Multiscale Techniques for Nonlinear Dynamical Systems: Applications and Theory

Jacob R. Price

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Hong Qian, Chair

Panos Stinis

Randy LeVeque

Program Authorized to Offer Degree:
Department of Applied Mathematics

University of Washington

Abstract

Multiscale Techniques for Nonlinear Dynamical Systems:
Applications and Theory

Jacob R. Price

Chair of the Supervisory Committee:
Professor Hong Qian
Department of Applied Mathematics

Most interesting real world systems can be understood at multiple scales of detail. A physical system such as a closed container of gas particles can be understood in terms of hydrodynamic flows, molecules and atoms exerting forces upon one another, or evolving wavefunctions for each component particle. A multiscale approach can be used to understand the interplay of different scales of detail in problems that lack time or spatial scale separation. We present the Mori-Zwanzig formalism as a general framework for understanding multiscale methods. Another popular multiscale method, the heterogeneous multiscale method, is shown to be a special case of this framework. The heterogeneous multiscale method framework is applied to a plasma physics problem. We then derive a new multiscale scheme from the Mori-Zwanzig formalism called the complete memory approximation, and apply it to the Korteweg-de Vries equation, the 3D Euler's equations, and Burgers' equation. Surprising scaling results shed light into the complex role played by memory in reduced order models of partial differential equations.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

Six years of intensive study and work has finally culminated in the production of this dissertation. The trope of the solitary academic would suggest that credit is due to hours of dedicated work and effort by a lone researcher. That idea is hopelessly out of date and inaccurate. I, like all other people, am embedded in a culture and social network that influences everything I do. I count myself lucky that the constellation of people in my life have offered me an environment in which I feel supported to pursue my passions. In fact, the completion of this document should be attributed not to myself, but to a lifetime of insightful teachers, supportive friends, and an endlessly loving family. I will try here to offer my humblest thanks to those people, with the caveat that this list will be necessarily incomplete. A comprehensive list would stretch far beyond the scope of this short section. I hope those who are left off will forgive me.

I owe much of my academic success to an unprecedented string of formative, wonderful teachers and mentors. My love of learning can be traced in part to Kay Nicholas and Jack Yates. My passion for mathematics developed in imitation of the tremendous passion expressed by Ted Buckless and Steven Thumser. At Kalamazoo College, a number of influential figures guided me to find my passion. I so adored my time there that I made it my goal to teach at just such an institution. Peggy Cauchy was the face of the mathematics department, and made it a wonderful place to be. John Fink was an exemplary advisor and an inspirational human. My passion for teaching was discovered when he enlisted me to teach recitation sections for linear algebra. Eric Barth showed me the kind of professor, research mentor, and advisor that I someday hope to be. I am grateful also to Michael Murillo, Jeff Haack, and Gil Shohet, who were peerless research partners and mentors for

two idyllic summers spent at Los Alamos National Laboratory. Finally, at the University of Washington, I found myself embedded in a department with countless inspirational and helpful mentors. Ricardo Hidalgo and the staff at Hall Health were an incredible resource for learning to effectively maintain my own mental health. I do not know how I would have completed graduate school were it not for the confidence and support offered by Bernard Deconinck as both a professor and a graduate program coordinator. I have similarly leaned upon the teaching expertise of Mark Kot, and have him to thank for offering me my first chance at teaching a college course. Our administrative staff: Lauren Lederer, Alan Perry, Keshanie Dissayanake, and Tony Garcia cannot be thanked enough for the work they put into this department. We simply could not function without them.

I have been lucky to participate in a number of formative groups at the University of Washington. I would like to express deep gratitude to the members of the Diversity Committee. The work you do is important and I feel so lucky to have been a part of it. The mathematics staff at Math Science Upward Bound: Dave Wolczyk, Michael Coats, and Riki Omokawa also do tremendous work in pursuing equity in mathematics and I feel lucky to have worked with each of them. I am grateful to the Teaching Discussion Group. Each week, I look forward to our discussions and it makes me glad to know that the next generation of instructors shares my passion for teaching. Finally, a great deal of thanks goes to Jayadev Athreya, the Washington Experimental Mathematics Lab, and my research students: Jesse Rivera, Landon Shorack, and Qingtong Zeng. This chance to explore research mentorship was absolutely invaluable.

The members of my defense committee deserve a tremendous amount of gratitude. Though he is no longer on the committee, I am grateful to Jim Pfaendtner, who helped steer me towards a fruitful research area and supported my career goals compassionately. Jayadev Athreya generously replaced Jim on short notice and I am so pleased that he will be able to participate in my final defense. Randy LeVeque has modeled what it means to

be a great researcher and genuinely enjoyable person. Hong Qian's zest for mathematics is positively infectious, and I am glad to have had fascinating and mind-boggling discussions with him. Finally, I am grateful to my advisor, Panos Stinis. Our collaboration has been tremendously productive and fascinating. However, I am most grateful to him for the genuine support and care he has shown towards me. I could not ask for a better mentor and I am so pleased our paths crossed when they did.

I must thank the friends who have been with me through this marathon of a doctorate. Adam Crawford, Darren Caulley, and Eric Christensen: relaxing evenings of movies and board games help to keep me sane and I feel grateful to have met you. Brian DeSilva, Jeremy Upsal, Kelsey Marcinko, Chris Swierczewski, Felix Ye, and countless others make the department an enjoyable place in which to work. Finally, my first friends in Seattle come from my own graduate school cohort. Trevor Caldwell, Yian Ma, Kameron Harris, and Donsub Rim were always delightful people with whom to spend time. Lowell Thompson has been a riotously entertaining and knowledgable office mate for a number of years. I will fondly remember our times sharing beers at College Inn and complaining about Scorelator. Finally, Ben Segal was an exemplary office mate and friend. I will never forget our time solving puzzles, playing games, eating sushi, and even traveling to Cambodia. The department has not been the same since your graduation. In July 2017, I met Katie Levy. She swiftly became my life partner and greatest supporter. This last year of school should have felt grueling and intimidating, but going home to her made it breeze by effortlessly. She makes me a better researcher, teacher, and person. I do not know how I could begin to thank her enough, and I am excited to see where life takes us next.

Last and certainly not least, my family has offered me endless support and enthusiasm. My extended family are some of my greatest cheerleaders. My siblings, Alex and Maddie, make me laugh and cheer on every success I have been fortunate enough to encounter. My parents, Jackie and Vince, are the ones who have made everything possible. I owe to them

everything and it is not possible to even begin listing all the ways I am in their debt. I could not have done anything in my life without their inspirational example and unconditional support.

DEDICATION

To my parents: Jackie and Vince.

Chapter 1
INTRODUCTION

Any physical phenomenon can be understood and modeled at a number of different scales. Consider a system consisting of gas particles in a closed container:

1. At the most fundamental level, the subatomic particles comprising the gas particles can be represented as wavefunctions which interact according to the equations of quantum mechanics.

2. If quantum details are discounted, the gas particles can be modeled as classical particles that interact according to Newton's laws in a framework referred to as molecular dynamics.

3. However, if individual particle trajectories are disregarded, we can employ statistical mechanical arguments to represent the system in terms of particle distribution functions that evolve according to kinetic theory.

4. Finally, if nonequilibrium effects are not important, we can represent the system as a continuum that evolves according to hydrodynamics.

When one decides to model a physical system using one of these physical frameworks, one is making an important implicit claim: the lower scales of detail *are not important*. In many cases, this claim holds true. Newtonian mechanics accurately capture everyday physical observations despite neglecting quantum mechanical considerations. This is because the microscopic details lead to effects that, on average, cancel with one another rendering them unimportant for the evolution of the system as a whole.

Modeling an interesting physical system using a single scale requires choosing the scale that is detailed enough such that the phenomenon is observable but simple enough that extraneous details are omitted. This way of thinking about modeling systems is not limited to physics, either. Biological problems might take place at the scale of molecular dynamics within living cells, network dynamics in neuroscience on the scale of cell populations, or continuum representations of population dynamics and ecological models. Chemical problems run from the quantum mechanical considerations of orbital dynamics, to the classical dynamics of particle collisions and reactions, to the continuum representation modeled effectively in reaction-diffusion systems. Even purely mathematical questions can contain implicit and explicit scale separation. When we use a Taylor series to locally linearize a dynamical system, we assume the nonlinear effects that vary greatly on a large scale are nearly uniform in the region of interest.

However, there are many cases where the assumptions required to model a system at a single scale do not hold. For example, there may be systems in which the interesting phenomena occur on a scale consistent with one modeling framework, while the relevant dynamics occurs on a scale consistent with a more detailed modeling framework. A single problem might include some regions in time or space that satisfy the assumptions needed for one modeling framework while other regions can be represented in a higher or lower detailed framework. We call problems that display this challenging structure *multiscale* problems.

In each of the aforementioned cases, there is some sufficiently detailed scale that fully captures the problem at all relevant times and positions. One could then solve any such multiscale problem by solving the entire problem at the level of detail dictated by the most challenging points. However, this gives rise to prohibitively large problems. Of growing interest in applied mathematics, especially in the era of Big Data, is *reduced order modeling*. Reduced order modeling seeks to reduce a prohibitively large system to a computationally realizable problem size (resolution) while maintaining the essential features of the dynamics. When the large system is multiscale in nature, one can use knowledge of each scale and their relationship to one another in order to construct suitable reduced order models (ROMs).

Mathematical modeling and applied mathematics generally has a rich history of multiscale methods both analytic and numerical. Before discussing an overarching framework for multiscale modeling, it would be prudent to highlight some key examples of problems that require a multiscale approach.

1.1 Examples of multiscale systems

Every area of science is replete with examples of systems that benefit from a multiscale modeling approach. These application areas motivate and inform the development of robust multiscale methods.

1.1.1 Physical systems with localized disturbances

Certain physical events such as explosions and high energy impacts, give rise to *shock waves*. A shock wave is a local disturbance in the underlying medium in which key physical quantities such as temperature, pressure, and density change nearly discontinuously. As the system evolves, this disturbance propagates through the system and eventually dissipates. Away from a shock, the material is considered to be near equilibrium and can be modeled well with continuum dynamics with a very coarse discretization. Near the shock, however, the state variables considered in the continuum framework vary exceptionally quickly. These sharp gradients necessitate a far finer discretization than was needed elsewhere. In particularly strong shocks, such as those occuring in inertial confinement fusion or meteorite impacts, the sharpness of the shock necessitates consideration of molecular details at the shock front.

Similarly, in material science one is interested in the propagation of a crack through an otherwise homogeneous material under stress. Away from the crack, material variables are changing slowly. At the crack, and especially the crack tip, material variables change almost discontinuously. Molecular detail might be necessary at the crack tip in order to predict the path of the crack through the material.

In both of these cases, multiscale methods can be employed to couple the different physical models needed at the localized disturbance and in the more homogeneous medium. The local

nature of the disturbance is an example of *scale separation* in space, which simplifies the construction of a multiscale method. Some examples of multiscale approaches to physical systems with localized disturbances are found in [2, 91, 118].

1.1.2 Nanomaterials and biochemical dynamics

The behavior of microscopic materials in solution is of interest in the fields of nanomaterials and biochemistry. Consider, for example, a protein molecule. This consists of a string of amino acids that determines the primary structure of the protein. The protein is then folded into secondary, tertiary, and quaternary structures. Modeled physically, there are collections of atoms or groups of atoms bound together subject to frequent random perturbations from impacts with solvent molecules. Scientists interested in modeling the dynamics of complicated molecules like proteins and synthetic polymers must wrestle with the untenable size of these systems (each particle in the molecule of interest has six degrees of freedom, and one might also simulate the motion of the solvent molecules). Multiscale methods represent a possible manner of mitigating the problem size in this setting. In a technique known as *coarse-graining*, one can collect groups of particles into pseudo-atoms and infer the interactions of pseudo-atoms from previously observed microscopic data [48, 69, 77, 79, 80, 96].

1.1.3 Global climate models

Modeling the global climate is an exceptionally important and tremendously challenging problem. The underlying dynamics are relatively well-understood, as the climate obeys key physical laws such as the conservation of mass, energy, and momentum. This, coupled with radiation absorption and emission, allows one to represent the global climate as a three-dimensional fluid dynamics problem. However, the size of the problem is enormous and perturbations can have long-lasting and wide-reaching effects [98]. Even for the finest resolutions available today, sub-grid effects cannot be perfectly captured. If the resolution is refined, it produces a problem whose size dwarfs the capabilities of modern supercomputers.

The development of multiscale methods allows one to represent features at a higher level of detail without reducing the resolution to an unreasonable level. Promising results have been achieved by incorporating cloud resolving models over each simulation cell in place of conventional cloud parameterizations [56, 57, 72, 107]. This has been expanded to include multiscale modeling of aerosol pollutants [113–115]. This method of coupling detailed sub-cell models to larger scale fluid dynamics presents a promising solution to the challenges of insufficient resolution in climate models.

1.1.4 Biological modeling

Systems that arise in biology can range from the molecular scale, through individual cells and collections of cells (sometimes called the mesoscale), to entire organs or full-body physiology. It is not uncommon for interesting problems to span several of these scales of interest. Multiscale methods are critical for constructing analytical and numerical approximations of these problems. Our strong understanding of individual cellular behavior cannot be expanded to population level behavior without taking multiscale considerations into account [37]. They have been employed in reaction-diffusion systems to utilize microscopic detail only at relevant boundaries [49]. As discussed previously, multiscale approaches have seen wide success in structural biology [38]. Cancer modeling benefits from a multiscale connection between microscopic molecular dynamics, mesoscopic disease dynamics, and macroscopic physiological consequences [35, 95]. Systems biology is an inherently multiscale domain and explicit multiscale techniques are critical for effectively understanding problems in the field [17].

1.2 Multiscale techniques in applied mathematics

Beyond problems in physical, chemical, and biological application areas that are inherently multiscale, there is also a rich history of multiscale techniques developed in applied mathematics independent of specific applications. If the application problems are the projects of a carpenter, these techniques represent the tools that have been honed to be used in any number of unrelated projects.

1.2.1 The multigrid method

For some linear problems, iterative methods such as Jacobi iteration, Gauss-Seidel iteration, and successive overrelaxation may converge quickly at first but more slowly over time. In these cases the multigrid method can be used to reduce the computational load of the remaining convergence. After several iterative steps, the error typically becomes significantly smoother. One can iteratively solve equations for the error on a coarser grid. The coarser grid yields a smaller system that iterates more quickly. Furthermore, the ratio of the wavenumbers of error modes to the grid cell size is now smaller, and this ratio is what dictates the convergence rate.

By applying this method recursively, one constructs a multiscale structure of solution representations on ever coarser grids. Each grid is applied to progressively smoother error functions and accelerates the convergence for lower and lower frequencies of error. A comprehensive overview can be found in [65]. Multigrid methods have been used to great success in investigating partial differential equations arising in fluid dynamics such as the Navier-Stokes equation [41] and Euler's equations [54].

1.2.2 Singular perturbation techniques

We call a problem singularly perturbed when it has a small positive parameter ϵ, but the solution is *not* well-approximated by setting $\epsilon = 0$. One example is a singularly perturbed differential equation such as:

$$\epsilon u'' + u' + u = 0. \tag{1.1}$$

In this case, the presence of a nonzero ϵ changes the problem from a first order differential equation into a second order differential equation, which is a fundamentally different problem. Singularly perturbed problems have a multiscale structure built in. Namely, there is the $O(1)$ scale associated with the terms that are not multiplied by ϵ and the $O(\epsilon)$ scale of the terms associated with the small parameter.

Consequently, many solution methods for singularly perturbed problems approach the

problem from a multiscale framework. The method of matched asymptotic expansions is useful for boundary-layer problems, which have multiple spatial scales. A boundary-layer problem is a problem in which the singularly perturbed term is only significant in a highly local region of the domain of independent variables. The method of matched asymptotic expansions uses different solutions in the larger domain and in the localized boundary layer and matches the two solutions to meet in a sufficiently smooth manner in the intermediate region [62].

In some singularly perturbed problems, the impact of the small parameter is not localized in space. Instead, the small parameter creates multiple timescales ("fast" and "slow") of interest. Problems with multiple timescales are amenable to *multiple-scale analysis*. In this case, one explicitly introduces fast-scale and slow-scale versions of time and then expresses the solution as a perturbation series in which each term in the series depends upon both the fast and slow time variables. The flexibility of the extra variables allows one to eliminate problematic ("secular") terms. A comprehensive presentation of singularly perturbed problems, including multiscale approaches to their solutions, is provided in [9].

1.2.3 Adaptive mesh refinement

The adaptive mesh refinement method for solving partial differential equations is another example of a fundamentally multiscale applied mathematics technique. In adaptive mesh refinement, the solution to a partial differential equation is represented on an initially coarse spatial grid. As the solution evolves, some regions are relatively smooth and are well-represented by the coarse grid. Other regions develop strong gradients, such as when a shock occurs. In adaptive mesh refinement, the grid is automatically made finer in regions where higher resolution are needed. This is an example of, quite literally, using multiple spatial scales in the regions and at the times where they are needed. A thorough overview of adaptive mesh refinement can be found in [84]. Adaptive mesh refinement has been used to great effect in simulating cosmological hydrodynamics [108], shock propagation in fluids [10], and general hyperbolic partial differential equations [11].

1.2.4 Parallel tempering

Multiscale techniques have also been developed in the field of sampling methods. Markov Chain Monte Carlo (MCMC) sampling methods can be used to rigorously sample from a chosen probability distribution. These sampling techniques can be framed in terms of a "temperature" indicating the likelihood of accepting samples that significantly alter the energy of the system. By running multiple MCMC systems at different temperatures in parellel and exchanging configurations between two systems with a Metropolis-Hastings acceptance criterion, one can achieve a significant improvement in mixing properties. This results in a swifter exploration of the probability distribution even for low-temperature systems. An early overview of the technique is [34]. The method has been applied with great success to the molecular dynamics of protein folding [106]. A related idea is to use different spatial resolutions of a path through a stochastic dynamical system. One can construct a conditional path sampling method in which the spatial resolution plays the role of the temperature [116]. In these cases, one leverages the different behaviors of systems at different temperature scales or resolutions.

1.3 General multiscale frameworks

In this dissertation, we will be interested in multiscale methods as a general tool. Consequently, it would be useful to have a general and rigorous framework for multiscale methods and the related concept of reduced order modeling. Here we will discuss two such frameworks: the Mori-Zwanzig formalism and the heterogeneous multiscale method.

1.3.1 The Mori-Zwanzig formalism

Historically, the contributions of Robert Zwanzig and Hazime Mori to statistical mechanics allowed the rigorous discussion of reduced order modeling [120]. The Mori-Zwanzig formalism is an exact reduction of the dynamics of a full (large) system to the dynamics of a reduced set of variables. This reduction is done by viewing the set of variables in terms of Liouvillian

distributions. In this way, the unwanted variables can be treated probabilistically. The impact of the unwanted variables on the reduced set of variables may then be represented in a number of different ways, such as with conditional probabilities. Most generally, the impact of the unwanted variables on the reduced set of variables manifests as noise and memory terms. A memory term is a term that depends on the history of the trajectory of a variable up until the current time. It is fundamental to reduced order models, and represents a significant computational challenge.

The Mori-Zwanzig formalism is helpful conceptually, but the difficulty of computing the memory term precludes a simple implementation. Recently, there have been tremendous strides in approximating the memory effect. Optimal prediction and the t-model [8, 12, 23–25, 46, 103] and higher order memory approximations [99, 100] have led to effective reduced order models for some specific applications. The higher order approximations are susceptible to instabilities. One potential method for stabilizing these approximations draws on ideas from renormalization group theory in physics. The form of the problem terms are retained in a class of reduced order memory approximation schemes that are parametrized by "renormalization coefficients." These renormalization coefficients modify the problem terms into "effective terms." They are chosen to match observed quantities. Initial efforts to construct renormalized versions of these approximations [104, 105] show promise, but are not well understood.

1.3.2 The heterogeneous multiscale method

A more modern, but less rigorous, model reduction technique is the heterogeneous multiscale method (HMM), popularized by Weinan E [117, 118]. In this approach, a system is known to be described at two levels of detail, one coarse and one fine. The fine level is understood as the fully detailed system, and the coarse level is some dimension-reduced description of the full system that accurately captures the dynamics of a reduced number of variables. HMM, motivated by the original ideas of singular perturbation, is unique in that it uses physical arguments and constrained simulations of the fine system to inform undetermined

parameters in the coarse simulation. This technique relies on an explicit scale separation between the coarse dynamics and fine dynamics.

HMM is well suited to multiphysics problems, which inherently involve time or spatial scale separations. It has been successfully applied to gas dynamics problems using molecular dynamics to compute hydrodynamic fluxes [33], microfluidics and complex fluids [81, 91], finite temperature solid simulations [67], and more general fast-slow differential equation systems [55]. HMM, however, is not effective at handling systems in which there is no clear dividing line between scales.

1.4 Structure of dissertation

In this dissertation, we will present several extensions and applications of multiscale techniques derived from the Mori-Zwanzig formalism and the heterogeneous multiscale method. In Chapter 2, we will present the Mori-Zwanzig formalism in full detail and comment upon the inherent advantages and challenges of developing reduced order models from it. In Chapter 3, we will present the heterogeneous multiscale method in greater detail. We will demonstrate that this framework is a special case of the Mori-Zwanzig formalism. Finally, we will present an HMM method developed for plasma physics simulation and comment upon the Mori-Zwanzig interpretation of the derived equations. In Chapter 4, we will return to the Mori-Zwanzig formalism by presenting a new method of approximating the memory term called the "complete memory approximation." This method of deriving a reduced order model from the Mori-Zwanzig formalism is more general than previously developed methods. Applications to the Korteweg-de Vries equation, the three dimensional Euler's equations, and Burgers' equation are presented, including surprising results relating to the optimal scaling of introduced parameters. The code relating to these results is accessible in [87]. Finally, in Chapter 5, we will provide an overview of the results and comment upon future promising directions for the work described here.

Chapter 2

THE MORI-ZWANZIG FORMALISM

In this chapter, we will derive the Mori-Zwanzig formalism and present a simple example of its application to a linear problem.

2.1 Derivation of the Mori-Zwanzig formalism

Consider a system of autonomous ordinary differential equations

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{R}(\mathbf{u}) \tag{2.1}$$

augmented with an initial condition $\mathbf{u}(0) = \mathbf{u}^0$. For example, if one considers a partial differential equation, spectral or finite volume methods can be employed to convert a PDE to a system of ODEs. A solution trajectory for (2.1) can be thought of as a function of the initial condition and time ($\mathbf{u}(\mathbf{u}^0, t)$). The state of the system at any given time is determined entirely by the full initial condition and the time. Let $\mathbf{u}(t) = \{u_k(t)\}$, $k \in F \cup G$. We separate $\mathbf{u}(t)$ into resolved variables $\hat{\mathbf{u}} = \{u_i(t)\}$, $i \in F$ and unresolved variables $\tilde{\mathbf{u}} = \{u_j(t)\}$, $j \in G$ where F and G are disjoint. Let $R_k(\mathbf{u})$ be the kth entry in the vector-valued function $\mathbf{R}(\mathbf{u})$.

We can transform this nonlinear system of ODEs (2.1) into a linear system of PDEs by way of the Liouvillian operator, also known as the generator of the Koopman operator [61]:

$$\mathcal{L} = \sum_{k \in F \cup G} R_k(\mathbf{u}^0) \frac{\partial}{\partial u_k^0}. \tag{2.2}$$

The following proof of this claim is based upon [23]. Consider a smooth function $f(\mathbf{x})$ where $\mathbf{x}$ is a vector with the same number of components as $\mathbf{u}$. We will show that $\phi(\mathbf{u}^0, t) =$

$f(\mathbf{u}(\mathbf{u}^0, t))$ satisfies the PDE:

$$\frac{\partial \phi(\mathbf{u}^0, t)}{\partial t} = \mathcal{L}\phi, \quad \phi(\mathbf{u}^0, 0) = f(\mathbf{u}^0). \tag{2.3}$$

First, we can see that $\phi(\mathbf{u}^0, 0) = f(\mathbf{u}(\mathbf{u}^0, 0)) = f(\mathbf{u}^0)$. Next, we take the time derivative:

$$\frac{\partial \phi}{\partial t} = \sum_{k \in F \cup G} \frac{\partial f(\mathbf{u}(\mathbf{u}^0, t))}{\partial x_k} \frac{du_k}{dt} = \sum_{k \in F \cup G} R_k(\mathbf{u}(\mathbf{u}^0, t)) \frac{\partial f(\mathbf{u}(\mathbf{u}^0, t))}{\partial x_k} \tag{2.4}$$

where we used (2.1).

Next, we will show that

$$R_k(\mathbf{u}(\mathbf{u}^0, t)) = \sum_{j \in F \cup G} R_j(\mathbf{u}^0) \frac{\partial u_k(\mathbf{u}^0, t)}{\partial u_j^0}. \tag{2.5}$$

We begin by defining $F_k(\mathbf{u}^0, t) = R_k(\mathbf{u}(\mathbf{u}^0, t)) - \sum_{j \in F \cup G} R_j(\mathbf{u}^0) \frac{\partial u_k(\mathbf{u}^0, t)}{\partial u_j^0}$. Observe that the initial condition of F_k is:

$$F_k(\mathbf{u}^0, 0) = R_k(\mathbf{u}(\mathbf{u}^0, 0)) - \sum_{j \in F \cup G} R_j(\mathbf{u}^0) \frac{\partial u_k(\mathbf{u}^0, 0)}{\partial u_j^0} = R_k(\mathbf{u}^0) - \sum_{j \in F \cup G} R_j(\mathbf{u}^0) \frac{\partial u_k^0}{\partial u_j^0} = 0. \tag{2.6}$$

Now take the time derivative:

$$\begin{aligned}
\frac{\partial F_k(\mathbf{u}^0,t)}{\partial t} =& \frac{\partial}{\partial t} R_k(\mathbf{u}(\mathbf{u}^0,t)) - \sum_{j\in F\cup G} R_j(\mathbf{u}^0)\frac{\partial}{\partial u_j^0}\left(\frac{\partial u_k(\mathbf{u}^0,t)}{\partial t}\right) \\
=& \sum_{i\in F\cup G} \frac{\partial R_k(\mathbf{u}(\mathbf{u}^0,t))}{\partial x_i}\frac{\partial u_i(\mathbf{u}^0,t)}{\partial t} - \sum_{j\in F\cup G} R_j(\mathbf{u}^0)\frac{\partial R_k(\mathbf{u}(\mathbf{u}^0,t))}{\partial u_j^0} \\
=& \sum_{i\in F\cup G} \frac{\partial R_k(\mathbf{u}(\mathbf{u}^0,t))}{\partial x_i} R_i(\mathbf{u}(\mathbf{u}^0,t)) - \sum_{j\in F\cup G} R_j(\mathbf{u}^0) \sum_{i\in F\cup G} \frac{\partial R_k(\mathbf{u}(\mathbf{u}^0,t))}{\partial x_i}\frac{\partial u_i(\mathbf{u}^0,t)}{\partial u_j^0} \\
=& \sum_{i\in F\cup G} \left[\frac{\partial R_k(\mathbf{u}(\mathbf{u}^0,t))}{\partial x_i}\left(R_i(\mathbf{u}(\mathbf{u}^0,t)) - \sum_{j\in F\cup G} R_j(\mathbf{u}^0)\frac{\partial u_i(\mathbf{u}^0,t)}{\partial u_j^0}\right)\right] \\
=& \sum_{i\in F\cup G} \frac{\partial R_k(\mathbf{u}(\mathbf{u}^0,t))}{\partial x_i} F_i(\mathbf{u}^0,t).
\end{aligned}$$

Thus, we have shown that the PDE defining the evolution of F_k is linear with zero initial condition. Consequently, $F_k(\mathbf{u}^0,t) = 0$ for all $k \in F \cup G$ at all times. This means:

$$R_k(\mathbf{u}(\mathbf{u}^0,t)) = \sum_{j\in F\cup G} R_j(\mathbf{u}^0)\frac{\partial u_k(\mathbf{u}^0,t)}{\partial u_j^0}.$$

Returning to (2.4), we substitute this in:

$$\begin{aligned}
\frac{\partial \phi}{\partial t} =& \sum_{k\in F\cup G} R_k(\mathbf{u}(\mathbf{u}^0,t)\frac{\partial f(\mathbf{u}(\mathbf{u}^0,t))}{\partial x_k} \\
=& \sum\left(\sum_{j\in F\cup G} R_j(\mathbf{u}^0)\frac{\partial u_k(\mathbf{u}^0,t)}{\partial u_j^0}\right)\frac{\partial f(\mathbf{u}(\mathbf{u}^0,t))}{\partial x_k} \\
=& \sum_{j\in F\cup G} R_j(\mathbf{u}^0)\left(\sum_{k\in F\cup G}\frac{\partial f(\mathbf{u}(\mathbf{u}^0,t))}{\partial x_k}\frac{\partial u_k(\mathbf{u}^0,t)}{\partial u_j^0}\right) \\
=& \sum_{j\in F\cup G} R_j(\mathbf{u}^0)\frac{\partial}{\partial u_j^0}\left(f(\mathbf{u}(\mathbf{u}^0,t))\right) \\
=& \mathcal{L}\phi.
\end{aligned}$$

Thus, the nonlinear ODE (2.1) and linear PDE (2.3) are equivalent. In particular, if we

consider ϕ_k the solution of (2.3) with $f(\mathbf{u}^0) = u_k^0$, $\phi_k(\mathbf{u}^0, t) = u_k(t)$. In essence, we are constructing the partial differential equation for which our original ODE system defines the backward characteristic curves. Equation (2.3), however, is linear. Using semigroup notation, which states $\phi_k(\mathbf{u}^0, t) = e^{t\mathcal{L}} u_k^0$, we can write

$$\frac{du_k(t)}{dt} = \frac{\partial \phi_k(\mathbf{u}^0, t)}{\partial t} = \frac{\partial}{\partial t} e^{t\mathcal{L}} u_k^0 = e^{t\mathcal{L}} \mathcal{L} u_k^0. \tag{2.7}$$

This is the third and most convenient way of expressing the dynamics of the system given an initial condition.

Consider the space of functions that depend upon $\mathbf{u}^0$. Let P be an orthogonal projection onto the subspace of functions depending only on the resolved variables $\hat{\mathbf{u}}^0$. For example, Pf might be the conditional expectation of f given the resolved variables and an assumed distribution for the unresolved variables. Let $Q = I - P$. Then, we can decompose an evolution operator $e^{t\mathcal{L}}$ using Dyson's formula:

$$e^{t\mathcal{L}} = e^{tQ\mathcal{L}} + \int_0^t e^{(t-s)\mathcal{L}} P\mathcal{L} e^{sQ\mathcal{L}} \, \mathrm{d}s. \tag{2.8}$$

A brief proof of Dyson's formula follows. Define

$$g(t) = e^{tQ\mathcal{L}} + \int_0^t e^{(t-s)\mathcal{L}} P\mathcal{L} e^{sQ\mathcal{L}} \, \mathrm{d}s, \tag{2.9}$$

and then take the derivative:

$$\begin{aligned}
\frac{dg}{dt} =& Q\mathcal{L} e^{tQ\mathcal{L}} + \mathcal{L} e^{t\mathcal{L}} \int_0^t e^{-s\mathcal{L}} P\mathcal{L} e^{sQ\mathcal{L}} \, \mathrm{d}s + \left(e^{(t-t)\mathcal{L}} P\mathcal{L} e^{tQ\mathcal{L}}\right)(1) - (e^{(t-0)\mathcal{L}} P\mathcal{L})(0) \\
=& \mathcal{L}\left(e^{tQ\mathcal{L}} + \int_0^t e^{(t-s)\mathcal{L}} P\mathcal{L} e^{sQ\mathcal{L}} \, \mathrm{d}s\right) \\
=& \mathcal{L} g.
\end{aligned}$$

Observe that the intial condition is $g(0) = 1$. Thus, g satisfies the ODE:

$$\frac{dg}{dt} = \mathcal{L}g, \quad g(0) = 1 \tag{2.10}$$

which implies $g(t) = e^{t\mathcal{L}}$. This constitutes a proof of Dyson's formula. Applying Dyson's formula to (2.7) gives:

$$\frac{du_k}{dt} = e^{t\mathcal{L}}P\mathcal{L}u_k^0 + e^{tQ\mathcal{L}}Q\mathcal{L}u_k^0 + \int_0^t e^{(t-s)\mathcal{L}}P\mathcal{L}e^{sQ\mathcal{L}}Q\mathcal{L}u_k^0 \, \mathrm{d}s. \tag{2.11}$$

This is the Mori-Zwanzig identity. It represents an alternative, but exact way of writing the full dynamics. The first term on the right hand side in (2.11) is called the Markov term, because it depends only on the instantaneous values of the resolved variables. The second term is called "noise" and the third is called "memory". We again project the dynamics,:

$$\frac{dPu_k}{dt} = Pe^{t\mathcal{L}}P\mathcal{L}u_k^0 + P\int_0^t e^{(t-s)\mathcal{L}}P\mathcal{L}e^{sQ\mathcal{L}}Q\mathcal{L}u_k^0 \, \mathrm{d}s. \tag{2.12}$$

Here, we made use of the fact that

$$Pe^{tQ\mathcal{L}}Q\mathcal{L}u_k^0 = P\left[I + tQ\mathcal{L} + t^2(Q\mathcal{L})^2 + \dots\right]Q\mathcal{L}u_k^0 = 0$$

because $PQ = 0$. For $k \in F$, (2.12) describes the projected dynamics of the resolved variables. For a concrete example, if P is indeed a conditional expectation, then (2.12) describes the trajectories of the expected values of the resolved variables.

The system is not closed, however, due to the presence of the orthogonal dynamics operator $e^{sQ\mathcal{L}}$ in the memory term. In order to simulate the dynamics of (2.12) exactly, one needs to evaluate the second term which requires the dynamics of the unresolved variables. One key fact must be understood: reducing a large system to one of comparatively fewer variables necessarily introduces a memory term encoding the interplay between the unresolved and resolved variables. Dropping both the noise and memory terms and simulating

only the "average" dynamics (the Markov term) may not accurately reflect the dynamics of the resolved variables in the full simulation. Any multiscale dynamical model must in some way approximate or compute the memory term, or argue convincingly why the memory term is negligible. In fact, we will demonstrate that even in cases where the magnitude of the memory term is very small, it must be included in order to produce accurate simulations of the resolved modes.

2.2 Application to a linear system

For an informative example, we will apply the Mori-Zwanzig formalism to reduce a linear system of ODEs:

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u}, \quad \mathbf{u}(0) = \mathbf{u}^0. \tag{2.13}$$

Consider breaking the state vector into n resolved variables $\hat{\mathbf{u}}$ and m unresolved variables $\tilde{\mathbf{u}}$ such that $\mathbf{u} = [\hat{\mathbf{u}}, \tilde{\mathbf{u}}]^T$. Furthermore, we will partition the matrix A into block matrices:

$$A = \begin{bmatrix} \hat{A} & \hat{A}' \\ \tilde{A}' & \tilde{A} \end{bmatrix}$$

where $\hat{A}$ is $n \times n$, $\hat{A}'$ is $n \times m$, $\tilde{A}'$ is $m \times n$, and $\tilde{A}$ is $m \times m$. Under this definition, $\mathcal{L}$ is defined as:

$$\mathcal{L} = \sum_{k=1}^{n} \left(\sum_{i=1}^{n} \hat{a}_{ki}\hat{u}_i^0 + \sum_{i=1}^{m} \hat{a}'_{ki}\tilde{u}_i^0 \right) \frac{\partial}{\partial \hat{u}_k^0} + \sum_{j=1}^{m} \left(\sum_{i=1}^{n} \tilde{a}_{ji}\hat{u}_i^0 + \sum_{i=1}^{m} \tilde{a}'_{ji}\tilde{u}_i^0 \right) \frac{\partial}{\partial \hat{u}_j^0}, \tag{2.14}$$

or more succinctly,

$$\mathcal{L}\hat{\mathbf{u}}^0 = \hat{A}\hat{\mathbf{u}}^0 + \hat{A}'\tilde{\mathbf{u}}^0, \qquad \mathcal{L}\tilde{\mathbf{u}}^0 = \tilde{A}\hat{\mathbf{u}}^0 + \tilde{A}'\tilde{\mathbf{u}}^0.$$

Define our projector P such that $P\hat{\mathbf{u}}^0 = \hat{\mathbf{u}}^0$ and $P\tilde{\mathbf{u}}^0 = 0$. Then, we can use the Mori-Zwanzig formalism to write a new equation for the evolution of the resolved variables $\hat{\mathbf{u}}$:

$$\frac{d\hat{\mathbf{u}}}{dt} = \underbrace{e^{t\mathcal{L}}P\mathcal{L}\hat{\mathbf{u}}^0}_{\text{Markov term}} + \underbrace{e^{tQ\mathcal{L}}Q\mathcal{L}\hat{\mathbf{u}}^0}_{\text{Noise term}} + \underbrace{\int_0^t e^{(t-s)\mathcal{L}}P\mathcal{L}e^{sQ\mathcal{L}}Q\mathcal{L}\hat{\mathbf{u}}^0\,\mathrm{d}s}_{\text{Memory term}}.$$

The Markov term is:

$$e^{t\mathcal{L}}P\mathcal{L}\hat{\mathbf{u}}^0 = e^{t\mathcal{L}}P\left[\hat{A}\hat{\mathbf{u}}^0 + \hat{A}'\tilde{\mathbf{u}}^0\right] = \hat{A}\hat{\mathbf{u}}.$$

The noise term can be written as the solution to:

$$\frac{\partial e^{tQ\mathcal{L}}Q\mathcal{L}\hat{\mathbf{u}}^0}{\partial t} = e^{tQ\mathcal{L}}Q\mathcal{L}Q\mathcal{L}\hat{\mathbf{u}}^0. \tag{2.15}$$

First we find $Q\mathcal{L}\hat{\mathbf{u}}^0$:

$$Q\mathcal{L}\hat{\mathbf{u}}^0 = \hat{A}\hat{\mathbf{u}}^0 + \hat{A}'\tilde{\mathbf{u}}^0 - \hat{A}\hat{\mathbf{u}}^0 = \hat{A}'\tilde{\mathbf{u}}^0. \tag{2.16}$$

Next we apply $Q\mathcal{L}$ again:

$$\begin{aligned}
\mathcal{L}Q\mathcal{L}\hat{\mathbf{u}}^0 =& \mathcal{L}\hat{A}'\tilde{\mathbf{u}}^0 = \hat{A}'\left[\tilde{A}'\hat{\mathbf{u}}^0 + \tilde{A}\tilde{\mathbf{u}}^0\right] \\
P\mathcal{L}Q\mathcal{L}\hat{\mathbf{u}}^0 =& P\left[\hat{A}'\left(\tilde{A}'\hat{\mathbf{u}}^0 + \tilde{A}\tilde{\mathbf{u}}^0\right)\right] = \hat{A}'\tilde{A}'\hat{\mathbf{u}}^0 \\
Q\mathcal{L}Q\mathcal{L}\hat{\mathbf{u}}^0 =& \hat{A}'\tilde{A}\tilde{\mathbf{u}}^0.
\end{aligned}$$

In fact, (2.15) is an ODE for the quantity $\mathbf{z} = e^{tQ\mathcal{L}}Q\mathcal{L}\hat{\mathbf{u}}^0 = e^{tQ\mathcal{L}}\hat{A}'\tilde{\mathbf{u}}^0$. We define:

$$B = \hat{A}'\tilde{A}(\hat{A}'^*\hat{A}')^{-1}\hat{A}'^* \tag{2.17}$$

where $B\hat{A}' = \hat{A}'\tilde{A}$. Then we see:

$$\begin{aligned}\frac{\partial e^{tQ\mathcal{L}}Q\mathcal{L}\hat{\mathbf{u}}^0}{\partial t} =& e^{tQ\mathcal{L}}Q\mathcal{L}Q\mathcal{L}\hat{\mathbf{u}}^0\\ \frac{d\mathbf{z}}{dt} =& e^{tQ\mathcal{L}}\hat{A}'\tilde{A}\tilde{\mathbf{u}}^0\\ =& e^{tQ\mathcal{L}}B\hat{A}'\tilde{\mathbf{u}}^0\\ =& B\mathbf{z}.\end{aligned}$$

The solution to this linear system is:

$$e^{tQ\mathcal{L}}Q\mathcal{L}\hat{\mathbf{u}}^0 = e^{tB}\mathbf{z}(0) = e^{tB}\hat{A}'\tilde{\mathbf{u}}^0. \tag{2.18}$$

Before continuing, we will show $e^{tB}\hat{A}' = \hat{A}'e^{t\tilde{A}}$. First, recall that $B\hat{A}' = \hat{A}'\tilde{A}$. Now we will show by induction that $B^k\hat{A}' = \hat{A}'\tilde{A}^k$. Suppose $B^{k-1}\hat{A}' = \hat{A}'\tilde{A}^{k-1}$. Then,

$$B^k\hat{A}' = BB^{k-1}\hat{A}' = B\hat{A}'\tilde{A}^{k-1} = \hat{A}'\tilde{A}^k.$$

Consequently, $e^{tB}\hat{A}' = \hat{A}'e^{t\tilde{A}}$. Thus, the noise term can be written as:

$$e^{tQ\mathcal{L}}Q\mathcal{L}\hat{\mathbf{u}}^0 = e^{tB}\mathbf{z}(0) = e^{tB}\hat{A}'\tilde{\mathbf{u}}^0 = \hat{A}'e^{t\tilde{A}}\tilde{\mathbf{u}}^0.$$

We can plug this into the memory integral to simplify it:

$$\begin{aligned}
\int_0^t e^{(t-s)\mathcal{L}} P\mathcal{L} e^{sQ\mathcal{L}} Q\mathcal{L}\hat{\mathbf{u}}^0 \,\mathrm{d}s &= \int_0^t e^{(t-s)\mathcal{L}} P\mathcal{L}\hat{A}' e^{s\tilde{A}}\tilde{\mathbf{u}}^0 \,\mathrm{d}s \\
&= \int_0^t e^{(t-s)\mathcal{L}} \hat{A}' e^{s\tilde{A}} P\left[\tilde{A}'\hat{\mathbf{u}}^0 + \tilde{A}\tilde{\mathbf{u}}^0\right] \,\mathrm{d}s \\
&= \int_0^t e^{(t-s)\mathcal{L}} \hat{A}' e^{s\tilde{A}} \hat{A}'\tilde{A}'\hat{\mathbf{u}}^0 \,\mathrm{d}s \\
&= \hat{A}' \int_0^t e^{s\tilde{A}} \tilde{A}'\hat{\mathbf{u}}(t-s) \,\mathrm{d}s \\
&= \hat{A}' \int_0^t e^{(t-s)\tilde{A}} \tilde{A}'\hat{\mathbf{u}}(s) \,\mathrm{d}s.
\end{aligned}$$

Therefore, the Mori-Zwanzig formalism defines the following evolution equation for the resolved modes:

$$\frac{d\hat{\mathbf{u}}}{dt} = \hat{A}\hat{\mathbf{u}} + \hat{A}' e^{t\tilde{A}}\tilde{\mathbf{u}}^0 + \hat{A}' \int_0^t e^{(t-s)\tilde{A}} \tilde{A}'\hat{\mathbf{u}}(s) \,\mathrm{d}s. \tag{2.19}$$

We can also find this same result by using an integrating factor to solve for the unresolved modes:

$$\begin{aligned}
\frac{d\tilde{\mathbf{u}}}{dt} &= \tilde{A}'\hat{\mathbf{u}} + \tilde{A}\tilde{\mathbf{u}} \\
\frac{d\tilde{\mathbf{u}}}{dt} - \tilde{A}\tilde{\mathbf{u}} &= \tilde{A}'\hat{\mathbf{u}} \\
e^{-t\tilde{A}}\left(\frac{d\tilde{\mathbf{u}}}{dt} - \tilde{A}\tilde{\mathbf{u}}\right) &= e^{-t\tilde{A}}\tilde{A}'\hat{\mathbf{u}} \\
e^{-t\tilde{A}}\tilde{\mathbf{u}} + C &= \int_0^t e^{-s\tilde{A}}\hat{\mathbf{u}}(s) \,\mathrm{d}s \\
\tilde{\mathbf{u}} &= e^{t\tilde{A}}\tilde{\mathbf{u}}^0 + \int_0^t e^{(t-s)\tilde{A}}\hat{\mathbf{u}}(s) \,\mathrm{d}s.
\end{aligned}$$

We plug this solution into the ODE for the resolved modes:

$$\begin{aligned}\frac{d\hat{\mathbf{u}}}{dt} =& \hat{A}\hat{\mathbf{u}} + \hat{A}'\tilde{\mathbf{u}} \\ =& \hat{A}\hat{\mathbf{u}} + \hat{A}' e^{t\tilde{A}}\tilde{\mathbf{u}}^0 + \hat{A}' \int_0^t e^{(t-s)\tilde{A}} \tilde{A}' \hat{\mathbf{u}}(s)\, \mathrm{d}s.\end{aligned}$$

2.2.1 Two-dimensional example

As a concrete example, consider the very simple system:

$$\begin{cases} \frac{dx}{dt} = x + y \\ \frac{dy}{dt} = x - y \end{cases}, \quad x(0) = x^0, \;\; y(0) = y^0. \tag{2.20}$$

Let us consider x the resolved variable and y the unresolved variable. In this case, there is not an obvious choice for "resolved" and "unresolved" variables. However, the Mori-Zwanzig formalism does not rely upon a clear separation between the two classes. It is my hope that this example helps to demonstrate the universality of the Mori-Zwanzig formalism approach to model reduction. In this case $\hat{A} = 1$, $\hat{A}' = 1$, $\tilde{A}' = 1$, and $\tilde{A} = -1$. This means $B = (1)(-1)((1)(1))^{-1}(1) = -1$. Therefore, the Mori-Zwanzig formalism reorganizes the ODE for x into:

$$\frac{dx}{dt} = x + e^{-t}y^0 + \int_0^t e^{-(t-s)} x(s)\, \mathrm{d}s. \tag{2.21}$$

If we were to instead solve for $y(t)$ explicitly using an integrating factor, we would find

$$y(t) = e^{-t}y^0 + \int_0^t e^{-(t-s)} x(s)\, \mathrm{d}s.$$

Upon plugging this into $\frac{dx}{dt} = x + y$, we are left with the same result as in (2.21). Thus, we see that when applied to a linear problem with a particular choice of P, the Mori-Zwanzig formalism yields the same solution as found through the integrating factor method.

Chapter 3

HETEROGENEOUS MULTISCALE METHOD

The heterogenous multiscale method (HMM), popular in the computational physics literature, represents another way of treating a system with multiple scales (resolved and unresolved). In HMM, we begin with two physical descriptions of the same system at different levels of detail. For example, the fine-grained scale might be molecular dynamics equations describing the trajectories of each individual atom and molecule in a fluid system, while the coarse-grained system might be hydrodynamic equations describing the evolution of the pressure and temperature of the same system. The coarse-grained system is also, in some way, incomplete. It may include parameters that are empirically determined, or it may break down under certain conditions. HMM provides a methodology for marrying the two scales of dynamics in such a way that the coarse model accurately reproduces the dynamics of the coarse variables as they would be in the full system. The emphasis is on increasing the accuracy of the coarse level by informing it through the fine level.

E designates the two most common types of HMM type A and type B [118]. In type A problems, the assumptions of the coarse model are broken by some spatially isolated defect or singularity. For example, if the coarse model is a hydrodynamic description, shocks or cracks do not evolve correctly because they break some of the assumptions (such as smoothness) that are used to derive the model. This is remedied by computing the fine simulation only in the neighborhood of the problem region and using those to set the ill-defined parameters. This represents a spatial separation of scales. Type A problems can be understood as a computational physics version of *matched asymptotics*. When microscale dynamics are spatially constrained, the macroscale and microscale dynamics can be treated as two coupled problems in which the slowly varying macroscale provides boundary conditions

to the quickly varying microscale.

In type B problems, the coarse model is not fully defined at any point in the domain. It contains some parameters, such as stress tensors or relaxation times, that must be prescribed. Historically, these terms have been found empirically, but this limits the types of simulations to only ones that can be reproduced in a laboratory setting. The strategy of HMM is to use small fine-grained simulations constrained by the local state of the coarse simulation to inform the undefined parameters. This process can be executed after every coarse-grain time step, or only when the parameters can be expected to have changed significantly. This represents a temporal separation of scales. Solutions to type B problems are a computational physics version of *multiple-scale analysis*. There are two timescales of relevance and one must simultaneously solve for both scales in order to accurately reproduce the solution.

In both types of problems, we can consider separately the fine-grained dynamics and the coarse-grained dynamics. Let $\mathbf{u}(t)$ be the state vector of the full (fine-grained) system and $\hat{\mathbf{u}}(t)$ be the state vector of the reduced (coarse-grained) system. *A priori* we have possibly nonlinear evolution equations for both the full and coarse-grained variables:

$$\frac{d\mathbf{u}}{dt} = \mathbf{R}(\mathbf{u}), \quad \frac{d\hat{\mathbf{u}}}{dt} = \hat{\mathbf{R}}(\hat{\mathbf{u}}, \alpha(\mathbf{u})) \tag{3.1}$$

where $\alpha(\mathbf{u})$ is the vector of parameters needed to close the reduced system. HMM connects the two scales by allowing α to be a function of the full state $\mathbf{u}(t)$ and using a fictitious, local approximation of the microscale to inform the macroscale through these parameters. Obviously, since we are not fully solving the problem for the microscale variables, we will need to develop a framework to approximate the parameter function given only the resolved variables.

3.1 Connecting Mori-Zwanzig and HMM

The Mori-Zwanzig formalism is a purposefully general and rigorous description of model reduction: the derivation of a reduced model for only the resolved variables. The end result

of HMM is a reduced model for only the coarse-grained variables that can be evolved nearly independently of the full system. In both cases, the effects of the full system persist in some way. It is reasonable to posit that the two methods are related, but to the author's knowledge this connection has not been explicitly articulated in the literature.

HMM can be reexpressed in terms of the general Mori-Zwanzig formalism, and the role of the memory term in HMM's effectiveness can be made clear. First, one must identify what is playing the role of the projection operator P in HMM. Statistical mechanics is devoted in part to studying the projection of detailed microscale dynamics onto lower order macroscale variables. Thus, P is often an expected value over specific ensembles of microscale states, with additional assumptions to eliminate or simplify complex terms. The Mori-Zwanzig formalism requires the macroscale variables to be expressible in terms of the microscale variables. For multiphysics problems, this is typically the case by definition. For example, the microscopic temperature is given by

$$T = \frac{1}{3N} \sum_{i=1}^{N} \frac{|\mathbf{p}_i|^2}{2m_i}$$

where N is the number of particles in the system, m_i is the mass of particle i, and $\mathbf{p}_i$ is the momentum of particle i. The expected value of this quantity is the definition of the hydrodynamic temperature, a macroscopic variable.

For a given macroscopic variable $\hat{u}_k$, we must identify the microscopic version $\hat{u}'_k$, and it must satisfy a few constraints. It must depend only on variables already present in the full system. It must also be chosen such that $P\hat{u}'_k = \hat{u}_k$. The physical arguments used to derive macroscopic dynamics from microscopic descriptions will often naturally produce these relationships. Common macroscale quantities such as temperature, pressure, density, and distribution functions satisfy these constraints.

Once these definitions are made, the Mori-Zwanzig formulation can be used to express the correct HMM dynamics and reveal what memory terms are present. The projection P should be applied to the microscale differential equations for the $\hat{u}'_k$ variables. This is the

process of deriving macroscopic evolution equations from their microscopic definitions. All macroscale descriptions of physical systems can be derived from the more detailed microscale systems under particular assumptions. After simplification, the resulting equations will depend largely on the $\hat{\mathbf{u}}$ in a term we designate $\hat{\mathbf{R}}^0(\hat{\mathbf{u}})$, but will typically also include some unsimplified term involving the full dynamics $\hat{\mathbf{R}}'(\mathbf{u})$. Often assumptions about these terms allow one to approximate them, though these assumptions typically introduce unknown parameters. These are the α parameters that are not fully determined in the macroscopic model.

The parallels between HMM and Mori-Zwanzig help us now better understand what these approximate terms represent. They correspond to the noise and memory terms of Mori-Zwanzig: quantities whose dynamics cannot be understood exclusively through functions of the reduced variables. In HMM, we make assumptions about the forms of these unknown terms, then capitalize on the structure of the projection operator P to compute the missing data. Though P is a many-to-one transformation, statistical mechanical arguments are made that any representative microscale state should provide the necessary memory information.

The form of the P in HMM is explicitly chosen in such a way that a pseudo inverse is possible. This is a (possibly noisy) map $P^\dagger$ from the resolved set of variables back to the full set of variables (or a suitable subset of them using statistical mechanical arguments). The result is a microscale simulation constrained by the macroscale state. The missing data can then be computed within this constrained microscale simulation. This entire procedure determines $\hat{\alpha}(\hat{\mathbf{u}})$. This result is used as an approximation of $\alpha(\mathbf{u})$ to evolve the macroscale state in HMM according to

$$\frac{d\hat{\mathbf{u}}}{dt} = \hat{\mathbf{R}}^0(\hat{\mathbf{u}}) + \hat{\mathbf{R}}'(\hat{\mathbf{u}}, \hat{\alpha}(\hat{\mathbf{u}})). \tag{3.2}$$

Computing the noise and memory terms directly is possible in HMM specifically because of scale separation. Scale arguments are used to reduce the size of the microscale simulation that must be completed in order to ascertain these quantities. For example, if a parameter is known to be defined locally, a simulation of the full dynamics in a region much smaller

than the full system may be used to approximate the parameter locally. If the macroscale model has sufficient resolution, this provides all the necessary information. Thus, HMM represents a method for approximating the memory terms that necessarily arise in reduced order dynamics "on the fly." It can be reframed in terms of the Mori-Zwanzig formalism. The projection P and the memory terms are treated without explicitly acknowledging them as such. It is less general than the Mori-Zwanzig formalism, though, in that it uses assumptions and bootstrap methods to approximate the noise and memory terms, rather than computing or approximating them directly.

3.2 HMM applied to plasma simulations

The research described in this section was conducted at Los Alamos National Laboratory with Jeff Haack, Gil Shohet, and Michael Murillo. The work described here is being prepared for submission to the Journal of Computational Physics, though it is still preliminary. This section includes contributions from the aforementioned individuals. It has been reproduced with their permission.

3.2.1 Introduction

HMM has been successfully applied to gas dynamics problems using molecular dynamics to compute hydrodynamic fluxes [33], as well as dynamic crack propagation [2], and heat conduction in fine scale structures [1]. Multiscale domain decomposition methods have been used in neutral gas kinetic problems to couple a local kinetic Boltzmann model to the Navier-Stokes equations to model hypersonic flows [63]. However, the HMM philosophy has not been applied to warm and hot dense plasma systems, in which neither pure kinetic nor molecular dynamics models are satisfactory for probing the length and time scales of interest. There have been multiscale models that couple kinetic electron motion to molecular dynamics models for ionic motion to simulate small systems of ultracold neutral plasmas [86] and hot dense plasmas [44, 74], however these do not fit within the HMM framework.

In this section, we describe a multiscale kinetic-MD model based on the HMM philosophy,

in which the relaxation times used in the kinetic multispecies Vlasov-Bhatnagar-Gross-Krook (VBGK) model are computed from localized MD simulations on a small subset of the physical domain of interest. The result is an adaptive hybrid simulation method that employs MD-accurate relaxation times to significantly improve the accuracy of the VBGK model, while requiring significantly less computational time than would be required by a full MD simulation. The multiscale coupling between the models is handled through "middleware", which interfaces between the MD and kinetic code and dynamically builds a linear regression model based on previous MD data to further reduce expensive MD calls.

3.2.2 Connecting molecular dynamics and kinetic models

At the microscopic level, a classical N-particle plasma system is characterized by the trajectories of the constituent particles in phase space. Given interparticle interaction potentials, the N-body Hamiltonian fully describes the dynamics of the system, and the evolution is uniquely determined by the initial conditions in phase space. Simulations of this type are called molecular dynamics; models of this type have been developed for various physical regimes [110]. In plasma systems in which microscopic effects are important, MD is the natural model for both theory and simulations [39,47,70], however it is limited by the maximum system size [85]. State of the art MD simulations can evolve $N = O(10^{12})$ particle systems, however these systems correspond to domains on the order of several μm^3 [78], which is much smaller than e.g. the size of an inertial confinement fusion capsule. Furthermore, MD typically can only probe time scales on the order of picoseconds to microseconds, and pose an additional challenge in that massive calculations can produce tens of terabytes of data per time step [40].

Rather than simulating individual particles, kinetic theory (KT) models describe the evolution of particle distributions $f_k(\mathbf{r}, \mathbf{v}, t)$ of each species k, which increase the time and space scales that can be simulated when compared to MD. The temporal and spatial scales that can be simulated are limited mainly by the dynamics of the system and the complexity of the collisional model. Kinetic models are appropriate in regimes where one expects

nonequilibrium phenomena; this can be quantified by the Knudsen number $\mathcal{K}$, which is the ratio of the mean free path between particle collisions and a characteristic length scale of the system. For very small Knudsen numbers, the collisions drive the system to equilibrium very quickly and hydrodynamic models are sufficient to describe the dynamics. However, for $\mathcal{K} > 0.1$, the assumptions of hydrodynamics begin to break down. These conditions occur in a variety of applications, in particular during the shock convergence phase of intertial confinement fusion (ICF) [4, 5, 50, 83, 92, 94, 119]. There is a rich history of plasma physics kinetic models; see [13, 14, 20, 59, 112]. Below, we mathematically derive the multispecies Vlasov-BGK (VBGK) kinetic equation from molecular dynamics, and use it to motivate the HMM model.

Kinetic equations can be explicitly derived from the molecular dynamics N-body Hamiltonian given an interaction potential. In order to avoid the complexities of evolving the electron motion we consider only the ionic motion. The screening effect of electrons is modeled implicitly through the Yukawa interparticle potential:

$$U_{kl}(\mathbf{r}, \mathbf{r}') = \frac{Z_k Z_l e^2}{4\pi|\mathbf{r} - \mathbf{r}'|} e^{-|\mathbf{r}-\mathbf{r}'|/\lambda}. \tag{3.3}$$

where λ is the screening length. The plasma system may be comprised of one or more species of ions, where ions of species k have mass m_k, number density n_k, temperature T_k, and charge $Z_k e$ (where e is the unit charge).

In a classical plasma system, the molecular dynamics simulation evolves every particle according to the Hamiltonian equations

$$\frac{d\mathbf{r}_i}{dt} = \frac{1}{m_i}\nabla_{\mathbf{v}_i} H, \qquad \frac{d\mathbf{v}_i}{dt} = -\frac{1}{m_i}\nabla_{\mathbf{r}_i} H \tag{3.4}$$

$$H(\{\mathbf{r}_i\}_{i=1}^N, \{\mathbf{v}_i\}_{i=1}^N) = \sum_i \left[\frac{m_i|\mathbf{v}_i|^2}{2} + \frac{1}{2}\sum_{j\neq i} U_{ij}(\mathbf{r}_i, \mathbf{r}_j)\right], \tag{3.5}$$

where $(\mathbf{r}_i(t), \mathbf{v}_i(t))$ is the position and velocity of particle i at time t, and U_{ij} is the Yukawa interparticle potential between particles i and j, defined in (3.3), where i is a particle of

species k and j is a particle of species l.

To derive a kinetic equation from this particle system, we begin by constructing a Klimontovich distribution [60] N_k for each species k that indicates the location $\mathbf{r}_i$ and velocity $\mathbf{v}_i$ of every particle of that species at a given time t:

$$N_k(\mathbf{r}, \mathbf{v}, \{\mathbf{r}_\alpha\}_{\alpha=1}^N, \{\mathbf{v}_\alpha\}_{\alpha=1}^N, t) = \sum_{i \in S_k} \delta\left(\mathbf{r} - \mathbf{r}_i\right) \delta\left(\mathbf{v} - \mathbf{v}_i\right). \tag{3.6}$$

$\delta(x)$ is the Dirac delta function, and S_k is the set of ion indices corresponding to species k. The ensemble expected value of N_k is the definition of f_k:

$$f_k(\mathbf{r}, \mathbf{v}, t) \equiv \langle N_k \rangle. \tag{3.7}$$

This ensemble average is taken with respect to all initial choices for $\mathbf{r}_i(0)$ and $\mathbf{v}_i(0)$ consistent with the initial distributions $f_k(\mathbf{r}, \mathbf{v}, 0)$. This defines the ensemble over which we consider each average. We will take the time derivative of N_k first, and then take the expected value.

$$\begin{aligned}
\frac{\partial N_k}{\partial t} &= \sum_{i \in S_k} \frac{\partial}{\partial t} \delta(\mathbf{r} - \mathbf{r}_i) \delta(\mathbf{v} - \mathbf{v}_i) \\
&= \sum_{i \in S_k} \left\{ \delta(\mathbf{v} - \mathbf{v}_i) \frac{\partial}{\partial t} \delta(\mathbf{r} - \mathbf{r}_i) + \delta(\mathbf{r} - \mathbf{r}_i) \frac{\partial}{\partial t} \delta(\mathbf{v} - \mathbf{v}_i) \right\} \\
&= \sum_{i \in S_k} \left\{ - \frac{\partial \mathbf{r}_i(t)}{\partial t} \delta(\mathbf{v} - \mathbf{v}_i) \cdot \nabla_{\mathbf{r}} \delta(\mathbf{r} - \mathbf{r}_i) - \frac{\partial \mathbf{v}_i}{\partial t} \delta(\mathbf{r} - \mathbf{r}_i) \cdot \nabla_{\mathbf{v}} \delta(\mathbf{v} - \mathbf{v}_i) \right\}.
\end{aligned} \tag{3.8}$$

At this point, we can substitute in the Hamiltonian equations for the evolution of $\mathbf{r}_i$ and $\mathbf{v}_i$:

$$\frac{\partial N_k}{\partial t} = \sum_{i \in S_k} \left\{ - \mathbf{v}_i \delta(\mathbf{v} - \mathbf{v}_i) \cdot \nabla_{\mathbf{r}} \delta(\mathbf{r} - \mathbf{r}_i) + \frac{1}{m_k} \delta(\mathbf{r} - \mathbf{r}_i) \nabla_{\mathbf{r}_i} \left[\sum_l \sum_{\substack{j \in S_l \\ j \neq i}} U_{kl}(\mathbf{r}_i, \mathbf{r}_j) \right] \cdot \nabla_{\mathbf{v}} \delta(\mathbf{v} - \mathbf{v}_i) \right\}. \tag{3.9}$$

Because $\mathbf{v}_i\delta(\mathbf{v}-\mathbf{v}_i)$ is only nonzero when $\mathbf{v}=\mathbf{v}_i$, we find

$$\mathbf{v}_i\delta(\mathbf{v}-\mathbf{v}_i)=\mathbf{v}\delta(\mathbf{v}-\mathbf{v}_i). \tag{3.10}$$

A similar relation holds for the second term. Using this and simplifying, we find

$$\frac{\partial N_k}{\partial t}=-\mathbf{v}\cdot\nabla_{\mathbf{r}}N_k+\frac{1}{m_k}\nabla_{\mathbf{r}}\left[\sum_l\sum_{j\in S_l}U_{\{kl\}}(\mathbf{r},\mathbf{r}_j)\right]\cdot\nabla_{\mathbf{v}}N_k. \tag{3.11}$$

We now take the ensemble average of both sides of this equation and obtain

$$\frac{\partial f_k}{\partial t}=-\mathbf{v}\cdot\nabla_{\mathbf{r}}f_k+\frac{1}{m_k}\left\langle\nabla_{\mathbf{r}}\left[\sum_l\sum_{j\in S_l}U_{\{kl\}}(\mathbf{r},\mathbf{r}_j)\right]\cdot\nabla_{\mathbf{v}}N_k\right\rangle. \tag{3.12}$$

In order to simplify the second term in (3.12), we use the definition of N_l,

$$\sum_{j\in S_l}U_{kl}(\mathbf{r},\mathbf{r}_j)=\iint U_{kl}(\mathbf{r},\mathbf{r}')N_l(\mathbf{r}',\mathbf{v}',t)\,\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{v}', \tag{3.13}$$

to obtain

$$\left\langle\nabla_{\mathbf{r}}\left[\sum_l\sum_{j\in S_l}U_{\{kl\}}(\mathbf{r},\mathbf{r}_j)\right]\cdot\nabla_{\mathbf{v}}N_k\right\rangle=\sum_l\iint\nabla_{\mathbf{r}}U_{kl}(\mathbf{r},\mathbf{r}')\cdot\nabla_{\mathbf{v}}\left\langle N_k(\mathbf{r},\mathbf{v})N_l(\mathbf{r}',\mathbf{v}')\right\rangle\,\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{v}'. \tag{3.14}$$

The product of two Klimontovich distribution functions is the two-species Klimontovich distribution function for that species pair, $N_{kl}(\mathbf{r},\mathbf{v},\mathbf{r}',\mathbf{v}',\{\mathbf{r}_i\}_{i=1}^n,\{\mathbf{v}_j\}_{j=1}^n,t)$. The expected value is the definition of the two species distribution function f_{kl}:

$$f_{kl}(\mathbf{r},\mathbf{v},\mathbf{r}',\mathbf{v}',t)\equiv\langle N_{kl}\rangle \tag{3.15}$$

Unfortunately, we have introduced this new unknown function f_{kl}, when we wish to express our kinetic system as a partial differential equation for f_k. Because the two-particle distri-

bution function can be written as the product of two one-particle distribution function when the two species are not correlated, we write f_{kl} without loss of generality as

$$f_{kl}(\mathbf{r}, \mathbf{v}, \mathbf{r}', \mathbf{v}', t) = f_k(\mathbf{r}, \mathbf{v}, t) f_l(\mathbf{r}', \mathbf{v}', t) + C_{kl} \tag{3.16}$$

where C_{kl} is a complicated remainder function. This can be thought of as separating interactions into mean field interactions and deviations from the mean field. This substitution yields:

$$\left\langle \nabla_{\mathbf{r}} \left[\sum_l \sum_{j \in S_l} U_{kl}(\mathbf{r}, \mathbf{r}_j) \right] \cdot \nabla_{\mathbf{v}} [N_k] \right\rangle = \nabla_{\mathbf{r}} \sum_l \left(\int U_{kl}(\mathbf{r}, \mathbf{r}') n_l(\mathbf{r}', t) \, \mathrm{d}\mathbf{r}' \right) \cdot \nabla_{\mathbf{v}} f_k + C'_{kl}. \tag{3.17}$$

C'_{kl} is a related remainder function describing the interactions between species k and species l not characterized by the mean field approximation.

To this point, we have not made use of our specific MD potential function, the Yukawa potential. For different problems, different choices of potential functions may be relevant. Additionally, we have included both ions and electrons in our list of particle species. We will now make simplifying assumptions regarding the potential and electron dynamics. Because the electron mass is much less than the ion mass, for our purposes we will assume that the electrons follow the ions adiabatically, producing an average density of

$$n_e(\mathbf{r}) = n_e^0 + \Delta n_e(\mathbf{r}), \tag{3.18}$$

where n_e^0 is the global average electron density and $\Delta n_e(\mathbf{r})$ are linear screening electron clouds that follow ions throughout the domain. The effects of these electron clouds are captured in MD by employing the Yukawa potential (3.3) between ions.

Due to the special nature of electron dynamics and our assumption above (3.18), we will find it convenient to abuse notation such that sums over all species will henceforth refer only

to ion species, with electrons handled separately. Then we can show:

$$\nabla_{\mathbf{r}} \sum_l \left(\int U_{kl}(\mathbf{r}, \mathbf{r}') n_l(\mathbf{r}', t) \, \mathrm{d}\mathbf{r}' \right) = Z_k e \nabla_{\mathbf{r}} \phi(\mathbf{r}, t), \tag{3.19}$$

where the electrostatic potential ϕ is

$$\phi(\mathbf{r}, t) = \int \rho(\mathbf{r}', t) \left(\frac{e^{-|\mathbf{r}-\mathbf{r}'|/\lambda}}{|\mathbf{r} - \mathbf{r}'|} \right) \mathrm{d}\mathbf{r}', \tag{3.20}$$

and the total charge density ρ is given by

$$\rho(\mathbf{r}, t) = -e n_e^0 + \sum_l Z_l e n_l(\mathbf{r}, t). \tag{3.21}$$

Here, the contribution of the background electron density n_e^0 to ρ shifts the charge density by a constant amount.

For three dimensional Yukawa systems, this electric potential can also be computed by solving the screened Poisson equation:

$$-\frac{1}{4\pi} \nabla_{\mathbf{r}}^2 \phi + \frac{1}{\lambda^2} \phi = \rho(\mathbf{r}, t). \tag{3.22}$$

This arises from the fact that the Green's function solution for (3.22) can be expressed as (3.20). We can now write the full partial differential kinetic equation for f_k:

$$\frac{\partial f_k}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} f_k - \frac{Z_k e}{m_k} \nabla_{\mathbf{r}} \phi \cdot \nabla_{\mathbf{v}} f_k = \sum_l C'_{kl}. \tag{3.23}$$

This defines our system, save for the unknown C'_{kl} remainder terms. These are typically called the "collisional" terms since they describe particle-particle interactions beyond the mean field electric potential interaction.

Different choices for the functional form of C'_{kl}, and thus f_{kl}, yield different kinetic models, e.g. Boltzmann. For this paper we choose the Bhatnagar-Gross-Krook (BGK) model as our

collisional model because it only requires microscale data in the form of scalar relaxation times, rather than higher dimensional cross section data. To obtain the BGK model, we take

$$C'_{kl} \approx \frac{f^{eq}_{kl}(\mathbf{r}, \mathbf{v}, t) - f_k(\mathbf{r}, \mathbf{v}, t)}{\tau_{kl}}, \tag{3.24}$$

where f^{eq}_{kl} is chosen to locally conserve mass, momentum, and energy, and τ_{kl} is a relaxation parameter. Indices k and l denote species pairs, including self-collisions. The target distributions f^{eq}_{kl} are Maxwellians with bulk velocity $\mathbf{u}_{kl}$, and temperature T_{kl}.

$$f^{eq}_{kl} = n_k \exp\left(-\frac{m_k|\mathbf{v} - \mathbf{u}_{kl}|^2}{2T_{kl}}\right). \tag{3.25}$$

The mixture quantities $\mathbf{u}_{kl}$ and T_{kl} depend on the collision rates τ_{kl}, τ_{lk} and are defined such that the collisional model conserves mass, momentum, and energy and satisfies Boltzmann's H-Theorem; see [45] for more detail.

To summarize, we have derived the Vlasov-BGK (VBGK) equation:

$$\begin{aligned} &\frac{\partial f_k}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} f_k - \frac{Z_k e}{m_k} \nabla_{\mathbf{r}} \phi \cdot \nabla_{\mathbf{v}} f_k = \sum_l \frac{f^{eq}_{kl} - f_k}{\tau_{kl}} \\ &-\frac{1}{4\pi} \nabla^2_{\mathbf{r}} \phi + \frac{1}{\lambda^2} \phi = \left(\sum_k Z_k e n_k - e n^0_e\right) \end{aligned} \tag{3.26}$$

from molecular dynamics. The system is closed when the τ_{kl} are specified.

3.2.3 The multiscale model

The BGK approximation greatly simplifies the kinetic model by encoding collisional details into a single parameter per species pair, τ_{kl}. However, standard formulas for relaxation times for plasmas rely on a series of approximations that are only appropriate in a narrow range of parameter space [52, 76]. These are phenomenological quantities that can only be computed explicitly under specific conditions and assumptions, and many physically relevant scenarios that may be encountered during a simulation, such as moderate coupling and

distributions too far out of equilibrium, violate the assumptions used in the derivation of the standard formulas. Furthermore, because the relaxation times depend on a very high dimensional parameter space, namely, the parameter space of possible distributions, we cannot realistically predict these scenarios *a priori* by precomputing the relaxation times in a manner equivalent to generating an equation of state table.

Instead, we compute τ_{kl} *in situ* using embedded MD simulations (Figure 3.1) using the HMM framework.

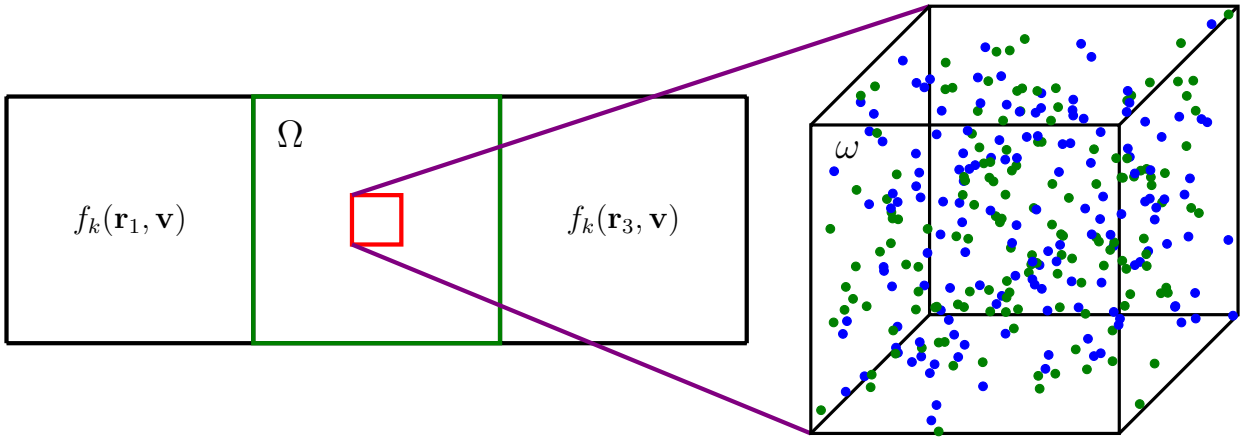

Figure 3.1: A sketch of an MD simulation on a domain ω embedded within a kinetic finite element Ω. The ions in ω are initialized with distributions $f_k(\mathbf{r}_2, \mathbf{v})$. The resultant trajectories inform the relaxation rates employed in Ω at future kinetic timesteps.

Initializing these MD simulations requires the definition of a suitable pseudoinverse of the projector, $P^\dagger$. We wish for the MD simulation to begin with particles distributed according to the current kinetic disctributions. A well known problem in initializing MD simulations is that if interparticle correlations are not consistent with the desired state, there will be an unphysical exchange between potential energy and kinetic energy. This initialization issue in MD is typically mitigated by "equilibrating" the system using non-Hamiltonian mechanics, which drives the system to a self-consistent $g(r)$ and precludes this nonphysical energy exchange [110]. A number of thermostats exist to initialize to a desired canonical distribu-

tion [18, 64], however we are concerned with possible nonequilibrium kinetic distributions as initial conditions, and a thermostat does not exist that generates an arbitrary nonequilibrium distribution initial state in MD.

We propose using a stochastic thermostat based on Langevin dynamics to match the density, momentum, and kinetic energy of the nonequilibrium kinetic distributions with their corresponding *equilibrium* particle distribution functions. This ensures that we have reasonable interparticle correlations that correspond to the first three moments of the distribution function. Then we then sample the distribution functions $f_k(\mathbf{v})$ and reassign each particle's velocity to match the potentially non-equilibrium kinetic state of the original distribution function. This is illustrated for a bump-on-tail example in Figure 3.2. It is important to

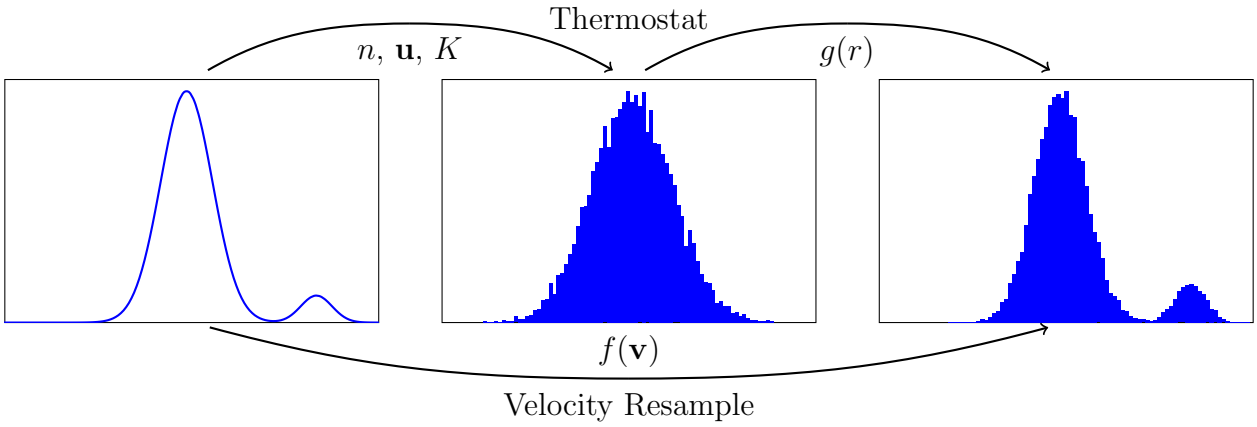

Figure 3.2: The process of initializing an MD simulation to a desired nonequilibrium distribution (left). First, the MD particles are driven by a Langevin thermostat to a Maxwellian whose first two moments match the desired distribution and has realistic interparticle correlations $g(r)$ (middle). Then, the particle velocities are resampled directly from the desired distribution (right). These steps are done simultaneously for all species in the cell.

note that by resampling the velocities, we lose the particle correlations that depend on both positions *and* velocities. This is a casualty of the multiscale method; there is not sufficient information to generate a macroscale and microscale that are perfectly consistent. Recall that our pseudoinverse projector $P^\dagger$ is not unique. However, we have not found that these

lost correlations have any appreciable effect on the evolution of the system.

There is no explicit formula for the relaxation rates τ_{kl}, and we must extract them from some statistical observable in the MD system. Because the relaxation rates describe the rate of change of the distribution functions due to the kinetic system being out of equilibrium, we compute τ_{kl} by ensuring that the rate of change of a quantity of interest matches between the kinetic and MD systems.

One such relevant quantity in kinetic theory is the Boltzmann H function. The H_k function for each species k is defined as:

$$H_k(\mathbf{r}, t) = \int f_k(\mathbf{r}, \mathbf{v}, t) \log(f_k(\mathbf{r}, \mathbf{v}, t)) \, \mathrm{d}\mathbf{v}. \tag{3.27}$$

Integrating the H_k function for each species over the domain and summing over species yields the negative entropy of the system. It can be proven that this total H is a decreasing function of time, and constant only when the system is in global equilibrium.

Although the total H decreases monotonically, H_k for each individual species may increase due to interactions with other species. We compute τ_{kl} by matching the MD and kinetic interpretations of the rate of change of H_k for each species at each position, due specifically to collisional interactions, including self-collisions. This guarantees that we compute τ_{kl} by forcing a fundamental, collision-dependent physical quantity to change in the kinetic model in accordance with the more detailed molecular model. For a detailed discussion of H_k in this problem, see Section 3.2.4. The most pertinent details can be found in the remainder of this subsection.

The rate of change of H_k according to the kinetic model when disregarding advection is

$$\frac{dH_k^{KT}}{dt} = \sum_l \frac{1}{\tau_{kl}(\mathbf{r})} \int (f_{kl}^{eq} - f_k) \log(f_k) \, \mathrm{d}\mathbf{v}. \tag{3.28}$$

Note that only the collisional terms in the Vlasov-BGK equation contribute towards this rate. This allows us to isolate the terms involving τ_{kl}. The equivalent rate of change in a

molecular model is

$$\frac{dH_k^{MD}}{dt} = \sum_l \left\langle \sum_{\substack{i \in S_k \\ j \in S_l}} \frac{\mathbf{F}_{ij}}{m_k} \cdot \nabla_{\mathbf{v}} f_k|_{\mathbf{v}=\mathbf{v}_i} \right\rangle, \tag{3.29}$$

where S_k is the set of indices corresponding to ions of species k, and $\mathbf{F}_{ij}$ is the force exerted on particle i by particle j. The MD data are computed as a time-averaged quantity, assuming ergodicity and that the kinetic variables do not change significantly over the course of a simulation. This last requirement limits the amount of time over which we can average the data. The resulting data is subject to statistical noise. This can be overcome by using a larger number of particles or aggregating the data from several MD simulations run in parallel. Because the MD simulation is over a much smaller volume than the kinetic finite volume cell, the quantities gathered in MD must be scaled up by the volume differential.

Note that, because both (3.28) and (3.29) are expressed as sums over all species, we can equate the two expressions term-wise to specifically match the entropy change of any species k due to interactions with any other species l. Equating the appropriate terms for species k and species l yields a system of two nonlinear equations specifying τ_{kl} and τ_{lk}, because f_{kl}^{eq} and f_{lk}^{eq} both depend nonlinearly on these rates. When both species are near equilibrium, the resulting system is linearly dependent. To account for statistical noise in the MD computation of dH_k^{MD}/dt, we attempted to solve this system in the least-squares sense using a nonlinear optimization method. However, we found that this led to a ill-posed optimization problem. We therefore choose the constraint of $\tau_{12}n_1 = \tau_{21}n_2$ to compute a unique solution for the cross-species relaxation terms. This constraint is suggested by the derivation of the BGK collision model from Boltzmann; it also arises as a consequence of deriving a BGK model that matches energy exchange rates with the Boltzmann model [45].

3.2.4 Derivation of the matching condition

We select τ_{kl} by matching the rates of change of the H_k in both MD and BGK. Conveniently, this choice of matching criteria isolates the effects of the collisional terms and decomposes into species-species interactions naturally in both physical models.

dH_k/dt in Molecular Dynamics

Consider first an MD simulation cell that is embedded within the finite volume at $\mathbf{r}$:

$$H_k^{MD} = \iint \langle N_k \rangle \log(\langle N_k \rangle) \, \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{v}, \tag{3.30}$$

where the $\mathbf{r}$ integral is over the MD cell. We assume the system is spatially homogeneous over the cell, so here N_k corresponds only to the velocity Klimontovich distribution $N_k = \sum_{i \in S_k} \delta(\mathbf{v} - \mathbf{v}_i)$, and the resulting $\mathbf{v}$ integral is multiplied by the volume V_{MD} of the MD cell. We can use the definition of the delta function to evaluate this integral:

$$H_k^{MD} = \left\langle V_{MD} \sum_{i \in S_k} \log(\langle N_k(\mathbf{v}_i) \rangle) \right\rangle. \tag{3.31}$$

Summed over all species, this should correspond to the negative entropy of the cell. This is not a typical expression for the negative entropy, so it is worthwhile to explore it. Observe that a particle contributes negatively to H_K^{MD} if it, on average, has a velocity that is unlikely. That is, the more particles found consistently in the tails of the distribution, the larger the entropy.

In order to take the time derivative of H_k^{MD}, we return to the integral formulation so that we can again take advantage of the delta function feature:

$$\begin{aligned} \frac{dH_k^{MD}}{dt} &= V_{MD} \left\langle \int \frac{\partial N_k}{\partial t} (\log(\langle N_k \rangle) + 1) \, \mathrm{d}\mathbf{v} \right\rangle \\ &= V_{MD} \left\langle \sum_{i \in S_k} \int \frac{\mathrm{d}\mathbf{v}_i}{dt} \delta(\mathbf{v} - \mathbf{v}_i) \cdot \nabla_{\mathbf{v}} [\log(\langle N_k \rangle)] \, \mathrm{d}\mathbf{v} \right\rangle \\ &= V_{MD} \left\langle \sum_{i \in S_k} \frac{\mathbf{F}_i}{m_k} \cdot \nabla_{\mathbf{v}} [\log(\langle N_k \rangle)]|_{\mathbf{v}=\mathbf{v}_i} \right\rangle \end{aligned} \tag{3.32}$$

where $\mathbf{F}_i$ is the force on particle i due to its interactions with every other particle in the cell. In reality, the entropy production in the cell depends also upon the force interactions

between ions inside the cell and those outside it. However, because we consider only the collisional aspects of entropy production, which are local, we disregard these interactions.

The force exerted on all particles of species k can be decomposed by species using

$$\sum_{i \in S_k} \mathbf{F}_i = \sum_l \sum_{\substack{i \in S_k \\ j \in S_l}} \mathbf{F}_{ij}$$

where $\mathbf{F}_{ij}$ is the force exerted on particle i by particle j. This species-wise decomposition allows us to specify interspecies relaxation times τ_{kl} individually.

Finally, we note that we initialize the MD simulation with $\langle N_k \rangle = f_k$, and assume that kinetic distribution functions do not change significantly during the MD simulation. Thus, we can replace $\langle N_k \rangle$ with f_k in this calculation as the distribution should not have moved significantly from its initial state. Thus, the rate of change of H_k in MD is:

$$\begin{aligned} \frac{dH_k^{MD}}{dt} &= V_{MD} \sum_l \frac{dH_{kl}^{MD}}{dt}, \\ \frac{dH_{kl}^{MD}}{dt} &= \left[\sum_{\substack{i \in S_k \\ j \in S_l}} \frac{\mathbf{F}_{ij}}{m_k} \cdot \sum_{i \in S_k} \nabla_{\mathbf{v}} [\log(f_k)]|_{\mathbf{v}=\mathbf{v}_i} \right] \end{aligned} \tag{3.33}$$

We record the velocities of and forces on each particle in order to calculate this quantity at each time step. The quantity is statistically noisy, because it is the derivative of a function that is not necessarily smooth. We attenuate the noise by taking a time average, assuming the system is ergodic and that dH_k^{MD}/dt does not change significantly during the MD simulation.

It is worth pausing again and considering this term more closely. H_k is decreasing, and entropy increasing whenever, on average, the force on a particle tends to point opposite the direction of increasing probability. That is, when a particle is being accelerated towards an unlikely velocity, the entropy increases. This makes intuitive sense. If f_k is a Maxwellian of

the form

$$f_k = A_k \mathrm{Exp}\left\{-\frac{m_k|\mathbf{v}-\mathbf{u}_k|^2}{2T_k}\right\},$$

then the rate of change of the entropy is:

$$\frac{dH_k^{MD}}{dt} = -\frac{V_{MD}}{T_k}\sum_l \left\langle \sum_{\substack{i\in S_k \\ j\in S_l}} \mathbf{F}_{ij}\cdot(\mathbf{v}_i-\mathbf{u}_k)\right\rangle. \tag{3.34}$$

We recover the classical result that the rate of change of the entropy is proportional to the inverse of the temperature.

dH_k/dt in Kinetic Theory

In kinetic theory, the amount of H_k inside a finite volume cell is defined as

$$H_k^{KT} = \iint f_k \log(f_k)\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{v}. \tag{3.35}$$

Here, the spatial integral is over the kinetic finite volume cell. f_k is considered uniform over each cell. The rate of change of this quantity according the the Vlasov-BGK equation can be found by taking the time derivative of each side and substituting in the expression for $\partial f_k/\partial t$:

$$\begin{aligned}\frac{dH_k^{KT}}{dt} =& -\iint \mathbf{v}\cdot\nabla_{\mathbf{r}}[f_k][\log(f_k)+1]\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{v} + \iint \frac{Z_k e}{m_k}\nabla_{\mathbf{r}}[\phi]\cdot\nabla_{\mathbf{v}}[f_k][\log(f_k)+1]\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{v} \\ &+ \sum_l \iint \frac{f_{kl}^{eq}-f_k}{\tau_{kl}}[\log(f_k)+1]\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{v}.\end{aligned} \tag{3.36}$$

The second integral vanishes by the divergence theorem because $\nabla_{\mathbf{v}}[f_k][\log(f_k)+1] = \nabla_{\mathbf{v}}[f_k\log(f_k)]$. The first integral is the transport of H_k by advection. This is handled well by the Vlasov portion of the Vlasov-BGK equation. Additionally, transport happens on a time and spatial scale far exceeding an MD simulation. Because no advection from neighboring cells is included in MD, it does not have an equivalent term, and equation (3.33) corresponds

to the collisional entropy in MD. Thus, we consider matching only the "collisional entropy" in BGK (the third integral) with the collisional entropy observed in MD.

Let the non-advective rate of change of entropy be denoted $\left(\frac{dH_k^{KT}}{dt}\right)_{\text{coll}}$. Mass conservation causes $\sum_l \frac{1}{\tau_{kl}} \iint f_{kl}^{eq} - f_k \, \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{v}$ to vanish. We can compute the spatial integral of the remaining portion of the integrand assuming uniformity, letting the volume of the finite volume be V_{KT}. What remains is:

$$\begin{aligned}\left(\frac{dH_k^{KT}}{dt}\right)_{\text{coll}} =& V_{KT} \sum_l \left(\frac{dH_{kl}^{KT}}{dt}\right)_{\text{coll}}, \\ \left(\frac{dH_{kl}^{KT}}{dt}\right)_{\text{coll}} =& \frac{1}{\tau_{kl}} \int (f_{kl}^{eq} - f_k) \log(f_k) \, \mathrm{d}\mathbf{v}.\end{aligned} \tag{3.37}$$

We seek to choose τ_{kl} and τ_{lk} such that equations (3.33) and (3.37) are the same.

Solving for the Relaxation Rates

We will guarantee the matching not just of the overall rate of H_k, but of the contributions to that rate due to each species l. That is, our matching condition is:

$$V_{MD} \frac{dH_{kl}^{MD}}{dt} = V_{KT} \left(\frac{dH_{kl}^{KT}}{dt}\right)_{\text{coll}} \tag{3.38}$$

for all k and l.

In the case of a single-species system, or computing the intraspecies relaxation rate of a multispecies system, we have one equation to satisfy. The equilibrium distribution f_{kk}^{eq} does not depend on τ_{kk}, so the matching condition is simply:

$$\tau_{kk} = \frac{V_{KT} \int (f_{kk}^{eq} - f_k) \log(f_k) \, \mathrm{d}\mathbf{v}}{V_{MD} \frac{dH_{kk}^{MD}}{dt}}. \tag{3.39}$$

For any two distinct species, two of our matching conditions are coupled due to the

dependence of f_{kl}^{eq} and f_{lk}^{eq} on τ_{kl} and τ_{lk}. The necessary matching conditions are:

$$\begin{aligned} V_{MD}\frac{dH_{kl}^{MD}}{dt} =& \frac{V_{KT}}{\tau_{kl}}\int (f_{kl}^{eq}(\tau_{kl},\tau_{lk}) - f_k)\log(f_k)\,\mathrm{d}\mathbf{v} \\ V_{MD}\frac{dH_{lk}^{MD}}{dt} =& \frac{V_{KT}}{\tau_{lk}}\int (f_{lk}^{eq}(\tau_{kl},\tau_{lk}) - f_l)\log(f_l)\,\mathrm{d}\mathbf{v}. \end{aligned} \tag{3.40}$$

Though it appears that we have two equations in two unknowns, these equations are linearly dependent when f_k and f_l are Maxwellian with the same bulk velocity (and the system is nearly singular when they are merely near equilibrium). This is the initial condition of a temperature relaxation problem, and is also a scenario one may encounter during simulations that we would like to handle. Additionally, the data from MD are often noisy. We handle both of these issues simultaneously by enforcing another constraint $\tau_{kl}n_k = \tau_{lk}n_l$. With this constraint, the cross-term matching conditions can also be solved explicitly.

The HMM interface

The HMM model is evolved by alternating between the kinetic model and MD. It remains to specify when the kinetic model interrogates the MD to get new collision rates. We choose to measure the ratio of the current and initial differences between the distributions $f_k(\mathbf{r},\mathbf{v})$ and their local equilibria $f_k^{eq}(\mathbf{r},\mathbf{v})$:

$$E_k(\mathbf{r}) = \frac{\|f_k^{eq}(\mathbf{r},t) - f_k(\mathbf{r},t)\|_2}{\|f_k^{eq}(\mathbf{r},t_0) - f_k(\mathbf{r},t_0)\|_2}, \tag{3.41}$$

where t_0 is the time the most recent kinetic phase began. When this ratio drops below some threshold, the kinetic model pauses and requests new collision times from the MD based on the current distribution functions f_k. The reference time t_0 is reset to the current time and the kinetic calculation continues with the new collision times.

To further reduce expensive MD calls, as well as to reduce the effects of the noise coming from the MD data, we use each MD data point to build a linear regression model for each of the collision times τ_{kl}. When the kinetic code requests new collision times, the linear

regression model is evaluated, and if the new collision rates suggested by the model are within $E_\tau\%$ of the previous rates, where E_τ is specified by the user, they are presented to the kinetic model without running MD. In order to ensure that the MD data is relatively fresh, we maintain a moving window of the previous N_{MD} simulations to inform the regression model.

Algorithm

The algorithm described in this chapter can be summarized as follows:

1. Given the initial macroscopic state in each finite volume cell, run a small MD simulation to compute initial values for each $\tau_{kl}(\mathbf{r})$.

2. Evolve the kinetic equation using the computed τ_{kl} in each cell. Calculate the ratio (3.41) to determine if a distribution has significantly changed.

3. Evaluate the linear regression model to determine new values for τ_{kl}

   (a) If the τ_{kl} suggested by the linear regression model is within $E_\tau\%$ of the previous value, provide this to the kinetic model.

   (b) If not, initialize a MD simulation to compute new values for τ_{kl} in those cells and update the regression model.

4. Repeat steps 2 and 3 until the end time of the simulation is reached.

The resulting simulation, while bottlenecked by MD simulations, is able to probe much larger systems for a much longer time than a full MD simulation. Furthermore, the relaxation times used in the Vlasov-BGK equation will cause the kinetic system to accurately track the entropy changes implied by the molecular description. This results in a model far more accurate and flexible than any current Vlasov-BGK models.

3.2.5 Numerical results

In this section, we present a relatively simple 0D-3V multispecies kinetic problem (zero dimension in physical space, three dimensions in velocity space) in order to explore the choices made in making this model. Future work will extend this model to more realistic 1D-3V case to model problems such as interface evolution in inertial confinement fusion.

Molecular dynamics implementation details

Given an initial configuration $\left(\{\mathbf{r}_i(0)\}_{i=1}^N, \{\mathbf{v}_i(0)\}_{i=1}^N\right)$, an MD simulation evolves according to 3.5, which can be reexpressed as:

$$\begin{aligned}
\frac{d\mathbf{r}_i}{dt} =& \frac{1}{m_i}\nabla_{\mathbf{v}_i}[H] = \mathbf{v}_i \\
\frac{d\mathbf{v}_i}{dt} =& -\frac{1}{m_i}\nabla_{\mathbf{r}_i}[H] = \frac{1}{m_i}\mathbf{F}_i(\mathbf{r}_1, \ldots, \mathbf{r}_N) \\
\mathbf{F}_i =& -\sum_{j\neq i} \nabla_{\mathbf{r}_i}[U_{ij}(\mathbf{r}_i, \mathbf{r}_j)],
\end{aligned} \tag{3.42}$$

where $\mathbf{F}_i$ is the force exerted on ion i due to all other ions. To compute the screening length, we assume a density-weighted mixture temperature and use the Debye length as the characteristic length:

$$\lambda = \lambda_D = \sqrt{\frac{T_e}{4\pi n_e}} \tag{3.43}$$

$$n_e = \sum_k Z_k n_k \tag{3.44}$$

$$T_e = \frac{\sum_k n_k T_k}{\sum_k n_k}. \tag{3.45}$$

We exploit the symmetries inherent in the Hamiltonian formulation by using the symplectic velocity Verlet integrator to evolve the MD system from the initial conditions, which guarantees conservation of energy when used with a stable time step. While computing all the interparticle potentials is an expensive $O(N^2)$ operation, we can exploit the exponential

decay of the Yukawa potential by using nearest neighbor linked lists and a cutoff radius to accelerate the simulation, yielding nominally $O(N)$ performance. The force computation and particle evolution is parallelized using OpenMP. To accelerate the MD simulations we selectively use a reversible multiple time step method [109] on close-together particles when we detect that energy is not conserved.

BGK implementation details

The 0D-3V Vlasov BGK equation is evolved with a second order SSP Runge-Kutta method [42]. For each species, we make a uniform discretization of velocity space with the maximum velocity chosen to be a multiple of the maximum expected thermal velocity in the simulation.

Temperature relaxation

We apply this model to a two-species, zero dimensional temperature relaxation problem. The system is comprised of aluminum and hydrogen ions with Maxwell-distributed initial conditions at equal density with zero bulk velocity. The hydrogen ions begin at $T_H = 1000$ eV while the aluminum ions begin at $T_{Al} = 500$ eV. The MD simulation employs periodic boundary conditions, simulating bulk phase. Over time, the system should relax to a thermal equilibrium with $T_{eq} = 750$.

We implemented several versions of our algorithm, and the results are depicted in Figure 3.3. First, in the solution labeled "HMM, with regression", we adjust the relaxation rates for species k whenever E_k (3.41) changes by 10%. If a linear regression of the five previously computed empirical relaxation rates implies new relaxation rates that differ by less than 10% of the previous value, we use that as our new rate. If the results implied by the regression change the rate by more than 10%, we instead spawn three MD simulations and average the results over those simulations to mitigate noise. We also use MD simulations if the extrapolation time exceeds the width of the window defined by the five rates upon which the regression is currently based. In the solution labeled "HMM, no regression", we adjust the relaxation rates for species k whenever E_k (3.41) changes by 10%. No regression is employed,

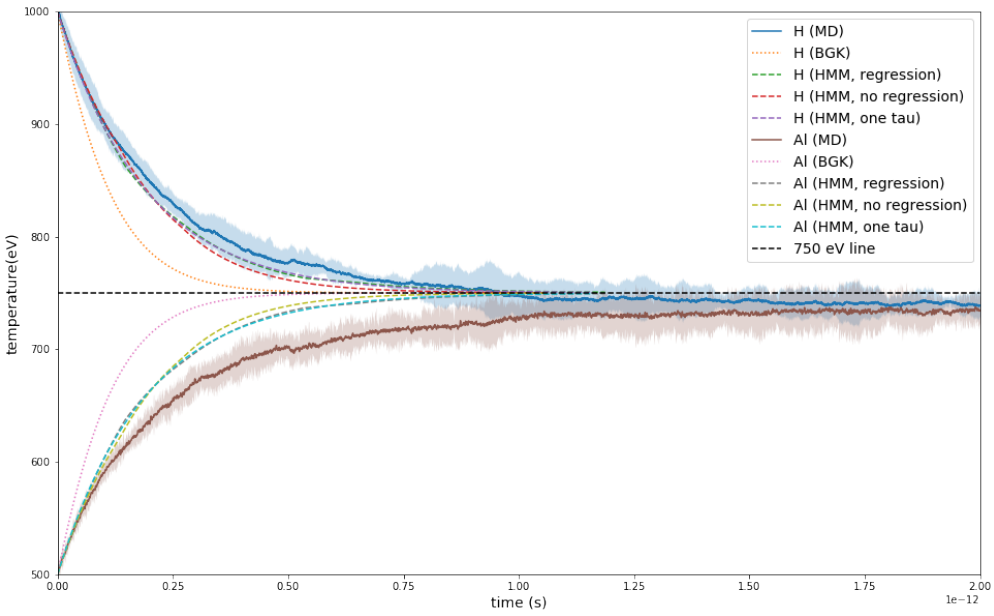

Figure 3.3: A comparison of several solution methods for a temperature relaxation problem. Equal densities of hydrogen ions (H) at 1000 eV and aluminum ions (Al) at 500 eV are allowed to reach equilibrium. The exact MD solution is averaged from five simulations and depicted with 95% confidence intervals. The "BGK" curves depict simulations using theoretical relaxation rates. The "HMM, regression" curves employ our described HMM method with linear regression used for relaxation rates that have not changed very dramatically. The "HMM, no regression" describe the same method, but with no linear regression employed to minimize calls to MD. The "HMM, one tau" uses five MD simulations to find suitable initial relaxation rates and then uses that for the duration of the simulation. Note that the "HMM, regression" and "HMM, one tau" perform the best over the course of the simulation.

and we use three MD simulations to recompute the relaxation rates whenever this threshold is crossed. In the solution labeled "HMM, one tau", we compute the relaxation rates only at the beginning using five MD simulations over which to average. We compare our results against the "exact" results computed from an average of five MD simulations (with 95% confidence intervals) and with a standard BGK simulation using theoretical values for τ_{kl} specific to temperature relaxation problems [45].

Inspecting Figure 3.3, we observe several things. First, the BGK simulation with theoretical relaxation rates performs quite poorly. Each of our HMM algorithms represents an

improvement upon the naive BGK results. Intriguingly, the single tau simulation and the regression simulation perform almost identically, and both are improvements upon the HMM without regression. The single tau simulation is computationally more efficient than the updating schemes, and so is the obvious choice here. We believe this is a consequence of the fact that this is a temperature relaxation problem and a single relaxation fixed rate dictates the dynamics. In a more complicated system, the relaxation rates would be expected to vary over the course of the simulation. It is also interesting to note that the MD simulations do not converge to 750 eV, but to a temperature slightly below that. This is because some of the energy in the system is converted to potential energy over the course of the simulation.

3.2.6 Derivation of kinetic moments from molecular dynamics

MD is taken as the ground truth from which the kinetic equations are derived as expected values. Similarly, we will here define the kinetic moments in MD, and define their KT interpretations as the expected values of those expressions. This is equivalent to defining microscopic functions $f(\mathbf{u})$ such that $Pf(\mathbf{u}) = \hat{f}(\hat{\mathbf{u}})$. These connections will be necessary for identifying how the macroscale state should constrain microscale simulations. In many cases, the result is the definition typically used in kinetic theory. However, in a few cases there are some subtle differences that must be taken into account when coupling the two theories. Consider the full system described in MD. Let Ω represent a region of the simulation corresponding to a particular BGK finite volume cell, and suppose units are chosen so that the volume of that cell is one.

The density of species k inside Ω according to MD is

$$n_k^\Omega = \sum_{\substack{i \in S_k \\ \mathbf{r}_i \in \Omega}} 1. \tag{3.46}$$

Taking the expected value, we find

$$\begin{aligned} n_k^{\Omega,KT} \equiv \langle n_k^{\Omega} \rangle = \left\langle \sum_{\substack{i \in S_k \\ \mathbf{r}_i \in \Omega}} 1 \right\rangle &= \iint \left\langle \sum_{\substack{i \in S_k \\ \mathbf{r}_i \in \Omega}} \delta(\mathbf{r} - \mathbf{r}_i)\delta(\mathbf{v} - \mathbf{v}_i) \right\rangle \mathrm{d}\mathbf{v}\mathrm{d}\mathbf{r} \\ &= \int_\Omega \int \left\langle \sum_{i \in S_k} \delta(\mathbf{r} - \mathbf{r}_i)\delta(\mathbf{v} - \mathbf{v}_i) \right\rangle \mathrm{d}\mathbf{v}\mathrm{d}\mathbf{r} \\ &= \int_\Omega \int f_k \, \mathrm{d}\mathbf{v}\mathrm{d}\mathbf{r} = \int f_k \, \mathrm{d}\mathbf{v}. \end{aligned}$$

In the last step we used the fact that the distribution is assumed to be uniform over the BGK cell, and that the cell volume is one. The result is the density of species k in terms of kinetic theory variables that is consistent with the MD interpretation.

The momentum of species k inside Ω according to MD is:

$$(n_k \mathbf{u}_k)^{\Omega} = \sum_{\substack{i \in S_k \\ \mathbf{r}_i \in \Omega}} m_k \mathbf{v}_i. \tag{3.47}$$

Taking the expected value, and using delta functions and intra-cell uniformity as above, we find

$$(n_k \mathbf{u}_k)^{\Omega,KT} \equiv \langle (n_k \mathbf{u}_k)^{\Omega} \rangle = m_k \int \mathbf{v} f_k \, \mathrm{d}\mathbf{v}.$$

The kinetic energy of species k inside Ω according to MD is

$$K_k^{\Omega} = \sum_{\substack{i \in S_k \\ \mathbf{r}_i \in \Omega}} \frac{m_k |\mathbf{v}_i|^2}{2}. \tag{3.48}$$

The BGK interpretation is found by taking the expected value:

$$K_k^{KT} \equiv \langle K_k^{\Omega} \rangle = \frac{m_k}{2} \int |\mathbf{v}|^2 f_k \, \mathrm{d}\mathbf{v}.$$

The potential energy in the full system is:

$$PE = \frac{1}{2}\sum_{k,l}\sum_{\substack{i\in S_k\\ j\in S_l}} U_{kl}(\mathbf{r}_i,\mathbf{r}_j). \tag{3.49}$$

Taking the expected value, we find:

$$PE^{KT} \equiv \langle PE\rangle = \sum_{k,l}\frac{1}{2}\iiiint U_{kl}(\mathbf{r},\mathbf{r}')f_{kl}\,\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{v}'\mathrm{d}\mathbf{v}\mathrm{d}\mathbf{r}.$$

This is different from the standard definition of potential energy in kinetic theory. That definition arises only upon making a mean field approximation:

$$PE^{BGK} \equiv \sum_{k,l}\frac{1}{2}\iint U_{kl}(\mathbf{r},\mathbf{r}')n_k(\mathbf{r})n_l(\mathbf{r}')\,\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{r}.$$

The residual between PE^{BGK} and PE^{KT} is the contribution of the collisional term to the potential energy. We call this PE^{coll}, and it has the form:

$$PE^{\text{coll}} = \sum_{k,l}\frac{1}{2}\iiiint U_{kl}(\mathbf{r},\mathbf{r}')C_{kl}\,\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{v}\mathrm{d}\mathbf{v}'\mathrm{d}\mathbf{r}$$

where $C_{kl} = f_{kl} - f_k f_l$ is the remainder of the mean field approximation. This term is not necessarily zero. What this means is that the standard definition of potential energy in KT differs from that implied by MD. There will be an error incurred in the potential energy, and thus total energy, defined in each scale. This error cannot be avoided, and so we cannot force the total energy to match between scales when initializing an MD simulation.

3.2.7 Residual equation

We can fully express the errors incurred by using the BGK approximation. Let f_k^B be the solution to the Vlasov-BGK equation with optimal choices for τ_{kl}. We define the residual as $f_k^r = f_k - f_k^B$. Starting with the exact initial condition (zero error), the residual will develop

as we evolve our HMM system according to:

$$\begin{aligned}\frac{\partial f_k^r}{\partial t} =& - \mathbf{v} \cdot \nabla_{\mathbf{r}}[f_k - f_k^B] - \frac{q_k}{m_k} \nabla_{\mathbf{r}}[\phi] \cdot \nabla_{\mathbf{v}}[f_k - f_k^B] \\ &+ \frac{1}{m_k} \sum_l \iint \nabla_{\mathbf{r}}[U_{kl}] \cdot \mathbf{v}[f_{kl} - f_k^B f_l^B] \, \mathrm{d}\mathbf{r}' \mathrm{d}\mathbf{v}' + \sum_l \frac{f_{kl}^{eq} - f_k^B}{\tau_{kl}}.\end{aligned}$$

At the initial time, the only error is from the third and fourth terms, but this error will cause the first two terms to become nonzero as time passes. If the third and fourth terms approximately cancel one another, the rate of error growth will be small, and the approximation will be valid for a significant amount of time. This is true whenever the BGK operator effectively accounts for the collisional effects of the system. An additional error is incurred in the HMM formulation: the τ_{kl} are not optimal throughout each timestep, nor perhaps even after updating (due to statistical noise). The resulting algorithm will perform necessarily worse than one with a theoretical optimal τ_{kl}.

We can also see how the different choice of potential energy for BGK introduces errors on top of the errors in distribution:

$$PE^r = \frac{1}{2} \sum_{k,l} \iiiint U_{kl}[f_{kl} - f_k^B f_l^B] \, \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{v}\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{v}'.$$

We can count on the total momentum and mass being conserved consistently, because even with our different evolution equations, they are conserved in both the exact and BGK models. The total energy we begin with in BGK will be conserved, but this will differ from the total energy in the real system. The exchange of energy between kinetic and potential energy will also be incorrect. Thus, neither the kinetic energy nor the potential energy in BGK will evolve in a way fully consistent with MD. Because the collisional term does not directly affect the kinetic or potential energy, and is the only term modified by updating τ_{kl}, BGK will conserve the wrong initial total energy during the whole simulation.

The error equations suggest the possibility of choosing the optimal τ_{kl} by minimizing

the difference between the third and fourth terms in the error equation. Unfortunately, this requires knowledge of f_{kl}, which we cannot easily gather from statistical data in an MD simulation. This is the topic of future work on this model.

3.2.8 Explicit Mori-Zwanzig interpretation

We can recast this specific HMM implementation in terms of the Mori-Zwanzig formalism. First, we specify that the variables in the full system (both resolved and unresolved) are comprised of the positions and velocities of each particle, $\mathbf{r}_i$ and $\mathbf{v}_i$. This is the microscale in HMM terminology. The operator $\mathcal{L}$, then, obeys the product rule and chain rule until it is applied directly to $\mathbf{r}_i$ or $\mathbf{v}_i$, in which case:

$$\mathcal{L}\mathbf{r}_i = \mathbf{v}_i \tag{3.50}$$

$$\mathcal{L}\mathbf{v}_i = -\frac{1}{m_i}\nabla_{\mathbf{r}_i}\left[\sum_l \sum_{\substack{j\in S_l \\ j\neq i}} U_{kl}(\mathbf{r}_i, \mathbf{r}_j)\right]. \tag{3.51}$$

Next, we identify the resolved variables. These are the state variables of the macroscopic system. That is, they are the distribution functions $f_k(\mathbf{r}, \mathbf{v}, t)$ in each finite volume cell. We assume they are discretized in velocity as well so that we have a large system of ODEs for the evolution of the f_k functions at each point in phase space. The microscopic interpretations of these variables are the Klimontovich distributions:

$$N_k(\mathbf{r}, \mathbf{v}, t) = \sum_{k\in S_k} \delta(\mathbf{r} - \mathbf{r}_i(t))\delta(\mathbf{v} - \mathbf{v}_i(t)).$$

We now must define a projection P that takes any function of the microscopic variables and expresses it as a function of only the macroscopic variables. Taking inspiration from the manner in which the Vlasov equation was derived from the microscopic equations, we let $Pg = \langle g \rangle$ where the expected value is taken over the ensemble of initial microstates consistent with the initial macrostate variables. This definition of P may yield functions

that depend upon higher order distribution functions from the BBGKY hierarchy, such as the two particle distribution function

$$f_{kl}(\mathbf{r},\mathbf{v},\mathbf{r}',\mathbf{v}',t) = \langle N_{kl}\rangle = \left\langle \sum_{i\in S_k, j\in S_l} \delta(\mathbf{r}-\mathbf{r}_i(t))\delta(\mathbf{v}-\mathbf{v}_i(t))\delta(\mathbf{r}'-\mathbf{r}_j(t))\delta(\mathbf{v}'-\mathbf{v}_j(t)) \right\rangle.$$

In order for our projection to project onto the space of functions that depend only on the one particle distribution functions, and not higher order ones, we introduce the following additional rule: $P(N_{k_1,\ldots,k_n}) = \prod_i f_{k_i}$. That is, the n-particle distribution function is approximated as the product of the n one-particle distribution functions. We now apply these definitions to the microscopic interpretations, N_k, of the macroscopic variables, f_k, in order to see the Markov term:

$$\begin{aligned}
\mathcal{L}N_k &= \sum_{i\in S_k} \mathcal{L}\left[\delta(\mathbf{r}-\mathbf{r}_i)\delta(\mathbf{v}-\mathbf{v}_i)\right] \\
&= \sum_{i\in S_k} \left\{\delta(\mathbf{v}-\mathbf{v}_i)\mathcal{L}\delta(\mathbf{r}-\mathbf{r}_i) + \delta(\mathbf{r}-\mathbf{r}_i)\mathcal{L}\delta(\mathbf{v}-\mathbf{v}_i)\right\} \\
&= \sum_{i\in S_k} \left\{-(\mathcal{L}\mathbf{r}_i)\delta(\mathbf{v}-\mathbf{v}_i)\cdot\nabla_{\mathbf{r}}[\delta(\mathbf{r}-\mathbf{r}_i)] - (\mathcal{L}\mathbf{v}_i)\delta(\mathbf{r}-\mathbf{r}_i)\cdot\nabla_{\mathbf{v}}[\delta(\mathbf{v}-\mathbf{v}_i)]\right\} \\
&= \sum_{i\in S_k} \left\{ -\mathbf{v}_i\delta(\mathbf{v}-\mathbf{v}_i)\cdot\nabla_{\mathbf{r}}[\delta(\mathbf{r}-\mathbf{r}_i)] + \frac{1}{m_k}\delta(\mathbf{r}-\mathbf{r}_i)\nabla_{\mathbf{r}_i}\left[\sum_l \sum_{\substack{j\in S_l \\ j\neq i}} U_{kl}(\mathbf{r}_i,\mathbf{r}_j)\right]\cdot\nabla_{\mathbf{v}}[\delta(\mathbf{v}-\mathbf{v}_i)] \right\} \\
&= -\mathbf{v}\cdot\nabla_{\mathbf{r}}[N_k] + \sum_{i\in S_k}\left\{ \sum_l \sum_{\substack{j\in S_l \\ j\neq i}} \frac{1}{m_k}\nabla_{\mathbf{r}}[U_{kl}(\mathbf{r}_i,\mathbf{r}_j)]\cdot\nabla_{\mathbf{v}}[\delta(\mathbf{r}-\mathbf{r}_i)\delta(\mathbf{v}-\mathbf{v}_i)] \right\}.
\end{aligned}$$

We now apply the projection operator, which acts only on the trajectories $\mathbf{r}_i$ and $\mathbf{v}_i$.

$$\begin{aligned}
P\mathcal{L}N_k =& -\mathbf{v}\cdot\nabla_{\mathbf{r}}[PN_k] + \sum_{i\in S_k}\left\{P\left(\sum_l\sum_{\substack{j\in S_l\\ j\neq i}}\frac{1}{m_k}\nabla_{\mathbf{r}}[U_{kl}(\mathbf{r}_i,\mathbf{r}_j)]\cdot\nabla_{\mathbf{v}}[\delta(\mathbf{r}-\mathbf{r}_i)\delta(\mathbf{v}-\mathbf{v}_i)]\right)\right\}\\
=& -\mathbf{v}\cdot\nabla_{\mathbf{r}}[f_k] + \frac{1}{m_k}\left(\sum_l\iint\nabla_{\mathbf{r}}[U_{kl}(\mathbf{r},\mathbf{r}')]\cdot\nabla_{\mathbf{v}}[PN_{kl}]\,\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{v}'\right)\\
=& -\mathbf{v}\cdot\nabla_{\mathbf{r}}[f_k] + \frac{1}{m_k}\left(\sum_l\iint\nabla_{\mathbf{r}}[U_{kl}(\mathbf{r},\mathbf{r}')]f_l\,\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{v}'\right)\cdot\nabla_{\mathbf{v}}[f_k]\\
=& -\mathbf{v}\cdot\nabla_{\mathbf{r}}[f_k] + \frac{Z_k e}{m_k}\nabla_{\mathbf{r}}[\phi]\cdot\nabla_{\mathbf{v}}[f_k].
\end{aligned}$$

This is the Vlasov equation. This proves that, under these definitions, the Markov term of the reduced order model is the Vlasov equation. This implies that the remainder, or corrections to the Vlasov model, correspond to the memory and noise terms. The HMM methodology uses knowledge of the forms of these remainders and the underlying microscale system to approximate these terms using small simulations constrained by the macroscale variables.

3.3 Summary

The heterogeneous multiscale method is an extremely successful multiscale simulation framework for computational physics. It is perfectly suited for systems with precisely two levels of important detail in which the coarser scale is the scale of interest. We have seen how it can be understood in terms of the Mori-Zwanzig formalism, and we have seen a novel specific plasma physics application. We have seen how the HMM strategy we derived in that case can be reframed in terms of Markov term and noise and memory terms.

There are natural extensions of the specific application described in this chapter, and we will revisit them in Chapter 5. The methods used here, such as using knowledge of the full system to inform coarse modeling methods and enforcing matching conditions, are common themes of multiscale modeling. We hope to continue to seek application areas which benefit

from the HMM methodology specifically. Problems that can be tackled by HMM have the following features:

1. the problem involves two (or at least a small number of) relevant scales of interest,

2. we understand how to effectively simulate the component scales at least theoretically if not practically,

3. the scales are explicitly separated in terms of timescale or spatial scale,

4. a matching condition can be employed to connect the scales.

With those features, one can use one's knowledge of the component scales to construct a custom multiscale model using the heterogeneous multiscale method. We plan to seek applications in industry and national laboratories that satisfy these features, so that we can continue to construct custom multiscale models and learn more about their construction and behavior.

Chapter 4

THE COMPLETE MEMORY APPROXIMATION

We now return our attention to the Mori-Zwanzig formalism. Unlike HMM, the Mori-Zwanzig formalism, as a multiscale method, does not assume an explicit separation of scales. It can, consequently, be used for multiscale problems in which there is a *continuum* of scales rather than some small discrete number each treated differently. Portions of the text in this chapter, specifically the content of Section 4.1 and 4.3, is drawn from [88], which has been submitted to the Journal of Multiscale Modeling and Simulation.

Recall the form of the projected dynamics in the Mori-Zwanzig formalism:

$$\frac{dPu_k}{dt} = Pe^{t\mathcal{L}}P\mathcal{L}u_k^0 + P\int_0^t e^{(t-s)\mathcal{L}}P\mathcal{L}e^{sQ\mathcal{L}}Q\mathcal{L}u_k^0\,\mathrm{d}s.$$

We will study how to develop reduced order models directly from this rigorous foundation. These will take the form of methods of approximating the memory integral in terms of only the resolved variables. Define the memory term as:

$$\mathcal{M}_k = P\int_0^t e^{(t-s)\mathcal{L}}P\mathcal{L}e^{sQ\mathcal{L}}Q\mathcal{L}u_k^0\,\mathrm{d}s. \tag{4.1}$$

4.1 Memory approximation methods

Several memory approximation methods have been developed and applied to a variety of problems. In this section, I will describe several such approximation methods and their assumptions. Finally, I will describe a novel memory approximation that avoids many of those assumptions.

4.1.1 The t-model

The simplest possible approximation of the memory integral is to assume the integrand is constant. In this case, the memory integral becomes:

$$\mathcal{M}_k \approx tPe^{t\mathcal{L}}P\mathcal{L}Q\mathcal{L}u_k^0. \tag{4.2}$$

We apply $P\mathcal{L}Q\mathcal{L}$ to the initial condition. The result is an expression that depends upon only the initial condition of the resolved modes (because the final operator applied is a projection operator). This is then evolved forward in time. Therefore, this term can be expressed in terms of only the trajectories of the resolved variables, as desired. The t-model has been used to successfully construct reduced order models for a variety of problems [12, 19, 24, 25, 46, 99, 101–105].

4.1.2 The BCH approximation

Though the t-model has been a useful tool for simulation, we are interested in constructing new reduced order models in a perturbative sense. One such method begins by rewriting the memory term by reversing our use of Dyson's formula

$$\begin{aligned}\mathcal{M}_k =& Pe^{t\mathcal{L}}Q\mathcal{L}u_k^0 - Pe^{tQ\mathcal{L}}Q\mathcal{L}u_k^0\\ =& Pe^{t\mathcal{L}}\left(Q\mathcal{L}u_k^0 - e^{-t\mathcal{L}}e^{tQ\mathcal{L}}QLu_k^0\right)\\ =& Pe^{t\mathcal{L}}\left(Q\mathcal{L}u_0^k - e^{C(t)}Q\mathcal{L}u_0^k\right)\end{aligned}$$

where $C(t) = -tP\mathcal{L} + [tP\mathcal{L}, tQ\mathcal{L}] + \ldots$ is the BCH series, which expands in powers of the commutator $[tP\mathcal{L}, tQ\mathcal{L}]$. In the event that $[tP\mathcal{L}, tQ\mathcal{L}]$ is small, which corresponds to cases in which the orthogonal and the projected dynamics approximately commute, $C(t)$ can be

approximated by $-tP\mathcal{L}$. The resulting approximation is:

$$\mathcal{M}_k \approx \sum_{j=1}^{\infty}(-1)^{j+1}\frac{t^j}{j!}Pe^{t\mathcal{L}}(P\mathcal{L})^jQ\mathcal{L}u_k^0. \tag{4.3}$$

We designate this method of approximating the memory term the "BCH approximation." Note that if we take $[tP\mathcal{L}, tQ\mathcal{L}] = 0$ exactly, we can derive this expression explicitly from the integral form of the memory by making use of the commutation of $P\mathcal{L}$ and $Q\mathcal{L}$ and expanding the exponential function as a series so we can compute the integral termwise. If $[tP\mathcal{L}, tQ\mathcal{L}]$ is merely small, this expansion represents the first order approximation in that power series.

We can find higher order BCH approximations of the memory term by truncating (4.3) at different values of j. The $j = 1$ term is the t-model. Higher order terms are always projected immediately before the evolution operator, meaning they represent terms that can be fully described in terms of the trajectories of resolved variables only. They can therefore be included in a reduced order model without modification, as by construction they do not depend explicitly on unresolved variables. Unfortunately, when the BCH approximation has been used to construct reduced order models beyond the t-model, the resulting terms have led to numerical instability. Some success has been found in *renormalizing* these terms, and we will return to that concept shortly. The BCH memory approximation was employed in [104, 105].

4.1.3 The complete memory approximation

The BCH approximation assumes that the commutator $[tP\mathcal{L}, tQ\mathcal{L}]$ is small. This may not remain true as simulations evolve. We sought to construct a new method for approximating the memory integral without relying upon this assumption.

We begin by rewriting the memory term using the definitions of $e^{-s\mathcal{L}}$ and $e^{sQ\mathcal{L}}$ and then

computing the integral termwise:

$$\begin{aligned}\mathcal{M}_k =& Pe^{t\mathcal{L}} \int_0^t \left(\sum_{i=0}^{\infty} \frac{(-1)^i s^i}{i!} \mathcal{L}^i \right) P\mathcal{L} \left(\sum_{j=0}^{\infty} \frac{s^j}{j!} (Q\mathcal{L})^j \right) Q\mathcal{L} u_k^0 \, \mathrm{d}s \\ =& Pe^{t\mathcal{L}} \left(\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \frac{(-1)^i t^{i+j+1}}{i!\, j!\, (i+j+1)} \mathcal{L}^i P\mathcal{L} (Q\mathcal{L})^j Q\mathcal{L} u_k^0 \right). \end{aligned} \tag{4.4}$$

We assume the integrand is sufficiently smooth that we can interchange the order of the integral and the infinite sums. This represents the major assumption associated with this memory approximation. In the case where $P\mathcal{L}$ and $Q\mathcal{L}$ commute, we can show that (4.3) is equivalent to (4.4). Consider t^n where n is some integer $n \geq 1$. The term in the BCH approximation (4.3) associated with this power of t is

$$\mathcal{M}_k^n = Pe^{t\mathcal{L}} (-1)^{n+1} \frac{t^n}{n!} (P\mathcal{L})^n Q\mathcal{L} u_k^0. \tag{4.5}$$

The term associated with this power of t in the full expansion (4.4) is:

$$\begin{aligned}\mathcal{M}_k'^n =& Pe^{t\mathcal{L}} \sum_{i=0}^{n-1} \frac{(-1)^i t^n}{i!\, (n-1-i)!\, n} \mathcal{L}^i P\mathcal{L} (Q\mathcal{L})^{n-1-i} Q\mathcal{L} u_k^0 \\ =& \frac{t^n}{n} Pe^{t\mathcal{L}} \sum_{i=0}^{n-1} \frac{(-1)^i}{i!\, (n-1-i)!} (P\mathcal{L} + Q\mathcal{L})^i P\mathcal{L} (Q\mathcal{L})^{n-1-i} Q\mathcal{L} u_k^0 \end{aligned} \tag{4.6}$$

We expand the binomial in (4.6) and use the fact that $P\mathcal{L}$ and $Q\mathcal{L}$ commute to group all $P\mathcal{L}$ and $Q\mathcal{L}$ together:

$$\begin{aligned}\mathcal{M}_k'^n =& \frac{t^n}{n} Pe^{t\mathcal{L}} \sum_{i=0}^{n-1} \frac{(-1)^i}{i!\, (n-1-i)!} \sum_{l=0}^{i} \frac{i!}{l!\, (i-l)!} (P\mathcal{L})^{i-l+1} (Q\mathcal{L})^{n-1-i+l} Q\mathcal{L} u_k^0 \\ =& \frac{t^n}{n} Pe^{t\mathcal{L}} \sum_{i=0}^{n-1} \sum_{l=0}^{i} \frac{(-1)^i}{l!\, (i-l)!\, (n-1-i)!} (P\mathcal{L})^{i-l+1} (Q\mathcal{L})^{n-1-i+l} Q\mathcal{L} u_k^0. \end{aligned}$$

We now implement a change of variables $a = i + 1$ and $b = i - l + 1$:

$$\begin{aligned}\mathcal{M}_k'^n =& \frac{t^n}{n} P e^{t\mathcal{L}} \sum_{a=1}^{n} \sum_{b=1}^{a} \frac{(-1)^{a+1}}{(a-b)!\,(b-1)!\,(n-a)!} (P\mathcal{L})^b (Q\mathcal{L})^{n-b} Q\mathcal{L} u_k^0 \\ =& \frac{t^n}{n} P e^{t\mathcal{L}} \sum_{b=1}^{n} \frac{1}{(b-1)!} \left(\sum_{a=b}^{n} \frac{(-1)^{a+1}}{(a-b)!\,(n-a)!} \right) (P\mathcal{L})^b (Q\mathcal{L})^{n-b} Q\mathcal{L} u_k^0. \end{aligned} \tag{4.7}$$

Consider the term in parentheses. For $b = n$, this term is:

$$\sum_{a=n}^{n} \frac{(-1)^{a+1}}{(a-b)!\,(n-a)!} = (-1)^{n+1}. \tag{4.8}$$

For $b < n$, let $c = a - b$. Then the term in parentheses in (4.7) is:

$$\begin{aligned}\sum_{a=b}^{n} \frac{(-1)^{a+1}}{(a-b)!\,(n-a)!} =& \sum_{c=0}^{n-b} \frac{(-1)^{b+c+1}}{c!\,(n-b-c)!} \\ =& \frac{(-1)^{b+1}}{(n-b)!} \sum_{c=0}^{n-b} \frac{(n-b)!}{c!\,((n-b)-c)!} (-1)^c (1)^{(n-b)-c} \\ =& \frac{(-1)^{b+1}}{(n-b)!} (1-1)^{n-b} \\ =& 0. \end{aligned}$$

Thus, the only nonzero term in (4.7) is when $a = b = n$, so

$$\mathcal{M}_k'^n = P e^{t\mathcal{L}} (-1)^{n+1} \frac{t^n}{n!} (P\mathcal{L})^n Q\mathcal{L} u_k^0 = \mathcal{M}_k^n.$$

Therefore, the BCH approximation $\mathcal{M}_k^n$ is a special case of this full expansion of the memory integral subject to the condition $P\mathcal{L}Q\mathcal{L} = Q\mathcal{L}P\mathcal{L}$. Now, consider the full expansion of the memory kernel (4.4). We begin to analyze this formulation of the memory by writing the

first few terms arranged by powers of t:

$$\mathcal{M}_k = tPe^{t\mathcal{L}}\left[P\mathcal{L}Q\mathcal{L}\right]u_k^0 - \frac{t^2}{2}Pe^{t\mathcal{L}}\left[\mathcal{L}Q\mathcal{L}Q\mathcal{L} - P\mathcal{L}P\mathcal{L}Q\mathcal{L}\right]u_k^0 + O(t^3) \tag{4.9}$$

The $O(t)$ term is the t-model once again. The $O(t^2)$ term, presents a new problem. The second term in it can be computed in a manner similar to the t-model, but the first term is not projected prior to its evolution. When the operator $\mathcal{L}$ is applied to any quantity, the result is a function of *all* modes, both resolved and unresolved. If we apply the evolution operator $e^{t\mathcal{L}}$ to a term of this form, we would need to evolve forward in time a quantity that depends upon unresolved modes, necessitating knowledge of the dynamics of the unresolved modes. This makes it impossible to compute as part of a reduced order model. In the BCH approximation, there were no terms that lacked a leading P, so this problem was avoided. To close the model in the resolved variables we start by constructing an additional reduced order model for the unclosed term. This term is $Pe^{t\mathcal{L}}\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0$. First note that

$$Pe^{t\mathcal{L}}\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0 = \frac{\partial}{\partial t}Pe^{t\mathcal{L}}P\mathcal{L}Q\mathcal{L}u_k^0. \tag{4.10}$$

That is, it is the derivative of the t-model term itself. Now consider a reduced order model for this derivative under the Mori-Zwanzig formalism again:

$$\begin{aligned}\frac{\partial}{\partial t}Pe^{t\mathcal{L}}P\mathcal{L}Q\mathcal{L}u_k^0 &= Pe^{t\mathcal{L}}P\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0 + P\int_0^t e^{(t-s)\mathcal{L}}P\mathcal{L}e^{sQ\mathcal{L}}Q\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0\,\mathrm{d}s \\ &= Pe^{t\mathcal{L}}P\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0 + Pe^{t\mathcal{L}}\left(\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{(-1)^i t^{i+j+1}}{i!\,j!\,(i+j+1)}\mathcal{L}^i P\mathcal{L}(Q\mathcal{L})^{j+1}P\mathcal{L}Q\mathcal{L}u_k^0\right).\end{aligned} \tag{4.11}$$

If we replace $Pe^{t\mathcal{L}}\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0$ in (4.9) with (4.11), only the first term of (4.11) contributes at the $O(t^2)$ level. All the double sum terms contribute at $O(t^3)$ and higher. We find

$$\mathcal{M}_k = tPe^{t\mathcal{L}}\left[P\mathcal{L}Q\mathcal{L}\right]u_k^0 - \frac{t^2}{2}Pe^{t\mathcal{L}}P\mathcal{L}\left[P\mathcal{L} - Q\mathcal{L}\right]Q\mathcal{L}u_k^0 + O(t^3) \tag{4.12}$$

where all the terms are now projected prior to evolution and so involve *only* resolved variables. Observe that now the second order term in (4.12) *no longer matches* the second order term in (4.3) even when $P\mathcal{L}$ and $Q\mathcal{L}$ commute. The reason is that the contribution of one term $Pe^{t\mathcal{L}}Q\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0$ has been rewritten as a sum that contributes only at $O(t^3)$ and higher. If $P\mathcal{L}$ and $Q\mathcal{L}$ commute, this sum representation of $Pe^{t\mathcal{L}}Q\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0$ would identically match $Pe^{t\mathcal{L}}P\mathcal{L}Q\mathcal{L}Q\mathcal{L}u_k^0$, and the two would indeed cancel. Thus, the BCH approximation remains a special case of the complete memory approximation, and the complete memory approximation remains an expansion of the memory term that does not neglect any terms.

We can naturally extend this to higher orders. Many of the terms in (4.4) contain a leading $\mathcal{L}$ like the "problem term" at $O(t^2)$ discussed above. These terms, if included as part of a reduced order model, would not be closed in the resolved variables. Here we should be explicit. These "problem terms" with a leading $\mathcal{L}$ in fact only depend on the resolved variables (due to the projector P applied after the evolution). However, because the initial condition is evolved forward in time *before* this projection, the functional dependence cannot be identified without knowledge of the evolution of the unresolved modes. In order to construct a closed reduced order model at higher orders, we need to approximate these terms. Consider one of these "problem terms," which we designate C:

$$C = t^i Pe^{t\mathcal{L}}\mathcal{L}B(u_k^0), \tag{4.13}$$

where B is some combination of projections and Liouvillian operators applied to u_k^0. The first step to dealing with problem terms like C is to rewrite it as a time derivative

$$C = t^i Pe^{t\mathcal{L}}\mathcal{L}B(u_k^0) = t^i P\frac{\partial}{\partial t}e^{t\mathcal{L}}B(u_k^0). \tag{4.14}$$

The Mori-Zwanzig formalism can similarly be used to expand this time derivative. The memory integral that appears can be approximated with the same double sum technique

employed above:

$$\begin{aligned} C =& t^i \left(P e^{t\mathcal{L}} P\mathcal{L}B(u_k^0) + P e^{t\mathcal{L}} \int_0^t e^{-s\mathcal{L}} P\mathcal{L} e^{sQ\mathcal{L}} Q\mathcal{L}B(u_k^0)\,\mathrm{d}s \right) \\ =& t^i \left[P e^{t\mathcal{L}} P\mathcal{L}B(u_k^0) + P e^{t\mathcal{L}} \left(\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \frac{(-1)^i t^{i+j+1}}{i!\,j!\,(i+j+1)} \mathcal{L}^i P\mathcal{L}(Q\mathcal{L})^j Q\mathcal{L}B(u_k^0) \right) \right]. \end{aligned} \tag{4.15}$$

By repeatedly applying this technique, we can construct an approximation for (4.4) in which every term has a leading P before the evolution operator is applied. The resulting series is written as:

$$\mathcal{M}_k = \sum_{i=1}^{\infty} \frac{(-1)^{i+1} t^i}{i!} R_k^i(\hat{\mathbf{u}}). \tag{4.16}$$

Recall that we designate $\hat{\mathbf{u}}$ as the resolved modes and that the operator P ensures that each R_k^i is a function of those modes only. Different approximation schemes can be constructed by truncating this series at different terms. As discussed above, the $O(t)$ term corresponds to the t-model

$$R_k^1(\hat{\mathbf{u}}) = P e^{t\mathcal{L}} P\mathcal{L}Q\mathcal{L}u_k^0, \tag{4.17}$$

and the $O(t^2)$ term is:

$$R_k^2(\hat{\mathbf{u}}) = P e^{t\mathcal{L}} P\mathcal{L}\left[P\mathcal{L} - Q\mathcal{L}\right] Q\mathcal{L}u_k^0. \tag{4.18}$$

We call this the t^2-model term. By grouping terms in the series (4.4) in powers of t and using the technique described above to further expand problem terms, we can uniquely define $R_k^i(\hat{\mathbf{u}})$ for any positive integer i. We automated this process in a Mathematica notebook, which is available in the `Renormalized_Mori_Zwanzig` git repository [87]. The $O(t^3)$ t^3-model term is:

$$R_k^3(\hat{\mathbf{u}}) = P e^{t\mathcal{L}} P\mathcal{L}[P\mathcal{L}P\mathcal{L} - 2P\mathcal{L}Q\mathcal{L} - 2Q\mathcal{L}P\mathcal{L} + Q\mathcal{L}Q\mathcal{L}]Q\mathcal{L}u_k^0. \tag{4.19}$$

Finally, the $O(t^4)$ t^4-model term is:

$$R_k^4(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathcal{L}\Big[P\mathcal{L}P\mathcal{L}P\mathcal{L} - 3P\mathcal{L}P\mathcal{L}Q\mathcal{L} - 5P\mathcal{L}Q\mathcal{L}P\mathcal{L} - 3Q\mathcal{L}P\mathcal{L}P\mathcal{L} + 3P\mathcal{L}Q\mathcal{L}Q\mathcal{L} + 5Q\mathcal{L}P\mathcal{L}Q\mathcal{L} + 3Q\mathcal{L}Q\mathcal{L}P\mathcal{L} - Q\mathcal{L}Q\mathcal{L}Q\mathcal{L}\Big]Q\mathcal{L}u_k^0. \quad (4.20)$$

Note that every term in the $O(t^3)$ term of the Taylor series has four applications of the $\mathcal{L}$ differential operator, while the $O(t^4)$ terms all have five applications of $\mathcal{L}$ with different permutations of projections. We believe that the complete memory approximation is an effective approximation scheme because it groups terms that have the same number of differential operators.

4.2 Renormalizing reduced order models

The complete memory approximation framework provides a series representation of the memory integral. Different ROMs can be created by truncating the series at different terms. In this case, the differential equation for a resolved mode is:

$$\frac{dPu_k}{dt} = R_k^0(\hat{\mathbf{u}}) + \sum_{i=1}^{n} \frac{(-1)^{i+1}t^i}{i!} R_k^i(\hat{\mathbf{u}}), \quad (4.21)$$

where $R_k^0(\hat{\mathbf{u}})$ is the Markov term, $R_k^1(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}[P\mathcal{L}Q\mathcal{L}]u_k^0$ is the t-model, and higher ordered terms are found by grouping similar powers of t in the complete memory approximation. For the examples we have tried, the resulting ROMs are unstable. This was the case for both the BCH memory approximation and the complete memory approximation, both of which can be written as (4.21) with suitable definitions of $R_k^i(\hat{\mathbf{u}})$. We will assume for the remainder of this text that we are using the $R_k^i(\hat{\mathbf{u}})$ as defined by the complete memory approximation.

The ROMs defined by the BCH memory approximation have been stabilized through *renormalization*. We attach additional coefficients to each term in the series, such that the terms represent an *effective memory*, given knowledge only of the resolved modes [104, 105].

The evolution equation for a reduced variable then becomes

$$\frac{dPu_k}{dt} = R_k^0(\hat{\mathbf{u}}) + \sum_{i=1}^{n} \alpha_i(t) t^i R_k^i(\hat{\mathbf{u}}). \tag{4.22}$$

Here, we allow the renormalization coefficients $\alpha_i(t)$ to be time dependent. This gives us the flexibility to allow the functional form of the effective memory to be dynamic if necessary. These coefficients must be chosen in a way that captures information we know about the memory term. Selecting them is done in a problem-specific manner, frequently by comparing the reduced system to a larger system prior to the point where the larger system becomes unresolved. Concrete examples and details will be provided in the following sections.

4.2.1 Analogies to renormalization in physics

Renormalization is a loaded term. However, we believe that it is an accurate description of the sentiment behind our procedure. It is also an inherently multiscale mode of thinking that interfaces well with the fundamental ideas of the Mori-Zwanzig formalism. We present here a brief overview of renormalization in the physics literature (adapted from [29]) and make analogies to our approach here.

Whenever we consider a physical problem, we select a physical model that is valid on a particular scale but fails to represent the whole truth. We call this the macroscale. For example, we model a fluid by its hydrodynamic variables instead of molecular trajectories. One can ask the question, "What should we do if we go are faced with a scale or problem where our physical model is not valid, but we *do not know* the more fundamental model that should be used?" Renormalization is a process for taking the known model and modifying it into an approximate but effective model beyond its domain of validity. In practice, we take functional forms from our physical model and modify their parameters to match reality. One of the earliest examples was when Stokes modeled the drag on a sphere in a fluid flow using a modified or effective mass for the sphere. This is very similar to our renormalization procedure. We assume the functional forms of our ROM terms are correct, but that they

are being used outside their domain of validity. We modify them with effective coefficients to match reality.

Renormalization in physics requires these modifying coefficients because using a model outside the scale of validity without adjusting parameters often leads to singularities. The concept of renormalization has also been used in the context of singular perturbation. If one attempts to solve a singularly perturbed problem using a standard perturbation expansion, the result is "secular terms" that violate the assumption of the perturbation (namely that there is a clear ordering of terms from large to small). One can eliminate the effects of these secular terms by changing from a standard perturbation expansion to one with effective coefficients. These coefficients are selected specifically to eliminate the assumption-violating behavior of the secular terms. All of these concepts were eventually formalized in renormalization group theory. The scale-dependent effective parameters form a group of transformations and studying the fixed points of the group flow leads to conclusions about phase transitions.

All of these ideas are echoed in our work. We have a reduced order model that would be effective if applied to enough modes to fully resolve the solution. But because we cannot fully resolve the systems we wish to simulate (either because they are finite but too large, or they develop shocks and become impossible to resolve with a finite number of variables), we are necessarily unable to capture the truth without making modifications. The instability of the models thus are not a shortcoming, but a signal that we are applying these models beyond their range of validity. By changing the parameters to effective ones through renormalization coefficients, we expand the range of validity. These effective parameters come from attempting to match real observations from a solution that we trust at least up to a point. Furthermore, as in the work from the renormalization group theory, we will find that these renormalization coefficients function best when they themselves have a strict scaling structure.

4.3 Application 1: The Korteweg-de Vries equation with small dispersion

The work in this section was submitted for publication at the SIAM Multiscale Modeling and Simulation journal in October 2017. It was accepted pending revisions which have since been submitted. It is also available in the arXiv at [88].

The Korteweg-de Vries equation with small dispersion ϵ is

$$u_t + uu_x + \epsilon^2 u_{xxx} = 0. \tag{4.23}$$

We will consider solving this equation on $[0, 2\pi]$ with periodic boundary conditions and initial condition $v^0 = u(x, 0)$.

Renormalized MZ models have already been applied to the Burgers equation which corresponds to the case $\epsilon = 0$ in (4.23) [104, 105]. The Burgers equation develops singularities in the form of shocks in finite time. We will revisit this case in Section 4.5. For $\epsilon \neq 0$, the dispersive term precludes a finite time singularity. Instead, the solution can be fully resolved with $O(1/\epsilon)$ Fourier modes [111]. Additionally, the presence of a dispersive term causes energy to flow from the resolved modes to the unresolved modes *and back*, unlike Burgers where there is a constant drain of energy out of the resolved variables. We have to make a comment here about terminology. In the dispersive equation community (which includes KdV), the square magnitude of a Fourier mode is called the "mass" of the mode while in the fluid dynamics community (which includes the study of Burgers) it is called the "energy" of the mode. In this section we will use the word "mass."

Because of the periodic boundary conditions, we use Fourier series as a basis for the solution. That is, let

$$u(x, t) = \sum_{k \in F \cup G} u_k(t) e^{ikx}$$

where $F = [-N, \ldots, N-1]$, $G = [-M, \ldots, -N-1, N, \ldots, M-1]$, and $F \cup G = [-M, \ldots, M-1]$ for $N < M$. To avoid an asymmetry between positive and negative modes we always set to zero the coefficient for the most negative mode ($-N$ and $-M$, depending on the context).

We call F the *resolved* modes and G the *unresolved* modes. Let $\mathbf{u} = \{u_k(t)\}_{k\in F\cup G}$. We partition $\mathbf{u} = (\hat{\mathbf{u}}, \tilde{\mathbf{u}})$ where $\hat{\mathbf{u}} = \{u_k\}_{k\in F}$ and $\tilde{\mathbf{u}} = \{u_k\}_{k\in G}$. Our goal is to construct a reduced order model for each component $\hat{u}_k(t)$ of $\hat{\mathbf{u}}$. $\mathbf{u}$ is the "full" model, which will be fully resolved if M exceeds the maximal resolution for the chosen ϵ.

The equation of motion for the Fourier mode u_k is

$$\frac{du_k}{dt} = R_k(\mathbf{u}) = i\epsilon^2 k^3 u_k - \frac{ik}{2} \sum_{\substack{p+q=k \\ p,q\in F\cup G}} u_p u_q. \tag{4.24}$$

The convolution sum in the second term on the right hand side can be computed efficiently by transforming data to real space and computing the sum there as the FFT of the product of the real space solution with itself.

We define $\mathcal{L}$ as in (2.2) such that $\mathcal{L}u_k^0 = R_k(\mathbf{u}^0)$. We must define the projection operator P. Consider a function $h(\mathbf{u}^0)$ that depends on all the Fourier modes. We define $Ph(\mathbf{u}^0) = Ph(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) = h(\hat{\mathbf{u}}^0, 0)$. That is, we set each unresolved variable to zero. Our choice of projection operator is motivated by our interest in studying the transfer of activity to smaller scales when starting with a smooth initial condition. The choice of projection facilitates the computation of the memory terms. In order to remain consistent with our projection operator choice, our initial condition must be $\mathbf{u}^0(x) = (\hat{\mathbf{u}}^0, 0)$. We must begin with an initial condition that does not have any unresolved modes activated. For example, we will use $v^0(x) = \sin(x)$, for which only the first Fourier mode is nonzero.

The Markov term for $u_k(t)$ for $k \in F$ is given by $Pe^{t\mathcal{L}}P\mathcal{L}u_k^0$:

$$R_k^0(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathcal{L}u_k^0 = Pe^{t\mathcal{L}}P\left[i\epsilon^2 k^3 u_k^0 - \frac{ik}{2}\sum_{\substack{p+q=k \\ p,q\in F\cup G}} u_p^0 u_q^0\right]$$

$$R_k^0(\hat{\mathbf{u}}) = i\epsilon^2 k^3 \hat{u}_k - \frac{ik}{2}\sum_{\substack{p+q=k \\ p,q\in F}} u_p u_q.$$

Note that the final P operator actually *commutes* with the nonlinear evolution operator. This is not a general rule, but in fact a consequence of the specific projector we chose. Because the unresolved modes are set to zero with zero variance, we are able to commute P and $e^{t\mathcal{L}}$. Since all our terms have a leading P, when we commute the P with $e^{t\mathcal{L}}$, it has no additional effect.

The Markov term has the same form as the full system, but has been restricted to sums over the resolved modes. We can easily compute the convolution sum in this expression using fast Fourier transforms, but only retaining the resolved modes of the result. In fact, it will be prudent to define a function representing a convolution of two vector valued functions $\mathbf{f}$ and $\mathbf{g}$ with their respective components labeled f_i and g_i. We define the convolution of $\mathbf{f}$ with $\mathbf{g}$ as:

$$C_k(\mathbf{f}(\mathbf{u}), \mathbf{g}(\mathbf{u})) = -\frac{ik}{2} \sum_{p+q=k} f_p(\mathbf{u}) g_q(\mathbf{u}). \tag{4.25}$$

With this definition, the Markov term is

$$R_k^0(\hat{\mathbf{u}}) = i\epsilon^2 k^3 u_k + C_k(\hat{\mathbf{u}}, \hat{\mathbf{u}}). \tag{4.26}$$

It will be useful to define the vector-valued functions $\hat{\mathbf{C}}(\mathbf{f}(\mathbf{u}), \mathbf{g}(\mathbf{u}))$ and $\tilde{\mathbf{C}}(\mathbf{f}(\mathbf{u}), \mathbf{g}(\mathbf{u}))$ whose components are the appropriate convolutions (4.25) with the unresolved modes set to zero and the resolved modes set to zero (respectively).

It can be shown that the Markov term conserves mass in the resolved modes. Thus, it does not allow any transfer of mass out of the resolved modes, which must occur if we are to accurately reproduce what would happen in the full system. That must be accomplished through the memory term. We can compute terms in the complete memory approximation to generate reduced order models for KdV. We will consider up to fourth order in t. This reduced order model for u_k can be written

$$\frac{dPu_k}{dt} = R_k^0(\hat{\mathbf{u}}) + \sum_{i=1}^{4} \alpha_i(t) t^i R_k^i(\hat{\mathbf{u}}). \tag{4.27}$$

The Markov term R_k^0 was computed above in (4.26). We next compute the t-model term $R_k^1(\hat{\mathbf{u}})$. First, we compute $Q\mathcal{L}u_k^0$ (again for $k \in F$):

$$\begin{aligned} Q\mathcal{L}u_k^0 &= \mathcal{L}u_k^0 - P\mathcal{L}u_k^0 \\ &= -\frac{ik}{2} \sum_{\substack{p+q=k \\ p\in F, q\in G}} u_p^0 u_q^0 - \frac{ik}{2} \sum_{\substack{p+q=k \\ p\in G, q\in F}} u_p^0 u_q^0 - \frac{ik}{2} \sum_{\substack{p+q=k \\ p,q\in G}} u_p^0 u_q^0. \end{aligned}$$

Next we compute $P\mathcal{L}Q\mathcal{L}u_k^0$:

$$\begin{aligned} P\mathcal{L}Q\mathcal{L}u_k^0 &= -\frac{ik}{2} \sum_{\substack{p+q=k \\ p\in F, q\in G}} P\mathcal{L}[u_p^0 u_q^0] - \frac{ik}{2} \sum_{\substack{p+q=k \\ p\in G, q\in F}} P\mathcal{L}[u_p^0 u_q^0] \\ &= 2\left(-\frac{ik}{2} \sum_{\substack{p+q=k \\ p\in F, q\in G}} u_p^0 \left(-\frac{iq}{2} \sum_{\substack{r+s=q \\ r,s\in F}} u_r^0 u_s^0 \right) \right). \end{aligned}$$

We have here repeatedly used the projection operator P to eliminate terms that become zero. We have also made use of the symmetry of the two sums that remained. Therefore, the t-model term is

$$R_k^1(\hat{\mathbf{u}}) = 2C_k(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})). \tag{4.28}$$

Because the convolutions involve a function with only resolved modes convolved with a function with only unresolved modes, the result is dealiased by construction if we augment our Fourier vectors with one additional mode. We can compute additional terms entirely in the shorthand established above by recognizing the following rules:

1. Because a convolution sum is a product of terms and $\mathcal{L}$ is a differential operator, it operates according to the product rule. That is, for every argument in a convolution, we get a term that is a duplicate of that convolution, but with $\mathcal{L}$ applied to that

argument. For example:

$$\mathcal{L}C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) = C_k(\mathcal{L}\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) + C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\mathcal{L}\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) + C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \mathcal{L}\hat{\mathbf{u}}^0)).$$

2. Each $\mathcal{L}\hat{\mathbf{u}}^0$ term is expanded as

$$\mathcal{L}\hat{\mathbf{u}}^0 = i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \hat{\mathbf{C}}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0),$$

where $(\hat{\mathbf{u}}^{k3})_j = j^3\hat{u}_j^0$. $\mathcal{L}\tilde{\mathbf{u}}^0$ is expanded in an identical manner, but with each term being the unresolved part, rather than the resolved part.

3. When $\mathcal{L}$ is applied to terms involving powers of k, the following occurs:

$$\begin{aligned}
\mathcal{L}(i\epsilon^2\hat{\mathbf{u}}^{k3})_k &= \mathcal{L}i\epsilon^2k^3u_k^0 \\
&= (i\epsilon^2k^3)^2u_k^0 + (i\epsilon^2k^3)[C_k(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2C_k(\hat{\mathbf{u}^0}, \tilde{\mathbf{u}}^0) + C_k(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0)] \\
&= \left(-\epsilon^4\hat{\mathbf{u}}^{k6} + i\epsilon^2[\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \hat{\mathbf{C}}^{k3}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0)]\right)_k
\end{aligned}$$

where $(\hat{\mathbf{u}}^{k6})_j = j^6u_j^0$, $(\hat{\mathbf{C}}^{k3}(\mathbf{a}, \mathbf{b}))_j = j^3C_j(\mathbf{a}, \mathbf{b})$ for $j \in F$. That is, the further expansion of powers of k proceed in an easily understandable manner.

The projection operator P is also simple to implement. When applied to a convolution, it applies to each term in the convolution. P sets equal to zero any occurrence of $\tilde{\mathbf{u}}^0$. Q can be represented as $I - P$. Finally, $Pe^{t\mathcal{L}}$ merely advances the initial conditions $\mathbf{u}^0$ to the current time $\mathbf{u}$ (and the P commutes with $e^{t\mathcal{L}}$ and has no effect as discussed above). The process of deriving these terms becomes quite exhausting, because each nested convolution expands into multiple terms with additional applications of $\mathcal{L}$. Each term in the complete memory approximation also contains more terms than those that preceded it. We implemented these definitions into a Mathematica notebook, which can then generate ROMs of any order for KdV. This is available in the git repository `Renormalized_Mori_Zwanzig` [87]. In fact,

through redefinitions of P and $\mathcal{L}$, this software can be used to derive ROMs of any order for a generic PDE with a generic projector. In the next two sections we will reuse this software. Using this, we computed expressions for R_k^2, R_k^3, and R_k^4. We will demonstrate the derivation of R_k^2 by hand here as an example. We have already computed $Q\mathcal{L}u_k^0$, but future applications of $Q\mathcal{L}$ will be computed by $Q\mathcal{L} = \mathcal{L} - P\mathcal{L}$. Thus, the R_k^2 is:

$$Pe^{t\mathcal{L}}[P\mathcal{L}P\mathcal{L}Q\mathcal{L} - P\mathcal{L}Q\mathcal{L}Q\mathcal{L}]u_k^0 = Pe^{t\mathcal{L}}[2P\mathcal{L}P\mathcal{L}Q\mathcal{L} - P\mathcal{L}\mathcal{L}Q\mathcal{L}]u_k^0.$$

The first term can be computed by applying $P\mathcal{L}$ to $P\mathcal{L}Q\mathcal{L}u_k^0$, which we have already found:

$$\begin{aligned}
P\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0 =& P\mathcal{L}\left[2C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0))\right] \\
=& P\Big[2C_k(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \hat{\mathbf{C}}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0), \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) \\
&+ 4C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(i\epsilon^2\hat{\mathbf{u}}^{k3} + \mathbf{C}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \hat{\mathbf{C}}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0), \hat{\mathbf{u}}^0))\Big] \\
=& 2C_k(i\epsilon^2\hat{\mathbf{u}}^{k3} + \mathbf{C}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0), \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) + 4C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0), \hat{\mathbf{u}}^0)).
\end{aligned}$$

The second term in the $O(t^2)$ expansion is:

$$\begin{aligned}
P\mathcal{L}\mathcal{L}Q\mathcal{L}u_k^0 =& P\mathcal{L}\mathcal{L}[2C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + C_k(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0)] \\
=& P\mathcal{L}\Big[2C_k(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \hat{\mathbf{C}}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0), \tilde{\mathbf{u}}^0) \\
&+ 2C_k(\hat{\mathbf{u}}^0, i\epsilon^2\tilde{\mathbf{u}}^{k3} + \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \tilde{\mathbf{C}}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0)) \\
&+ 2C_k(\tilde{\mathbf{u}}^0, i\epsilon^2\tilde{\mathbf{u}}^{k3} + \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \tilde{\mathbf{C}}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0))\Big].
\end{aligned}$$

We will apply the projection P in the same step that we apply $\mathcal{L}$ to save space writing out

terms that will be eliminated. The result is:

$$\begin{aligned}
P\mathcal{L}\mathcal{L}Q\mathcal{L}u_k^0 =& 4C_k(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0), \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) \\
&+ 2C_k(\hat{\mathbf{u}}^0, i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0))) \\
&+ 2C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0), \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)).
\end{aligned}$$

Combining these two results gives us the $O(t^2)$ term of the complete memory approximation:

$$\begin{aligned}
R_k^2(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}[2P\mathcal{L}P\mathcal{L}Q\mathcal{L} - P\mathcal{L}\mathcal{L}Q\mathcal{L}]u_k^0 \\
=& -2C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&- 2C_k(\hat{\mathbf{u}}, i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&- 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))).
\end{aligned} \tag{4.29}$$

One can clearly see how frustrating it would be to compute the $O(t^3)$ and $O(t^4)$ terms by hand. Instead, we use the automated procedure to derive these expressions. We have reproduced them here:

$$\begin{aligned}
R_k^3(\hat{\mathbf{u}}) =& 2C_k(\hat{\mathbf{u}}, -\epsilon^4\tilde{\mathbf{C}}^{k6}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, 2i\epsilon^2\hat{\mathbf{u}}^{k3} + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&+ 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, -\epsilon^4\hat{\mathbf{u}}^{k6} + i\epsilon^2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&+ i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, -2(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&+ 2\tilde{\mathbf{C}}(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&+ 2\tilde{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&+ 6C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))).
\end{aligned} \tag{4.30}$$

$$
\begin{aligned}
&R_k^4(\hat{\mathbf{u}}) = \\
&2C_k(\hat{\mathbf{u}}, i\epsilon^6\tilde{\mathbf{C}}^{k9}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 2\epsilon^4\tilde{\mathbf{C}}^{k6}(\hat{\mathbf{u}}, 3i\epsilon^2\hat{\mathbf{u}}^{k3} + 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\quad - 2i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, -3\epsilon^4\hat{\mathbf{u}}^{k6} + 3i\epsilon^2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, 3i\epsilon^2\hat{\mathbf{u}}^{k3} + 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 5\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) + i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) \\
&\qquad\qquad - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, 3i\epsilon^2\hat{\mathbf{u}}^{k3} + 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&\quad - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^6\hat{\mathbf{u}}^{k9} + \epsilon^4\hat{\mathbf{C}}^{k6}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 2i\epsilon^2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, \epsilon^4\hat{\mathbf{u}}^{k6} - i\epsilon^2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad\quad - 3i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, 5i\epsilon^2\hat{\mathbf{u}}^{k3} + 5\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&\qquad - 2\hat{\mathbf{C}}(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad - 6\hat{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad - \epsilon^4\tilde{\mathbf{C}}^{k6}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, -3i\epsilon^2\hat{\mathbf{u}}^{k3} - 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, -3\epsilon^4\hat{\mathbf{u}}^{k6} + 3i\epsilon^2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, 3i\epsilon^2\hat{\mathbf{u}}^{k3} + 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 5\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) + i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) \\
&\qquad\quad + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, -3i\epsilon^2\hat{\mathbf{u}}^{k3} - 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&\qquad + 2\tilde{\mathbf{C}}(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), 3i\epsilon^2\hat{\mathbf{u}}^{k3} + 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad + 2\tilde{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&\quad + 2i\epsilon^2\tilde{\mathbf{C}}^{k3}(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), -3\epsilon^2\hat{\mathbf{u}}^{k3} - 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\quad - 2\tilde{\mathbf{C}}(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), 3\epsilon^4\hat{\mathbf{u}}^{k6} - 3i\epsilon^2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, -3i\epsilon^2\hat{\mathbf{u}}^{k3} - 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 5\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad\qquad - i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, 3i\epsilon^2\hat{\mathbf{u}}^{k3} + 3\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&\quad + 2\tilde{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), \epsilon^4\hat{\mathbf{u}}^{k6} - i\epsilon^2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 2\mathbf{C}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad\qquad - 3i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, 5i\epsilon^2\hat{\mathbf{u}}^{k3} + 5\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&\quad - 2i\epsilon^2\tilde{\mathbf{C}}^{k3}(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&- 8C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), -\epsilon^4\tilde{\mathbf{C}}^{k6}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}}, -2i\epsilon^2\hat{\mathbf{u}}^{k3} - 2\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, -\epsilon^4\hat{\mathbf{u}}^{k6} + i\epsilon^2\hat{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) + i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) \\
&\qquad\quad + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, -2i\epsilon^2\hat{\mathbf{u}}^{k3} - 2\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&\qquad + 2\tilde{\mathbf{C}}(i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) \\
&\qquad + 2\tilde{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&+ 48C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})), i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \\
&- 6C_k(i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + \tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})), \\
&\qquad\qquad i\epsilon^2\tilde{\mathbf{C}}^{k3}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, i\epsilon^2\hat{\mathbf{u}}^{k3} + \hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}) + e\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))) \qquad (4.31)
\end{aligned}
$$

These terms involve convolutions of unresolved terms with other unresolved terms, necessitating that we further augment our vectors using the 3/2-rule to dealias the results. The functional form of these higher-order terms in the complete memory approximation are quite complicated, and it is unlikely that a mathematical modeler would propose them. However, because they are derived from the dynamics themselves through the Mori-Zwanzig formalism, we find that they inherit significant structure from the full KdV equation.

4.3.1 Renormalization Coefficients

The non-renormalized ROMs computed from the complete memory approximation (4.21) are numerically unstable, so we focus on the renormalized ROMs (4.22). We must develop a procedure for computing the renormalization coefficients α_i. An important quantity in a ROM for KdV is the mass in a Fourier mode:

$$M_k(t) = |u_k(t)|^2. \tag{4.32}$$

For select values of ϵ, we computed the exact solution using a sufficient number of modes to ensure the solution was fully resolved, from which we compute the rate of change of the mass:

$$\Delta M_k(t) = u_k(t)\overline{R_k(\mathbf{u})} + \overline{u_k(t)}R_k(\mathbf{u}). \tag{4.33}$$

The mass in a subset of modes is not monotonically decreasing. Instead, there is a "mass rebound" as the dispersive term opposes the formation of a shock, and mass returns from high-frequency modes to low-frequency modes.

We also computed $R_k^i(\hat{\mathbf{u}})$ for ROMs constructed for those same subsets using truncated versions $\hat{\mathbf{u}}$ of the exact solution $\mathbf{u}$. From this, we can measure the impact each R_k^i term has upon the rate of change of the mass in individual modes. This is given by:

$$\Delta M_k^i(t) = u_k(t)\overline{R_k^i(\hat{\mathbf{u}})} + \overline{u_k(t)}R_k^i(\hat{\mathbf{u}}). \tag{4.34}$$

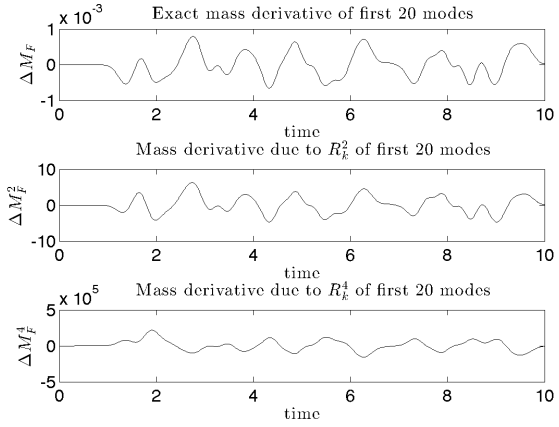

Figure 4.1: (Top) The exact net mass derivative of the first $N = 20$ positive modes of the solution to the KdV equation with $\epsilon = 0.1$ up to time 10 ($\sum_{k=1}^{20} \Delta M_k(t)$). (Middle) The rate of change of the mass in the first $N = 20$ positive modes for the R_k^2 term in an ROM of size $N = 20$ provided the truncated exact solution $\hat{\mathbf{u}}$ as input ($\sum_{k=1}^{20} \Delta M_k^2(t)$). It seems to be proportional to the top figure with a positive constant of proportionality. (Bottom) The rate of change of the mass in the first $N = 20$ positive modes for the R_k^4 term in an ROM of size $N = 20$ provided the truncated exact solution $\hat{\mathbf{u}}$ as input ($\sum_{k=1}^{20} \Delta M_k^4(t)$). It seems to be proportional to the top figure with a negative constant of proportionality.

Any net rate of change of mass in the resolved modes must be accounted for by these memory terms alone, because the Markov term conserves mass in the resolved modes.

We found that the $\sum_{k\in F} \Delta M_k^2$ and $\sum_{k\in F} \Delta M_k^4$ terms closely mirror the exact net mass derivative $\sum_{k\in F} \Delta M_k$ for a given subset of modes, as depicted in Figure 4.1. From this, we draw several conclusions. First, this close agreement suggests that the complete memory approximation is in some sense a "correct" way of expanding the memory integral. Second, we observe that the rate of change of mass due to the R_k^2 and R_k^4 terms are approximately proportional to the exact mass derivative at all times. In an ROM, these terms would be multiplied by t^2 and t^4, respectively, eliminating the agreement with the exact results as time passes. One possible means of counteracting this effect is to propose a functional form

for the renormalization coefficients of

$$\alpha_i(t) = \beta_i t^{-i}, \tag{4.35}$$

such that the time dependence of the renormalization coefficient cancels the time dependence in the memory terms. We use this as an ansatz that is qualitatively suggested by the results in Figure 4.1, though other forms could be considered.

For a fixed ϵ, we used a least squares fit to identify optimal choices for the constants β_i. Suppose we have the exact solution at times $t^* = \{t_0, \ldots, t_m\}$. The cost function for a given set of resolved modes F is:

$$C_F(\beta_1, \beta_2, \beta_3, \beta_4) = \sum_{t \in t^*} \sum_{k \in F} \left[\Delta M_k(t) - \Delta M_k^0(t) - \sum_{i=1}^{4} \beta_i \Delta M_k^i(t) \right]^2. \tag{4.36}$$

That is, we sought to minimize the error in representing the derivative of the mass of each individual mode at $\{t_0, \ldots, t_m\}$. We tried several other cost functions, such as the square errors of the real and imaginary parts of the derivative of each mode, without finding significant differences in the optimizing coefficients.

We conducted least squares fits for many sets of resolved modes and choices of dispersion ϵ. In each case, we used data for $t_i \in [t_0, t_m]$ in increments of 0.001. In practice, we found the results were insensitive to the specific details of the data used. As long as the width of the interval, $t_m - t_0$, was greater than 2 time units, the coefficients were not substantially different. They seem to "lock in" on a particular value (Figure 4.2). The optimal coefficients for the second and fourth terms were remarkably stable to the set of timesteps used. On the other hand, the optimal coefficients for the first and third terms varied significantly as we adjusted the timesteps used in the fit. In fact, as the length of the fitted timeframe grew, these coefficients tended closer towards zero. This suggests to us that the first order and third order coefficients are effectively fitting noise, and instead should be discarded in

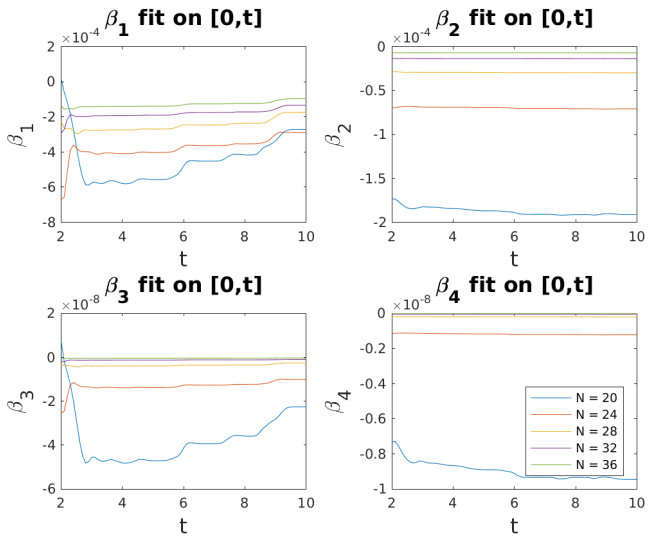

Figure 4.2: Optimal coefficients for ROMs of resolution $N = 20, 24, \ldots, 36$ for $\epsilon = 0.1$. Fits were computed by minimizing (4.36) on different sets of times t^*. In each plot, we fit over $t^* = [0, T]$ where $T = 2, 2.1, \ldots, 10$. Note that the second order and fourth order coefficients are impressively stable, while the first and third order coefficients are less stable (even varying in sign in some cases). For this reason, we suspect the second and fourth order models are the most important to include.

favor of only the second and fourth order coefficients. For this reason, we hypothesize that $\beta_1 = \beta_3 = 0$. We also see that, as N increases the magnitude of each coefficient seems to grow smaller. For all subsequent calculations, we used $t^* = \{0, 0.1, \ldots, 10\}$ for our fits. We found that the optimal coefficients displayed power law behavior *both* in ϵ and in the size of the reduced order model N. We used an additional least squares fit to identify the scaling coefficients for each variable.

The power law behavior of the coefficients suggests the presence of incomplete similarity for appropriate non-dimensional parameters [7]. After we nondimensionalize the problem with characteristic length L and velocity U, (4.23) becomes

$$u_t + uu_x + \frac{1}{Re^2} u_{xxx} = 0, \tag{4.37}$$

where $Re = \frac{\sqrt{U}L}{\epsilon}$ is a "dispersive" Reynolds number. From (4.22) we know that the renormalization coefficients $\alpha_i(t)$ are non-dimensional. This is because the terms $t^i R_k^i$ that they multiply in (4.22) have the correct dimensions by construction, since they are produced through the MZ formalism and not added by hand. One can also recognize that these terms are nondimensional because they take the place of clearly nondimensional coefficients in the Taylor series expansion (4.4). Also, from (4.35) we know that the prefactors β_i have dimension $T^i = (\frac{L}{U})^i$. If in addition to the Reynolds number Re we define the non-dimensional parameters $\Pi_i = \frac{\beta_i}{(L/U)^i}$ and $\Lambda = \frac{N}{L^{-1}}$ we can represent the renormalization coefficients as

$$\Pi_i = a_i Re^{b_i} \Lambda^{c_i}. \tag{4.38}$$

Note that we here understand N as the maximal wavenumber, which has units $(\text{length})^{-1}$. It happens to also be the number of variables we evolve, referred to as the *resolution*. Figure 4.3 presents the results of the least squares fit of $\log(\Pi_2)$ and $\log(\Pi_4)$ against $\log(Re)$ and $\log(\Lambda)$. For $i = 1$ and $i = 3$, the data suggested $a_i = 0$, so the power law exponents b_i and c_i were irrelevant and we exclude them from Figure 4.3. It is notable that the odd numbered terms in the memory expansion seem to be unnecessary for capturing the dynamics. For $i = 2$, we found $a_2 = -1.2006$, $b_2 = 3.7013$, and $c_2 = -5.7384$. For $i = 4$, we found $a_4 = -0.3318$, $b_4 = 7.4052$, and $c_4 = -11.4715$ (we will refer to reduced order models of this form with these renormalization coefficients as the fourth order complete renormalized model). We conducted a similar fit with Π_1 and Π_3 set to zero *a priori* and found the resulting parameters for Π_2 and Π_4 changed very little. This corroborates our choice to set Π_1 and Π_3 to zero explicitly. We note that b_4 and c_4 are roughly double b_2 and c_2 respectively. Furthermore, note that the signs of the renormalization coefficients are negative. In the non-renormalized model, the coefficients on R_k^2 and R_k^4 are $-1/2$ and $-1/24$, respectively. Therefore, we see the sign of the renormalization coefficients matches that of the unrenormalized model. It seems likely that these power laws belie an even deeper structure in the memory inherited by the ROMs from the original KdV equation than we

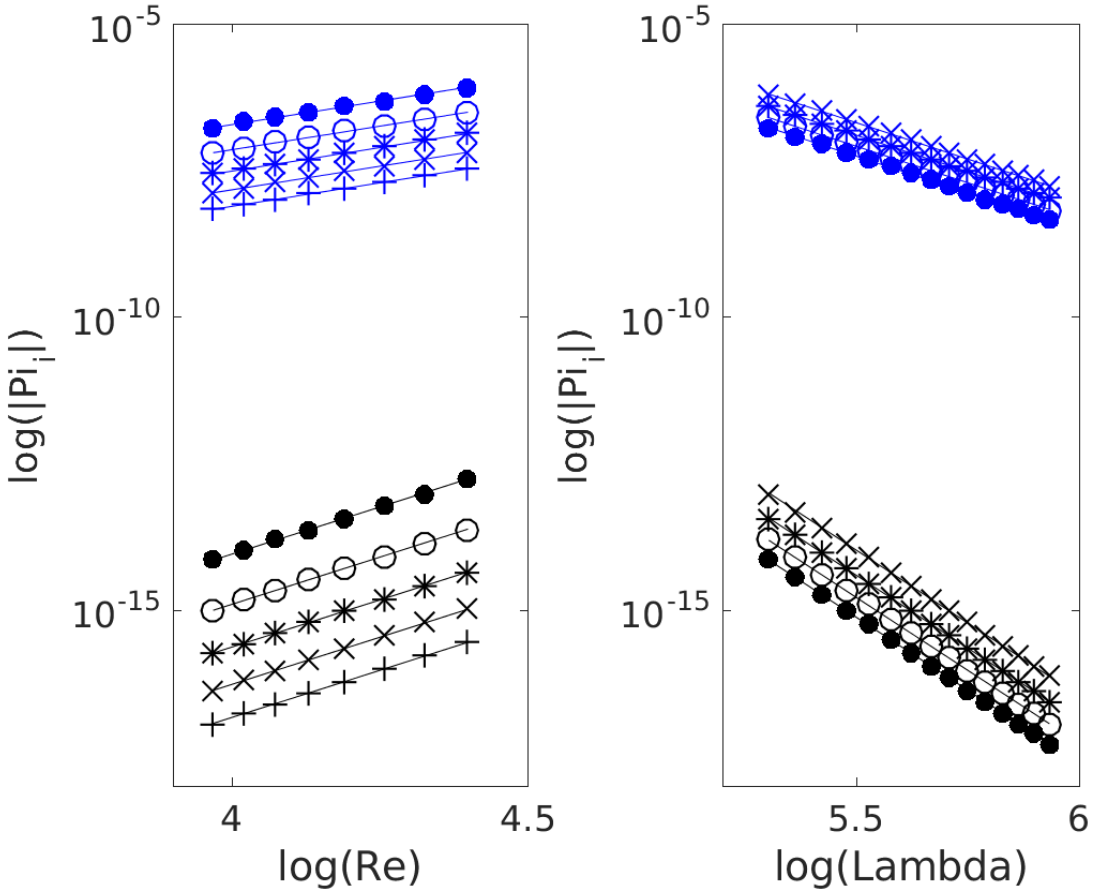

Figure 4.3: Log-log plots of the absolute values of the nondimensionalized renormalization coefficients Π_i. Computed by minimizing (4.36) using data from time $[0, 10]$ for a variety of values of dispersive Reynolds numbers Re and resolutions Λ with an initial condition $v_0(x) = \sin(x)$. (Left) $\log(|\Pi_2|)$ (blue, upper) and $\log(|\Pi_4|)$ (black, lower) plotted against $\log(Re)$ for Λ corresponding to N equal to 32 (dots), 38 (circles), 44 (stars), 50 (crosses), and 56 (pluses). (Right) $\log(|\Pi_2|)$ (blue, upper) and $\log(|\Pi_4|)$ (black, lower) plotted against $\log(\Lambda)$ for Reynolds numbers corresponding to ϵ equal to 0.1 (dots), 0.09 (circles), 0.08 (stars), and 0.07 (crosses). The linear relationships suggest power law behavior in both Re and Λ, which implies power law behavior in ϵ and N of the dimensional formulation. We present only the results for $i = 2, 4$ because simulations indicate the coefficients vanish for $i = 1, 3$.

have uncovered here. This is a subject of future investigation by the authors.

The non-dimensional renormalization coefficients Π_i in (4.38) exhibit incomplete similarity in *both* parameters Re and Λ. If we keep the resolution of the reduced model fixed (Λ fixed), then $\Pi_i \to \infty$ as $Re \to \infty$. This is a hallmark of a singular perturbation problem which is to be expected since for $Re = \infty$ we recover the Burgers equation which develops shocks in finite time, while KdV does not. We will see in Section 4.5 that the renormalized ROMs for Burgers are, as one would expect, markedly different from those of KdV. Similarly, if we keep Re fixed, and let the resolution of the reduced model $\Lambda \to \infty$ then $\Pi_i \to 0$ which is also expected since in this limit there is no need for a memory term.

We used a similar approach to find different renormalization coefficients $\alpha_i'(t)$ for a ROM that includes only the first two terms of the complete memory approximation:

$$\frac{d\hat{u}_k}{dt} = R_k^0(\hat{\mathbf{u}}) + \sum_{i=1}^{2} \alpha_i'(t) t^i R_k^i(\hat{\mathbf{u}}).$$

Similarly, we assume the time dependence of the coefficients is $\alpha_i'(t) = \beta_i' t^{-i}$ for constants β_i'. Let Π_i' be the non-dimensionalized form of β_i'. It also obeys a power law formula:

$$\Pi_i' = d_i Re^{e_i} \Lambda_1^{f_i}, \tag{4.39}$$

where $d_1 = 0$ (so the t-model again is not included), and $d_2 = -0.7923$, $e_2 = 3.7832$, and $f_2 = -5.8254$. Observe that the scaling coefficients for Λ_1 and Re are very similar to the results for the second order term in the fourth order model, but that the prefactor is different (we will call reduced order models with this form and renormalization coefficients the second order complete renormalized model). Once again, the fact that the t-model is unimportant for capturing the memory effects of the KdV equation is noteworthy, as is the scaling structure inherent in the R_k^2 coefficient.

Finally, the reader might be puzzled by the fact that the R_k^2 coefficient is negative in both cases, since the rate of change of mass entering and exiting the resolved modes due

to the non-renormalized R_k^2 was observed to be approximately proportional (with a positive constant of proportionality) to the associated exact quantity (Figure 4.1). The negative coefficient means the net rate of change of mass in and out of the resolved modes is exactly opposite the desired result. However, the coefficient was not fit to the net rate of change of mass in the resolved modes, but the rate of change of mass in each individual mode (see (4.36)). The results, as will be seen, are accurate in spite of this inconsistency. This curious phenomenon is also the subject of further investigation.

4.3.2 Results

Because a fully resolved solution is possible for sufficiently large values of ϵ, we are able to test these new renormalized reduced order models for both stability and accuracy by comparing them against a numerically accurate solution. A natural comparison for our ROMs is the "average dynamics" given by the Markov term alone, disregarding memory terms. All simulations are conducted using a time step of $\Delta t = 0.001$ using an implicit-explicit integration scheme that has proven effective for the KdV equation with periodic boundary conditions [32, 58].

There are several metrics for identifying the accuracy of a reduced order model of this kind. First, we have seen from the exact solution that for a given subset of modes F in the exact solution, mass should flow both in and out as time passes. This corresponds to mass passing from the resolved modes to the unresolved modes and back. A ROM for the same subset of modes should accurately capture this net in and out flow of mass. The Markov term alone conserves mass in the resolved modes, and so is unable capture this effect at all. An example of this dynamic mass loss and gain is depicted in Figure 4.4. Observe that the second order ROM drains mass when the exact system should be gaining it, and vice versa. This is a consequence of the fact that the renormalization coefficient is negative, while when summed over the resolved modes, the R_k^2 term itself appears contribute a positive multiple of the derivative of the total mass in the resolved modes (as seen in Figure 4.1). The fourth order ROM, on the other hand, drains too much mass in the initial three units of time, but

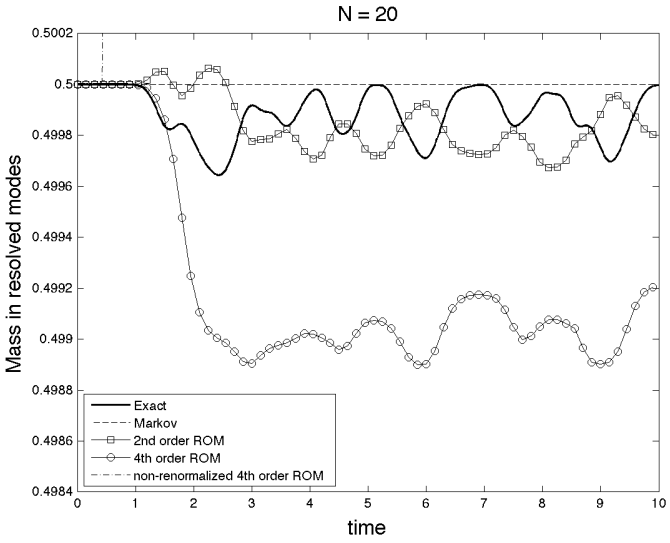

Figure 4.4: The mass in the first $N = 20$ positive Fourier modes for an initial condition of $\sin(x)$ according to several models. The solid curve depicts the exact solution, found by running a simulation with $N = 256$ positive modes. The dashed curve depicts the Markov model with $N = 20$ modes. The solid curve with squares depicts the renormalized second order complete ROM, while the solid curve with circles depicts the renormalized fourth order complete memory ROM. The dash-dotted curve is a non-renormalized version of the 4th order ROM, which is unstable.

then gains and drains mass consistently with the exact rate. The addition of the fourth order term seems to improve the model's ability to capture mass transfer in and out of the resolved modes. Note, finally, how little the mass in the resolved modes actually changes. In the $N = 20$ case, less than 0.08% ever leaves the resolved modes. It is remarkable that this little amount of mass flow in and out of the resolved variables carries with it the rich structure for the renormalized coefficients presented in Section 4.3.1. For this reason, intuition would suggest that the memory is not important, and that the Markov model will be sufficient. This proves to not be the case.

The need for the inclusion of memory can be seen by inspecting our solutions in real space. Figure 4.5 depicts the solution for the first $N = 20$ modes as depicted in real space

at a variety of times. The initial condition leads to a near-shock, but the dispersive term resists the creation, leading to the generation of a number of solitons which then interact with one another for the remaining time. After sufficient time, the Markov model is out of phase with the exact solution, while the fourth order ROM tracks it well for long times.

We compute the L_2 norm of the difference between the exact real space solution and that predicted by the ROM. We divide this error by the L_2 norm of the exact solution, producing a global relative error between the exact and approximated solution trajectories. Here, it is reasonable to also compute the error of the Markov approximation for comparison. Figure 4.6 depicts the global relative error at time $t = 100$ for ROMs of several different resolutions with $\epsilon = 0.1$ (the error results for other values of ϵ are qualitatively similar). Recall that the formulae for the renormalization coefficients were fit only with data between $t \in [0, 10]$, so it appears that these coefficients are valid for a long time.

The accuracy of complete ROMs is not achieved until N is sufficiently large. For $\epsilon = 0.1$ this is around $N = 20$. For $\epsilon = 0.09$, it is approximately $N = 24$. Qualitatively, our results suggest that a stable and accurate ROM can only be constructed for sets of resolved modes whose "full models" comprise at least half of the modes needed for a fully resolved simulation. Because KdV is a dispersive problem, the total mass is conserved. Therefore, a reduced order model can only hope to capture the dynamic flow of a fixed amount of mass if the full model upon which it is based contains a large proportion of the modes through which the mass must flow.

As $\epsilon \to 0$, the initial draining of mass passes to higher frequencies before beginning to rebound. When $\epsilon = 0$, the problem becomes Burgers' equation, which has a finite time singularity, and the mass (in the Burgers literature called energy) cascades to higher frequencies indefinitely. For this case, a renormalized model based upon the BCH approximation in which the coefficients $\alpha_i(t)$ are actually *constant* has been seen to perform very well [104]. We will revisit this in Section 4.5. As $\epsilon \to 0$, the renormalization coefficients as computed by our formulae (4.38) and (4.39) grow to infinity. This suggests that, as ϵ becomes small, the assumptions made in deriving our complete renormalized ROMs fail to hold. As mentioned

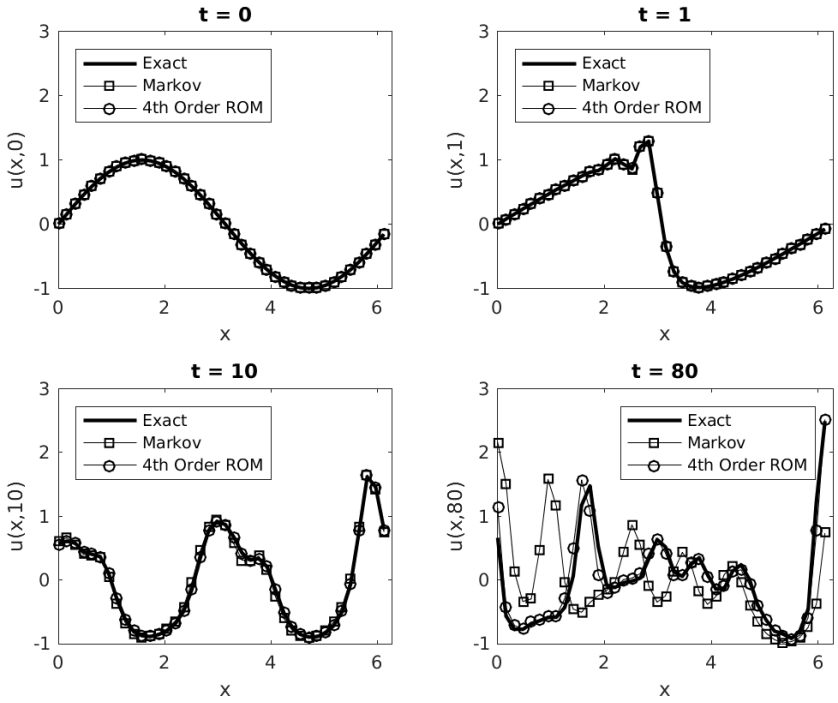

Figure 4.5: The real space representation of the solution to KdV through $N = 20$ positive modes with $\epsilon = 0.1$ and initial condition $v^0(x) = \sin(x)$ according to several models. The initial condition is depicted in the upper left. At the upper right, there is steepening of the gradient due to the nonlinearity in the equation. In the lower left, the sharp gradient has been converted into a number of interacting solitons. In the bottom right, the interactions of the solitons have caused the Markov model to lose phase with the exact solution, while the accuracy of our reduced order model persists.

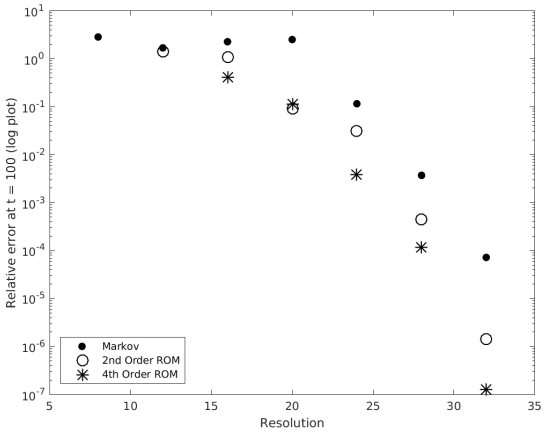

Figure 4.6: The relative global error at time $t = 100$ for several models plotted on a logarithmic axis with $\epsilon = 0.1$ and initial condition $v^0(x) = \sin(x)$. The error is computed as the ratio between the L_2 norm of the real space error and the L_2 norm of the exact real space solution. All three ROMs improve as the number of resolved modes increases, but the complete memory second order and fourth order ROMs are consistently more accurate than the Markov model alone.

before, this is an indication of the singular nature of the perturbation problem as $\epsilon \to 0$.

In deriving the governing parameters of the power laws (4.38) and (4.39), we used data for $\epsilon = 0.1, 0.095, \ldots 0.065$. Using the power laws we recovered, we conducted a simulation for $\epsilon = 0.01$, depicted in Figure 4.7. Clearly, the ROM which keeps up to a 4th order memory term is much more accurate than the Markov ROM. This signifies that the power laws can be extrapolated beyond the parameter regimes over which they were derived. This is a major objective of any model reduction methodology because it allows the construction of accurate ROMs for parameter regimes where a brute force calculation could be infeasible.

Finally, we must comment on the computational cost of these reduced order models. As observed, stable and accurate ROMs can only be constructed for resolutions that are approximately four times smaller than a fully resolved solution. The FFTs employed in an ROM are large enough to contain the "full model," which is twice as large as the ROM itself. Thus, in the best case scenario, we will be using FFTs that are half as large as those

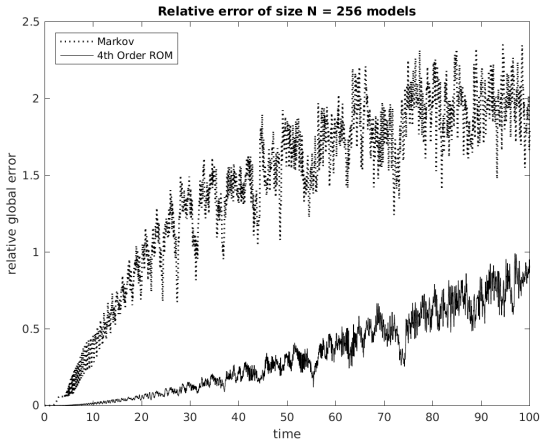

Figure 4.7: Relative global error over time of several reduced models of size $N = 256$ and $\epsilon = 0.01$. The exact solution is taken to be the behavior of the first $N = 256$ modes in a full simulation of size $N = 512$. Simulations were conducted with $\Delta t = 10^{-4}$. The Markov model (dashed curve) becomes inaccurate within the first twenty time units. The error of the fourth order ROM (solid curve), has grown to only 75% after 100 time units.

needed for the full solution. The second order ROM uses six FFTs and IFFTs per timestep, while the fourth order ROM requires 22. A fully resolved simulation, on the other hand only uses one FFT and IFFT per timestep, though it is twice as large. Consequently, the ROMs produced in this paper are necessarily less efficient than simply solving the problem outright. Their value is not found in producing efficient computational schemes, but in shedding light on the role of memory in dispersive problems.

4.3.3 Discussion

We have applied our complete memory approximation to the construction of ROMs for the Korteweg-de Vries equation. The functional forms of these ROMs are complex, and would be unlikely to be discovered by classical mathematical modeling techniques. Instead, these terms are derived from the dynamics themselves. The resulting ROMs are numerically unstable, unless each term is tamed with specific renormalization coefficients. We find that

these renormalization coefficients decay algebraically in time, canceling the time dependence originally found in the Taylor expansions. This differs from previous efforts to renormalize ROMs derived from the Mori-Zwanzig formalism, which became stable with renormalization coefficients that were constant in time [104, 105].

The renormalization coefficients that most effectively capture the correct dynamics of the resolved modes also suggest significant structure in the memory. The odd terms in the memory expansion seem to be unimportant for capturing the dynamics of the memory. This is at odds with previous results for Burgers' equation, for which the odd terms seemed to be more important [105]. It is possible that the even terms in the memory expansion account for the memory of dispersive terms, while the odd terms account for the memory of dissipative terms. To clarify the situation, the authors plan to apply it to dissipative problems (such as Burgers' equation) and problems with dissipation and dispersion (such as the KdV-Burgers equation). For the case of KdV-Burgers equation, averaging and renormalization for traveling waves was considered in [22].

The structure of the memory is also evident in the fact that the coefficients of the even terms in the memory expansion obey strict power laws in ϵ and N, with the power law dependence for the R_k^4 term being approximately twice as large as that of the R_k^2 term. Due to this structure, we can construct ROMs for any ϵ and N without needing to know the exact solution for that specific ϵ. The highly structured nature of the renormalization coefficients suggest that the complete memory approximation is, in some manner, a natural way of describing the dynamics of a subset of modes in this problem.

We found qualitatively that accurate ROMs were only possible when the "full model" on which the reduced model is based is nearly fully resolved itself. This was not the case for dissipative problems, and we hypothesize that it is the dispersive nature of the problem that necessitates large "full models." Once accurate ROMs can be constructed, however, they prove to be much more effective than the Markov term alone. This is noteworthy, because the amount of mass leaving the resolved modes is exceptionally small. This indicates that, even when the memory term is extremely small, ignoring it will lead to inaccurate results,

especially during long time simulations. This is a hallmark of a singular perturbation problem (in our case the perturbation is the small magnitude of the dispersion coefficient ϵ). Also, this agrees with the discovered incomplete similarity exhibited by the renormalized coefficients [7].

The results of the complete memory approximation and its renormalization are promising and important. First, they support the need for memory terms in reduced models. Second, the suggested distinction of different order MZ memory terms as relevant for different physical mechanisms points towards a systematic classification of the memory. This realization supports the advantage of model reduction starting from an exact formalism like MZ over adding terms by hand. Finally, the success of renormalization in producing stable and accurate models for long times indicates an elegant and effective way of incorporating dynamic information about the scales we aim to resolve.

4.4 Application 2: The three-dimensional Euler's equations

The work in this section is also available in the arXiv at [89]. We intend to submit it for publication in June 2018.

4.4.1 Introduction

The behavior of solutions of the 3D Euler equations for incompressible fluid flow is one of the most challenging problems in the analysis of PDEs and scientific computing. The common root of these difficulties lies in the complexity of the dynamics implied by the equations. On the theoretical side, it is unknown whether general smooth initial conditions give rise to a finite time singularity. On the numerical side, computations that begin with smooth initial conditions quickly give rise to a degree of complexity (turbulence) that exhausts the available computational power. Despite the accumulated knowledge on the theoretical and numerical fronts [3, 6, 21, 26–28, 30, 36, 43, 51, 53, 66, 68, 71, 73, 75, 82, 97], there is still no conclusive evidence. A limiting factor in the numerical exploration of this problem is the necessary system size for fully resolved computations. Consequently, the development of reduced order models for Euler's equations would greatly help in this effort.

The MZ formalism has been previously used to develop reduced order models for Euler's equations [46, 99, 104]. Those models were based on approximations of the memory term which assume various degrees of "long memory" i.e., the assumption that the unresolved variables evolve on timescales that are comparable to the resolved variables. Such an assumption is eminently plausible for the Euler equations (and high-Reynolds number fluid flows in general), given the vast range of active scales present in the solution. The complete memory approximation also assumes long memory but avoids all the simplifying approximations applied before. Thus, it can incorporate all the complex effects present in the Euler dynamics. In this section we present results for renormalized MZ reduced models of the 3D Euler equations stemming from the complete memory approximation. The renormalized reduced models are stable and we evolve them for long times. We use the predicted evolution to estimate several quantities related to the occurrence of a finite-time singularity (blow-up). We find that our results are indeed consistent with a finite-time singularity.

4.4.2 Problem statement

The three dimensional Euler's equations are given by:

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p, \quad \nabla \cdot \mathbf{u} = 0 \tag{4.40}$$

where $\mathbf{u}(x,t)$ is a three-dimensional vector that is periodic in all three dimensions and p is the pressure. Let the initial condition be $\mathbf{u}(x,0) = \mathbf{v}^0$. We will use the Taylor-Green initial condition:

$$\mathbf{v}^0 = \begin{bmatrix} \sin(x)\cos(y)\cos(z) \\ -\cos(x)\sin(y)\cos(z) \\ 0 \end{bmatrix}. \tag{4.41}$$

The Taylor-Green initial condition is very smooth. We are interested in studying the cascade of energy into higher frequency modes as time evolves. We will write (4.40) in vector form in order to keep track of terms more carefully. Let u^x be the component of u in the x dimension

(and similarly for y and z):

$$\begin{bmatrix} \frac{\partial u^x}{\partial t} \\ \frac{\partial u^y}{\partial t} \\ \frac{\partial u^z}{\partial t} \end{bmatrix} + \begin{bmatrix} u^x \frac{\partial u^x}{\partial x} + u^y \frac{\partial u^x}{\partial y} + u^z \frac{\partial u^x}{\partial z} \\ u^x \frac{\partial u^y}{\partial x} + u^y \frac{\partial u^y}{\partial y} + u^z \frac{\partial u^y}{\partial z} \\ u^x \frac{\partial u^z}{\partial x} + u^y \frac{\partial u^z}{\partial y} + u^z \frac{\partial u^z}{\partial z} \end{bmatrix} = - \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{bmatrix}. \tag{4.42}$$

In order to specify the pressure p, we will apply $\nabla\cdot$ to both sides of the equation and use $\nabla \cdot \mathbf{u} = 0$.

$$\nabla \cdot \begin{bmatrix} \frac{\partial u^x}{\partial t} \\ \frac{\partial u^y}{\partial t} \\ \frac{\partial u^z}{\partial t} \end{bmatrix} + \nabla \cdot \begin{bmatrix} u^x \frac{\partial u^x}{\partial x} + u^y \frac{\partial u^x}{\partial y} + u^z \frac{\partial u^x}{\partial z} \\ u^x \frac{\partial u^y}{\partial x} + u^y \frac{\partial u^y}{\partial y} + u^z \frac{\partial u^y}{\partial z} \\ u^x \frac{\partial u^z}{\partial x} + u^y \frac{\partial u^z}{\partial y} + u^z \frac{\partial u^z}{\partial z} \end{bmatrix} = -\nabla \cdot \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{bmatrix}$$

$$\begin{aligned} &\frac{\partial u^x}{\partial x}\frac{\partial u^x}{\partial x} + u^x \frac{\partial^2 u^x}{\partial x^2} + \frac{\partial u^y}{\partial x}\frac{\partial u^x}{\partial y} + u^y \frac{\partial^2 u^x}{\partial x \partial y} + \frac{\partial u^z}{\partial x}\frac{\partial u^x}{\partial z} + u^z \frac{\partial^2 u^x}{\partial x \partial z} \\ &+ \frac{\partial u^x}{\partial y}\frac{\partial u^y}{\partial x} + u^x \frac{\partial^2 u^y}{\partial x \partial y} + \frac{\partial u^y}{\partial y}\frac{\partial u^y}{\partial y} + u^y \frac{\partial^2 u^y}{\partial y^2} + \frac{\partial u^z}{\partial y}\frac{\partial u^y}{\partial z} + u^z \frac{\partial^2 u^y}{\partial y \partial z} \\ &+ \frac{\partial u^x}{\partial z}\frac{\partial u^z}{\partial x} + u^z \frac{\partial^2 u^z}{\partial x \partial z} + \frac{\partial u^y}{\partial z}\frac{\partial u^z}{\partial y} + u^y \frac{\partial^2 u^z}{\partial y \partial z} + \frac{\partial u^z}{\partial z}\frac{\partial u^z}{\partial z} + u^z \frac{\partial^2 u^z}{\partial^2 z} = -\left[\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial x^y 2} + \frac{\partial^2 p}{\partial z^2}\right]. \end{aligned}$$

Note that:

$$u^x \frac{\partial^2 u^x}{\partial x^2} + u^x \frac{\partial^2 u^y}{\partial x \partial y} + u^x \frac{\partial^2 u^z}{\partial x \partial z} = u^x \frac{\partial}{\partial x}\left[\nabla \cdot \mathbf{u}\right] = 0$$

and similary for the $u^y \frac{\partial}{\partial y}$ and $u^z \frac{\partial}{\partial z}$ terms. Thus:

$$\begin{aligned} &\frac{\partial u^x}{\partial x}\frac{\partial u^x}{\partial x} + \frac{\partial u^y}{\partial x}\frac{\partial u^x}{\partial y} + \frac{\partial u^z}{\partial x}\frac{\partial u^x}{\partial z} + \frac{\partial u^x}{\partial y}\frac{\partial u^y}{\partial x} + \frac{\partial u^y}{\partial y}\frac{\partial u^y}{\partial y} + \frac{\partial u^z}{\partial y}\frac{\partial u^y}{\partial z} \\ &\qquad + \frac{\partial u^x}{\partial z}\frac{\partial u^z}{\partial x} + \frac{\partial u^y}{\partial z}\frac{\partial u^z}{\partial y} + \frac{\partial u^z}{\partial z}\frac{\partial u^z}{\partial z} = -\left[\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial x^y 2} + \frac{\partial^2 p}{\partial z^2}\right] \end{aligned}$$

Because $\mathbf{u}$ is periodic in all three dimensions, we will write it as Fourier series:

$$\mathbf{u}(x,t) = \sum_{\mathbf{k}} \mathbf{u}_{\mathbf{k}}(t) e^{i\mathbf{k}\cdot\mathbf{x}} \tag{4.43}$$

where $\mathbf{k}$ is a three-dimensional wavevector, and the sum is over all possible integer-valued wavevectors. Also let $p = \sum_{\mathbf{k}} p_{\mathbf{k}}(t) e^{i\mathbf{k}\cdot\mathbf{x}}$. We will be able to find a compact expression for $p_{\mathbf{k}}$. Consider first the right hand side:

$$-\left[\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial^2 y} + \frac{\partial^2 p}{\partial z^2}\right] = -\left[\sum_{\mathbf{k}} -k_x^2 p_{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{x}} + \sum_{\mathbf{k}} -k_y^2 p_{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{x}} + \sum_{\mathbf{k}} -k_z^2 p_{\mathbf{k}}\mathbf{k} e^{i\mathbf{k}\cdot\mathbf{x}}\right] = \sum_{\mathbf{k}} |\mathbf{k}|^2 p_{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{x}}.$$

Now, the left hand side:

$$\begin{aligned}
&\frac{\partial u^x}{\partial x}\frac{\partial u^x}{\partial x} + \frac{\partial u^y}{\partial x}\frac{\partial u^x}{\partial y} + \frac{\partial u^z}{\partial x}\frac{\partial u^x}{\partial z} + \frac{\partial u^x}{\partial y}\frac{\partial u^y}{\partial x} + \frac{\partial u^y}{\partial y}\frac{\partial u^y}{\partial y} + \frac{\partial u^z}{\partial y}\frac{\partial u^y}{\partial z} + \frac{\partial u^x}{\partial z}\frac{\partial u^z}{\partial x} + \frac{\partial u^y}{\partial z}\frac{\partial u^z}{\partial y} + \frac{\partial u^z}{\partial z}\frac{\partial u^z}{\partial z} \\
&= \left(\sum_{\mathbf{p}} i p_x u^x_{\mathbf{p}} e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_x u^x_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) + \left(\sum_{\mathbf{p}} i p_x u^y_{\mathbf{p}} e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_y u^x_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) \\
&\quad + \left(\sum_{\mathbf{p}} i p_x u^z_{\mathbf{p}} e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_z u^x_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) + \left(\sum_{\mathbf{p}} i p_y u^x_{\mathbf{p}} e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_x u^y_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) \\
&\quad + \left(\sum_{\mathbf{p}} i p_y u^y_p e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_y u^y_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) + \left(\sum_{\mathbf{p}} i p_y u^z_{\mathbf{p}} e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_z u^y_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) \\
&\quad + \left(\sum_{\mathbf{p}} i p_z u^x_{\mathbf{p}} e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_x u^z_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) + \left(\sum_{\mathbf{p}} i p_z u^y_{\mathbf{p}} e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_y u^z_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) \\
&\quad + \left(\sum_{\mathbf{p}} i p_z u^z_{\mathbf{p}} e^{i\mathbf{p}\cdot\mathbf{x}}\right)\left(\sum_{\mathbf{q}} i q_z u^z_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{x}}\right) \\
&= -\sum_{\mathbf{k}} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \big[p_x q_x u^x_{\mathbf{p}} u^x_{\mathbf{q}} + p_x q_y u^y_{\mathbf{p}} u^x_{\mathbf{q}} + p_x q_z u^z_{\mathbf{p}} u^x_{\mathbf{q}} + p_y q_x u^x_{\mathbf{p}} u^y_{\mathbf{q}} + p_y q_y u^y_{\mathbf{p}} u^y_{\mathbf{q}} + p_y q_z u^z_{\mathbf{p}} u^y_{\mathbf{q}} \\
&\qquad\qquad + p_z q_x u^x_{\mathbf{p}} u^z_{\mathbf{q}} + p_z q_y u^y_{\mathbf{p}} u^z_{\mathbf{q}} + p_z q_z u^z_{\mathbf{p}} u^z_{\mathbf{q}}\big] e^{i\mathbf{k}\cdot\mathbf{x}} \\
&= -\sum \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{p}\cdot\mathbf{u}_{\mathbf{q}})(\mathbf{q}\cdot\mathbf{u}_{\mathbf{p}}) e^{i\mathbf{k}\cdot\mathbf{x}}.
\end{aligned}$$

Now observe that:

$$\nabla \cdot \mathbf{u} = 0 \implies 0 = \sum_{\mathbf{k}} [ik_x u^x_{\mathbf{k}} + ik_y u^y_{\mathbf{k}} + ik_z u^z_{\mathbf{k}}] e^{i\mathbf{k}\cdot\mathbf{x}}$$
$$0 = i \sum_{\mathbf{k}} (\mathbf{k} \cdot \mathbf{u_k}) e^{i\mathbf{k}\cdot\mathbf{x}}$$

Multiplying both sides by $e^{-i\mathbf{j}\cdot\mathbf{x}}$ and integrating yields:

$$0 = i(\mathbf{j} \cdot \mathbf{u_j})$$

so we know that $\mathbf{k} \cdot \mathbf{u_k} = 0$. Thus,

$$\mathbf{p} \cdot \mathbf{u_q} = \mathbf{p} \cdot \mathbf{u_q} + 0 = \mathbf{p} \cdot \mathbf{u_q} + \mathbf{q} \cdot \mathbf{u_q} = \mathbf{k} \cdot \mathbf{u_q},$$

and similarly for the other dot product. Thus, the left hand side we had been simplifying is:

$$-\sum_{\mathbf{k}} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{k} \cdot \mathbf{u_q})(\mathbf{k} \cdot \mathbf{u_p}) e^{i\mathbf{k}\cdot\mathbf{x}}$$

Multiplying the left and right hand sides by $e^{-i\mathbf{j}\cdot\mathbf{x}}$ leaves:

$$|\mathbf{j}|^2 p_{\mathbf{j}} = -\sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{j} \cdot \mathbf{u_q})(\mathbf{j} \cdot \mathbf{u_p}) \implies p_{\mathbf{j}} = -\frac{1}{|\mathbf{j}|^2} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{j} \cdot \mathbf{u_q})(\mathbf{j} \cdot \mathbf{u_p}).$$

We now return to the original equation, knowing we can plug this in for $p_{\mathbf{k}}$ later on. We write each term as a Fourier sum:

$$\begin{bmatrix} \frac{\partial u^x}{\partial t} \\ \frac{\partial u^y}{\partial t} \\ \frac{\partial u^z}{\partial t} \end{bmatrix} = \sum_{\mathbf{k}} \frac{d\mathbf{u_k}}{dt} e^{i\mathbf{k}\cdot\mathbf{x}}$$

$$\begin{bmatrix} u^x \frac{\partial u^x}{\partial x} + u^y \frac{\partial u^x}{\partial y} + u^z \frac{\partial u^x}{\partial z} \\ u^x \frac{\partial u^y}{\partial x} + u^y \frac{\partial u^y}{\partial y} + u^z \frac{\partial u^y}{\partial z} \\ u^x \frac{\partial u^z}{\partial x} + u^y \frac{\partial u^z}{\partial y} + u^z \frac{\partial u^z}{\partial z} \end{bmatrix} = \sum_{\mathbf{k}} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \begin{bmatrix} iq_x u^x_{\mathbf{p}} u^x_{\mathbf{q}} + iq_y u^y_{\mathbf{p}} u^x_{\mathbf{q}} + iq_z u^z_{\mathbf{p}} u^x_{\mathbf{q}} \\ iq_x u^x_{\mathbf{p}} u^y_{\mathbf{q}} + iq_y u^y_{\mathbf{p}} u^y_{\mathbf{q}} + iq_z u^z_{\mathbf{p}} u^y_{\mathbf{q}} \\ iq_x u^x_{\mathbf{p}} u^z_{\mathbf{q}} + iq_y u^y_{\mathbf{p}} u^z_{\mathbf{q}} + iq_z u^z_{\mathbf{p}} u^z_{\mathbf{q}} \end{bmatrix} e^{i\mathbf{k}\cdot\mathbf{x}}$$

$$= i \sum_{\mathbf{k}} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{q} \cdot \mathbf{u}_{\mathbf{p}}) \mathbf{u}_{\mathbf{q}} e^{i\mathbf{k}\cdot\mathbf{x}}$$

$$= i \sum_{\mathbf{k}} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{k} \cdot \mathbf{u}_{\mathbf{p}}) \mathbf{u}_{\mathbf{q}} e^{i\mathbf{k}\cdot\mathbf{x}}$$

$$-\begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{bmatrix} = -\sum_{\mathbf{k}} \begin{bmatrix} ik_x p_{\mathbf{k}} \\ ik_y p_{\mathbf{k}} \\ ik_z p_{\mathbf{k}} \end{bmatrix} e^{i\mathbf{k}\cdot\mathbf{x}} = -i \sum_{\mathbf{k}} \mathbf{k} p_{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{x}} = i \sum_{\mathbf{k}} \frac{\mathbf{k}}{|\mathbf{k}|^2} \left(\sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{k} \cdot \mathbf{u}_{\mathbf{p}})(\mathbf{k} \cdot \mathbf{u}_{\mathbf{q}}) \right) e^{i\mathbf{k}\cdot\mathbf{x}}$$

Multiplying each term by $e^{-i\mathbf{j}\cdot\mathbf{x}}$ and integrating yields this equation for the evolution of a Fourier mode:

$$\frac{d\mathbf{u}_k}{dt} = -i \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{k} \cdot \mathbf{u}_{\mathbf{p}}) \mathbf{u}_{\mathbf{q}} + \frac{\mathbf{k}}{|\mathbf{k}|^2} (\mathbf{k} \cdot \mathbf{u}_{\mathbf{p}})(\mathbf{k} \cdot \mathbf{u}_{\mathbf{q}}) \tag{4.44}$$

Written in a more compact form,

$$\frac{d\mathbf{u}_{\mathbf{k}}}{dt} = -i \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \mathbf{k} \cdot \mathbf{u}_{\mathbf{p}} A_{\mathbf{k}} \mathbf{u}_{\mathbf{q}} \tag{4.45}$$

where

$$A_k = I - \frac{\mathbf{k}\mathbf{k}^T}{|\mathbf{k}|^2}. \tag{4.46}$$

4.4.3 Implementation

Now we should think about how to implement this computationally. For a given state vector $\mathbf{u}$, the derivative of element $\mathbf{k}$ is given by (4.45). We would like to evaluate the right hand side efficiently. We first consider, in real space, the quantity $\mathbf{u}\mathbf{u}^T$:

$$\mathbf{u}\mathbf{u}^T = \begin{bmatrix} u^x u^x & u^x u^y & u^x u^z \\ u^y u^x & u^y u^y & u^y u^z \\ u^z u^x & u^z u^y & u^z u^z \end{bmatrix} = \sum_{\mathbf{k}} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \begin{bmatrix} u^x_{\mathbf{p}} u^x_{\mathbf{q}} & u^y_{\mathbf{p}} u^x_{\mathbf{q}} & u^z_{\mathbf{p}} u^x_{\mathbf{q}} \\ u^x_{\mathbf{p}} u^y_{\mathbf{q}} & u^y_{\mathbf{p}} u^y_{\mathbf{q}} & u^z_{\mathbf{p}} u^y_{\mathbf{q}} \\ u^x_{\mathbf{p}} u^z_{\mathbf{q}} & u^y_{\mathbf{p}} u^z_{\mathbf{q}} & u^z_{\mathbf{p}} u^z_{\mathbf{q}} \end{bmatrix} e^{i\mathbf{k}\cdot\mathbf{x}}.$$

The fast Fourier transform of this is found by multiplying by $e^{-i\mathbf{j}\cdot\mathbf{x}}$ and integrating. Thus,

$$[FFT(\mathbf{u}\mathbf{u}^T)]_{\mathbf{k}} = \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \begin{bmatrix} u^x_{\mathbf{p}} u^x_{\mathbf{q}} & u^y_{\mathbf{p}} u^x_{\mathbf{q}} & u^z_{\mathbf{p}} u^x_{\mathbf{q}} \\ u^x_{\mathbf{p}} u^y_{\mathbf{q}} & u^y_{\mathbf{p}} u^y_{\mathbf{q}} & u^z_{\mathbf{p}} u^y_{\mathbf{q}} \\ u^x_{\mathbf{p}} u^z_{\mathbf{q}} & u^y_{\mathbf{p}} u^z_{\mathbf{q}} & u^z_{\mathbf{p}} u^z_{\mathbf{q}} \end{bmatrix}.$$

Observe the two terms in the right hand side of (4.45). First is,

$$-i \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{k}\cdot\mathbf{u}_{\mathbf{p}})\mathbf{u}_{\mathbf{q}} = -i \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \mathbf{u}_{\mathbf{q}}\mathbf{u}_{\mathbf{p}}^T\mathbf{k} = -i \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \begin{bmatrix} u^x_{\mathbf{p}} u^x_{\mathbf{q}} & u^y_{\mathbf{p}} u^x_{\mathbf{q}} & u^z_{\mathbf{p}} u^x_{\mathbf{q}} \\ u^x_{\mathbf{p}} u^y_{\mathbf{q}} & u^y_{\mathbf{p}} u^y_{\mathbf{q}} & u^z_{\mathbf{p}} u^y_{\mathbf{q}} \\ u^x_{\mathbf{p}} u^z_{\mathbf{q}} & u^y_{\mathbf{p}} u^z_{\mathbf{q}} & u^z_{\mathbf{p}} u^z_{\mathbf{q}} \end{bmatrix} \mathbf{k} = -i[FFT(\mathbf{u}\mathbf{u}^T)]_{\mathbf{k}}\mathbf{k}.$$

Thus, it can be computed easily by transforming into real space, multiplying the state vectors, and returning to Fourier space. Similarly, the second term is:

$$\begin{aligned} i \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \mathbf{k}\cdot\mathbf{u}_{\mathbf{p}} \frac{\mathbf{k}\mathbf{k}^T}{|\mathbf{k}|^2}\mathbf{u}_{\mathbf{q}} &= \frac{i\mathbf{k}}{|\mathbf{k}|^2} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{k}\cdot\mathbf{u}_{\mathbf{p}})(\mathbf{k}\cdot\mathbf{u}_{\mathbf{q}}) \\ &= \frac{i\mathbf{k}}{|\mathbf{k}|^2} \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} \Big[k_x k_x u^x_{\mathbf{p}} u^x_{\mathbf{q}} + k_x k_y u^y_{\mathbf{p}} u^x_{\mathbf{q}} + k_x k_z u^z_{\mathbf{p}} u^x_{\mathbf{q}} \\ &\qquad\qquad k_y k_x u^x_{\mathbf{p}} u^y_{\mathbf{q}} + k_y k_y u^y_{\mathbf{p}} u^y_{\mathbf{q}} + k_y k_z u^z_{\mathbf{p}} u^y_{\mathbf{q}} \\ &\qquad\qquad k_z k_x u^x_{\mathbf{p}} u^z_{\mathbf{q}} + k_z k_y u^y_{\mathbf{p}} u^z_{\mathbf{q}} + k_z k_z u^z_{\mathbf{p}} u^z_{\mathbf{q}} \Big] \\ &= \frac{i\mathbf{k}}{|\mathbf{k}|^2} \left(\mathbf{k}^T [FFT(\mathbf{u}\mathbf{u}^T)]_{\mathbf{k}}\mathbf{k} \right). \end{aligned}$$

Thus, we can also compute the other part of the right hand side using the same real space calculation. For the remainder of this section, we used Matlab's built-in integrator `ode45` to solve the described differential equations. This uses a version of Runge-Kutta-Fehlberg with adaptive stepsize selection. We set the initial step as 10^{-3} because the very small initial rate of change causes the algorithm to choose an overly ambitious starting step. We also set the maximum relative error to 10^{-10}.

4.4.4 Reduced models

Unlike the KdV case, the reduced models for Euler are simpler to derive. This is largely due to the fact that the RHS has only one term, so the models do not telescope out in a challenging fashion. We do have to note that the convolution is no longer symmetric. With the benefit of hindsight, we can define our functions ahead of time to make our life easier when applying repeated differential operators.

Let $\mathbf{u}(t)$ be the solution to Euler's equations. Consider the Fourier components $\mathbf{u_k}(t)$, where $\mathbf{k} \in F \cup G$. Let F be the set of resolved modes. That is $F = \{\mathbf{k} \in [-N, N-1]^3\}$. Let $F \cup G = \{\mathbf{k} \in [-M, M-1]^3\}$. Define $\hat{\mathbf{u}} = \{\mathbf{u_k} \mid \mathbf{k} \in F\}$ and $\tilde{\mathbf{u}} = \{\mathbf{u_k} \mid \mathbf{k} \in G\}$.

Following the Mori-Zwanzig formalism, we define

$$\mathcal{L} = \sum_{\mathbf{k} \in F \cup G} \begin{bmatrix} R^x_{\mathbf{k}}(\mathbf{u}^0)\frac{\partial}{\partial u^{0,x}_{\mathbf{k}}} & 0 & 0 \\ 0 & R^y_{\mathbf{k}}(\mathbf{u}^0)\frac{\partial}{\partial u^{0,y}_{\mathbf{k}}} & 0 \\ 0 & 0 & R^z_{\mathbf{k}}(\mathbf{u}^0)\frac{\partial}{\partial u^{0,z}_{\mathbf{k}}} \end{bmatrix}$$

where $R^i_{\mathbf{k}}$ is the component of $\mathbf{R_k}$ in the ith direction and $u^{0,i}_{\mathbf{k}}$ is the component of the initial condition of $\mathbf{u_k}$ in the ith direction. Observe that:

$$\mathcal{L}\mathbf{u}^0_{\mathbf{k}} = \mathbf{R_k}(\mathbf{u}^0)$$

as desired. Therefore, we can see

$$\mathcal{L}\mathbf{u}_{\mathbf{k}}^0 = -i \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p},\mathbf{q},\mathbf{k}\in F\cup G}} \mathbf{k}\cdot\mathbf{u}_{\mathbf{p}}^0 A_{\mathbf{k}}\mathbf{u}_{\mathbf{q}}^0.$$

Let's define the convolution $\mathbf{C}$ of two vectors. Let $\mathbf{C}_{\mathbf{k}}$ be the component of $\mathbf{C}$ corresponding to the wavevector $\mathbf{k}$:

$$\mathbf{C}_{\mathbf{k}}(\mathbf{v},\mathbf{w}) = -i \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p},\mathbf{q}\in F\cup G}} \mathbf{k}\cdot\mathbf{v}_{\mathbf{p}} A_{\mathbf{k}}\mathbf{w}_{\mathbf{q}}. \tag{4.47}$$

Under this definition, we can see that $\mathcal{L}\mathbf{u}_{\mathbf{k}}^0 = \mathbf{C}_{\mathbf{k}}(\mathbf{u}^0,\mathbf{u}^0)$, or under the Mori-Zwanzig formalism that

$$\frac{d\mathbf{u}_{\mathbf{k}}}{dt} = e^{t\mathcal{L}}\mathbf{C}_{\mathbf{k}}(\mathbf{u}^0,\mathbf{u}^0) = \mathbf{C}_{\mathbf{k}}(\mathbf{u},\mathbf{u}). \tag{4.48}$$

For this to be the exact full solution, we would need $M = \infty$. For our implementation, M will necessarily be the maximal mode we choose to retain. We now define the projector P in the same manner as in Section 4.3

$$Ph(\mathbf{u}^0) = Ph(\hat{\mathbf{u}}^0,\tilde{\mathbf{u}}^0) = h(\hat{\mathbf{u}}^0,0).$$

That is, it simply sets all unresolved modes to zero. This again lets it commute with the evolution operator and cancel with the leading P that appears in all the terms of our ROMs. Let $Q = I - P$. Then we can begin constructing the terms of a complete memory approximation:

$$\frac{dP\mathbf{u}_{\mathbf{k}}}{dt} = \mathbf{R}_{\mathbf{k}}^0(\hat{\mathbf{u}}) + \sum_{i=1}^{4} \alpha_i(t) t^i \mathbf{R}_{\mathbf{k}}^i(\hat{\mathbf{u}}). \tag{4.49}$$

Markov Term

The Markov term is

$$\mathbf{R}_{\mathbf{k}}^0(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathcal{L}\mathbf{u}_{\mathbf{k}}^0.$$

Using the definitions above, we find:

$$\mathbf{R}^0_{\mathbf{k}}(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathbf{C}_{\mathbf{k}}(\mathbf{u}^0, \mathbf{u}^0) = Pe^{t\mathcal{L}}\mathbf{C}_{\mathbf{k}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) = \mathbf{C}_{\mathbf{k}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}).$$

Here, we used the fact that P sets the unresolved modes to zero. Through a slight abuse of notation, consider $\hat{\mathbf{u}}^0$ to be the array of $\mathbf{u}$ where all $\mathbf{u}_{\mathbf{j}} = 0$ for $\mathbf{j} \in G$ (and similarly for $\tilde{\mathbf{u}}$).

First-order term

The t-model is

$$\mathbf{R}^1_{\mathbf{k}}(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathcal{L}Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}}.$$

We can simplify this as:

$$\begin{aligned}\mathbf{R}^1_{\mathbf{k}}(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P\mathcal{L}Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}} = Pe^{t\mathcal{L}}P\mathcal{L}[\mathcal{L} - P\mathcal{L}]\mathbf{u}^0_{\mathbf{k}} \\ =& Pe^{t\mathcal{L}}P\mathcal{L}[\mathbf{C}_{\mathbf{k}}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{u}}^0, \hat{\mathbf{u}}^0) + \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0)]\end{aligned}$$

In order to proceed, we must investigate how $\mathcal{L}$ operates on a convolution sum:

$$\begin{aligned}\mathcal{L}\mathbf{C}_{\mathbf{k}}(\mathbf{v}, \mathbf{w}) =& \mathcal{L}\left(-i \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p},\mathbf{q}\in F\cup G}} \mathbf{k}\cdot\mathbf{v}_{\mathbf{p}} A_{\mathbf{k}}\mathbf{w}_{\mathbf{q}}\right) \\ =& -i \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p},\mathbf{q}\in F\cup G}} \mathbf{k}\cdot(\mathcal{L}\mathbf{v}_{\mathbf{p}}) A_{\mathbf{k}}\mathbf{w}_{\mathbf{q}} - i \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p},\mathbf{q}\in F\cup G}} \mathbf{k}\cdot\mathbf{v}_{\mathbf{p}} A_{\mathbf{k}}(\mathcal{L}\mathbf{w}_{\mathbf{q}}) \\ =& \mathbf{C}_{\mathbf{k}}(\mathcal{L}\mathbf{v}, \mathbf{w}) + \mathbf{C}_{\mathbf{k}}(\mathbf{v}, \mathcal{L}\mathbf{w}).\end{aligned}$$

We see it obeys the product rule. Now consider how $\mathcal{L}$ operates on $\hat{\mathbf{u}}^0$ and $\tilde{\mathbf{u}}^0$. Consider $\mathbf{p} \in F$ and $\mathbf{q} \in G$:

$$\mathcal{L}\mathbf{u}^0_{\mathbf{p}} = -i \sum_{\substack{\mathbf{r}+\mathbf{s}=\mathbf{p} \\ \mathbf{r},\mathbf{s}\in F\cup G, \mathbf{p}\in F}} \mathbf{p}\cdot\mathbf{u}^0_{\mathbf{r}} A_{\mathbf{p}} \mathbf{u}^0_{\mathbf{s}} = \mathbf{C}_{\mathbf{p}}(\mathbf{u}^0,\mathbf{u}^0)$$
$$\mathcal{L}\mathbf{u}^0_{\mathbf{q}} = -i \sum_{\substack{\mathbf{r}+\mathbf{s}=\mathbf{q} \\ \mathbf{r},\mathbf{s}\in F\cup G, \mathbf{q}\in G}} \mathbf{p}\cdot\mathbf{u}^0_{\mathbf{r}} A_{\mathbf{p}} \mathbf{u}^0_{\mathbf{s}} = \mathbf{C}_{\mathbf{q}}(\mathbf{u}^0,\mathbf{u}^0).$$

Define the following arrays. Let $\hat{\mathbf{C}}(\mathbf{v},\mathbf{w})$ be the convolution of $\mathbf{v}$ and $\mathbf{w}$, but set all indices corresponding to $\mathbf{j} \in G$ to zero. Let $\tilde{\mathbf{C}}(\mathbf{v},\mathbf{w})$ be the convolution of $\mathbf{v}$ and $\mathbf{w}$, but set all indices corresponding to $\mathbf{i} \in F$ to zero. Thus, $\mathbf{C}(\mathbf{v},\mathbf{w}) = \hat{\mathbf{C}}(\mathbf{v},\mathbf{w}) + \tilde{\mathbf{C}}(\mathbf{v},\mathbf{w})$. Observe that, under this definition (and based upon the componentwise derivations above), $\mathcal{L}\hat{\mathbf{u}}^0 = \hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)$ and $\mathcal{L}\tilde{\mathbf{u}}^0 = \tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)$. Thus,

$$\begin{aligned}\mathbf{R}^1_{\mathbf{k}}(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P[\mathbf{C}_{\mathbf{k}}(\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\tilde{\mathbf{u}}^0) + \mathbf{C}_{\mathbf{k}}(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)) \\ &+ \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\hat{\mathbf{u}}^0) + \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{u}}^0,\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)) \\ &+ \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\tilde{\mathbf{u}}^0) + \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{u}}^0,\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0))].\end{aligned}$$

In order to apply the projector, observe that $P\mathbf{C}_{\mathbf{k}}(\mathbf{v},\mathbf{w}) = \mathbf{C}_{\mathbf{k}}(P\mathbf{v},P\mathbf{w})$. Note that $P\mathbf{u}^0 = \hat{\mathbf{u}}^0$, $P\hat{\mathbf{u}}^0 = \hat{\mathbf{u}}^0$, and $P\tilde{\mathbf{u}}^0 = 0$. Finally, note that $\mathbf{C}_{\mathbf{k}}(\mathbf{v},0) = \mathbf{C}_{\mathbf{k}}(0,\mathbf{w}) = 0$. This yields:

$$\begin{aligned}\mathbf{R}^1_{\mathbf{k}}(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P[\mathbf{C}_{\mathbf{k}}(\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\tilde{\mathbf{u}}^0) + \mathbf{C}_{\mathbf{k}}(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)) \\ &+ \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\hat{\mathbf{u}}^0) + \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{u}}^0,\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)) \\ &+ \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\tilde{\mathbf{u}}^0) + \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{u}}^0,\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0))] \\ =& Pe^{t\mathcal{L}}[\mathbf{C}_{\mathbf{k}}(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0)) + \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0),\hat{\mathbf{u}}^0)] \\ =& \mathbf{C}_{\mathbf{k}}(\hat{\mathbf{u}},\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})) + \mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}),\hat{\mathbf{u}}).\end{aligned}$$

This term involves convolutions of the same two terms in both permutations. It is useful to define a function:

$$\mathbf{D}(\mathbf{v},\mathbf{w}) = \mathbf{C}(\mathbf{v},\mathbf{w}) + \mathbf{C}(\mathbf{w},\mathbf{v})$$

and the related functions $\mathbf{D_k}$, $\hat{\mathbf{D}}$, and $\tilde{\mathbf{D}}$ defined in terms of the equivalent convolutions. With this notation, we can write the t-model term in a single expression:

$$\mathbf{R}^1_{\mathbf{k}}(\hat{\mathbf{u}}) = \mathbf{D_k}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})). \tag{4.50}$$

Second-order term

The set of rules derived in the previous section will allow us to proceed to higher terms. The t^2-model is:

$$\mathbf{R}^2_{\mathbf{k}}(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathcal{L}[P\mathcal{L} - Q\mathcal{L}]Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}}.$$

First note that, with our newly defined $\mathbf{D}$ function,

$$Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}} = \mathbf{D_k}(\hat{\mathbf{u}}, \tilde{\mathbf{u}}) + \mathbf{C_k}(\tilde{\mathbf{u}}, \tilde{\mathbf{u}}).$$

We will also need to understand how $\mathcal{L}$ operates upon $\mathbf{D}$:

$$\begin{aligned}
\mathcal{L}\mathbf{D}(\mathbf{v},\mathbf{w}) =& \mathcal{L}\mathbf{C}(\mathbf{v},\mathbf{w}) + \mathcal{L}\mathbf{C}(\mathbf{w},\mathbf{v}) \\
=& \mathbf{C}(\mathcal{L}\mathbf{v},\mathbf{w}) + \mathbf{C}(\mathbf{v},\mathcal{L}\mathbf{w}) + \mathbf{C}(\mathcal{L}\mathbf{w},\mathbf{v}) + \mathbf{C}(\mathbf{w},\mathcal{L}\mathbf{v}) \\
=& \mathbf{D}(\mathcal{L}\mathbf{v},\mathbf{w}) + \mathbf{D}(\mathbf{v},\mathcal{L}\mathbf{w}).
\end{aligned}$$

Thus, it operates in the same manner it did upon $\mathbf{C}$. The projector P similarly is applied to each term within the expression

$$P\mathbf{D}(\mathbf{v},\mathbf{w}) = \mathbf{D}(P\mathbf{v}, P\mathbf{w}).$$

Starting from this, we derive an expression for the t^2-term:

$$\begin{aligned}
\mathbf{R}^2_{\mathbf{k}}(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P\mathcal{L}(2P\mathcal{L}-\mathcal{L})[\mathbf{D}_{\mathbf{k}}(\hat{\mathbf{u}}^0,\tilde{\mathbf{u}}^0)+\mathbf{C}_{\mathbf{k}}(\tilde{\mathbf{u}}^0,\tilde{\mathbf{u}}^0)] \\
=& Pe^{t\mathcal{L}}P\mathcal{L}[2\mathbf{D}_{\mathbf{k}}(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0)) - \mathbf{D}_{\mathbf{k}}(\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\tilde{\mathbf{u}}^0) \\
& \qquad - \mathbf{D}_{\mathbf{k}}(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)) - \mathbf{D}_{\mathbf{k}}(\tilde{\mathbf{u}}^0,\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0))] \\
=& Pe^{t\mathcal{L}}[\mathbf{D}_{\mathbf{k}}(\hat{\mathbf{u}}^0,\tilde{\mathbf{D}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0)-\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0),\hat{\mathbf{u}}^0)) - \mathbf{D}_{\mathbf{k}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0),\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0))] \\
=& \mathbf{D}_{\mathbf{k}}(\hat{\mathbf{u}},\tilde{\mathbf{D}}(\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})-\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}),\hat{\mathbf{u}})) - \mathbf{D}_{\mathbf{k}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}),\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})).
\end{aligned}$$

Third-order Term

We will make use of the terms we have already computed in order to simplify our derivation of the third order term. Under the complete memory approximation, the third term is:

$$\mathbf{R}^3_{\mathbf{k}}(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathcal{L}[PLPL - 2PLQL - 2QLPL + QLQL]Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}}.$$

Once again, we rewrite it with no $Q\mathcal{L}$ terms.

$$\begin{aligned}
\mathbf{R}^3_{\mathbf{k}}(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P\mathcal{L}[PLPL - 2PLQL - 2QLPL + QLQL]Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}} \\
=& Pe^{t\mathcal{L}}P\mathcal{L}[3P\mathcal{L}(2P\mathcal{L}-P\mathcal{L}) - 3\mathcal{L}P\mathcal{L} + \mathcal{L}\mathcal{L}]Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}}
\end{aligned}$$

We recognize that we have already computed an expression for $P\mathcal{L}(2P\mathcal{L}-\mathcal{L})Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}}$ in the previous subsection. This leaves two additional terms to compute before simplifying and

applying the final $P\mathcal{L}$:

$$\begin{aligned}
\mathcal{L}P\mathcal{L}Q\mathcal{L}\mathbf{u}_\mathbf{k}^0 =& \mathcal{L}[\mathbf{D}_\mathbf{k}(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0))] \\
=& \mathbf{D}_\mathbf{k}(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) + \mathbf{D}_\mathbf{k}(\hat{\mathbf{u}}^0, \tilde{\mathbf{D}}(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \hat{\mathbf{u}}^0)) \\
\mathcal{L}\mathcal{L}Q\mathcal{L}\mathbf{u}_\mathbf{k}^0 =& \mathcal{L}\mathcal{L}[\mathbf{D}_\mathbf{k}(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + \mathbf{C}_\mathbf{k}(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0)] \\
=& \mathcal{L}[\mathbf{D}_\mathbf{k}(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{u}}^0) + \mathbf{D}_\mathbf{k}(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)) + \mathbf{D}_\mathbf{k}(\tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{u}}^0)] \\
=& \mathbf{D}_\mathbf{k}(\hat{\mathbf{D}}(\mathbf{C}(\mathbf{u}^0, \mathbf{u}^0), \mathbf{u}^0), \tilde{\mathbf{u}}^0) + 2\mathbf{D}_\mathbf{k}(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)) \\
& + \mathbf{D}_\mathbf{k}(\mathbf{u}^0, \tilde{\mathbf{D}}(\mathbf{C}(\mathbf{u}^0, \mathbf{u}^0), \mathbf{u}^0)) + \mathbf{D}_\mathbf{k}(\tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)).
\end{aligned}$$

Combining these three computed terms with the correct coefficients yields:

$$\begin{aligned}
[3P\mathcal{L}(2P\mathcal{L} - P\mathcal{L}) - 3\mathcal{L}P\mathcal{L} + \mathcal{L}\mathcal{L}]Q\mathcal{L}\mathbf{u}_\mathbf{k}^0 =& \mathbf{D}_\mathbf{k}(\hat{\mathbf{u}}^0, \tilde{\mathbf{D}}(-3\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0) + 3\hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0), \hat{\mathbf{u}}^0)) \\
& - 6\mathbf{C}_\mathbf{k}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0), \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) - 3\mathbf{D}_\mathbf{k}(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) \\
& + \mathbf{D}_\mathbf{k}(\hat{\mathbf{D}}(\mathbf{C}(\mathbf{u}^0, \mathbf{u}^0), \mathbf{u}^0), \tilde{\mathbf{u}}^0) + 2\mathbf{D}_\mathbf{k}(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)) \\
& + \mathbf{D}_\mathbf{k}(\mathbf{u}^0, \tilde{\mathbf{D}}(\mathbf{C}(\mathbf{u}^0, \mathbf{u}^0), \mathbf{u}^0)) + 2\mathbf{C}_\mathbf{k}(\tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)).
\end{aligned}$$

The t^3-term is found once we apply $P\mathcal{L}$ to this expression, yielding:

$$\begin{aligned}
\mathbf{R}_\mathbf{k}^3(\hat{\mathbf{u}}) =& \mathbf{D}_\mathbf{k}(\hat{\mathbf{u}}, \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
& + \tilde{\mathbf{D}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{D}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
& + 3\mathbf{D}_\mathbf{k}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))).
\end{aligned}$$

Fourth-order term

Our final included model will be the t^4-model. The derivation of these models is quite tedious. For this reason, but we make use of our symbolic tools described above [87]. The result for the fourth order model is:

$$
\begin{aligned}
\mathbf{R}^4_{\mathbf{k}}(\hat{\mathbf{u}}) =& e^{t\mathcal{L}}P\mathcal{L}[PLPLPL - 3PLPLQL - 5PLQLPL + 3PLQLQL \\
&\quad - 3QLPLPL + 5QLPLQL + 3QLQLPL - QLQLQL]Q\mathcal{L}\mathbf{u}^0_{\mathbf{k}} \\
=& \mathbf{D_k}(\hat{\mathbf{u}}, \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{D}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + 3\hat{\mathbf{D}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&\quad + \tilde{\mathbf{D}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) - \tilde{\mathbf{D}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&\quad + \hat{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{D}}(\hat{\mathbf{u}}, 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 5\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&\quad + \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{D}}(\hat{\mathbf{u}}, 5\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{D}}(\hat{\mathbf{u}}, 3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})))) \\
&\quad + \tilde{\mathbf{D}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \hat{\mathbf{D}}(\hat{\mathbf{u}}, 3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 5\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&\quad + \tilde{\mathbf{D}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \hat{\mathbf{D}}(\hat{\mathbf{u}}, 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{D}}(\hat{\mathbf{u}}, 5\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})))) \\
&- 4\mathbf{D_k}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{D}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{D}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&\quad + \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})))) \\
&- 3\mathbf{D_k}(\tilde{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})), \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&- 3\mathbf{D_k}(\tilde{\mathbf{D}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})), \tilde{\mathbf{D}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))).
\end{aligned}
$$

4.4.5 Renormalization coefficients

With these terms computed, we can express a renormalized reduced order model as:

$$\frac{dP\mathbf{u_k}}{dt} = \mathbf{R}^0_{\mathbf{k}}(\hat{\mathbf{u}}) + \sum_{i=1}^{n} \alpha_i(t) t^i \mathbf{R}^i_{\mathbf{k}}(\hat{\mathbf{u}})$$

for $n = 1, 2, 3, 4$. If we do not renormalize and $\alpha_i(t) = \frac{(-1)^{i+1}}{i!}$, we find that the simulations are unstable for all except the t-model alone. Instead, we will choose renormalization coefficients to stabilize the models. In the previous section, ROMs of the Korteweg-de Vries equation were stabilized with algebraically decaying renormalization coefficients [88]. In past work, it was found that ROMs of Burgers' equation and Euler's equations derived from the BCH approximation were stabilized with constant renormalization coefficients [104, 105]. We will

consider both cases as possible ansatzes. To be explicit,

$$\alpha_i(t) = a_i t^{-i} \tag{4.51}$$

$$\alpha_i'(t) = a_i'. \tag{4.52}$$

We will use the rates of change of the energy in each resolved mode as the quantities we attempt to match in the renormalization process. This choice is reasonable because it is known that energy moves from low-frequency modes to high-frequency modes as Euler's equations are evolved, and the Markov term is incapable of capturing this, since it conserves energy in the resolved modes. Thus, it makes an excellent heuristic for the effectiveness of a memory approximation. The energy of a mode is defined as:

$$E_{\mathbf{k}}(t) = |\mathbf{u}_{\mathbf{k}}|^2. \tag{4.53}$$

The rate of change of the energy in a particular mode in the full model is:

$$\Delta E_{\mathbf{k}}(t) = \mathbf{R}_{\mathbf{k}}(\mathbf{u}) \cdot \overline{\mathbf{u}}_{\mathbf{k}} + \mathbf{u}_{\mathbf{k}} \cdot \overline{\mathbf{R}}_{\mathbf{k}}(\mathbf{u}). \tag{4.54}$$

In a reduced order model, each term in the series has its own contribution to the energy derivative:

$$\Delta E^i_{\mathbf{k}}(t) = \mathbf{R}^i_{\mathbf{k}}(\hat{\mathbf{u}}) \cdot \overline{\mathbf{u}}_{\mathbf{k}} + \mathbf{u}_{\mathbf{k}} \cdot \overline{\mathbf{R}}^i_{\mathbf{k}}(\hat{\mathbf{u}}). \tag{4.55}$$

Given an exact energy derivative, our renormalization coefficients will be chosen to minimize the difference between $\Delta E_{\mathbf{k}}$ and

$$\Delta \hat{E}_{\mathbf{k},n}(t) = \Delta E^0_{\mathbf{k}}(t) + \sum_{i=1}^{n} \alpha_i(t) t^i \Delta E^i_{\mathbf{k}}(t).$$

We will renormalize against data $\Delta E_{\mathbf{k}}$ produced by a full model that we trust has not yet become unresolved. We use the Markov model to simulate this "full" system of size

$F \cup G = \{\mathbf{k} \mid \mathbf{k} \in [-M, M-1]^3\}$ up to time T. This produces a time series $\mathbf{u}(t)$ for $t = 0, \ldots, T$. We must identify the timesteps which correspond to times we are confident that this simulation is still resolved. We assume the transfer of energy is largely local. This means that the energy that begins in low-frequency modes at the beginning will begin to drain into modes with increasing $|\mathbf{k}|$ as time evolves. The assumption of local energy transfer means that we can assume a simulation is still resolved as long as energy has not yet reached the edge of the computational domain.

We process the data to find a conservative estimate of the resolved timesteps. At each timestep, we define $\hat{\mathbf{u}}(t)$ to be a restricted version of the calculated solution of size $F = \{\mathbf{k} \mid \mathbf{k} \in [-M/2, M/2-1]^3\}$ at the observed timesteps. We can calculate to first order the amount of energy flowing out of these modes through the t-model:

$$\Delta E_F(t) = t \sum_{\mathbf{k} \in F} \mathbf{R}^1_{\mathbf{k}}(\hat{\mathbf{u}}) \cdot \overline{\mathbf{u}}_{\mathbf{k}} + \mathbf{u}_{\mathbf{k}} \cdot \overline{\mathbf{R}}^1_{\mathbf{k}}(\hat{\mathbf{u}}). \tag{4.56}$$

We restrict ourselves to timesteps where the full data are still resolved by limiting ourselves to a specific set of timesteps $t^* = \{t \mid 10^{-16} < \Delta E_F(t) < 10^{-10}\}$. If there is so little energy leaving the cube corresponding to the inner 1/27th of the simulated domain, we conclude that the energy has not yet cascaded to the edge of the domain and the simulation is still resolved. We double-checked this by computing:

$$\Delta E_G(t) = t \sum_{\mathbf{k} \in G} \mathbf{R}^1_{\mathbf{k}}(\mathbf{u}) \cdot \overline{\mathbf{u}}_{\mathbf{k}} + \mathbf{u}_{\mathbf{k}} \cdot \overline{\mathbf{R}}^1_{\mathbf{k}}(\mathbf{u}), \tag{4.57}$$

which is a first-order approximation of the amount of energy that *should* be flowing out of the full system were it larger. We found that for all $t \in t^*$ this quantity was below machine precision. This makes us more confident that the timesteps t^* can be trusted as resolved. Thus, we can use it for fitting our reduced order models.

Consider a reduced order model of resolution N that includes reduced order models up through order n. By this we mean the ROM would compute the solutions for the wavenum-

bers in $F_N = \{\mathbf{k} \mid \mathbf{k} \in [-N, N-1]^3\}$. We assume the renormalization coefficients depend upon the system size N. We computed these coefficients using a least squares fit. For the algebraically decaying coefficients $\alpha_i(t) = a_i t^{-i}$, we minimized:

$$C_{N,n}(\mathbf{a}) = \sum_{\mathbf{k}\in F_N} \sum_{t\in t^*} \left(\Delta E_{\mathbf{k}} - \Delta E^0_{\mathbf{k}} - \sum_{i=1}^{n} a_i \Delta E^i_{\mathbf{k}} \right)^2 \tag{4.58}$$

where $\mathbf{a}$ is the vector of renormalization coefficients. For the constant renormalization coefficients $\alpha'_i(t) = a'_i$, we minimized:

$$C'_{N,n}(\mathbf{a}') = \sum_{\mathbf{k}\in F_N} \sum_{t\in t^*} \left(\Delta E_{\mathbf{k}} - \Delta E^0_{\mathbf{k}} - \sum_{i=1}^{n} a'_i t^i \Delta E^i_{\mathbf{k}} \right)^2 . \tag{4.59}$$

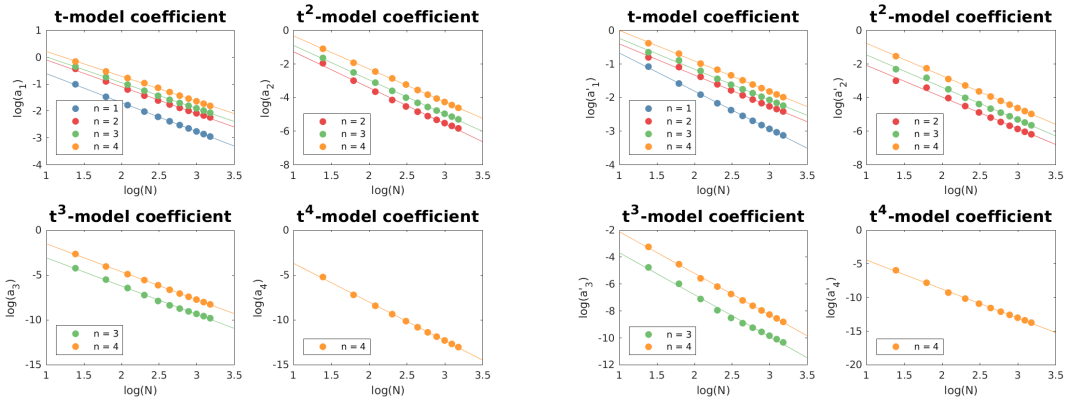

Figure 4.8: Optimal algebraically decaying renormalization coefficients a_i (left) and constant renormalization coefficients a'_i (right) plotted on a log-log scale against the system resolution N. The optimal coefficients were calculated by minimizing (4.58) and (4.59), respectively, for $N = 4, 6, \ldots, 24$ and $n = 1, 2, 3, 4$ against data produced by a still-resolved $M = 48$ full simulation.

We ran a full simulation of size $M = 48$ and used it to compute renormalization coefficients for reduced order models of size $N = 4, 6, \ldots, 24$ with algebraically decaying renormalization coefficients and constant renormalization coefficients. In each case, we considered

models that included up through $n = 1,2,3,4$ terms from the complete memory approximation. In all cases, there was an apparent algebraic dependence upon N for the constants $\mathbf{a}$ and $\mathbf{a}'$ as depicted in Figure 4.8. We also used simulations of size $M = 24$ and $M = 32$ to compute renormalization coefficients for reduced order models of size $N = 4, 6, \ldots, 12$ and $N = 4, 6, \ldots, 16$, respectively. The coefficients found in each of these calculations were similar. As the largest simulation, we trust the results from the $M = 48$ simulation the most.

Due to the apparent algebraic dependence, we conclude that the functional form of the renormalization coefficients is:

$$\alpha_i(t) = \beta_i^n N^{\gamma_i^n} t^{-i} \tag{4.60}$$

$$\alpha_i'(t) = \beta_i'^n N^{\gamma_i'^n}. \tag{4.61}$$

We identified the parameters β_i^n, γ_i^n, $\beta_i'^n$ and $\gamma_i'^n$ by computing a linear least squares fit of $\log(\alpha_i)$ and $\log(N)$. The resulting coefficients and the correlation coefficient of the linear least-squares fit r^2 are presented in Table 4.1.

There are a few comments we can make from these data. First, we see from the correlation coefficients that these fits are quite good. It appears the scaling law form is a good representation of the functional form of the renormalization coefficients. We see also that the exponent in those scaling laws (γ_i^n and $\gamma_i'^n$) seem relatively independent of the number of terms n included in the reduced model. The prefactors β_i^n and $\beta_i'^n$, however, slowly grow in magnitude as more terms are included. We also see that the coefficients of the even terms are negative while the coefficients of the odd terms are positive in all cases. Thus, the renormalized coefficients agree with the unrenormalized coefficients in sign though not in magnitude. Unlike the KdV case, there is no indication that only some of the memory approximation terms should be included in our ROMs.

n	β_1^n	β_2^n	β_3^n	β_4^n	γ_1^n	γ_2^n	γ_3^n	γ_4^n	r_1	r_2	r_3	r_4
1	1.591				−1.077				1.000			
2	2.448	−2.341			−0.999	−2.136			0.998	0.996		
3	2.650	−3.094	1.068		−0.962	−2.042	−3.148		1.000	0.999	0.998	
4	3.110	−5.006	4.924	−1.828	−0.924	−1.959	−3.115	−4.312	0.999	1.000	1.000	0.999

n	$\beta_1'^n$	$\beta_2'^n$	$\beta_3'^n$	$\beta_4'^n$	$\gamma_1'^n$	$\gamma_2'^n$	$\gamma_3'^n$	$\gamma_4'^n$	r_1'	r_2'	r_3'	r_4'
1	1.574				−1.132				0.999			
2	1.677	−0.805			−0.926	−1.879			0.998	0.994		
3	1.955	−1.570	0.573		−0.915	−1.925	−3.122		0.997	0.996	0.994	
4	2.454	−3.124	2.628	−0.860	−0.906	−1.924	−3.091	−4.306	0.999	1.000	0.999	0.998

Table 4.1: Scaling laws that approximate the observed optimal correlation coefficients for 3D Euler's equations. The top table contains scaling laws for algebraically decaying renormalization coefficients $\alpha_i(t) = \beta_i^n N^{\gamma_i^n} t^{-i}$ while the bottom table contains scaling laws for constant renormalization coefficients $\alpha_i'(t) = \beta_i'^n N^{\gamma_i'^n}$. We computed an $M = 48$ full simulation to construct $\Delta E_{\mathbf{k}}$. The algebraically decaying renormalization coefficient scaling laws were found by minimizing (4.58) with $n = 1, 2, 3, 4$ and $N = 4, 6, \ldots, 24$, then conducting a linear least squares fit of $\log(a_i)$ against $\log(N)$. The correlation coefficient r^2 of this log-log fit is also provided. The same method was used to compute the constant renormalization coefficient scaling laws.

4.4.6 Results

The behavior of the solution to the three-dimensional Euler's equations with a smooth initial condition remains unknown. Consequently, we cannot compare the results of our ROMs to the exact solution for accuracy. Instead, we endeavour to produce ROMs that remain *stable* over a long time. We will have to rely upon secondary means of inferring the accuracy of the resultant ROMs. Our results, not fully validated as they are, can be interpreted as evidence that is suggestive of long-term behavior of a subset of Fourier modes evolved according to Euler's equations.

First, it should be noted that renormalized ROMs with *constant* renormalization coefficients proved to be unstable for $n > 1$. The t-model is stable, by construction, but the addition of higher-order terms, even when renormalized, rendered the simulations unstable. As the order of the ROM increases for a fixed resolution, the time at which the model

becomes unstable becomes earlier (see Figure 4.9). Consequently, we conclude that the constant renormalization coefficients are not the correct choice for producing stable ROMs for Euler's equations.

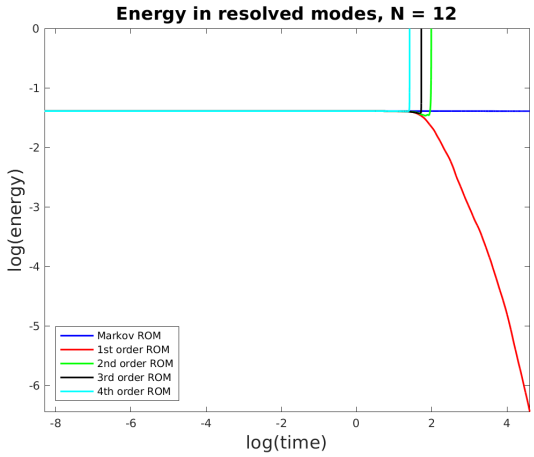

Figure 4.9: The energy contained in the resolved modes of several ROMs of resolution $N = 12$ using *constant* renormalization coefficients as described in the bottom of Table 4.1 depicted on a log-log plot. The Markov model is stable but does not drain any energy. The first order ROM is stable by construction. All other ROMs are unstable, and the time of instability grows smaller as the order of the ROM increases. These results are qualitatively the same for other resolutions N.

On the other hand, the ROMs with renormalization coefficients that *decay algebraically with time* led to solutions that remained stable until at least $t = 1000$. When we fix a resolution N and simulate ROMs that include up through degree $n = 1, 2, 3, 4$, as seen in Figure 4.10, the results converge quickly with increasing order. Each additional term in a ROM is more expensive to compute, and the fast convergence gives us confidence that including additional terms will only minimally affect our results. Thus, we will assume that the fourth order ROMs represent the most accurate simulations of the dynamics of the resolved modes.

As one increases the resolution of our fourth order ROMs, several fascinating patterns emerge. First, we consider the energy contained in the resolved modes. The Markov term

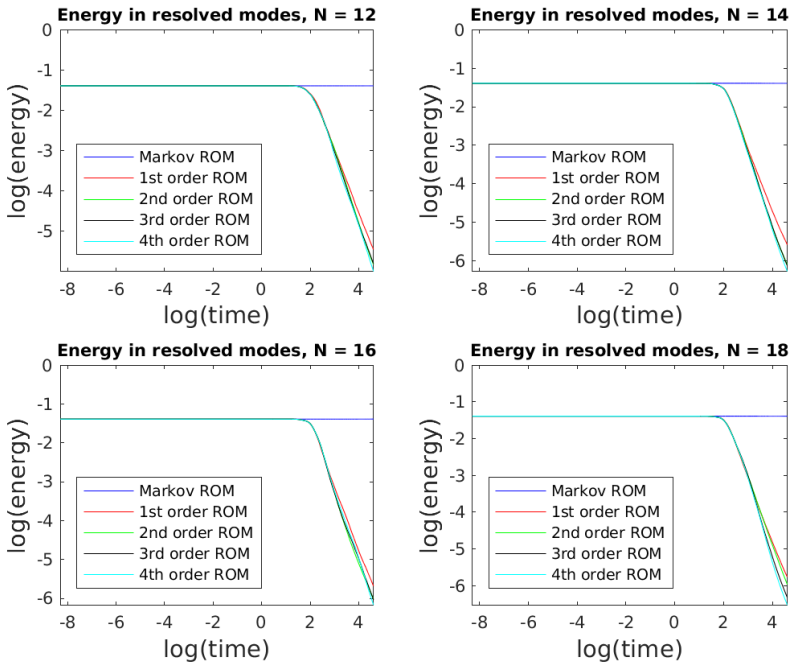

Figure 4.10: The energy contained in the resolved modes of several ROMs of resolution $N = 12, 14, 16, 18$ up to time $t = 100$ depicted on a log-log plot. The Markov model does not drain energy at all. The other four ROMs use algebraically decaying renormalization coefficients as described in the top of Table 4.1. Note that the behavior of the energy appears to converge as the order of the model increases, indicating that we are in the perturbative regime. The results for other resolutions N are qualitatively similar and the convergence appears to be faster as N grows larger.

conserves energy, so any draining of energy is accomplished by the memory terms alone. Figure 4.11 depicts the energy decay of ROMs with resolution $N = 4, 6, \ldots, 24$ up to time $t = 1000$ on a log-log plot. We see that in all cases there is monotonic energy decay. As time goes on, the results become stratified: the amount of energy remaining in the system at a given time *decreases* as the resolution of the model grows. This indicates significant activity in the high-frequency modes that increases with the resolution of the ROM. This is one point of evidence for a singularity. Were there not a singularity, one would not expect each larger ROM to drain more energy.

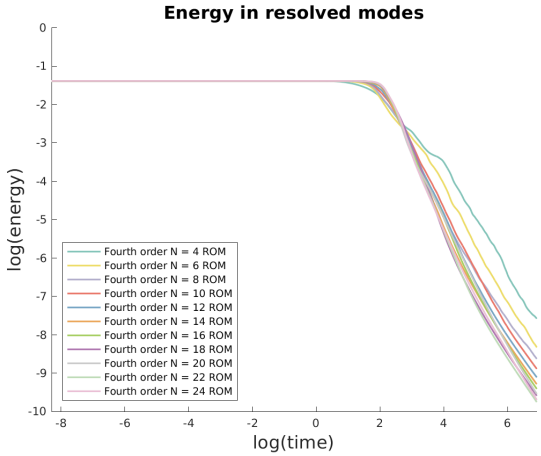

Figure 4.11: The energy contained in the resolved modes of fourth order ROMs of resolutions $N = 4, 6, \ldots, 24$ up to time $t = 1000$ depicted on a log-log plot. These simulations use algebraically decaying renormalization coefficients as described in the top half of Table 4.1. The qualitative behavor is the same for each resolution: energy does not drain at first, until it suddenly begins to do so at a nearly constant rate. After a long time, it gradually shifts to a different constant drain rate. The time at which the energy drain begins becomes later as N increases, and the initial slope grows steeper with increasing N. Both appear to have limiting values. The total energy ejected by the end of the simulation grows monotonically with increasing N.

We also see several other patterns as the resolution increases. The time at which the energy begins to drain from the resolved modes becomes later with increasing resolution. We measure the time at which the energy drain begins by identifying the point at which 10% of the initial energy has left the system. These data can be found in Table 4.2. This quantity seems to be converging towards a fixed value. If the time at which energy drain begins for the model of resolution N is designated T_N, we plot $T_{N+2} - T_N$ against T_N in Figure 4.12. A best linear fit intersects the x-axis at $t = 7.817$, suggesting this as a limiting value. The decay of energy is also relatively linear on a log-log plot (Figure 4.2), indicating algebraic energy ejection from the resolved modes. The slope indicates the exponent of the decay. We computed the slope from the data for which between 50% and 90% of the initial energy has left the system. These slopes seem to converge towards a approximately -2.5.

Finally, we see that the rate of energy ejection eventually becomes less steep. We computed the slope from the data after 99.5% of the initial energy had left the system. These new ejection rates seem to converge towards -1.5. All these observations are enumerated in Table 4.2.

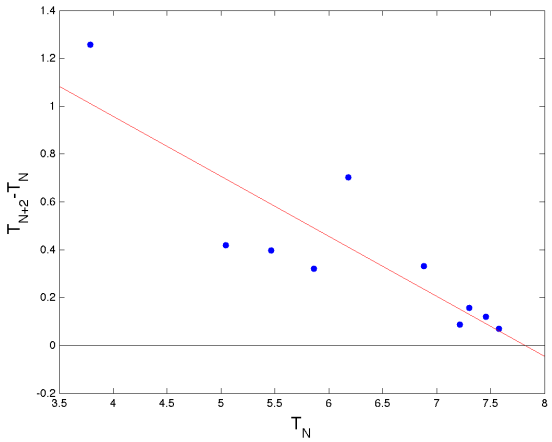

Figure 4.12: The times at which energy begins draining in reduced order models of resolution N, T_N, plotted against $T_{N+2} - T_N$. The limiting value of T_N corresponds to when this quantity intersects the x-axis. A best fit line (red) intersects the x-axis (black) at $t = 7.817$.

Next, we computed several other interesting dynamic quantities from our ROM simulations. In each of these quantities, we found perturbative convergence as the order of the model increases. This suggests that we can continue to treat the fourth order ROM as the most accurate result. The curl of the solution is called the *vorticity* in fluid dynamics. The enstrophy is defined as the L^2-norm of the vorticity:

$$e(t) = \int |\nabla \times \mathbf{u}|^2 \, \mathrm{d}\mathbf{x}, \tag{4.62}$$

where $\nabla \times \mathbf{u}$ is the vorticity. If a singularity does occur, we would expect a peak in the enstrophy to occur at the time of the singularity. Furthermore, we would expect this peak to grow larger as the resolution is increased (and become infinite as the resolution grows to

ROM resolution N	Initial decay time	Initial decay rate	Second decay rate
4	3.786	-0.808	-1.208
6	5.044	-1.003	-1.291
8	5.463	-1.575	-1.163
10	5.860	-1.646	-1.298
12	6.181	-1.916	-1.274
14	6.882	-1.981	-1.265
16	7.213	-2.005	-1.357
18	7.301	-1.989	-1.298
20	7.458	-2.208	-1.485
22	7.578	-2.226	-1.348
24	7.648	-2.414	-1.413

Table 4.2: Observations of the energy decay in reduced order models of 3D Euler's equations. Simulations use algebraically decaying renormalization coefficients as described in the top of Table 4.1. When plotted on a log-log plot, the energy begins to decay at a fixed rate after a certain amount of time. The time at which the energy begins to decay is calculated as the time at which 10% of the intial energy has left the resolved modes. The initial decay rate is the slope of a least-squares fit line to the log-log data for which between 50% and 90% of the initial energy has left the resolved modes. The second decay rate is the slope of a least-squares fit line to the log-log data after 99.5% of the data has left the resolved modes.

infinity). Indeed, these are exactly the results we observe (Figure 4.13). Furthermore, we observe that the time at which the enstrophy peaks roughly coincides with the time at which the energy begins to flow out of the system at a fixed algebraic rate.

The maximum of the vorticity

$$||\boldsymbol{\omega}||_\infty(t) = \max|\nabla \times \mathbf{u}| \tag{4.63}$$

is the best indicator of singular behavior [31]. During a singularity this quantity will blow up even if the enstrophy does not. In the presence of a singularity, one would again expect a peak in the maximum of the vorticity at some finite time, and we would expect this peak to grow towards infinity as the resolution is increased. We observe this here as well, and again the time of the peak seems to roughly coincide with the point at which energy begins

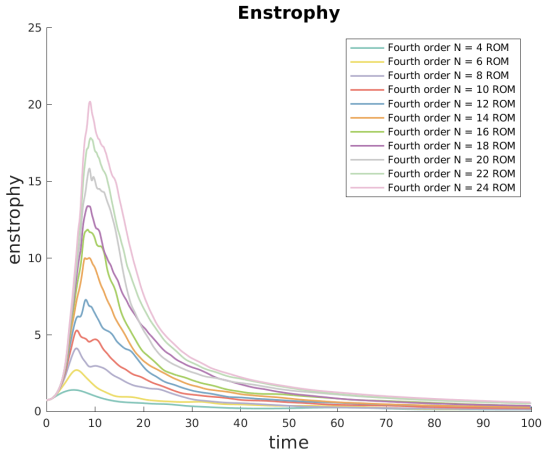

Figure 4.13: The enstrophy plotted against time for fourth order ROMs of size $N = 4, 6, \ldots, 24$ using algebraically decaying renormalization coefficients as described in the top half of Table 4.1 up to time $t = 100$. The enstrophy for each ROM begins small, grows to a maximum at a finite time, and then decays. As N increases, the maximum value achieved increases, and the time at which this maximum is achieved appears to converge. If this pattern continues and the enstrophy approaches infinity at some finite time as the number of simulated modes increases, we can conclude that a finite-time singularity occurs.

flowing out of the system (Figure 4.14). Data on the times and heights of the enstrophy and maximal vorticity peaks are presented in Table 4.3.

4.4.7 Discussion

Our reduced order models represent an advancement in the ability to simulate these equations. Without an exact solution to validate against, it is difficult to ascertain whether our results are accurate in addition to stable. However, there are a few hints: the convergence of behavior in Figure 4.10 indicates that our ROMs have a perturbative structure. That is, each additional order in the ROM modifies the solution less and less. It appears to be converging towards *something* and it is not unreasonable to think that it is the exact solution. Next, Table 4.1 demonstrates that adding additional terms does not significantly change the scaling laws for other terms. Each additional term is making *corrections* to previously

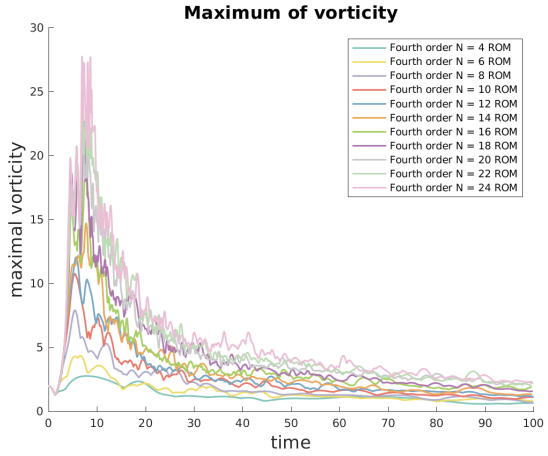

Figure 4.14: The maximal vorticity plotted against time for fourth order ROMs of size $N = 4, 6, \ldots, 24$ using algebraically decaying renormalization coefficients as described in the top of Table 4.1 up to time $t = 100$. The maximal vorticity grows to a maximum at a finite time like the enstrophy. As the resolution increases, the maximum value achieved increases. This is further suggestive that a finite-time singularity occurs.

captured behavior, but their contributions seem to be somewhat orthogonal to one another. These results together give us confidence in trusting these results.

The simulations are rather expensive. Each convolution requires a three-dimensional FFT and IFFT. Furthermore, for a system of resolution N, the FFTs are of size $(2 \times 3/2 \times N)^3$. The factor of $3/2$ is needed to dealias the results, and the factor of 2 comes from the fact that we include both positive and negative modes. The Markov model requires only one FFT. The first order ROM requires an additional two convolutions. The second order ROM adds another five, while the third order ROM has an additional nine on top of that. The fourth order ROM requires all past convolutions, plus another 20. Thus, the simulation cost grows very quickly as the degree of the ROM increases. On the other hand, as stated above, we are *incapable* of performing a brute force calculation beyond a few units of time even on modern cutting-edge high-resolution simulations. With our models we can integrate out to time $t = 1000$ for $N = 24$ in only a few days on a laptop computer. The results in Figures 4.11,

ROM resolution N	$\max e(t)$	Time of maximum	$\max \|\|\omega\|\|_\infty(t)$	Time of maximum
4	1.407	5.511	2.777	8.001
6	2.702	6.191	4.358	6.582
8	4.115	6.133	7.921	5.408
10	5.281	6.249	10.756	5.425
12	7.281	8.066	12.065	5.697
14	10.000	8.769	14.712	7.747
16	11.848	8.453	19.500	7.613
18	13.383	8.521	22.655	7.263
20	15.823	8.825	22.785	7.086
22	17.812	9.074	25.133	8.161
24	20.182	8.948	27.745	6.921

Table 4.3: Dependence of peaks of enstrophy and maximum vorticity on resolution. Simulations use algebraically decaying renormalization coefficients as described in the top of Table 4.1. The maximum value of the enstrophy $e(t)$, the maximal vorticity $||\omega||_\infty(t)$, and the times those maxima are achieved are presented. Note that the maxima grow as the resolution increases. In fact, they appear to increase nearly linearly. If this pattern continues, we expect the limit of $\max e(t)$ and $\max||\omega||_\infty(t)$ to be infinite as $N \to \infty$, which indicates that a singularity occurs. Note also that the time of the maximum appears to be converging. This time appears to be somewhere around $t = 9$ for the maximum of the enstrophy and $t = 8$ for the maximum of the vorticity. Compare this to the limiting time of the drain of energy, which is roughly $t = 8$.

4.13, and 4.14 and Tables 4.2 and 4.3 all provide strong evidence of a singularity occurring in Euler's equations with the Taylor-Green initial condition around $t = 8$. Our simulations are able to pass through the singularity, which would not be possible for a non-reduced system even with infinite computational power (if a singularity does indeed occur).

We have found that algebraically decaying renormalization coefficients are necessary to produce stable reduced order models. We found an algebraic dependence of the prefactors upon the resolution N. It is possible that a different time dependence for the renormalization coefficients could yield stable and accurate simulations. This is an area of future inquiry by the authors. The algebraic decay of energy from the resolved modes is suggestive of a finite time singularity. The increasing peaks of the enstrophy and maximum of the vorticity provide

further evidence for the formation of a singularity. The convergence of each of these events to a fixed time leads us to conclude that these reduced order models provide strong evidence for a finite time singularity developing from a smooth initial condition in Euler's equations. This is an important and long-standing open problem in fluid dynamics.

4.5 Application 3: Burgers' Equation

As a final application, we return to the first application to which renormalized models of this variety were applied: Burgers' equation,

$$u_t + uu_x = 0, \tag{4.64}$$

on a periodic domain with $u(x,0) = \sin(x)$. Note that this is the same as KdV with $\epsilon = 0$. The dynamics, however, are quite different (as is often the case with a singularly perturbed problem). This problem produces a shock after a few units of time. Once the shock forms, it dominates the dynamics of the system. In previous work, renormalized models of the BCH variety were used to construct reduced order models of the solution [104, 105]. We revisit this problem now with the complete memory approximation and our understanding of the types of observed renormalization coefficients. Once more, let

$$u(x,t) = \sum_{k \in F \cup G} u_k(t) e^{ikx}$$

where $F = [-N, \ldots, N-1]$, $G = [-M, \ldots, -N-1, N, \ldots, M-1]$, and $F \cup G = [-M, \ldots, M-1]$ for $N < M$ (as was the case for KdV). Let $\mathbf{u} = \{u_k(t)\}_{k \in F \cup G}$, and let $\hat{\mathbf{u}} = \{u_k\}_{k \in F}$ and $\tilde{\mathbf{u}} = \{u_k\}_{k \in G}$. The equation of motion for the Fourier mode u_k is

$$\frac{du_k}{dt} = R_k(\mathbf{u}) = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p,q \in F \cup G}} u_p u_q. \tag{4.65}$$

The convolution sum is identical to that found in KdV, and so it can also be conveniently computed in real space. We define $\mathcal{L}$ as in (2.2) such that $\mathcal{L}u_k^0 = R_k(\mathbf{u}^0)$ and the projector P in the usual way, $Pf(\mathbf{u}^0) = Pf(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) = f(\hat{\mathbf{u}}^0, 0)$. We can thus proceed to the derivation of reduced model terms.

4.5.1 Reduced Models

Let us focus on constructing a reduced order model for u_k where $k \in F$. Our Mathematica software [87] allowed us to complete this process in a few seconds (after about an hour of setting up the problem), which makes us optimistic about the ability to extend this methodology to other problems quickly. The convolution sum of interest in this case is:

$$C_k(\mathbf{v}, \mathbf{w}) = -\frac{ik}{2} \sum_{k \in F \cup G} v_p w_q, \tag{4.66}$$

such that the full model can be written:

$$\frac{du_k}{dt} = e^{t\mathcal{L}} C_k(\mathbf{u}^0, \mathbf{u}^0) = C_k(\mathbf{u}, \mathbf{u}). \tag{4.67}$$

With finite resolution, this model is unable to resolve the solution. We instead begin constructing the terms of a complete memory approximation of the projected dynamics:

$$\frac{dPu_k}{dt} = R_k^0(\hat{\mathbf{u}}) + \sum_{i=1}^{n} \alpha_i(t) t^i R_k^i(\hat{\mathbf{u}}). \tag{4.68}$$

Markov Term

The Markov term is

$$R_k^0(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}} P\mathcal{L}\mathbf{u}_k^0.$$

Using the definitions above, we find:

$$\hat{R}_k^0(\mathbf{u}) = Pe^{t\mathcal{L}}PC_k(\mathbf{u}^0, \mathbf{u}^0) = C_k(\hat{\mathbf{u}}, \hat{\mathbf{u}}).$$

First-order term

The t-model is

$$R_k^1(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathcal{L}Q\mathcal{L}\mathbf{u}_k^0.$$

This can be rewritten:

$$\begin{aligned} R_k^1(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P\mathcal{L}Q\mathcal{L}\mathbf{u}_k^0 \\ =& Pe^{t\mathcal{L}}P\mathcal{L}[2C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + C_k(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0)]. \end{aligned}$$

Once again, our operator $\mathcal{L}$ operates upon a convolution through the product rule. It operates upon $\hat{\mathbf{u}}$ and $\tilde{\mathbf{u}}$ in a simple manner:

$$\mathcal{L}\hat{\mathbf{u}}^0 = \hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \qquad \mathcal{L}\tilde{\mathbf{u}}^0 = \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)$$

where $\hat{\mathbf{C}}(\mathbf{v}, \mathbf{w})$ is the convolution of $\mathbf{v}$ and $\mathbf{w}$, but with the modes $k \in G$ set to zero and $\tilde{\mathbf{C}}(\mathbf{v}, \mathbf{w})$ is the convolution of $\mathbf{v}$ and $\mathbf{w}$ with the modes $k \in F$ set to zero. Therefore,

$$\begin{aligned} R_k^1(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P\mathcal{L}[2C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{u}}^0) + C_k(\tilde{\mathbf{u}}^0, \tilde{\mathbf{u}}^0)] \\ =& Pe^{t\mathcal{L}}P[2C_k(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{u}}^0) + 2C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)) + 2C_k(\tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{u}}^0)] \\ =& Pe^{t\mathcal{L}}[2C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0))] \\ =& 2C_k(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})). \end{aligned}$$

Second-order term

The t^2-model is:

$$R_k^2(\hat{\mathbf{u}}) = Pe^{t\mathcal{L}}P\mathcal{L}[P\mathcal{L} - Q\mathcal{L}]Q\mathcal{L}\mathbf{u}_\mathbf{k}^0 = Pe^{t\mathcal{L}}P\mathcal{L}[2P\mathcal{L} - \mathcal{L}]Q\mathcal{L}u_k^0.$$

The first term we have already computed:

$$2P\mathcal{L}Q\mathcal{L}u_k^0 = 4C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)).$$

The second term is:

$$-\mathcal{L}Q\mathcal{L}u_k^0 = -2C_k(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{u}}^0) - 2C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)) - 2C_k(\tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{u}}^0).$$

Combining them and applying $P\mathcal{L}$ to the result allows us to derive the second order term:

$$\begin{aligned}
R_k^2(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P\mathcal{L}[4C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0)) - 2C_k(\hat{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{u}}^0) - 2C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0)) + \\
& - 2C_k(\tilde{\mathbf{C}}(\mathbf{u}^0, \mathbf{u}^0), \tilde{\mathbf{u}}^0)] \\
=& Pe^{t\mathcal{L}}[4C_k(\hat{\mathbf{u}}^0, \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0))) - 2C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0), \tilde{\mathbf{C}}(\hat{\mathbf{u}}^0, \hat{\mathbf{u}}^0))] \\
=& 4C_k(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) - 2C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))
\end{aligned}$$

Third-order Term

Recall that the third term of the complete memory approximation is:

$$\begin{aligned}
R_k^3(\hat{\mathbf{u}}) =& Pe^{t\mathcal{L}}P\mathcal{L}[PLPL - 2PLQL - 2QLPL + QLQL]Q\mathcal{L}\mathbf{u}_\mathbf{k}^0 \\
=& Pe^{t\mathcal{L}}P\mathcal{L}[3P\mathcal{L}(2P\mathcal{L} - \mathcal{L}) - 3\mathcal{L}P\mathcal{L} + \mathcal{L}\mathcal{L}]Q\mathcal{L}\mathbf{u}_\mathbf{k}^0
\end{aligned}$$

We derived $P\mathcal{L}(2P\mathcal{L}-\mathcal{L})Q\mathcal{L}\mathbf{u}_\mathbf{k}^0$ in the previous subsection. Next we compute:

$$\begin{aligned}
\mathcal{L}P\mathcal{L}Q\mathcal{L}u_k^0 =&\mathcal{L}[2C_k(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0))]\\
=&2C_k(\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0))+4C_k(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)))\\
\mathcal{L}\mathcal{L}Q\mathcal{L}u_k^0 =&\mathcal{L}[2C_k(\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\tilde{\mathbf{u}}^0)+2C_k(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0))+2C_k(\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),\tilde{\mathbf{u}}^0)]\\
=&4C_k(\tilde{\mathbf{u}}^0,\mathbf{C}(\mathbf{C}(\mathbf{u}^0,\mathbf{u}^0),\mathbf{u}^0))+2C_k(\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),2\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)+\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0))\\
&+4C_k(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\mathbf{C}(\mathbf{u}^0,\mathbf{u}^0),\mathbf{u}^0))
\end{aligned}$$

Combining these three terms yields:

$$\begin{aligned}
[3P\mathcal{L}(2P\mathcal{L}-\mathcal{L})-3\mathcal{L}P\mathcal{L}+\mathcal{L}\mathcal{L}]Q\mathcal{L}\mathbf{u}_\mathbf{k}^0 =& C_k(\tilde{\mathbf{u}}^0,4\mathbf{C}(\mathbf{u}^0,\mathbf{C}(\mathbf{u}^0,\mathbf{u}^0)))\\
&+C_k(\hat{\mathbf{u}}^0,\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,-24\hat{\mathbf{C}}(\hat{\mathbf{u}}^0,\tilde{\mathbf{u}}^0)-12\hat{\mathbf{C}}(\tilde{\mathbf{u}}^0,\tilde{\mathbf{u}}^0)-12\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0))\\
&\qquad+\tilde{\mathbf{C}}(\mathbf{u}^0,4\mathbf{C}(\mathbf{u}^0,\mathbf{u}^0)))\\
&-6C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0),\tilde{\mathbf{C}}(\hat{\mathbf{u}}^0,\hat{\mathbf{u}}^0)+\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0))\\
&+2C_k(\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0),2\hat{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)+\tilde{\mathbf{C}}(\mathbf{u}^0,\mathbf{u}^0)).
\end{aligned}$$

The t^3-term is found once we apply $P\mathcal{L}$ to this expression, yielding:

$$\begin{aligned}
R_k^3(\hat{\mathbf{u}}) =&Pe^{t\mathcal{L}}P\mathcal{L}[3P\mathcal{L}(2P\mathcal{L}-\mathcal{L})-3\mathcal{L}P\mathcal{L}+\mathcal{L}\mathcal{L}]Q\mathcal{L}\mathbf{u}_\mathbf{k}^0\\
=&4C_k(\hat{\mathbf{u}},2\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})-2\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))+\tilde{\mathbf{C}}(\hat{\mathbf{u}},-2\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})+\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})))\\
&\qquad+\tilde{\mathbf{C}}(\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}),\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})-\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))+\tilde{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}),\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})))\\
&+12C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}),\tilde{\mathbf{C}}(\hat{\mathbf{u}},-\hat{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}})+\tilde{\mathbf{C}}(\hat{\mathbf{u}},\hat{\mathbf{u}}))).
\end{aligned}$$

Fourth-order term

The fourth order term, according to our symbolic computations, is:

$$
\begin{aligned}
\hat{R}_k^4(\mathbf{u}) =& e^{t\mathcal{L}} P\mathcal{L}[PLPLPL - 3PLPLQL - 5PLQLPL + 3PLQLQL \\
&- 3QLPLPL + 5QLPLQL + 3QLQLPL - QLQLQL]Q\mathcal{L}\mathbf{u}_\mathbf{k}^0 \\
=& 8C_k(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}, -5\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&+ \hat{\mathbf{C}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + 3\hat{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) \\
&+ 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, -3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 5\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}, 3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&+ \tilde{\mathbf{C}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), -3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) - \hat{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&+ \tilde{\mathbf{C}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \hat{\mathbf{C}}(\hat{\mathbf{u}}, 3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 5\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}, -3\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&+ \tilde{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \hat{\mathbf{C}}(\hat{\mathbf{u}}, -\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}, 5\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - 3\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})))) \\
&+ 16C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, -\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}, 2\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) - \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&+ \tilde{\mathbf{C}}(\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), -\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})) - \tilde{\mathbf{C}}(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&+ 24C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, -\hat{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) + 2\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))) \\
&- 24C_k(\tilde{\mathbf{C}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}})), \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \tilde{\mathbf{C}}(\hat{\mathbf{u}}, \hat{\mathbf{u}}))).
\end{aligned}
$$

4.5.2 Renormalization coefficients

We can now express a renormalized reduced order model as:

$$\frac{dPu_k}{dt} = R_k^0(\hat{\mathbf{u}}) + \sum_{i=1}^{n} \alpha_i(t) t^i R_k^i(\hat{\mathbf{u}}).$$

We find that the t-model is stable, but that higher-order models are generally unstable. In [104, 105], the BCH approximation of the memory term was renormalized with constant renormalization coefficients. However, in Sections 4.3 and 4.4, we found that coefficients with algebraic time decay performed optimally. We will explore both of these possibilities

here. That is, let

$$\alpha_i(t) = a_i t^{-i} \tag{4.69}$$

$$\alpha_i'(t) = a_i'. \tag{4.70}$$

As in the previous sections, the energy of a mode is defined as:

$$E_k(t) = |\mathbf{u}_k|^2, \tag{4.71}$$

and the rate of change of the energy in a particular mode in the full model is:

$$\Delta E_k(t) = R_k(\mathbf{u})\overline{u}_k + u_k\overline{R}_k(\mathbf{u}). \tag{4.72}$$

Once more we isolate the termwise contribution to this rate of change:

$$\Delta E_k^i(t) = R_k^i(\mathbf{u})\overline{u}_k + u_k\overline{R}_k^i(\mathbf{u}). \tag{4.73}$$

After computing the exact energy derivative for each mode at a variety of times, our renormalization coefficients are found by minimizing the difference between ΔE_k and

$$\Delta\hat{E}_{k,n}(t) = \sum_{i=0}^{n} \alpha_i(t) t^i \Delta E_k^i.$$

The exact solution for comparison was computed in real space using the upwind method with $\Delta x = \frac{2\pi}{10000}$ and $\Delta t = 10^{-4}$ (which satisfies the CFL stability condition). We solved the system on $t = [0, 10]$. The solution was then transformed into Fourier space for comparison. We call this exact solution for the Fourier modes $k \in [-5000, 4999]$ $\mathbf{u}(t)$. Consider t^* to be the set of times used in our fits.

Consider reduced order models of resolution N that includes up through order n. By this we consider our set of resolved modes $F_N = \{k \mid k \in [-N, N-1]\}$. We assume the

renormalization coefficients depend upon the system size N, as this has been observed in our past examples and in previous work. The cost function for the algebraically decaying coefficients was:

$$C_n(\mathbf{a}) = \sum_{k \in F_N} \sum_{t \in t^*} \left(\Delta E_k - \Delta E_k^0 - \sum_{i=1}^{n} a_i \Delta E_k^i \right)^2 \tag{4.74}$$

where $\mathbf{a}$ is the vector of renormalization coefficients. For the constant coefficients, we minimized:

$$C_n'(\mathbf{a}') = \sum_{k \in F_N} \sum_{t \in t^*} \left(\Delta E_k - \Delta E_k^0 - \sum_{i=1}^{n} a_i' t^i \Delta E_k^i \right)^2 . \tag{4.75}$$

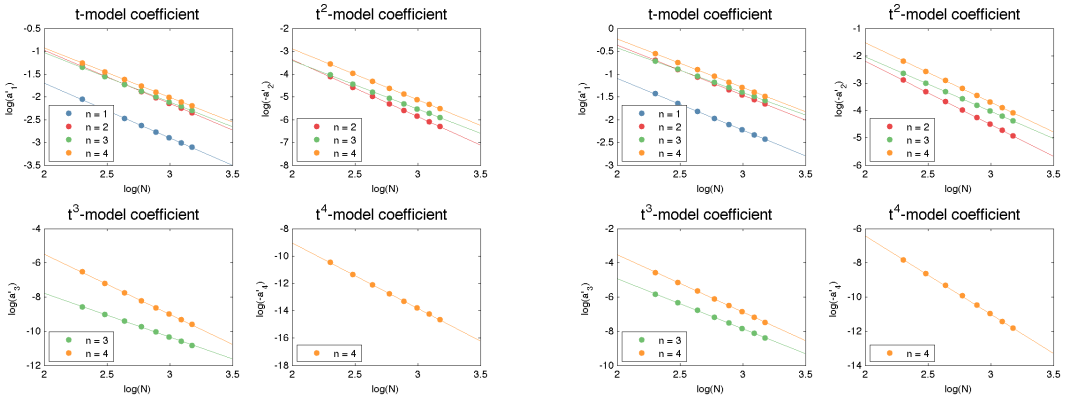

Figure 4.15: Optimal algebraically decaying renormalization coefficients a_i (left) and constant renormalization coefficients a_i' (right) plotted on a log-log scale against the system resolution N. The optimal coefficients were calculated for $N = 10, 12, \ldots, 24$ and $n = 1, 2, 3, 4$ against data produced by an upwind solution with $\Delta x = \frac{2\pi}{10000}$.

We used our exact solution with $t^* = \{0, 0.1, \ldots, 10\}$ to compute renormalization coefficients for reduced order models of size $N = 10, 12, \ldots, 24$ with algebraically decaying coefficients and constant coefficients. We considered models that included up through $n = 1$,2,3,4 terms from the complete memory approximation. The results are depicted in Figure 4.15. We also used $t^* = [0, t_M]$ where $t_M \in [0, 10]$ and found that, once the width of the fitting window was at least two the resultant coefficients did not differ significantly. We find again

that the functional form of the renormalization coefficients is:

$$\alpha_i(t) = \beta_i^n N^{\gamma_i^n} t^{-i} \tag{4.76}$$

$$\alpha_i'(t) = \beta_i'^n N^{\gamma_i'^n}. \tag{4.77}$$

We found β_i^n, γ_i^n, $\beta_i'^n$ and $\gamma_i'^n$ by computing a linear least squares fit of $\log(\alpha_i)$ and $\log(N)$. The resulting coefficients and the correlation coefficient of the linear least-squares fit r^2 are presented in Table 4.4.

There are a few comments we can make from these data. First, the scaling laws are quite strict, as can be seen from the near perfect (up to three digits) correlation coefficients. Second, one can compare these results to Table 4.1 and find that the scaling exponents γ_i^n and $\gamma_i'^n$ are nearly equal for Burgers' equation and Euler's equations. From these data, it is not immediately apparent which functional form for the renormalization coefficients should be preferred, so we will conduct experiments with both varieties. We do observe the same result as before: the signs of the renormalization coefficients agree with those of the nonrenormalized model.

4.5.3 Results

We solved each of our ROMs using `ode45` with the relative tolerance set to 10^{-10}. Solutions were computed for $n = 1, 2, 3, 4$ and $N = 4, 6, \ldots, 24$ for both types of coefficients. The results for the energy decay for $N = 8, 10, 12, 14$, which are representative of other simulated resolutions, are depicted in Figure 4.16. We observe a number of curious things. First, the ROMs with algebraically decaying renormalization coefficients are stable. Furthermore, they appear to display perturbative convergence (each additional term has less of an impact on the energy trajectory). However, the results are exceedingly inaccurate. This leads us to believe that algebraic time decay is not the proper functional form for Burgers' equation. The constant coefficient ROMs are intriguing as well. The $n = 1$ and $n = 3$ ROMs are stable and somewhat accurate, though the $n = 3$ model displays curious oscillations. The $n = 2$

n	β_1^n	β_2^n	β_3^n	β_4^n	γ_1^n	γ_2^n	γ_3^n	γ_4^n	r_1	r_2	r_3	r_4
1	2.025				-1.202				1.000			
2	3.880	-5.081			-1.167	-2.495			1.000	1.000		
3	3.089	-2.319	0.072		-1.081	-2.126	-2.569		1.000	1.000	0.999	
4	3.450	-4.845	4.548	-1.647	-1.083	-2.233	-3.507	-4.772	1.000	1.000	1.000	1.000

n	$\beta_1'^n$	$\beta_2'^n$	$\beta_3'^n$	$\beta_4'^n$	$\gamma_1'^n$	$\gamma_2'^n$	$\gamma_3'^n$	$\gamma_4'^n$	r_1'	r_2'	r_3'	r_4'
1	3.261				-1.137				1.000			
2	6.171	-11.747			-1.094	-2.325			1.000	1.000		
3	4.673	-6.841	2.602		-0.981	-1.982	-2.937		1.000	1.000	1.000	
4	6.721	-16.383	23.392	-15.627	-1.067	-2.162	-3.342	-4.579	1.000	1.000	1.000	1.000

Table 4.4: Scaling laws that approximate the observed optimal correlation coefficients for Burgers' equation. The top table contains scaling laws for algebraically decaying coefficients $\alpha_i(t) = \beta_i^n N^{\gamma_i^n} t^{-i}$ while the bottom table contains scaling laws for constant coefficients $\alpha_i'(t) = \beta_i'^n N^{\gamma_i'^n}$. The optimal coefficients were calculated for $N = 10, 12, \ldots, 24$ and $n = 1, 2, 3, 4$ against data produced by an upwind solution with $\Delta x = \frac{2\pi}{10000}$. We then conducted a linear least squares fit of $\log(\alpha_i)$ and $\log(\alpha_i')$ against $\log(N)$. The correlation coefficient r^2 of this log-log fit is also provided.

and $n = 4$ models are less accurate and stable. There is also, it appears, a lack of obvious perturbative convergence in these models.

We are able to compare our results to the exact solution. Because the decaying coefficient ROMs are so obviously inaccurate, we only consider the constant coefficient ROMs. We calculated the relative error (in the two norm metric) of our result as a function of time. Thus, for a given solution from an ROM of resolution N and degree n $\mathbf{u}^{N,n}(t)$ and exact solution $\mathbf{u}(t)$, we calculated

$$E_N^n(t) = \sum_{k \in F} \frac{|u_k^{N,n}(t) - u_k(t)|^2}{|u_k(t)|^2}. \tag{4.78}$$

These results are found in Figure 4.17. Notably, we see that the oscillations in the energy decay for $n = 3$ for large N manifest as a lack of accuracy. For small N, the $n = 3$ model yields an increase in accuracy over the $n = 1$ model, suggesting convergence. This is not the

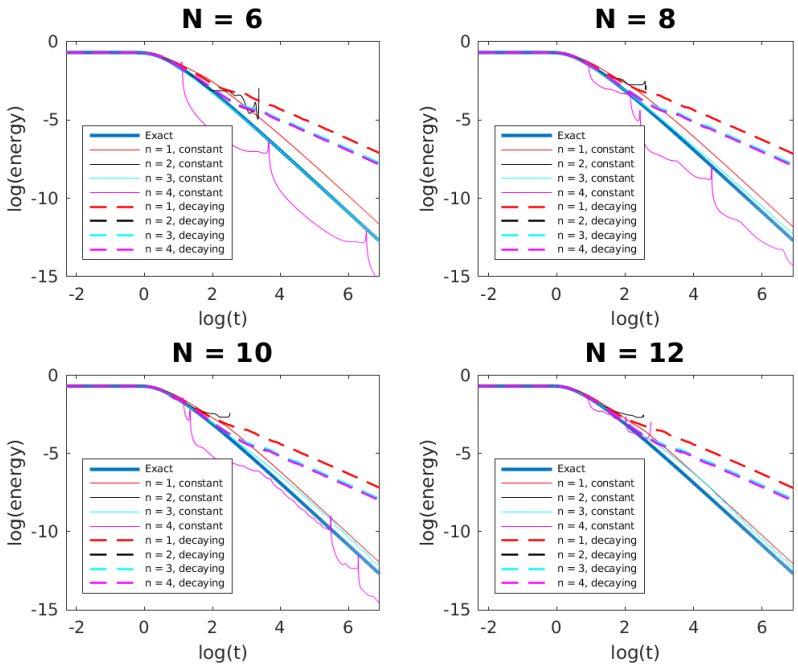

Figure 4.16: Energy decay of Burgers' equation according to an exact solution and ROMs of degree $n = 1, 2, 3, 4$ and resolution $N = 6, 8, 10, 12$ with both constant and decaying coefficients. Note that the decaying coefficient ROMs are all stable, converge as the number of terms grows, but are highly inaccurate. The constant coefficient ROMs are stable for $n = 1, 3$ and seem to converge towards the exact solution. However, as the resolution N increases, oscillations begin to appear in the $n = 3$ case. The constant coefficient $n = 2, 4$ ROMs appear to be unstable and display curious oscillations.

case for large N, in which the $n = 3$ error oscillates about the error from the $n = 1$ model. It is interesting that, in the cases where there are not oscillations we see the relative error grow to a maximal value and remain fixed as the simulation continues. One also observes that the $n = 1$ and $n = 3$ models are stable and accurate, while the $n = 2$ and $n = 4$ models are not. Perhaps one must add terms in pairs due to the alternating sign of the renormalization coefficients in this case. It would be interesting to explore an $n = 5$ model to see if this pattern continues.

Our results for Burgers' equation remain preliminary, but agree in part with past results

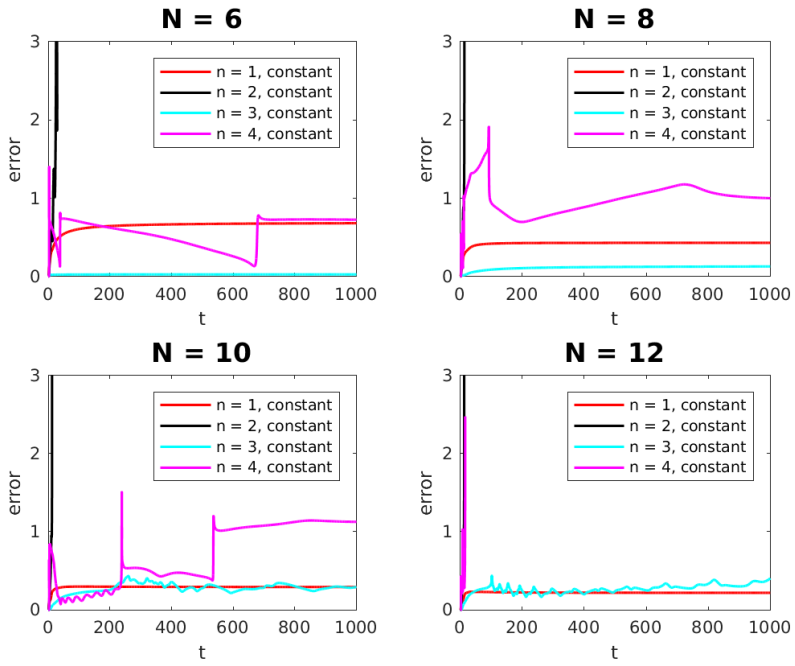

Figure 4.17: Relative error of Burgers' equation according to an exact solution and ROMs of degree $n = 1, 2, 3, 4$ and resolution $N = 6, 8, 10, 12$ with constant renormalization coefficients. The algebraically decaying coefficient models were observed to be inaccurate in capturing the energy evolution, so their errors are not depicted. Note that, for small N, the error is smaller for the $n = 3$ model. Furthermore, both the $n = 1$ and $n = 3$ models appear to reach a maximal error and then retain that error as time evolves. Note that for larger N, the oscillations in the $n = 3$ model lead to errors on the same order as the $n = 1$ model (a lack of convergence towards the exact solution).

which used the BCH approximation of the memory term. We find that constant renormalization coefficient models appear to perform better than decaying renormalization coefficient models (though other time dependencies should be explored). Furthermore, we find that terms must be added in pairs beyond the Markov model, perhaps due to the alternating signs of the coefficients. We once more observe scaling law behavior in the coefficients, which is promising. However, the unclear convergence of the models towards the exact solution is alarming, as are the unexplained oscillations. We will continue to investigate this problem

and in the process learn still more about the complete memory approximation.

4.6 The complete memory approximation method

By applying the complete memory approximation to three separate problems, we have developed a procedure for constructing reduced order models for systems of ODEs (in our case, solutions to PDEs on a periodic domain expressed as a Fourier series).

0. Derive the terms of the complete memory approximation to desired degree in terms of $\mathcal{L}$, P, and Q using symbolic software.

1. Define $\mathcal{L}$ such that $\frac{du_k}{dt} = \mathcal{L}u_k^0$ for a given system variable.

2. Define P as desired. Simpler choices lead to simpler ROMs, but more complicated or problem-informed choices could yield useful results.

3. Derive the series terms $R_k^n(\hat{\mathbf{u}})$ using symbolic software.

4. Compute a reference solution $\mathbf{u}(t)$ that you are confident is fully resolved at least for some time.

5. Choose a dynamic quantity to match. Examples include rates of change of energy or rates of change of particular modes.

6. From the exact solution, compute the contributions to the dynamic quantity given the solution at each time. For a variety of time-dependent functional forms, use least squares fits to choose renormalization coefficients to match the dynamic quantity.

7. Identify patterns in the renormalization coefficients and test strong candidates for stability and accuracy.

8. Use the renormalized models to explore results beyond the capacity of full simulations or to search for patterns in the contributions of memory terms.

The zeroth step must be done only once. Each other step has been demonstrated in Sections 4.3, 4.4, and 4.5. Step one is dictated by the chosen ODE system. For step two, in each case we have defined P to be a conditional expectation in which the unresolved modes are taken to be identically zero. This has made the construction of ROMs simpler, but is not a necessity. Step three can be made simpler through the use of our symbolic software [87]. Steps four and five are delicate. One must understand the system well to identify how to gain sufficient reference data, and one can use knowledge of the problem to select suitable matching quantities. Step six is similarly challenging, as we have found evidence for algebraically decaying renormalization coefficients (KdV and Euler's equations) and constant-coefficient renormalization coefficients (Burgers' equation). A more systematic approach for identifying and understanding suitable time dependence of renormalization coefficients is needed. Steps seven and eight can proceed quickly once the previous steps have been completed. They promise to yield fascinating results.

The steps described above create a roadmap for explorations of the complete memory approximation. The complete memory approximation is derived directly from the Mori-Zwanzig formalism, which is an exact representation of the system dynamics. Furthermore, we do not neglect any terms (as in the BCH approximation to the memory). Instead, we allow the memory term to inform the structure of the reduced order models. The fact that these models are unstable is unfortunate, but not surprising: we are using approximations that are only exact when the system is fully resolved to probe systems that are not resolved. This is analogous to the situation in physics for which renormalization was the solution. We find that this is the case here as well. The effective renormalization coefficients we found display surprising and fascinating structure. Furthermore, the reduced order models allow us to explore the behavior of solutions beyond our current computational limits, such as providing increasing evidence for a finite-time singularity in Euler's equations. There is still much to be learned about the complete memory approximation, and by following the above steps we can apply it to any number of new problems, each of which will shed more light on the role of memory in efficiently simulating dynamical systems.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Summary of results

Multiscale problems are ubiquitous in all areas of science and applied mathematics. Problem-specific multiscale simulations methods (such as the heterogeneous multiscale method) and general purpose multiscale techniques (such as adaptive mesh refinement) have been developed to solve important and challenging problems. The Mori-Zwanzig formalism is a framework for thinking about multiscale methods in a rigorous and general sense. In this dissertation, we have endeavoured to expand both the theory and practice of multiscale simulation methods. In Chapter 2, we began with a detailed overview of the Mori-Zwanzig formalism as a mathematical tool. One can reframe any system of ODEs into a linear PDE through the Liouvillian operator $\mathcal{L}$. In this form, projection techniques are more convenient. We can define a projection operator P that maps functions of every variable onto functions that depend only upon a subset of the variables (called the resolved variables). With these defined, we can break the dynamics of the resolved variables into a Markov term, a noise term, and a memory term. Using the Mori-Zwanzig formalism to construct reduced order models requires one to wrestle with how to approximate the noise and memory terms, which are connected to the unresolved variables. We demonstrated all of these ideas with a simple example of a linear system of differential equations.

In Chapter 3, we introduced the heterogeneous multiscale method, which is a simulation framework that has been used to much success in the computational physics community. We demonstrated how the general approach can be interpreted as a special case of the Mori-Zwanzig formalism in which the terms relating to the unresolved variables are approximated with a constrained approximation of the underlying microscale dynamics. Next, we presented

results applying the heterogeneous multiscale method to plasma physics applications. We successfully coupled a kinetic macroscale solver to a molecular dynamics microscale solver in order to increase the accuracy of the macroscale solver. We applied this methodology to a zero-dimensional temperature relaxation problem, and are eager to expand to higher-dimensional problems, in which the improvements will become much more significant. We also demonstrated how this specific case of HMM can be mapped onto the Mori-Zwanzig formalism framework.

In Chapter 4, we return our attention to the Mori-Zwanzig formalism. We confine ourselves to studying only the projected dynamics, which eliminates the noise term. Under most projections, this amounts to considering only the average behavior and so cannot produce actual dynamical paths. In the examples we presented, however, the projected dynamics are actual dynamical paths because our projectors have zero variance in the unresolved modes. We present the BCH approximation method, which has been used to some success in the past, but requires assumptions about the commutativity of the projected and the orthogonal dynamics. We develop a new method of approximating the memory term, which we call the complete memory approximation. This does not require any assumptions about the dynamics, except that the solutions of the orthogonal dynamics equations are sufficiently smooth that we can interchange an infinite sum and an integral. This introduces new terms that are not expressible in terms of only the resolved variables. These terms can be themselves expressed through the Mori-Zwanzig formalism and approximated by expanding the evolution operators and integrating termwise. By repeating this process, we derive a telescoping but deterministic method for constructing reduced order models of arbitrary order. This entire approach is called the "complete memory approximation." We furthermore develop symbolic software to complete this process automatically. The resulting reduced order models are quite complicated, and are unlikely to be proposed by a mathematical modeler. However, these models are derived from the equations themselves and represent in some way the "correct" form of reduced order models for a system.

Reduced order models constructed naively from the complete memory approximation are

unstable, like past reduced order models derived from the Mori-Zwanzig formalism. We stabilize our methods through the process of *renormalization*. That is, we acknowledge that our reduced order model is necessarily incomplete. We maintain the functional form of the terms we derived, but we modify their coefficients to make them *effective* terms. In order to choose these renormalization coefficients, we must have access to the exact solution at least up to some time. In our case, we are studying how equations in which smooth initial conditions develop increasingly complex activity. The early stages of the simulation, however, can be found exactly. We then applied this methodology to three different partial differential equations on periodic domains: the Korteweg-de Vries equation, the three-dimensional Euler's equations, and Burgers' equation. Each case showed us different interesting aspects of the complete memory approximation.

In KdV, we found that the renormalization coefficients required algebraically decaying time dependence that canceled the time dependence in the Taylor series. Furthermore, we found the prefactor of this time dependence itself obeyed a scaling law in both the degree of dispersion ϵ and the resolution N. The limiting behavior of these coefficients suggest that these coefficients will grow without bound as $\epsilon \to 0$ (at $\epsilon = 0$ we have Burgers' equation). This discontinuous behavior in the coefficients is a consequence of the singularly perturbed nature of the problem. Furthermore, we demonstrated that even when the amount of energy leaving the resolved modes is very small, the memory term cannot be completely disregarded without sacrificing accuracy. We found that only the even terms are necessary for capturing these dynamics. The implications of these results are fascinating and merit significant further study.

In Euler's equations, we found that the algebraic time decay of the renormalization coefficients was again necessary. We found an algebraic dependence of the prefactors upon the resolution N once more. We were unable to compare our results to exact solutions, since modern simulations of Euler's equations have not yet extended beyond a few units of time. We did, however, demonstrate that the renormalized complete memory approximation appears to be perturbative in nature: each subsequent term provides increasingly smaller

corrections. This can be seen by the convergence of the energy evolution and the insensitivity of the renormalization coefficients to the number of included terms. The algebraic decay of energy from the resolved modes evokes the formation of a singularity in Burgers' equation. The increasing peaks of the enstrophy and maximum of the vorticity provide further evidence for the formation of a singularity. The convergence of each of these events to a fixed time leads us to conclude that these reduced order models provide strong evidence for a finite time singularity developing from a smooth intial condition in Euler's equations. This is an important and long-standing open problem in computational fluid dynamics. Our simulations were able to proceed up to $t = 1000$ and likely beyond. This is hundreds of times farther than state-of-the-art full scale simulations. Reduced order models allow us to utilize the multiscale structure of problems to evolve only a subset of the variables in the system.

In Burgers' equation, we found once more that the renormalization coefficients have a very particular structure. This time, the constant renormalization coefficients performed best. The value of this constant obeyed a strict scaling law with the resolution of the model N. Furthermore, we found the curious case that we achieved stability only when we added terms beyond the t-model in pairs. We found unsettling evidence that models can be stable, but converge to the wrong solution (as in the algebraically decaying renormalization coefficient models). We also found strange oscillations that eliminated the convergence of our models. There is obviously much to be learned from a continued study of Burgers' equation, and we expect those results to inform our interpretation of the results from KdV and Euler's equations as well.

In all cases, we found the sign of the renormalization coefficients agreed with the sign of the unrenormalized model terms. The dramatic scaling laws for the prefactors are exceedingly unlikely to arise from chance, and we believe they are emblematic of the unique nature of the complete memory approximation. The general nature of the Mori-Zwanzig formalism and the lack of assumptions needed to employ the complete memory approximation combine to make a powerful and interesting new tool for studying reduced order models and memory.

5.2 Future directions

The development of the HMM plasma method and the complete memory approximation open many doors for future inquiry. We will comment here upon some promising avenues we plan to explore in the coming years.

5.2.1 Improving the HMM plasma algorithm

The multiscale plasma simulation algorithm described in Chapter 3 shows great promise, and there are a number of obvious and less obvious extensions. The current state of the algorithm is implemented only for zero-dimensional problems, which do not benefit tremendously from the multiscale approach. We are preparing those results for publication now. A simple extension is to generalize the implementation to include multidimensional problems. In particular, a one-dimensional shock problem would be an interesting test case. This example is prevalent in inertial confinement fusion simulations, and the adaptive multiscale technique should scale well to this problem.

Another extension draws further inspiration from machine learning. Each time one computes the relaxation rates from a microscale simulation, one could store the result and refer to it again if one encounters a similar macroscale state. For nearby macroscale states, interpolation methods called "kriging", which come with built-in error approximations, could be employed to infer relaxation rates. This framework would minimize the frequency with which one needs to generate microscale data, which is the limiting constraint of the simulation. This is inspired by the work described in [93]. In our case, however, the macroscale variables are continuous distribution functions. In order to practice interpolation in a space of reasonable dimension, we need to represent our macroscale state with only a few variables. We have several ideas for approaching this, such as representing the solution in terms of a basis of commonly-encountered distributional shapes. Successfully improving this multiscale method will make it easier to scale up to large problems, including ones of national interest such as nuclear fusion simulations.

5.2.2 Further investigation of the complete memory approximation

The complete memory approximation is a novel and interesting method for constructing reduced order models from the Mori-Zwanzig formalism. We have found that the renormalization coefficients it requires inherit tremendous structure from the full system it is approximating. There are many directions to take this new tool, both in terms of understanding the tool itself and applying it to interesting problems.

Interrogating the parity of terms

We were surprised to find that only even terms were needed in reduced order models for KdV, while even and odd terms were needed for Euler's equations. For Burgers' equation, we found that stable models only appeared when pairs of terms were included. We would like to investigate this phenomenon, and have several theories. KdV is a dispersive equation. Dispersive effects are associated with equations that have an odd number of derivatives. The even terms in the complete memory approximation have an odd number of differential operators. Thus, we hypothesize that the even terms of the complete memory approximation are associated with dispersive behavior. While this explains our results for KdV, it does not fit with what we know of Burgers' equation. Burgers' equation is dissipative, and dissipation is associated with even numbers of differential operators. One would expect that *only* the odd terms would be needed for Burgers' equation, but instead we find that the terms must be added in pairs with an odd term being the highest term. Finally, Euler's equations are completely different. ROMs constructed for Euler's equations are stable for models with even and odd maximal terms.

We plan to expore the role of dispersion and dissipation in the complete memory approximation by invesigating the KdV-Burgers equation:

$$u_t + uu_x = \alpha u_{xx} + \epsilon^2 u_{xxx} \tag{5.1}$$

on a periodic domain. This equation exhibits both dissipation and dispersion. By adjusting

the ratio of α to ϵ, we can dictate the degree of dispersion and dissipation. We hope to study the optimal renormalization coefficients for different ratios of these parameters. It is our hope that we will see interesting limiting behavior in the coefficients: perhaps the even-term coefficients will be large when dispersion is the predominant force, but will tend towards zero as the dissipative term becomes more prominent (with the odd-term coefficients doing the opposite). The study of the limiting behavior of renormalization coefficients is reminiscent of renormalization group theory in physics, in which one begins to study the limiting behavior of renormalization coefficients in order to learn about critical behavior.

Time dependent renormalization coefficients

We observed effective renormalization coefficients with two types of time dependence: algebraic decay to cancel the time dependence inherent in the Talor series and constant. It is possible that other time dependences could be relevant for these or other models. We hope to explore other possible time dependences without generating ansatzes *a priori*. In fact, we first discovered the algebraic time decay of the KdV renormalization coefficients by comparing the exact evolution of a matching quantity and the value of each term in the complete memory approximation expansion over time. We have access to an exact time series and n other time series with which to fit (where n is the maximal degree in the truncated series). We would like to find simple functions of time that allow us to match the exact time series with our component time series.

One might make an ansatz that the time dependence is algebraic with an undefined exponent $\alpha_i(t) = a_i t^{\eta_i}$, and use a nonlinear least squares optimization to find the optimal prefactors and exponents. We tried this, hoping to discover $\eta_i = -i$ for KdV. In our experience, when the nonlinear optimization method was supplied with a guess *exceptionally close* to $\eta_i = -i$, it found that to be the minimum. For guesses further away, the optimization algorithm failed to converge. It is possible that this extreme sensitivity is indicative of critical behavior near the optimal exponent, which would be worthy of further investigation.

We could also expand our "ansatz" to a library of possible time dependences and allow the

data to identify the best possible fit from a large collection of possible time dependences. This approach would employ modern machine learning ideas. Data-driven model discovery such as this has been the topic of intense inquiry in recent years [15, 16, 90] and we intend to explore these concepts as we consider other possible functional dependences for the renormalization coefficients.

Larger scale complete memory approximation simulations

We have described applying the complete memory approximation to the Korteweg-de Vries equation, Euler's equations, and Burgers' equation. In each case, our simulations were run on a reasonably powerful desktop computer. It is our hope that the complete memory approximation will be useful in larger scale computations as well. We are confident that there are little new results of interest to gain from larger simulations of Burgers' equation and KdV. Burgers' equation consists of a shock with little other interesting dynamics. We believe we have already accurately captured these results and that larger resolution simulations will reveal little more. The KdV equation can be fully resolved with a finite number of modes, and we believe we have comprehensively studied the effect of applying complete memory approximation to it.

The three-dimensional Euler's equations, however, remain an open question. Our evidence is *suggestive* of a finite-time singularity, but this evidence would become more persuasive with larger-scale simulations. We were only able to simulate relatively small 24^3 mode simulations using our memory approximation. This precludes interesting visualization of the results becuase they lack sufficient resolution. Furthermore, it is possible that new and exciting effects only become apparent at sufficient resolution. We would be interested in applying our reduced order models to a more finely-resolved simulation on a powerful computer.

Application to other systems

Aside from the KdV-Burgers equation, there are countless other partial differential equations that we would be interested in investigating with our novel reduced order model framework. One interesting case is the critical nonlinear Schrödinger equation:

$$iu_t + \Delta u + |u|^4 u = 0. \tag{5.2}$$

The unrenormalized t-model alone has been used to simulate beyond a finite-time singularity [103]. It would be interesting to investigate how both the additional terms of the complete memory approximation and the renormalization coefficients affect these results. It would also be interesting to generalize those results to higher dimensions. There are countless other interesting PDEs on periodic domains that could be studied in the manner described in this dissertation. The symbolic notebook we developed allows one to quickly study new equations of this form [87].

Throughout this text, we have applied the complete memory approximation to problems on a periodic domain and with a very specific projector P. There is nothing in the derivation that requires the use of Fourier modes or forces us to use this simple projector. If one wishes to turn this new memory approximation into a more general-purpose tool, one would need to study how to apply it to more complicated bases and projections. In those cases, the orthogonal dynamics (the "noise" term in the Mori-Zwanzig formalism) might become important. A logical next step is to apply this formalism to realistic problems with projections such as conditional expectations. We aspire to turn this new approximation method into a general tool for reduced-order modeling that can be applied to problems in many different scientific domains.

5.3 Conclusion

We have strived in this dissertation to provide a useful overview of multiscale methods from an applied mathematics perspective. We focused our attention on practical computational

methods, such as HMM, and useful theoretical formulations, such as the Mori-Zwanzig formalism. Our most significant contribution to the field of multiscale modeling is the complete memory approximation, which allows us to construct reduced order models of arbitrary order from the Mori-Zwanzig formalism with minimal additional assumptions. We have demonstrated the usefulness of this tool in three separate applications, including one of the biggest open questions in computational fluid dynamics. The remarkable structure found from these applications suggests that this framework is indeed an important and essential means of understanding the role of memory in reduced order models. The code described can be accessed at [87]. We hope this tool will allow us to simulate reduced order models of systems beyond the capabilities of modern computation, and consequently allow us to probe new and exciting numerical solutions.

BIBLIOGRAPHY

[1] A. Abdulle and A. Nonnenmacher, *A short and versatile finite element multiscale code for homogenization problems*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 2839–2859.

[2] ——, *Adaptive finite element heterogeneous multiscale method for homogenization problems*, Comput. Meth. Appl. Mech. Engr., 200 (2011), pp. 2710–2726.

[3] D. Agafontsev, E. Kuznetsov, and A. Mailybaev, *Development of high vorticity structures in incompressible 3d Euler equations*, Physics of Fluids, 27 (2015), p. 085102.

[4] P. Amendt, S. C. Wilks, C. Bellei, C. K. Li, and R. D. Petrasso, *The potential role of electric fields and plasma barodiffusion on the inertial confinement fusion database*, Phys. Plasmas, 18 (2011), p. 056308.

[5] S. Atzeni, A. Schiavi, F. Califano, F. Cattani, F. Cornolti, D. Del Sarto, T. Liseykina, A. MacChi, and F. Pegoraro, *Fluid and kinetic simulation of inertial confinement fusion plasmas*, Computer Physics Communications, 169 (2005), pp. 153–159.

[6] D. Ayala and B. Protas, *Extreme vortex states and the growth of enstrophy in three-dimensional incompressible flows*, Journal of Fluid Mechanics, 818 (2017), pp. 772–806.

[7] G. I. Barenblatt, *Scaling*, Cambridge University Press, 2003.

[8] C. L. Beck, S. Lall, T. Liang, and M. West, *Model reduction, optimal prediction, and the mori-zwanzig representation of markov chains*, in Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, Dec 2009, pp. 3282–3287.

[9] C. M. Bender and S. A. Orszag, *Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory*, Springer Science & Business Media, 2013.

[10] M. J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, Journal of Computational Physics, 82 (1989), pp. 64–84.

[11] M. J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, Journal of Computational Physics, 53 (1984), pp. 484–512.

[12] D. Bernstein, *Optimal prediction of Burgers's equation*, Multiscale Modeling & Simulation, 6 (2007), pp. 27–52.

[13] G. F. Bertsch, H. Kruse, and S. D. Gupta, *Boltzmann equation for heavy ion collisions*, Phys. Rev. C, 29 (1984), pp. 673–675.

[14] A. V. Bobylev and K. Nanbu, *Theory of collision algorithms for gases and plasmas based on the Boltzmann equation and the Landau-Fokker-Planck equation*, Phys. Rev. E, 61 (2000), pp. 4576–4586.

[15] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proceedings of the National Academy of Sciences, 113 (2016), pp. 3932–3937.

[16] S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz, *Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems*, SIAM Journal on Applied Dynamical Systems, 13 (2014), pp. 1716–1732.

[17] S. Bunyavanich and E. E. Schadt, *Systems biology of asthma and allergic diseases: A multiscale approach*, Journal of Allergy and Clinical Immunology, 135 (2015), pp. 31–42.

[18] G. Bussi and M. Parrinello, *Stochastic thermostats: Comparison of local and global schemes*, Computer Physics Communications, 179 (2008), pp. 26 – 29. Special issue based on the Conference on Computational Physics 2007.

[19] A. J. Chandy and S. H. Frankel, *The t-model as a large eddy simulation model for the Navier-Stokes equations*, Multiscale Modeling & Simulation, 8 (2009), pp. 445–462.

[20] S. Chapman and T. G. Cowling, *The mathematical theory of non-uniform gases: An account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*, Cambridge University Press, 1970.

[21] A. J. Chorin, *Vorticity and turbulence*, vol. 103, Springer Science & Business Media, 1994.

[22] ——, *Averaging and renormalization for the Korteveg-deVries-Burgers equation*, Proceedings of the National Academy of Sciences, 100 (2003), pp. 9674–9679.

[23] A. J. Chorin, O. H. Hald, and R. Kupferman, *Optimal prediction and the Mori-Zwanzig representation of irreversible processes*, Proceedings of the National Academy of Sciences, 97 (2000), pp. 2968–2973.

[24] ——, *Optimal prediction with memory*, Physica D: Nonlinear Phenomena, 166 (2002), pp. 239–257.

[25] A. J. Chorin and P. Stinis, *Problem reduction, renormalization, and memory*, Communications in Applied Mathematics and Computational Science, 1 (2007), pp. 1–27.

[26] P. Constantin, *Geometric statistics in turbulence*, SIAM Review, 36 (1994), pp. 73–98.

[27] ——, *Euler and Navier-Stokes equations*, Publicacions Matemàtiques, (2008), pp. 235–265.

[28] ——, *Analysis of Hydrodynamic Models*, SIAM, 2017.

[29] B. Delamotte, *A hint of renormalization*, American Journal of Physics, 72 (2004), pp. 170–184.

[30] J. Deng, T. Y. Hou, and X. Yu, *Improved geometric conditions for non-blowup of the 3D incompressible Euler equation*, Communications in Partial Differential Equations, 31 (2006), pp. 293–306.

[31] C. R. Doering and J. D. Gibbon, *Applied analysis of the Navier-Stokes equations*, vol. 12, Cambridge University Press, 1995.

[32] T. A. Driscoll, *A composite Runge-Kutta method for the spectral solution of semi-linear PDEs*, Journal of Computational Physics, 182 (2002), pp. 357–367.

[33] W. E and X. Li, *Analysis of the heterogeneous multiscale method for gas dynamics*, Methods Appl. Anal., 11 (2004), pp. 557–572.

[34] D. J. Earl and M. W. Deem, *Parallel tempering: Theory, applications, and new perspectives*, Physical Chemistry Chemical Physics, 7 (2005), pp. 3910–3916.

[35] T. Eissing, L. Kuepfer, C. Becker, M. Block, K. Coboeken, T. Gaub, L. Goerlitz, J. Jaeger, R. Loosen, B. Ludewig, et al., *A computational systems biology software platform for multiscale modeling and simulation: Integrating whole-body physiology, disease biology, and molecular reaction networks*, Frontiers in Physiology, 2 (2011), p. 4.

[36] T. M. Elgindi and I.-J. Jeong, *Finite-time singularity formation for strong solutions to the axi-symmetric 3D Euler equations*, arXiv preprint arXiv:1802.09936, (2018).

[37] R. Erban and H. G. Othmer, *From signal transduction to spatial pattern formation in E. coli: A paradigm for multiscale modeling in biology*, Multiscale Modeling & Simulation, 3 (2005), pp. 362–394.

[38] M. Feig, J. Karanicolas, and C. L. Brooks III, *MMTSB Tool Set: Enhanced sampling and multiscale modeling methods for applications in structural biology*, Journal of Molecular Graphics and Modelling, 22 (2004), pp. 377–395.

[39] R. Gendrin, *Effects of heavy ions on microscopic plasma physics in the magnetosphere*, in High-Latitude Space Plasma Physics, Springer, 1983, pp. 415–436.

[40] T. C. Germann and K. Kadau, *Trillion-atom molecular dynamics becomes a reality*, International Journal of Modern Physics C, 19 (2008), pp. 1315–1319.

[41] U. Ghia, K. N. Ghia, and C. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, Journal of Computational Physics, 48 (1982), pp. 387–411.

[42] S. Gottlieb, *On high order strong stability preserving Runge-Kutta and multi step time discretizations*, J. Sci. Comput., (2005), pp. 105–128.

[43] T. Grafke and R. Grauer, *Lagrangian and geometric analysis of finite-time Euler singularities*, arXiv preprint arXiv:1212.0573, (2012).

[44] F. R. Graziani, J. D. Bauer, and M. S. Murillo, *Kinetic theory molecular dynamics and hot dense matter: Theoretical foundations*, Phys. Rev. E, 90 (2014), p. 033104.

[45] J. Haack, C. Hauck, and M. Murillo, *A conservative entropic multispecies BGK model*, J. Stat. Phys., 168 (2017), pp. 826–856.

[46] O. H. Hald and P. Stinis, *Optimal prediction and the rate of decay for solutions of the Euler equations in two and three dimensions*, Proceedings of the National Academy of Sciences, 104 (2007), pp. 6527–6532.

[47] A. Hasegawa, *Plasma instabilities and nonlinear effects*, Springer, 2012.

[48] T. K. Haxton, R. V. Mannige, R. N. Zuckermann, and S. Whitelam, *Modeling sequence-specific polymers using anisotropic coarse-grained sites allows quantitative comparison with experiment*, Journal of Chemical Theory and Computation, 11 (2014), pp. 303–315.

[49] A. Hellander, S. Hellander, and P. Lo?tstedt, *Coupled mesoscopic and microscopic simulation of stochastic reaction-diffusion processes in mixed dimensions*, Multiscale Modeling & Simulation, 10 (2012), pp. 585–611.

[50] N. M. Hoffman, G. B. Zimmerman, K. Molvig, H. G. Rinderknecht, M. J. Rosenberg, B. J. Albright, A. N. Simakov, H. Sio, A. B. Zylstra, M. G. Johnson, F. H. Sguin, J. A. Frenje, C. K. Li, R. D. Petrasso, D. M. Higdon, G. Srinivasan, V. Y. Glebov, C. Stoeckl, W. Seka, and T. C. Sangster, *Approximate models for the ion-kinetic regime in inertial-confinement-fusion capsule implosions*, Physics of Plasmas, 22 (2015), p. 052707.

[51] T. Y. Hou and R. Li, *Dynamic depletion of vortex stretching and non-blowup of the 3-D incompressible Euler equations*, Journal of Nonlinear Science, 16 (2006), pp. 639–664.

[52] J. Huba, *NRL Plasma Formulary*, Naval Research Laboratory, 2013.

[53] P. Isett, *Hölder Continuous Euler Flows in Three Dimensions with Compact Support in Time*, Princeton University Press, 2017.

[54] A. Jameson, *Solution of the Euler equations for two dimensional transonic flow by a multigrid method*, Appl. Math. Comput, 13 (1983), pp. 327–355.

[55] D. Kelly and E. Vanden-Eijnden, *Fluctuations in the heterogeneous multiscale methods for fast-slow systems*, arXiv preprint arXiv:1601.02147, (2016).

[56] M. Khairoutdinov, D. Randall, and C. DeMott, *Simulations of the atmospheric general circulation using a cloud-resolving model as a superparameterization of physical processes*, Journal of the Atmospheric Sciences, 62 (2005), pp. 2136–2154.

[57] M. F. Khairoutdinov and D. A. Randall, *A cloud resolving model as a cloud parameterization in the NCAR Community Climate System Model: Preliminary results*, Geophysical Research Letters, 28 (2001), pp. 3617–3620.

[58] C. Klein et al., *Fourth order time-stepping for low dispersion Korteweg-de Vries and nonlinear Schrödinger equation*, Electronic Transactions on Numerical Analysis, 29 (2008), pp. 116–135.

[59] I. I. Klimontovich, *Kinetic theory of nonideal gas and nonideal plasma*, Moscow Izdatel Nauka, 1 (1975).

[60] U. L. Klimontovich and A. Dobroslavsky, *The kinetic theory of electromagnetic processes*, vol. 10, Springer-Verlag Berlin, 1983.

[61] B. O. Koopman, *Hamiltonian systems and transformation in Hilbert space*, Proceedings of the National Academy of Sciences, 17 (1931), pp. 315–318.

[62] P. A. Lagerstrom, *Matched asymptotic expansions: Ideas and techniques*, vol. 76, Springer Science & Business Media, 2013.

[63] P. Le Tallec and J. Perlat, *Coupling kinetic models with Navier-Stokes equations*, CFD Review, 2 (1998), pp. 833–855.

[64] B. Leimkuhler, E. Noorizadeh, and O. Penrose, *Comparing the efficiencies of stochastic isothermal molecular dynamics methods*, Journal of Statistical Physics, 143 (2011), pp. 921–942.

[65] R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*, vol. 98, SIAM, 2007.

[66] D. Li and Y. G. Sinai, *Blow ups of complex solutions of the 3d Navier-Stokes system and renormalization group method*, Journal of the European Mathematical Society, 10 (2008), pp. 267–313.

[67] X. Li and E. Weinan, *Multiscale modeling of the dynamics of solids at finite temperature*, Journal of the Mechanics and Physics of Solids, 53 (2005), pp. 1650–1685.

[68] G. Luo and T. Y. Hou, *Potentially singular solutions of the 3d axisymmetric Euler equations*, Proceedings of the National Academy of Sciences, 111 (2014), pp. 12968–12973.

[69] E. Lyman, J. Pfaendtner, and G. A. Voth, *Systematic multiscale parameterization of heterogeneous elastic network models of proteins*, Biophysical Journal, 95 (2008), pp. 4183–4192.

[70] R. L. Lysak and C. W. Carlson, *The effect of microscopic turbulence on magnetosphere-ionosphere coupling*, Geophysical Research Letters, 8 (1981), pp. 269–272.

[71] A. J. Majda and A. L. Bertozzi, *Vorticity and incompressible flow*, vol. 27, Cambridge University Press, 2002.

[72] R. Marchand, J. Haynes, G. G. Mace, T. Ackerman, and G. Stephens, *A comparison of simulated cloud radar output from the multiscale modeling framework global climate model with CloudSat cloud radar observations*, Journal of Geophysical Research: Atmospheres, 114 (2009).

[73] C. Marchioro and M. Pulvirenti, *Mathematical theory of incompressible nonviscous fluids*, vol. 96, Springer, 1994.

[74] D. Michta, M. Surh, and F. Graziani, *Kinetic theory molecular dynamics; numerical considerations*, High Energy Density Physics, 9 (2013), pp. 696 – 701.

[75] H. Moffatt, S. Kida, and K. Ohkitani, *Stretched vortices–the sinews of turbulence; large-Reynolds-number asymptotics*, Journal of Fluid Mechanics, 259 (1994), pp. 241–264.

[76] T. Morse, *Energy and momentum exchange between nonequipartition gases*, Physics of Fluids, 6 (1963), pp. 1420–1427.

[77] F. Müller-Plathe, *Coarse-graining in polymer simulation: From the atomistic to the mesoscopic scale and back*, ChemPhysChem, 3 (2002), pp. 754–769.

[78] V. Nikolskiy and V. Stegailov, *Floating-point performance of ARM cores and their efficiency in classical molecular dynamics*, Journal of Physics: Conference Series, 681 (2016), p. 012049.

[79] W. Noid, *Perspective: Coarse-grained models for biomolecular systems*, The Journal of Chemical Physics, 139 (2013), p. 090901.

[80] W. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, and H. C. Andersen, *The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models*, The Journal of Chemical Physics, 128 (2008), p. 244114.

[81] S. T. O'Connell and P. A. Thompson, *Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flows*, Physical Review E, 52 (1995), p. R5792.

[82] P. Orlandi, S. Pirozzoli, and G. Carnevale, *Vortex events in Euler and Navier-Stokes simulations with smooth initial conditions*, Journal of Fluid Mechanics, 690 (2012), pp. 288–320.

[83] B. E. Peigney, O. Larroche, and V. Tikhonchuk, *Ion kinetic effects on the ignition and burn of inertial confinement fusion targets: A multi-scale approach*, Physics of Plasmas, 21 (2014), p. 122709.

[84] T. Plewa, T. Linde, and V. G. Weirs, *Adaptive mesh refinement-theory and applications*, Lecture notes in computational science and engineering, 41 (2005), pp. 3–5.

[85] S. Plimpton, *Computational limits of classical molecular dynamics simulations*, Computational Materials Science, 4 (1995), pp. 361–364.

[86] T. Pohl, T. Pattard, and J. M. Rost, *Kinetic modeling and molecular dynamics simulation of ultracold neutral plasmas including ionic correlations*, Phys. Rev. A, 70 (2004), p. 033416.

[87] J. Price, *Dissertation release of renormalized Mori-Zwanzig git repository.* `http://doi.org/10.5281/zenodo.1256167`, 2018.

[88] J. Price and P. Stinis, *Renormalized reduced order models with memory for long time prediction*, arXiv preprint arXiv:1707.01955, (2017).

[89] ——, *Renormalization and blow-up for the 3d euler equations*, arXiv preprint arXiv:1805.08766, (2018).

[90] J. L. Proctor, S. L. Brunton, B. W. Brunton, and J. Kutz, *Exploiting sparsity and equation-free architectures in complex systems*, The European Physical Journal Special Topics, 223 (2014), pp. 2665–2684.

[91] W. Ren and E. Weinan, *Heterogeneous multiscale method for the modeling of complex fluids and micro-fluidics*, Journal of Computational Physics, 204 (2005), pp. 1–26.

[92] H. G. Rinderknecht, H. Sio, C. K. Li, A. B. Zylstra, M. J. Rosenberg, P. Amendt, J. Delettrez, C. Bellei, J. A. Frenje, M. Gatu Johnson, F. H. Séguin, R. D. Petrasso, R. Betti, V. Y. Glebov, D. D. Meyerhofer, T. C. Sangster, C. Stoeckl, O. Landen, V. A. Smalyuk, S. Wilks, A. Greenwood, and A. Nikroo, *First observations of nonhydrodynamic mix at the fuel-shell interface in shock-driven inertial confinement implosions*, Phys. Rev. Lett., 112 (2014), p. 135001.

[93] D. Roehm, R. S. Pavel, K. Barros, B. Rouet-Leduc, A. L. McPherson, T. C. Germann, and C. Junghans, *Distributed database kriging for adaptive sampling (D2KAS)*, Computer Physics Communications, 192 (2015), pp. 138–147.

[94] M. J. Rosenberg et al., *Investigation of ion kinetic effects in direct-drive exploding-pusher implosions at the NIF*, Phys. Plasmas, 21 (2014), p. 122712.

[95] S. Schnell, R. Grima, and P. K. Maini, *Multiscale modeling in biology: New insights into cancer illustrate how mathematical tools are enhancing the understanding of life from the smallest scale to the grandest*, American Scientist, 95 (2007), pp. 134–142.

[96] M. S. Shell, *The relative entropy is fundamental to multiscale and inverse thermodynamic problems*, J. Chem. Phys, 129 (2008), p. 108.

[97] C.-W. Shu, W.-S. Don, D. Gottlieb, O. Schilling, and L. Jameson, *Numerical convergence study of nearly incompressible, inviscid Taylor-Green vortex flow*, SIAM Journal on Scientific Computing, 24 (2005), pp. 1–27.

[98] K. Steinhaeuser, A. R. Ganguly, and N. V. Chawla, *Multivariate and multiscale dependence in the global climate system revealed through complex networks*, Climate Dynamics, 39 (2012), pp. 889–895.

[99] P. Stinis, *Higher order Mori-Zwanzig models for the Euler equations*, Multiscale Modeling & Simulation, 6 (2007), pp. 741–760.

[100] ——, *A phase transition approach to detecting singularities of partial differential equations*, Communications in Applied Mathematics and Computational Science, 4 (2009), pp. 217–239.

[101] ——, *Mori-Zwanzig reduced models for uncertainty quantification I: Parametric uncertainty*, arXiv preprint arXiv:1211.4285, (2012).

[102] ——, *Mori-Zwanzig reduced models for uncertainty quantification II: Initial condition uncertainty*, arXiv preprint arXiv:1212.6360, (2012).

[103] ——, *Numerical computation of solutions of the critical nonlinear Schrödinger equation after the singularity*, Multiscale Modeling & Simulation, 10 (2012), pp. 48–60.

[104] ——, *Renormalized reduced models for singular PDEs*, Communications in Applied Mathematics and Computational Science, 8 (2013), pp. 39–66.

[105] ——, *Renormalized Mori-Zwanzig-reduced models for systems without scale separation*, in Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 471, The Royal Society, 2015, p. 20140446.

[106] Y. Sugita and Y. Okamoto, *Replica-exchange molecular dynamics method for protein folding*, Chemical Physics Letters, 314 (1999), pp. 141–151.

[107] W.-K. Tao, J.-D. Chern, R. Atlas, D. Randall, M. Khairoutdinov, J.-L. Li, D. E. Waliser, A. Hou, X. Lin, C. Peters-Lidard, et al., *A multiscale modeling system: Developments, applications, and critical issues*, Bulletin of the American Meteorological Society, 90 (2009), pp. 515–534.

[108] R. Teyssier, *Cosmological hydrodynamics with adaptive mesh refinement-a new high resolution code called RAMSES*, Astronomy & Astrophysics, 385 (2002), pp. 337–364.

[109] M. Tuckerman, B. J. Berne, and G. J. Martyna, *Reversible multiple time scale molecular dynamics*, The Journal of Chemical Physics, 97 (1992), pp. 1990–2001.

[110] M. E. Tuckerman and G. J. Martyna, *Understanding modern molecular dynamics: Techniques and applications*, The Journal of Physical Chemistry B, 104 (2000), pp. 159–178.

[111] S. Venakides, *The zero dispersion limit of the Korteweg-de Vries equation with periodic initial data*, Transactions of the American Mathematical Society, (1987), pp. 189–226.

[112] A. Vlasov, *The vibrational properties of an electron gas*, Physics-Uspekhi, 10 (1968), pp. 721–733.

[113] M. Wang, S. Ghan, R. Easter, M. Ovchinnikov, X. Liu, E. Kassianov, Y. Qian, W. Gustafson Jr, V. Larson, D. Schanen, et al., *The multi-scale aerosol-climate model PNNL-MMF: Model description and evaluation*, Geoscientific Model Development, 4 (2011), p. 137.

[114] M. Wang, S. Ghan, M. Ovchinnikov, X. Liu, R. Easter, E. Kassianov, Y. Qian, and H. Morrison, *Aerosol indirect effects in a multi-scale aerosol-climate model PNNL-MMF*, Atmospheric Chemistry and Physics, 11 (2011), p. 5431.

[115] Y. Wang, M. Wang, R. Zhang, S. J. Ghan, Y. Lin, J. Hu, B. Pan, M. Levy, J. H. Jiang, and M. J. Molina, *Assessing the effects of anthropogenic aerosols on Pacific storm track using a multiscale global climate model*, Proceedings of the National Academy of Sciences, 111 (2014), pp. 6894–6899.

[116] J. Q. Weare, *Efficient conditional path sampling of stochastic differential equations*, University of California, Berkeley, 2007.

[117] E. Weinan, B. Engquist, et al., *The heterogeneous multiscale methods*, Communications in Mathematical Sciences, 1 (2003), pp. 87–132.

[118] E. Weinan, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden, *Heterogeneous multiscale methods: A review*, Commun. Comput. Phys, 2 (2007), pp. 367–450.

[119] L. Yin, B. J. Albright, W. Taitano, E. L. Vold, L. Chacón, and A. N. Simakov, *Plasma kinetic effects on interfacial mix*, Phys. Plasmas, 23 (2016), p. 112302.

[120] R. Zwanzig, *Memory effects in irreversible thermodynamics*, Physical Review, 124 (1961), p. 983.

VITA

Jacob Price received his Bachelor of Arts degrees in Mathematics and Physics with Department Honors from both departments and Summa cum Laude distinction from Kalamazoo College in 2012. Subsequently, Jacob joined the Department of Applied Mathematics at the University of Washington and obtained a Masters of Science in Applied Mathematics in 2013. He conducted research at Los Alamos National Laboratory in 2015 and 2016 and will be joining the faculty at the University of Puget Sound in 2018 as a tenure-track Assistant Professor of Mathematics.