ABSTRACT

DANIEL, BENJAMIN COLTON JANTZEN. Physics-Constrained Neural ODEs and Randomized Rank-Revealing QR with Applications in Fiber Coating Dynamics. (Under the direction of Hangjie Ji).

Thin liquid films flowing down a vertical cylindrical fiber, known as *fiber coating*, exhibit complex interfacial dynamics, leading to irregular wavy patterns and traveling droplets. These free surface flows, driven by Rayleigh-Plateau instabilities, play a crucial role in various engineering applications, including thermal desalination and water vapor and ultra-fine particle capture. In the inertialess limit, the evolution of droplet profiles in fiber coating systems is governed by a lubrication equation, which is a fourth-order parabolic degenerate partial differential equation (PDE). For large scale control and optimal experimental design problems involving fiber coating systems, repeatedly solving this PDE can be computationally expensive. To overcome this challenge, data-driven approaches are essential to accelerate the process. In this dissertation, we present two novel approaches to address distinct challenges in data-driven learning and subset selection with applications in fiber coating dynamics.

We first propose a family of Physics-Constrained Neural Ordinary Differential Equation (NODE) models to learn the droplet dynamics from PDE data. This approach learns the temporal modes from the proper orthogonal decomposition (POD) of the data while incorporating key solution properties into the learning process. Specifically, we leverage the structure of the NODE to enforce conserved mass and entropy conditions derived from the fiber coating PDE. We analytically demonstrate that NODE models enforcing the entropy condition guarantee the positivity of solutions at the POD reconstruction level. Numerical results show that while the vanilla NODE can produce unphysical profiles with negative liquid heights and fails to conserve mass, the proposed NODE models robustly generate predictions that preserve mass conservation and ensure the positivity of the reconstructed solutions.

Next, we analyze a randomized two-stage algorithm for rank-revealing QR (RRQR) factorizations. RRQRs have applications in solving least squares problems and creating low-rank approximations. Computing a RRQR involves identifying linearly independent columns of a matrix, a problem known as subset selection. Subset selection is a combinatorially challenging problem, and for large matrices, finding the optimal set of columns is computationally infeasible. Because of this, subset selection algorithms seek only to find an approximately optimal subset of linearly independent columns, which can be done efficiently.

Our focus is on the randomized two-stage RRQR algorithm, which uses randomization techniques to approximate the dominant right singular vectors of a matrix. These approxi-

mated singular vectors are then used to select the desired columns. In this work, we establish structural and probabilistic singular value bounds to demonstrate the effectiveness of the RRQR algorithm. Numerical results using various test matrices, including simulation data from the fiber coating PDE demonstrate the rank-revealing properties of the algorithm and validate the established bounds.

© Copyright 2025 by Benjamin Colton Jantzen Daniel

All Rights Reserved

Physics-Constrained Neural ODEs and Randomized Rank-Revealing QR with Applications in Fiber Coating Dynamics

by
Benjamin Colton Jantzen Daniel

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina
2025

APPROVED BY:

Mohammad Farazmand	Mansoor Haider
Ilse Ipsen	Arvind Saibaba

Hangjie Ji
Chair of Advisory Committee

DEDICATION

To my family and friends.

BIOGRAPHY

Benjamin Daniel was born in San Antonio, Texas. He got his high school diploma from Chantilly High School in 2014, and graduated from James Madison University with a degree in Mathematics in 2018. He is currently working on his Doctorate in Applied Mathematics at North Carolina State University under the direction of Hangjie Ji.

ACKNOWLEDGEMENTS

My graduate career would not have been possible without the support and encouragement of numerous individuals who dedicated significant time and energy to my success.

First, I would like to thank my advisor, Dr. Hangjie Ji for her guidance and patience. Somebody who, despite her constantly busy schedules, always had time available for me. I thank Dr. Ilse Ipsen and Dr. Arvind Saibaba for their work and contributions in Chapter 4 of this dissertation, and for ingraining a passion for numerical linear algebra into me that will stay there indefinitely. I thank my committee members, Dr. Mohammad Farazmand and Dr. Mansoor Haider for their helpful comments and feedback. This dissertation would look substantially different without the ideas they've discussed with me. Thanks to Dr. Aritra Mitra for agreeing to be my graduate school representative.

To all my friends at NC State; Tim, Michael, Catie, John, Will, Chris, and Kristen, to name a few; thank you all for making this journey enjoyable. I also want to extend thanks to everybody I've played Frisbee or board games with; it's been fun.

Thank you to parents, Mark and Christa, and my Paw Paw Merlie for their constant support, and to my siblings, Tad, Jon, Josiah, and Leah for always finding ways to keep me entertained.

Lastly, I would like to acknowledge the funding that supported my research through National Science Foundation grants DMS-1745654 and DMS-2309774.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

This dissertation focuses on the development of physics-constrained neural ODEs and the analysis of randomized rank-revealing QR (RRQR) algorithms, with applications in fiber coating dynamics. We review physics-based models for fiber coating dynamics in §1.1 and motivate the importance of a neural ODE-based approach to learning droplet dynamics in §1.2. In §1.3, we introduce main concepts of randomized RRQR algorithms, followed by a discussion of the structure of the dissertation in §1.4.

1.1 Physics-based Models for Fiber Coating Dynamics

Thin liquid films flowing down a vertical fiber, known as *fiber coating*, exhibit intricate interface dynamics, including droplet formation, coalescence, and traveling wave patterns (Quéré 1990). These phenomena are driven by Rayleigh-Plateau instability and play an important role in a variety of engineering applications, including gas absorption (Chinju et al. 2000), mass and heat exchangers (Zeng et al. 2017), particle capturing (Sadeghpour et al. 2019), and desalination (Sadeghpour et al. 2021). Understanding the droplet behaviors in fiber coating dynamics is crucial for optimizing experimental designs for practical applications.

Many studies on fiber coating dynamics have been devoted to physics-based modeling under different assumptions, typically leading to evolution equations for the height of the

free surface that incorporate the interplay between gravity, surface tension, and effects of the substrate geometry (e.g. Ruyer-Quil et al. (2008)). Recent advances in the control (Biswal et al. 2024) and large-scale engineering applications (Sadeghpour et al. 2021) involving fiber coating systems also necessitate the development of data-driven reduced-order modeling techniques. Leveraging experimental measurements and high-fidelity simulation data to construct surrogate models for the rapid prediction of droplet dynamics can significantly accelerate computation in these applications.

Earlier experimental works by Kliakhandler et al. (2001) showed that there exist three distinct flow regimes in fiber coating dynamics at different flow rates: a convective regime characterized by droplet coalescence and irregular wavy patterns, a Rayleigh-Plateau regime with a train of stable traveling droplets, and an isolated droplet regime featuring widely spaced traveling droplets separated by smaller and slowly moving waves. The flow regimes can be predicted by Orr-Sommerfeld stability analysis (Ruyer-Quil et al. 2008; Solorio and Sen 1987), and it has also been experimentally and analytically demonstrated that altering liquid properties, fiber radius, and nozzle geometries can also change the flow regimes (Sadeghpour et al. 2017; Ji et al. 2020).

This classical fluid dynamics problem has attracted many researchers in the last two decades. For fiber coating dynamics with high and moderate Reynolds numbers, methods such as full Navier-Stokes analysis, weighted residual integral boundary layer (WRIBL) modeling, and the control-volume approach have been investigated. The WRIBL approach applies a Galerkin weighted residual method to account for moderate inertia effects and streamwise viscous dissipation (Ruyer-Quil et al. 2009, 2008), yielding a system of coupled equations for both film thickness and local flow rate. The WRIBL model has also been used to demonstrate the impact of nozzle geometry on downstream flow dynamics (Ji et al. 2020). The control-volume model characterizes the conservation of mass and axial momentum with an imposed axial velocity profile, leading to a coupled system for film thickness and axial velocity (Ruan et al. 2021; Taranets et al. 2024). To model the flow at low Reynolds numbers, under the thin-film assumption that the characteristic film thickness is significantly smaller than a certain characteristic length scale, the classical lubrication approximation has been used to describe the evolution of the film thickness (Chang and Demekhin 1999; Frenkel 1992; Kalliadasis and Chang 1994; Craster and Matar 2006). Many of these models have also been validated against experimental observations (Craster and Matar 2006; Ji et al. 2019; Ruyer-Quil et al. 2009; Duprat et al. 2009).

Recent works on physics-based modeling for fiber coating dynamics have focused on the incorporation of thermocapillary effects (Ji et al. 2021), rotation (Liu and Ding 2020), surfactant-laden fluids (Nair and Sharma 2020), chemical reactions (Chattopadhyay and Ji

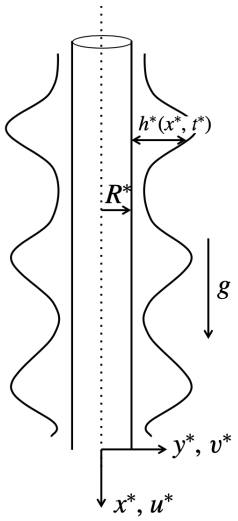

Figure 1.1: Schematic figure of a thin liquid flowing down a vertical cylindrical fiber (Ji et al. 2019), where R^* represents the fiber radius, h^* represents the liquid film thickness, g is the gravitational constant, and u^*, v^* represent the velocity in the axial and radial directions. Here, the symbol * indicates dimensional variables and we drop the * for variables in the dimensionless model (1.1).

2024; Chattopadhyay 2024; Li and Chao 2020), wind effects (Cazaubiel and Carlson 2023), and fiber geometry into the system (Xie et al. 2021). While most models assume axisymmetric droplet profiles, the work of Gabbard and Bostwick (2021) investigated asymmetric instabilities in droplets flowing down a vertical fiber. The well-posedness of PDE solutions for a lubrication-type PDE and a control-volume model for fiber coating have been studied by Ji et al. (2022) and Taranets et al. (2024). More recently, boundary control algorithms (Biswal et al. 2024) and positivity-preserving numerical schemes for simulations on coarse grids (Kim et al. 2024) have been developed for fiber coating systems.

In this work, we will focus on a lubrication model for fiber coating dynamics. Craster and Matar (2006) developed a lubrication model for fiber coating dynamics assuming that the characteristic film thickness is much smaller than the capillary length. The work of Ji et al. (2019) built upon this model and proposed a model that incorporates slip condition, fully nonlinear curvature terms and a film stabilization term in the dynamic pressure.

We consider the formulation of the fiber coating model given in Ji et al. (2019) and Craster and Matar (2006). Let h be the dimensionless film thickness of an axisymmetric thin liquid flowing down a vertical cylindrical fiber of radius R, depicted in Figure 1.1. The dynamics of

the film thickness $h(x,t)$ satisfies the fourth-order nonlinear degenerate PDE,

$$\frac{\partial}{\partial t}\left(h+\frac{\alpha}{2}h^2\right)+\frac{\partial}{\partial x}\left[M(h)\left(1-\frac{\partial}{\partial x}\left[Z(h)-\frac{\partial^2 h}{\partial x^2}\right]\right)\right]=0, \tag{1.1a}$$

where the mobility function $M(h)$ and the azimuthal curvature $Z(h)$ take the form

$$M(h)=\frac{h^3\phi(\alpha h)}{3\phi(\alpha)}, \quad Z(h)=\frac{\alpha}{\eta(1+\alpha h)}, \tag{1.1b}$$

where the shape factor function $\phi(X)$ is given by

$$\phi(X)=\frac{3}{16X^3}\left[(1+X)^4(4\ln(1+X)-3)+4(1+X)^2-1\right]. \tag{1.1c}$$

Equation (1.1) is a fourth-order nonlinear degenerate partial differential equation for the dimensionless film thickness h. Here, the term $\frac{\partial}{\partial x}M(h)$ accounts for the gravitational effect, $\alpha=\mathcal{H}/R$ is the aspect ratio of the Nusselt film thickness $\mathcal{H}$ and the radius of the fiber R, η is a scaling ratio dependent on the characteristic lengthscales in the axial and radial directions. The surface tension plays a dual role in the flow instability; the azimuthal curvature term $Z(h)$ is destabilizing, while the streamwise surface tension term h_{xx} is stabilizing. The evolution equation (1.1) is consistent with the model proposed in Craster and Matar (2006) up to a rescaling and is identical to the film stabilization model discussed in Ji et al. (2019) without the film stabilization term. This model characterizes all three flow regimes discussed previously and has been verified against experimental data (Ji et al. 2019; Craster and Matar 2006).

1.2 Data-driven Models and Neural ODEs

While extensive modeling and analytical works have been established for these liquid film dynamics, numerically solving the PDE models can be computationally expensive, particularly when repeated simulations or simulations on large spatial or temporal domains are needed. Nevertheless, such simulations are essential for their applications in optimal control problems and experimental designs (Biswal et al. 2024; Fu et al. 2024). This motivates us to develop reduced-order data-driven modeling techniques for fiber coating dynamics.

Reduced-order models are efficient alternatives to full-order models in scientific simulations, offering significant computational savings while maintaining accuracy in modeling complex dynamical systems. Various data-driven techniques, such as the proper orthogonal decomposition (POD) and dynamic mode decomposition (DMD) (Schmid 2010), have been

developed for constructing reduced-order models.

The POD decomposes a function $h(x,t)$ into a set of temporal and spatial functions, called coefficients. These coefficients can be used to form a reduced basis for $h(x,t)$ that maximizes the variance of the reduced order model. Recently, machine learning approaches such as recurrent neural networks and neural ODEs (NODEs) have been used to solve for the temporal coefficients of the POD (Baker et al. 2023; Kani and Elsheikh 2017, 2019). The vanilla NODE has several weaknesses, such as a potentially high computational cost for both the forward evaluation and solving the adjoint equation due to the stiffness of the learned model (Xia et al. 2021). In addition, NODE can suffer from vanishing in the adjoint state, making NODEs unreliable for learning long-term dynamics (Bengio et al. 1994). Several variants of NODEs have been developed to address some of these issues. A brief selection of NODE variants include the augmented NODE (Dupont et al. 2019), which creates a more expressive model by expanding the dimensionality of the hidden state; the Heavy Ball NODE (Xia et al. 2021), which leverages momentum to fix the vanishing gradient and stiffness issues; and the Stochastic Neural ODE (Li et al. 2019), which incorporates stochasticity and allows for modeling systems where uncertainty or noise plays a significant role.

A weakness of using NODEs and their variants directly for the fiber coating model is that the machine learning architecture ignores many important physical properties of the system. For example, the mobility function $M(h) \propto h^3$ in the fiber coating model (1.1) means the PDE is degenerate, and well-posed classical solutions cease to exist when h becomes non-positive. Thus, if h becomes zero or negative at any point while solving the model numerically, the numerical solution develops a singularity in time and ceases to exist beyond this critical time. Therefore, constructing positivity-preserving solutions is critical in developing robust reduced-order models for fiber coating dynamics. We are also interested in enforcing the conservation of mass in our solution to (1.1), which would preserve the volume of liquid on the fiber and yield more physically realistic droplet dynamics.

Enforcing conserved quantities and other physical constraints is critical in developing reduced-order models for real-world systems. Recently, Anderson and Farazmand (2022) developed the method of reduced-order nonlinear solutions (RONS) for nonlinear PDEs with conserved quantities. This approach has been extended to enforcing conserved quantities in Galerkin-type truncations (Hilliard and Farazmand 2024). Researchers have also previously extended neural networks by incorporating prior knowledge about conserved quantities to learn dynamical systems. Greydanus et al. (2019) proposed the Hamiltonian neural network, which conserves the system energy. Finzi et al. (2020a) proposed LieConv, a convolutional neural network that conserves linear and angular momentum, and later extended the Hamiltonian neural network to a system with holonomic constraints (Finzi et al. 2020b). Matsubara et al.

(2020) proposed a neural network that preserves total mass. Conservation laws were recently incorporated into NODEs by Matsubara and Yaguchi (2023), who developed a variant of the NODE that preserves first-order integrals by projecting the dynamics onto a manifold where those integrals are constant. In Chapter 3, we use techniques from Anderson and Farazmand (2022) to develop a family of physics-constrained NODEs for learning the POD modes of fiber coating data.

1.3 Randomized Algorithms for Rank-Revealing QR

Column subset selection, which we will often refer to as simply *subset selection*, is the process of selecting columns of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ such that the selected columns form an approximate basis for the range of $\mathbf{A}$, denoted by $\mathcal{R}(\mathbf{A})$, and the discarded columns can be approximated by a linear combination of the selected ones. If $\mathbf{A}$ has rank equal to k, then we can select k columns that fulfill these requirements exactly. More often, $\mathbf{A}$ will not have rank exactly equal to k, but rather have approximate rank k, in which case, we want to find a good rank-k approximation to $\mathcal{R}(\mathbf{A})$.

The famed Singular Value Decomposition (SVD) provides an orthonormal basis for the best-possible rank$-k$ approximation to $\mathcal{R}(\mathbf{A})$ through its dominant k left singular vectors. However, the SVD can be expensive to compute for large matrices, making its use impractical. In addition, updating the SVD when a new row or column is added to $\mathbf{A}$ can also be expensive. Subset selection can provide a faster way to find an approximate basis, and while this basis is not guaranteed to be optimal (and will not be in most cases), it has the bonus of interpretability, as the elements in the basis are columns of $\mathbf{A}$.

Subset selection is sometimes done with a QR decomposition $\mathbf{A}\mathbf{\Pi} = \mathbf{Q}\mathbf{R}$, where $\mathbf{\Pi} \in \mathbb{R}^{n \times n}$ is a permutation matrix. The leading k columns of $\mathbf{\Pi}$ correspond to the k selected columns of $\mathbf{A}$, and the upper triangular matrix $\mathbf{R}$ has submatrices that provide information about the singular values of $\mathbf{A}$. This decomposition is called a Rank-Revealing QR factorization and has applications in rank-deficient least squares problems, low-rank approximations, and nonsymmetric eigenproblems (Chan and Hansen 1992; Gu and Eisenstat 1996).

Many researchers have developed and analyzed RRQR algorithms. In 1965, Businger and Golub's QR with column pivoting (Businger and Golub 1966) was the first such algorithm to be developed. Then in 1976, Golub, Klema, and Stewart developed an RRQR factorization that uses QR with column pivoting on the dominant right singular vectors of $\mathbf{A}$ (Golub et al. 1976). A practical limitation of this technique is that it assumes the singular vectors are known or can be computed efficiently. If the matrix is large, then these vectors may be expensive to compute and other techniques may need to be considered. Hong and Pan (1992)

proved the existence of RRQRs by showing there always exists a permutation matrix that produces a singular value approximation that is accurate up to a factor of n and k. Gu and Eisenstat (1996) expanded on this work by providing an algorithm for a strong RRQR, which achieves estimates of singular values with similar theoretical bounds, and can be computed in polynomial time.

Recently, many RRQRs algorithms are being developed for use on large problems. Traditional QR with column pivoting passes over the trailing columns of $\mathbf{A}$ each time a new pivot is selected. This makes this technique not viable in problems where communication is a bottleneck. Duersch and Gu (2017) developed an RRQR algorithm that avoids communication by using random sampling to block columns of $\mathbf{A}$ together and performs pivots as blocks. Martinsson (2015) independently developed a similar technique around the same time. Demmel et al. (2015) developed and analyzed an RRQR algorithm that uses tournament pivoting to avoid repeating full passes over $\mathbf{A}$ on the selection of each pivot. Armstrong et al. (2023) developed a RRQR suitable for large matrices by introducing randomization into the Golub, Klema, and Stewart algorithm.

Other rank-revealing factorizations that use column selection include the CUR decomposition (Cheng et al. 2005) and interpolative decomposition (Mahoney and Drineas 2009). The CUR decomposition and interpolative decomposition use columns and/or rows of $\mathbf{A}$ in the factorization itself, which capitalizes on any structure $\mathbf{A}$ may have, such as sparsity or non-negativity (Martinsson 2016). The CUR decomposition is a factorization $\hat{\mathbf{A}} := \mathbf{CUR}$ where $\mathbf{C} \in \mathbb{R}^{m\times k}$ consists of columns of $\mathbf{A}$, $\mathbf{U} \in \mathbb{R}^{k\times k}$, and $\mathbf{R} \in \mathbb{R}^{k\times n}$ consists of rows of $\mathbf{A}$, where the columns and rows of $\mathbf{A}$ are chosen such that $\hat{\mathbf{A}}$ is a reasonable low-rank approximation of $\mathbf{A}$. A (column) interpolative decomposition of $\mathbf{A}$ is a factorization $\mathbf{A} = \mathbf{CX}$, where $\mathbf{C} \in \mathbb{R}^{m\times k}$ is made up of columns of $\mathbf{A}$, and $\mathbf{X} \in \mathbb{R}^{k\times n}$ has the $k \times k$ identity matrix as a submatrix and has no elements that exceed 1 in magnitude.

Another rank-revealing factorizations related to the RRQR is the UTV factorization, $\mathbf{A} = \mathbf{UTV}^{\mathrm{T}}$, where $\mathbf{T} \in \mathbb{R}^{m\times n}$ is triangular and $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are orthogonal (Golub and Van Loan 2013, Section 5.4.6). The RRQR is a special case of a UTV factorization where $\mathbf{V}$ is a permutation matrix. Randomized techniques for computing a UTV factorization are presented in Martinsson et al. (2019), and algorithms for UTV factorizations have been developed for problems that require optimization for parallel computing (Heavner et al. 2021) and matrices too large to store in RAM (Heavner et al. 2020).

1.4 Structure of Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, we review background information pertaining to the results in this dissertation, including the formulation of the fiber coating model, POD, NODEs, and RRQR. In Chapter 3, we focus on developing a physics-constrained data-driven learning algorithm for fiber coating dynamics based on NODEs and POD. Motivated by recent works on developing quantity-preserving reduced-order models (Anderson and Farazmand 2022; Matsubara and Yaguchi 2023), our algorithm enforces mass conservation and solution positivity in NODEs at the POD reconstruction level. Then, in Chapter 4, we analyze a two-step approach for a randomized rank-revealing QR for a matrix $\mathbf{A}$. First, we find a matrix $\mathbf{W}$ whose columns form an orthogonal basis that approximates the range of $\mathbf{A}$. Then, we select rows of $\mathbf{W}$ and use that same permutation to obtain columns of $\mathbf{A}$. We analyze a realization of this technique proposed in Armstrong et al. (2023) that uses randomized subspace iteration (Halko et al. 2011, Algorithm 4.4) to obtain the matrix $\mathbf{W}$ and uses a strong rank-revealing QR (Gu and Eisenstat 1996) to select rows from that $\mathbf{W}$. This analysis provides theoretical bounds for all singular values of the selected columns and the discarded columns of $\mathbf{A}$, which will then be demonstrated using simulation data from the fiber coating model.

CHAPTER

2

BACKGROUND

2.1 Model Formulation of the Fiber Coating Equation

In this section, we review the derivation of the fiber coating model, a lubrication-type equation characterizing the dynamics of thin liquid fluids flowing down a vertical cylindrical fiber, following the methods in Ji et al. (2019) and Ruyer-Quil et al. (2008). We consider a flow of two-dimensional axisymmetric Newtonian fluid down a vertical cylindrical fiber, given in Figure 2.1 and assume the fluid has constant density and viscosity. We begin with the dimensional incompressible Navier-Stokes equations and the equation of continuity,

$$\frac{D\vec{u}^*}{Dt^*} = -\frac{1}{\rho}\nabla p^* + \mathbf{g} + \nu\nabla^2\vec{u}^*, \tag{2.1}$$

$$\nabla \cdot \vec{u}^* = 0 \tag{2.2}$$

where parameters with star superscripts represent dimensional quantities, with t^* being time, $\vec{u}^*$ the velocity vector,

$$\frac{D\vec{u}^*}{Dt^*} = \frac{\partial\vec{u}^*}{\partial t^*} + (\vec{u}^* \cdot \nabla\vec{u}^*)$$

is the material derivative, ρ is density, p^* is pressure, $\mathbf{g}$ is the gravity vector, and $\nu = \mu/\rho$ is kinematic viscosity, where μ is dynamic viscosity.

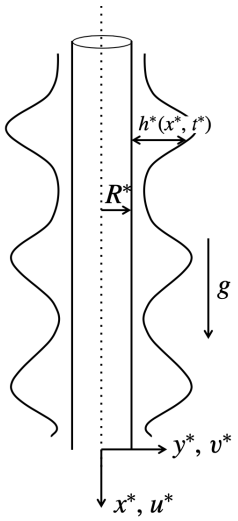

Figure 2.1: Schematic figure of a thin liquid flowing down a vertical cylindrical fiber (Ji et al. 2019).

Since the liquid flows down a cylindrical fiber, we express the equation (2.1) and (2.2) in cylindrical coordinates. Let $\vec{u}^* = [u^*, v^*]^{\mathrm{T}}$, where u^* and v^* are the velocities in the axial (x^*) and radial (y^*) directions, respectively. We omit the angular velocity since the flow is assumed axisymmetic. The material derivative and Laplace operator can be written in cylindrical coordinates as

$$\frac{D\vec{u}^*}{Dt^*} = \frac{\partial \vec{u}^*}{\partial t^*} + (\vec{u}^* \cdot \nabla \vec{u}^*) = \begin{bmatrix} u^*_{t^*} + u^* u^*_{x^*} + v^* u^*_{y^*} \\ v^*_{t^*} + u^* v^*_{x^*} + v^* v^*_{y^*} \end{bmatrix},$$

$$\nabla^2 \vec{u}^* = \begin{bmatrix} u^*_{x^*x^*} + \frac{1}{y^*} u^*_{y^*} + u^*_{y^*y^*} \\ \frac{1}{y^*} v^*_{y^*} + v^*_{y^*y^*} - \frac{1}{(y^*)^2} v^* + v^*_{x^*x^*} \end{bmatrix}.$$

Then the Navier-Stokes and continuity equations (2.1) and (2.2) can be written as

$$u^*_{t^*} + u^* u^*_{x^*} + v^* u^*_{y^*} = -\frac{1}{\rho} p^*_{x^*} + g + \nu \left(u^*_{x^*x^*} + \frac{1}{y^*} u^*_{y^*} + u^*_{y^*y^*} \right), \tag{2.3}$$

$$v^*_{t^*} + u^* v^*_{x^*} + v^* v^*_{y^*} = -\frac{1}{\rho} p^*_{y^*} + \nu \left(\frac{1}{y^*} v^*_{y^*} + v^*_{y^*y^*} - \frac{1}{(y^*)^2} v^* + v^*_{x^*x^*} \right). \tag{2.4}$$

$$v^*_{y^*} + \frac{1}{y^*} v^* + u^*_{x^*} = 0. \tag{2.5}$$

At the interface between the fiber and fluid, we impose no-penetration ($v^* = 0$) and

no-slip ($u^* = 0$) boundary conditions at $y^* = R^*$, where R^* is the radius of the fiber. This gives $\vec{u}^* = 0$ at $y^* = R^*$ and states that there is no velocity in either direction at the interface. Let $h^*(x^*, t^*)$ be the thickness of the fluid. The normal and shear stress balances at the free surface, $y^* = h^* + R^*$, are given by

$$p^* = \frac{2\mu}{1 + (h^*_{x^*})^2} \left[(h^*_{x^*})^2 u^*_{x^*} - h^*_{x^*}(v^*_{x^*} + u^*_{y^*}) + v^*_{y^*} \right] + \frac{\sigma}{(1 + (h^*_{x^*})^2)^{3/2}} \left(\frac{1 + (h^*_{x^*})^2}{h^* + R^*} - h^*_{x^*x^*} \right), \tag{2.6}$$

$$\left(1 - (h^*_{x^*})^2\right)(v^*_{x^*} + u^*_{y^*}) + 2h^*_{x^*}(v^*_{y^*} - u^*_{x^*}) = 0, \tag{2.7}$$

where σ is surface tension and scales the total curvature of the free interface. See Appendix A.1 for detailed derivations of (2.6) and (2.7). We assume no flow through the free surface, which yields a kinematic boundary condition

$$h^*_{t^*} + u^* h^*_{x^*} = v^* \qquad \text{at } y^* = h^* + R^*. \tag{2.8}$$

We now simplify the model by making it non-dimensional following the work of Duprat et al. (2009). The length scale in the radial direction, y, is the Nusselt film thickness, denoted by $\mathcal{H}$, and the length scale in the stream-wise direction, x, is $\mathcal{L} = \mathcal{H}/\varepsilon$, where ε is given by $\varepsilon = (\rho g \mathcal{H}^2/\sigma)^{1/3} = \mathrm{We}^{1/3}$, where the Weber number, We, compares the capillary length $l_c = \sqrt{\sigma/(\rho g)}$ to the radial length scale $\mathcal{H}$. Here, the Nusselt film thickness is determined by the thickness of a spatially-uniform fluid film coating the fiber without any spatial perturbations. The characteristic stream-wise velocity is $\mathcal{U} = (g\mathcal{H}^2)/\nu$, the characteristic radial velocity is $\mathcal{V} = \epsilon\mathcal{U}$, and the pressure and time scales are given by $\mathcal{P} = \rho g \mathcal{L}$ and $\mathcal{T} = (\nu\mathcal{L})/(g\mathcal{H}^2)$, respectively.

We non-dimensionalize the model by letting $y^* = \mathcal{H}y$, $x^* = \mathcal{L}x$, $u^* = \mathcal{U}u$, $v^* = \mathcal{V}v$, $p^* = \mathcal{P}p$, and $t^* = \mathcal{T}\tilde{t}$. With these scales, we write the dimensionless Navier–Stokes equations (2.3) and (2.4) as

$$\varepsilon^2 \mathrm{Re}(u_{\tilde{t}} + uu_x + vu_y) = -p_x + 1 + \frac{u_y}{y} + u_{yy} + \varepsilon^2 u_{xx}, \tag{2.9}$$

$$\varepsilon^4 \mathrm{Re}(v_{\tilde{t}} + vv_u + uv_x) = -p_y + \varepsilon^2 \left(\frac{v_y}{y} + v_{yy} - \frac{v}{y^2} \right) + \varepsilon^4 v_{xx}, \tag{2.10}$$

where $\mathrm{Re} = \mathcal{U}\mathcal{L}/\nu$ is the Reynolds number. The dimensionless continuity equation (2.5) is

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{v}{y} = 0, \tag{2.11}$$

and the dimensionless balances of normal (2.6) and tangential stresses (2.7) at $y = h + R$ are now given by

$$p = \frac{\alpha}{\varepsilon^2(1+\alpha h)(1+\varepsilon^2 h_x^2)^{1/2}} - \frac{h_{xx}}{(1+\varepsilon^2 h_x^2)^{3/2}} + \frac{\varepsilon^4}{1+\varepsilon^2 h_x^2}\left[h_x^2 u_x - h_x v_x + \varepsilon^2(-h_x u_y + v_y)\right], \tag{2.12}$$

$$(1-\varepsilon^2 h_x^2)(\varepsilon^2 v_x + u_y) + 2\varepsilon^2 h_x(v_y - u_x) = 0, \tag{2.13}$$

where $\alpha = \mathcal{H}/R^*$ is the aspect ratio of the characteristic film thickness and fiber radius. The dimensionless fiber radius is $R = R^*/\mathcal{H}$. In the right-hand side of (2.12), the first term represents the destabilizing azimuthal curvature, which causes the formation of droplets, and the second term is the stabilizing stream-wise curvature of the free surface, which stabilizes the liquid film. The dimensionless no-penetration and no-slip boundary conditions are

$$v = 0 \text{ and } u = 0 \qquad \text{at } y = R, \tag{2.14}$$

and the dimensionless kinematic boundary condition (2.8) is

$$h_{\tilde{t}} + u h_x = v \qquad \text{at } y = R + h. \tag{2.15}$$

To further reduce the model, we assume that the viscous forces and inertial forces are of the same magnitude, giving $\mathrm{Re} = \mathcal{O}(1)$. We also assume that $\mathcal{H}$ is small, giving $\varepsilon \ll 1$ and that $\alpha/\varepsilon^2 = \mathcal{O}(1)$. With these assumptions, we omit all terms of order $\mathcal{O}(\varepsilon^2)$ in the dimensionless Navier-Stokes equations (2.9) and (2.10) to get

$$1 - \frac{\partial p}{\partial x} + \frac{\partial^2 u}{\partial y^2} + \frac{1}{y}\frac{\partial u}{\partial y} = 0, \tag{2.16}$$

$$-\frac{\partial p}{\partial y} = 0, \tag{2.17}$$

and the balance of tangential stresses at the free surface (2.13) becomes

$$\frac{\partial u}{\partial y} = 0 \qquad \text{at } y = h + R. \tag{2.18}$$

Furthermore, the azimuthal and streamwise curvature terms in (2.12) can be linearized using

Taylor series, giving

$$\frac{\alpha}{\varepsilon^2(1+\alpha h)\sqrt{1+\varepsilon^2 h_x^2}} = \frac{\alpha}{\varepsilon^2(1+\alpha h)}\left(1-\frac{1}{2}\varepsilon^2 h_x^2+\mathcal{O}(\varepsilon^4)\right),$$

$$\frac{h_{xx}}{(1+\varepsilon^2 h_x^2)^{3/2}} = h_{xx}\left(1-\frac{3}{2}\varepsilon^2 h_x^2+\mathcal{O}(\varepsilon^4)\right).$$

Omitting the terms of order $\mathcal{O}(\varepsilon^2)$ reduces the balance of normal stresses at the free surface (2.12) to

$$p = \frac{\alpha}{\eta(1+\alpha h)} - \frac{\partial^2 h}{\partial x^2}, \tag{2.19}$$

where $\eta = \varepsilon^2$ is a scaling parameter.

We now consider a uniform Nusselt flow without any interfacial instabilities, denoted by $u_0(y)$. This is a simplified base state of the model where the flow changes in the radial direction, y, and is constant in t and x. Since u_0 is not dependent on x, (2.16) gives an expression an ODE describing u_0,

$$1 + \frac{du_0}{dy} + \frac{1}{y}\frac{du_0}{dy} = 0 \tag{2.20}$$

with no-slip and shear-stress boundary conditions

$$u_0 = 0 \text{ at } y = R \quad \text{and} \quad \frac{du_0}{dy} = 0 \text{ at } y = h + R. \tag{2.21}$$

We take the L_2 inner product of the continuity equation (2.16) with the Nusselt uniform solution u_0, to obtain

$$\int_R^{h+R}\left(1+\frac{\partial^2 u}{\partial y^2}+\frac{1}{y}\frac{\partial u}{\partial y}\right)u_0 y dy = \int_R^{h+R}\frac{\partial p}{\partial x}u_0 y dy. \tag{2.22}$$

Since u_0 satisfies equation (2.20), by integrating by parts and applying the no-slip and no-penetration boundary conditions (2.14) and the shear stress condition (2.18), we obtain the local flow rate q satisfying

$$q = \left(1-\frac{\partial p}{\partial x}\right)q_0, \tag{2.23}$$

where

$$q_0 = \frac{1}{R}\int_R^{R+h} u_0 y dy \tag{2.24}$$

is the flow rate of the uniform Nusselt layer per unit circumference length. We refer to Appendix A.1 for a detailed derivation. To obtain an expression for q_0, we solve the initial

value problem given by (2.20) and (2.21) for u_0 to obtain

$$u_0(y) = -\frac{1}{4}(y^2 - R^2) + \frac{1}{2}(h + R^2)\ln\left(\frac{y}{R}\right). \tag{2.25}$$

Using the expression (2.25) for u_0, we evaluate the integral in (2.24) to obtain

$$q_0 = \frac{h^3}{3}\phi(\alpha h), \tag{2.26}$$

where $\phi(X)$ is given by

$$\phi(X) = \frac{3}{16X^3}\left((1+X)^4(4\ln(1+X) - 3) + 4(1+X)^2 - 1\right).$$

Since $\phi(X)$ incorporates the geometry of the fiber, we call ϕ the shape factor.

Now we use the rescaled kinematic boundary conditions (2.15), no-slip and no-penetration boundary conditions (2.14), and the shear stress condition (2.18) to get the mass conservation equation,

$$(1+\alpha h)\frac{\partial h}{\partial \tilde{t}} + \frac{\partial q}{\partial x} = 0, \quad \text{where } q = \frac{1}{R}\int_R^{h+R} uydy. \tag{2.27}$$

This equation balances the time derivative of the cross-sectional area of the fluid, $(1 + \alpha h)\left(\partial h/\partial \tilde{t}\right)$, with the flux, $\partial q/\partial x$. We refer to Appendix A.1 for a detailed derivation of the mass conservation equation (2.27).

We finish the derivation by combining the mass conservation equation (2.27) with our expressions for p, q, and q_0 (equations (2.19), (2.23), and (2.26), respectively) and rescaling the time by letting $\tilde{t} = \phi(\alpha)t$ to obtain the dimensionless equation for $h(x,t)$,

$$\frac{\partial}{\partial t}\left(h + \frac{\alpha}{2}h^2\right) + \frac{\partial}{\partial x}\left(\frac{h^3\phi(\alpha h)}{3\phi(\alpha)}\left[1 - \frac{\partial}{\partial x}\left(\frac{\alpha}{\eta(1+\alpha h)} - \frac{\partial^2 h}{\partial x^2}\right)\right]\right) = 0.$$

For simplicity, we define the mobility function as $M(h) := (h^3\phi(\alpha h))/(3\phi(\alpha))$ The time rescaling from $\tilde{t}$ to t gives a normalized mobility function such that $M(1) = 1/3$. It is also convenient to define the azimuthal curvature as $Z(h) := \alpha/(\eta(1+\alpha h))$ and rewrite our governing equation as

$$\frac{\partial}{\partial t}\left(h + \frac{\alpha}{2}h^2\right) + \frac{\partial}{\partial x}\left(M(h)\left[1 - \frac{\partial}{\partial x}\left(Z(h) - \frac{\partial^2 h}{\partial x^2}\right)\right]\right) = 0. \tag{2.28}$$

Equation (2.28) is a degenerate, nonlinear, fourth-order PDE that models the film thickness of a fluid flowing down a vertical cylindrical fiber. This model incorporates surface tension, gravity and azimuthal instabilities, but neglects inertia and stream-wise viscous effects (Ji

et al. 2019).

2.2 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) is a data-driven method that takes the solution to a dynamical system and computes orthogonal modes spanning the same solution space (Benner et al. 2015). This decomposition allows expressing the (mean-subtracted) solution as a sum of separable equations such that the spatial and temporal modes are ordered based on the amount of variation they describe in the original solution, enabling dimensionality reduction by truncating lower-order modes. The k-dimensional POD approximation of $h(x,t)$ is

$$h(x,t) - \overline{h}(t) \approx \sum_{i=1}^{k} u_i(t) v_i(x), \tag{2.29}$$

where $\overline{h}(t)$ is the spatial mean of $h(x,t)$ at time t, and the functions $v_i(x)$ are orthogonal. To compute the (discretized) POD, we start with a set of data, which are snapshots derived from either physical observations or numerical simulations. These snapshots are consolidated into a matrix $\mathbf{A} \in \mathbb{R}^{N_t \times N_x}$, where rows of $\mathbf{A}$ correspond to the solution h at specific time steps, N_t is the number of snapshots, and N_x is the number of spatial grid points in the data. The dominant eigenvectors of the covariance matrix of $\mathbf{A}$ form the orthonormal basis $\{v_i(x)\}$. The temporal modes, $\{u_i(t)\}$, are the coefficients obtained when projecting the data onto the spatial modes $\{v_i(x)\}$. Given a discretization of the solution h, the proper orthogonal decomposition is given by the singular value decomposition (SVD). The SVD is also pivotal in our discussion on subset selection, so we review it below.

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$, and let $k \in \mathbb{N}$ denote the target rank with $1 \leq k \leq \operatorname{rank}(\mathbf{A})$ and $k < n$. We define the thin singular value decomposition as a factorization $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$, where $\mathbf{U} \in \mathbb{R}^{m \times n}$ has orthonormal columns, $\mathbf{V} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix, and $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with diagonal elements ordered as

$$\sigma_1(\mathbf{A}) \geq \sigma_2(\mathbf{A}) \geq \ldots \geq \sigma_n(\mathbf{A}) \geq 0.$$

We partition the SVD of $\mathbf{A}$ as

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_k & \mathbf{U}_\perp \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_k & \\ & \mathbf{\Sigma}_\perp \end{bmatrix}, \quad \text{and} \quad \mathbf{V} = \begin{bmatrix} \mathbf{V}_k & \mathbf{V}_\perp \end{bmatrix}, \tag{2.30}$$

where $\mathbf{U}_k \in \mathbb{R}^{m \times k}$ and $\mathbf{V}_k \in \mathbb{R}^{n \times k}$ have orthonormal columns and $\mathbf{\Sigma}_k \in \mathbb{R}^{k \times k}$ is diagonal.

The partitioning in (2.30) is useful for generating low-rank approximations of $\mathbf{A}$. We define a rank-k truncated SVD of $\mathbf{A}$ as

$$\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{T}}.$$

The Eckart-Young theorem (Golub and Van Loan 2013, Theorem 2.4.8) states that $\mathbf{A}_k$ is a best rank k approximation of $\mathbf{A}$ in the spectral and Frobenius norm, satisfying

$$\min_{\substack{\mathbf{B}\in\mathbb{R}^{m\times n}\\ \mathrm{rank}(\mathbf{B})=k}} \|\mathbf{A}-\mathbf{B}\|_2 = \|\mathbf{A}-\mathbf{A}_k\|_2 = \sigma_{k+1}(\mathbf{A}), \quad \text{and} \tag{2.31}$$

$$\min_{\substack{\mathbf{B}\in\mathbb{R}^{m\times n}\\ \mathrm{rank}(\mathbf{B})=k}} \|\mathbf{A}-\mathbf{B}\|_F = \|\mathbf{A}-\mathbf{A}_k\|_F = \sqrt{\sum_{i=k+1}^{n} \sigma_i(\mathbf{A})^2}. \tag{2.32}$$

The columns of $\mathbf{U\Sigma}$ are the discretized temporal modes of the POD, $\{u_i(t)\}$ and columns of $\mathbf{V}$ are the discretized spatial modes, $\{v_i(x)\}$.

2.3 Neural ODEs

Recently, machine learning approaches such as recurrent neural networks and neural ODEs (NODEs) have been used to learn the temporal modes of the POD (Baker et al. 2023; Kani and Elsheikh 2017, 2019). Neural ODEs are a class of continuous-depth neural networks. While traditional deep learning methods use discrete layers to generate predictions, the NODE framework considers this as a continuous process, which allows the use of an ODE solver to adapt the depth of the model to the complexity of the dynamics, rather than relying on a fixed depth. The vanilla NODE is formulated as a first-order ODE for the states $\mathbf{u}(t) \in \mathbb{R}^m$,

$$\frac{d}{dt}\mathbf{u}(t) = \mathbf{f}(\mathbf{u}(t), t, \boldsymbol{\theta}), \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{2.33}$$

where $\mathbf{f}(\mathbf{u}, t, \boldsymbol{\theta}) \in \mathbb{R}^m$ represents a neural network parameterized by learnable parameters $\boldsymbol{\theta}$ (Chen et al. 2018). NODEs and their variants have been widely applied to learn from irregularly-sampled sequential data and are particularly suitable for learning complex dynamical systems (Chen et al. 2018; Kidger et al. 2020; Norcliffe et al. 2020), which are trained by efficient algorithms (Quaglino et al. 2020; Kelly et al. 2020).

To train the NODE with a given loss function $\mathcal{L}$ that depends on the state $\mathbf{u}$, the parameters $\boldsymbol{\theta}$, and training data, we apply the adjoint sensitivity method. First, we solve the ODE (2.33) from $t = 0$ to the terminal time $t = T$ with the initial condition $\mathbf{u}(0) = \mathbf{u}_0$ using a classical ODE solver to solve the ODE. Then, to update the parameters, we compute the

gradient of the loss function $\mathcal{L}$ with respect to $\boldsymbol{\theta}$,

$$\frac{d\mathcal{L}}{d\boldsymbol{\theta}} = \int_0^T \mathbf{a}(t)^{\mathrm{T}} \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}(\mathbf{u}(t), t, \boldsymbol{\theta})\, dt, \tag{2.34}$$

where the adjoint state $\mathbf{a}(t) = \partial\mathcal{L}/\partial\mathbf{u}(t)$ satisfies the backward adjoint equation

$$\frac{d}{dt}\mathbf{a}(t) = -\mathbf{a}(t)^{\mathrm{T}} \frac{\partial}{\partial \mathbf{a}}\mathbf{f}(\mathbf{u}(t), t, \boldsymbol{\theta}), \quad \mathbf{a}(T) = \mathbf{a}_T. \tag{2.35}$$

Once we solve the adjoint equation (2.35) numerically from $t = T$ to $t = 0$ and evaluate the gradient (2.34), we update $\boldsymbol{\theta}$ with a suitable optimizer and repeat this process for a given number of iterations.

Traditional machine learning models, including NODEs, do not generally respect physical properties of the model. This can lead to unphysical or unstable results in predictions, motivating research into invariant-preserving methods. Various neural networks have been developed in order to obey conservation laws, such as Lagrangian Neural Networks (Lutter et al. 2019; Cranmer et al. 2020), Hamiltonian Neural Networks (Greydanus et al. 2019), LieConv (Finzi et al. 2020a), and Deep conservation (Lee and Carlberg 2021). Recently, the CFINDE (Matsubara et al. 2020) incorporates conservation laws given by first integral into NODEs rather than a residual neural network. In addition to conservation laws, researchers have proposed neural networks to learn Lyapunov functions, which play a role in the stability of ODEs, and are expected to be non-increasing, rather than constant (Kolter and Manek 2019; Takeishi and Kawahara 2020; Chu et al. 2024).

2.3.1 Heavy Ball NODE

The classical NODE (2.33) has several weaknesses, such as a potentially high computational cost for both the forward evaluation and solving the adjoint equation (Xia et al. 2021). In addition, the adjoint state $\mathbf{a}(t)$ can vanish as t gets large, making NODEs unreliable for learning long-term dynamics. Many variants of neural ODEs have been developed to improve the NODE. Of these variants, we briefly review the heavy ball neural ODE (HBNODE) proposed in Xia et al. (2021), as a HBNODE-based algorithm was introduced in Baker et al. (2023) to learn the POD of complex dynamics.

The HBNODE is a variant of the vanilla NODE inspired by the heavy ball gradient descent method (Xia et al. 2021). It is formulated as a second-order neural ODE for $\mathbf{u}(t)$,

$$\frac{d^2\mathbf{u}(t)}{dt^2} + \gamma\frac{d\mathbf{u}(t)}{dt} = f(\mathbf{u}(t), t, \boldsymbol{\theta}). \tag{2.36}$$

Here, $\gamma \geq 0$ denotes a damping parameter, which can be a tunable hyperparameter or learned during training. The adjoint equation for the (2.36) is also a HBNODE, satisfying

$$\frac{d^2\mathbf{a}(t)}{dt^2} - \gamma\frac{d\mathbf{a}(t)}{dt} = \mathbf{a}(t)\frac{\partial f}{\partial \mathbf{u}}(\mathbf{u}(t), t, \boldsymbol{\theta}), \tag{2.37}$$

which is solved backwards from $t = T$ to $t = 0$. Letting $\tau = T - t$ and $\mathbf{b}(\tau) = \mathbf{a}(T - \tau)$, we can rewrite the HBNODE adjoint equation as

$$\frac{d^2\mathbf{b}(\tau)}{d\tau^2} - \gamma\frac{d\mathbf{b}(\tau)}{d\tau} = \mathbf{b}(\tau)\frac{\partial f}{\partial \mathbf{u}}(\mathbf{u}(T-\tau), T-\tau, \boldsymbol{\theta}),$$

which is also a HBNODE with the same damping parameter γ. Because of this, solving the adjoint equation will have similar costs to the forward solve, meaning that if the forward pass is accelerated, backwards propagation will be accelerated as well.

The ability to learn long-term dependencies is essential for the success of deep learning models when dealing with sequential data. A well-known bottleneck for training residual neural networks (RNNs) is the vanishing gradient phenomenon. As a continuous analogue of RNNS, NODEs may suffer from this phenomenon in the form of a vanishing adjoint state $\mathbf{a}(t) = \partial\mathcal{L}/\partial\mathbf{u}(t)$. When the vanishing gradient issue occurs, the adjoint state, $\mathbf{a}(t)$, quickly decreases as $T - t$ increases. Consequently, the effect of the adjoint state on $\partial\mathcal{L}/\partial\theta$ becomes negligible for large $T - t$.

Specifically, the expression for the adjoint state of the NODE is

$$\frac{\partial\mathcal{L}}{\partial\mathbf{u}(t)} = \frac{\partial\mathcal{L}}{\partial\mathbf{u}(T)}\frac{\partial\mathbf{u}(T)}{\partial\mathbf{u}(t)} = \frac{\partial\mathcal{L}}{\partial\mathbf{u}(T)}\exp\left(-\int_T^t \frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\mathbf{u}, \tau, \boldsymbol{\theta})d\tau\right).$$

For the HBNODE, with $\mathbf{m}(t) = d\mathbf{u}(t)/dt$, the adjoint state is given by

$$\begin{bmatrix}\frac{\partial\mathcal{L}}{\partial\mathbf{u}(t)} & \frac{\partial\mathcal{L}}{\partial\mathbf{m}(t)}\end{bmatrix} = \begin{bmatrix}\frac{\partial\mathcal{L}}{\partial\mathbf{u}(T)} & \frac{\partial\mathcal{L}}{\partial\mathbf{m}(T)}\end{bmatrix}\begin{bmatrix}\frac{\partial\mathbf{u}(T)}{\partial\mathbf{u}(t)} & \frac{\partial\mathbf{u}(T)}{\partial\mathbf{m}(t)} \\ \frac{\partial\mathbf{m}(T)}{\partial\mathbf{u}(t)} & \frac{\partial\mathbf{m}(T)}{\partial\mathbf{m}(t)}\end{bmatrix} = \frac{\partial\mathcal{L}}{\partial\mathbf{u}(T)}\exp\Bigg(-\underbrace{\int_T^t \begin{bmatrix}\mathbf{0} & \mathbf{I} \\ \frac{\partial f}{\partial\mathbf{h}} & -\gamma\mathbf{I}\end{bmatrix} d\tau}_{:=\mathbf{M}}\Bigg). \tag{2.38}$$

The eigenvalues of $-\mathbf{M}$ can be paired in a way such that the sum of each pair equals $(t-T)\gamma$ (Xia et al. 2021, Proposition 4). Therefore, while the adjoint state of NODEs may vanish as $T - t$ increases, the adjoint state of HBNODEs will not, highlighting the advantages of using HBNODEs for capturing long-range dependencies.

2.3.2 NODE and HBNODE Comparison

The work of Baker et al. (2023) demonstrated that HBNODE significantly outperforms the vanilla NODE in learning the POD of complex dynamics. In this subsection, we perform a numerical experiment to compare the performance of the vanilla NODE and HBNODE in learning the POD of the fiber-coating dynamics governed by PDE (1.1) following the techniques in Baker et al. (2023), and highlight the disadvantages of directly applying NODE or HBNODE to the fiber coating data.

To learn the k dominant temporal modes of the POD, we use a sequence-to-sequence architecture, where an RNN encoder is used to transform the input sequence into the latent states that evolve according to the NODE (2.33) or the HBNODE (2.36). Specifically, this encoder takes as input the temporal POD modes on a training interval from time $t = t_1$ to $t = t_N$, denoted by $\{\mathbf{u}(t_j)\}_{j=1}^{N}$, and outputs a latent sequence $\{\mathbf{w}(t_j)\}_{j=1}^{N}$, where $\mathbf{u} \in \mathbb{R}^k$, $\mathbf{w} \in \mathbb{R}^m$, and $m > k$. The NODE for the latent states $\mathbf{w}$ is defined as

$$\frac{d\mathbf{w}}{dt} = f(\mathbf{w}, \boldsymbol{\theta}), \qquad \mathbf{w}(0) = \mathbf{w}_0, \tag{2.39}$$

and the corresponding HBNODE is given by

$$\frac{d^2\mathbf{w}}{dt^2} + \gamma \frac{d\mathbf{w}}{dt} = f(\mathbf{w}, \boldsymbol{\theta}), \qquad \mathbf{w}(0) = \mathbf{w}_0, \tag{2.40}$$

where γ is a tunable parameter, and $f(\mathbf{w}, \boldsymbol{\theta}) \in \mathbb{R}^m$ is parameterized by a neural network. We use the NODE or HBNODE to advance the latent sequence to obtain the predicted values of the latent sequence $\{\mathbf{w}(t_j)\}_{j=s}^{M}$ from time $t = t_s$ to $t = t_M$, where M is the total number of time snapshots, and $s = M - N + 1$. Finally, an RNN decoder is used to recover the predicted temporal POD modes $\{\hat{\mathbf{u}}(t_j)\}_{j=s}^{M}$ from time $t = t_s$ to $t = t_M$, where $\hat{\mathbf{u}} \in \mathbb{R}^k$.

We define the loss function $\mathcal{L}$ using the mean squared error to measure the error between the labeled data and the prediction,

$$\mathcal{L}(\mathbf{u}, \hat{\mathbf{u}}) = \frac{1}{N \cdot N_s} \sum_{i=1}^{N_s} \sum_{j=s}^{M} \|\mathbf{u}^{(i)}(t_j) - \hat{\mathbf{u}}^{(i)}(t_j)\|_2^2, \tag{2.41}$$

where the superscripts indicate the sample indices, and N_s is the number of samples.

For this experiment, we use the simulation data obtained by numerically solving the fiber coating equation (1.1) over a periodic domain with size $L = 5$. The dimensionless system parameters are set to $\alpha = 2.299$ and $\eta = 0.231$, corresponding to a laboratory experiment

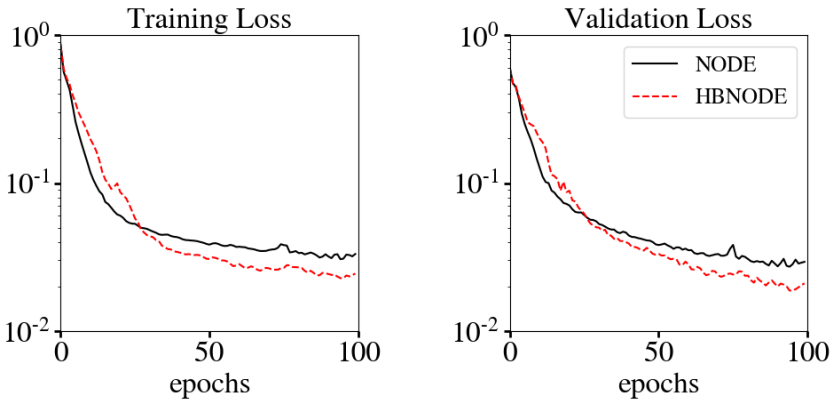

Figure 2.2: A comparison of the mean training and validation loss using the NODE (black) and HBNODE (red dashed) over 5 training processes.

discussed in Ji et al. (2019). We choose the initial data of the form

$$h_0(x) = c_0 + c_1 \sin(\pi x/L) + c_2 \sin(2\pi x/L) + c_3 \sin(4\pi x/L) + c_4 \sin(8\pi x/L), \tag{2.42}$$

where $c_0 = 0.355$, $c_i \in \{0.01, 0.02, 0.03, 0.04\}$ for $i = 1, \ldots, 4$. We select the amplitudes of the sine modes in the initial condition to satisfy $c_1 > \max\{c_2, c_3, c_4\}$ to promote the formation of single-peak droplets. This choice leads to a total of $N_s = 36$ different samples. The PDE (1.1) is numerically solved using second-order centered finite differences with implicit time-stepping on a uniform grid, where the number of grid points is $N_x = 801$, and the terminal time for the simulation is $T = 600$. Starting from each initial condition, after early-stage transients, the numerical solution to (1.1) typically evolves into a single-peak wave moving at a constant speed to the right. We select the temporal domain for the learning task to be from $t = 150$ to $t = 600$, with $M = 91$ uniformly spaced time snapshots. For each sample, we organize the simulation results into a data matrix $\mathbf{A} \in \mathbb{R}^{91\times 801}$, where the j^{th} row of $\mathbf{A}$ corresponds to the j^{th} time snapshot of the sample for $1 \le j \le 91$.

We then compute the POD for the single-peak dataset over the time interval from $t = 150$ to 600 over which the droplet dynamics is close to the traveling wave regime. The temporal modes for these flows oscillate quasi-periodically and the singular values rapidly decay. The rank-k POD approximation with $k = 4$ models gives an average relative error across all samples of

$$\frac{\|\mathbf{A} - \mathbf{A}_k\|_F}{\|\mathbf{A}\|_F} \approx 0.11,$$

where $\mathbf{A}_k$ is the rank-k truncated SVD of $\mathbf{A}$. We select $k = 4$ as the number of POD modes

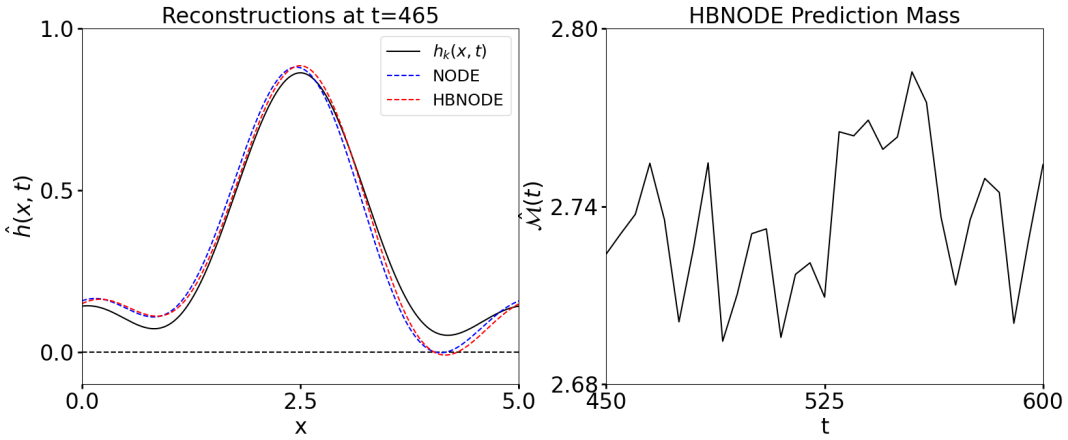

Figure 2.3: (Left) A comparison of the reconstructions of a time snapshot of the NODE and HBNODE. Both reconstructions fall slightly below the line $\hat{h}(x,t) = 0$, yielding an unphysical reconstruction. (Right) The mass of the reconstructed solution $\hat{h}(x,t)$ generated by the HBNODE over the prediction times.

for this experiment.

We randomly divide samples into $N_r = 30$ training samples and $N_v = 6$ test samples, using $t \in [150, 445]$ as the training set for each sample to predict values for $t \in [450, 600]$. This corresponds to using $N = 60$ for the input sequence $\{\mathbf{u}(t_j)\}_{j=1}^{N}$ from $t = t_1$ to $t = t_N$. For the design of the RNN encoder and decoder, we set $m = 4k = 16$ for the transformation between the temporal modes $\mathbf{u} \in \mathbb{R}^k$ and the latent states $\mathbf{w} \in \mathbb{R}^m$. The RNN encoder consists of three fully connected layers with $8k = 32$ nodes in each hidden layer, each followed by an activation function $\tanh(x)$. The decoder consists of a single fully connected layer that transforms the latent state $\mathbf{w} \in \mathbb{R}^m$ to the predicted temporal modes $\hat{\mathbf{u}} \in \mathbb{R}^k$. For the function $f(\mathbf{w}, \boldsymbol{\theta})$ on the right-hand side of (2.39) and (2.40), we use a single hidden layer with $4k$ nodes followed by a sigmoid activation function. We use an AdamW optimizer to train the network with a batch size of 6 and iterate over 100 epochs. The blackbox integrator is selected to be DOPRI-5 (Dormand and Prince 1980) with a relative tolerance of 10^{-6}.

Figure 2.2 compares the performance of NODE and HBNODE on the single-peak dataset over 5 training processes with different initializations. The training loss is the loss function (2.41) averaged over only the training samples, defined as

$$\mathcal{L}_r(\mathbf{u}, \hat{\mathbf{u}}) = \frac{1}{N \cdot N_r} \sum_{i=1}^{N_r} \sum_{j=s}^{M} \|\mathbf{u}^{(i)}(t_j) - \hat{\mathbf{u}}^{(i)}(t_j)\|_2^2,$$

and similarly the validation loss is loss function (2.41) averaged over only the validation

samples, given as

$$\mathcal{L}_v(\mathbf{u}, \hat{\mathbf{u}}) = \frac{1}{N \cdot N_v} \sum_{i=N_r+1}^{N_s} \sum_{j=s}^{M} \|\mathbf{u}^{(i)}(t_j) - \hat{\mathbf{u}}^{(i)}(t_j)\|_2^2.$$

This figure shows that in both training and validation, the HBNODE has a smaller loss than the NODE. This is consistent with the observations in Baker et al. (2023), which demonstrated that the HBNODE outperforms the vanilla NODE in various sequential learning tasks. Figure 2.3 (left) presents the reconstructions of the NODE and HBNODE solutions of a test sample at $t = 465$. In this particular snapshot, the difference in accuracy between the NODE and HBNODE reconstructions is not pronounced. However, both snapshots produce negative reconstructions at around $x = 4$, which is a physically unrealistic prediction. Furthermore, we consider the mass of the HBNODE reconstruction, defined as

$$\hat{\mathcal{M}}(t) = \int_0^L \hat{h}(x,t) + \frac{\alpha}{2}\hat{h}(x,t)^2 dx.$$

Figure 2.3 (right) shows the mass of the HBNODE reconstruction of this sample over the prediction times. In the true solution, $h(x,t)$, this mass is conserved, but in the reconstruction, the mass varies from about 2.694 to 2.785. This is yet another unphysical observation.

While the NODE and HBNODE are able to accurately approximate fiber coating dynamics, their reconstructions can become negative and do not conserve mass. In Chapter 3, we propose physics-constrained NODEs, which is a modification to the NODE framework that enforces physical constraints into the reconstructions.

2.4 Rank-Revealing QR Factorizations

Next, we discuss the Rank-Revealing QR (RRQR) factorization and define the strong RRQR and the principal angles between subspaces, which will be useful in the analysis of randomized subset selection algorithms in Chapter 4.

2.4.1 The Rank-Revealing QR Factorization

In this section, we define the Rank-Revealing QRs (RRQRs) and go over some important properties of RRQRs. Let all the assumptions in Section 2.2 hold and let $\mathbf{\Pi} \in \mathbb{R}^{n \times n}$ be a permutation matrix. Then we partition $\mathbf{A\Pi}$ as

$$\mathbf{A}\begin{bmatrix}\mathbf{\Pi}_1 & \mathbf{\Pi}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}, \tag{2.43}$$

where $\mathbf{\Pi}_1 \in \mathbb{R}^{n\times k}$, $\mathbf{\Pi}_2 \in \mathbb{R}^{n\times(n-k)}$, $\mathbf{A}_1 \in \mathbb{R}^{m\times k}$, and $\mathbf{A}_2 \in \mathbb{R}^{m\times(n-k)}$. Instead of working with $\mathbf{A}_1$ and $\mathbf{A}_2$ directly, it is often easier to consider the (thin) QR factorization $\mathbf{A\Pi} = \mathbf{QR}$, partitioned as

$$\mathbf{A}\begin{bmatrix}\mathbf{\Pi}_1 & \mathbf{\Pi}_2\end{bmatrix} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{R}_{11} & \mathbf{R}_{12}\\ \mathbf{0} & \mathbf{R}_{22}\end{bmatrix}, \tag{2.44}$$

where $\mathbf{Q} \in \mathbb{R}^{m\times n}$ has orthonormal columns, $\mathbf{R} \in \mathbb{R}^{n\times n}$ is upper-triangular, $\mathbf{Q}_1 \in \mathbb{R}^{m\times k}$, $\mathbf{Q}_2 \in \mathbb{R}^{m\times(n-k)}$, $\mathbf{R}_{11} \in \mathbb{R}^{k\times k}$, and $\mathbf{R}_{22} \in \mathbb{R}^{(n-k)\times(n-k)}$. By the orthogonal invariance of singular values, the singular values of $\mathbf{A}$ are equal to the singular values of $\mathbf{R}$. As a result, the interlacing property of singular values (Horn and Johnson 1991, Corollary 3.1.3) gives

$$\sigma_i(\mathbf{R}_{11}) \le \sigma_i(\mathbf{A}) \qquad 1 \le i \le k \tag{2.45}$$

$$\sigma_{k+j}(\mathbf{A}) \le \sigma_j(\mathbf{R}_{22}) \qquad 1 \le j \le n-k. \tag{2.46}$$

Furthermore, we call this QR factorization (2.44) of $\mathbf{A\Pi}$ a rank-revealing QR factorization (RRQR) of $\mathbf{A}$ if there exists low-degree polynomials $p_1(n,k)$ and $p_2(n,k)$ such that the following inequalities hold:

$$\frac{\sigma_i(\mathbf{A})}{p_1(n,k)} \le \sigma_i(\mathbf{R}_{11}) \qquad 1 \le i \le k \tag{2.47}$$

$$\sigma_j(\mathbf{R}_{22}) \le \sigma_{k+j}(\mathbf{A})p_2(n,k) \qquad 1 \le j \le n-k. \tag{2.48}$$

If we combine the inequalities given by the interlacing property of singular values, (2.45) and (2.46), which hold for any QR decomposition, and the inequalities that the RRQR satisfies, (2.47) and (2.48), then for any RRQR, we have

$$\frac{\sigma_i(\mathbf{A})}{p_1(n,k)} \le \sigma_i(\mathbf{R}_{11}) \le \sigma_i(\mathbf{A}) \qquad 1 \le i \le k$$

$$\sigma_{k+j}(\mathbf{A}) \le \sigma_j(\mathbf{R}_{22}) \le \sigma_{k+j}(\mathbf{A})p_2(n,k) \qquad 1 \le j \le n-k.$$

This tells us the singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ can be bounded above and below by the singular values of $\mathbf{A}$ up to a multiplicative factor of $p_1(n,k)$ or $p_2(n,k)$. If p_1 and p_2 are relatively small, then the singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ give us information about the singular values of $\mathbf{A}$.

In addition, if $\mathbf{A}_1$ is full rank, (in other words, if $\mathbf{\Pi}_1$ selects linearly independent columns

of $\mathbf{A}$), then (Broadbent et al. 2010, (2.2)) states

$$\sigma_i(\mathbf{A}_1) = \sigma_i(\mathbf{R}_{11}) \qquad 1 \leq i \leq k, \tag{2.49}$$

$$\sigma_j\left(\left(\mathbf{I} - \mathbf{A}_1\mathbf{A}_1^{\dagger}\right)\mathbf{A}_2\right) = \sigma_j(\mathbf{R}_{22}) \qquad 1 \leq j \leq n-k. \tag{2.50}$$

This means that the singular values of the submatrices $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ are the singular values of $\mathbf{A}_1$ and $\left(\mathbf{I} - \mathbf{A}_1\mathbf{A}_1^{\dagger}\right)\mathbf{A}_2$, the projection of $\mathbf{A}_2$ onto the null space of $\mathbf{A}_1$, giving us information about the set of columns that $\mathbf{\Pi}_1$ selects. Therefore, RRQR algorithms can be used to perform subset selection by providing a method to select a set of linearly independent columns of $\mathbf{A}$ (Chan and Hansen 1992).

We call a RRQR strong (Gu and Eisenstat 1996, §1.2) if for a parameter $f \geq 1$, we can bound the singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ as follows:

$$\frac{\sigma_i(\mathbf{A})}{\beta(n,k)} \leq \sigma_i(\mathbf{R}_{11}) \leq \sigma_i(\mathbf{A}) \qquad 1 \leq i \leq k$$

$$\sigma_{k+j}(\mathbf{A}) \leq \sigma_j(\mathbf{R}_{22}) \leq \beta(n,k)\sigma_{j+k}(\mathbf{A}) \qquad 1 \leq j \leq n-k,$$

where $\beta(n,k) = \sqrt{1 + f^2k(n-k)}$. If f is chosen to be a small power of n, there are algorithms to compute a strong RRQR requiring $\mathcal{O}(mn^2)$ flops (Gu and Eisenstat 1996, §1.2).

The strong RRQR algorithm can be modified for matrices in $\mathbb{R}^{k\times n}$. In this case, the factorization is

$$\mathbf{A\Pi} = \mathbf{Q}\begin{bmatrix}\mathbf{R}_{11} & \mathbf{R}_{12}\end{bmatrix}, \tag{2.51}$$

where $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ is a permutation matrix, $\mathbf{Q} \in \mathbb{R}^{k\times k}$ is an orthogonal matrix, $\mathbf{R}_{11} \in \mathbb{R}^{k\times k}$ is upper triangular, $\mathbf{R}_{12} \in \mathbb{R}^{k\times(n-k)}$ and the following inequality holds

$$\frac{\sigma_i(\mathbf{A})}{\beta(n,k)} \leq \sigma_i(\mathbf{R}_{11}) \leq \sigma_i(\mathbf{A}) \quad 1 \leq i \leq k. \tag{2.52}$$

The following inequality describes the conditioning of the columns selected by a strong RRQR applied to a matrix with orthonormal rows.

Proposition 1. *Let $\mathbf{W} \in \mathbb{R}^{n\times k}$ have orthonormal columns and let $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi} = \mathbf{QR}$ be computed using a strong RRQR of $\mathbf{W}^{\mathrm{T}}$ with parameter $f \geq 1$. Then $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular and*

$$\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2 \leq \beta(n,k).$$

2.4.2 Principal Angles Between Subspaces

The distance between two subspaces can be measured by the principal angles between them. These angles can be computed by using matrices whose columns form orthonormal bases for the compared subspaces. Let $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times k}$ have orthonormal columns. Then $\mathcal{X} := \mathcal{R}(\mathbf{X})$ and $\mathcal{Y} := \mathcal{R}(\mathbf{Y})$ are k-dimensional subspaces of $\mathbb{R}^n$. We denote by $\mathbf{\Phi}(\mathcal{X}, \mathcal{Y})$, the vector of principal angles between $\mathcal{X}$ and $\mathcal{Y}$ in non-decreasing order, with $\varphi_i(\mathcal{X}, \mathcal{Y})$ representing the i^{th} principal angle for $1 \leq i \leq k$. It follows from the expressions in (Zhu and Knyazev 2012, Theorem 2.1 and Property 2.1) that

$$\cos(\mathbf{\Phi}(\mathcal{X}, \mathcal{Y})) = \begin{bmatrix} \sigma_1(\mathbf{X}^T\mathbf{Y}) \\ \vdots \\ \sigma_k(\mathbf{X}^T\mathbf{Y}) \end{bmatrix} \quad \text{and} \quad \sin(\mathbf{\Phi}(\mathcal{X}, \mathcal{Y})) = \begin{bmatrix} \sigma_k(\mathbf{X}\mathbf{X}^T(\mathbf{I} - \mathbf{Y}\mathbf{Y}^T)) \\ \vdots \\ \sigma_1(\mathbf{X}\mathbf{X}^T(\mathbf{I} - \mathbf{Y}\mathbf{Y}^T)) \end{bmatrix}.$$

If $\mathbf{X}^{\mathrm{T}}\mathbf{Y}$ is nonsingular, this gives the following expressions for the largest principal angles:

$$\cos(\varphi_k(\mathcal{X}, \mathcal{Y})) = \frac{1}{\|(\mathbf{X}^T\mathbf{Y})^{-1}\|_2}, \tag{2.53}$$

$$\sin(\varphi_k(\mathcal{X}, \mathcal{Y})) = \|\mathbf{X}\mathbf{X}^T(\mathbf{I} - \mathbf{Y}\mathbf{Y}^T)\|_2. \tag{2.54}$$

CHAPTER

3

PHYSICS-CONSTRAINED NEURAL ODES

In this chapter, we present physics-constrained neural ordinary differential equations (NODEs) for learning POD representations of PDE solutions of the fiber coating model. These NODEs are designed to learn the dynamics of the given data while respecting the underlying properties of the physics-based PDE model that generates the data.

We begin by recalling the fiber coating model derived in Chapter 2.1. Let $h(x,t)$ be the dimensionless film thickness of an axisymmetric thin liquid flowing down a vertical cylindrical fiber. The dynamics of $h(x,t)$ are governed by the fourth-order nonlinear degenerate parabolic equation

$$\frac{\partial}{\partial t}\left(h+\frac{\alpha}{2}h^2\right)+\frac{\partial}{\partial x}\left[M(h)\left(1-\frac{\partial}{\partial x}\left[Z(h)-\frac{\partial^2 h}{\partial x^2}\right]\right)\right]=0, \tag{3.1}$$

where the mobility function $M(h)$ and the azimuthal curvature $Z(h)$ take the form

$$M(h)=\frac{h^3\phi(\alpha h)}{3\phi(\alpha)}, \quad Z(h)=\frac{\alpha}{\eta(1+\alpha h)}, \tag{3.2}$$

where $\alpha=\mathcal{H}/R$ is the aspect ratio between the characteristic film thickness $\mathcal{H}$ and the fiber radius R, $\eta=(\mathcal{H}/\mathcal{L})^2$, $\mathcal{L}$ is the characteristic length scale, and the shape factor function

$\phi(X)$ is given by

$$\phi(X) = \frac{3}{16X^3}\left[(1+X)^4(4\ln(1+X)-3)+4(1+X)^2-1\right]. \tag{3.3}$$

Instead of directly working with simulation data from the PDE model (3.1), we exploit the low rank structure of the data and consider the k-dimensional POD approximation of the solution $h(x,t)$,

$$h(x,t) \approx \sum_{i=1}^{k} u_i(t)v_i(x), \tag{3.4}$$

where k is the total number of POD modes used, and $\{u_i(t)\}_{i=1}^k$ and $\{v_i(x)\}_{i=1}^k$ are the temporal and spatial modes discussed in Section 2.2. We define $\mathbf{u}(t) = [u_1(t), \cdots, u_k(t)]^{\mathrm{T}} : \mathbb{R}^+ \to \mathbb{R}^k$ as a time-dependent vector containing the temporal modes.

Given temporal modes at discrete times $t = t_i$, $0 \le i \le N-1$, we train a neural ODE,

$$\frac{d}{dt}\hat{\mathbf{u}}(t) = \mathbf{f}(\hat{\mathbf{u}}(t), \boldsymbol{\theta}), \quad \hat{\mathbf{u}}(t_i) = \mathbf{u}(t_i), \quad t_i \le t \le t_{i+1}, \tag{3.5}$$

with a right-hand side function $\mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) \in \mathbb{R}^k$ that learns the dynamics of $\mathbf{u}(t)$. The training strategy will be described in §3.4. Let $\hat{\mathbf{u}} = [\hat{u}_1(t), \cdots, \hat{u}_k(t)]^{\mathrm{T}}$ be the solution to the neural ODE (3.5). We use the solution $\hat{\mathbf{u}}(t)$ to create a reconstruction of the film thickness $h(x,t)$, defined as

$$h(x,t) \approx \hat{h}(x, \hat{\mathbf{u}}(t)) = \sum_{i=1}^{k} \hat{u}_i(t)v_i(x). \tag{3.6}$$

While the vanilla NODE in (3.5) can be trained to provide accurate reconstructions by minimizing a given loss function, it often can provide unphysical reconstructions, such as predicting a negative $h(x,t)$ or not conserving the volume of the fluid. In order to provide more physical reconstructions, we need to modify the right-hand side of the NODE to ensure it respects important properties of the model (3.1).

The structure of this chapter is as follows. In §3.1, we discuss important properties of the fiber coating model we want to preserve, namely the conservation of mass and bounded entropy constraint. In Section 3.2, we prove that enforcing a bounded entropy constraint yields positive solution reconstructions. Then, in §3.3, we derive physics-constrained NODEs that uphold the conservation of mass and bounded entropy. In §3.4, we discuss the data preparation and the implementation of physics-constrained NODEs for our experiments, and finally, in §3.5 we present results that demonstrate the accuracy of the physics-constrained NODE reconstructions that preserve key solution properties.

3.1 Properties of the Fiber Coating Model

In this subsection, we describe some properties of the fiber coating model (3.1). Specifically, we focus on the Rayleigh-Plateau regime, where the flow is characterized by traveling droplets, and we only consider the flow dynamics away from the inlet. This allows us to equip the PDE with periodic boundary conditions, as the flow near the inlet depends on the inlet geometry (Ji et al. 2020) and is not spatially periodic.

We assume there exists a solution $h \in C^4(L_T)$ for the problem (3.1), where L_T is the space $[0, L] \times [0, T]$. Over a periodic domain $0 \leq x \leq L$, the solution $h(x, t)$ obeys the conservation of mass, satisfying

$$\mathcal{M}(t) = \mathcal{M}(0), \qquad \text{where } \mathcal{M}(t) := \int_0^L \left(h + \frac{\alpha}{2}h^2\right) dx, \tag{3.7}$$

where $\mathcal{M}(t)$ represents the total volume of liquid over the cylindrical substrate at time t, and $\mathcal{M}(0)$ is the mass of the initial data $h_0(x) = h(x, 0)$.

Following the work in Kim et al. (2024), we define the entropy $\mathcal{E}(t)$ of the solution of the PDE (3.1) at time t as

$$\mathcal{E}(t) := \int_0^L G(h(x, t))dx, \tag{3.8}$$

where $G(h)$ is defined via its derivative,

$$G'(h) = (1 + \alpha h) \int_A^h \frac{1}{M(s)} ds, \tag{3.9}$$

where $A > 0$ is a positive constant that ensures that the integral is well-defined. Calculating the time derivative of $\int_0^L G(h)\ dx$ yields

$$\frac{d}{dt}\int_0^L G(h)dx = \int_0^L G'(h)h_t\ dx = \int_0^L h_x \left[1 - (Z(h) - h_{xx})_x\right]\ dx. \tag{3.10}$$

Applying integration by parts with periodic boundary conditions we obtain

$$\frac{d}{dt}\int_0^L G(h)dx = \int_0^L -h_{xx}^2 + h_{xx}Z(h)\ dx = \int_0^L -\left(h_{xx} - \tfrac{1}{2}Z(h)\right)^2 + \left(\tfrac{1}{2}Z(h)\right)^2\ dx. \tag{3.11}$$

This leads to

$$\frac{d}{dt}\int_0^L G(h)dx \leq \int_0^L \left(\frac{Z(h)}{2}\right)^2\ dx, \tag{3.12}$$

which, after integrating over time, yields the entropy estimate of the PDE (3.1),

$$\mathcal{E}(t) = \int_0^L G(h(x,t))\ dx \leq \mathcal{E}(0) + \int_0^t \int_0^L \left(\frac{Z(h(x,s))}{2}\right)^2\ dx\ ds. \tag{3.13}$$

The investigation of such entropy properties can be traced back to earlier works by Bernis and Friedman (1990) on the non-negativity of weak solutions in higher-order degenerate parabolic PDEs. Similar entropy properties have also been used in numerical and analytical works for other lubrication-type equations (Zhornitskaya and Bertozzi 1999; Bertozzi and Pugh 1994). In the context of fiber coating equations, the study on the entropy conditions is limited, with exceptions in the work of Ji et al. (2022), where entropy estimates are utilized to study the existence of weak solutions in a generalized fiber coating equation, and in Kim et al. (2024), where a bounded entropy method was developed in which the mobility functional $M(h)$ was discretized in a specific way to preserve the positivity of numerical solutions even on a coarse grid.

3.2 Positivity-Preserving Reconstructed Solutions

Next, we use the entropy estimate from Section 3.1 to develop a positivity-preserving NODE algorithm for the fiber coating model (3.1). We begin by using the first equality in (3.11) to obtain a formulation for the derivative of the entropy,

$$\frac{d}{dt}\mathcal{E}(t) = \int_0^L h_{xx}Z(h) - h_{xx}^2\ dx. \tag{3.14}$$

In the continuous space, the inequality (3.13) shows that if the derivative of entropy satisfies (3.14), then the entropy has an upper bound.

We consider the domain $[0, L]$ divided into N_x equally spaced grid points with grid size $\Delta x = L/N_x$. Let $\hat{h}(x_i, t)$ be a semi-discretized solution, reconstructed by (3.6), which is discrete in space at the grid point x_i and continuous in time t. We use a centered finite difference to compute the discrete analogue of the second-order space derivative. Following convention in finite differences, we define the second-order centered finite difference of h as

$$\hat{h}_{\bar{x}x}(x_i, t) = \frac{1}{(\Delta x)^2}\left[\hat{h}(x_{i+1}, t) - 2\hat{h}(x_i, t) + \hat{h}(x_{i-1}, t)\right] \tag{3.15}$$

In the following theorem, we show that if the reconstructed solution $\hat{h}$ follows the discretized version of (3.14), then we can bound the discretized entropy estimate, given by $\hat{\mathcal{E}}(t)$ from above. Furthermore, this allows us to bound the reconstructed solution $\hat{h}$ from below by a

positive constant.

Theorem 1. *Let $\hat{h}(x_i,t)$ be the semi-discretized reconstructed solution defined in* (3.6) *discretized in space at time $t>0$, $\hat{h}(x_i,0)>0$ for $1\le i\le N_x$, and the mobility functional $M(\hat{h})=\mathcal{O}(\hat{h}^3)\ge 0$. If the following discretized entropy condition is satisfied,*

$$\frac{d}{dt}\hat{\mathcal{E}}=\sum_{i=1}^{N_x}\left(Z[\hat{h}(x_i,t)]\hat{h}_{\bar{x}x}(x_i,t)-\hat{h}^2_{\bar{x}x}(x_i,t)\right)\Delta x, \tag{3.16}$$

where $\hat{h}_{\bar{x}x}$ is the second-order centered finite difference approximation defined in (3.15). *Then at any time $T>0$, there exists $\delta>0$ such that $\hat{h}(x_i,T)\ge\delta>0$.*

Proof. The proof of the theorem follows the argument in Kim et al. (2024).

Since $\hat{h}(x_i,0)>0$ and $M(\hat{h}(x_i,0))\ge 0$ for $1\le i\le N_x$, it holds from (3.9) that $\hat{\mathcal{E}}(0)$ is well defined, and we let $\hat{\mathcal{E}}(0)=C_1$. By the assumption of the discretized entropy condition (3.16) , a similar derivation to (3.11) and (3.12) gives

$$\frac{d}{dt}\hat{\mathcal{E}}=\sum_{i=1}^{N_x}\left(Z[\hat{h}(x_i,t)]\hat{h}_{\bar{x}x}(x_i,t)-\hat{h}^2_{\bar{x}x}(x_i,t)\right)\Delta x\le\sum_{i=1}^{N_x}\left(\frac{Z(\hat{h}(x_i,t))}{2}\right)^2\Delta x.$$

Furthermore, we have

$$\left|Z(\hat{h}(x_i,t))\right|=\left|\frac{\alpha}{\eta(1+\alpha\hat{h}(x_i,t))}\right|\le\frac{\alpha}{\eta},\qquad 1\le i\le N_x,$$

which means $Z(\hat{h}(x_i,t))^2$ is bounded and we let

$$\frac{d}{dt}\hat{\mathcal{E}}\le\sum_{i=1}^{N_x}\left(\frac{Z(\hat{h}(x_i,t))}{2}\right)^2\Delta x\le C_2.$$

Since $\hat{\mathcal{E}}(0)=C_1$ and $\dfrac{d}{dt}\hat{\mathcal{E}}\le C_2$, for $T>0$, we have

$$\hat{\mathcal{E}}(T)\le\hat{\mathcal{E}}(0)+C_2T=C_1+C_2T=C<\infty.$$

A direct computation using $M(\hat{h})=\mathcal{O}(\hat{h}^3)$ shows

$$G(\hat{h}(x_i,T))=\frac{1}{2\hat{h}(x_i,T)}+\frac{\alpha}{2}\ln\left(\hat{h}(x_i,T)\right)+\mathcal{O}(1),\qquad 1\le i\le N_x.$$

Let $\delta = \min_{1\le i\le N_x} \hat{h}(x_i, T)$. We assume that $\delta \to 0$. It follows that $\lim_{\delta\to 0^+} G(\delta) = \infty$ and

$$\lim_{\delta\to 0^+} \hat{\mathcal{E}}(T) = \lim_{\delta\to 0^+} \sum_{i=1}^{N_x} G(\hat{h}(x_i, T))\Delta x \ge \lim_{\delta\to 0^+} G(\delta)\Delta x = \infty.$$

This contradicts $\hat{\mathcal{E}}(T) \le C$. Thus, $\delta \not\to 0$ and $\delta = \min_{1\le i\le N_x} \hat{h}(x_i, T) > 0$.

□

The existence of a lower bound $\delta = \min_{1\le i\le N_x} \hat{h}(x_i, T) > 0$ implies that $\hat{h}(x_i, t) > 0$ for $1 \le i \le N_x$ and for all t. Thus, if we can create a NODE model such that the reconstruction satisfies the condition (3.16), we guarantee the NODE reconstruction $\hat{h}$ will be positive everywhere.

3.3 Physics-Constrained NODEs

In this subsection, we present the formulation of Physics-Constrained NODEs for learning POD representations of PDE solutions from the fiber coating model (3.1). Following earlier works on enforcing conserved quantities in reduced-order solutions and neural differential equations (Anderson and Farazmand 2022; Hilliard and Farazmand 2024; Matsubara and Yaguchi 2023), our approach enforces the conservation of mass (3.7) and the boundedness of the entropy (3.13) in the reduced-order solutions.

3.3.1 MECNODE Derivation

We follow the approaches in Hilliard and Farazmand (2024) and Matsubara et al. (2020) to derive an ODE system for the temporal modes $\mathbf{u}(t)$ that enforces the conservation of mass (3.7) and discretized entropy condition (3.16). We consider the following optimization problem,

$$\min_{\mathbf{u}_t\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{u}_t - \mathbf{f}(\mathbf{u}, \boldsymbol{\theta})\|_2^2, \tag{3.17a}$$

subject to the constraints

$$\hat{\mathcal{M}}(\mathbf{u}) = \mathcal{M}_0, \tag{3.17b}$$

$$\frac{d}{dt}\hat{\mathcal{E}}(\mathbf{u}) = E(\mathbf{u}), \tag{3.17c}$$

where $E(\mathbf{u})$ is given by

$$E(\mathbf{u}) = \sum_{i=1}^{N_x} \left(Z[\hat{h}(x_i,t)]\hat{h}_{\bar{x}x}(x_i,t) - \hat{h}^2_{\bar{x}x}(x_i,t) \right) \Delta x. \tag{3.18}$$

This choice of E ensures that the reconstruction $\hat{h}$ satisfies the assumptions in Theorem 1, and thus guarantees a positive reconstruction. The cost functional $J(\mathbf{u}, \mathbf{u}_t) = \frac{1}{2}\|\mathbf{u}_t - \mathbf{f}(\mathbf{u}, \boldsymbol{\theta})\|_2^2$ quantifies the instantaneous error between the time derivative $\mathbf{u}_t$ and the NODE representation $\mathbf{f}(\mathbf{u}, \boldsymbol{\theta})$. At each time step, the NODE evolves the states $\mathbf{u}$ in the direction that minimizes the cost functional J subject to the constraints (3.17b) and (3.17c).

The conservation of mass (3.7) and the bounded entropy condition (3.13) are enforced at the reconstructed reduced-order solution level, where the reconstructed mass $\hat{\mathcal{M}}$ and entropy functional $\hat{\mathcal{E}}$ are given by

$$\hat{\mathcal{M}}(\mathbf{u}) = \mathcal{M}(\hat{h}), \quad \hat{\mathcal{E}}(\mathbf{u}) = \mathcal{E}(\hat{h}), \tag{3.19}$$

Since $\hat{\mathcal{M}}$ is a conserved quantity in time, the condition (3.17b) leads to

$$\frac{d}{dt}\hat{\mathcal{M}}(\mathbf{u}) = \langle \nabla_{\mathbf{u}}\hat{\mathcal{M}}, \mathbf{u}_t \rangle = 0, \tag{3.20}$$

and similarly, the condition (3.17c) can be represented by

$$\frac{d}{dt}\hat{\mathcal{E}}(\mathbf{u}) = \langle \nabla_{\mathbf{u}}\hat{\mathcal{E}}, \mathbf{u}_t \rangle = E(\mathbf{u}). \tag{3.21}$$

We now combine the constraints (3.20) and (3.21) to form the augmented loss functional with Lagrange multipliers λ_1 and λ_2,

$$\mathcal{J}(\mathbf{u}, \mathbf{u}_t, \lambda_1, \lambda_2) = \frac{1}{2}\|\mathbf{u}_t - \mathbf{f}(\mathbf{u}, \boldsymbol{\theta})\|_2^2 + \lambda_1 \left\langle \nabla_{\mathbf{u}}\hat{\mathcal{M}}, \mathbf{u}_t \right\rangle + \lambda_2 \left(\left\langle \nabla_{\mathbf{u}}\hat{\mathcal{E}}, \mathbf{u}_t \right\rangle - E(\mathbf{u}) \right), \tag{3.22}$$

where $\langle \cdot, \cdot \rangle$ denotes the L_2 inner product. Based on the KKT condition, any minimizer of $\mathcal{J}$ must satisfy

$$0 = \frac{\partial J}{\partial \mathbf{u}_t} = \mathbf{u}_t - f(\mathbf{u}, \boldsymbol{\theta}) + \lambda_1 \nabla_{\mathbf{u}}\hat{\mathcal{M}} + \lambda_2 \nabla_{\mathbf{u}}\hat{\mathcal{E}}, \tag{3.23}$$

$$0 = \frac{\partial J}{\partial \lambda_1} = \left\langle \nabla_{\mathbf{u}}\hat{\mathcal{M}}, \mathbf{u}_t \right\rangle, \tag{3.24}$$

$$0 = \frac{\partial J}{\partial \lambda_2} = \left\langle \nabla_{\mathbf{u}}\hat{\mathcal{E}}, \mathbf{u}_t \right\rangle - E(\mathbf{u}). \tag{3.25}$$

From (3.23), we have $\mathbf{u}_t = \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) - \lambda_1 \nabla_{\mathbf{u}}\hat{\mathcal{M}} - \lambda_2 \nabla_{\mathbf{u}}\hat{\mathcal{E}}$. Substituting this into both (3.24)

and (3.25) and expanding inner products yields a system of equations for the multipliers

$$\langle \nabla_{\mathbf{u}} \hat{\mathcal{M}}, \nabla_{\mathbf{u}} \hat{\mathcal{M}} \rangle \lambda_1 + \langle \nabla_{\mathbf{u}} \hat{\mathcal{M}}, \nabla_{\mathbf{u}} \hat{\mathcal{E}} \rangle \lambda_2 = \langle \nabla_{\mathbf{u}} \hat{\mathcal{M}}, f(\mathbf{u}, \boldsymbol{\theta}) \rangle$$
$$\langle \nabla_{\mathbf{u}} \hat{\mathcal{E}}, \nabla_{\mathbf{u}} \hat{\mathcal{M}} \rangle \lambda_1 + \langle \nabla_{\mathbf{u}} \hat{\mathcal{E}}, \nabla_{\mathbf{u}} \hat{\mathcal{E}} \rangle \lambda_2 = \langle \nabla_{\mathbf{u}} \hat{\mathcal{E}}, \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) \rangle - E(\mathbf{u}),$$

which can be expressed as a linear system $\mathbf{C}\boldsymbol{\lambda} = \mathbf{b}$, where

$$\mathbf{C} = \begin{bmatrix} \langle \nabla_{\mathbf{u}} \hat{\mathcal{M}}, \nabla_{\mathbf{u}} \hat{\mathcal{M}} \rangle & \langle \nabla_{\mathbf{u}} \hat{\mathcal{M}}, \nabla_{\mathbf{u}} \hat{\mathcal{E}} \rangle \\ \langle \nabla_{\mathbf{u}} \hat{\mathcal{E}}, \nabla_{\mathbf{u}} \hat{\mathcal{M}} \rangle & \langle \nabla_{\mathbf{u}} \hat{\mathcal{E}}, \nabla_{\mathbf{u}} \hat{\mathcal{E}} \rangle \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \langle \nabla_{\mathbf{u}} \hat{\mathcal{M}}, \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) \rangle \\ \langle \nabla_{\mathbf{u}} \hat{\mathcal{E}}, \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) \rangle - E(\mathbf{u}) \end{bmatrix}.$$

The matrix $\mathbf{C}$ can be factored as $\mathbf{C} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q}$, where $\mathbf{Q} = \begin{bmatrix} \nabla_{\mathbf{u}} \hat{\mathcal{M}} & \nabla_{\mathbf{u}} \hat{\mathcal{E}} \end{bmatrix} \in \mathbb{R}^{k \times 2}$. We assume that $\nabla_{\mathbf{u}} \hat{\mathcal{M}}$ and $\nabla_{\mathbf{u}} \hat{\mathcal{E}}$ are linearly independent. Therefore, $\mathbf{C}$ is nonsingular and we can solve the system $\mathbf{C}\boldsymbol{\lambda} = \mathbf{b}$ for $\boldsymbol{\lambda}$ to obtain $\boldsymbol{\lambda} = \left(\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\right)^{-1} \mathbf{b}$. From (3.23), we have $\mathbf{u}_t = \mathbf{f} - \mathbf{Q}\boldsymbol{\lambda}$, which when combined with $\boldsymbol{\lambda} = \left(\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\right)^{-1} \mathbf{b}$ and $\mathbf{b} = \mathbf{Q}^{\mathrm{T}}\mathbf{f} - \begin{bmatrix} 0 & E(\mathbf{u}) \end{bmatrix}^{\mathrm{T}}$ yields

$$\mathbf{u}_t = \mathbf{f} - \mathbf{Q} \left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\right)^{-1} \left(\mathbf{Q}^{\mathrm{T}}\mathbf{f} - \begin{bmatrix} 0 \\ E(\mathbf{u}) \end{bmatrix} \right). \tag{3.26}$$

Then we expand the equation (3.26) to get

$$\mathbf{u}_t = \left(\mathbf{I} - \mathbf{Q} \left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\right)^{-1} \mathbf{Q}^{\mathrm{T}} \right) \mathbf{f} + \frac{E(\mathbf{u})}{d} \left((\nabla_{\mathbf{u}} \hat{\mathcal{M}}^{\mathrm{T}} \nabla_{\mathbf{u}} \hat{\mathcal{M}}) \mathbf{I} - \nabla_{\mathbf{u}} \hat{\mathcal{M}} \nabla_{\mathbf{u}} \hat{\mathcal{M}}^{\mathrm{T}} \right) \nabla_{\mathbf{u}} \hat{\mathcal{E}}, \tag{3.27}$$

where $\mathbf{I} \in \mathbb{R}^{k \times k}$ is the identity matrix and

$$d = \det(\mathbf{C}) = \langle \nabla_{\mathbf{u}} \hat{\mathcal{M}}, \nabla_{\mathbf{u}} \hat{\mathcal{M}} \rangle \langle \nabla_{\mathbf{u}} \hat{\mathcal{E}}, \nabla_{\mathbf{u}} \hat{\mathcal{E}} \rangle - \langle \nabla_{\mathbf{u}} \hat{\mathcal{M}}, \nabla_{\mathbf{u}} \hat{\mathcal{E}} \rangle^2.$$

Since $\mathbf{C}$ is nonsingular, $d \neq 0$ and (3.27) is well-defined.

3.3.2 Physics-Constrained NODE models

Now we propose three different physics-constrained NODES for learning fiber coating dynamics. Based on the differential equation (3.27), we propose the Mass and Entropy-Constrained Neural ODE (MECNODE) for the fiber coating dynamics as follows.

Definition 1 (MECNODE)**.** *Let $\hat{\mathcal{M}}$ and $\hat{\mathcal{E}}$ be the mass and entropy as defined in* (3.19)*. Let $\mathbf{Q} = \begin{bmatrix} \nabla_{\mathbf{u}} \hat{\mathcal{M}} & \nabla_{\mathbf{u}} \hat{\mathcal{E}} \end{bmatrix} \in \mathbb{R}^{k \times 2}$ for $k \geq 3$, and $E(\mathbf{u}) > 0$. Assume that $\nabla_{\mathbf{u}} \hat{\mathcal{M}}$ and $\nabla_{\mathbf{u}} \hat{\mathcal{E}}$ are linearly independent. Then the mass and entropy-constrained neural ODE (MECNODE) is*

formulated as an initial value problem

$$\mathbf{u}_t = (\mathbf{I} - \mathbf{P}_\mathbf{Q})\mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) + E(\mathbf{u})\mathbf{Z}\nabla_\mathbf{u}\hat{\mathcal{E}}, \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{3.28}$$

where $\mathbf{f} : \mathbb{R}^k \to \mathbb{R}^k$ *represents a neural network parameterized by trainable parameters* $\boldsymbol{\theta}$*,* $\mathbf{P}_\mathbf{Q} = \mathbf{Q}\left(\mathbf{Q}^\mathrm{T}\mathbf{Q}\right)^{-1}\mathbf{Q}^\mathrm{T} \in \mathbb{R}^{k\times k}$ *is the orthogonal projector onto the range of* $\mathbf{Q}$*, and* $\mathbf{Z}$ *is given by*

$$\mathbf{Z} = \frac{1}{d}\left((\nabla_\mathbf{u}\hat{\mathcal{M}}^\mathrm{T}\nabla_\mathbf{u}\hat{\mathcal{M}})\mathbf{I} - \nabla_\mathbf{u}\hat{\mathcal{M}}\nabla_\mathbf{u}\hat{\mathcal{M}}^\mathrm{T}\right) \in \mathbb{R}^{k\times k},$$

where $d = \langle\nabla_\mathbf{u}\hat{\mathcal{M}}, \nabla_\mathbf{u}\hat{\mathcal{M}}\rangle\langle\nabla_\mathbf{u}\hat{\mathcal{E}}, \nabla_\mathbf{u}\hat{\mathcal{E}}\rangle - \langle\nabla_\mathbf{u}\hat{\mathcal{M}}, \nabla_\mathbf{u}\hat{\mathcal{E}}\rangle^2$*.*

Remark 1. *The vector* $\mathbf{Z}\nabla_\mathbf{u}\hat{\mathcal{E}}$ *can be written as* $\mathbf{y}/\|\mathbf{y}\|_2^2$*, where* $\mathbf{y} \in \mathbb{R}^k$ *is the projection of* $\nabla_\mathbf{u}\hat{\mathcal{E}}$ *onto the null space of* $\nabla_\mathbf{u}\hat{\mathcal{M}}$*, written as*

$$\mathbf{y} = \left(\mathbf{I} - \frac{\nabla_\mathbf{u}\hat{\mathcal{M}}\nabla_\mathbf{u}\hat{\mathcal{M}}^\mathrm{T}}{\nabla_\mathbf{u}\hat{\mathcal{M}}^\mathrm{T}\nabla_\mathbf{u}\hat{\mathcal{M}}}\right)\nabla_\mathbf{u}\hat{\mathcal{E}}.$$

With this substitution, we can equivalently write the MECNODE equation(3.28) *as*

$$\mathbf{u}_t = (\mathbf{I} - \mathbf{P}_\mathbf{Q})\mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) + E(\mathbf{u})\frac{\mathbf{y}}{\|\mathbf{y}\|_2^2}, \quad \mathbf{u}(0) = \mathbf{u}_0.$$

The constrained optimization problem (3.17) may not always have a solution. For a minimizer to exist, the level sets $\{\mathbf{u} \in \mathbb{R}^k \mid \nabla_\mathbf{u}\hat{\mathcal{M}} = 0\}$ and $\{\mathbf{u} \in \mathbb{R}^k \mid \nabla_\mathbf{u}\hat{\mathcal{E}} = E(\mathbf{u})\}$ must have a non-empty intersection. Otherwise, the system is overdetermined, and a minimizer does not exist (Anderson and Farazmand 2022). If the minimizer exists, then the solution satisfies (3.28). In our experiments, provided that the target rank k is greater than the number of constraints (in this case $k > 2$), we have not observed an empty intersection of those level sets. The entropy constraint $\frac{d}{dt}\hat{\mathcal{E}}(\mathbf{u}) = E(\mathbf{u})$ in the MECNODE can be used to ensure that the reconstruction $\hat{h}(x_i, t)$ satisfies the conditions in Theorem 1, thus guaranteeing the positivity of the reconstructed solution $\hat{h}(x_i, t)$.

Next, we present two variants of physics-constrained NODE models, each preserving only one of the properties: conservation of mass or the entropy condition. In the first case, we are only be interested in conserving the mass of the reconstructed solution. Thus, we drop the entropy constraint and arrive at the following optimization problem,

$$\min_{\mathbf{u}_t\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{u}_t - \mathbf{f}(\mathbf{u}, \boldsymbol{\theta})\|_2^2, \tag{3.29}$$

subject to the constraint

$$\hat{\mathcal{M}}(\mathbf{u}) = \mathcal{M}_0. \tag{3.30}$$

Applying a similar process to the derivation of the MECNODE equation (3.27), the optimization problem (3.29) with the constraint (3.30) yields the solution

$$\mathbf{u}_t = \left(\mathbf{I} - \frac{\nabla_{\mathbf{u}} \hat{\mathcal{M}} \nabla_{\mathbf{u}} \hat{\mathcal{M}}^{\mathrm{T}}}{\nabla_{\mathbf{u}} \hat{\mathcal{M}}^{\mathrm{T}} \nabla_{\mathbf{u}} \hat{\mathcal{M}}} \right) \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}).$$

From this formulation, we define the Mass-Constrained NODE as follows.

Definition 2 (MCNODE)**.** *For the optimization problem* (3.29) *with the constraint on the conservation of mass* (3.30)*, we obtain a Mass-Constrained NODE of the form*

$$\mathbf{u}_t = (\mathbf{I} - \mathbf{P}_{\nabla_{\mathbf{u}} \hat{\mathcal{M}}}) \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}), \quad \mathbf{P}_{\nabla_{\mathbf{u}} \hat{\mathcal{M}}} = \frac{\nabla_{\mathbf{u}} \hat{\mathcal{M}} \nabla_{\mathbf{u}} \hat{\mathcal{M}}^{\mathrm{T}}}{\nabla_{\mathbf{u}} \hat{\mathcal{M}}^{\mathrm{T}} \nabla_{\mathbf{u}} \hat{\mathcal{M}}}, \qquad \mathbf{u}(0) = \mathbf{u}_0, \tag{3.31}$$

where $\mathbf{P}_{\nabla_{\mathbf{u}} \hat{\mathcal{M}}}$ *is the orthogonal projector onto the range of* $\nabla_{\mathbf{u}} \hat{\mathcal{M}}$*. If* $\hat{\mathbf{u}}(t)$ *is a solution to the initial value problem* (3.31)*, then the reconstruction* $\hat{h}$ *in* (3.6) *satisfies the condition* $\mathcal{M}(\hat{h}) = \hat{\mathcal{M}}_0$*, where* $\hat{\mathcal{M}}_0 = \hat{\mathcal{M}}(\mathbf{u}_0)$*. We will refer to the mass-constrained NODE* (3.31) *as MCNODE.*

Similar to the MECNODE, we require that the number of POD modes k to satisfy $k \geq 2$ to ensure the optimization problem is well-posed. We note that this formulation is consistent with the continuous FINDE model proposed in the work of Matsubara and Yaguchi (2023). While the continuous FINDE uses a neural ODE to evolve the dynamics, the MCNODE evolves the temporal modes of the POD instead, exploiting the low-rank structure of the data for computational speedup.

Additionally, we consider a variation of the optimization problem (3.17) that only enforces the entropy constraint,

$$\min_{\mathbf{u}_t \in \mathbb{R}^k} \frac{1}{2} \|\mathbf{u}_t - \mathbf{f}(\mathbf{u}, \boldsymbol{\theta})\|_2^2, \tag{3.32}$$

subject to the constraint

$$\frac{d}{dt} \hat{\mathcal{E}}(\mathbf{u}) = E(\mathbf{u}). \tag{3.33}$$

The solution to this optimization problem is

$$\mathbf{u}_t = \left(\mathbf{I} - \frac{\nabla_{\mathbf{u}} \hat{\mathcal{E}} \nabla_{\mathbf{u}} \hat{\mathcal{E}}^{\mathrm{T}}}{\nabla_{\mathbf{u}} \hat{\mathcal{E}}^{\mathrm{T}} \nabla_{\mathbf{u}} \hat{\mathcal{E}}} \right) \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) + \frac{E(\mathbf{u})}{\nabla_{\mathbf{u}} \hat{\mathcal{E}}^{\mathrm{T}} \nabla_{\mathbf{u}} \hat{\mathcal{E}}} \nabla_{\mathbf{u}} \hat{\mathcal{E}}.$$

From this initial value problem, we propose the Entropy-Constrained NODE as follows.

Table 3.1: A table comparing the physical properties enforced by the NODE, MCNODE, ECNODE, and MECNODE.

	Conserves Mass	Positive Reconstruction
NODE (3.5)	X	X
MCNODE (3.31)	✓	X
ECNODE (3.34)	X	✓
MECNODE (3.28)	✓	✓

Definition 3 (ECNODE)**.** *For the optimization problem* (3.32) *with only the constraint on the bounded entropy,* $\frac{d}{dt}\hat{\mathcal{E}}(\mathbf{u}) = E(\mathbf{u})$*, we obtain an entropy-constrained NODE of the form*

$$\mathbf{u}_t = (\mathbf{I} - \mathbf{P}_{\nabla_{\mathbf{u}}\hat{\mathcal{E}}})\mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) + \frac{E}{\nabla_{\mathbf{u}}\hat{\mathcal{E}}^{\mathrm{T}}\nabla_{\mathbf{u}}\hat{\mathcal{E}}}\nabla_{\mathbf{u}}\hat{\mathcal{E}}, \quad \mathbf{P}_{\nabla_{\mathbf{u}}\hat{\mathcal{E}}} = \frac{\nabla_{\mathbf{u}}\hat{\mathcal{E}}\nabla_{\mathbf{u}}\hat{\mathcal{E}}^{\mathrm{T}}}{\nabla_{\mathbf{u}}\hat{\mathcal{E}}^{\mathrm{T}}\nabla_{\mathbf{u}}\hat{\mathcal{E}}}, \qquad \mathbf{u}(0) = \mathbf{u}_0, \quad (3.34)$$

where $\mathbf{P}_{\nabla_{\mathbf{u}}\hat{\mathcal{E}}}$ *is the orthogonal projector onto the range of* $\nabla_{\mathbf{u}}\hat{\mathcal{E}}$*. If* $\hat{\mathbf{u}}(t)$ *is a solution to the initial value problem* (3.34)*, then the reconstruction* $\hat{h}$ *satisfies* $\frac{d}{dt}\mathcal{E}(\hat{h}) = E(\mathbf{u})$*. We will refer to the entropy-constrained NODE* (3.34) *as the ECNODE.*

In order for the optimization problem (3.32) to be well-posed, we need the number of POD modes to exceed the number of constraints, necessitating that $k \geq 2$. Similar to the MECNODE, the ECNODE can be prescribed an $E(\mathbf{u})$ to ensure that the reconstruction $\hat{h}(x_i, t)$ satisfies the conditions in Theorem 1, guaranteeing the positivity of $\hat{h}(x_i, t)$.

We summarize the physical properties each NODE enforces in Table 3.1. The "Conserves Mass" column of the table indicates whether the reconstruction $\hat{h}(x, t)$ of a given model satisfies the conservation of mass (3.7), and the "Positive Reconstruction" column specifies which models guarantee that the reconstruction satisfies $\hat{h}(x, t) > 0$ through the bounded entropy constraint.

3.4 Experimental Set-up

In this subsection, we describe the data generation and processing procedures for the training data (see §3.4.1 – 3.4.2), as well as the implementation details of the neural ODE models used in our experiments (see §3.4.3).

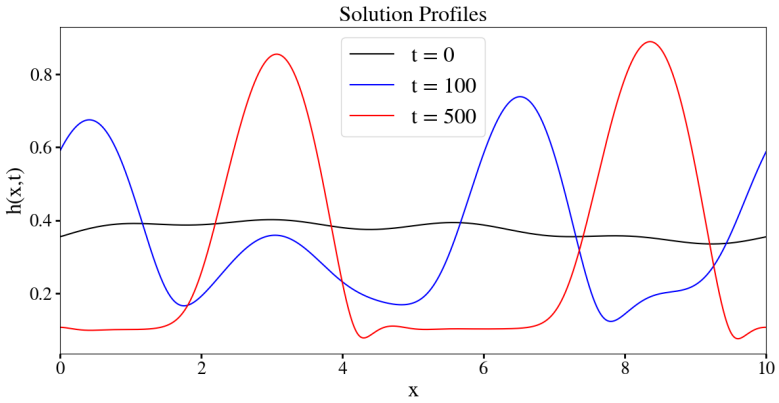

Figure 3.1: Profiles of the solution $h(x,t)$ to the fiber coating model (3.1) starting from nearly flat initial data and evolving into two-peak waves.

3.4.1 Data Preparation

We obtain our training data by numerically solving the fiber coating equation (3.1) subject to periodic boundary conditions. Throughout this section, we set the dimensionless system parameters as $\alpha = 2.299$ and $\eta = 0.23105$. These parameters correspond to a laboratory experiment discussed in Ji et al. (2019). We choose the initial data of the form

$$h_0(x) = c_0 + c_1 \sin(\pi x/L) + c_2 \sin(2\pi x/L) + c_3 \sin(4\pi x/L) + c_4 \sin(8\pi x/L), \tag{3.35}$$

where $c_0 = 0.355$ and c_i takes a value in the set $\{0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04\}$ for $i = 1, \ldots, 4$, giving $7^4 = 2401$ possible initial conditions. To avoid fast droplet coalescence in the two-peak system, we select the amplitudes of the sine modes in the initial condition (3.35) to satisfy $c_1 > \max\{c_2, c_3, c_4\}$. This choice reduces the parameter space to 441 possible initial conditions. Of these possible initial conditions, we randomly sample $N_s = 100$ of them without replacement, which will be further divided randomly into N_r training samples, N_v validations samples, and N_t testing samples.

Due to the long wave instability nature of the fiber coating model, the dynamics of the PDE solution highly depends on the domain size L. We use a domain size of $L = 10$, where the nearly flat initial data gradually evolves into a two-peak moving wave, shown in Figure 3.1. These two-peak systems represent two moving droplets with inter-droplet interactions at a slower timescale.

The fiber coating equation (3.1) is numerically solved for each of our chosen initial conditions using second-order centered finite differences with implicit time-stepping on a

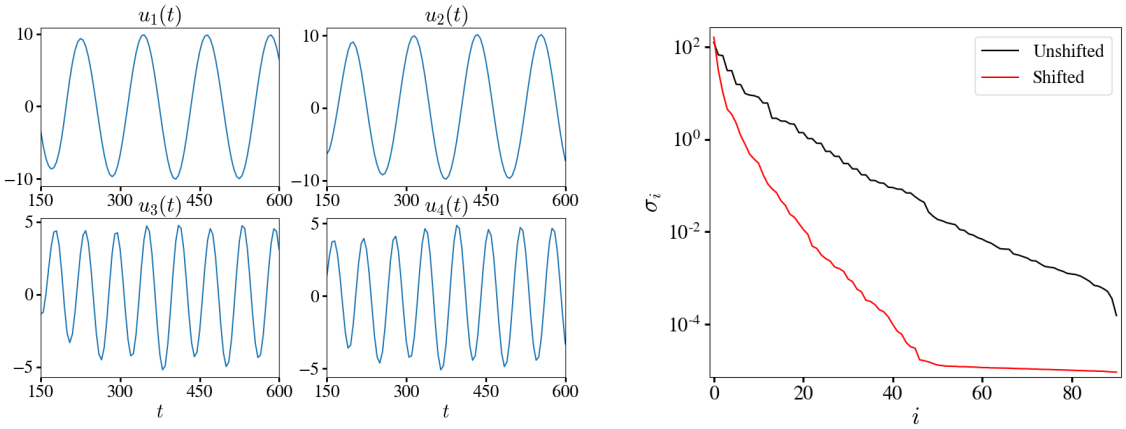

Figure 3.2: (Left) The top four dominant POD modes of a sample in the unshifted two-peak dataset; (Right) the singular value distributions of a sample before (in black) and after (in red) translational symmetry reduction (3.36), labeled as unshifted and shifted. The dots indicate the top k modes included in the POD reconstruction.

uniform grid, setting the number of grid points as $N_x = 1601$ and a terminal time $t_M = 600$. We choose the temporal domain for the learning task to be from $t = 200$ to $t = 600$ with $M = 81$ uniformly spaced time snapshots. For each sample, we put the numerical solution into a data matrix $\mathbf{A} \in \mathbb{R}^{81\times 1601}$, where the j^{th} row of $\mathbf{A}$ corresponds to the j^{th} time snapshot of the sample for $1 \leq j \leq 81$, forming the data matrix

$$\mathbf{A} = \begin{bmatrix} h(x_1,t_1) & h(x_2,t_1) & \dots & h(x_{N_x},t_1) \\ h(x_1,t_2) & h(x_2,t_2) & \dots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ h(x_1,t_M) & \dots & \dots & h(x_{N_x},t_M) \end{bmatrix}.$$

3.4.2 Learning Two-Peak Dynamics with Symmetry Reduction

The next step is to take the POD of each data matrix $\mathbf{A}$ to obtain its temporal and spatial modes. However, training a NODE with a large number of modes is difficult since it often results in an increased system stiffness. Figure 3.2 (left) depicts the first four temporal modes of a sample in the two-peak dataset and the remaining modes involve more high frequency patterns.

To reduce training complexity and to focus more on the inter-droplet interactions, we reduce the translational symmetry in the advection-dominated two-peak dataset by applying

the transformation

$$\tilde{h}(x,t) = h(x - x_c(t), t), \quad \text{where} \quad x_c(t) = \frac{1}{\mathcal{M}} \int_0^L \left(h(x,t) + \frac{\alpha}{2} h(x,t)^2 \right) x \, dx. \tag{3.36}$$

Here, $x_c(t)$ is the x-coordinate of the center of mass, and the shifted solution $\tilde{h}(x,t)$ represents the droplet profiles in a moving reference frame with its center of mass at $x = 0$ for all time. Similar techniques have been used for symmetry reduction in multi-transport equations (Reiss et al. 2018) and turbulent flows (Fedele et al. 2015).

Figure 3.2 (right) depicts a comparison of the singular value distributions of a sample in the two-peak dataset before and after the transformation (3.36). This plot shows that introducing the symmetry reduction yields a more rapid singular value decay, and therefore allows for better POD reconstructions with fewer modes, improving both accuracy and training time of the model. For simplicity, we drop the tilde and will use $h(x,t)$ to represent the transformed solution for the rest of the chapter.

For the POD on the transformed dataset, we elect to use just $k = 4$ modes. This gives an average relative error across all samples of

$$\frac{\|\mathbf{A} - \mathbf{A}_k\|_F}{\|\mathbf{A}\|_F} \approx 0.005,$$

where $\mathbf{A}_k$ is the rank-k truncated SVD of $\mathbf{A}$. The POD reconstruction of the shifted data using $k = 4$ modes yields greater accuracy than the POD reconstruction of the untransformed data using 27 modes. Due to the spatial resolution of the discretization, the center of mass x_c does not necessarily lie on a spatial grid point, and the higher-order temporal modes of the transformed data display highly oscillatory behavior. To amend this, we apply a Gaussian filter over time to each of the temporal modes. This filter smoothens the temporal modes, which makes the NODE system for the temporal modes easier to train without substantially affecting the accuracy of the reconstructions. The average relative error between all samples of the POD reconstruction and the post-filter reconstruction is 0.006, and the average relative error between the true PDE data and the post-filter reconstruction is 0.008.

3.4.3 (MEC)NODE Implementation Details

To implement physics-constrained NODEs discussed in §3.3, we need to use (3.7) and (3.8) to compute the gradients of the reconstructed mass and entropy with respect to the POD

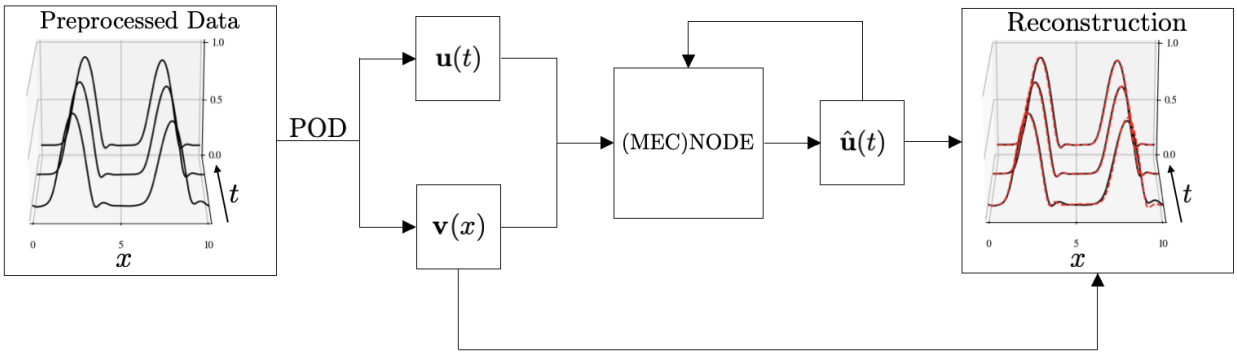

Figure 3.3: A diagram showing the training process of the various NODE models. The POD is used on the preprocessed data to obtain temporal and spatial modes, which are then used as inputs to train the NODE models. After training, the predictions $\hat{\mathbf{u}}$ and spatial modes $\mathbf{v}$ are used to create reconstructions.

modes,

$$\nabla_{\mathbf{u}}\hat{\mathcal{M}} = \alpha\mathbf{u}(t) + \int_0^L \mathbf{v}(x)dx, \quad \nabla_{\mathbf{u}}\hat{\mathcal{E}} = \hat{\mathcal{E}}'(\hat{h})\nabla_{\mathbf{u}}\hat{h} = \int_0^L G'(\hat{h})\mathbf{v}(x)dx, \tag{3.37}$$

where $\mathbf{u} = \begin{bmatrix} u_1 & \dots & u_k \end{bmatrix}^{\mathrm{T}}$ contains the dominant temporal modes, $\mathbf{v} = \begin{bmatrix} v_1 & \dots & v_k \end{bmatrix}^{\mathrm{T}}$ contains the dominant spatial modes, and $G'(\hat{h})$ is given by the definition in (3.9). The formula for $\nabla_{\mathbf{u}}\hat{\mathcal{M}}$ in (3.37) can be easily implemented using the spatial modes $\mathbf{v}$ from the POD, while the formula for $\nabla_{\mathbf{u}}\hat{\mathcal{E}}$ is more complicated and involves fully reconstructing $\hat{h}$ and using it in an integral form. We use the leading order term of a series expression of $G'(h)$ in (3.9) around $h = 0$ to approximate $G'(\hat{h})$ by

$$G'(\hat{h}) = (1 + \alpha\hat{h}) \int_A^{\hat{h}} \frac{1}{M(s)} ds \approx \frac{3}{2}\phi(\alpha)(1 + \alpha\hat{h})(A^{-2} - \hat{h}^{-2}), \tag{3.38}$$

where we choose the parameter $A = 0.1$. For the implementation of (3.38) in the ECNODE and MECNODE, the reconstruction $\hat{h}$ is computed using (3.6).

Since $\nabla_{\mathbf{u}}\hat{\mathcal{M}}$ and $\nabla_{\mathbf{u}}\hat{\mathcal{E}}$ depend on the spatial modes $\mathbf{v}$, the physics-constrained NODEs must also take $\mathbf{v}$ as an input. Figure 3.3 shows the training pipeline. The preprocessed data (after symmetry reduction) is decomposed into its temporal modes $\mathbf{u}(t)$ and spatial modes $\mathbf{v}(x)$, which are then fed into one of the NODE models for $\hat{\mathbf{u}}$, given by

$$\frac{d\hat{\mathbf{u}}}{dt} = \mathbf{F}(\hat{\mathbf{u}}, \mathbf{v}, \boldsymbol{\theta}), \quad \hat{\mathbf{u}}(t_i) = \mathbf{u}(t_i), \quad t_i \le t \le t_{i+1}, \tag{3.39}$$

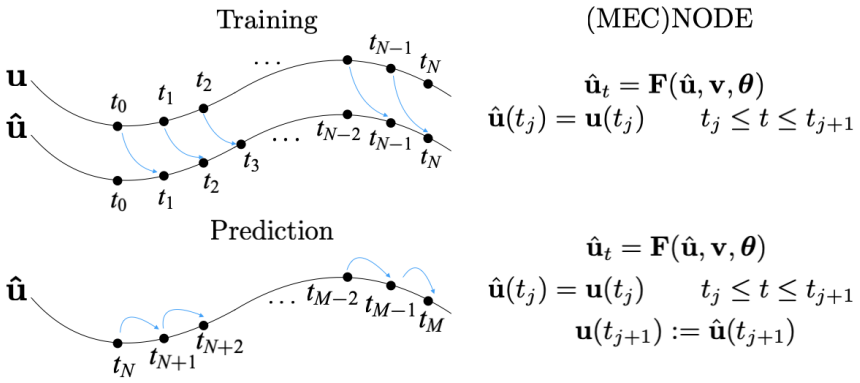

Figure 3.4: A schematic illustrating the training and prediction process of a NODE model (3.39) using a point-to-point evolution for the temporal modes $\hat{\mathbf{u}}_i$. The function $\mathbf{F}$ represents the right-hand side of either a vanilla NODE or a physics-constrained NODE.

where $i = 0, 1, \cdots, N-1$, and $\mathbf{F}$ represents the right-hand side function of the vanilla NODE (3.5) or a physics-constrained NODE, such as the MCNODE (3.31), ECNODE (3.34), or MECNODE (3.28). During training, for each time interval $t_i \le t \le t_{i+1}$, given the ground truth temporal modes $\mathbf{u}(t_i)$ from the preprocessed data, the NODE solution $\hat{\mathbf{u}}$ evolves from the initial state $\mathbf{u}(t_i)$ to reach the prediction $\hat{\mathbf{u}}(t_{i+1})$ at time $t = t_{i+1}$. The loss function $\mathcal{L}$ is defined as the mean squared error between the ground truth data $\mathbf{u}$ and the prediction $\hat{\mathbf{u}}$,

$$\mathcal{L}(\mathbf{u}, \hat{\mathbf{u}}) = \frac{1}{N_s \cdot N \cdot k} \sum_{j=1}^{N} \sum_{\ell=1}^{N_s} \|\mathbf{u}^\ell(t_j) - \hat{\mathbf{u}}^\ell(t_j)\|_2^2, \tag{3.40}$$

where $\mathbf{u}^\ell$ refers to the temporal modes of the ℓ^{th} sample. This piece-wise training strategy allows us to leverage the data from the temporal mode sequence and is different from the sequence-to-sequence architecture used in Baker et al. (2023).

After training is complete, we make predictions for the temporal modes $\hat{\mathbf{u}}$ in time by solving the NODE (3.39) for $t_N \le t \le t_M$, where the initial data is given by $\hat{\mathbf{u}}(t_N) = \mathbf{u}(t_N)$. The predictions $\hat{\mathbf{u}}$ and spatial modes $\mathbf{v}$ are used to create the prediction for the droplet profile $\hat{h}(x, t)$ in time as defined in (3.6). The training and inference processes for learning and predicting the k dominant temporal modes of h are depicted in Figure 3.4.

The function $F(\hat{\mathbf{u}}, \mathbf{v}, \boldsymbol{\theta})$ in the right-hand side of each NODE model is represented by a fully-connected neural network with 3 layers, each containing $16k$ neurons, with the first two layers using a hyperbolic tangent activation function and the last layer using an ELU

activation function. Each sample uses $N = 60$ evenly-spaced time snapshots for $t \in [200, 500]$ as their training set to predict values for $t \in [500, 600]$. We use a Dormand-Prince method (Dormand and Prince 1980) with a relative tolerance of 10^{-6}. We train the model using the training samples for $N_i = 100$ epochs and save the model from the epoch that performs best on the validation samples to use for predictions on the test samples. The NODEs were trained on a CPU with the training time ranging from about 30 minutes to 2.5 hours.

3.5 Results

In this subsection, we present the results of using the proposed physics-constrained NODEs to learn the POD modes of the fiber coating solutions while enforcing mass and entropy conditions. These NODE models are applied to learn the dynamics of two-peak waves in a fiber coating system. We demonstrate the advantages of physics-constrained NODE models in predicting future dynamics in a practical setting.

3.5.1 Comparison of NODE, MCNODE, and ECNODE

We begin by comparing the performance of the vanilla NODE (3.5), MCNODE (3.31), and ECNODE (3.34) with only $k = 2$ modes. This comparison demonstrate the effects of preserving either the mass or entropy condition in learning fiber coating dynamics. The MECNODE model (3.28) is not considered here, as with 2 modes, the optimization problem (3.17) is over-constrained.

To ensure the ECNODE model (3.34) preserves the positivity of the reconstructed solution, the POD approximation of the training data, $h_k(x, t)$, needs to satisfy the positivity condition (see Theorem 1). However, for the preprocessed training samples discussed in Section 3.4.1, the $2-$mode POD reconstructions for 5 out of the 100 training samples yield a reconstruction $\hat{h}_2(x, t)$ that becomes negative at a critical location x for some time t. Consequently, we remove these samples for this experiment and use $N_r = 60$ training samples, $N_v = 20$ validation samples and $N_t = 15$ testing samples.

In Figure 3.5, we show typical time snapshots of reconstructed solutions (dashed curves) from a test sample predicted by the NODE, MCNODE, and ECNODE models using $k = 2$ modes, compared against the ground-truth data (solid curves). The ground-truth data captures the later stage dynamics of two droplets moving apart from each other with a gradually increasing inter-droplet spacing over time. While all three models capture the shape and dynamics of the right droplet well, the NODE and the MCNODE do not consistently maintain the positivity of the reconstructed solution. At time $t = 550$, both NODE and

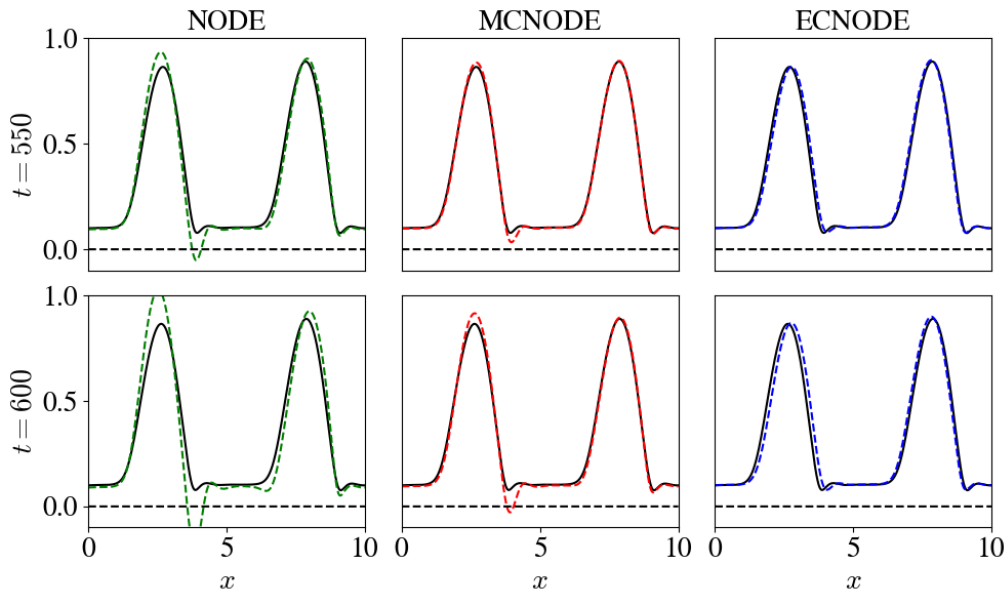

Figure 3.5: Reconstructed solutions (dashed curves) predicted by the trained NODE, MCNODE, and ECNODE models, along with ground-truth data (solid curves), at times $t = 550$ and $t = 600$ of a test sample using $k = 2$ modes.

MCNODE reconstructions display enlarged capillary wave near the right contact line of the left droplet, and at $t = 600$, both reconstructions dip below $\hat{h} = 0$. In contrast, the predicted solution by the ECNODE model remains positive over time, though it exhibits a slight phase shift in the left droplet.

Next, we compare the performance of the models using four metrics to evaluate the quality of the reconstructions over the testing samples. The training error is the average relative L_2 error over the training times $t_1 \leq t \leq t_N$, defined for each sample as

$$E_{\text{train}} = \frac{1}{k} \sum_{i=1}^{k} \sqrt{\frac{\sum_{j=1}^{N} (u_i(t_j) - \hat{u}_i(t_j))^2}{\sum_{j=1}^{N} u_i(t_j)^2}}. \tag{3.41}$$

The prediction error is the average relative L_2 error over the prediction times $t_{N+1} \leq t \leq t_M$, defined for each sample as

$$E_{\text{pred}} = \frac{1}{k} \sum_{i=1}^{k} \sqrt{\frac{\sum_{j=N+1}^{M} (u_i(t_j) - \hat{u}_i(t_j))^2}{\sum_{j=N+1}^{M} u_i(t_j)^2}}. \tag{3.42}$$

The mass error is the relative error of the mass of t_{i+1} and t_i for all prediction times, defined

Table 3.2: A table comparing the performance of NODE and proposed physics-constrained NODEs trained with 2 modes across testing samples.

	E_{train}	E_{pred}	E_{mass}	Minimum $\hat{h}(x,t)$
NODE	0.01523	0.06167	0.07451	-0.2055
MCNODE	0.01406	0.03165	$3.875 \cdot 10^{-8}$	-0.02805
ECNODE	0.01806	0.04524	0.08818	0.03379

for each sample as

$$E_{\text{mass}} = \sum_{j=N+1}^{M} \frac{|\mathcal{M}(\hat{h}(t_j)) - \mathcal{M}(\hat{h}(t_N))|}{\mathcal{M}(\hat{h}(t_N))}, \tag{3.43}$$

and the minimum $\hat{h}(x,t)$ represents the minimal height that the reconstructed solution $\hat{h}(x,t)$ takes across all test samples.

Table 3.2 presents the performance comparison of the NODE, MCNODE, and ECNODE across all testing samples. First, we note that the training error does not vary significantly across models, with all three models yielding a training error on the order or 10^{-2}. Since the MCNODE satisfies the constraint (3.30), the mass error should be zero if the ODE is solved exactly. Thus, we expect the mass of the MCNODE reconstruction to be conserved within the accuracy of the ODE solver. Table 3.2 shows that the errors for the MCNODE model are on the order of 10^{-8}, demonstrating that the MCNODE model enforces the conservation of mass (3.7). Enforcing this conservation law also improves the accuracy of the predictions, with the average prediction error of the MCNODE being 51% of the prediction error of the NODE. The minimum values that $\hat{h}$ takes for both the NODE and MCNODE reconstructions are negative, leading to unphysical solutions (see Figure 3.5). Enforcing the entropy constraint in the ECNODE model results in a positive solution, in accordance with Theorem 1, and improves the prediction error compared to the vanilla NODE.

3.5.2 Performance of the MECNODE model

Next, we examine the performance of the MECNODE model (3.28) and compare it against the vanilla NODE (3.5), ECNODE (3.34), and MCNODE (3.31) models. In this case, we use $k = 4$ modes for the POD reconstruction of the training samples so that the optimization problem for MECNODE is not over-constrained. With $k = 4$ modes, the POD reconstructions for all 100 samples are positive. We divide the samples into $N_r = 60$ training samples, $N_v = 20$ validation samples, and $N_t = 20$ test samples.

Figure 3.6 depicts snapshot reconstructions (dashed curves) of a test sample using the

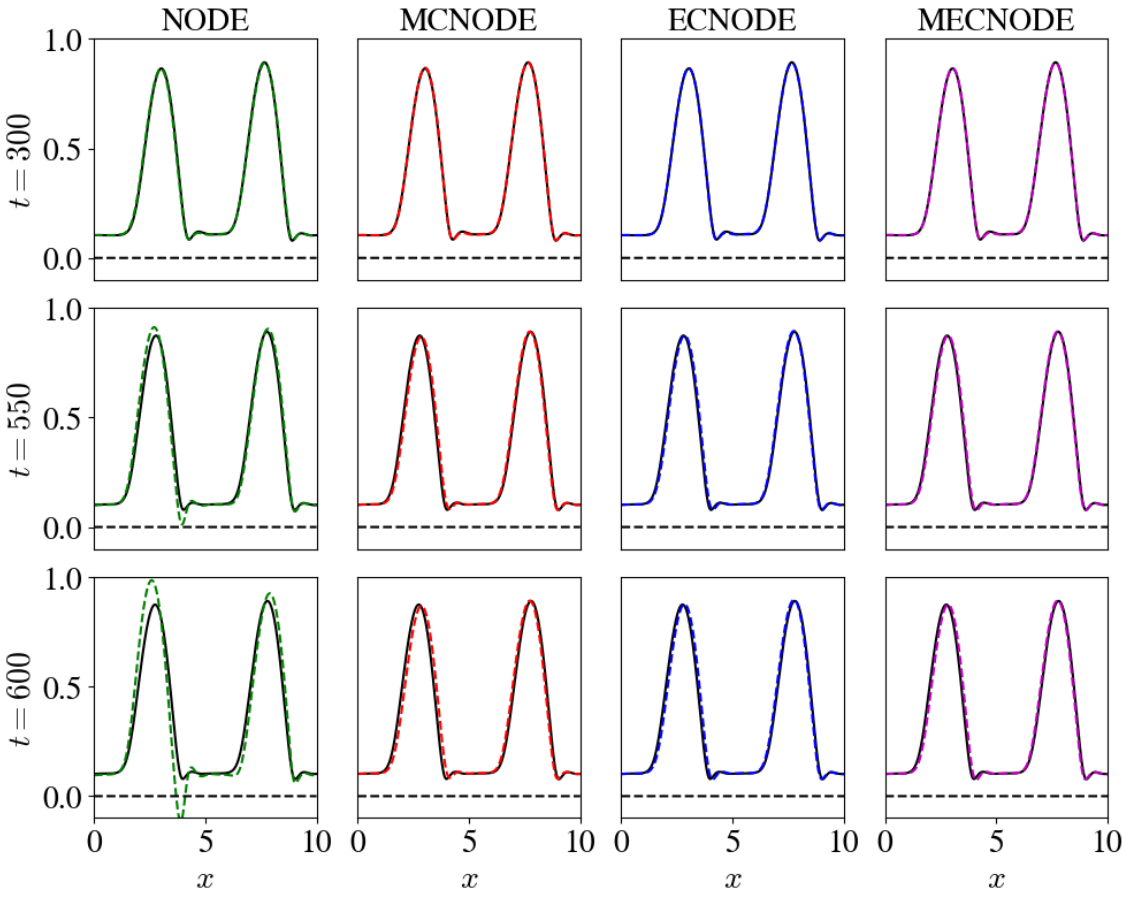

Figure 3.6: Reconstructions (dashed curves) of a test sample from all four trained models at times $t = 300$, $t = 550$, and $t = 600$ using $k = 4$ modes, compared against the ground-truth data (solid curves).

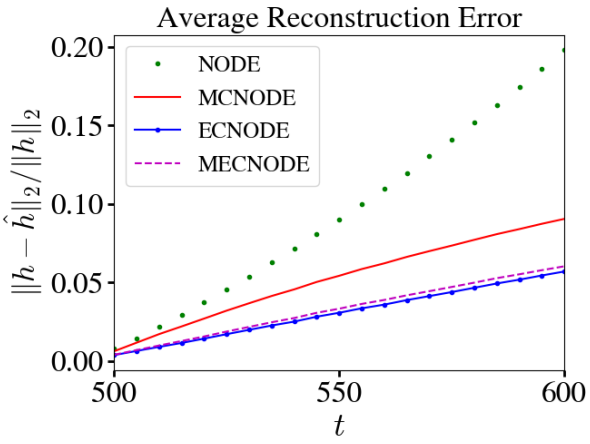

Figure 3.7: Average reconstruction error across all test samples over the prediction times $[500, 600]$.

vanilla NODE, MCNODE, ECNODE, and MECNODE, compared against the ground-truth data (solid curves). The reconstructed solutions from the four models align well with the ground-truth data as the inter-droplet spacing increases from time $t = 300$ to $t = 600$. However, we observe that the reconstructions become less accurate as t increases. This trend is further illustrated in Figure 3.7, which plots the average reconstruction error across all test samples over prediction time. The vanilla NODE exhibits a much faster increase in error compared to the physics-constrained NODEs. In addition, the NODE produces an unphysical prediction as the droplet height $\hat{h}$ falls below zero at $t = 600$, similar to the behavior observed in Figure 3.5 for NODE and MCNODE reconstructions with $k = 2$ modes.

We now examine the mass of the NODE reconstructions. Since the POD approximation does not conserve mass, we do not expect mass conservation in our reconstructions over the training time $[200, 500)$, as this aligns with the POD data. However, the mass during the prediction interval $[500, 600]$ should be conserved by both MCNODE and MECNODE. Figure 3.8 (left) depicts the reconstructed mass over time of each NODE model using the same test sample as in Figure 3.6. We observe that the NODE and ECNODE reconstructions violate the mass conservation over the prediction times $[505, 600]$. To quantify this violation, we examine the relative error in mass over the prediction times in Figure 3.8 (right). The NODE and ECNODE do not conserve mass, exhibiting the mass error that varies by over an order of magnitude in time. In contrast, the mass-conserving models, the MCNODE and MECNODE, maintain mass conservation at a level around 10^{-9}.

In Table 3.3, we compare the errors over the testing samples of models. The training error (3.41), prediction error (3.42), and the mass error (3.43) are all defined as before. Since the

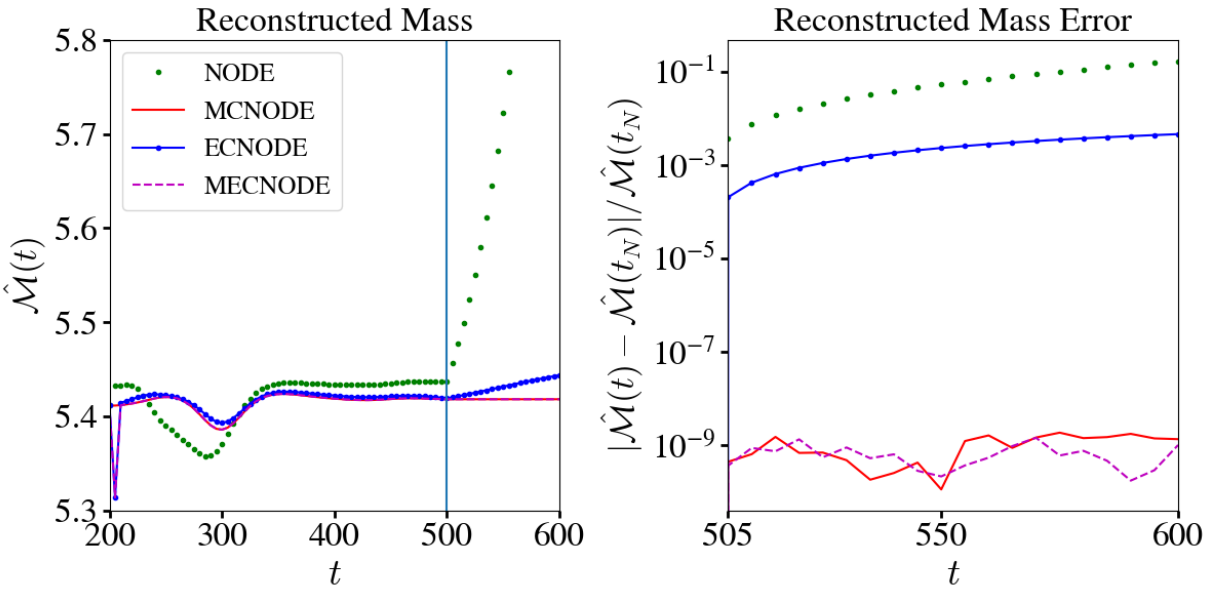

Figure 3.8: (Left) A plot of the reconstructed mass over time of the trained models over a test sample. (Right) The relative error the reconstructed mass of over the prediction times.

Table 3.3: A table comparing various errors in NODE and physics-constrained NODEs.

	E_{train}	E_{pred}	E_{mass}	Minimum $\hat{h}(x,t)$
NODE	0.01275	0.09471	0.3629	-0.5738
MCNODE	0.009154	0.05206	$9.589 \cdot 10^{-9}$	0.05653
ECNODE	0.01119	0.03045	0.04057	0.05951
MECNODE	0.008902	0.03279	$6.189 \cdot 10^{-9}$	0.05647

MCNODE and MECNODE satisfy the constraint $\mathcal{M}(\hat{h}) = \mathcal{M}_0$, the mass error should be on the order of the accuracy of the ODE solver. Table 3.3 shows the mass error for the MCNODE model and MECNODE are of the order of 10^{-9}, demonstrating that the MECNODE and MCNODE models properly enforce the conservation of mass (3.7). Enforcing this conservation law also improves the accuracy of the predictions, with the average prediction error of the MCNODE being 55% of the prediction error of the NODE.

Enforcing the entropy constraint not only guarantees a positive reconstructed solution, but also improves the prediction error, with the ECNODE prediction error being 34.6% of that of the vanilla NODE. Table 3.3 also shows the ECNODE has an improved mass error over the vanilla NODE. The improvement in the mass error is partially due to a nonlinear relationship between the gradients of the entropy and the mass, where by (3.37) and (3.38)

we have

$$A^2\nabla_{\mathbf{u}}\hat{\mathcal{E}} = A^2\int_0^L G'(\hat{h})\mathbf{v}(x)dx \approx \int_0^L \frac{3}{2}\phi(\alpha)(1+\alpha\hat{h})\left(1-\frac{A^2}{\hat{h}^2}\right)\mathbf{v}(x)dx. \tag{3.44}$$

Assuming that $0 < A \ll \hat{h}$, we obtain the leading-order representation of (3.44) as

$$\begin{aligned} A^2\nabla_{\mathbf{u}}\hat{\mathcal{E}} &\approx \frac{3}{2}\phi(\alpha)\int_0^L (1+\alpha\hat{h})\mathbf{v}(x)dx \\ &= \frac{3}{2}\phi(\alpha)\left(\alpha\mathbf{u}(t) + \int_0^L \mathbf{v}(x)dx\right) \\ &= \frac{3}{2}\phi(\alpha)\nabla_{\mathbf{u}}\hat{\mathcal{M}}, \end{aligned}$$

where the last equality follows from (3.37). It follows that if A is small, then we have $\nabla_{\mathbf{u}}\hat{\mathcal{E}} \approx (3\phi(\alpha)/(2A^2))\nabla_{\mathbf{u}}\hat{\mathcal{M}}$, and $\nabla_{\mathbf{u}}\hat{\mathcal{E}}$ may be close to being linearly dependent to $\nabla_{\mathbf{u}}\hat{\mathcal{M}}$. The MECNODE achieves similar improvements in the prediction error as the ECNODE, along with a positive reconstruction, while also conserving the mass up to the accuracy of the ODE solver.

3.6 Conclusions and Discussion

The developed physics-constrained NODE models leverage key conditions derived from the physics-based PDE model and the flexibility of data-driven methodologies. By enforcing mass conservation and bounded entropy conditions as optimization constraints, we designed NODE variants that preserve conserved quantities and ensure solution positivity at the POD level, as demonstrated both analytically and numerically. Unlike vanilla NODE, which may produce unphysical predictions for droplet profiles, our algorithm learns the POD representation of droplet dynamics while ensuring important solution properties are preserved. The trained (MEC)NODE models can function as surrogate models for the rapid evaluation of droplet dynamics in control systems and serve as building blocks for predicting droplet dynamics in large-scale fiber coating systems.

Our approach can be easily extended to other NODE-based learning tasks for physical systems with conserved quantities (Hilliard and Farazmand 2024). To ensure the positivity-preserving property of learned solutions from other PDE models, the bounded entropy condition imposed in the ECNODE and MECNODE models must be adapted to reflect the underlying structure of the specific PDEs. Positivity-preserving numerical methods have been extensively developed for high-order nonlinear parabolic equations modeling free surface

flows (Zhornitskaya and Bertozzi 1999), quantum hydrodynamics (Jüngel 2001; Braukhoff and Jüngel 2020), and other gradient flows (Düring et al. 2010). These numerical schemes are built on physical entropy dissipation principles and have been shown to exhibit desirable convergence and stability properties. To apply our physics-constrained NODE approach to these PDE solutions, the corresponding physical entropy condition must be appropriately incorporated into the NODE models. While the numerical examples in this chapter focus on a one-dimensional case, the proposed NODE model is expected to extend naturally to higher-dimensional problems.

It would also be of interest to extend the proposed physics-based NODE framework to other types of Neural ODE models (Dupont et al. 2019; Xia et al. 2021) for learning POD representations of physical systems. For instance, Baker et al. (2023) demonstrated that the heavy ball NODE (HBNODE) outperforms the vanilla NODE in learning the POD of complex systems due to the well-structured spectrum of the model. However, directly incorporating the conserved quantities and entropy conditions into the HBNODE model may compromise the structure of the model, potentially leading to reduced training performance and generalizability. A balanced approach that leverages both the structure of HBNODE and the physical constraints derived from PDE models could result in improved model performance.

CHAPTER

4

AN ANALYSIS OF A RANDOMIZED ALGORITHM FOR COMPUTING RANK-REVEALING QR FACTORIZATIONS

In many real-world machine learning applications, datasets can be massive, and not all data equally affect model performance. Some data may be redundant, irrelevant, or even harmful. Feature selection methods remove undesirable data to improve model performance and reduce computational time (Guyon and Elisseeff 2003). Given a data matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, one way of performing feature selection is by selecting important rows/columns of $\mathbf{A}$, which can be done with rank-revealing QR (RRQR) algorithms. RRQRs are also used in other applications, such as rank-deficient least squares problems, low-rank approximations, and nonsymmetric eigenproblems (Chan and Hansen 1992; Gu and Eisenstat 1996).

QR with Column Pivoting (Businger and Golub 1966) was the first RRQR algorithm. In general, the QR with Column Pivoting algorithm works well in practice, but there are adversarial matrices where it can fail arbitrarily badly, such as the Kahan matrix (Kahan 1966). Golub, Klema, and Stewart (GKS) developed an RRQR factorization that uses QR with column pivoting on the dominant right singular vectors of $\mathbf{A}$ instead of using $\mathbf{A}$ directly (Golub et al. 1976). However, this technique requires the computation of the SVD of $\mathbf{A}$, which could

be prohibitively expensive for large problems. Armstrong et al. (2023) developed a variation of the GKS algorithm more suitable for large matrices by incorporating randomization into the SVD.

In this chapter, we perform an analysis of a two-step approach for computing a rank-revealing QR of a matrix $\mathbf{A}$. The first step of this approach is to find a matrix $\mathbf{W} \in \mathbb{R}^{n\times k}$ whose columns form an orthonormal basis for a k-dimensional space that approximates the range of $\mathbf{A}$. Afterwards, we select columns of $\mathbf{W}^{\mathrm{T}}$ (or equivalently, rows of $\mathbf{W}$) and use the same indices to select columns of $\mathbf{A}$. We also further analyze the randomized GKS algorithm (Armstrong et al. 2023) as an implementation of this two-step approach. This analysis provides theoretical bounds for all singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ for algorithms that follow this approach, offering stronger guarantees than those that currently exist. We then perform numerical experiments showing the effectiveness of our analysis on several test problems, including the fiber-coating model.

We begin by describing the assumptions used throughout this chapter. For convenience, we define

$$\gamma := \frac{\sigma_{k+1}(\mathbf{A})}{\sigma_k(\mathbf{A})}$$

as the inverse of the singular value gap from k to $k+1$.

Assumptions. Throughout the chapter, we use the following assumptions about the matrix $\mathbf{A}$ and the target rank k:

1. $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$,
2. $1 \leq k \leq \operatorname{rank}(\mathbf{A})$ and $k < n$,
3. $\sigma_{k+1}(\mathbf{A}) < \sigma_k(\mathbf{A})$.

These assumptions guarantee that $\sigma_k(\mathbf{A}) \neq 0$, that $0 \leq \gamma < 1$, and that $\mathcal{R}(\mathbf{V}_k)$ is well-defined.

The structure of this chapter is as follows. In §4.1, we review an algorithm proposed by Golub, Klema, and Stewart (GKS) and show that this algorithm provides a bound for all singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$. In §4.2, we propose a generalized approach of the GKS Algorithm and analyze an implementation of this approach given by Armstrong et al. (2023). In §4.3, we analyze the general approach and algorithm and provide bounds for the singular values and norms of the matrices $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ produced by this algorithm. Then, in §4.4, we provide numerical results to demonstrate the effectiveness of the algorithm and the analysis. In §4.5, we show the proofs of the theorems in this chapter. Lastly, §4.6 provides a conclusion and ideas for future work.

4.1 Golub, Klema, and Stewart RRQR

An early algorithm for rank-revealing QR factorizations was given by Golub, Klema, and Stewart (GKS) in Golub et al. (1976). Their idea was to recognize from $\mathbf{A}\mathbf{\Pi} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}\mathbf{\Pi}$ that permuting columns of $\mathbf{A}$ is equivalent to permuting columns of $\mathbf{V}^{\mathrm{T}}$. So if we select well-conditioned columns of $\mathbf{V}_k^{\mathrm{T}}$, then the same permutation applied to $\mathbf{A}$ would select linearly independent columns of $\mathbf{A}$ as well. Since the GKS algorithm uses $\mathbf{\Pi}_1$ to select linearly independent columns of $\mathbf{V}_k^{\mathrm{T}}$, it follows that the matrix $\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular. Golub, Klema, and Stewart gave a lower bound for the smallest singular value of $\mathbf{R}_{11}$ and the largest singular value of $\mathbf{R}_{22}$. In Theorem 2, we expand these results to all singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$.

Theorem 2. *Let* $\mathbf{A}\mathbf{\Pi} = \mathbf{Q}\mathbf{R}$ *be partitioned as in* (2.44) *with* $\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1$ *nonsingular. Then the following hold:*

$$\frac{\sigma_i(\mathbf{A})}{\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} \leq \sigma_i(\mathbf{R}_{11}) \qquad 1 \leq i \leq k, \tag{4.1}$$

$$\sigma_j(\mathbf{R}_{22}) \leq \sigma_{k+j}(\mathbf{A})\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2 \qquad 1 \leq j \leq n-k. \tag{4.2}$$

Proof. See §4.5.2. □

To provide a geometric interpretation of Theorem 2, we recall our discussion of principal angles between subspaces (2.53), which states that

$$\frac{1}{\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} = \cos(\varphi_k(\mathcal{R}(\mathbf{V}_k), \mathcal{R}(\mathbf{\Pi}_1))).$$

Substituting this into Theorem 2 yields

$$\sigma_i(\mathbf{A})\cos(\varphi_k(\mathcal{R}(\mathbf{V}_k), \mathcal{R}(\mathbf{\Pi}_1))) \leq \sigma_i(\mathbf{R}_{11}) \leq \sigma_i(\mathbf{A}) \qquad 1 \leq i \leq k,$$

$$\sigma_{k+j}(\mathbf{A}) \leq \sigma_j(\mathbf{R}_{22}) \leq \frac{\sigma_{k+j}(\mathbf{A})}{\cos(\varphi_k(\mathcal{R}(\mathbf{V}_k), \mathcal{R}(\mathbf{\Pi}_1)))} \qquad 1 \leq j \leq n-k.$$

If $\mathcal{R}(\mathbf{V}_k)$ is close to $\mathcal{R}(\mathbf{\Pi}_1)$, then $\cos(\varphi_k(\mathcal{R}(\mathbf{V}_k), \mathcal{R}(\mathbf{\Pi}_1)))$ is close to 1. In that case, the singular values of $\mathbf{R}_{11}$ are close to the k dominant singular values of $\mathbf{A}$ and the singular values of $\mathbf{R}_{22}$ are close to the $n-k$ small singular values of $\mathbf{A}$.

4.2 Rank-Revealing QR Algorithm

In this section, we discuss a technique for computing an RRQR that is similar in spirit to the algorithm proposed by Golub, Klema, and Stewart (Golub et al. 1976). This technique

is presented in Algorithm Skeleton 1. The idea of the technique is that instead of selecting columns from $\mathbf{A}$ directly or from $\mathbf{V}_k^T$ as is done in the GKS algorithm, we opt to select columns from some $\mathbf{W}^{\mathrm{T}}$, where $\mathbf{W}$ has orthonormal columns and $\mathcal{R}(\mathbf{W})$ approximates $\mathcal{R}(\mathbf{V}_k)$. This $\mathbf{W}$ can come from any rank-revealing factorization of $\mathbf{A}$, such as the SVD, UTV factorization, randomized variants of the preceding, and many others. If $\mathbf{W}$ can be computed more efficiently than $\mathbf{V}_k$, then this technique can be faster than a RRQR from the GKS algorithm.

Algorithm Skeleton 1 Generalized GKS RRQR

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$, Target rank $1 \leq k \leq \operatorname{rank}(A)$.
Output: $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ that selects columns of $\mathbf{A}$, matrix $\mathbf{Q} \in \mathbb{R}^{m\times n}$ with orthonomal columns and upper-triangular matrix $\mathbf{R} \in \mathbb{R}^{n\times n}$ such that $\mathbf{A\Pi} = \mathbf{QR}$.

1: Find $\mathbf{W} \in \mathbb{R}^{n\times k}$ with orthonormal columns such that $\mathcal{R}(\mathbf{W}) \approx \mathcal{R}(\mathbf{V}_k)$.
2: Select columns of $\mathbf{W}^{\mathrm{T}}$ to obtain $\mathbf{\Pi}$
3: Compute the thin QR factorization $\mathbf{A\Pi} = \mathbf{QR}$.

One efficient way to compute the aforementioned $\mathbf{W}$ is by using Randomized SVD with subspace iterations (Halko et al. 2011, Algorithm 4.4). From there we can select columns of $\mathbf{W}$ using a strong RRQR. This leads to the Randomized GKS algorithm proposed in (Armstrong et al. 2023, Algorithm 4.2) which we present in Algorithm 2.

Given a matrix $\mathbf{A}$, a target rank k, and an oversampling parameter p (for our analysis, we frequently use $p = 0$), Algorithm 2 computes a matrix $\mathbf{W} \in \mathbb{R}^{n\times k}$ with orthonormal columns such that $\mathcal{R}(\mathbf{W}) \approx \mathcal{R}(\mathbf{V}_k)$. The idea is to use a Gaussian matrix $\mathbf{\Omega}$ to randomly sample $k + p$ times from the columns of $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^q\mathbf{A}^{\mathrm{T}}$. This sampling creates a sketch matrix $\mathbf{Y} := (\mathbf{A}^{\mathrm{T}}\mathbf{A})^q\mathbf{A}^{\mathrm{T}}\mathbf{\Omega}$. If $\mathbf{U}^{\mathrm{T}}\mathbf{\Omega}$ has full rank, then it follows that $\mathcal{R}(\mathbf{Y}) \approx \mathcal{R}(\mathbf{A}^{\mathrm{T}})$. The algorithm then computes a thin QR factorization of $\mathbf{Y}$ to obtain $\mathbf{Q} \in \mathbb{R}^{n\times(k+p)}$ with columns that form an orthonormal basis for $\mathcal{R}(\mathbf{Y})$. Finally, it computes the k dominant right singular vectors of $\mathbf{AQQ}^{\mathrm{T}}$ to obtain the desired $\mathbf{W}$.

Computing $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^q\mathbf{A}^{\mathrm{T}}$ directly is numerically unstable, and round-off errors can ruin the accuracy of the small singular values. Instead, we orthogonalize the product after each multiplication with $\mathbf{A}$ or $\mathbf{A}^{\mathrm{T}}$ to avoid this instability. In exact arithmetic, the matrices $\mathbf{Q}$ we obtain by computing the product directly and by orthogonalizing at each step are identical (Halko et al. 2011, Remark 4.3).

To consider the cost of Algorithm 2, we need to consider the RRQR method being used to select columns of $\mathbf{W}^{\mathrm{T}}$. Since the strong RRQR may be expensive to compute, QR with column pivoting can be used instead. QR with column pivoting has a cost of $\mathcal{O}(kn^2)$ flops. In

Algorithm 2 Randomized GKS RRQR

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$, target rank $1 \leq k \leq \text{rank}(A)$, oversampling parameter $p \geq 0$ with with $k + p < n$, number of subspace iterations $q \geq 0$, and strong RRQR parameter $f \geq 1$.

Output: Permutation matrix $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ that selects columns of $\mathbf{A}$, matrix $\mathbf{Q} \in \mathbb{R}^{m\times n}$ with orthonomal columns and upper-triangular matrix $\mathbf{R} \in \mathbb{R}^{n\times n}$ such that $\mathbf{A\Pi} = \mathbf{QR}$.

Draw a standard Gaussian matrix $\mathbf{\Omega} \in \mathbb{R}^{m\times(k+p)}$
Compute the $n \times (k + p)$ matrix $\mathbf{Y}_0 = \mathbf{A}^{\mathrm{T}}\mathbf{\Omega}$
Compute thin QR $\mathbf{Y}_0 = \mathbf{Q}_0\mathbf{R}_0$, with $\mathbf{Q}_0 \in \mathbb{R}^{n\times(k+p)}, \mathbf{R}_0 \in \mathbb{R}^{(k+p)\times(k+p)}$
for $j = 1, 2, \ldots, q$ **do**
    Form the $m \times (k + p)$ matrix $\tilde{\mathbf{Y}}_j = \mathbf{AQ}_{j-1}$
    Compute thin QR of $\tilde{\mathbf{Y}}_j = \tilde{\mathbf{Q}}_j\tilde{\mathbf{R}}_j$, with $\tilde{\mathbf{Q}}_j \in \mathbb{R}^{m\times(k+p)}, \tilde{\mathbf{R}}_j \in \mathbb{R}^{(k+p)\times(k+p)}$
    Form the $n \times (k + p)$ matrix $\mathbf{Y}_j = \mathbf{A}^{\mathrm{T}}\tilde{\mathbf{Q}}_{\mathbf{j}}$
    Compute thin QR of $\mathbf{Y}_j = \mathbf{Q}_j\mathbf{R}_j$, with $\mathbf{Q}_j \in \mathbb{R}^{n\times(k+p)}, \mathbf{R}_j \in \mathbb{R}^{(k+p)\times(k+p)}$
end for
Form the $m \times (k + p)$ matrix $\mathbf{B} = \mathbf{AQ}_q$
Find rank-k truncated SVD of $\mathbf{B}$, $\mathbf{B}_k = \hat{\mathbf{U}}_k\hat{\mathbf{\Sigma}}_k\hat{\mathbf{V}}_k^{\mathrm{T}}$
Form the $n \times k$ matrix $\mathbf{W} = \mathbf{Q}_q\hat{\mathbf{V}}_k$
Compute a strong RRQR of $\mathbf{W}^{\mathrm{T}}$, $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$.
Compute the thin QR factorization $\mathbf{A\Pi} = \mathbf{QR}$.

our experience, QR with column pivoting on $\mathbf{W}^{\mathrm{T}}$ often produces a strong RRQR with some constant $f \leq 2$, so it serves as an economical alternative without sacrificing much accuracy. A similar analysis can be found in (Feng et al. 2019, Appendex E). We consider the flop count of each line in Algorithm 2, assuming $\mathbf{A}$ is dense. The cost of computing

1. $\mathbf{\Omega} \in \mathbb{R}^{(k+p)\times m}$ is $\mathcal{O}(m(k + p))$ flops.
2. $\mathbf{Y}_0$ is $2mn(k + p)$ flops.

For $j = 1, 2, \ldots, q$:

4. the QR factorization of $\mathbf{Y}_0^{\mathrm{T}}$ is $2m(k + p)^2$ flops (Golub and Van Loan 2013, 5.2.8).
5. $\tilde{\mathbf{Y}}_j$ is $2mn(k + p)$ flops.
6. the QR factorization of $\tilde{\mathbf{Y}}_j$ is $2n(k + p)^2$ flops.
7. $\mathbf{Y}_j$ is $2mn(k + p)$ flops.
8. the QR factorization of $\mathbf{Y}_j$ is $2m(k + p)^2$ flops.

End for.

10. $\mathbf{B}$ is $2mn(k+p)$ flops.
11. the rank-k truncated SVD of $\mathbf{B}$ is $2n(k+p)^2 + 11(k+p)^3$ flops (Golub and Van Loan 2013, 8.6.3).
12. $\mathbf{W}$ is $2n(k+p)^2$ flop.
13. $\mathbf{\Pi}$ is $\mathcal{O}(nk^2)$ flops using QR with Column Pivoting (Businger and Golub 1966).
14. the QR factorization $\mathbf{A\Pi} = \mathbf{QR}$ is $2mn^2$ flops.

In total, the highest asymptotic cost in computing $\mathbf{\Pi}$ is lines $2, 5, 7$, and 10 at $2mn(k+p)$ flops. These computations are performed $2q+2$ times in total, meaning the approximate cost of selecting columns of $\mathbf{A}$ is $(4q+4)mn(k+p)$ flops. Usually, $q \leq 2$ and $k \ll n$, making the number of flops to compute $\mathbf{\Pi}$ $\mathcal{O}(mnk)$. After computing $\mathbf{\Pi}$, Algorithm 2 computes the QR factorization of $\mathbf{A\Pi}$ in line 14, which takes $2mn^2$ flops. Since the computational cost of finding $\mathbf{\Pi}$ is lower-order than the cost of computing the QR of $\mathbf{A\Pi}$, we conclude that the flop count of Algorithm 2 is asymptotically equivalent to that of the standard QR factorization. However, the additional lower-order terms and the permuting of the columns of $\mathbf{A}$ will slightly increase the computational time of Algorithm 2 compared to the standard QR.

4.3 Main Results

In this section, we provide bounds showing that RRQR produced by the general technique in Algorithm Skeleton 1 or our implementation in Algorithm 2 accurately captures the singular values of $\mathbf{A}$. Specifically, we establish lower bounds for the singular values of $\mathbf{R}_{11}$ and upper bounds for the singular values of $\mathbf{R}_{22}$.

4.3.1 Singular Value Bounds for $\mathbf{R}_{22}$

We begin our analysis by providing bounds for the singular values of $\mathbf{R}_{22}$. For the following theorem, we only assume that $\mathbf{W}$ has orthonormal columns, and that $\mathbf{\Pi}_1$ selects a sufficiently well-conditioned set of columns from $\mathbf{W}$, so that $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular. This theorem offers insight into why the general technique in Algorithm 1 yields an $\mathbf{R}_{22}$ submatrix with meaningful singular values.

Theorem 3 ($\mathbf{R}_{22}$ bounds for Algorithm Skeleton 1)**.** *Let* $\mathbf{A}$ *and* k *satisfy the Assumptions,* $\mathbf{W} \in \mathbb{R}^{n\times k}$ *have orthonormal columns, and* $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ *be a permutation matrix such that* $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ *is nonsingular. Let* $\mathbf{A\Pi} = \mathbf{QR}$ *be a QR factorization partitioned as in* (2.44)*. Then*

$$\sigma_j\left(\mathbf{R}_{22}\right) \leq \sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2 \qquad 1 \leq j \leq n-k. \tag{4.3}$$

Proof. See §4.5.3. □

Using the expression for the cosine of the largest principal angle given in (2.53), we can write (4.3) geometrically as

$$\sigma_j\left(\mathbf{R}_{22}\right) \leq \frac{\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))}{\cos(\varphi_k(\mathcal{R}(\mathbf{W}), \mathcal{R}(\mathbf{\Pi}_1)))} \qquad 1 \leq j \leq n-k.$$

To justify the relationship between this theorem and Algorithm Skeleton 1, we consider the cosine of the largest principal angle between $\mathcal{R}(\mathbf{W})$ and $\mathcal{R}(\mathbf{\Pi}_1)$, $\cos(\varphi_k(\mathcal{R}(\mathbf{W}), \mathcal{R}(\mathbf{\Pi}_1)))$, and $\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))$, individually. The cosine of the largest principal angle between $\mathcal{R}(\mathbf{W})$ and $\mathcal{R}(\mathbf{\Pi}_1)$ quantifies how well $\mathcal{R}(\mathbf{\Pi}_1)$ approximates $\mathcal{R}(\mathbf{W})$. Ideally, $\mathcal{R}(\mathbf{\Pi}_1) = \mathcal{R}(\mathbf{W})$. In this case, all principal angles are 0 and so the cosine of the largest principal angle is 1. Thus, the singular values of $\mathbf{R}_{22}$ are the singular values of the low rank error, $\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}$. However, if $\mathcal{R}(\mathbf{\Pi}_1)$ is nearly orthogonal to $\mathcal{R}(\mathbf{W})$, then the cosine of the largest principal angle is close to 0 and the bound loses tightness. For this reason, we choose $\mathbf{\Pi}_1$ such that $\mathcal{R}(\mathbf{\Pi}_1)$ captures $\mathcal{R}(\mathbf{W})$. A good choice of $\mathbf{\Pi}$ to achieve this is one obtained by performing a rank-revealing QR on $\mathbf{W}^{\mathrm{T}}$.

Furthermore, the singular values of $\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}$ are dependent on the overlap between $\mathcal{R}(\mathbf{W})$ and $\mathcal{R}(\mathbf{V}_k)$. If $\mathcal{R}(\mathbf{W}) = \mathcal{R}(\mathbf{V}_k)$, the first $n-k$ singular values of $\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}$ are exactly the $n-k$ small singular values of $\mathbf{A}$. On the other hand, if $\mathcal{R}(\mathbf{W})$ is a subspace of $\mathcal{R}(\mathbf{V}_\perp)$, then the singular values of $\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}$ are the large singular values of $\mathbf{A}$. Given that this bound is better if the singular values of $\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}$ are small, we want to choose $\mathbf{W}$ such that $\mathcal{R}(\mathbf{W}) \approx \mathcal{R}(\mathbf{V}_k)$. Additionally, when $\mathbf{W} = \mathbf{V}_k$, the bound in Theorem 3 becomes the GKS bounds (4.2), as shown by the following remark.

Remark 2. *In Theorem 3, we consider* $\mathbf{W}$ *as an approximation to the* k *dominant right singular vectors of* $\mathbf{A}$*. In the special case where* $\mathbf{W}$ *is exactly* $\mathbf{V}_k$*,*

$$\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{V}_k\mathbf{V}_k^{\mathrm{T}})) = \sigma_{k+j}(\mathbf{A}) \qquad 1 \leq j \leq n-k,$$

and the bound in Theorem 3 becomes

$$\sigma_j\left(\mathbf{R}_{22}\right) \leq \sigma_{k+j}(\mathbf{A})\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2 \qquad 1 \leq j \leq n-k,$$

which is identical to (4.2).

We now implement the RRQR given in Algorithm 2. The following theorem provides structural bounds for all singular values of $\mathbf{R}_{22}$ in the case without oversampling.

Theorem 4 (Structural $\mathbf{R}_{22}$ Bounds for Algorithm 2)**.** *Let* $\mathbf{A}$ *and* k *satisfy the Assumptions. Let* $\mathbf{\Omega} \in \mathbb{R}^{m\times k}$ *and* $\mathbf{W} \in \mathbb{R}^{n\times k}$ *be computed by Algorithm 2 without oversampling* ($p = 0$)*. Let* $\mathbf{\Pi}$ *be computed by Algorithm 2. We partition the QR factorization* $\mathbf{A\Pi} = \mathbf{QR}$ *as in* (2.44) *and* $\mathbf{U}^{\mathrm{T}}\mathbf{\Omega}$ *as*

$$\mathbf{U}^{\mathrm{T}}\mathbf{\Omega} = \begin{bmatrix}\mathbf{\Omega}_1 \\ \mathbf{\Omega}_2\end{bmatrix},$$

where $\mathbf{\Omega}_1 \in \mathbb{R}^{k\times k}$ *and* $\mathbf{\Omega}_2 \in \mathbb{R}^{(m-k)\times k}$*. Assume* $\mathbf{\Omega}_1$ *is nonsingular. Then*

$$\sigma_j\left(\mathbf{R}_{22}\right) \leq \beta(n,k)\sqrt{\sigma_{j+k}(\mathbf{A})^2 + \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2^2} \qquad 1 \leq j \leq n-k.$$

Proof. See §4.5.3. □

The matrix $\mathbf{\Omega}$ used by Algorithm 2 is a Gaussian random matrix. Since the Gaussian distribution is rotationally invariant, $\mathbf{U}^{\mathrm{T}}\mathbf{\Omega}$ is also a Gaussian random matrix. Also, $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are disjoint submatrices of $\mathbf{U}^{\mathrm{T}}\mathbf{\Omega}$, so these two matrices are Gaussian and statistically independent. Furthermore, the columns of a Gaussian matrix are almost surely linearly independent, so the assumption that $\mathbf{\Omega}_1$ is nonsingular holds with probability one (Halko et al. 2011, p. 274).

The term $\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-}1\|_2^2$ can be controlled by γ^{4q}. Since $\gamma < 1$ by assumption, we can choose q large enough to make $\gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2^2$ smaller than any given tolerance with high probability. The required q may be extremely large if γ is close to one, so a small γ is preferred, corresponding to a large singular value gap between k and $k+1$. If $\gamma = 0$, corresponding to $\mathbf{A}$ being rank k, then Theorem 4 implies that all singular values of $\mathbf{R}_{22}$ are 0, yielding $\mathbf{R}_{22} = \mathbf{0}$.

In this next theorem, we use probabilistic techniques from random matrix theory to bound the singular values of the submatrix $\mathbf{R}_{22}$ generated by Algorithm 2 with high probability. For this result, we assume no oversampling. We consider the analysis with oversampling in §4.3.2.

Theorem 5 (Probabilistic $\mathbf{R}_{22}$ Bounds for Algorithm 2)**.** *Let* $\mathbf{A}$ *and* k *satisfy the Assumptions and let* $\mathbf{\Pi}$ *be computed by Algorithm 2. Let* $\mathbf{A\Pi} = \mathbf{QR}$ *be partitioned as in* (2.44)*. For any*

$0 < \Delta < 1$, *let*

$$D = \frac{2\sqrt{k}}{\Delta}\left[\|\boldsymbol{\Sigma}_\perp\|_F + \left(\sqrt{k} + \sqrt{\ln\left(\frac{4}{\Delta^2}\right)}\right)\|\boldsymbol{\Sigma}_\perp\|_2\right]$$

and $\beta(n,k) = \sqrt{1+k(n-k)}$. *Then with probability* $1-\Delta$,

$$\sigma_j\left(\mathbf{R}_{22}\right) \leq \beta(n,k)\sqrt{\sigma_{j+k}(\mathbf{A})^2 + \gamma^{4q}D^2} \qquad 1 \leq j \leq n-k.$$

Proof. See §4.5.3. □

As in the previous result, the effects of D on the upper bound can be controlled by the number of subspace iterations q. Since $\gamma < 1$, for a fixed probability Δ, we can choose q large enough to make $\gamma^{4q}D^2$ smaller than any given tolerance. The required q may be extremely large if γ is close to one, so a large singular value gap between k and $k+1$ is preferred. If $\gamma^{4q}D^2$ is sufficiently small, we obtain

$$\sigma_j\left(\mathbf{R}_{22}\right) \leq \beta(n,k)\sqrt{\sigma_{j+k}(\mathbf{A})^2 + \gamma^{4q}D^2} \approx \beta(n,k)\sigma_{j+k}(\mathbf{A}), \qquad 1 \leq j \leq n-k.$$

and Algorithm 2 gives similar guarantees as a strong RRQR.

4.3.2 Oversampling Analysis for $\mathbf{R}_{22}$

In §4.3.1, we presented the bounds for the singular values of $\mathbf{R}_{22}$ without oversampling. In this subsection, we derive spectral and Frobenius norm bounds of $\mathbf{R}_{22}$ relating to Algorithms 1 and 2 when oversampling is applied. In practice, oversampling is commonly used in randomized algorithms to reduce the risk of losing accuracy in the computation of the small singular values (Martinsson et al. 2019). In our experience, oversampling does not significantly affect the performance of Algorithm 2, but it does allow us to form stronger probabilistic upper bounds for the singular values of $\mathbf{R}_{22}$, which makes considering oversampling important in the analysis of this algorithm.

Our first oversampling theorem makes a more specific assumption on $\mathbf{W}$ than the one from Algorithm Skeleton 1. We now assume that $\mathbf{W}$ consists of the k dominant right singular vectors of $\mathbf{AQQ}^{\mathrm{T}}$, where $\mathbf{Q} \in \mathbb{R}^{n\times(k+p)}$ is any matrix with orthonormal columns. The assumption that $\boldsymbol{\Pi}$ is chosen such that $\mathbf{W}^{\mathrm{T}}\boldsymbol{\Pi}_1$ is nonsingular stays the same.

Theorem 6 (Oversampling $\mathbf{R}_{22}$ Bounds)**.** *Let* $\mathbf{A}$ *and* k *satisfy the Assumptions,* $\mathbf{Q} \in \mathbb{R}^{n\times(k+p)}$ *have orthonormal columns, and* $\mathbf{W} \in \mathbb{R}^{n\times k}$ *have columns consisting of the* k *dominant right*

singular vectors of $\mathbf{A}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$. *Let* $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ *be nonsingular. Then*

$$\|\mathbf{R}_{22}\|_\xi \leq \|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2\sqrt{\|\mathbf{\Sigma}_\perp\|_\xi^2 + \|\mathbf{A}_k(\mathbf{I}-\mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_F^2}, \tag{4.4}$$

where $\|\cdot\|_\xi$ *denotes either the spectral or Frobenius norm.*

Proof. See §4.5.4. □

Algorithm 2 constructs $\mathbf{Q}_q \in \mathbb{R}^{n\times(k+p)}$, a matrix whose columns form an orthonormal basis for the range of $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^q\mathbf{A}^{\mathrm{T}}\mathbf{\Omega}$, and $\mathbf{W} \in \mathbb{R}^{n\times k}$ has columns consisting of the k dominant right singular vectors of $\mathbf{A}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$. The quantity $\|\mathbf{A}_k - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\|_F$ quantifies how well this basis captures $\mathcal{R}(\mathbf{A}_k)$, with a smaller norm corresponding to a better basis. Generally, increasing the oversampling parameter p will increase the rank of the orthogonal projector $\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$, which decreases $\|\mathbf{A}_k - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\|_F$ and improves the approximate basis. The following theorem shows that this algorithm's accuracy depends on the the singular values of $\mathbf{A}$ and interaction between $\mathbf{U}$ and $\mathbf{\Omega}$.

Theorem 7 (Structural $\mathbf{R}_{22}$ Oversampling Bounds for Algorithm 2)**.** *Let* $\mathbf{A}$ *and* k *satisfy the Assumptions. Let* $\mathbf{\Omega} \in \mathbb{R}^{m\times k}$ *be computed by Algorithm 2. Let* $\mathbf{\Pi}$ *be computed by Algorithm 2. We partition the QR factorization* $\mathbf{A}\mathbf{\Pi} = \mathbf{Q}\mathbf{R}$ *as in* (2.44) *and* $\mathbf{U}^{\mathrm{T}}\mathbf{\Omega}$ *as*

$$\mathbf{U}^{\mathrm{T}}\mathbf{\Omega} = \begin{bmatrix}\mathbf{\Omega}_1\\ \mathbf{\Omega}_2\end{bmatrix},$$

where $\mathbf{\Omega}_1 \in \mathbb{R}^{k\times(k+p)}$ *and* $\mathbf{\Omega}_2 \in \mathbb{R}^{(m-k)\times(k+p)}$. *Assume* $\mathbf{\Omega}_1$ *has full row rank. Then*

$$\|\mathbf{R}_{22}\|_\xi \leq \beta(n,k)\sqrt{\|\mathbf{\Sigma}_\perp\|_\xi^2 + \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F^2}, \tag{4.5}$$

where $\|\cdot\|_\xi$ *denotes either the spectral or Frobenius norm.*

Proof. See §4.5.4. □

If $k \geq 2$ and $p \geq 2$, then from (Halko et al. 2011, Proposition 10.2), we have $\mathbb{E}(\|\mathbf{\Omega}_1^\dagger\|_F^2) = \frac{k}{p-1}$. Thus, if we increase the oversampling, we expect $\|\mathbf{\Omega}_1^\dagger\|_F$ to decrease, and consequently the bound in Theorem 7 becomes tighter.

Just as in the case without oversampling, we consider the singular value distribution of $\mathbf{A}$ and interaction between $\mathbf{U}$ and $\mathbf{\Omega}$. If $\mathbf{\Omega} = \mathbf{U}_k$ or if $\mathbf{A}$ has exact rank k, then $\gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F^2 = 0$ and it follows from Theorem 7 that

$$\|\mathbf{R}_{22}\|_\xi \leq \beta(n,k)\|\mathbf{\Sigma}_\perp\|_\xi.$$

In this case, Algorithm 2 can provide a similar guarantee as the strong RRQR on the spectral and Frobenius norm of $\mathbf{R}_{22}$, even without performing subspace iterations.

Also similar to the case without oversampling, the effects of the randomness in Algorithm 2 appear in Theorem 7 in the term $\|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2 \mathbf{\Omega}_1^{-1}\|_F$. We now probabilistically bound this term to obtain the following theorem.

Theorem 8 (Probabilistic $\mathbf{R}_{22}$ Oversampling Bounds for Algorithm 2)**.** *Let* $\mathbf{A}$ *satisfy the Assumptions,* $\mathbf{A\Pi} = \mathbf{QR}$ *be computed by Algorithm 2 with oversampling parameter* $p \geq 4$*, and* $0 < \Delta < 1$ *be a failure probability. We define*

$$F = \frac{3k}{p+1}\left(\frac{2}{\Delta}\right)^{\frac{2}{p}}\left(\|\mathbf{\Sigma}_\perp\|_F + \left[\sqrt{\ln\left(\frac{4}{\Delta^2}\right)} + \sqrt{k+p}\right]\sigma_{k+1}(\mathbf{A})\right)^2.$$

Then with probability at least $1-\Delta$*, we have*

$$\|\mathbf{R}_{22}\|_\xi \leq \beta(n,k)\sqrt{\|\mathbf{\Sigma}_\perp\|_\xi^2 + \gamma^{4q}F^2}. \tag{4.6}$$

Proof. See §4.5.4. □

The term F decreases as p increases. This implies that increasing the amount of oversampling gives a tighter upper bound on $\mathbf{R}_{22}$. Furthermore, the effects of F on this bound can be controlled by having a large singular value gap (γ small) and/or taking many subspace iterations (q large).

4.3.3 Singular Value Bounds for $\mathbf{R}_{11}$

In this subsection, we bound the singular values of $\mathbf{R}_{11}$ produced methods satisfying Algorithm Skeleton 1 and our implementation in Algorithm 2. For the analysis of Algorithm Skeleton 1, it is reasonable to want to generalize the $\mathbf{R}_{11}$ bounds for the GKS algorithm in Theorem 2 to a corresponding bound that uses $\mathbf{W}$ instead of $\mathbf{V}_k$. That corresponding bound to Theorem 3 for $\mathbf{R}_{11}$ would be

$$\frac{\sigma_i(\mathbf{AWW}^{\mathrm{T}})}{\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} \leq \sigma_i(\mathbf{R}_{11}) \qquad 1 \leq i \leq k. \tag{4.7}$$

However, this result does not hold in general.

Example 1. *Consider the case where* $m = n = 2$ *and* $k = 1$ *with*

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \qquad \mathbf{W} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \textit{and} \quad \mathbf{\Pi} = \mathbf{I}.$$

Then

$$\frac{\|\mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_2}{\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} = \frac{\sqrt{5}}{2} > 1 = \|\mathbf{R}_{11}\|_2,$$

and (4.7) *fails.*

Another consideration is that if we add additional assumptions that $\mathbf{W}$ is a good approximation of $\mathbf{V}_k$ and that $\mathbf{\Pi}_1$ chooses columns of $\mathbf{W}^{\mathrm{T}}$ well, then perhaps we could make (4.7) hold. However, this is also not the case.

Example 2. *Consider the case where* $m = n = 2$ *and* $k = 1$ *with*

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}.$$

Then $\mathbf{A}$ *has SVD* $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$, *where*

$$\mathbf{U} = \mathbf{I}, \quad \mathbf{\Sigma} = \begin{bmatrix} \sqrt{2} & \\ & 0 \end{bmatrix}, \quad \textit{and} \quad \mathbf{V} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Let $0 < \epsilon \ll \left(1 - \frac{1}{\sqrt{2}}\right)$ *and* $\delta = \frac{1}{\sqrt{2}} - \frac{1}{2}\sqrt{-4\epsilon^2 - 4\sqrt{2}\epsilon + 2} > 0$. *This choice of* δ *makes*

$$\mathbf{W} = \begin{bmatrix} \frac{1}{\sqrt{2}} + \epsilon \\ \frac{1}{\sqrt{2}} - \delta \end{bmatrix}$$

have unit norm. Since the first element of $\mathbf{W}$ *is larger than the second, we choose* $\mathbf{\Pi} = \mathbf{I}$ *to make* $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ *as large as possible. Then let* $\mathbf{A}\mathbf{\Pi} = \mathbf{Q}\mathbf{R}$, *where* $\mathbf{Q} = \mathbf{I}$ *and* $\mathbf{R} = \mathbf{A}$ *and*

$$\frac{\|\mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_2}{\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} = \left(\epsilon + \frac{1}{\sqrt{2}}\right)^2 + \left(\epsilon + \frac{1}{\sqrt{2}}\right)\sqrt{1 - \left(\epsilon + \frac{1}{\sqrt{2}}\right)^2} > 1 = \|\mathbf{R}_{11}\|.$$

For ϵ near 0, $\mathbf{W}$ is a small perturbation of $\mathbf{V}_k$ and $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ is as large as possible, however (4.7) still does not hold.

We now provide a theoretical lower bound for the singular values of $\mathbf{R}_{11}$ for a RRQR computed by the general RRQR technique given in Algorithm Skeleton 1. Since the bound (4.7) fails, we need to give a different lower bound for the singular values of $\mathbf{R}_{11}$.

Theorem 9 ($\mathbf{R}_{11}$ Bounds for Algorithm Skeleton 1)**.** *Let* $\mathbf{A}$ *and* k *satisfy the Assumptions,*

$\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ *be nonsingular and let*

$$\sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k)) < \frac{1}{\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2}. \tag{4.8}$$

Then $\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1$ *is nonsingular, and*

$$\sigma_i(\mathbf{A})\left(\frac{1}{\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} - \sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))\right) \le \sigma_i(\mathbf{R}_{11}) \qquad 1 \le i \le k.$$

Proof. See §4.5.5. □

The additional assumption states that $\mathcal{R}(\mathbf{W})$ is sufficiently close to $\mathcal{R}(\mathbf{V}_k)$, and forces the lower bound to be strictly greater than zero. The term multiplying $\sigma_i(\mathbf{A})$ in the bound is the difference between two trigonometric functions,

$$\cos\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{\Pi}_1)) - \sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k)).$$

If $\mathcal{R}(\mathbf{\Pi}_1)$ is close to $\mathcal{R}(\mathbf{W})$, then $\cos\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{\Pi})_1) \approx 1$ improving the tightness of the bound. The sine term measures the overlap between $\mathcal{R}(\mathbf{V}_k)$ and $\mathcal{R}(\mathbf{W})$. If $\mathcal{R}(\mathbf{W}) \approx \mathcal{R}(\mathbf{V}_k)$, then $\sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))$ is small and the lower bound in Theorem 9 approximates (4.7). Furthermore, when $\mathbf{W} = \mathbf{V}_k$, the bound in Theorem 9 becomes (4.1), as shown by the following remark.

Remark 3. *We consider Theorem 9 when* $\mathbf{W} = \mathbf{V}_k$*. In this case,*

$$\sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k)) = 0.$$

Thus, the assumption in (4.8) *is satisfied and the bound in Theorem 9 simplifies to*

$$\frac{\sigma_i(\mathbf{A})}{\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} \le \sigma_i(\mathbf{R}_{11}), \qquad 1 \le j \le k,$$

which is the same as (4.1)*.*

We now consider the structural $\mathbf{R}_{11}$ bounds for Algorithm 2. The following theorem bounds all singular values of $\mathbf{R}_{11}$ for the RRQR associated with Algorithm 2.

Theorem 10 (Structural $\mathbf{R}_{11}$ Bounds for Algorithm 2)**.** *Let* $\mathbf{A}$ *and* k *satisfy the Assumptions. Let* $\mathbf{\Omega} \in \mathbb{R}^{m\times(k+p)}$ *and let* $\mathbf{W} \in \mathbb{R}^{n\times k}$ *and* $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ *be computed by Algorithm 2 with*

oversampling $p \geq 4$ *and*

$$q > \frac{\ln(\sigma_k(\mathbf{A})^2) - \ln\left(\|\boldsymbol{\Sigma}_\perp \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1^\dagger\|_2^2(\beta(n,k)^2 - 1)\right)}{4\ln(\gamma)}.$$

We partition the QR factorization $\mathbf{A}\boldsymbol{\Pi} = \mathbf{QR}$ *as in* (2.44) *and partition* $\mathbf{U}^{\mathrm{T}}\boldsymbol{\Omega}$ *as*

$$\mathbf{U}^{\mathrm{T}}\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{\Omega}_1 \\ \boldsymbol{\Omega}_2 \end{bmatrix},$$

where $\boldsymbol{\Omega}_1 \in \mathbb{R}^{k\times k}$ *and* $\boldsymbol{\Omega}_2 \in \mathbb{R}^{(m-k)\times k}$*. Assume* $\boldsymbol{\Omega}_1$ *is nonsingular. Then*

$$\left(\frac{1}{\beta(n,k)} - \frac{\gamma^{2q}\|\boldsymbol{\Sigma}_\perp \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1^\dagger\|_2}{\sqrt{\sigma_k(\mathbf{A})^2 + \gamma^{4q}\|\boldsymbol{\Sigma}_\perp \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1^\dagger\|_2^2}}\right)\sigma_i(\mathbf{A}) \leq \sigma_i(\mathbf{R}_{11}) \quad 1 \leq i \leq k,$$

where $\beta(n,k) = \sqrt{1 + f^2 k(n-k)}$.

Proof. See §4.5.5. □

The interpretation of this result is similar to that of Theorem 4. There are special cases where Theorem 10 shows that Algorithm 2 provides strong RRQR guarantees for the singular values of $\mathbf{R}_{11}$, as highlighted in the following remark.

Remark 4. *If* $\mathbf{A}$ *has exact rank* k *or if* $\boldsymbol{\Omega} = \mathbf{U}_k$*, then from Theorem 10,*

$$\frac{\sigma_i(\mathbf{A})}{\beta(n,k)} \leq \sigma_i(\mathbf{R}_{11}) \qquad 1 \leq i \leq k, \tag{4.9}$$

and Algorithm 2 has the same guarantees on $\mathbf{R}_{11}$ *as a strong RRQR.*

In this next theorem, we provide a probabilistic upper bound for the singular values of the submatrix $\mathbf{R}_{11}$ produced by Algorithm 2.

Theorem 11 (Probabilistic $\mathbf{R}_{11}$ Bounds for Algorithm 2)**.** *Let* $\mathbf{A}$ *and* k *satisfy the Assumptions. Let* $\boldsymbol{\Pi}$ *be obtained by Algorithm 2 with strong RRQR parameter* $f > 1$*, oversampling* $p \geq 4$ *and with the number of subspace iterations satisfying*

$$q > \frac{\ln(\sigma_k(\mathbf{A})^2) - \ln\left(F^2(\beta(n,k)^2 - 1)\right)}{4\ln(\gamma)},$$

where

$$F = \frac{3k}{p+1}\left(\frac{2}{\Delta}\right)^{\frac{2}{p}}\left(\|\boldsymbol{\Sigma}_\perp\|_F + \left[\sqrt{\ln\left(\frac{4}{\Delta^2}\right)} + \sqrt{k+p}\right]\sigma_{k+1}(\mathbf{A})\right)^2$$

for any $0 < \Delta < 1$. *Let* $\mathbf{A\Pi} = \mathbf{QR}$ *be a QR factorization partitioned as in* (2.44). *Then with probability at least* $1 - \Delta$, *we have*

$$\left(\frac{1}{\beta(n,k)} - \frac{\gamma^{2q} F}{\sqrt{\sigma_k(\mathbf{A})^2 + \gamma^{4q} F^2}} \right) \sigma_i(\mathbf{A}) \leq \sigma_i(\mathbf{R}_{11}) \qquad 1 \leq i \leq k,$$

where $\beta(n,k) = \sqrt{1 + f^2 k(n-k)}$.

Proof. See §4.5.5. □

The additional assumption that q is sufficiently large guarantees that the lower bound is positive. As in Theorem 10, the effects of F on this bound can be controlled by q. If $\gamma^{4q} F^2$ is sufficiently small, we obtain

$$\frac{\sigma_i(\mathbf{A})}{\beta(n,k)} \approx \left(\frac{1}{\beta(n,k)} - \frac{\gamma^{2q+1} C}{\sqrt{1 + \gamma^{4q+2} C^2}} \right) \sigma_i(\mathbf{A}) \leq \sigma_i(\mathbf{R}_{11}) \qquad 1 \leq i \leq k,$$

and Algorithm 2 gives similar guarantees as a strong RRQR.

4.4 Numerical Results

In this section, we conduct numerical experiments to show the accuracy of Algorithm 2 and the bounds in our derived theorems. To do so, we implement Algorithm 2 and take the rank-revealing QR factorization $\mathbf{A\Pi} = \mathbf{QR}$. We focus on the singular values of the submatrices $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ and compare them to the singular values of the input matrix $\mathbf{A}$ and our derived singular value bounds.

4.4.1 Test Matrices

For our first experiment, we consider three 150×100 test matrices. To construct these matrices, we begin with diagonal matrices $\mathbf{\Sigma}$ containing our desired singular value distributions. We use the following singular value distributions, as depicted in Figure 4.1:

1. Exp. Decay: $\mathbf{\Sigma}$ has diagonal entries that exponentially decay starting at 1.
2. S Matrix: $\mathbf{\Sigma}$ has diagonal entries 1 until around the 15^{th} singular value, then quickly decays to 10^{-4}.
3. Linear: $\mathbf{\Sigma}$ has diagonal entries that linearly decrease from 1 to 0.

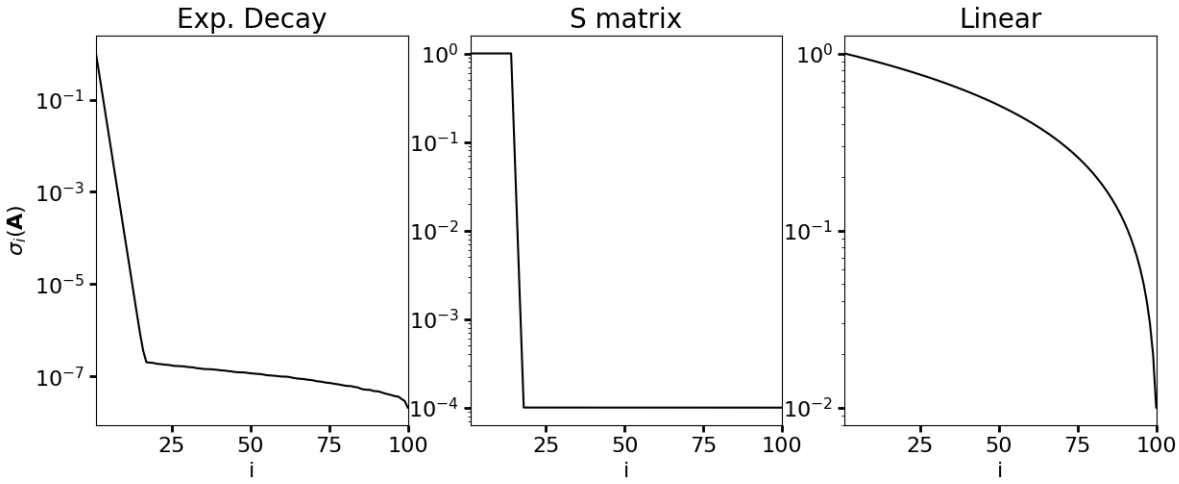

Figure 4.1: Singular value distributions of the three test matrices. The singular value decay of Exp. Decay matrix plateaus due to the perturbation by $\epsilon\mathbf{G}$.

To construct these matrices, orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ are generated by performing a QR factorization on a standard Gaussian matrix. This process gives singular vector matrices $\mathbf{U}$ and $\mathbf{V}$ that have Haar measure, which is undesirable. To disrupt this structure, we perturb the SVD so our test matrices are of the form $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}} + \epsilon\mathbf{G}$, where $\mathbf{\Sigma}$ is a diagonal matrix with the corresponding singular value distribution, $\epsilon = 10^{-8}$, and $\mathbf{G}$ is a standard Gaussian matrix. Due to this perturbation, the singular value distribution of our test matrices will not match the prescribed $\mathbf{\Sigma}$ exactly. However, since the perturbation is small, the resulting perturbed singular value distribution does not change significantly in terms of absolute error.

To demonstrate the tightness of the bounds, we run Algorithm 2 on these three test matrices. The bounds depend on the parameters chosen for Algorithm 2. We use a target rank of $k = 15$, which provides a well-defined singular value gap for the Exp. Decay and S matrices. For Theorems 11 and 6, we opt for a failure probability on the order of machine epsilon (in double precision), choosing $\Delta = 10^{-16}$. To choose our oversampling parameter p, we recall the term $F(\mathbf{A}, k, p, \Delta)$ that appears in Theorems 11 and 6,

$$F = \frac{3k}{p+1}\left(\frac{2}{\Delta}\right)^{\frac{2}{p}}\left(\|\mathbf{\Sigma}_\perp\|_F + \left[\sqrt{\ln\left(\frac{4}{\Delta^2}\right)} + \sqrt{k+p}\right]\sigma_{k+1}(\mathbf{A})\right)^2. \tag{4.10}$$

Given that $\Delta = 10^{-16}$, the factor $(2/\Delta)^{2/p}$ can be astronomically large without the appropriate amount of oversampling. We choose $p = 20$ so that $(2/\Delta)^{2/p} < 100$. Given the choice of $p = 20$, we can now compute the minimum q required for Theorem 11. For the Linear matrix,

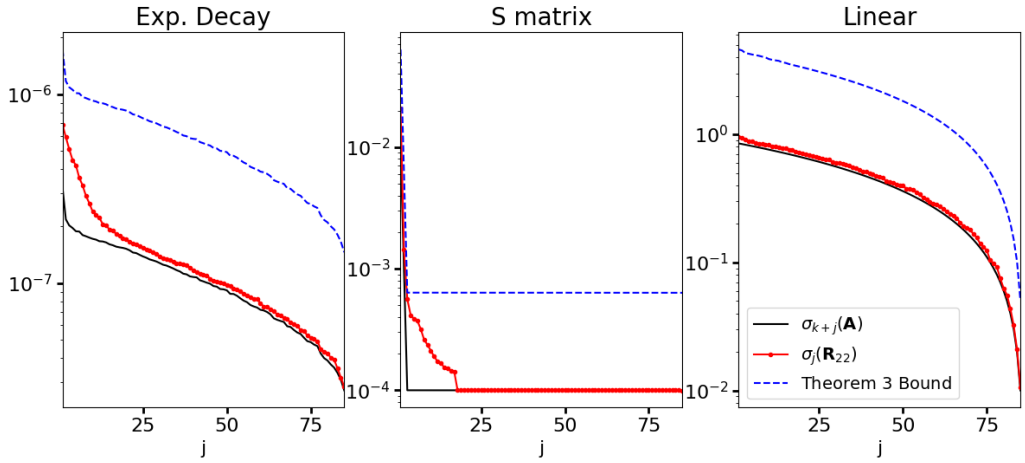

Figure 4.2: The computed singular values of $\mathbf{R}_{22}$ (red solid-with-dots), the lower bound given by the small singular values of $\mathbf{A}$ (black solid), and the upper bound given by Theorem 3 (blue dashed)

the minimum number of iterations needed for the bound to hold is $q = 503$; for the Exp. Decay matrix, we need $q = 6$ iterations, and for the S matrix, we need $q = 3$ iterations. The large number of iterations required for the Linear matrix is due to an insufficient gap after the k-th singular value. Taking 503 iterations for the analysis of the Linear matrix is impractical, so instead we choose $q = 6$ so that the analysis will work on the other two test matrices. Empirically, $q = 2$ is often enough to obtain a high accuracy using the randomized SVD with subspace iterations (Halko et al. 2011, §1.6), but our analysis suggests a few more iterations are necessary to guarantee the accuracy of the algorithm for all singular values.

We consider the tightness of the bounds for $\mathbf{R}_{22}$ that pertain to any realization of the Skeleton Algorithm Skeleton 1, given by Theorem 3. We repeat the bound below, but refer the readers to Theorem 3 for specifics on the necessary assumptions,

$$\sigma_{k+j}(\mathbf{A}) \leq \sigma_j(\mathbf{R}_{22}) \leq \sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2 \quad 1 \leq j \leq n-k.$$

Figure 4.2 depicts the accuracy of Algorithm 2 as well as the bounds given in Theorem 3. We observe that the bounds in Theorem 3 constrain the singular values of $\mathbf{R}_{22}$ to an interval slightly less than an order of magnitude in size. Additionally, the upper bound is typically tighter for the large singular values of $\mathbf{R}_{22}$ than for the small ones.

In Table 4.1, we present quantities related to the $\mathbf{R}_{22}$ bounds with oversampling in both the 2−norm and Frobenius norm. The $\|\mathbf{R}_{22}\|$ columns denote the corresponding norm of $\mathbf{R}_{22}$.

Table 4.1: A table comparing the norm of $\mathbf{R}_{22}$ with the Theorem 8 and Strong RRQR bounds for three test matrices.

	2−Norm			**Frobenius Norm**		
	$\|\mathbf{R}_{22}\|$	Thm. 8	Strong	$\|\mathbf{\Sigma}_{22}\|$	Thm. 8	Strong
Decay	$6.889 \cdot 10^{-7}$	$2.282 \cdot 10^{-5}$	$2.282 \cdot 10^{-5}$	$1.718 \cdot 10^{-6}$	$7.926 \cdot 10^{-5}$	$7.926 \cdot 10^{-5}$
S Matrix	0.02077	0.7142	0.7142	0.02086	0.7207	0.7207
Linear	0.9557	96235.5	60.708	4.9509	96236	325.993

The Theorem 8 columns show the upper bound given by

$$\|\mathbf{R}_{22}\|_\xi \leq \beta(n,k)\sqrt{\|\mathbf{\Sigma}_\perp\|_\xi^2 + \gamma^{4q}F^2},$$

in either the 2−norm or the Frobenius norm. Lastly, the columns labeled "Strong" display the strong RRQR bound in the corresponding norm,

$$\|\mathbf{R}_{22}\|_\xi \leq \beta(n,k)\|\Sigma_\perp\|_\xi.$$

For the Exp. Decay and S Matrix, the bounds from Theorem 6 perform just as well as the strong RRQR bound (to four decimal places), suggesting that for matrices with a sufficient singular value gap, Algorithm 2 is very close to a strong RRQR. However, the bounds from Theorem 6 do not demonstrate tightness for the Linear matrix in either norm, which is expected given its unsubstantial singular value gap. This indicates that the analysis is not well-suited for matrices lacking a low-rank structure.

We next consider the bounds for the singular values of $\mathbf{R}_{11}$ given by Theorems 9 and 11, as depicted in Figure 4.3. For the Exp. Decay and S matrix, the lower bound from Theorem 9 estimates the order of magnitude of the singular values of $\mathbf{R}_{11}$, and the bound from Theorem 11 is close to the strong RRQR bound. Since the S matrix has a large singular value gap ($\gamma \approx 1/10$), the bound from Theorem 11 is very tight, appearing indistinguishable from the strong RRQR bound. For the Linear matrix, we observe that Theorem 9 and Theorem 11 do not provide a lower bound on the singular values of $\mathbf{R}_{11}$. This occurs because randomized SVD performs suboptimally when the singular value gap of the matrix is small, causing the required assumption $\mathcal{R}(\mathbf{W}) \approx \mathcal{R}(\mathbf{V}_k)$ to fail. Empirically, Algorithm 2 still performs well on the Linear matrix, but since this matrix lacks a gap or low-rank structure, it is not well-suited for our analysis.

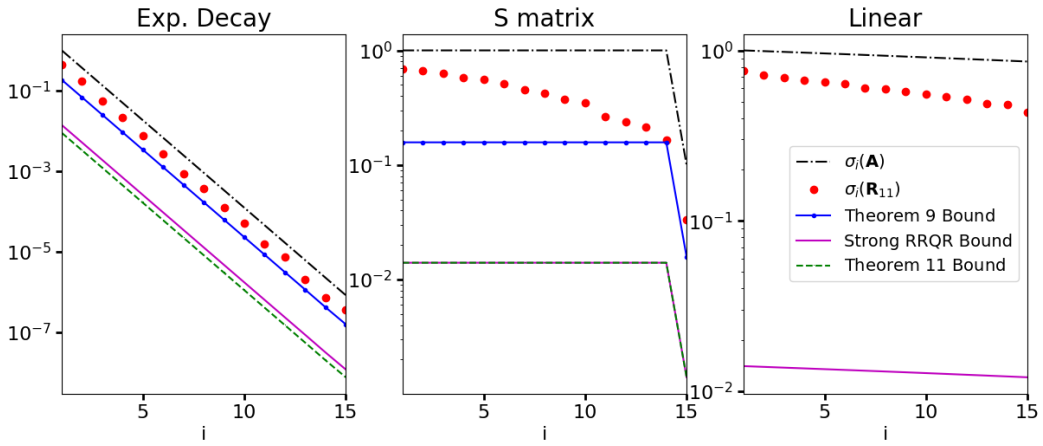

Figure 4.3: The computed singular values of $\mathbf{R}_{11}$ (red dots), the upper bound given by the singular values of $\mathbf{A}$ (black dash-dot), and the lower bounds given by Theorem 9 (blue dotted), Strong RRQR (purple solid), and Theorem 11 (green dashed).

4.4.2 Fiber Coating Data

When training the physics-constrained NODE on the fiber coating data, selecting a subset of time snapshots may improve both performance and efficiency. This selection can be achieved using an RRQR algorithm, such as Algorithm 2. In this subsection, we demonstrate the effectiveness of Algorithm 2 on numerical solutions to the fiber coating model, previously discussed in Chapter 3 (see (3.1)),

$$\frac{\partial}{\partial t}\left(h+\frac{\alpha}{2}h^2\right)+\frac{\partial}{\partial x}\left[M(h)\left(1-\frac{\partial}{\partial x}\left[Z(h)-\frac{\partial^2 h}{\partial x^2}\right]\right)\right]=0, \tag{4.11}$$

subject to the initial condition

$$h(x,0)=0.355+0.035\sin\left(\pi x/L\right)+0.02\sin\left(2\pi x/L\right)+0.01\sin\left(4\pi x/L\right)+0.025\sin\left(8\pi x/L\right)$$

and periodic boundary conditions. The system parameters $\alpha=2.299$ and $\eta=0.23105$ are identical to those used in Chapter 3. The evolution of numerical solutions to the PDE (4.11) with a domain size $L=20$ is depicted in Figure 4.4. The figure shows that the nearly flat initial data quickly develops multiple humps during the early stages (see the snapshot at $t=100$) and eventually transitions into a four-peak droplet system (see the snapshot at $t=500$). Here, we choose a larger domain ($L=20$) compared to the examples considered in Chapter 3 with $L=10$ (see Figure 3.1) to capture more interesting transient dynamics and inter-droplet interactions in the simulation data.

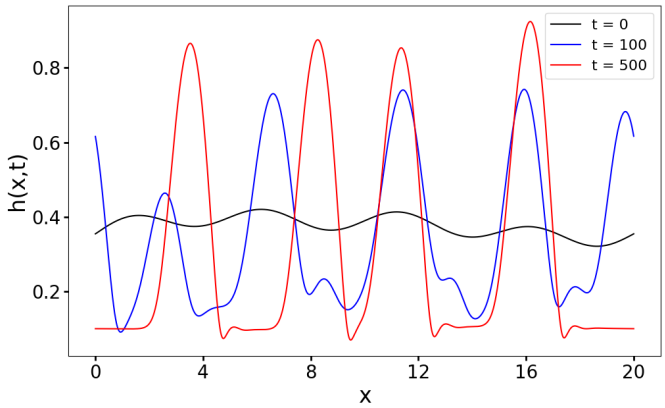

Figure 4.4: Profiles of the solution $h(x,t)$ to the fiber coating model (4.11) with $L = 20$, evolving into four-peak waves.

The fiber coating equation (4.11) is numerically solved using the given initial and boundary conditions with second-order centered finite differences and implicit time-stepping on a uniform grid, where the number of grid points is $N_x = 10001$ and the terminal time is $t_M = 1000$. We use the time interval $[200, 1000]$ with $M = 801$ uniformly spaced time snapshots and put the solution into a data matrix $\mathbf{A} \in \mathbb{R}^{10001\times 801}$, where the j^{th} column of $\mathbf{A}$ corresponds to the j^{th} time snapshot of the sample for $1 \leq j \leq 801$.

Then we perform an RRQR on the matrix $\mathbf{A}$ with a target rank $k = 23$. This choice of k gives an inverse singular value gap of $\gamma \approx 0.5764$. For the randomized subspace iterations, we choose an oversampling parameter $p = 20$ and use $q = 12$ subspace iterations, which is the minimum number required to satisfy the assumptions in Theorem 11. In practice, this number of subspace iterations is higher than conventional, but our analysis suggests that, given the relatively small singular value gap, an increased number of iterations may be necessary to bound the small singular values with high probability. For the strong rank-revealing QR, we use an SRRQR constant of $f = 2$.

Figure 4.5 shows the singular values of $\mathbf{R}_{11}$ (red dots) and $\mathbf{R}_{22}$ (red curves) produced by Algorithm 2. We also plot the minimal assumption bounds (green and blue curves) from Theorems 3 and 9, respectively, for the singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$. These plots show that the singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ provide reasonable approximations of the singular values of $\mathbf{A}$, demonstrating that Algorithm 2 performs well on the fiber coating data. Furthermore, the upper and lower bounds exhibit approximately an order of magnitude gap, confirming that the singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ remain within a small range relative to the singular values of $\mathbf{A}$.

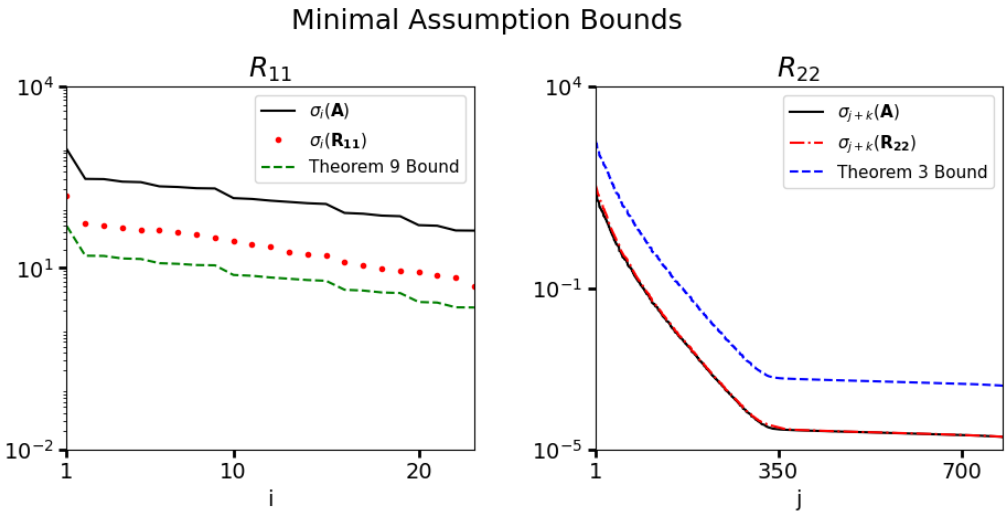

Figure 4.5: Plots of the singular values of $\mathbf{R}_{11}$ (red dots) and $\mathbf{R}_{22}$ (red dash-dotted line) produced by Algorithm 2, along with the bounds for the singular values provided by Theorem 3 (blue dashed), Theorem 9 (green dashed). The singular values of $\mathbf{A}$ (black solid) serve as an upper bound for the singular values of $\mathbf{R}_{11}$ and a lower bound for the singular values of $\mathbf{R}_{22}$.

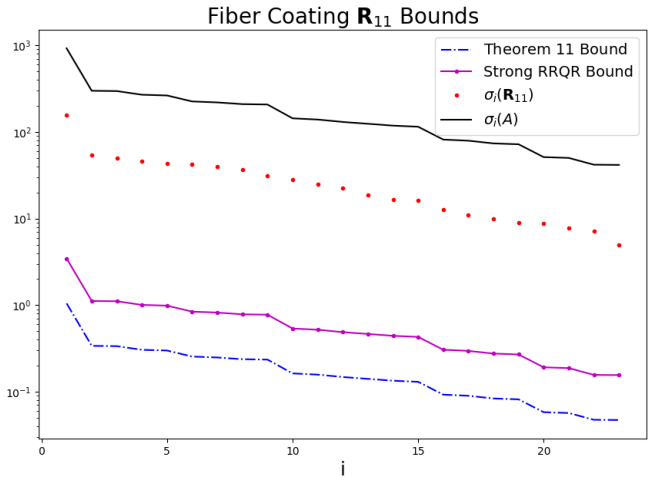

Figure 4.6: The computed singular values of $\mathbf{R}_{11}$ (red dots), the upper bound given by the singular values of $\mathbf{A}$ (black solid), and the lower bounds given by the Strong RRQR (purple dot-solid), and Theorem 11 (blue dot-dashed).

Table 4.2: A table comparing the norm of $\mathbf{R}_{22}$ with the Theorem 8 and Strong RRQR bounds for the Fiber Coating data.

	$\|\mathbf{R}_{22}\|$	Thm. 8	Strong
2−norm	42.5377	6422.61	6422.54
Fro. norm	95.2630	16574.77	16574.74

Next, we consider the $\mathbf{R}_{11}$ bound given in Theorem 11. Figure 4.6 depicts the singular values of $\mathbf{A}$ and $\mathbf{R}_{11}$, along with the lower bound from Theorem 11 and the strong RRQR lower bound. Since the fiber coating data matrix is 10001×801, the strong RRQR factor $\beta(n, k)$ is 267.54, yielding a lower bound that is not as tight as the upper bound given by the singular values of $\mathbf{A}$. The Theorem 11 bound falls below the strong RRQR bound, with the two bounds differing by about a factor of 3.

In Table 4.2, we present quantities related to the $\mathbf{R}_{22}$ bounds with oversampling in both the 2− and Frobenius norms. Similar to the table with the test matrices, all of these bounds are evaluated in both norms. The $\|\mathbf{R}_{22}\|$ columns denote the corresponding norm of $\mathbf{R}_{22}$, the Theorem 8 columns show the upper bound of from that theorem, and the "Strong" columns show the strong RRQR bound. From the table, we observe that the Theorem 8 bound does a much better job of matching the Strong RRQR bound than Theorem 11 did for $\mathbf{R}_{11}$. However, both the Theorem 8 bound and the Strong RRQR bound vastly overestimate the true norm due to the large multiplicative factor $\beta(n, k)$ present in both.

4.4.3 Synopsis of Numerical Experiments

To summarize the numerical results, the singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ provided by Algorithm 2 give good approximations of the singular values of $\mathbf{A}$ for all test matrices and the fiber coating data. The minimal assumption bounds from Theorems 3 and 9 provide tight bounds, ensuring that the computed singular values cannot differ significantly from the true singular values of $\mathbf{A}$. Furthermore, the probabilistic bounds for Algorithm 2 given by Theorems 11 and 8 are usually close to the strong RRQR bound, or at least differ by only a small factor, with one notable exception.

However, this analysis has its limitations, as highlighted by the Linear test matrix. In the absence of a reasonable singular value gap, several issues can arise, including:

1. The assumptions required by the $\mathbf{R}_{11}$ theorems may not hold.

2. The number of subspace iterations required for the analysis may become unreasonable.

3. The bound in Theorem 8 may not be close to the Strong RRQR bound.

Despite these challenges, Algorithm 2 performs well on matrices without a substantial singular value gap. However, for matrices lacking such a gap, this analysis is not advisable.

4.5 Proofs

In this section, we present proofs for the theorems in Section 4.3. To support these proofs, we first review the necessary background knowledge and auxiliary lemmas.

4.5.1 Preliminaries and Auxiliary Lemmas

Singular Value Inequalities

The singular values of matrices satisfy the following two inequalities (Horn and Johnson 1991, Theorem 3.3.16). If $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ with $m \geq n$ and $\mathbf{C} \in \mathbb{R}^{n\times \ell}$. Then the following inequalities hold,

$$\sigma_{j+k}(\mathbf{A}+\mathbf{B}) \leq \sigma_j(\mathbf{A}) + \sigma_{k+1}(\mathbf{B}) \qquad 1 \leq j \leq n,\ k+j \leq n. \tag{4.12}$$

$$\sigma_j(\mathbf{AC}) \leq \sigma_j(\mathbf{A})\|\mathbf{C}\|_2 \qquad 1 \leq j \leq n. \tag{4.13}$$

Eigenvalues

We continue with a discussion on eigenvalues. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\lambda \in \mathbb{C}$ and $\mathbf{x} \in \mathbb{R}^n$. Then we call λ an eigenvalue of $\mathbf{A}$ and $\mathbf{x}$ an eigenvector of $\mathbf{A}$ if $\mathbf{Ax} = \lambda\mathbf{x}$. We focus specifically on eigenvalues of symmetric matrices. If $\mathbf{A}$ is symmetric, then all of its eigenvalues are real and are ordered in non-increasing order, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. We say a symmetric matrix is positive-semidefinite if all its eigenvalues are greater than or equal to 0. Eigenvalues inequalities are often useful in bounding singular values due to a relationship between the eigenvalues and singular values of a matrix. One such relationship is (Trefethen and Bau 1997, Theorem 5.4).

$$\sigma_i^2(\mathbf{A}) = \lambda_i(\mathbf{A}^{\mathrm{T}}\mathbf{A}) \qquad 1 \leq i \leq n. \tag{4.14}$$

Symmetric matrices have a partial order relation among them. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ be symmetric. We define the Loewner partial order, denoted by $\preceq$, with the relation $\mathbf{A} \preceq \mathbf{B}$ if $\mathbf{B} - \mathbf{A}$ is symmetric positive-semidefinite. Below, we review two useful properties of the Loewner partial order.

Lemma 1. *(Horn and Johnson 2013, Theorem 7.7.2a, Corollary 7.7.4c) If* $\mathbf{A} \preceq \mathbf{B}$ *and* $\mathbf{S} \in \mathbb{R}^{n \times n}$*, then*

$$\mathbf{S}\mathbf{A}\mathbf{S}^{\mathrm{T}} \preceq \mathbf{S}\mathbf{B}\mathbf{S}^{\mathrm{T}} \tag{4.15}$$

$$\lambda_i(\mathbf{A}) \leq \lambda_i(\mathbf{B}) \qquad 1 \leq i \leq n. \tag{4.16}$$

It follows from (4.16) that if $\mathbf{A} \preceq \mathbf{B}$,

$$\operatorname{trace}(\mathbf{A}) \leq \operatorname{trace}(\mathbf{B}). \tag{4.17}$$

The following eigenvalue inequality is attributed to Aronszajn (Bhatia 1997, Theorem III.2.9).

Lemma 2 (Aronszajn's Inequality)**.** *Let* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *be a symmetric matrix partitioned as*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^{\mathrm{T}} & \mathbf{A}_{22} \end{bmatrix}, \tag{4.18}$$

where $\mathbf{A}_{11} \in \mathbb{R}^{k \times k}$ *and* $\mathbf{A}_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$*. Then*

$$\lambda_{i+j-1}(\mathbf{A}) + \lambda_n(\mathbf{A}) \leq \lambda_i(\mathbf{A}_{11}) + \lambda_j(\mathbf{A}_{22}). \quad 1 \leq i \leq k, \quad 1 \leq j \leq n-k.$$

We now look at a specific application of Aronszajn's Inequality.

Corollary 1. *Let* $\mathbf{A}$ *be symmetric positive-semidefinite and partitioned as in* (4.18)*. Then*

$$\lambda_j(\mathbf{A}) \leq \lambda_1(\mathbf{A}_{11}) + \lambda_j(\mathbf{A}_{22}), \qquad 1 \leq j \leq n-k. \tag{4.19}$$

Proof. Applying Lemma 2 to $\mathbf{A}$ with $i = 1$ gives

$$\lambda_j(\mathbf{A}) + \lambda_n(\mathbf{A}) \leq \lambda_1(\mathbf{A}_{11}) + \lambda_j(\mathbf{A}_{22}), \qquad 1 \leq j \leq n-k.$$

Since $\mathbf{A}$ is symmetric positive-semidefinite, $\lambda_n(\mathbf{A})$ is non-negative and

$$\lambda_j(\mathbf{A}) \leq \lambda_j(\mathbf{A}) + \lambda_n(\mathbf{A}) \leq \lambda_1(\mathbf{A}_{11}) + \lambda_j(\mathbf{A}_{22}), \qquad 1 \leq j \leq n-k.$$

□

Majorization

We move on to discuss majorization results regarding principal angles between subspaces. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and let $x_1^{\downarrow} \geq \ldots \geq x_n^{\downarrow}$ denote the entries of $\mathbf{x}$ arranged in decreasing order. We

say $\mathbf{x} \prec_w \mathbf{y}$, or $\mathbf{y}$ weakly majorizes $\mathbf{x}$, if

$$\sum_{j=1}^{\ell} x_j^{\downarrow} \leq \sum_{j=1}^{\ell} y_j^{\downarrow} \qquad 1 \leq \ell \leq n.$$

We recall a result regarding the perturbation of the principal angles.

Lemma 3. *(Knyazev and Argentati 2007, Theorem* 3.2*) Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ be $k-$dimensional subspaces of $\mathbb{R}^n$. Then*

$$|\cos \mathbf{\Phi}(\mathcal{X}, \mathcal{Z}) - \cos \mathbf{\Phi}(\mathcal{Y}, \mathcal{Z})| \prec_w \sin \mathbf{\Phi}(\mathcal{X}, \mathcal{Y}),$$

where $|\cdot|$ is the component-wise absolute value. This lemma implies the following corollary regarding the largest principal angles between these subspaces.

Corollary 2. *Under the assumptions of Lemma 3,*

$$|\cos \varphi_k(\mathcal{X}, \mathcal{Z}) - \cos \varphi_k(\mathcal{Y}, \mathcal{Z})| \leq \sin \varphi_k(\mathcal{X}, \mathcal{Y}). \tag{4.20}$$

Proof. Since $\cos \varphi_k(\mathcal{X}, \mathcal{Z}) - \cos \varphi_k(\mathcal{Y}, \mathcal{Z})$ is a component of the vector $\cos \mathbf{\Phi}(\mathcal{X}, \mathcal{Z}) - \cos \mathbf{\Phi}(\mathcal{Y}, \mathcal{Z})$,

$$|\cos \varphi_k(\mathcal{X}, \mathcal{Z}) - \cos \varphi_k(\mathcal{Y}, \mathcal{Z})| \leq \|\cos \mathbf{\Phi}(\mathcal{X}, \mathcal{Z}) - \cos \mathbf{\Phi}(\mathcal{Y}, \mathcal{Z})\|_\infty. \tag{4.21}$$

If $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$ with $\mathbf{x} \prec_w \mathbf{y}$ and $\mathbf{x}$ has non-negative entries, then $\|\mathbf{x}\|_\infty \leq \|\mathbf{y}\|_\infty$ (Bhatia 1997, p.45). Thus, Lemma 3 gives

$$\|\cos \mathbf{\Phi}(\mathcal{X}, \mathcal{Z}) - \cos \mathbf{\Phi}(\mathcal{Y}, \mathcal{Z})\|_\infty \leq \|\sin \mathbf{\Phi}(\mathcal{X}, \mathcal{Y})\|_\infty.$$

Combining this with (4.21) yields

$$|\cos \varphi_k(\mathcal{X}, \mathcal{Z}) - \cos \varphi_k(\mathcal{Y}, \mathcal{Z})| \leq \|\sin \mathbf{\Phi}(\mathcal{X}, \mathcal{Y})\|_\infty. \tag{4.22}$$

Since $\sin(\varphi)$ is an increasing function on $\left[0, \frac{\pi}{2}\right]$, $\|\sin \mathbf{\Phi}(\mathcal{X}, \mathcal{Y})\|_\infty = \sin \varphi_k(\mathcal{X}, \mathcal{Y})$. Substituting this into 4.22 gives the result.

□

Structural Lemmas

The following lemmas are necessary in proving the structural bounds for Algorithm 2. We begin with a modified version of the Eckart-Young theorem, similar to (Gu 2015, Theorem 3.5).

Lemma 4. *Let* $\mathbf{A}$ *satisfy the Assumptions,* $\mathbf{Q} \in \mathbb{R}^{n\times(k+p)}$ *have orthonormal columns and let* $\mathbf{B}_k$ *be the rank-k truncated SVD of* $\mathbf{AQ}$*. Then* $\mathbf{B}_k$ *is an optimal solution to*

$$\min_{\substack{\mathbf{C}\in\mathbb{R}^{m\times(k+p)}\\ \operatorname{rank}(\mathbf{C})\leq k}} \|\mathbf{A}-\mathbf{C}\mathbf{Q}^{\mathrm{T}}\|_F = \|\mathbf{A}-\mathbf{B}_k\mathbf{Q}^{\mathrm{T}}\|_F.$$

Proof. We consider the squared version of the objective function,

$$\|\mathbf{A}-\mathbf{C}\mathbf{Q}^{\mathrm{T}}\|_F^2 = \|\mathbf{A}(\mathbf{I}-\mathbf{Q}\mathbf{Q}^{\mathrm{T}})+(\mathbf{A}\mathbf{Q}-\mathbf{C})\mathbf{Q}^{\mathrm{T}}\|_F^2.$$

Since $(\mathbf{I}-\mathbf{Q}\mathbf{Q}^{\mathrm{T}})\mathbf{Q}=\mathbf{0}$, matrix Pythagoras (Drineas and Mahoney 2017, Lemma 5) yields

$$\|\mathbf{A}-\mathbf{C}\mathbf{Q}^{\mathrm{T}}\|_F^2 = \|\mathbf{A}(\mathbf{I}-\mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_F^2+\|(\mathbf{A}\mathbf{Q}-\mathbf{C})\mathbf{Q}^{\mathrm{T}}\|_F^2.$$

The term $\|\mathbf{A}(\mathbf{I}-\mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_F$ is independent of $\mathbf{C}$, so the matrix $\mathbf{C}$ that minimizes $\|(\mathbf{A}\mathbf{Q}-\mathbf{C})\mathbf{Q}^{\mathrm{T}}\|_F^2$ also minimizes $\|\mathbf{A}-\mathbf{C}\mathbf{Q}^{\mathrm{T}}\|_F^2$. By the orthogonal invariance of singular values and (2.32), we know that the desired matrix is the rank-k truncated SVD of $\mathbf{AQ}$, which is $\mathbf{B}_k$.

□

The following lemma is useful in the analysis of the randomized SVD with subspace iterations, and its statement and proof come from the analysis in (Halko et al. 2011, §9.2)

Lemma 5 (Halko et al. (2011) §9.2)**.** *Let* $\mathbf{\Omega} \in \mathbb{R}^{m\times(k+p)}$ *and consider the partition* $\mathbf{U}^{\mathrm{T}}\mathbf{\Omega}$ *given by*

$$\mathbf{U}^{\mathrm{T}}\mathbf{\Omega} = \begin{bmatrix}\mathbf{\Omega}_1\\ \mathbf{\Omega}_2\end{bmatrix},$$

where $\mathbf{\Omega}_1 \in \mathbb{R}^{k\times(k+p)}$ *and* $\mathbf{\Omega}_2 \in \mathbb{R}^{(m-k)\times(k+p)}$*. Assume* $\mathbf{\Omega}_1$ *has full rank. Let*

$$\mathbf{F} := \mathbf{\Sigma}_\perp^{2q+1}\mathbf{\Omega}_2\mathbf{\Omega}_1^{\dagger}\mathbf{\Sigma}_k^{-(2q+1)} \qquad \textit{and} \qquad \mathbf{Z} := \begin{bmatrix}\mathbf{I}\\ \mathbf{F}\end{bmatrix}.$$

Let $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^q\mathbf{A}^{\mathrm{T}}\mathbf{\Omega} = \mathbf{Q}\mathbf{R}$ *be a thin QR factorization. Then*

$$\mathbf{Z}\mathbf{Z}^{\dagger} \preceq \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V}.$$

Proof. We begin by substituting the SVD of $\mathbf{A}$ into $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^q\mathbf{A}^{\mathrm{T}}\mathbf{\Omega} = \mathbf{QR}$ to obtain $\mathbf{QR} = \mathbf{V}\mathbf{\Sigma}^{2q+1}\mathbf{U}^{\mathrm{T}}\mathbf{\Omega}$. Premultiplying by $\mathbf{V}^{\mathrm{T}}$ and partitioning $\mathbf{\Sigma}$ and $\mathbf{U}^{\mathrm{T}}\mathbf{\Omega}$ yields

$$\mathbf{V}^{\mathrm{T}}\mathbf{QR} = \begin{bmatrix} \mathbf{\Sigma}_k^{2q+1} & \\ & \mathbf{\Sigma}_\perp^{2q+1} \end{bmatrix} \begin{bmatrix} \mathbf{\Omega}_1 \\ \mathbf{\Omega}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_k^{2q+1}\mathbf{\Omega}_1 \\ \mathbf{\Sigma}_\perp^{2q+1}\mathbf{\Omega}_2 \end{bmatrix}.$$

Since $k \leq \operatorname{rank}(\mathbf{A})$, $\mathbf{\Sigma}_k$ is nonsingular. Thus, we can postmultiply by $\mathbf{\Omega}_1^\dagger\mathbf{\Sigma}_k^{-(2q+1)}$ to obtain

$$\mathbf{V}^{\mathrm{T}}\mathbf{QR}\mathbf{\Omega}_1^\dagger\mathbf{\Sigma}_k^{-(2q+1)} = \begin{bmatrix} \mathbf{I} \\ \mathbf{\Sigma}_\perp^{2q+1}\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\mathbf{\Sigma}_k^{-(2q+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{F} \end{bmatrix} = \mathbf{Z}.$$

We now consider $\mathbf{P}_{\mathbf{Z}}$ and $\mathbf{P}_{\mathbf{V}^{\mathrm{T}}\mathbf{Q}}$, the orthogonal projectors onto $\mathcal{R}(\mathbf{Z})$ and $\mathcal{R}(\mathbf{V}^{\mathrm{T}}\mathbf{Q})$, respectively. Since $\mathbf{P}_{\mathbf{Z}}$ is an orthogonal projector, $\mathbf{P}_{\mathbf{Z}} \preceq \mathbf{I}$. The conjugation rule (4.15) yields

$$\mathbf{P}_{\mathbf{V}^{\mathrm{T}}\mathbf{Q}}\mathbf{P}_{\mathbf{Z}}\mathbf{P}_{\mathbf{V}^{\mathrm{T}}\mathbf{Q}} \preceq \mathbf{P}_{\mathbf{V}^{\mathrm{T}}\mathbf{Q}}. \tag{4.23}$$

From the expression of $\mathbf{Z}$, we have $\mathcal{R}(\mathbf{Z}) \subset \mathcal{R}(\mathbf{V}^{\mathrm{T}}\mathbf{Q})$. This implies that

$$\mathbf{P}_{\mathbf{V}^{\mathrm{T}}\mathbf{Q}}\mathbf{P}_{\mathbf{Z}}\mathbf{P}_{\mathbf{V}^{\mathrm{T}}\mathbf{Q}} = \mathbf{P}_{\mathbf{Z}}.$$

Substituting this equation into (4.23) gives $\mathbf{P}_{\mathbf{Z}} \preceq \mathbf{P}_{\mathbf{V}^{\mathrm{T}}\mathbf{Q}}$. Using the expressions for orthogonal projectors, we obtain $\mathbf{Z}\mathbf{Z}^\dagger \preceq \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V}$.

□

Probabilistic Lemmas

In order to prove Theorems 5, 8, and 11, we first need to give some probabilistic results. Our probabilistic analysis is similar to that in (Halko et al. 2011, §10). We begin by giving a large deviation bound for the 2-norm of the inverse of a Gaussian matrix.

Lemma 6 (Edelman)**.** *(Sankar et al. 2003, Theorem* 3.4*) Let* $\mathbf{G} \in \mathbb{R}^{n\times n}$ *be a random Gaussian matrix with variance* σ*, then for* $x > 0$*,*

$$\mathbb{P}\left(\left\|\mathbf{G}^{-1}\right\|_2 \geq x\right) \leq \frac{\sqrt{n}}{x\sigma}.$$

The next two results are tools to carry out our probabilistic analysis.

Lemma 7. *(Halko et al. 2011, Prop.* 10.1*) For fixed matrices* $\mathbf{S}$*,* $\mathbf{T}$ *and standard Gaussian matrix* $\mathbf{\Omega}$*, we have*

$$\mathbb{E}\|\mathbf{S}\mathbf{\Omega}\mathbf{T}\|_2 \leq \|\mathbf{S}\|_2\|\mathbf{T}\|_F + \|\mathbf{S}\|_F\|\mathbf{T}\|_2.$$

Lemma 8. *(Halko et al. 2011, Prop. 10.3) Let $\mathbf{\Omega} \in \mathbb{R}^{m\times n}$ be a standard Gaussian matrix and $h : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a Lipschitz function with Lipschitz constant $L > 0$, i.e.*

$$|h(\mathbf{X}) - h(\mathbf{Y})| \leq L\|\mathbf{X} - \mathbf{Y}\|_F \qquad \mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m\times n}.$$

Then for any $t > 0$,

$$\mathbb{P}(h(\mathbf{\Omega}) \geq \mathbb{E}(h(\mathbf{\Omega})) + Lt) \leq e^{-\frac{t^2}{2}}.$$

With these theorems, we can now give probabilistic bounds for the quantity $\|\mathbf{Z}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2$, where $\mathbf{Z} \in \mathbb{R}^{(n-k)\times(n-k)}$ is arbitrary.

Lemma 9. *Let $\mathbf{Z} \in \mathbb{R}^{(n-k)\times(n-k)}$, and let $\mathbf{\Omega}_1 \in \mathbb{R}^{k\times k}$ and $\mathbf{\Omega}_2 \in \mathbb{R}^{(n-k)\times k}$ be standard random Gaussian matrices with $\mathbf{\Omega}_1$ nonsingular. Then for any probability $0 < \Delta < 1$,*

$$\mathbb{P}\left(\|\mathbf{Z}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2 \geq \frac{2k}{\Delta}\left[\|\mathbf{Z}\|_F + \left(\sqrt{k} + \sqrt{\ln\left(\frac{4}{\Delta^2}\right)}\right)\|\mathbf{Z}\|_2\right]\right) \leq \Delta.$$

Proof. Since $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are independent, we can analyze how the error depends on $\mathbf{\Omega}_2$ by conditioning on $\mathbf{\Omega}_1$. We define the event

$$E_x = \{\mathbf{\Omega}_1 : \|\mathbf{\Omega}_1^{-1}\|_2 \leq x\}.$$

By Lemma 6, we have

$$\mathbb{P}(E_x^{\mathrm{c}}) \leq \frac{\sqrt{k}}{x}.$$

Consider the function $h : \mathbb{R}^{(n-k)\times k} \to \mathbb{R}$ defined by

$$h(\mathbf{X}) = \|\mathbf{Z}\mathbf{X}\mathbf{\Omega}_1^{-1}\|_2.$$

We now show that h is Lipschitz. Indeed, we use the reverse triangle inequality to obtain

$$|h(\mathbf{X}) - h(\mathbf{Y})| = \left|\|\mathbf{Z}\mathbf{X}\mathbf{\Omega}_1^{-1}\|_2 - \|\mathbf{Z}\mathbf{Y}\mathbf{\Omega}_1^{-1}\|_2\right| \leq \|\mathbf{Z}\mathbf{X}\mathbf{\Omega}_1^{-1} - \mathbf{Z}\mathbf{Y}\mathbf{\Omega}_1^{-1}\|_2.$$

Factoring and using submultiplicativity, we obtain

$$|h(\mathbf{X}) - h(\mathbf{Y})| \leq \|\mathbf{Z}(\mathbf{X} - \mathbf{Y})\mathbf{\Omega}_1^{-1}\|_2 \leq \|\mathbf{Z}\|_2\|\mathbf{\Omega}_1^{-1}\|_2\|\mathbf{X} - \mathbf{Y}\|_2.$$

Therefore, h is Lipschitz with constant $L = \|\mathbf{Z}\|_2\|\mathbf{\Omega}_1^{-1}\|_2$. Applying Lemma 7 and using

$\|\mathbf{\Omega}_1^{-1}\|_F \leq \sqrt{k}\|\mathbf{\Omega}_1^{-1}\|_2$, we have

$$\mathbb{E}\|\mathbf{Z}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2 \leq \left(\sqrt{k}\|\mathbf{Z}\|_2 + \|\mathbf{Z}\|_F\right)\|\mathbf{\Omega}_1^{-1}\|_2. \tag{4.24}$$

Given E_x, we have $\|\mathbf{\Omega}_1^{-1}\|_1 \leq x$. Thus, we can combine (4.24) and Lemma 8 to obtain

$$\mathbb{P}\left(\|\mathbf{Z}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2 \geq \left[\left(\sqrt{k}+t\right)\|\mathbf{Z}\|_2 + \|\mathbf{Z}\|_F\right]x \;\middle|\; E_x\right) \leq e^{-\frac{t^2}{2}}.$$

We remove the conditioning on $\mathbf{\Omega}_1^{-1}$ to obtain

$$\mathbb{P}\left(\|\mathbf{Z}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2 \geq \left[\left(\sqrt{k}+t\right)\|\mathbf{Z}\|_2 + \|\mathbf{Z}\|_F\right]x\right) \leq e^{-\frac{t^2}{2}} + \frac{\sqrt{k}}{x}.$$

We now let $0 < \Delta < 1$ and choose $t = \sqrt{\ln\left(\frac{4}{\Delta^2}\right)}$ and $x = \frac{2\sqrt{k}}{\Delta}$. These choices make $e^{-\frac{t^2}{2}} + \frac{\sqrt{k}}{x} = \Delta$, and we obtain the result,

$$\mathbb{P}\left(\|\mathbf{Z}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2 \geq \frac{2\sqrt{k}}{\Delta}\left[\|\mathbf{Z}\|_F + \left(\sqrt{k} + \sqrt{\ln\left(\frac{4}{\Delta^2}\right)}\right)\|\mathbf{Z}\|_2\right]\right) \leq \Delta.$$

□

4.5.2 Proof of Theorem 2

Proof. We prove this theorem by proving the bounds in (4.1) and (4.2) separately. We begin with (4.1), the lower bound for the singular values of $\mathbf{R}_{11}$. By (2.49), the singular values of $\mathbf{R}_{11}$ are the same as the singular values of $\mathbf{A}_1 = \mathbf{A}\mathbf{\Pi}_1$. Thus, it is sufficient to show

$$\frac{\sigma_i(\mathbf{A})}{\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} \leq \sigma_i(\mathbf{A}\mathbf{\Pi}_1) \qquad 1 \leq i \leq k.$$

Using the SVD $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ and the fact that singular values are orthogonally invariant,

$$\sigma_i(\mathbf{A}\mathbf{\Pi}_1) = \sigma_i(\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}\mathbf{\Pi}_1) \qquad 1 \leq i \leq k. \tag{4.25}$$

Now we show $\sigma_i(\mathbf{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1) \leq \sigma_i(\mathbf{A}\mathbf{\Pi}_1)$. Partitioning $\mathbf{\Sigma}$ and $\mathbf{V}$ as in (2.30) gives

$$\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}\mathbf{\Pi}_1 = \begin{bmatrix}\mathbf{\Sigma}_k & \\ & \mathbf{\Sigma}_\perp\end{bmatrix}\begin{bmatrix}\mathbf{V}_k^{\mathrm{T}} \\ \mathbf{V}_\perp^{\mathrm{T}}\end{bmatrix}\mathbf{\Pi}_1 = \begin{bmatrix}\mathbf{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1 \\ \mathbf{\Sigma}_\perp\mathbf{V}_\perp^{\mathrm{T}}\mathbf{\Pi}_1\end{bmatrix}.$$

Since the singular values of a submatrix do not exceed the singular values of the matrix

(Horn and Johnson 1991, Corollary 3.1.3), this implies

$$\sigma_i(\mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{T}} \mathbf{\Pi}_1) \leq \sigma_i(\mathbf{\Sigma} \mathbf{V}^{\mathrm{T}} \mathbf{\Pi}_1) \qquad 1 \leq i \leq k.$$

Combining this with (4.25) yields

$$\sigma_i(\mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{T}} \mathbf{\Pi}_1) \leq \sigma_i(\mathbf{A} \mathbf{\Pi}_1) \qquad 1 \leq i \leq k. \tag{4.26}$$

Now, we write

$$\sigma_i(\mathbf{A}) = \sigma_i(\mathbf{\Sigma}_k) = \sigma_i(\mathbf{\Sigma}_k (\mathbf{V}_k^{\mathrm{T}} \mathbf{\Pi}_1)(\mathbf{V}_k^{\mathrm{T}} \mathbf{\Pi}_1)^{-1}) \qquad 1 \leq i \leq k.$$

Applying the multiplicative inequality (4.13) and (4.26) yields

$$\sigma_i(\mathbf{A}) \leq \sigma_i(\mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{T}} \mathbf{\Pi}_1) \|(\mathbf{V}_k^{\mathrm{T}} \mathbf{\Pi}_1)^{-1}\|_2 \leq \sigma_i(\mathbf{A} \mathbf{\Pi}_1) \|(\mathbf{V}_k^{\mathrm{T}} \mathbf{\Pi}_1)^{-1}\|_2 \qquad 1 \leq i \leq k.$$

Dividing both sides by $\|(\mathbf{V}_k^{\mathrm{T}} \mathbf{\Pi}_1)^{-1}\|_2$ proves (4.1).

We move on to proving (4.2). Let $\mathbf{A\Pi} = \mathbf{QR}$ be partitioned as in (2.44). We prove the statement as follows: We first equate the singular values of $\mathbf{R}_{22}$ and $\mathbf{Q}_2^T \mathbf{A}$. Then we introduce a special oblique projector to obtain an equivalent expression for $\mathbf{Q}_2^{\mathrm{T}} \mathbf{A}$. The proof is completed by using singular value inequalities on this new expression.

Multiplying the right-hand side in (2.44) yields

$$\begin{bmatrix} \mathbf{A\Pi}_1 & \mathbf{A\Pi}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 \mathbf{R}_{11} & \mathbf{Q}_1 \mathbf{R}_{12} + \mathbf{Q}_2 \mathbf{R}_{22} \end{bmatrix}.$$

This gives $\mathbf{A\Pi}_1 = \mathbf{Q}_1 \mathbf{R}_{11}$ and $\mathbf{A\Pi}_2 = \mathbf{Q}_1 \mathbf{R}_{12} + \mathbf{Q}_2 \mathbf{R}_{22}$. We use these equations to relate the singular values of $\mathbf{R}_{22}$ and $\mathbf{Q}_2^{\mathrm{T}} \mathbf{A}$. From

$$\mathbf{Q}_2^{\mathrm{T}} \mathbf{A\Pi} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ & \mathbf{R}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{R}_{22} \end{bmatrix}$$

follows

$$\sigma_j(\mathbf{Q}_2^{\mathrm{T}} \mathbf{A\Pi}) = \sigma_j\left(\begin{bmatrix} \mathbf{0} & \mathbf{R}_{22} \end{bmatrix}\right) = \sigma_j(\mathbf{R}_{22}) \qquad 1 \leq j \leq n-k.$$

Since singular values are orthogonally invariant,

$$\sigma_j(\mathbf{R}_{22}) = \sigma_j(\mathbf{Q}_2^{\mathrm{T}} \mathbf{A}) \qquad 1 \leq j \leq n-k. \tag{4.27}$$

Equation (4.27) states that it is sufficient to focus on $\mathbf{Q}_2^{\mathrm{T}} \mathbf{A}$ instead. We obtain the desired

bounds on the singular values of $\mathbf{Q}_2^{\mathrm{T}}\mathbf{A}$ by introducing a certain oblique projector.

Let $\mathbf{P} \in \mathbb{R}^{n\times n}$ be the oblique projector onto $\mathcal{R}(\mathbf{\Pi}_1)$ along $\mathcal{R}(\mathbf{V}_k)$, given by the expression

$$\mathbf{P} := \mathbf{\Pi}_1(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\mathbf{V}_k^{\mathrm{T}}.$$

We observe $\mathbf{Q}_2^{\mathrm{T}}\mathbf{A}\mathbf{P} = \mathbf{0}$ and $\mathbf{A}_k\mathbf{P} = \mathbf{A}_k$. These two facts yield

$$\mathbf{Q}_2^{\mathrm{T}}\mathbf{A} = \mathbf{Q}_2^{\mathrm{T}}(\mathbf{A} - \mathbf{A}_k)(\mathbf{I} - \mathbf{P}).$$

Combining this with (4.27), we obtain

$$\sigma_j(\mathbf{R}_{22}) = \sigma_j(\mathbf{Q}_2^{\mathrm{T}}(\mathbf{A} - \mathbf{A}_k)(\mathbf{I} - \mathbf{P})) \qquad 1 \le j \le n-k.$$

Applying the multiplicative inequality (4.13) gives

$$\sigma_j(\mathbf{R}_{22}) \le \sigma_j(\mathbf{A} - \mathbf{A}_k)\|\mathbf{Q}_2^{\mathrm{T}}\|_2\|(\mathbf{I} - \mathbf{P})\|_2 \qquad 1 \le j \le n-k.$$

Since $\mathbf{Q}_2^{\mathrm{T}}$ has orthonormal rows, its $2-$norm is equal to 1. In addition, $\mathrm{rank}(\mathbf{P}) = k$, so $\mathbf{P}$ is a non-zero, non-identity projector, so by (Szyld 2006, Therorem 2.1), $\|\mathbf{I} - \mathbf{P}\|_2 = \|\mathbf{P}\|_2$ and thus,

$$\sigma_j(\mathbf{R}_{22}) \le \sigma_j(\mathbf{A} - \mathbf{A}_k)\|\mathbf{\Pi}_1(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\mathbf{V}_k^{\mathrm{T}}\|_2 \qquad 1 \le j \le n-k.$$

The computation of $\sigma_j(\mathbf{A} - \mathbf{A}_k)$ and the orthogonal invariance of the 2-norm gives the desired result,

$$\sigma_j(\mathbf{R}_{22}) \le \sigma_{k+j}(\mathbf{A})\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2 \qquad 1 \le j \le n-k.$$

□

4.5.3 Proofs of Singular Value Bounds for $\mathbf{R}_{22}$

In this subsection, we present the proofs of the theorems provided in §4.3.1 on the singular value bounds for $\mathbf{R}_{22}$. We show the proofs in the order in which the corresponding theorems appear in §4.3.1.

Proof of Theorem 3 ($\mathbf{R}_{22}$ bounds for Algorithm Skeleton 1)

Proof. This proof is similar in structure to the proof of (4.2) in Theorem 2 Let $\mathbf{P}$ be the oblique projector onto $\mathcal{R}(\mathbf{\Pi}_1)$ along $\mathcal{R}(\mathbf{W})$, which takes the form

$$\mathbf{P} := \mathbf{\Pi}_1(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\mathbf{W}^{\mathrm{T}}.$$

It is clear that $\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{P} = \mathbf{W}\mathbf{W}^{\mathrm{T}}$, and since $\mathbf{Q}_2\mathbf{A}\mathbf{\Pi}_1 = \mathbf{0}$, it follows that $\mathbf{Q}_2^{\mathrm{T}}\mathbf{A}\mathbf{P} = \mathbf{0}$ as well. These two facts yield

$$\mathbf{Q}_2^{\mathrm{T}}\mathbf{A} = \mathbf{Q}_2^{\mathrm{T}}\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})(\mathbf{I} - \mathbf{P}).$$

Recall from (4.27) that $\mathbf{R}_{22}$ and $\mathbf{Q}_2^{\mathrm{T}}\mathbf{A}$ have the same singular values. So

$$\sigma_j(\mathbf{R}_{22}) = \sigma_j(\mathbf{Q}_2^{\mathrm{T}}\mathbf{A}) = \sigma_j(\mathbf{Q}_2^{\mathrm{T}}\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})(\mathbf{I} - \mathbf{P})) \qquad 1 \leq j \leq n - k.$$

Applying the multiplicative singular value inequality (4.13) gives

$$\sigma_j(\mathbf{R}_{22}) \leq \sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))\|\mathbf{Q}_2^{\mathrm{T}}\|_2\|(\mathbf{I} - \mathbf{P})\|_2 \qquad 1 \leq j \leq n - k.$$

Since $\mathbf{Q}_2^{\mathrm{T}}$ is a submatrix of an orthogonal matrix, its norm is at most 1. Since $\mathbf{P} \in \mathbb{R}^{n\times n}$ rank-k projection, it is non-zero, and not the identity matrix, so by (Szyld 2006, Theorem 2.1), $\|\mathbf{I} - \mathbf{P}\|_2 = \|\mathbf{P}\|_2$ and thus,

$$\sigma_j(\mathbf{R}_{22}) \leq \sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))\|\mathbf{P}\|_2 \qquad 1 \leq j \leq n - k.$$

The orthogonal invariance of the 2-norm implies $\|\mathbf{P}\|_2 = \|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2$, which, when substituted into the above equation completes the proof.

□

Proof of Theorem 4 (Structural $\mathbf{R}_{22}$ Bounds for Algorithm 2)

Proof. By Proposition 1, $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular and Theorem 3 gives

$$\sigma_j(\mathbf{R}_{22}) \leq \sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2 \qquad 1 \leq j \leq n - k.$$

Furthermore, Proposition 1 also gives that $\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2 \leq \beta(n,k)$. Thus,

$$\sigma_j(\mathbf{R}_{22}) \leq \beta(n,k)\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})) \qquad 1 \leq j \leq n - k.$$

From here, it suffices to prove

$$\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})) \leq \sqrt{\sigma_{j+k}(\mathbf{A})^2 + \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2^2}. \tag{4.28}$$

The proof of (4.28) is done in several steps. We begin by finding a block matrix with the same eigenvalues as $\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})$. We then apply Corollary 1 to bound the eigenvalues of that block matrix using the eigenvalues of its diagonal blocks, and finish the proof by using

the Loewner partial order to bound the eigenvalues of the diagonal blocks.

We begin by considering the squared singular values of $\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})$. We take the SVD of $\mathbf{A}$ and use the orthogonal invariance of singular values and (4.14) to obtain

$$\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))^2 = \lambda_j(\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{V}\mathbf{\Sigma}) \quad 1 \le j \le n-k.$$

Since $\mathcal{R}(\mathbf{W}) = \mathcal{R}(\mathbf{Q})$, it follows that $\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{Q}\mathbf{Q}^{\mathrm{T}}$, and

$$\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))^2 = \lambda_j(\mathbf{\Sigma}(\mathbf{I} - \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V})\mathbf{\Sigma}) \qquad 1 \le j \le n-k. \tag{4.29}$$

For convenience, we define

$$\mathbf{F} := \mathbf{\Sigma}_{\perp}^{2q+1}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\mathbf{\Sigma}_k^{-(2q+1)} \qquad \text{and} \qquad \mathbf{Z} := \begin{bmatrix}\mathbf{I}\\ \mathbf{F}\end{bmatrix}.$$

By Lemma 5, we have $\mathbf{Z}\mathbf{Z}^{\dagger} \preceq \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V}$. It follows from (4.15) and (4.16) that

$$\lambda_j(\mathbf{\Sigma}(\mathbf{I} - \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V})\mathbf{\Sigma}) \le \lambda_j(\mathbf{\Sigma}\left(\mathbf{I} - \mathbf{Z}\mathbf{Z}^{\dagger}\right)\mathbf{\Sigma}) \qquad 1 \le j \le n-k. \tag{4.30}$$

We now consider the term $\mathbf{\Sigma}\left(\mathbf{I} - \mathbf{Z}\mathbf{Z}^{\dagger}\right)\mathbf{\Sigma}$. We partition this product as the block matrix

$$\begin{aligned}\mathbf{\Sigma}(\mathbf{I} - \mathbf{Z}\mathbf{Z}^{\dagger})\mathbf{\Sigma} &= \begin{bmatrix}\mathbf{\Sigma}_k(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1})\mathbf{\Sigma}_k & \star\\ \star & \mathbf{\Sigma}_{\perp}(\mathbf{I} - \mathbf{F}(\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}})\mathbf{\Sigma}_{\perp}\end{bmatrix}\\ &:= \begin{bmatrix}\mathbf{B}_{11} & \star\\ \star & \mathbf{B}_{22}\end{bmatrix},\end{aligned}$$

where $\star$ denotes blocks that are unimportant to the computation. Applying Corollary 1, we obtain

$$\lambda_j\left(\mathbf{\Sigma}\left(\mathbf{I} - \mathbf{Z}\mathbf{Z}^{\dagger}\right)\mathbf{\Sigma}\right) \le \lambda_1(\mathbf{B}_{11}) + \lambda_j(\mathbf{B}_{22}) \quad 1 \le j \le n-k. \tag{4.31}$$

Combining (4.29), (4.30), and (4.31) gives

$$\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))^2 \le \lambda_1(\mathbf{B}_{11}) + \lambda_j(\mathbf{B}_{22}) \quad 1 \le j \le n-k. \tag{4.32}$$

We finish the proof by bounding $\lambda_1(\mathbf{B}_{11})$ and $\lambda_j(\mathbf{B}_{22})$. We begin by showing $\mathbf{B}_{11} \preceq \mathbf{\Sigma}_k\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{\Sigma}_k$. By (Halko et al. 2011, Proposition 8.2), we have $(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}) \preceq \mathbf{F}^{\mathrm{T}}\mathbf{F}$. Using the conjugation rule for the Loewner partial order (4.15), we conjugate both sides by $\mathbf{\Sigma}_k$ to

obtain

$$\mathbf{B}_{11} = \mathbf{\Sigma}_k(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1})\mathbf{\Sigma}_k \preceq \mathbf{\Sigma}_k\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{\Sigma}_k. \tag{4.33}$$

Statements (4.16) and (4.14) imply that $\lambda_1(\mathbf{B}_{11})$ is bounded as follows,

$$\lambda_1(\mathbf{B}_{11}) \leq \lambda_1(\mathbf{\Sigma}_k\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{\Sigma}_k) = \|\mathbf{F}\mathbf{\Sigma}_k\|_2^2 \tag{4.34}$$

By the definition of $\mathbf{F}$ and submultiplicativity,

$$\|\mathbf{F}\mathbf{\Sigma}_k\|_2 \leq \|\mathbf{\Sigma}_\perp^{2q}\|_2\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2\|\mathbf{\Sigma}_k^{-2q}\|_2 = \gamma^{2q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2,$$

Combining this with (4.34) yields

$$\lambda_1(\mathbf{B}_{11}) \leq \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2^2. \tag{4.35}$$

This completes the upper bound for the largest eigenvalues of $\mathbf{B}_{11}$. We now bound the eigenvalues of $\mathbf{B}_{22}$ by showing $\mathbf{B}_{22} \preceq \mathbf{\Sigma}_\perp^2$. Since $\mathbf{F}(\mathbf{I}+\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}$ is symmetric positive-definite, $\mathbf{I} - \mathbf{F}(\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}} \preceq \mathbf{I}$. Applying (4.15) with $\mathbf{S} = \mathbf{\Sigma}_\perp$ yields

$$\mathbf{B}_{22} = \mathbf{\Sigma}_\perp(\mathbf{I} - \mathbf{F}(\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}})\mathbf{\Sigma}_\perp \preceq \mathbf{\Sigma}_\perp^2.$$

Thus, by (4.16), we can bound eigenvalues as follows,

$$\lambda_j(\mathbf{B}_{22}) \leq \lambda_j(\mathbf{\Sigma}_\perp^2) = \sigma_{j+k}(\mathbf{A})^2 \qquad 1 \leq j \leq n-k. \tag{4.36}$$

Finally, we use the bounds of the squared singular values of $\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})$ (4.32), alongside the bounds for the eigenvalues of $\mathbf{B}_1$ (4.35), and $\mathbf{B}_2$ (4.36), to obtain

$$\sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))^2 \leq \sigma_{j+k}(\mathbf{A})^2 + \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2^2 \qquad 1 \leq j \leq n-k.$$

Taking square roots finishes the proof.

□

Proof of Theorem 5 (Probabilistic $\mathbf{R}_{22}$ Bounds for Algorithm 2)

Proof. Theorem 4 gives

$$\sigma_j(\mathbf{R}_{22}) \leq \beta(n,k)\sqrt{\sigma_{j+k}(A)^2 + \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2^2} \qquad 1 \leq j \leq n-k. \tag{4.37}$$

Applying Lemma 9 with $\mathbf{Z} = \boldsymbol{\Sigma}_\perp$ yields

$$\mathbb{P}\left(\|\boldsymbol{\Sigma}_\perp \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1^{-1}\|_2 \geq \frac{2\sqrt{k}}{\Delta} \left[\|\boldsymbol{\Sigma}_\perp\|_F + \left(\sqrt{k} + \sqrt{\ln\left(\frac{4}{\Delta^2}\right)} \right) \|\boldsymbol{\Sigma}_\perp\|_2 \right] \right) \leq \Delta.$$

It follows that $\|\boldsymbol{\Sigma}_\perp \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1^{-1}\|_2 \leq D$ with probability at least $1 - \Delta$. Combining that with (4.37) yields the result.

□

4.5.4 Proofs of Oversampling Theorems for $\mathbf{R}_{22}$

Next, we present proofs of theorems discussed in §4.3.2 from the oversampling analysis of the matrix $\mathbf{R}_{22}$.

Proof of Theorem 6 (Minimal Assumption Oversampling $\mathbf{R}_{22}$ Bounds)

Proof. We first take the bound in the Frobenius norm. By Theorem 3, we have

$$\sigma_j\left(\mathbf{R}_{22}\right) \leq \sigma_j(\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}}))\|(\mathbf{W}^{\mathrm{T}}\boldsymbol{\Pi}_1)^{-1}\|_2 \qquad 1 \leq j \leq n-k.$$

Since the equality holds for all singular values, it holds for the spectral and Frobenius norms, giving

$$\|\mathbf{R}_{22}\|_\xi \leq \|\mathbf{A}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\|_\xi \|(\mathbf{W}^{\mathrm{T}}\boldsymbol{\Pi}_1)^{-1}\|_2 \qquad 1 \leq j \leq n-k. \tag{4.38}$$

The inequality (4.38) is analogous to (Gu 2015, Theorem 3.5 and (3.7)). The difference between them is that in (4.38), the projectors $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ and $\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$ are being multiplied on the right instead of the left. The proof of (4.38) proceeds similarly to the corresponding theorems in Gu (2015). We begin by proving Theorem 6 in the Frobenius Norm, and the spectral norm case will follow as a consequence. Given (4.38), it remains to be shown that

$$\|\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_F^2 \leq \|\boldsymbol{\Sigma}_\perp\|_F^2 + \|\mathbf{A}_k - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\|_F^2. \tag{4.39}$$

Let $\mathbf{B} = \mathbf{A}\mathbf{Q}$ have SVD $\mathbf{B} = \hat{\mathbf{U}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{V}}^{\mathrm{T}}$. Then $\mathbf{W} = \mathbf{Q}\hat{\mathbf{V}}_k$ and we have

$$\mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{A}\mathbf{Q}\hat{\mathbf{V}}_k\hat{\mathbf{V}}_k^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}} = \mathbf{B}\hat{\mathbf{V}}_k\hat{\mathbf{V}}_k^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}} = \mathbf{B}_k\mathbf{Q}^{\mathrm{T}},$$

where the last equality follows from $\hat{\mathbf{V}}_k$ being the k dominant singular vectors of $\mathbf{B}$. As a consequence of Lemma 4, we have $\|\mathbf{A} - \mathbf{B}_k\mathbf{Q}^{\mathrm{T}}\|_F \leq \|\mathbf{A} - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\|_F$. Since $\mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{B}_k\mathbf{Q}^{\mathrm{T}}$,

this implies

$$\|\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_F \leq \|\mathbf{A} - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\|_F.$$

We square both sides and add and subtract $\mathbf{A}_k$ on the right-hand side to obtain

$$\|\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_F^2 \leq \|(\mathbf{A} - \mathbf{A}_k) + (\mathbf{A}_k - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_F^2.$$

Since $(\mathbf{A} - \mathbf{A}_k)^{\mathrm{T}}\mathbf{A}_k = \mathbf{0}$, by matrix Pythagoras (Drineas and Mahoney 2017, Lemma 5), we have

$$\|\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \|\mathbf{A}_k - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\|_F^2. \tag{4.40}$$

We finish the Frobenius norm case by writing $\|\mathbf{A} - \mathbf{A}_k\|_F = \|\mathbf{\Sigma}_\perp\|_F$.

We move on to the spectral norm case of the theorem. Since $\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}$ and $\mathbf{A} - \mathbf{A}_k$ are rank$-(n-k)$, we can write the Frobenius norms of those matrices as

$$\|\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_F^2 = \sum_{j=1}^{n-k} \sigma_j^2(\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}) \quad \text{and} \quad \|\mathbf{A} - \mathbf{A}_k\|_F^2 = \sum_{j=1}^{n-k} \sigma_j^2(\mathbf{A} - \mathbf{A}_k).$$

Combining these equations with (4.40) yields

$$\sum_{j=1}^{n-k} \sigma_j^2(\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}) \leq \|\mathbf{A}_k - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\|_F^2 + \sum_{j=1}^{n-k} \sigma_j^2(\mathbf{A} - \mathbf{A}_k).$$

Subtracting $\sum_{j=2}^{n-k} \sigma_j^2(\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}})$ from both sides and using the definition of the spectral norm yields

$$\begin{aligned} \|\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_2^2 \leq \|\mathbf{A}_k - \mathbf{A}_k\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\|_F^2 + \|\mathbf{\Sigma}_\perp\|_2^2 \\ + \sum_{j=2}^{n-k} \left(\sigma_j^2(\mathbf{A} - \mathbf{A}_k) - \sigma_j^2(\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}})\right). \end{aligned} \tag{4.41}$$

The additive singular value inequality (4.12) using the matrices $\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}$ and $\mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}$ gives

$$\begin{aligned} \sigma_{k+j}(\mathbf{A}) &= \sigma_{k+j}(\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}} + \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}) \\ &\leq \sigma_j(\mathbf{A} - \mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}) + \sigma_{k+1}(\mathbf{A}\mathbf{W}\mathbf{W}^{\mathrm{T}}) \quad 1 \leq j \leq n-k. \end{aligned}$$

Since $\operatorname{rank}(\mathbf{AWW}^{\mathrm{T}}) \leq k$, we have $\sigma_{k+1}(\mathbf{AWW}^{\mathrm{T}}) = 0$, so the previous equations becomes

$$\sigma_{k+j}(\mathbf{A}) \leq \sigma_j(\mathbf{A} - \mathbf{AWW}^{\mathrm{T}}) \qquad 1 \leq j \leq n-k.$$

Therefore, $\sigma_j^2(\mathbf{A} - \mathbf{A}_k) - \sigma_j^2(\mathbf{A} - \mathbf{AWW}^{\mathrm{T}}) \leq 0$ for all $2 \leq j \leq n-k$. Combining this with (4.41), we obtain

$$\|\mathbf{A} - \mathbf{AWW}^{\mathrm{T}}\|_2^2 \leq \|\mathbf{\Sigma}_\perp\|_2^2 + \|\mathbf{A}_k - \mathbf{A}_k\mathbf{QQ}^{\mathrm{T}}\|_F^2.$$

□

Proof of Theorem 7 (Structural $\mathbf{R}_{22}$ Oversampling Bounds for Algorithm 2)

Proof. By Proposition 1, $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular and Theorem 6 gives

$$\|\mathbf{R}_{22}\|_\xi \leq \beta(n,k)\sqrt{\|\mathbf{\Sigma}_\perp\|_\xi^2 + \|\mathbf{A}_k(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\|_F^2}. \tag{4.42}$$

It suffices to show

$$\|\mathbf{A}_k(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\|_F \leq \gamma^{2q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F.$$

We begin by writing the squared Frobenius norm of $\mathbf{A}_k(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})$ in terms of a trace,

$$\|\mathbf{A}_k(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\|_F^2 = \operatorname{trace}\left(\mathbf{A}_k(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\mathbf{A}_k^{\mathrm{T}}\right).$$

Substituting the SVD $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}$ and using the orthogonal invariance of the Frobenius norm yields

$$\|\mathbf{A}_k(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\|_F^2 = \operatorname{trace}\left(\mathbf{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\mathbf{V}_k^{\mathrm{T}}\mathbf{\Sigma}_k\right).$$

Since $\mathbf{V}$ is orthogonal, $\mathbf{VV}^{\mathrm{T}} = \mathbf{I}$ and

$$\|\mathbf{A}_k(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\|_F^2 = \operatorname{trace}\left(\mathbf{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}\mathbf{VV}^{\mathrm{T}}(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\mathbf{VV}^{\mathrm{T}}\mathbf{V}_k^{\mathrm{T}}\mathbf{\Sigma}_k\right).$$

We multiply out $\mathbf{V}^{\mathrm{T}}\mathbf{V}_k = \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}$ and $\mathbf{V}^{\mathrm{T}}(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\mathbf{V} = (\mathbf{I} - \mathbf{V}^{\mathrm{T}}\mathbf{QQ}^{\mathrm{T}}\mathbf{V})$ to obtain

$$\|\mathbf{A}_k(\mathbf{I} - \mathbf{QQ}^{\mathrm{T}})\|_F^2 = \operatorname{trace}\left(\mathbf{\Sigma}_k\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}(\mathbf{I} - \mathbf{V}^{\mathrm{T}}\mathbf{QQ}^{\mathrm{T}}\mathbf{V})\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}\mathbf{\Sigma}_k\right). \tag{4.43}$$

We define

$$\mathbf{F} := \mathbf{\Sigma}_\perp^{2q+1}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\mathbf{\Sigma}_k^{-(2q+1)} \qquad \text{and} \qquad \mathbf{Z} =: \begin{bmatrix}\mathbf{I}\\ \mathbf{F}\end{bmatrix}.$$

By Lemma 5, we have $\mathbf{Z}\mathbf{Z}^\dagger \preceq \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V}$. It follows from applying (4.15) with $\mathbf{S} = \mathbf{\Sigma}_k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$ that

$$\mathbf{\Sigma}_k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} (\mathbf{I} - \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V}) \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{\Sigma}_k \preceq \mathbf{\Sigma}_k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} (\mathbf{I} - \mathbf{Z}\mathbf{Z}^\dagger) \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{\Sigma}_k.$$

Thus, property (4.17) of the Loewner partial order implies

$$\operatorname{trace}\left(\mathbf{\Sigma}_k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} (\mathbf{I} - \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V}) \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{\Sigma}_k\right) \leq \operatorname{trace}\left(\mathbf{\Sigma}_k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} (\mathbf{I} - \mathbf{Z}\mathbf{Z}^\dagger) \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{\Sigma}_k\right).$$

Combining this with (4.43) yields

$$\|\mathbf{A}_k(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_F^2 \leq \operatorname{trace}\left(\mathbf{\Sigma}_k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} (\mathbf{I} - \mathbf{Z}\mathbf{Z}^\dagger) \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{\Sigma}_k\right). \tag{4.44}$$

Expressing the projector $\mathbf{I} - \mathbf{Z}\mathbf{Z}^\dagger$ in terms of $\mathbf{F}$ gives

$$\mathbf{I} - \mathbf{Z}\mathbf{Z}^\dagger = \begin{bmatrix}\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1} & \star\\ \star & \star\end{bmatrix},$$

where $\star$ denotes blocks that are unimportant to the computation. We substitute this into (4.44), to obtain

$$\|\mathbf{A}_k(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_F^2 \leq \operatorname{trace}\left(\mathbf{\Sigma}_k(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1})\mathbf{\Sigma}_k\right). \tag{4.45}$$

We now show $\mathbf{\Sigma}_k(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1})\mathbf{\Sigma}_k \preceq \mathbf{\Sigma}_k\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{\Sigma}_k$. By (Halko et al. 2011, Proposition 8.2), $(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}) \preceq \mathbf{F}^{\mathrm{T}}\mathbf{F}$. Using (4.15), we can conjugate by $\mathbf{\Sigma}_k$ and preserve the ordering, yielding

$$\mathbf{\Sigma}_k(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1})\mathbf{\Sigma}_k \preceq \mathbf{\Sigma}_k\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{\Sigma}_k.$$

As a consequence of (4.16),

$$\operatorname{trace}\left(\mathbf{\Sigma}_k(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1})\mathbf{\Sigma}_k\right) \leq \operatorname{trace}\left(\mathbf{\Sigma}_k\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{\Sigma}_k\right).$$

Combining this with (4.45) we obtain the string of inequalities

$$\|\mathbf{A}_k(\mathbf{I}-\mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_F^2 \leq \operatorname{trace}\left(\mathbf{\Sigma}_k(\mathbf{I}-(\mathbf{I}+\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1})\mathbf{\Sigma}_k\right) \leq \operatorname{trace}\left(\mathbf{\Sigma}_k\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{\Sigma}_k\right). \tag{4.46}$$

The term on the right can be bounded using strong submultiplicativity as

$$\operatorname{trace}\left(\mathbf{\Sigma}_k\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{\Sigma}_k\right) = \|\mathbf{F}\mathbf{\Sigma}_k\|_F^2 = \|\mathbf{\Sigma}_\perp^{2q+1}\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\mathbf{\Sigma}_k^{-2q}\|_F^2 \leq \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F^2.$$

This alongside (4.46) gives $\|\mathbf{A}_k(\mathbf{I}-\mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_F^2 \leq \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F^2$. Taking the square root of both sides completes the proof.

□

Proof of Theorem 8 (Probabilistic $\mathbf{R}_{22}$ Oversampling Bounds for Algorithm 2)

Proof. From Theorem 7, it is sufficient to show $\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F^2 \leq F^2$. The proof structure follows the proof of (Halko et al. 2011, Theorem 10.8). We begin with a strong submultiplicative inequality to obtain

$$\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F^2 \leq \|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\|_2^2\|\mathbf{\Omega}_1^\dagger\|_F^2.$$

We now bound each of the two terms individually. We give a large deviation bounds for $\|\mathbf{\Omega}_1^\dagger\|_F^2$. By (Halko et al. 2011, Prop 10.4), for any $x \geq 1$,

$$\mathbb{P}\left(\|\mathbf{\Omega}_1^\dagger\|_F \geq \sqrt{\frac{3k}{p+1}}x\right) \leq x^{-p}. \tag{4.47}$$

We choose $x = \left(\frac{2}{\Delta}\right)^{\frac{1}{p}}$ to obtain

$$\mathbb{P}\left(\|\mathbf{\Omega}_1^\dagger\|_F \geq \sqrt{\frac{3k}{p+1}}\left(\frac{2}{\Delta}\right)^{\frac{1}{p}}\right) \leq \frac{\Delta}{2}. \tag{4.48}$$

We now bound $\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\|_2$ by considering the function $h : \mathbb{R}^{(n-k)\times k} \to \mathbb{R}$ defined by

$$h(\mathbf{X}) = \|\mathbf{\Sigma}_\perp\mathbf{X}\|_2.$$

To show that h is Lipschitz, we use the reverse triangle inequality to obtain

$$|h(\mathbf{X}) - h(\mathbf{Y})| = |\|\mathbf{\Sigma}_\perp\mathbf{X}\|_2 - \|\mathbf{\Sigma}_\perp\mathbf{Y}\|_2| \leq \|\mathbf{\Sigma}_\perp\mathbf{X} - \mathbf{\Sigma}_\perp\mathbf{Y}\|_2 \leq \|\mathbf{\Sigma}_\perp\|_2\|\mathbf{X}-\mathbf{Y}\|_2.$$

Since the spectral norm of any matrix is bounded by its Frobenius norm, h is Lipschitz with

the constant $L = \|\mathbf{\Sigma}_\perp\|_2$. Applying Lemma 7, we have

$$\mathbb{E}\|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2\|_2 \leq \sqrt{k+p}\|\mathbf{\Sigma}_\perp\|_2 + \|\mathbf{\Sigma}_\perp\|_F. \tag{4.49}$$

Since h is Lipschitz, we can use (4.49) and Lemma 8 to obtain that

$$\mathbb{P}\left(\|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2\|_2 \geq \left(\sqrt{k+p}+t\right)\|\mathbf{\Sigma}_\perp\|_2 + \|\mathbf{\Sigma}_\perp\|_F\right) \leq e^{-\frac{t^2}{2}}.$$

In the above, choose $t = \sqrt{\ln\left(\frac{4}{\Delta^2}\right)}$ so that

$$\mathbb{P}\left(\|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2\|_2 \geq \left(\sqrt{k+p}+\sqrt{\ln\left(\frac{4}{\Delta^2}\right)}\right)\|\mathbf{\Sigma}_\perp\|_2 + \|\mathbf{\Sigma}_\perp\|_F\right) \leq \frac{\Delta}{2}. \tag{4.50}$$

For convenience, we define

$$F_1 = \frac{3k}{p+1}\left(\frac{2}{\Delta}\right)^{\frac{2}{p}}$$
$$F_2 = \left(\|\mathbf{\Sigma}_\perp\|_F + \left[\sqrt{\ln\left(\frac{4}{\Delta^2}\right)}+\sqrt{k+p}\right]\sigma_{k+1}(\mathbf{A})\right)^2$$

We now consider the joint probability $\|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2\|_2^2\|\mathbf{\Omega}_1^\dagger\|_F^2$. We define the three events on $\mathbf{\Omega}$

$$E_1 = \left\{\mathbf{\Omega} : \|\mathbf{\Omega}_1^\dagger\|_F^2 \geq F_1\right\}, \qquad E_2 = \left\{\mathbf{\Omega} : \|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2\|_2^2 \geq F_2\right\}, \quad \text{and}$$

$$E_\Omega = \left\{\mathbf{\Omega} : \|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2\|_2^2\|\mathbf{\Omega}_1^\dagger\|_F^2 \geq F_1F_2\right\}.$$

For E_Ω to occur, either E_1 or E_2 has to occur. Thus, $E_\Omega \subset E_1 \cup E_2$ and

$$\mathbb{P}(E_\Omega) \leq \mathbb{P}(E_1 \cup E_2) \leq \mathbb{P}(E_1) + \mathbb{P}(E_2).$$

From (4.48) and (4.50), we know $\mathbb{P}(E_1) \leq \frac{\Delta}{2}$ and $\mathbb{P}(E_2) \leq \frac{\Delta}{2}$, so

$$\mathbb{P}(E_\Omega) \leq \frac{\Delta}{2} + \frac{\Delta}{2} = \Delta$$

and with probability at least $1-\Delta$,

$$\|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2\|_2^2\|\mathbf{\Omega}_1^\dagger\|_F^2 \leq \frac{3k}{p+1}\left(\frac{2}{\Delta}\right)^{\frac{2}{p}}\left(\|\mathbf{\Sigma}_\perp\|_F + \left[\sqrt{\ln\left(\frac{4}{\Delta^2}\right)}+\sqrt{k+p}\right]\sigma_{k+1}(\mathbf{A})\right)^2 = F^2.$$

□

4.5.5 Proofs of Singular Value Bounds for $\mathbf{R}_{11}$

In this section, we continue our analysis by proving the singular value bounds pertaining to $\mathbf{R}_{11}$ provided in §4.3.3.

Proof of Theorem 9 ($\mathbf{R}_{11}$ Bounds for Algorithm Skeleton 1)

Proof. This proof proceeds by establishing a relationship between the principal angles between $\mathcal{R}(\mathbf{W})$, $\mathcal{R}(\mathbf{\Pi}_1)$, and $\mathcal{R}(\mathbf{V}_k)$. We then use that relationship to show $\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular and prove the bound.

We begin by applying Lemma 2 with $\mathcal{R}(\mathbf{W})$, $\mathcal{R}(\mathbf{\Pi}_1)$, and $\mathcal{R}(\mathbf{V}_k)$ yields

$$|\cos\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{\Pi}_1)) - \cos\varphi_k(\mathcal{R}(\mathbf{V}_k),\mathcal{R}(\mathbf{\Pi}_1))| \leq \sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k)). \tag{4.51}$$

Since $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular, (2.53) implies

$$\cos\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{\Pi}_1)) = \frac{1}{\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2}.$$

Combining this and (4.51) yields

$$\frac{1}{\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} - \sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k)) \leq \cos\varphi_k(\mathcal{R}(\mathbf{V}_k),\mathcal{R}(\mathbf{\Pi}_1)). \tag{4.52}$$

The assumption in (4.8) implies the above lower bound is strictly greater than 0. Thus, $\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular and by (2.53),

$$\cos\varphi_k(\mathcal{R}(\mathbf{V}_k),\mathcal{R}(\mathbf{\Pi}_1)) = \frac{1}{\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2}.$$

Substituting the above expression in (4.52) and multiplying by $\sigma_i(\mathbf{A})$ gives

$$\sigma_i(\mathbf{A})\left(\frac{1}{\|(\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} - \sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))\right) \leq \frac{\sigma_i(\mathbf{A})}{\|(\mathbf{V}_k^{\mathrm{T}}\mathbf{\Pi}_1)^{-1}\|_2} \qquad 1 \leq i \leq k.$$

Applying (4.1) completes the proof.

□

Proof of Theorem 10 (Structural $\mathbf{R}_{11}$ Bounds for Algorithm 2)

Proof. By Proposition 1, $\mathbf{W}^{\mathrm{T}}\mathbf{\Pi}_1$ is nonsingular and Theorem 9 yields

$$\sigma_i(\mathbf{A})\left(\frac{1}{\beta(n,k)} - \sin\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))\right) \leq \sigma_i(\mathbf{R}_{11}) \qquad 1 \leq i \leq k.$$

What remains to be shown is that

$$\sin\left(\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))\right) \leq \frac{\gamma^{2q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2}{\sqrt{\sigma_k(\mathbf{A})^2 + \gamma^{4q}\|\mathbf{\Sigma}_\perp\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\|_2^2}}.$$

This proof proceeds by proving

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))) = \lambda_1\left(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\right),$$

where $\mathbf{F} = \mathbf{\Sigma}_\perp^{2q+1}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\mathbf{\Sigma}_k^{-(2q+1)}$. This allows us to use theory developed in the proof of Theorem 4 to bound $\lambda_1\left(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\right)$ by the desired quantity. We begin with the squared expression for $\sin(\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k)))$ given in (2.54) and substitute $\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{Q}\mathbf{Q}^{\mathrm{T}}$ to obtain

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))) = \|\mathbf{V}_k\mathbf{V}_k^{\mathrm{T}}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_2^2.$$

Orthogonal invariance of the spectral norm and (4.14) yields

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))) = \|\mathbf{V}_k^{\mathrm{T}}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathrm{T}})\|_2^2 = \lambda_1\left(\mathbf{V}_k^{\mathrm{T}}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathrm{T}})\mathbf{V}_k\right).$$

Since $\mathbf{V}$ is orthonormal, $\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{I}$ and

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))) = \lambda_1\left(\mathbf{V}_k^{\mathrm{T}}\mathbf{V}\mathbf{V}^{\mathrm{T}}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathrm{T}})\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{V}_k\right).$$

We multiply out $\mathbf{V}^{\mathrm{T}}\mathbf{V}_k = \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}$ and $\mathbf{V}^{\mathrm{T}}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathrm{T}})\mathbf{V} = (\mathbf{I} - \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V})$ to obtain

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}),\mathcal{R}(\mathbf{V}_k))) = \lambda_1\left(\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}(\mathbf{I} - \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V})\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}\right).$$

We define

$$\mathbf{F} := \mathbf{\Sigma}_\perp^{2q+1}\mathbf{\Omega}_2\mathbf{\Omega}_1^{-1}\mathbf{\Sigma}_k^{-(2q+1)} \qquad \text{and} \qquad \mathbf{Z} = \begin{bmatrix}\mathbf{I}\\ \mathbf{F}\end{bmatrix}.$$

By Lemma 5, we have $\mathbf{Z}\mathbf{Z}^\dagger \preceq \mathbf{V}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{V}$. Thus,

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}), \mathcal{R}(\mathbf{V}_k))) \leq \lambda_1 \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} (\mathbf{I} - \mathbf{Z}\mathbf{Z}^\dagger) \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \right). \tag{4.53}$$

Multiplying out the upper-left block of $\mathbf{Z}\mathbf{Z}^\dagger$, we have

$$\mathbf{I} - \mathbf{Z}\mathbf{Z}^\dagger = \begin{bmatrix} \mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1} & \star \\ \star & \star \end{bmatrix}.$$

Substituting this expression into (4.53) and multiplying yields

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}), \mathcal{R}(\mathbf{V}_k))) \leq \lambda_1 \left(\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1} \right). \tag{4.54}$$

We substitute $\mathbf{I} - (\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1} = \mathbf{F}^{\mathrm{T}}\mathbf{F}(\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}$ to obtain

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}), \mathcal{R}(\mathbf{V}_k))) \leq \lambda_1 \left(\mathbf{F}^{\mathrm{T}}\mathbf{F}(\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1} \right).$$

A direct computation of $\lambda_1 \left(\mathbf{F}^{\mathrm{T}}\mathbf{F}(\mathbf{I} + \mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1} \right)$ shows

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}), \mathcal{R}(\mathbf{V}_k))) \leq \frac{\|\mathbf{F}\|_2^2}{1 + \|\mathbf{F}\|_2^2}.$$

By submultiplicativity, $\|\mathbf{F}\|_2 \leq \gamma^{2q} \|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2 \mathbf{\Omega}_1^{-1}\|_2 / \sigma_k(\mathbf{A})$. Furthermore, $f(x) = \dfrac{x^2}{1+x^2}$ is increasing for $x \geq 0$. Therefore,

$$\sin^2(\varphi_k(\mathcal{R}(\mathbf{W}), \mathcal{R}(\mathbf{V}_k))) \leq \frac{\gamma^{4q} \|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2 \mathbf{\Omega}_1^{-1}\|_2^2}{\sigma_k(\mathbf{A})^2 + \gamma^{4q} \|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2 \mathbf{\Omega}_1^{-1}\|_2^2}.$$

Taking the square root of both sides completes the proof.

□

Proof of Theorem 11 (Probabilistic $\mathbf{R}_{11}$ Bounds for Algorithm 2)

Proof. In the proof of Theorem 8, we showed that with probability at least $1 - \Delta$,

$$\|\mathbf{\Sigma}_\perp \mathbf{\Omega}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \leq \frac{3k}{p+1} \left(\frac{2}{\Delta} \right)^{\frac{2}{p}} \left(\|\mathbf{\Sigma}_\perp\|_F + \left[\sqrt{\ln\left(\frac{4}{\Delta^2} \right)} + \sqrt{k+p} \right] \sigma_{k+1}(\mathbf{A}) \right)^2 = F^2.$$

Combining that with Theorem 10 yields the result. The assumption on q is given so that the lower-bound is positive.

□

4.6 Conclusions and Discussion

The generalized GKS approach presented in Algorithm Skeleton 1 provides a framework for producing rank-revealing QR (RRQR) factorizations. In addition to computing RRQRs, the permutation matrix $\mathbf{\Pi}$ computed by this algorithm can be used in many applications, as reviewed in Chan and Hansen (1992). For example, in low-rank approximations, which are constructed by setting $\mathbf{R}_{22} = \mathbf{0}$ in (2.44), one can express the approximation as $\mathbf{A} \approx \mathbf{Q}_1 \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \end{bmatrix} \mathbf{\Pi}^{\mathrm{T}}$, or equivalently, by the interpolative decomposition $\mathbf{A} \approx \mathbf{A}_1 \begin{bmatrix} \mathbf{I}_k & \mathbf{R}_{11}^{-1}\mathbf{R}_{12} \end{bmatrix} \mathbf{\Pi}^{\mathrm{T}}$. Other important applications of RRQRs include rank-deficient least squares problems and subspace identification.

In this chapter, we analyzed Algorithm 2, originally proposed in Armstrong et al. (2023) as a special case of the generalized GKS approach, and derived both structural and probabilistic bounds for the algorithm. This theory builds on the analysis in Armstrong et al. (2023), where we proposed bounds for every singular value of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$, rather than just the extreme ones. These bounds depended heavily on the singular value gap and can be tuned to provide strong theoretical guarantees by increasing the number of subspace iterations performed.

We demonstrated the effectiveness of Algorithm 2 by applying it to three test matrices alongside simulation data of the fiber coating model. We numerically computed the singular values of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ and verified that in the presence of a singular value gap, the bounds presented in §4.3 provided reasonable bounds on these singular values.

Potential improvements on this work include developing theorems that bound all the singular values of $\mathbf{R}_{22}$ which incorporate oversampling, as this work is limited to only the $2-$ and Frobenius norm in that context. It would also be meaningful to develop bounds that do not rely as heavily on the existence of a singular value gap. Algorithm 2 seems to work well on matrices without such a gap, so it would be of interest to prove that such performance is assured with high probability.

Further work on this topic includes replacing $\mathbf{W}$ from the Randomized SVD with one obtained from a randomized UTV factorization (Martinsson et al. 2019). This approach can lead to computational speedups while providing similar theoretical guarantees. UTV factorizations also have implementations that take advantage of parallel computing (Heavner et al. 2021) and are effective on matrices that are too large to fit in RAM (Heavner et al. 2020). For performing a rank-revealing QR on $\mathbf{W}^{\mathrm{T}}$ stored out of core, we can use communication-

avoiding RRQRs (Demmel et al. 2015) to minimize data transfer, yielding a communication-avoiding implementation of the generalized GKS algorithm.

CHAPTER

5

CONCLUSION AND FUTURE WORK

In this dissertation, we developed physics-constrained neural ODEs to learn the dynamics of thin liquid films flowing down vertical cylindrical fibers. Our approach uses neural ODEs to learn the POD representation of the dynamics while enforcing desirable physical constraints to improve the long-term accuracy of the learned solution. While POD-based learning using NODEs has been shown to be an effective method to accurately model physical phenomena (Baker et al. 2023), these NODEs can often produce unphysical results, which can make them unreliable for applications involving large-scale systems and control tasks. We solved this problem by incorporating the mass conservation and bounded entropy conditions into the NODE model, which yielded POD representations of the droplet dynamics that maintain conserved quantities and ensure the positivity of the reconstructed solution.

Several interesting open questions and opportunities for further research emerge from the limitations and assumptions we make in the development of physics-constrained NODEs. We will briefly discuss the limitations of our approach, compare it against other existing algorithms such the POD-Galerkin method and encoder-decoder framework, and provide an overview of potential extensions of our algorithm.

In this work, we imposed periodic boundary conditions on the fiber coating model (1.1), which is a simplification of the real physical system. For liquids flowing down a vertical cylinder, the upstream fluid dynamics near the flow inlet strongly depend on the nozzle

geometry and the flow rate (Ji et al. 2020). Such complex flow characteristics have been utilized in the modeling of liquid flows along the entire cylindrical domain (Ruyer-Quil et al. 2008, 2009) and in the design of optimal boundary controls (Biswal et al. 2024). The assumption of a periodic domain is only valid for downstream dynamics in the Rayleigh-Plateau regime or isolated droplet regime far away from the nozzle inlet. Incorporating various types of inlet and outlet boundary conditions into the physics-constrained NODE models would make the established NODE models more applicable to physical control systems. For example, in the optimal boundary control method developed for a simplified fiber coating model in Biswal et al. (2024), the authors considered time-dependent inlet flow conditions and Neumann outlet boundary conditions. Dirichlet-Neumann inlet boundary conditions and soft outlet boundary conditions have also been considered in the works of Ruyer-Quil et al. (2008) and Ji et al. (2020). While the POD-based NODE algorithm does not necessarily depend on the boundary conditions of the underlying PDE model, the imposed constraints – such as the mass-conserving and positivity-preserving constraints – take different forms with non-periodic boundary conditions. Therefore, generalizing the physics-constrained NODEs to fiber-coating models equipped with other boundary conditions would require modifying the imposed constraints to account for the specific boundary conditions.

A current limitation of our approach is the reliance on a predefined set of parameters, such as the domain size (period) L, the total amount of liquid mass $\mathcal{M}$, and the scaling parameters α and η. Due to the imposed periodic boundary conditions, with other parameters fixed, the current setting of our model can only account for the Raleigh-Plateau and isolated droplet regimes, where the droplet dynamics are parameterized by the mass $\mathcal{M}$ and the period L. A numerical study in Ji et al. (2019) shows that the liquid film profiles, propagation speed, and stability of the traveling wave solutions to (1.1) can be identified by the paired parameters $\mathcal{M}$ and L. Similar results have also been identified for liquid films flowing along the inner surface of a cylinder (Camassa et al. 2021). However, in a setting with a specified flow rate imposed at the inlet boundary, the period L and the liquid mass $\mathcal{M}$ within the period strongly depend on the intrinsic frequency of the gravity-driven flow and interaction between the moving droplets. In this case, one could choose to use higher-order weighted residual models (Ruyer-Quil et al. 2008) to simulate the liquid film dynamics along the entire cylinder and to extract the average period downstream from the numerical profiles (Ji et al. 2021). When experimental data are available, an alternative approach is to extract droplet frequency and droplet size information from sequential experimental images and perform a time average to obtain droplet characteristics.

Expanding our current approach to accommodate a larger parameter space could enable its application to a wider range of scenarios with varying liquid properties, flow rates, and

cylinder radii. This would require modifying the scaling parameters, as well as the period and mass of the solution. However, such modifications could fundamentally alter the droplet dynamics, including the number of peaks and flow regimes, making the training process significantly more challenging. An interesting direction for future work would be to investigate the generalizability of our algorithm and its variants to droplet dynamics in other regimes.

The developed physics-constrained NODE can be easily adapted to other PDE models involving constrained quantities. Following the work of Matsubara and Yaguchi (2023) and the derivation in Section 3.3.2, any conservation law represented by a first integral can naturally be incorporated into the physics-constrained NODE by formulating a constrained optimization problem. Furthermore, it would also be of interest to extend the current learning algorithm to PDE models that involve non-mass-conserving effects, such as thin film equations with condensation or evaporation effects (Ji and Witelski 2018), and to compare the performance of the learned model against that of coarse-grained dynamical system models derived asymptotically from the PDE (Ji and Witelski 2024). To preserve the positivity of learned reduced-order solutions for other PDEs, one could develop NODEs that mimic positivity-preserving numerical schemes based on similar entropy estimates developed for fiber coating dynamics. Such schemes have been developed for thin film equations (Zhornitskaya and Bertozzi 1999), quantum hydrodynamics (Jüngel 2001; Braukhoff and Jüngel 2020), and other gradient flows (Düring et al. 2010). We refer to Kim (2023) for an extensive review of recent works on positivity-preserving schemes. A natural extension of our physics-constrained NODE approach to these PDEs involves encoding the relevant entropy conditions into the NODE models.

The physics-constrained NODE is related to a family of POD-NODE reduced-order modeling algorithms that learn POD coefficients using deep neural networks (Rojas et al. 2021; Dutta et al. 2021; Baker et al. 2023). These algorithms typically use an encoder (such as an RNN encoder) to project the POD temporal coefficients onto a latent sequence that evolves according to the NODE, followed by a decoder to provide the prediction for the POD coefficients in future times. Incorporating an encoder-decoder framework can significantly increase the generalizability of the learning algorithm. However, enforcing constraints such as mass conservation and entropy conditions in POD-NODE models coupled with encoder-decoder is nontrivial. Furthermore, the work in Baker et al. (2023) demonstrates that HBNODEs outperform vanilla NODEs in learning long-term trends from POD data due to their spectral properties. While the physics-constrained NODEs can be generalized to HBNODEs, this approach can destroy the spectral structure of HBNODEs and lead to degraded performance. Designing specialized encoder-decoder architectures and NODE variants that accommodate PDE structures and seamlessly integrate physics-based constraints

would be of interest for future work.

The POD-NODE approach is also comparable to POD-Galerkin projection (Deane et al. 1991), a commonly used technique for reduced order models. The POD-Galerkin reduced order method has been widely applied to computational fluid dynamics and Hamiltonian systems (Akhtar et al. 2009; Balajewicz et al. 2016; Gong et al. 2017). This approach leverages the orthogonality of the POD modes and projects the governing PDE onto the space spanned by the POD modes using Galerkin methods. The resulting low-dimensional dynamical system provides an efficient approximation of the original PDE. When combined with finite volume method (Stabile et al. 2017), finite element method (Ullmann et al. 2016), and other numerical methods, this framework has led to the development of many efficient and reliable PDE solvers. More recently, POD-Galerkin has also been integrated with physics-informed neural networks (PINNs) to solve inverse problems (Hijazi et al. 2023). Compared to Galerkin-based POD reduced-order models, NODE-based POD learning algorithms do not account for the underlying PDE when constructing backbone NODE models. Instead, they define the right-hand side function of the ODEs as a neural network that learns from training data. Future work could explore a thorough comparison between POD-NODE and POD-Galerkin approaches, particularly in scenarios where learning from both simulation data and experimental measurements is needed.

Our approach assumes access to high-resolution measurement data, which is not a realistic assumption in large-scale field experiments or real-world control applications. The practicality of physics-constrained NODE models can be enhanced by incorporating methods that account for noise and uncertainty. From a modeling perspective, certain types of noise can be incorporated into the PDE using various stochastic thin-film equations (Fischer and Grün 2018; Metzger and Grün 2022). However, the well-posedness of stochastic thin-film equations suitable for fiber coating systems remains an open question. From a data collection point of view, optimal sensor placement may also be necessary for fiber coating applications. An optimal sensor placement task aims to identify a limited number of sensor locations from a set of candidate points to minimize uncertainty in the reconstructed solution profile based on measurements, under certain criteria such as D-optimality and mutual information (Krause et al. 2008). When formulated as a column subset selection problem, the sensor placement task can be numerically solved using the Randomized GKS algorithms (Eswar et al. 2024), including the algorithm analyzed in Chapter 4.

We also analyzed a rank-revealing QR algorithm and performed a robust analysis that provides bounds for every singular value of the $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$ submatrices. A limitation of this analysis is that it relies on a large singular value gap. If the gap is not substantial, one could theoretically perform multiple subspace iterations to achieve a tighter bound, but

this is computationally inefficient and may be slower than directly computing the singular value decomposition. Therefore, it would be of interest to develop $\mathbf{R}_{11}$ bounds that do not impose additional requirements on the number of subspace iterations, ensuring the practical implementation of Algorithm 2.

We applied this algorithm to simulation data of the fiber-coating PDE and showed that it selects a good set of time snapshots from the data. In our numerical experiments, we performed the subset selection algorithm on unshifted fiber-coating data. For more complex problems, it may be of interest to perform Algorithm 2 on shifted data, particularly in the presence of multiple droplets or complex droplet interactions. Furthermore, it would be valuable to further explore the application of subset selection via rank-revealing QR algorithms to POD-based learning algorithms for the dynamics of free surface flows. Specifically, one may use rank-revealing QR for feature selection as part of data preprocessing before training a physics-constrained NODE and evaluate how feature selection affects the model performance.

REFERENCES

Akhtar, I., Nayfeh, A. H., and Ribbens, C. J. (2009). On the stability and extension of reduced-order galerkin models in incompressible flows: a numerical study of vortex shedding. *Theoretical and Computational Fluid Dynamics*, 23(3):213–237.

Anderson, W. and Farazmand, M. (2022). Evolution of nonlinear reduced-order solutions for pdes with conserved quantities. *SIAM Journal on Scientific Computing*, 44(1):A176–A197.

Armstrong, R., Buzali, A., and Damle, A. (2023). Structure-aware analyses and algorithms for interpolative decompositions. *CoRR*, abs/2310.09452.

Baker, J., Cherkaev, E., Narayan, A., and Wang, B. (2023). Learning proper orthogonal decomposition of complex dynamics using heavy-ball neural odes. *Journal of Scientific Computing*, 95(2):54.

Balajewicz, M., Tezaur, I., and Dowell, E. (2016). Minimal subspace rotation on the stiefel manifold for stabilization and enhancement of projection-based reduced order models for the compressible navier–stokes equations. *Journal of Computational Physics*, 321:224–241.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Benner, P., Gugercin, S., and Willcox, K. (2015). A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531.

Bernis, F. and Friedman, A. (1990). Higher order nonlinear degenerate parabolic equations. *Journal of differential equations*, 83(1):179–206.

Bertozzi, A. L. and Pugh, M. (1994). The lubrication approximation for thin viscous films: the moving contact line with a'porous media'cut-off of van der waals interactions. *Nonlinearity*, 7(6):1535.

Bhatia, R. (1997). *Matrix Analysis*. Springer-Verlag, New York.

Biswal, S., Ji, H., Elamvazhuthi, K., and Bertozzi, A. L. (2024). Optimal boundary control of a model thin-film fiber coating model. *Physica D: Nonlinear Phenomena*, 457:133942.

Braukhoff, M. and Jüngel, A. (2020). Entropy-dissipating finite-difference schemes for nonlinear fourth-order parabolic equations. *arXiv preprint arXiv:2001.03742*.

Broadbent, M. E., Brown, M., and Penner, K. (2010). Subset selection algorithms: Randomized vs. deterministic. *SIAM undergraduate research online*, 3(1):50–71.

Businger, P. and Golub, G. (1966). Linear least squares solutions by householder transformations. *Numerische Mathematik*, 7:269–276.

Camassa, R., Marzuola, J. L., Ogrosky, H. R., and Swygert, S. (2021). On the stability of traveling wave solutions to thin-film and long-wave models for film flows inside a tube. *Physica D: Nonlinear Phenomena*, 415:132750.

Cazaubiel, A. and Carlson, A. (2023). Influence of wind on a viscous liquid film flowing down a thread. *Physical Review Fluids*, 8(5):054002.

Chan, T. F. and Hansen, P. C. (1992). Some applications of the rank revealing QR factorization. *SIAM Journal on Scientific and Statistical Computing*, 13(3):727–741.

Chang, H.-C. and Demekhin, E. A. (1999). Mechanism for drop formation on a coated vertical fibre. *Journal of Fluid Mechanics*, 380(1):233–255.

Chattopadhyay, S. (2024). Thermocapillary thin films on rotating cylinders with wall slip and exothermic reactions. *International Journal of Heat and Mass Transfer*, 233:126027.

Chattopadhyay, S. and Ji, H. (2024). Modeling reactive film flows down a heated fiber. *Chemical Engineering Science*, 300:120551.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6572–6583.

Cheng, H., Gimbutas, Z., Martinsson, P. G., and Rokhlin, V. (2005). On the compression of low rank matrices. *SIAM Journal on Scientific Computing*, 26(4):1389–1404.

Chinju, H., Uchiyama, K., and Mori, Y. (2000). "string-of-beads" flow of liquids on vertical wires for gas absorption. *AIChE Journal*, 46:937 – 945.

Chu, H., Miyatake, Y., Cui, W., Wei, S., and Furihata, D. (2024). Structure-preserving physics-informed neural networks with energy or lyapunov structure. *arXiv preprint arXiv:2401.04986*.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. (2020). Lagrangian neural networks.

Craster, R. V. and Matar, O. K. (2006). On viscous beads flowing down a vertical fibre. *Journal of Fluid Mechanics*, 553:85–105.

Deane, A. E., Kevrekidis, I. G., Karniadakis, G. E., and Orszag, S. A. (1991). Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders. *Physics of Fluids A: Fluid Dynamics*, 3(10):2337–2354.

Demmel, J. W., Grigori, L., Gu, M., and Xiang, H. (2015). Communication avoiding rank revealing QR factorization with column pivoting. *SIAM Journal on Matrix Analysis and Applications*, 36(1):55–89.

Dormand, J. R. and Prince, P. J. (1980). A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26.

Drineas, P. and Mahoney, M. W. (2017). Lectures on randomized numerical linear algebra.

Duersch, J. A. and Gu, M. (2017). Randomized QR with column pivoting. *SIAM Journal on Scientific Computing*, 39(4):C263–C291.

Dupont, E., Doucet, A., and Teh, Y. W. (2019). Augmented neural odes. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Duprat, C., Ruyer-Quil, C., and Giorgiutti-Dauphiné, F. (2009). Spatial evolution of a film flowing down a fiber. *Physics of Fluids*, 21(4):042–109.

Düring, B., Matthes, D., and Milišic, J. P. (2010). A gradient flow scheme for nonlinear fourth order equations. *Discrete Contin. Dyn. Syst. Ser. B*, 14(3):935–959.

Dutta, S., Rivera-Casillas, P., Cecil, O., Farthing, M., Perracchione, E., Putti, M., et al. (2021). Data-driven reduced order modeling of environmental hydrodynamics using deep autoencoders and neural odes. In *9th International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2021*, pages 1–16. International Center for Numerical Methods in Engineering.

Eswar, S., Rao, V., and Saibaba, A. K. (2024). Bayesian d-optimal experimental designs via column subset selection.

Fedele, F., Abessi, O., and Roberts, P. J. (2015). Symmetry reduction of turbulent pipe flows. *Journal of Fluid Mechanics*, 779:390–410.

Feng, Y., Xiao, J., and Gu, M. (2019). Flip-flop spectrum-revealing QR factorization and its applications to singular value decomposition. *Electronic Transactions on Numerical Analysis*, 51:469–494.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. (2020a). Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Finzi, M., Wang, K. A., and Wilson, A. G. (2020b). Simplifying hamiltonian and lagrangian neural networks via explicit constraints. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13880–13889. Curran Associates, Inc.

Fischer, J. and Grün, G. (2018). Existence of positive solutions to stochastic thin-film equations. *SIAM Journal on Mathematical Analysis*, 50(1):411–455.

Frenkel, A. L. (1992). Nonlinear theory of strongly undulating thin films flowing down vertical cylinders. *Europhysics Letters*, 18(7):583.

Fu, G., Ji, H., Pazner, W., and Li, W. (2024). Mean field control of droplet dynamics with high order finite element computations. *arXiv preprint arXiv:2402.05923*.

Gabbard, C. T. and Bostwick, J. B. (2021). Asymmetric instability in thin-film flow down a fiber. *Physical Review Fluids*, 6(3):034005.

Golub, G., Klema, V., and Stewart, G. (1976). Rank degeneracy and least squares problems. Technical Report STAN-CS-76-559, Computer Science Department, Stanford University.

Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 4th edition.

Gong, Y., Wang, Q., and Wang, Z. (2017). Structure-preserving galerkin pod reduced-order modeling of hamiltonian systems. *Computer Methods in Applied Mechanics and Engineering*, 315:780–798.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Gu, M. (2015). Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3):A1139–A1173.

Gu, M. and Eisenstat, S. C. (1996). Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3(null):1157–1182.

Halko, N., Martinsson, P. G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

Heavner, N., Igual, F. D., Quintana-Ortí, G., and Martinsson, P. (2021). Efficient algorithms for computing a rank-revealing UTV factorization on parallel computing architectures. *CoRR*, abs/2104.05782.

Heavner, N., Martinsson, P.-G., and Quintana-Ortí, G. (2020). Computing rank-revealing factorizations of matrices stored out-of-core.

Hijazi, S., Freitag, M., and Landwehr, N. (2023). Pod-galerkin reduced order models and physics-informed neural networks for solving inverse problems for the navier–stokes equations. *Advanced Modeling and Simulation in Engineering Sciences*, 10(1):5.

Hilliard, Z. T. and Farazmand, M. (2024). Enforcing conserved quantities in galerkin truncation and finite volume discretization. *Nonlinear Dynamics*, 112:14051–14069.

Hong, Y. and Pan, C.-T. (1992). Rank-revealing QR factorizations and the singular value decomposition. *Mathematics of Computation*, 58(107):213–232.

Horn, R. and Johnson, C. (2013). *Matrix Analysis*. Cambridge University Press, New York, 2nd edition.

Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis*. Cambridge University Press, Cambridge.

Ji, H., Falcon, C., Sadeghpour, A., Zeng, Z., Ju, Y. S., and Bertozzi, A. L. (2019). Dynamics of thin liquid films on vertical cylindrical fibres. *Journal of Fluid Mechanics*, 865:303–327.

Ji, H., Falcon, C., Sedighi, E., Sadeghpour, A., Ju, Y. S., and Bertozzi, A. L. (2021). Thermally-driven coalescence in thin liquid film flowing down a fibre. *Journal of Fluid Mechanics*, 916:A19.

Ji, H., Sadeghpour, A., Ju, Y. S., and Bertozzi, A. L. (2020). Modelling film flows down a fibre influenced by nozzle geometry. *Journal of Fluid Mechanics*, 901:R6.

Ji, H., Taranets, R., and Chugunova, M. (2022). On travelling wave solutions of a model of a liquid film flowing down a fibre. *European Journal of Applied Mathematics*, 33(5):864–893.

Ji, H. and Witelski, T. P. (2018). Instability and dynamics of volatile thin films. *Physical Review Fluids*, 3(2):024001.

Ji, H. and Witelski, T. P. (2024). Coarsening of thin films with weak condensation. *SIAM Journal on Applied Mathematics*, 84(2):362–386.

Jüngel, A. (2001). A positivity-preserving numerical scheme for a nonlinear fourth order parabolic system. *SIAM journal on numerical analysis*, 39(2):385–406.

Kahan, W. (1966). Numerical linear algebra. *Canadian Mathematical Bulletin*, 9(5):757–801.

Kalliadasis, S. and Chang, H.-C. (1994). Drop formation during coating of vertical fibres. *Journal of Fluid Mechanics*, 261:135–168.

Kani, J. N. and Elsheikh, A. H. (2017). Dr-rnn: A deep residual recurrent neural network for model reduction.

Kani, J. N. and Elsheikh, A. H. (2019). Reduced-order modeling of subsurface multi-phase flow models using deep residual recurrent neural networks. *Transport in Porous Media*, 126.

Kelly, J., Bettencourt, J., Johnson, M. J., and Duvenaud, D. K. (2020). Learning differential equations that are easy to solve. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4370–4380. Curran Associates, Inc.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. (2020). Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707.

Kim, B. (2023). *Theory of Positivity-Preserving Numerical Methods for Thin Viscous Liquids Flowing Down Vertical Fibers*. University of California, Los Angeles.

Kim, B., Ji, H., Bertozzi, A. L., Sadeghpour, A., and Sungtaek Ju, Y. (2024). A positivity-preserving numerical method for a thin liquid film on a vertical cylindrical fiber. *Journal of Computational Physics*, 496:112560.

Kliakhandler, I. L., Davis, S. H., and Bankoff, S. G. (2001). Viscous beads on vertical fibre. *Journal of Fluid Mechanics*, 429:381–390.

Knyazev, A. V. and Argentati, M. E. (2007). Majorization for changes in angles between subspaces, ritz values, and graph laplacian spectra. *SIAM Journal on Matrix Analysis and Applications*, 29(1):15–32.

Kolter, J. Z. and Manek, G. (2019). Learning stable deep dynamics models. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2).

Lee, K. and Carlberg, K. T. (2021). Deep conservation: A latent-dynamics model for exact satisfaction of physical conservation laws. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):277–285.

Li, P. and Chao, Y. (2020). Marangoni instability of self-rewetting films modulated by chemical reactions flowing down a vertical fibre. *Chemical Engineering Science*, 227:115936.

Li, X., Wong, T.-K. L., Chen, R. T. Q., and Duvenaud, D. K. (2019). Scalable gradients and variational inference for stochastic differential equations. In *Second Symposium on Advances in Approximate Bayesian Inference*.

Liu, R. and Ding, Z. (2020). Instabilities and bifurcations of liquid films flowing down a rotating fibre. *Journal of Fluid Mechanics*, 899:A14.

Lutter, M., Ritter, C., and Peters, J. (2019). Deep lagrangian networks: Using physics as model prior for deep learning. *CoRR*, abs/1907.04490.

Mahoney, M. W. and Drineas, P. (2009). Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702.

Martinsson, P. (2016). The interpolative decomposition (id).

Martinsson, P. G. (2015). Blocked rank-revealing qr factorizations: How randomized sampling can be used to avoid single-vector pivoting.

Martinsson, P.-G., Quintana-Ortí, G., and Heavner, N. (2019). RandUTV: A blocked randomized algorithm for computing a rank-revealing UTV factorization. *ACM Trans. Math. Softw.*, 45(1).

Matsubara, T., Ishikawa, A., and Yaguchi, T. (2020). Deep energy-based modeling of discrete-time physics. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Matsubara, T. and Yaguchi, T. (2023). Finde: Neural differential equations for finding and preserving invariant quantities.

Metzger, S. and Grün, G. (2022). Existence of nonnegative solutions to stochastic thin-film equations in two space dimensions. *Interfaces and Free Boundaries*, 24(3):307–387.

Nair, A. and Sharma, G. (2020). Stability of surfactant-laden liquid film flow over a cylindrical rod. *Physical Review E*, 102(2):023111.

Norcliffe, A., Bodnar, C., Day, B., Simidjievski, N., and Liò, P. (2020). On second order behaviour in augmented Neural ODEs. In *Advances in Neural Information Processing Systems*.

Quaglino, A., Gallieri, M., Masci, J., and Koutník, J. (2020). Snode: Spectral discretization of Neural ODEs for system identification. In *International Conference on Learning Representations*.

Quéré, D. (1990). Thin films flowing on vertical fibers. *Europhysics Letters*, 13(8):721.

Reiss, J., Schulze, P., Sesterhenn, J., and Mehrmann, V. (2018). The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40(3):A1322–A1344.

Rojas, C. J., Dengel, A., and Ribeiro, M. D. (2021). Reduced-order model for fluid flows via neural ordinary differential equations. *arXiv preprint arXiv:2102.02248*.

Ruan, Y., Nadim, A., Duvvoori, L., and Chugunova, M. (2021). Liquid films falling down a vertical fiber: modeling, simulations and experiments. *Fluids*, 6(8):281.

Ruyer-Quil, C., Treveleyan, P., Giorgiutti-Dauphine, F., Duprat, C., and Kalliadasis, S. (2008). Modelling film flows down a fibre. *Journal of Fluid Mechanics*, 603:431–462.

Ruyer-Quil, C., Trevelyan, S. P. M. J., Giorgiutti-Dauphiné, F., Duprat, C., and Kalliadasis, S. (2009). Film flows down a fiber: Modeling and influence of streamwise viscous diffusion. *The European Physical Journal Special Topics*, 166(1):89–92.

Sadeghpour, A., Oroumiyeh, F., Zhu, Y., Ko, D. D., Ji, H., Bertozzi, A. L., and Ju, Y. S. (2021). Experimental study of a string-based counterflow wet electrostatic precipitator for collection of fine and ultrafine particles. *Journal of the Air & Waste Management Association*, 71(7):851–865.

Sadeghpour, A., Zeng, Z., Ji, H., Ebrahimi, N. D., Bertozzi, A. L., and Ju, Y. S. (2019). Water vapor capturing using an array of traveling liquid beads for desalination and water treatment. *Science Advances*, 5(4):eaav7662.

Sadeghpour, A., Zeng, Z., and Ju, Y. S. (2017). Effects of nozzle geometry on the fluid dynamics of thin liquid films flowing down vertical strings in the rayleigh–plateau regime. *Langmuir*, 33:6292–6299.

Sankar, A., Spielman, D. A., and Teng, S. (2003). Smoothed analysis of the condition numbers and growth factors of matrices. *CoRR*, cs.NA/0310022.

Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28.

Solorio, F. J. and Sen, M. (1987). Linear stability of a cylindrical falling film. *Journal of fluid mechanics*, 183:365–377.

Stabile, G., Hijazi, S., Mola, A., Lorenzi, S., Rozza, G., et al. (2017). Pod-galerkin reduced order methods for cfd using finite volume discretisation: vortex shedding around a circular cylinder. *Communications in Applied and Industrial Mathematics*, 8(1):210–236.

Szyld, D. (2006). The many proofs of an identity on the norm of oblique projections. *Numerical Algorithms*, 42:309–323.

Takeishi, N. and Kawahara, Y. (2020). Learning dynamics models with stable invariant sets.

Taranets, R. M., Ji, H., and Chugunova, M. (2024). On weak solutions of a control-volume model for liquid films flowing down a fibre. *Discrete and Continuous Dynamical Systems-B*, pages 0–0.

Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra*. SIAM, Philadelphia.

Ullmann, S., Rotkvic, M., and Lang, J. (2016). POD-Galerkin reduced-order modeling with adaptive finite element snapshots. *Journal of Computational Physics*, 325:244–258.

Xia, H., Suliafu, V., Ji, H., Nguyen, T., Bertozzi, A., Osher, S., and Wang, B. (2021). Heavy ball neural ordinary differential equations. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18646–18659. Curran Associates, Inc.

Xie, Q., Liu, R., Wang, X., and Chen, X. (2021). Investigation of flow dynamics of thin viscous films down differently shaped fibers. *Applied Physics Letters*, 119(20).

Zeng, Z., Sadeghpour, A., Warrier, G., and Ju, Y. S. (2017). Experimental study of heat transfer between thin liquid films flowing down a vertical string in the rayleigh-plateau instability regime and a counterflowing gas stream. *International Journal of Heat and Mass Transfer*, 108:830–840.

Zhornitskaya, L. and Bertozzi, A. L. (1999). Positivity-preserving numerical schemes for lubrication-type equations. *SIAM Journal on Numerical Analysis*, 37(2):523–555.

Zhu, P. and Knyazev, A. (2012). Angles between subspaces and their tangents. *Journal of Numerical Mathematics*, 21.

APPENDIX

APPENDIX

A

APPENDIX

A.1 Detailed Derivations of Equations (2.6), (2.7), (2.27), and (2.23).

Derivation of (2.6) and (2.7)

We derive (2.6) and (2.7) by considering the stress balance at the free surface $y^* = R^* + h^*$. The stress tensor $\mathbf{T}$ for an incompressible fluid is defined as

$$\mathbf{T} = -p^*\mathbf{I} + \mu(\nabla\vec{u}^* + (\nabla\vec{u}^*)^{\mathrm{T}}) = \begin{bmatrix} -p^* + 2\mu u^*_{x^*} & \mu(u^*_{y^*} + v^*_{x^*}) \\ \mu(u^*_{y^*} + v^*_{x^*}) & -p^* + 2\mu v^*_{y^*} \end{bmatrix}$$

At the free surface $y^* = R^* + h^*$, we have a normal vector $\mathbf{n} = \dfrac{1}{\sqrt{1 + (h^*_{x^*})^2}} \begin{bmatrix} -h^*_{x^*} \\ 1 \end{bmatrix}$, and a tangent vector $\mathbf{t} = \dfrac{1}{\sqrt{1 + (h^*_{x^*})^2}} \begin{bmatrix} 1 \\ h^*_{x^*} \end{bmatrix}$. Then the surface tension is defined as $\mathbf{t}_s = -\kappa\sigma\mathbf{n}$, where we assume σ is a constant and the curvature κ is given by

$$\kappa = \frac{h^*_{x^*x^*}}{(1 + (h^*_{x^*})^2)^{3/2}} - \frac{1}{(h^* + R^*)(1 + (h^*_{x^*})^2)^{3/2}},$$

where the first and second terms represent the destabilizing azimuthal curvature and the stabilizing streamwise curvature, respectively.

To derive (2.6), we balance the normal stress on the interface with the surface tension to obtain $\mathbf{Tn} = \mathbf{t}_s$. By left-multiplying both sides of this equation by $\mathbf{n}^{\mathrm{T}}$ and rearranging, we obtain

$$-\mathbf{n}^{\mathrm{T}}\mathbf{T}\mathbf{n} + \mathbf{n}^{\mathrm{T}}\mathbf{t}_s = \mathbf{0}.$$

Expanding this expression, we obtain

$$-p^* + \frac{2\mu}{1+(h^*_{x^*})^2}\left((h^*_{x^*})^2 u^*_{x^*} - h^*_{x^*}(v^*_{x^*} + u^*_{y^*}) + v^*_{y^*}\right) + \frac{\sigma}{(1+(h^*_{x^*})^2)^{3/2}}\left(\frac{1+(h^*_{x^*})^2}{h^* + R^*} - h^*_{x^*x^*}\right) = 0,$$

which we can add p^* to both sides to and obtain (2.6).

To derive (2.7), we balance the shear stress at the interface, which is unaffected by the surface tension, yielding $\mathbf{Tt} = \mathbf{0}$. Left-multiplying both sides of this equation by $\mathbf{n}^{\mathrm{T}}$ gives $\mathbf{n}^{\mathrm{T}}\mathbf{T}\mathbf{t} = \mathbf{0}$. Expanding this expression gives (2.7),

$$(1-(h^*_{x^*})^2)(v^*_{x^*} + u^*_{y^*}) + 2h^*_{x^*}(v^*_{y^*} - u^*_{x^*}) = 0.$$

Derivation of (2.27)

We will show that

$$(1+\alpha h)\frac{\partial h}{\partial \tilde{t}} + \frac{\partial q}{\partial x} = 0.$$

We use the definition for q in (2.27) and $\alpha = 1/R$ to obtain

$$(1+\alpha h)\frac{\partial h}{\partial \tilde{t}} + \frac{\partial q}{\partial x} = \left(1 + \frac{h}{R}\right)\frac{\partial h}{\partial \tilde{t}} + \frac{\partial}{\partial x}\left(\frac{1}{R}\int_R^{h+R} uy\ dy\right).$$

Factoring out an R^{-1} and applying Leibniz rule to evaluate $\dfrac{\partial}{\partial x}\displaystyle\int_R^{h+R} uy\ dy$ gives

$$(1+\alpha h)\frac{\partial h}{\partial \tilde{t}} + \frac{\partial q}{\partial x} = \frac{1}{R}\left((h+R)\frac{\partial h}{\partial \tilde{t}} + (h+R)\frac{\partial h}{\partial x}\left(u\big|_{y=h+R}\right) + \int_R^{h+R} u_x y\ dy\right).$$

Applying the kinematic boundary condition (2.15) and continuity equation (2.11) yields

$$(1+\alpha h)\frac{\partial h}{\partial \tilde{t}} + \frac{\partial q}{\partial x} = \frac{1}{R}\left((h+R)\left(v\big|_{y=h+R}\right) - \int_R^{h+R}(yv_y + v)\,dy\right).$$

Integrating by parts gives the desired equation,

$$(1+\alpha h)\frac{\partial h}{\partial \tilde{t}} + \frac{\partial q}{\partial x} = \frac{1}{R}\left((h+R)\left(v\big|_{y=h+R}\right) - (h+R)\left(v\big|_{y=h+R}\right)\right) = 0.$$

Derivation of (2.23)

We will show that $(1-p_x)q_0 = q$. By the definition of q and q_0 in (2.23) and (2.24), we have

$$Rp_xq_0 + Rq = p_x\int_R^{h+R} u_0 y\ dy + \int_R^{h+R} uy\ dy.$$

Since p is not a function of y (see (2.17)), we can put p_x inside the integral and apply (2.22) to obtain

$$Rp_xq_0 + Rq = \int_R^{h+R}\left(1 + u_{yy} + \frac{u_y}{y}\right) u_0 y\ dy + \int_R^{h+R} uy\ dy$$

Distributing $u_0 y$ and integration by parts gives

$$Rp_xq_0+Rq = \int_R^{h+R} u_0 y\ dy+(u_y u_0 y)\Big|_R^{h+R} - \int_R^{h+R} u_y(u_0+yu_{0_y})\ dy+\int_R^{h+R} u_y u_0\ dy+\int_R^{h+R} uy\ dy.$$

Using the no-slip boundary conditions (2.14) and the shear stress condition (2.18), we simplify this to

$$Rp_xq_0 + Rq = \int_R^{h+R} u_0 y\ dy - \int_R^{h+R} u_y u_{0_y} y\ dy + \int_R^{h+R} uy\ dy.$$

We then integrate by parts and apply the boundary conditions on u_0 (2.21) to get

$$Rp_xq_0 + Rq = \int_R^{h+R} u_0 y\ dy - \left(yu_{0_y}u\right)\Big|_R^{h+R} + \int_R^{h+R} u(u_{0_y} + u_{0_{yy}}y)\ dy + \int_R^{h+R} uy\ dy.$$

From the boundary conditions on u_0 (2.21), $\left(yu_{0_y}u\right)\Big|_R^{h+R} = 0$, and

$$Rp_xq_0 + Rq = \int_R^{h+R} u_0 y\ dy + \int_R^{h+R}\left(1 + u_{0_{yy}} + \frac{u_{0_y}}{y}\right) uy\ dy,$$

which, since u_0 satisfies (2.20), gives

$$Rp_xq_0 + Rq = \int_R^{h+R} u_0 y\ dy = Rq_0.$$

Dividing both sides by R and re-arranging the terms yields the desired result (2.23).