrec.2014.01.008.

[104] B. E. Launder and D. B. Spalding. *Lectures in Mathematical Models of Turbulence*. London and New York: Academic Press, 1972. ISBN: 0124380506.

[105] B.E. Launder and D.B. Spalding. "Mathematical models of turbulence". In: *Academic Press, Massachusetts* (1972).

[106] B.E. Launder and D.B. Spalding. "The numerical computation of turbulent flows". In: *Computer Methods in Applied Mechanics and Engineering* 3.2 (Mar. 1974), pp. 269–289. ISSN: 0045-7825. DOI: 10.1016/0045-7825(74)90029-2. URL: http://dx.doi.org/10.1016/0045-7825(74)90029-2.

[107] Kookjin Lee and Kevin T. Carlberg. "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders". In: *Journal of Computational Physics* 404 (Mar. 2020), p. 108973. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.108973. URL: http://dx.doi.org/10.1016/j.jcp.2019.108973.

[108] Huanrong Li, Zhendong Luo, and Jing Chen. "Numerical simulation based on POD for two-dimensional solute transport problems". In: *Applied Mathematical Modelling* 35.5 (May 2011), pp. 2489–2498. ISSN: 0307-904X. DOI: 10.1016/j.apm.2010.11.064. URL: http://dx.doi.org/10.1016/j.apm.2010.11.064.

[109] Kangji Li et al. "A fast-POD model for simulation and control of indoor thermal environment of buildings". In: *Building and Environment* 60 (Feb. 2013), pp. 150–157. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2012.11.020. URL: http://dx.doi.org/10.1016/j.buildenv.2012.11.020.

[110] Qianqian Li et al. "A Robust Registration Algorithm for Image-Guided Surgical Robot". In: *IEEE Access* 6 (2018), pp. 42950–42960. ISSN: 2169-3536. DOI: 10.1109/access.2018.2853601. URL: http://dx.doi.org/10.1109/ACCESS.2018.2853601.

[111] Qianxiao Li et al. "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.10 (Oct. 2017). ISSN: 1089-7682. DOI: 10.1063/1.4993854. URL: http://dx.doi.org/10.1063/1.4993854.

[112] Junda Liu et al. "Optimization of the Liquid Steel Flow Behavior in the Tundish through Water Model Experiment, Numerical Simulation and Industrial Trial". In: *Metals* 12.9 (Sept. 2022), p. 1480. ISSN: 2075-4701. DOI: 10.3390/met12091480. URL: http://dx.doi.org/10.3390/met12091480.

[113] Wing Kam Liu, Shaofan Li, and Harold S. Park. "Eighty Years of the Finite Element Method: Birth, Evolution, and Future". In: *Archives of Computational Methods in Engineering* 29.6 (June 2022), pp. 4431–4453.

ISSN: 1886-1784. DOI: 10.1007/s11831-022-09740-9. URL: http://dx.doi.org/10.1007/s11831-022-09740-9.

[114] Stefano Lorenzi et al. "POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations". In: *Computer Methods in Applied Mechanics and Engineering* 311 (Nov. 2016), pp. 151–179. ISSN: 0045-7825. DOI: 10.1016/j.cma.2016.08.006. URL: http://dx.doi.org/10.1016/j.cma.2016.08.006.

[115] Zhixin Lu, Brian R. Hunt, and Edward Ott. "Attractor reconstruction by machine learning". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.6 (June 2018). ISSN: 1089-7682. DOI: 10.1063/1.5039508. URL: http://dx.doi.org/10.1063/1.5039508.

[116] Mantas Lukoševičius. "A Practical Guide to Applying Echo State Networks". In: *Neural Networks: Tricks of the Trade*. Springer Berlin Heidelberg, 2012, pp. 659–686. ISBN: 9783642352898. DOI: 10.1007/978-3-642-35289-8_36. URL: http://dx.doi.org/10.1007/978-3-642-35289-8_36.

[117] Mantas Lukoševičius and Herbert Jaeger. "Reservoir computing approaches to recurrent neural network training". In: *Computer Science Review* 3.3 (Aug. 2009), pp. 127–149. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2009.03.005. URL: http://dx.doi.org/10.1016/j.cosrev.2009.03.005.

[118] Wolfgang Maass, Thomas Natschläger, and Henry Markram. "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations". In: *Neural Computation* 14.11 (Nov. 2002), pp. 2531–2560. ISSN: 1530-888X. DOI: 10.1162/089976602760407955. URL: http://dx.doi.org/10.1162/089976602760407955.

[119] Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. "Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders". In: *Physics of Fluids* 33.3 (Mar. 2021). ISSN: 1089-7666. DOI: 10.1063/5.0039986. URL: http://dx.doi.org/10.1063/5.0039986.

[120] Dipak Mazumdar. "Review, Analysis, and Modeling of Continuous Casting Tundish Systems". In: *steel research international* 90.4 (Sept. 2018). ISSN: 1869-344X. DOI: 10.1002/srin.201800279. URL: http://dx.doi.org/10.1002/srin.201800279.

[121] Dipak Mazumdar. "Review, analysis, and modeling of continuous casting tundish systems". In: *steel research international* 90.4 (2019), p. 1800279.

[122] Dipak Mazumdar and Roderick I. L. Guthrie. "The Physical and Mathematical Modelling of Continuous Casting Tundish System." In: *ISIJ International* 39.6 (1999), pp. 524–547. ISSN: 0915-1559. DOI: 10.2355/isijinternational.39.524. URL: http://dx.doi.org/10.2355/isijinternational.39.524.

[123] Wolfshtien Micha. "Turbulent Convection in Impinging Jets". PhD thesis. Imperial College, London University, London, UK, 1968.

[124] Rambod Mojgani and Maciej Balajewicz. "Arbitrary Lagrangian Eulerian framework for efficient projection-based reduction of convection dominated nonlinear flows". In: *APS Division of Fluid Dynamics Meeting Abstracts*. 2017, pp. M1–008.

[125] Tony C. W. Mok and Albert C. S. Chung. "Large Deformation Diffeomorphic Image Registration with Laplacian Pyramid Networks". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Springer International Publishing, 2020, pp. 211–221. ISBN: 9783030597160. DOI: 10.1007/978-3-030-59716-0_21. URL: http://dx.doi.org/10.1007/978-3-030-59716-0_21.

[126] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*. Springer International Publishing, 2016. ISBN: 9783319168746. DOI: 10.1007/978-3-319-16874-6. URL: http://dx.doi.org/10.1007/978-3-319-16874-6.

[127] Florian Müller, Patrick Jenny, and Daniel W. Meyer. "Probabilistic collocation and lagrangian sampling for advective tracer transport in randomly heterogeneous porous media". In: *Advances in Water Resources* 34.12 (Dec. 2011), pp. 1527–1538. ISSN: 0309-1708. DOI: 10.1016/j.advwatres.2011.09.005. URL: http://dx.doi.org/10.1016/j.advwatres.2011.09.005.

[128] A. Muralikrishna, S. Bagui, and Dipak Mazumdar. "Modeling and Measurements of Intermixing Time in a Water Model of a Four Strand Steelmaking Tundish System". In: *Transactions of the Indian Institute of Metals* 66.3 (Mar. 2013), pp. 281–295. ISSN: 0975-1645. DOI: 10.1007/s12666-013-0270-x. URL: http://dx.doi.org/10.1007/s12666-013-0270-x.

[129] K.S. Narendra and K. Parthasarathy. "Identification and control of dynamical systems using neural networks". In: *IEEE Transactions on Neural Networks* 1.1 (Mar. 1990), pp. 4–27. ISSN: 1045-9227. DOI: 10.1109/72.80202. URL: http://dx.doi.org/10.1109/72.80202.

[130] Monica Nonino and Davide Torlo. "Calibration-Based ALE Model Order Reduction for Hyperbolic Problems with Self-Similar Travelling Discontinuities". In: *Journal of Scientific Computing* 101.3 (Oct. 2024). ISSN: 1573-7691. DOI: 10.1007/s10915-024-02694-z. URL: http://dx.doi.org/10.1007/s10915-024-02694-z.

[131] Monica Nonino et al. "A reduced basis method by means of transport maps for a fluid–structure interaction problem with slowly decaying Kolmogorov *n*-width". In: *Advances in Computational Science and Engineering* 1.1 (2023), pp. 36–58. ISSN: 2837-1739. DOI: 10.3934/acse.2023002. URL: http://dx.doi.org/10.3934/acse.2023002.

[132] J. Tinsley Oden and L. Carter Wellford. "Analysis of flow of viscous fluids by the finite-element method." In: *AIAA Journal* 10.12 (Dec. 1972), pp. 1590–1599. ISSN: 1533-385X. DOI: 10.2514/3.6691. URL: http://dx.doi.org/10.2514/3.6691.

[133] Hans-Jürgen Odenthal, Ralf Böiling, and Herbert Pfeifer. "Numerical and Physical Simulation of Tundish Fluid Flow Phenomena". In: *steel research international* 74.1 (Jan. 2003), pp. 44–55. ISSN: 1611-3683. DOI: 10.1002/srin.200300160. URL: http://dx.doi.org/10.1002/srin.200300160.

[134] MARIO OHLBERGER and STEPHAN RAVE. "REDUCED BASIS METHODS: SUCCESS, LIMITATIONS AND FUTURE CHALLENGES". In: *Proceedings of ALGORITMY*. 2016, pp. 1–12.

[135] Mario Ohlberger and Stephan Rave. "Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing". In: *Comptes Rendus. Mathématique* 351.23–24 (Nov. 2013), pp. 901–906. ISSN: 1778-3569. DOI: 10.1016/j.crma.2013.10.028. URL: http://dx.doi.org/10.1016/j.crma.2013.10.028.

[136] Davide Papapicco et al. "The Neural Network shifted-proper orthogonal decomposition: A machine learning approach for non-linear reduction of hyperbolic equations". In: *Computer Methods in Applied Mechanics and Engineering* 392 (Mar. 2022), p. 114687. ISSN: 0045-7825. DOI: 10.1016/j.cma.2022.114687. URL: http://dx.doi.org/10.1016/j.cma.2022.114687.

[137] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Proceedings of Machine Learning Research. Atlanta, Georgia, USA: PMLR, June 2013, pp. 1310–1318.

[138] Damiano Pasetto, Alberto Guadagnini, and Mario Putti. "POD-based Monte Carlo approach for the solution of regional scale groundwater flow driven by randomly distributed recharge". In: *Advances in Water Resources* 34.11 (Nov. 2011), pp. 1450–1463. ISSN: 0309-1708. DOI: 10.1016/j.advwatres.2011.07.003. URL: http://dx.doi.org/10.1016/j.advwatres.2011.07.003.

[139] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *NIPS Workshop on Automatic differentiation*. 2017.

[140] Jaideep Pathak et al. "Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach". In: *Physical Review Letters* 120.2 (Jan. 2018). ISSN: 1079-7114. DOI: 10.1103/physrevlett.120.024102. URL: http://dx.doi.org/10.1103/PhysRevLett.120.024102.

[141] Jaideep Pathak et al. "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.12 (Dec. 2017). ISSN: 1089-7682. DOI: 10.1063/1.5010300. URL: http://dx.doi.org/10.1063/1.5010300.

[142] Benjamin Peherstorfer. "Breaking the Kolmogorov Barrier with Nonlinear Model Reduction". In: *Notices of the American Mathematical Society* 69.05 (May 2022), p. 1. ISSN: 1088-9477. DOI: 10.1090/noti2475. URL: http://dx.doi.org/10.1090/noti2475.

[143] Benjamin Peherstorfer. "Model Reduction for Transport-Dominated Problems via Online Adaptive Bases and Adaptive Sampling". In: *SIAM Journal on Scientific Computing* 42.5 (Jan. 2020), A2803–A2836. ISSN: 1095-7197. DOI: 10.1137/19m1257275. URL: http://dx.doi.org/10.1137/19M1257275.

[144] Benjamin Peherstorfer and Karen Willcox. "Data-driven operator inference for nonintrusive projection-based model reduction". In: *Computer Methods in Applied Mechanics and Engineering* 306 (July 2016), pp. 196–215. ISSN: 0045-7825. DOI: 10.1016/j.cma.2016.03.025. URL: http://dx.doi.org/10.1016/j.cma.2016.03.025.

[145] Benjamin Peherstorfer and Karen Willcox. "Online Adaptive Model Reduction for Nonlinear Systems via Low-Rank Updates". In: *SIAM Journal on Scientific Computing* 37.4 (Jan. 2015), A2123–A2150. ISSN: 1095-7197. DOI: 10.1137/140989169. URL: http://dx.doi.org/10.1137/140989169.

[146] Benjamin Peherstorfer et al. "Localized Discrete Empirical Interpolation Method". In: *SIAM Journal on Scientific Computing* 36.1 (Jan. 2014), A168–A192. ISSN: 1095-7197. DOI: 10.1137/130924408. URL: http://dx.doi.org/10.1137/130924408.

[147] Federico Pichi et al. "An artificial neural network approach to bifurcating phenomena in computational fluid dynamics". In: *Computers & Fluids* 254 (Mar. 2023), p. 105813. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2023.105813. URL: http://dx.doi.org/10.1016/j.compfluid.2023.105813.

[148] J Pieprzyca. "Cast strand's structure quality research in respect to method of the liquid steel flow through cc device's tundish". In: *Metalurgija* 52.2 (2013), pp. 157–160.

[149] Allan Pinkus. *n-Widths in Approximation Theory*. Springer Berlin Heidelberg, 1985. ISBN: 9783642698941. DOI: 10.1007/978-3-642-69894-1. URL: http://dx.doi.org/10.1007/978-3-642-69894-1.

[150] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced Basis Methods for Partial Differential Equations*. Springer International Publishing, 2016. ISBN: 9783319154312. DOI: 10.1007/978-3-319-15431-2. URL: http://dx.doi.org/10.1007/978-3-319-15431-2.

[151] K. Raghavendra et al. "Mathematical modelling of single and multi-strand tundish for inclusion analysis". In: *Applied Mathematical Modelling* 37.9 (May 2013), pp. 6284–6300. ISSN: 0307-904X. DOI: 10.1016/j.apm.2013.01.013. URL: http://dx.doi.org/10.1016/j.apm.2013.01.013.

[152] Arun Prakash Raghupathy et al. "Boundary-condition-independent reduced-order modeling of complex 2D objects by POD-Galerkin methodology". In: *2009 25th Annual IEEE Semiconductor Thermal Measurement and Management Symposium*. IEEE, 2009, pp. 208–215. DOI: 10.1109/stherm.2009.4810765. URL: http://dx.doi.org/10.1109/STHERM.2009.4810765.

[153] M. Raissi, P. Perdikaris, and G.E. Karniadakis. "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 (Feb. 2019), pp. 686–707. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.045. URL: http://dx.doi.org/10.1016/j.jcp.2018.10.045.

[154] Akhter Mohiuddin Rather, Arun Agarwal, and V.N. Sastry. "Recurrent neural network and a hybrid model for prediction of stock returns". In:

Expert Systems with Applications 42.6 (Apr. 2015), pp. 3234–3241. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2014.12.003. URL: http://dx.doi.org/10.1016/j.eswa.2014.12.003.

[155] S. S. Ravindran. "A reduced-order approach for optimal control of fluids using proper orthogonal decomposition". In: *International Journal for Numerical Methods in Fluids* 34.5 (2000), pp. 425–448. ISSN: 1097-0363. DOI: 10.1002/1097-0363(20001115)34:5<425::aid-fld67>3.0.co;2-w. URL: http://dx.doi.org/10.1002/1097-0363(20001115)34:5%3C425::AID-FLD67%3E3.0.CO;2-W.

[156] J. Reiss et al. "The Shifted Proper Orthogonal Decomposition: A Mode Decomposition for Multiple Transport Phenomena". In: *SIAM Journal on Scientific Computing* 40.3 (Jan. 2018), A1322–A1344. ISSN: 1095-7197. DOI: 10.1137/17m1140571. URL: http://dx.doi.org/10.1137/17M1140571.

[157] Ying Ren, Wen Yang, and Lifeng Zhang. "Deformation of Non-metallic Inclusions in Steel during Rolling Process: A Review". In: *ISIJ International* 62.11 (Nov. 2022), pp. 2159–2171. ISSN: 1347-5460. DOI: 10.2355/isijinternational.isijint-2022-235. URL: http://dx.doi.org/10.2355/isijinternational.ISIJINT-2022-235.

[158] Donsub Rim, Scott Moe, and Randall J. LeVeque. "Transport Reversal for Model Reduction of Hyperbolic Partial Differential Equations". In: *SIAM/ASA Journal on Uncertainty Quantification* 6.1 (Jan. 2018), pp. 118–150. ISSN: 2166-2525. DOI: 10.1137/17m1113679. URL: http://dx.doi.org/10.1137/17M1113679.

[159] Francesco Romor, Giovanni Stabile, and Gianluigi Rozza. "Non-linear Manifold Reduced-Order Models with Convolutional Autoencoders and Reduced Over-Collocation Method". In: *Journal of Scientific Computing* 94.3 (Feb. 2023). ISSN: 1573-7691. DOI: 10.1007/s10915-023-02128-2. URL: http://dx.doi.org/10.1007/s10915-023-02128-2.

[160] Gianluigi Rozza, Giovanni Stabile, and Francesco Ballarin. *Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics*. Society for Industrial and Applied Mathematics, Jan. 2022. ISBN: 9781611977257. DOI: 10.1137/1.9781611977257. URL: http://dx.doi.org/10.1137/1.9781611977257.

[161] Antje Rückert et al. "Particle Distribution and Separation in Continuous Casting Tundish". In: *Steel Research International* 80.8 (2009), pp. 568–574. DOI: 10.2374/SRI09SP043.

[162] Akshai K. Runchal. "Brian Spalding: CFD & Reality: A Personal Recollection". In: *Journal of the Franklin Institute* 351.1 (Jan. 2014), pp. 65–87. ISSN: 0016-0032. DOI: 10.1016/j.jfranklin.2012.12.004. URL: http://dx.doi.org/10.1016/j.jfranklin.2012.12.004.

[163] Akshai K. Runchal. "Emergence of Computational Fluid Dynamics at Imperial College (1965–1975): A Personal Recollection". In: *Journal of Heat Transfer* 135.1 (Dec. 2012). ISSN: 1528-8943. DOI: 10.1115/1.4007655. URL: http://dx.doi.org/10.1115/1.4007655.

[164] Akshai K. Runchal. "Transport Processes in Steady Two-Dimensional Separated Flows". PhD thesis. Imperial College, London University, London, UK, 1969.

[165] J.B. Rutzmoser et al. "Generalization of quadratic manifolds for reduced order modeling of nonlinear structural dynamics". In: *Computers &; Structures* 192 (Nov. 2017), pp. 196–209. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2017.06.003. URL: http://dx.doi.org/10.1016/j.compstruc.2017.06.003.

[166] Yogeshwar Sahai and R. Ahuja. "Steel Flow and Mixing of Melt in Steelmaking Tundishes". In: *Ironmaking & Steelmaking* 13.5 (1986), pp. 241–247.

[167] Yogeshwar Sahai and Toshihiko Emi. "Melt Flow Characterization in Continuous Casting Tundishes." In: *ISIJ International* 36.6 (1996), pp. 667–672. ISSN: 0915-1559. DOI: 10.2355/isijinternational.36.667. URL: http://dx.doi.org/10.2355/isijinternational.36.667.

[168] Yogeshwar Sahai and Toshihiko Emi. *Tundish Technology for Clean Steel Production*. WORLD SCIENTIFIC, Dec. 2007. ISBN: 9789812790767. DOI: 10.1142/6426. URL: http://dx.doi.org/10.1142/6426.

[169] Nirav Vasant Shah et al. "Finite element based Model Order Reduction for parametrized one-way coupled steady state linear thermo-mechanical problems". In: *Finite Elements in Analysis and Design* 212 (Dec. 2022), p. 103837. ISSN: 0168-874X. DOI: 10.1016/j.finel.2022.103837. URL: http://dx.doi.org/10.1016/j.finel.2022.103837.

[170] Chi-Wang Shu. "Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws". In: *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*. Springer Berlin Heidelberg, 1998, pp. 325–432. ISBN: 9783540498049. DOI: 10.1007/bfb0096355. URL: http://dx.doi.org/10.1007/BFb0096355.

[171] Md Irfanul Haque Siddiqui and Pradeep Kumar Jha. "Assessment of Turbulence Models for Prediction of Intermixed Amount with Free Surface Variation Using Coupled Level-Set Volume of Fluid Method". In: *ISIJ International* 54.11 (2014), pp. 2578–2587. ISSN: 1347-5460. DOI: 10.2355/isijinternational.54.2578. URL: http://dx.doi.org/10.2355/isijinternational.54.2578.

[172] Md Irfanul Haque Siddiqui and Pradeep Kumar Jha. "Effect of Inflow Rate Variation on Intermixing in a Steelmaking Tundish during Ladle Change-Over". In: *steel research international* 87.6 (Sept. 2015), pp. 733–744. ISSN: 1869-344X. DOI: 10.1002/srin.201500210. URL: http://dx.doi.org/10.1002/srin.201500210.

[173] Md Irfanul Haque Siddiqui and Pradeep Kumar Jha. "Numerical Simulation of Flow-Induced Wall Shear Stresses in Three Different Shapes of Multi-Strand Steelmaking Tundishes". In: *steel research international* 86.7 (Oct. 2014), pp. 799–807. ISSN: 1869-344X. DOI: 10.1002/srin.201400217. URL: http://dx.doi.org/10.1002/srin.201400217.

[174] Md Irfanul Haque Siddiqui and Man-Hoe Kim. "Optimization of flow control devices to minimize the grade mixing in steelmaking tundish". In: *Journal of Mechanical Science and Technology* 32.7 (July 2018), pp. 3213–3221. ISSN: 1976-3824. DOI: 10.1007/s12206-018-0624-8. URL: http://dx.doi.org/10.1007/s12206-018-0624-8.

[175] Md Irfanul Haque Siddiqui and Man-Hoe Kim. "Two-Phase Numerical Modeling of Grade Intermixing in a Steelmaking Tundish". In: *Metals* 9.1 (Jan. 2019), p. 40. ISSN: 2075-4701. DOI: 10.3390/met9010040. URL: http://dx.doi.org/10.3390/met9010040.

[176] Md Irfanul Haque Siddiqui et al. "Mathematical modeling of continuous casting tundish: a review". In: *VW Appl. Sci* 3.1 (2021), pp. 92–103.

[177] Lawrence Sirovich. "Turbulence and the dynamics of coherent structures. II. Symmetries and transformations". In: *Quarterly of Applied Mathematics* 45.3 (Oct. 1987), pp. 573–582. ISSN: 1552-4485. DOI: 10.1090/qam/910463. URL: http://dx.doi.org/10.1090/qam/910463.

[178] Sokratia Georgaka Sokratia Georgaka et al. "Parametric POD-Galerkin Model Order Reduction for Unsteady-State Heat Transfer Problems". In: *Communications in Computational Physics* 27.1 (Jan. 2020), pp. 1–32. ISSN: 1815-2406. DOI: 10.4208/cicp.oa-2018-0207. URL: http://dx.doi.org/10.4208/cicp.OA-2018-0207.

[179] Seth C. Spiegel, HT Huynh, and James R. DeBonis. "A Survey of the Isentropic Euler Vortex Problem using High-Order Methods". In: *22nd*

AIAA Computational Fluid Dynamics Conference. American Institute of Aeronautics and Astronautics, June 2015. DOI: 10.2514/6.2015-2444. URL: http://dx.doi.org/10.2514/6.2015-2444.

[180] Giovanni Stabile et al. *ITHACA-FV: In real Time Highly Advanced Computational Applications for Finite Volumes - ROMs for OpenFOAM*. https://github.com/ithaca-fv/ITHACA-FV. 2025.

[181] Giovanni Stabile and Gianluigi Rozza. "Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations". In: *Computers & Fluids* 173 (Sept. 2018), pp. 273–284. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2018.01.035. URL: http://dx.doi.org/10.1016/j.compfluid.2018.01.035.

[182] Giovanni Stabile et al. "A reduced order variational multiscale approach for turbulent flows". In: *Advances in Computational Mathematics* 45.5–6 (June 2019), pp. 2349–2368. ISSN: 1572-9044. DOI: 10.1007/s10444-019-09712-x. URL: http://dx.doi.org/10.1007/s10444-019-09712-x.

[183] G. W. Stewart. "On the Early History of the Singular Value Decomposition". In: *SIAM Review* 35.4 (Dec. 1993), pp. 551–566. ISSN: 1095-7200. DOI: 10.1137/1035134. URL: http://dx.doi.org/10.1137/1035134.

[184] Patankar V. Suhas. "Heat and mass transfer in turbulent boundary layers". PhD thesis. Imperial College, London University, London, UK, 1967.

[185] T. Taddei, S. Perotto, and A. Quarteroni. "Reduced basis techniques for nonlinear conservation laws". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 49.3 (Apr. 2015), pp. 787–814. ISSN: 1290-3841. DOI: 10.1051/m2an/2014054. URL: http://dx.doi.org/10.1051/m2an/2014054.

[186] Tommaso Taddei. "A Registration Method for Model Order Reduction: Data Compression and Geometry Reduction". In: *SIAM Journal on Scientific Computing* 42.2 (Jan. 2020), A997–A1027. ISSN: 1095-7197. DOI: 10.1137/19m1271270. URL: http://dx.doi.org/10.1137/19M1271270.

[187] Gouhei Tanaka et al. "Recent advances in physical reservoir computing: A review". In: *Neural Networks* 115 (July 2019), pp. 100–123. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2019.03.005. URL: http://dx.doi.org/10.1016/j.neunet.2019.03.005.

[188] Marco Tezzele et al. "Chapter 9: Nonintrusive Data-Driven Reduced Order Models in Computational Fluid Dynamics". In: *Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics*. Society for Industrial and Applied Mathematics, Jan. 2022, pp. 203–222. ISBN:

9781611977257. DOI: 10.1137/1.9781611977257.ch9. URL: http://dx.doi.org/10.1137/1.9781611977257.ch9.

[189] B. G. Thomas. "Modeling study of intermixing in tundish and strand during a continuous-casting grade transition". English (US). In: *Iron and Steelmaker (I and SM)* 24.11 (1997), pp. 83–96. ISSN: 0275-8687.

[190] Brian G. Thomas. "Review on Modeling and Simulation of Continuous Casting". In: *steel research international* 89.1 (Dec. 2017). ISSN: 1869-344X. DOI: 10.1002/srin.201700312. URL: http://dx.doi.org/10.1002/srin.201700312.

[191] Markéta Tkadlečková et al. "Numerical Analysis of RTD Curves and Inclusions Removal in a Multi-Strand Asymmetric Tundish with Different Configuration of Impact Pad". In: *Metals* 10.7 (June 2020), p. 849. ISSN: 2075-4701. DOI: 10.3390/met10070849. URL: http://dx.doi.org/10.3390/met10070849.

[192] Davide Torlo. "Model Reduction for Advection Dominated Hyperbolic Problems in an ALE Framework: Offline and Online Phases". In: *arXiv preprint* (Mar. 2020). DOI: 10.48550/arxiv.2003.13735. URL: https://arxiv.org/abs/2003.13735.

[193] Anurag Tripathi. "Mathematical modelling of flow control in a tundish using electro-magnetic forces". In: *Applied Mathematical Modelling* 35.10 (Oct. 2011), pp. 5075–5090. ISSN: 0307-904X. DOI: 10.1016/j.apm.2011.04.028. URL: http://dx.doi.org/10.1016/j.apm.2011.04.028.

[194] Nathan Trouvain et al. "ReservoirPy: An Efficient and User-Friendly Library to Design Echo State Networks". In: *Artificial Neural Networks and Machine Learning – ICANN 2020*. Springer International Publishing, 2020, pp. 494–505. DOI: 10.1007/978-3-030-61616-8_40. URL: https://doi.org/10.1007/978-3-030-61616-8_40.

[195] Onno Ubbink. *Numerical prediction of two fluid systems with sharp interfaces*. Jan. 1997. URL: http://hdl.handle.net/10044/1/8604.

[196] H Versteeg and W Malalasekera. *An introduction to computational fluid dynamics*. en. 2nd ed. Philadelphia, PA: Prentice Hall, Feb. 2007.

[197] D. Verstraeten et al. "An experimental unification of reservoir computing methods". In: *Neural Networks* 20.3 (Apr. 2007), pp. 391–403. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2007.04.003. URL: http://dx.doi.org/10.1016/j.neunet.2007.04.003.

[198] P.R. Vlachas et al. "Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics". In: *Neural Networks* 126 (June 2020), pp. 191–

217. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2020.02.016. URL: http://dx.doi.org/10.1016/j.neunet.2020.02.016.

[199] Pantelis R. Vlachas et al. *Backpropagation Algorithms and Reservoir Computing in Recurrent Neural Networks for the Forecasting of Complex Spatiotemporal Dynamics*. 2019. DOI: 10.48550/ARXIV.1910.05266. URL: https://arxiv.org/abs/1910.05266.

[200] Pantelis R. Vlachas et al. "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2213 (May 2018), p. 20170844. ISSN: 1471-2946. DOI: 10.1098/rspa.2017.0844. URL: http://dx.doi.org/10.1098/rspa.2017.0844.

[201] Qiang Wang et al. "Numerical Simulation on Refractory Wear and Inclusion Formation in Continuous Casting Tundish". In: *Metallurgical and Materials Transactions B* 52.3 (2021), pp. 1344–1356. ISSN: 1543-1916. DOI: 10.1007/s11663-021-02097-7. URL: http://dx.doi.org/10.1007/s11663-021-02097-7.

[202] Yi Wang et al. "A comparative study of POD interpolation and POD projection methods for fast and accurate prediction of heat transfer problems". In: *International Journal of Heat and Mass Transfer* 55.17–18 (Aug. 2012), pp. 4827–4836. ISSN: 0017-9310. DOI: 10.1016/j.ijheatmasstransfer.2012.04.053. URL: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2012.04.053.

[203] Z.J. Wang et al. "High-order CFD methods: current status and perspective". In: *International Journal for Numerical Methods in Fluids* 72.8 (Jan. 2013), pp. 811–845. ISSN: 1097-0363. DOI: 10.1002/fld.3767. URL: http://dx.doi.org/10.1002/fld.3767.

[204] Ziming Wang et al. "Residence Time Distribution (RTD) Applications in Continuous Casting Tundish: A Review and New Perspectives". In: *Metals* 12.8 (Aug. 2022), p. 1366. ISSN: 2075-4701. DOI: 10.3390/met12081366. URL: http://dx.doi.org/10.3390/met12081366.

[205] Kyle Washabaugh et al. "Nonlinear Model Reduction for CFD Problems Using Local Reduced-Order Bases". In: *42nd AIAA Fluid Dynamics Conference and Exhibit*. American Institute of Aeronautics and Astronautics, June 2012. DOI: 10.2514/6.2012-2686. URL: http://dx.doi.org/10.2514/6.2012-2686.

[206] N.P. Waterson and H. Deconinck. "Design principles for bounded higher-order convection schemes – a unified approach". In: *Journal of Computational Physics* 224.1 (May 2007), pp. 182–207. ISSN: 0021-9991. DOI:

10.1016/j.jcp.2007.01.021. URL: http://dx.doi.org/10.1016/j.jcp.2007.01.021.

[207] H. G. Weller et al. "A tensorial approach to computational continuum mechanics using object-oriented techniques". In: *Computers in Physics* 12.6 (Nov. 1998), pp. 620–631. ISSN: 0894-1866. DOI: 10.1063/1.168744. URL: http://dx.doi.org/10.1063/1.168744.

[208] Tilmann Wittig, Rolf Schuhmann, and Thomas Weiland. "Model order reduction for large systems in computational electromagnetics". In: *Linear Algebra and its Applications* 415.2–3 (June 2006), pp. 499–530. ISSN: 0024-3795. DOI: 10.1016/j.laa.2004.06.023. URL: http://dx.doi.org/10.1016/j.laa.2004.06.023.

[209] World Steel Association. *World Steel in Figures 2024*. 2024. URL: https://worldsteel.org/wp-content/uploads/World-Steel-in-Figures-2024.pdf.

[210] D. Xiao et al. "Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation". In: *International Journal for Numerical Methods in Fluids* 79.11 (July 2015), pp. 580–595. ISSN: 1097-0363. DOI: 10.1002/fld.4066. URL: http://dx.doi.org/10.1002/fld.4066.

[211] Kyongmin Yeo and Igor Melnyk. "Deep learning algorithm for data-driven simulation of noisy dynamical system". In: *Journal of Computational Physics* 376 (Jan. 2019), pp. 1212–1231. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.024. URL: http://dx.doi.org/10.1016/j.jcp.2018.10.024.

[212] Lifeng Zhang and Brian G. Thomas. "Inclusions in Continuous Casting of Steel". In: *Proceedings of the XXIV National Steelmaking Symposium*. Morelia, Michoacán, Mexico, Nov. 2003, pp. 184–198. URL: https://ccc.illinois.edu/PDF%20Files/Presentations/03_Inclusions_in_Continuous_Casting_of_Steel.pdf.

[213] Lifeng Zhang et al. "Modelling of non-metallic inclusions in steel". In: *Mineral Processing and Extractive Metallurgy* 129.2 (Mar. 2020), pp. 184–206. ISSN: 2572-665X. DOI: 10.1080/25726641.2020.1738087. URL: http://dx.doi.org/10.1080/25726641.2020.1738087.

[214] Y.-T. Zhang and C.-W. Shu. " ENO and WENO Schemes ". In: *Handbook of Numerical Methods for Hyperbolic Problems - Basic and Fundamental Issues*. Elsevier, 2016, pp. 103–122. DOI: 10.1016/bs.hna.2016.09.009. URL: http://dx.doi.org/10.1016/bs.hna.2016.09.009.

[215] Zhiwei Zhao et al. "Diffeomorphism neural operator for various domains and parameters of partial differential equations". In: *Communications Physics* 8.1 (Jan. 2025). ISSN: 2399-3650. DOI: 10.1038/s42005-024-01911-3. URL: http://dx.doi.org/10.1038/s42005-024-01911-3.

[216] Roland S. Zimmermann and Ulrich Parlitz. "Observing spatio-temporal dynamics of excitable media using reservoir computing". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.4 (Apr. 2018). ISSN: 1089-7682. DOI: 10.1063/1.5022276. URL: http://dx.doi.org/10.1063/1.5022276.

[217] Beata Zorawski et al. *Automated transport separation using the neural shifted proper orthogonal decomposition*. 2024. DOI: 10.48550/ARXIV.2407.17539. URL: https://arxiv.org/abs/2407.17539.

[218] Jing Zou et al. "A review of deep learning-based deformable medical image registration". In: *Frontiers in Oncology* 12 (Dec. 2022). ISSN: 2234-943X. DOI: 10.3389/fonc.2022.1047215. URL: http://dx.doi.org/10.3389/fonc.2022.1047215.