Variational Multiscale Modeling and Memory Effects in Turbulent Flow Simulations

by
Eric J. Parish

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2018

Doctoral Committee:

Associate Professor Karthik Duraisamy, Chair
Associate Professor Krzysztof Fidkowski
Associate Professor Eric Johnsen
Professor Venkat Raman
Professor Philip Roe

Eric J. Parish
parish@umich.edu
ORCID iD: 0000-0002-5478-2998

For my Grandma. She would not understand any of this, but would love it anyway.

ACKNOWLEDGEMENTS

First and foremost, I wish to thank my advisor, Professor Karthik Duraisamy. I have learned so much throughout my Ph.D. and you have been an integral part of that. I am grateful to you for giving me the flexibility to pursue research that interests me as well as for providing intellectual support and insight along the way. This dissertation would not have been possible without you. I also wish to thank you for your guidance outside of research. You have been incredibly supportive and helpful over this past year as I have started to make the transition to life after graduate school.

In addition, I wish to thank the many other people who have helped me along the way:

- To my friends at Michigan. You have helped me to reach this point in many ways. From you I have not only learned what it means to be a good researcher, but also what it means to be a good friend.

- To my lab mates at Michigan for many insightful discussions as well as for all the fun times we have had outside of the lab.

- To my undergraduate advisor, Professor Dellenback. Your dedication to teaching and interest in your students changed my career path. Taking your Thermodynamics course led to me changing majors, helped me find my research interests, and was one of the paradigm shifts in my life that led me to this

point.

- To another mentor in my undergraduate degree, Professor Michael Stoellinger. You introduced me to research and were a major reason I became interested in CFD. I thank you for taking the time to teach me and for helping me to find my path.

- To the members of my dissertation committee: Professor's Krzysztof Fidkowski, Eric Johnsen, Venkat Raman, and Phil Roe. Thank you for your time, advice, and guidance along the way.

- To Panos Stinis for his work on Mori-Zwanzig. Without this foundation, this dissertation would not be possible.

- To Rachel. You have given me the support to make it through this final stretch of the Ph.D. I could write so many things about you here, but more than anything I want to thank you for making me smile every day.

- To the many, many other friends, family, and mentors who have helped me along the way.

Lastly I want to thank my family. You have been there for me along every step of the way. From watching me play golf in my undergraduate degree to proofreading my journal papers, your support has been unwavering. Matt - you give me perspective everyday about what it means to be a good brother and person. Mom and Dad - you are always there for me when I need someone to talk to. Our conversations on Sunday mornings were one of the highlights of every week at Michigan. I love you all and am blessed to have you.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Effective computational models of multiscale problems have to account for the impact of unresolved physics on the resolved scales. This dissertation advances our fundamental understanding of multiscale models and develops a mathematically rigorous closure modeling framework by combining the Mori-Zwanzig (MZ) formalism of statistical mechanics with the variational multiscale (VMS) method. This approach leverages scale-separation projectors as well as phase-space projectors to provide a systematic modeling approach that is applicable to complex non-linear partial differential equations.

The MZ-VMS framework leads to a closure term that is non-local in time and appears as a convolution or memory integral. The resulting non-Markovian system is used as a starting point for model development. Several new insights are uncovered: It is shown that unresolved scales lead to memory effects that are driven by an orthogonal projection of the coarse-scale residual and, in the case of finite elements, inter-element jumps. Connections between MZ-based methods, artificial viscosity, and VMS models are explored.

The MZ-VMS framework is investigated in the context of turbulent flows. Large eddy simulations of Burgers' equation, turbulent flows, and magnetohydrodynamic turbulence using spectral and discontinuous Galerkin methods are explored. In the spectral method case, we show that MZ-VMS models lead to substantial improvements in the prediction of coarse-grained quantities of interest. Applications to

discontinuous Galerkin methods show that modern flux schemes can inherently capture memory effects, and that it is possible to guarantee non-linear stability and conservation via the MZ-VMS approach. We conclude by demonstrating how ideas from MZ-VMS can be adapted for shock-capturing and filtering methods.

CHAPTER I

Introduction

1.1 Computational Physics for Multiscale Problems

Computational physics refers to the field of study where one uses numerical methods and algorithms to solve physical problems. The emergence of computational physics over the past half century has led to significant advancements in technology, engineering products, and our fundamental understanding of physical processes. The field has provided scientists with the tools to predict and analyze complex systems via numerical approximations.

The simulation of multiscale problems remains a challenging topic in the computational physics community. Multiscale systems, which are characterized by a wide range of length and time scales, are omnipresent in science and engineering. Multiscale problems are prevalent in turbulent flows, magnetohydrodynamics, quantum mechanics, fracture dynamics, and many other fields. Though the governing equations of multiscale systems are often known, pursuing numerical solutions is made challenging by the large number of degrees of freedom (often on the order of billions) which are required to accurately characterize the system. A large number of degrees of freedom directly translates to a large computational and financial cost. For many multiscale problems of interest, this cost is so high that numerical solutions are in-

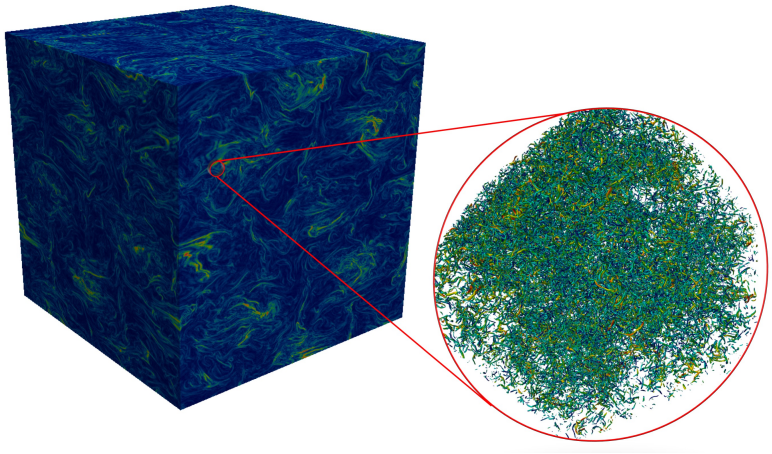


Figure 1.1: Turbulent flows, which display a wide range of length and time scales, are prominent examples of multiscale systems.

feasible; even on the most powerful supercomputers in the world. This issue has led to the development of multiscale modeling techniques. These techniques seek to approximately solve multiscale systems to a desired level of accuracy at a much lower computational cost.

1.2 Turbulence

This dissertation presents a general multiscale modeling strategy and evaluates it in the context of high Reynolds number turbulent flows. Turbulence refers to the state of a fluid (i.e. a gas or liquid) in a regime that is characterized by highly non-linear and chaotic behavior. Turbulence is present in many practical applications, such as flight aerodynamics and weather forecasting, and is a pacing research area in fluid mechanics. While turbulence is governed by a set of partial differential equations that have been known for centuries (the Navier-Stokes equations), to this date no theory exists which allows for a compact parameterization of turbulence. Richard Feynman has referred to turbulence as "the most important unsolved problem of classical physics" [37].

The simulation of turbulent flows is a prototypical multiscale problem. Turbulence is understood to have a continuous cascade of scales and energy [55]. This cascade begins with the large energetic flow structures and continues until the turbulent fluctuations are dissipated into heat at the finest scales. The range of scales present in this cascade is driven by the balance of inertial and viscous forces, the ratio of which is referred to as the Reynolds number. As the Reynolds number increases (i.e. inertial forces dominate), so does the range of scales. Most problems of engineering and scientific interest exist in high Reynolds number regimes. Due to this disparity in scales at high Reynolds numbers, numerical simulations that resolve all relevant scales are prohibitively expensive. This fact and other challenges in computational physics have spurred successful research efforts in areas such as computational hardware, distributed memory computing, linear and non-linear solver technologies, and algorithmic development. Despite these algorithmic and technological advances, direct numerical simulation (DNS) methodologies remain computationally infeasible for the high Reynolds number turbulent flows commonly encountered in real world applications [65].

1.3 Governing Equations of Motion and Closure

The aforementioned challenges associated with the solution of high Reynolds number turbulent flows has not prohibited scientists and engineers from using numerical tools to tackle real world problems. Generally, the solution of high Reynolds number turbulent flows is made tractable by developing a set of surrogate equations which contain a reduced range of scales. This reduction in scales is typically obtained by applying a filtering operation to the equations of motion. To describe this, consider

the set of partial differential equations,

$$\frac{\partial u}{\partial t} + \mathcal{R}(u) = 0 \qquad\qquad x \in \Omega, t \in (0, t), \tag{1.1}$$

subject to appropriate boundary and initial conditions and where $\mathcal{R}$ is a potentially non-linear differential operator. Typically, Eq. 1.1 contains a wide range of scales. This range of scales can be reduced by applying a filter to Eq. 1.1,

$$\mathcal{G}\frac{\partial u}{\partial t} + \mathcal{G}\mathcal{R}(u) = 0. \tag{1.2}$$

Defining the filtered variable as $\mathcal{G}u = \tilde{u}$, adding and subtracting $\mathcal{R}(\tilde{u})$, and assuming time-invariance of the filter leads to the filtered equations of motion,

$$\frac{\partial \tilde{u}}{\partial t} + \mathcal{R}(\tilde{u}) + \Big[\mathcal{G}\mathcal{R}(u) - \mathcal{R}(\tilde{u})\Big] = 0. \tag{1.3}$$

Equation 1.3 describes the evolution of the filtered variable. The idea is that the range of scales present in Eq. 1.3 is significantly less than what is present in Eq. 1.1. The reader will note, however, that the bracketed term in Eq. 1.3 contains the full state, u. This is an unknown variable in the filtered problem and results in the so-called *closure problem*. This unclosed term is typically replaced by a *subgrid-scale model* that only depends on the filtered state,

$$\frac{\partial \tilde{u}}{\partial t} + \mathcal{R}(\tilde{u}) + \mathcal{M}(\tilde{u}) = 0. \tag{1.4}$$

The type of filtering operation applied to the equations of motion and type of subgrid-scale model defines the modeled approach. The Reynolds-averaged Navier-Stokes (RANS) approach, large eddy simulation (LES) approach, and variational multiscale method constitute the most common modeling strategies in fluid mechanics.

1.4 Reynolds-Averaged Navier-Stokes

The RANS approach is the most widely used method for the practical simulation of turbulent flows. RANS methods, which are derived from applying an averaging op-

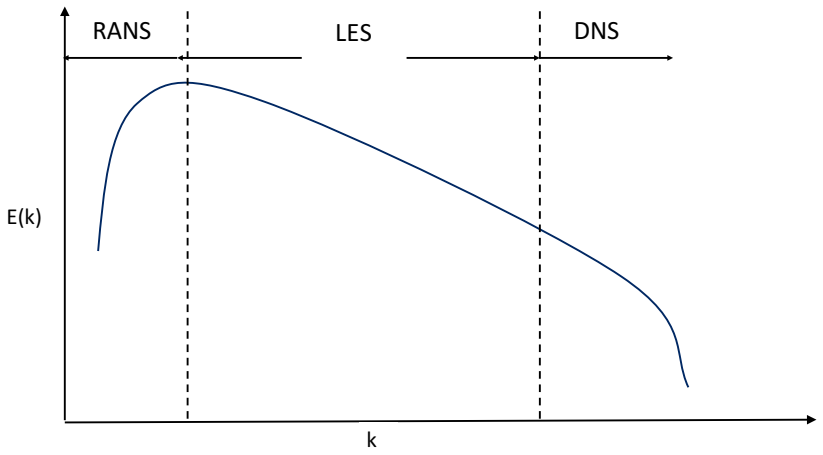


Figure 1.2: Graphical illustration of the energy spectrum in a turbulent flow. The wavenumber is given by k, while $E(k)$ denotes the energy. RANS methods model all turbulence and only resolve the large-scale mean flow structures. LES methods resolve into the inertial subrange of the turbulence spectrum and model only the fine-scales. DNS methods resolve all scales.

erator (such as ensemble or temporal averaging) to the equations of motion, model all scales of turbulence (see Figure 1.2). RANS methods require what is referred to as a turbulence/subgrid/closure model. These models augment the equations of motion with additional algebraic relations and/or transport equations and attempt to model the impact of the turbulent fluctuations on the mean flow. RANS models, which are typically developed from theory and experimental data, are generally "tuned" such that they accurately reproduce mean statistics of canonical turbulent flows. RANS-based methods have been shown to be successful for computing first order statistics for a variety of problems, including external aerodynamics [13, 65], internal flows [101], and reacting flows [35]. These methods, however, are known to be inaccurate for flows that deviate from conditions for which the models were developed and tuned [65, 33]. The limitations of RANS methods are a current topic of debate in the community [34]. RANS methods will not be considered in this dissertation.

1.5 Large Eddy Simulation

Advances in computational hardware and the lack of accuracy of RANS methods has led to great interest in large eddy simulation. Large eddy simulation approaches provide an intermediary level of accuracy between RANS-based methods and direct numerical simulation. Large eddy simulation approaches, which are derived by applying a low-pass filter to the equations of motion [88], attempt to resolve the coherent, energetic scales of a turbulent flow while modeling the unresolved, "subgrid" turbulent structures (see again Figure 1.2). In contrast, RANS-based approaches directly model all scales of turbulence. This dissertation will focus on the LES approach.

A variety of LES methodologies exist in the literature [66]. These methods are typically devised through the use of both filtering operations and subgrid-scale models. They can be broadly classified as either *explicit methods* or *implicit methods*. Explicit methods rely on the inclusion of additional subgrid-scale models and/or filtering operations to control numerical error and accuracy. Implicit methods do not include additional subgrid-scale models. Instead they rely on the inherent properties of a numerical scheme to control error and maintain stability.

1.6 The Variational Multiscale Method

In the late 1990's, Tom Hughes established a modeling paradigm known as the variational multiscale method. The variational multiscale method (VMS) [48, 49] is analogous to large eddy simulation and provides a general starting point to derive closure/subgrid/stabilization models. The VMS procedure, which starts from the weighted residual form of the governing equations, is centered around a sum decomposition of the solution u in terms of the resolved coarse scales $\tilde{u}$ and unresolved or fine scales u'. The key challenge in VMS is to obtain a representation for u' in

terms of $\tilde{u}$. This is typically done by virtue of approximated Green's functions [24] or physics-based subgrid-scale models [50]. Given the approximation to the fine scales, the impact of the fine scales on the coarse scales can then be numerically computed. In Hughes' pioneering work, it was demonstrated that classic stabilization techniques, such as adjoint stabilization and streamline upwind Petrov-Galerkin, can be derived via specific approximations to the Green's function and manipulations of the multiscale equations. The variational multiscale method has since gained significant attention, and reviews may be found in Refs [48, 49, 51, 24].

The VMS procedure - by itself - does not eliminate the closure problem. Representation of the fine-scale state in terms of the coarse-scale state, by virtue of a Green's function or otherwise, is required. For the non-linear unsteady problems commonly encountered in science and engineering, the development of such relations has proved challenging. The vast majority of methods used today are still based on Hughes' pioneering work where the VMS procedure was presented for a linear Dirichlet problem. There is a significant need to further investigate the development of VMS models.

1.7 The Mori-Zwanzig Formalism

A less publicized, but conceptually similar approach to VMS is the optimal prediction framework developed by Chorin and co-workers [20, 22, 42, 21]. Chorin's optimal prediction framework, which is a reformulation of the Mori-Zwanzig (MZ) formalism of statistical mechanics [69, 106], can be viewed as a model order reduction strategy for ordinary differential equations. The framework centers around the use of projection operators that separate the phase space of an ordinary differential equation into resolved and unresolved subspaces. Using this decomposition, a high-

dimensional non-linear Markovian dynamical system can be recast into an equivalent, lower-dimensional non-Markovian system. This formulation is mathematically exact. In the lower-dimensional system, which is commonly referred to as the generalized Langevin equation (GLE), the effect of the fine scales on the coarse scales is non-local in time and appears as a convolution integral. This term is typically referred to as memory. The memory term emerging from the MZ formalism is analogous to the Green's function emerging from the VMS procedure. While the memory integral is intractable in general non-linear problems, it serves as a mathematically rigorous starting point for the construction of closure models. The MZ formalism is an appealing approach for LES as – in theory – it allows for the construction of exact closure models. In a more practical sense, it provides a systematic framework for developing models.

In addition to the pioneering work of Chorin and collaborators, a significant body of research on Mori-Zwanzig approaches exists in the literature. The MZ formalism has been examined extensively in fields such as molecular dynamics [58] and uncertainty quantification [61]. In the context of the solution of partial differential equations, the main body of work has examined approximations to the memory term and their application to the semi-discrete systems emerging from Fourier-Galerkin spatial discretizations. Stinis and coworkers [46, 92, 94, 93, 82] have developed several models for approximating the memory and have examined their performance on Burgers' equation and the Euler equations. In the same spirit, the work of Zhu, Dominy, and Venturi [105, 104] examines additional approximations to the memory as well as error estimates for various MZ models.

The Mori-Zwanzig procedure, however, has received minimal attention as a *practical tool* for closure modeling in the context of the numerical solution of partial

differential equations. The vast majority of the work has been undertaken on either idealized systems or problems that pertain to the statistical mechanics community. Prior to this dissertation work, only one paper in the literature has examined the application of an MZ-based model to the Navier-Stokes equations [15]. This work was performed in the context of spectral methods and its extension to any practical technique was unclear. In fluid dynamics, MZ-related modeling has been pursued exclusively on the semi-discrete systems emerging from Fourier-Galerkin spectral methods. In addition to the aforementioned work on the Navier-Stokes equations, MZ-based models have been investigated in the context of Fourier-Galerkin simulations of Burgers' equation [7, 95] and the Euler equations [92, 46]. This body of work has demonstrated the potential of using the MZ procedure for model development. The Fourier-Galerkin spectral method, however, is limited to canonical problems and periodic domains.

The extension of the Mori-Zwanzig procedure to more general numerical methods has remained unclear. This is largely due to the challenges in formulating the MZ framework in a systematic manner for the ordinary differential equations that emerge from the numerical discretization of partial differential equations. These challenges stem from the fact that the key concept in MZ is a tractable decomposition of the discrete unknowns into a coarse-scale resolved set and a fine-scale unresolved set. While this separation is clear in the context of Fourier-Galerkin methods, it is less straightforward for more general numerical schemes. Another road-block to the further development of MZ methods is a lack of robust models to the memory and insight into such models.

1.8 Dissertation Setting

This dissertation work seeks to advance the state of the art in our understanding and modeling of under-resolved simulations of turbulent flow. This is achieved through addressing the following questions:

1. How can ideas from Mori-Zwanzig be used to develop a formal modeling approach?
2. Can commonalities be drawn between MZ-based models and more established methods (such as residual-based methods)?
3. How do MZ-based methods compare to more established subgrid-scale models?
4. What are the shortcomings in existing MZ-methods and how can they be improved?

This work presents a unique and unifying perspective on under-resolution in numerical methods, closure, and memory effects.

1.9 Contributions

They key contributions and publications stemming from this dissertation are:

1. A formulation of of Chorin's optimal prediction framework for variational multiscale-type discretizations of partial differential equations [76].
2. A rigorous investigation of the performance of existing MZ models for Fourier-Galerkin simulations of turbulent flows [75, 73].
3. An establishment of connections between MZ-based methods and residual-based closures [76].
4. Application of MZ-based methods to the discontinuous Galerkin method and an establishment of connections between MZ-based methods, upwinding, and

artificial viscosity. This work marks the first time that MZ-based approaches have been applied to finite element methods [76].

5. A demonstration that, in discontinuous Galerkin simulations, unclosed effects lead to a memory term that is driven by the coarse-scale residual and inter-element jumps [76].
6. The development of an autonomous dynamic MZ-VMS model that combines ideas from the MZ and fluid mechanics communities [74, 72].
7. A formulation of an auxiliary ODE system that can be used to approximate the orthogonal dynamics [43].
8. An investigation of MZ-VMS methods as resolution indicators and limiters. The MZ-VMS memory term is used as an indicator to determine if a solution is under-resolved and if limiting is needed.
9. The development of an open-source spectral method solver for the triply periodic Navier-Stokes equations and an open-source discontinuous Galerkin solver for the compressible Navier-Stokes equations.

These contributions are additionally presented in the following publications:

1. Parish E.J. and Duraisamy K., "A Unified Framework for Multiscale Modeling using Mori-Zwanzig and the Variational Multiscale Method," *arXiv 1712.09669v1*, 2017.

2. Parish, E. J. and Duraisamy K., "A Dynamic Subgrid Scale Model for Large Eddy Simulations Based on the Mori-Zwanzig Formalism", *Journal of Computational Physics*, 2017.
3. Gouasmi, A., Parish, E.J., and Duraisamy, K., "Characterizing Memory Effects

in Coarse-Grained Nonlinear Systems Using the Mori-Zwanzig formalism," *Proc. Royal Society A*, 2017

4. Parish E.J. and Duraisamy K. "Non-Markovian closure models for large eddy simulations using the Mori-Zwanzig formalism," *Phys. Rev. Fluids*, 2017.

1.10 Outline

The structure of this dissertation is as follows: Chapter II will discuss the mathematical foundations of large eddy simulation, the Galerkin method, and the variational multiscale method. Chapter III will outline the Mori-Zwanzig formalism and variational multiscale framework (MZ-VMS). This framework will integrate the variational multiscale method with the Mori-Zwanzig formalism. The memory and orthogonal dynamics will be discussed. Chapter IV will outline subgrid-scale models based on the Mori-Zwanzig formalism. Chapter V will investigate the performance of MZ-VMS models applied to spectral methods. The Burgers' equation, incompressible Navier-Stokes equations, and equations of ideal magnetohydrodynamics will be considered. Chapter VI will investigate MZ-VMS models applied to finite element methods. The focus will be on the discontinuous Galerkin approach. Burgers' equation and the Navier-Stokes equations will again be considered. Conclusions and perspectives will be provided in Chapter VII

CHAPTER II

Galerkin Methods and Multiscale Formulations

2.1 Multiscale Partial Differential Equations

This dissertation focuses on the numerical solution of the set of multiscale partial differential equations,

$$\frac{\partial u}{\partial t} + \mathcal{R}(u) = f; \qquad x \in \Omega, t \in (0,t); \qquad \Omega \subset R^d, \tag{2.1}$$

subject to appropriate boundary and initial conditions. Note that $\mathcal{R}$ is a potentially non-linear differential operator. A variety of numerical methods have been developed with the goal of solving Eq. 2.1. Finite difference, finite volume, finite element, and spectral methods are several such examples. Each numerical method has its own inherent advantages and disadvantages which may include speed, accuracy, and robustness. All these numerical methods consist of transforming the continuous set of partial differential equations into a semi-discrete system of ordinary differential equations.

This dissertation focuses on the case where the physics described by Eq. 2.1 are multiscale. In these cases, the dimensionality of the algebraic system resulting from spatial discretization becomes prohibitively large. A practical example of this is the simulation of turbulent flows at high Reynolds numbers. To circumvent this issue, we consider the large eddy simulation approach.

2.2 Large Eddy Simulation

Large eddy simulation (LES) originates from the field of fluid mechanics and refers to the general technique where the largest scales of a partial differential equation are directly resolved on a numerical grid while the smaller "subgrid" eddies are modeled. The LES approach has its origins in Smagorinsky's pioneering work on atmospheric flows [88] in the mid 1960's. Further work by Deardoff [31] and Schumann [87] in the 1970's extended LES to the engineering community. The explosion in computing power since Smagorinsky's initial work has led to great interest in LES. Topical areas such as numerical methods, subgrid-scale modeling, and mesh generation have all seen substantial advances.

The LES technique attempts to alleviate the cumbersome numerical cost of solving Eq. 2.1 by attempting to solve Eq. 2.1 only for the large scale features. The basis of LES is the decomposition of the flow field into a resolved *coarse-scale* and unresolved *fine-scale* component (see Figure 2.1). The methodology through which this decomposition is obtained distinguishes different LES methods. Traditionally, the separation is obtained through the use of a filter. Gaussian and sharp-spectral cutoff filters are two such examples.

2.2.1 Filtered Equations of Motion

The governing equations for LES are typically obtained by applying a filter $\mathcal{G}(\mathbf{x}, \Delta)$, where Δ is the filter width, to Eq. 2.1,

$$\mathcal{G}\frac{\partial u}{\partial t} + \mathcal{G}\mathcal{R}(u) = \mathcal{G}f. \tag{2.2}$$

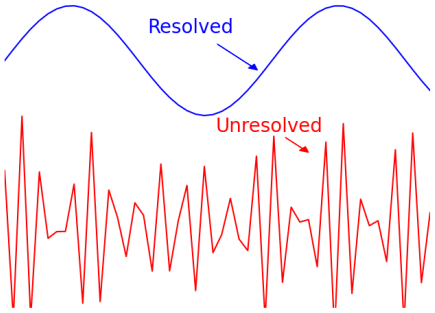


Figure 2.1: Decomposition of a velocity signal into the low-frequency resolved coarse scales and high-frequency unresolved fine scales. Figure concept adapted from [48].

Defining the filtered variable as $\mathcal{G}u = \tilde{u}$, adding and subtracting $\mathcal{R}(\tilde{u})$, and assuming time-invariance of the filter leads to the filtered equations of motion,

$$\frac{\partial \tilde{u}}{\partial t} + \mathcal{R}(\tilde{u}) + \Big[\mathcal{G}\mathcal{R}(u) - \mathcal{R}(\tilde{u})\Big] = \mathcal{G}f. \tag{2.3}$$

Equation 2.3 describes the dynamics of the filtered state, $\tilde{u}$. The reader will note that the bracketed term in Eq. 2.3 depends on both the filtered state $\tilde{u}$ and the full state, u. This is problematic as Eq. 2.3 only evolves the filtered state. The complete state u is unknown. This is referred to as the closure problem. Typically, some model is devised to approximate the impact of the fine scales on the coarse scales to obtain a closed system,

$$\frac{\partial \tilde{u}}{\partial t} + \mathcal{R}(\tilde{u}) + \mathcal{M}(\tilde{u}) = \mathcal{G}f. \tag{2.4}$$

The model term $\mathcal{M}(\tilde{u})$ is referred to as a *subgrid-scale model* and trys to approximate

$$\mathcal{M}(\tilde{u}) \approx \Big[\mathcal{G}\mathcal{R}(u) - \mathcal{R}(\tilde{u})\Big]. \tag{2.5}$$

The construction of this model $\mathcal{M}$ will be a primary focus of this dissertation. It is noted that a common alternative to Eq. 2.3 is obtained by adding and subtracting

$\mathcal{GR}(\tilde{u})$ to Eq. 2.2 (opposed to $\mathcal{R}(\tilde{u})$) to obtain,

$$\frac{\partial \tilde{u}}{\partial t} + \mathcal{GR}(\tilde{u}) + \Big[\mathcal{GR}(u) - \mathcal{GR}(\tilde{u})\Big] = \mathcal{G}f. \tag{2.6}$$

In this case, the subgrid-scale model is responsible for approximating $\mathcal{GM}(\tilde{u}) \approx \mathcal{GR}(u) - \mathcal{GR}(\tilde{u})$.

2.3 Implicit and Explicit LES Methodologies

A variety of LES methodologies exist. These methods differ in how filtering operations and subgrid-scale models are applied. Almost all LES methods are in one of four categories:

- Implicitly filtered explicitly modeled LES
- Explicitly filtered explicitly modeled LES
- Explicitly filtered implicitly modeled LES
- Implicitly filtered implicitly modeled LES

The first implicit/explicit refers to the application of filtering operators, while the second refers to the use of subgrid-scale models. This terminology is non-standard. In the literature LES methods are typically referred to as either explicit or implicit methods. In general, explicit methods refer to the case where a subgrid-scale model is used, while implicit methods refer to the case where no subgrid-scale model is used (see, for instance, [80, 30]). Implicit and explicit LES should not be confused with implicitly filtered and explicitly filtered LES. Implicitly filtered LES refers to the case where the grid and numerics are understood to filter the LES solution, while explicitly filtered LES refers to the case where an explicit filtering operation is applied [9, 64]. It is noted that explicitly filtered LES with no subgrid-scale model is sometimes referred to as implicit LES (as in [80]).

To illustrate each method, we introduce Burgers' equation. Burgers' equation is given by,

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0 \qquad x \in \Omega. \tag{2.7}$$

Burgers' equation is a one-dimensional analogue to the Euler equations of gas dynamics and displays similarities in the nonlinear convective term. Application of a time-invariant commuting spatial filter (such as a sharp-spectral cutoff filter), adding and subtracting $\frac{1}{2}\partial_x \tilde{u}^2$, and rearranging leads to the filtered Burgers equation,

$$\frac{\partial \tilde{u}}{\partial t} + \frac{1}{2}\frac{\partial \tilde{u}^2}{\partial x} = \frac{1}{2}\frac{\partial}{\partial x}\left(\tilde{u}^2 - \mathcal{G}u^2\right). \tag{2.8}$$

An alternative formulation is obtained by applying a time-invariant commuting spatial filter, adding and subtracting $\frac{1}{2}\mathcal{G}\partial_x \tilde{u}^2$, and rearranging,

$$\frac{\partial \tilde{u}}{\partial t} + \mathcal{G}\frac{1}{2}\frac{\partial \tilde{u}^2}{\partial x} = \frac{1}{2}\mathcal{G}\frac{\partial}{\partial x}\left(\tilde{u}^2 - u^2\right). \tag{2.9}$$

Equation 2.8 can be loosely viewed as an implicitly filtered formulation, while 2.9 can be viewed as an explicitly filtered formulation. Note that Equations 2.8 and 2.9 are manipulations of the same equation, and are thus equivalent. As is seen in the following discussion, however, their practical implementation differs.

2.3.1 Implicitly Filtered Explicitly Modeled Large Eddy Simulation

The implicitly filtered explicitly modeled LES approach consists of solving the equations of motion on an under-resolved grid and adding additional terms that are responsible for parameterizing how the unresolved turbulence effects the large scale grid-resolved flow. Implicitly filtered explicitly modeled LES is often referred to as *explicit LES*. This is in reference to the fact that an explicit subgrid-scale turbulence model is used.

To illustrate the concept, consider a forward in time Euler update to Eq. 2.8 using an implicitly filtered explicitly modeled LES approach with a central in space finite difference method,

$$\tilde{u}_j^{n+1} = \tilde{u}_j^n - \Delta t \left[\frac{1}{2} \frac{\tilde{u}_{j+1}^{n^2} - \tilde{u}_{j-1}^{n^2}}{2\Delta x} + \mathcal{M}(\tilde{u}) \right]. \tag{2.10}$$

The reader will note the presence of the subgrid-scale model $\mathcal{M}(\tilde{u})$. They will additionally see that an explicit filtering process does not appear in the update. This is somewhat unsettling as the governing equations of LES are defined by a filtering operation. In the implicitly filtered explicitly modeled LES approach, the finite support of the numerical grid and discrete operators are viewed as an *implicit* filter.

Implicit Filters in Finite Difference Methods

The notional argument of an implicit filter is often presented by showing the equivalence between a finite difference derivative and an exact derivative of a filtered quantity,

$$\frac{\partial u}{\partial x} \approx \frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} = \frac{\partial}{\partial x} \left[\frac{1}{2\Delta x} \int_{x-\Delta x}^{x+\Delta x} u dx \right] = \frac{\partial \tilde{u}}{\partial x}. \tag{2.11}$$

While the above shows how a discrete derivative may act as an implicit filter, the argument is only notional [64]. For instance, Eq. 2.11 shows an equivalence between the finite difference of an *unfiltered variable* and the exact derivative of a *filtered variable*. An additional issue with the above argument is its extension to multidimensional problems. The filtered Navier-Stokes equations in three dimensions, for instance, require a three-dimensional filter. The filters associated with a finite difference operator, in contrast, are all one-dimensional functions. As a result, the "implicitly filtered" Navier-Stokes equations that are obtained by virtue of finite difference operators cannot be formally derived from the Navier-Stokes equations themselves.

2.3.2 Explicitly Filtered Explicitly Modeled Large Eddy Simulation

With the theoretical issues raised above regarding implicitly filtered LES, particularly regarding implicit filtering in the context of finite difference methods, *explicitly filtered explicitly modeled LES* has received considerable attention from the research community. In explicitly filtered explicitly modeled LES, an explicit filtering operation is used in addition to an explicit subgrid-scale model. This explicit filtering operation is applied in a variety of flavors, with the most common being the application of an explicit filter to the non-linear convective terms or the application of an explicit filter to the entire solution after each time-step [80]. To illustrate this first approach, we consider the update to Eq. 2.9 using a forward Euler method (note that the approach can be used with any time-marching method) and the explicitly filtered explicitly modeled LES approach,

$$\tilde{u}_j^{n+1} = \tilde{u}_j^n - \mathcal{G}\Delta t\left[\frac{1}{2}\frac{\tilde{u}_{j+1}^{n^2} - \tilde{u}_{j-1}^{n^2}}{2\Delta x} + \mathcal{M}(\tilde{u})\right]. \tag{2.12}$$

In contrast to Eq. 2.10, Eq. 2.12 explicitly contains a filtering operation. The explicit filtering operation provides more control over numerical errors and allows for the filtering operation to be decoupled from the mesh. Explicit filtering, however, can reduce the effective resolution of a numerical method. This in turn can degrade the accuracy of the numerical method.

2.3.3 Explicitly Filtered Implicitly Modeled Large Eddy Simulation

Explicitly filtered implicitly modeled LES is again based on the application of an explicit filter to the solution. The explicitly filtered implicitly modeled LES method *does not* make use of an explicit subgrid-scale model. Instead, it relies on the explicit filtering procedure and numerics to act as an *implicit subgrid-scale model* [8]. To demonstrate this, we again consider the discrete update to Eq. 2.9 with an explicitly

filtered implicitly modeled LES approach,

$$\tilde{u}_j^{n+1} = \tilde{u}_j^n - \mathcal{G}\Delta t\left[\frac{1}{2}\frac{\tilde{u}_j^{n^2} - \tilde{u}_{j-1}^{n^2}}{\Delta x}\right]. \tag{2.13}$$

The reader will observe that Eq. 2.13 does not contain a model term $\mathcal{M}(\tilde{u})$. The reader will additionally observe that the central difference method used in Eqns. 2.10 and 2.12 has been replaced with a first order "upwind" difference method[1]. This numerical method acts as an implicit subgrid-scale model. The notion of how numerics can act as an implicit subgrid-scale model (note that Eq. 2.11 shows how numerics can act as an implicit filter, opposed to an implicit subgrid-scale model) is apparent by examining the leading order truncation term in a first order finite difference via a Taylor series expansion,

$$\frac{u(x+\Delta x) - u(x)}{\Delta x} = \frac{\partial u}{\partial x} + \Delta x\frac{\partial u^2}{\partial x^2} + \mathcal{O}(\Delta x^2). \tag{2.14}$$

In Eq. 2.14, it is seen that the first order finite difference approximation is equivalent to an exact derivative augmented with a first order diffusive term (plus higher order terms). The diffusive term is the leading order error term and acts as a type of *implicit subgrid-scale model* through the addition of artificial viscosity. The explicit filter, in addition to the implicit subgrid-scale model defined by the numerics, allows for control over numerical errors. Similar to the explicitly filtered explicitly modeled LES approach, however, the use of an explicit filtering operation can reduce the effective resolution of a numerical method. This can in turn degrade the accuracy. Explicitly filtered implicitly modeled LES is sometimes referred to as *implicit LES*. This is in reference to the fact that no subgrid-scale model is being used. As is discussed in the next section, however, many implicit LES approaches do not make use of an explicit filtering operation.

[1]The differencing in Eq 2.13 is upwind if $u > 0$.

2.3.4 Implicitly Filtered Implicitly Modeled Large Eddy Simulation

The implicitly filtered implicitly modeled LES approach, which is generally referred to as implicit LES (ILES), contains no explicit filtering operations or models. The method relies directly on the discretization to act as an implicit filter and an implicit subgrid-scale model. For a forward in time update, the method is described by,

$$\tilde{u}_j^{n+1} = \tilde{u}_j^n - \Delta t \left[\frac{1}{2} \frac{\tilde{u}_j^{n^2} - \tilde{u}_{j-1}^{n^2}}{\Delta x} \right]. \tag{2.15}$$

Implicit LES consists of solving the governing equations with a dissipative/stabilized numerical method on a coarse-discretization that does not resolve all scales. The approach has enjoyed success due to its simplicity and computational efficiency. Implicit LES, however, is not equipped with a notion of grid convergence (except to a direct numerical simulation result) and is highly dependent on the properties of the numerical method. Nonetheless, implicit LES methodologies have found significant traction in many communities. The high-order discontinuous Galerkin community [36, 32, 30] is one such example. As will be seen in later sections, high-order discontinuous Galerkin methods have a well-defined implicit filter and effective implicit subgrid-scale models.

2.3.5 Temporal Filtering

To this point the discussion has revolved primarily around spatial filtering. Most multiscale systems, however, display a wide range of both spatial and temporal scales. This advocates for the development of temporal and spatio-temporal filtering schemes [83, 12]. While such formulations are perhaps more rigorous, they have yet to gain significant traction in the community. The impact of temporal filtering will

not be considered in this dissertation.[2]

2.3.6 Community Consensus

The preceding discussion highlights the many flavors of LES methodologies. At this time, there is no agreement in the community upon which approach is best [81] and the implicit/explicit discussion is a point of contention[3]. As a result of these open questions and challenges, the development of LES methods and models remains an active research area.

2.4 Numerical Methods in LES

The LES methodologies described to this point have been discussed primarily at the continuous level. Numerical methods are required to discretize and solve the filtered Navier-Stokes equations. As was alluded to in the previous sections, the numerical method used in LES plays a significant role on the end calculation[4]. This is due to numerical methods having varying implicit filters and implicit subgrid-scale models.

The most prominent numerical methods in large eddy simulation are finite difference methods and weighted residual methods. Weighted residual methods encompass spectral, finite element, and finite volume methods. Both methods have been successfully used in a variety of contexts. As was apparent in the previous section, however, finite difference methods suffer from the lack of concise definition of a filter. As will be seen, weighted residual methods provide a cleaner setting for LES.

[2]The author believes that this should be an area of future work.

[3]The author spent the 2016 summer at the Center for Turbulence Research (CTR). At CTR, a heated debate between high-order and low-order methods, in conjunction with implicit vs explicit LES, occurred at least weekly!

[4]In the author's opinion, much of the disagreement regarding different LES methods stems from the fact that LES is too often discussed at the continuous level, rather than in association with a numerical method

2.5 The Weighted Residual and Galerkin Method

The weighted residual method is a procedure to solve ordinary and partial differential equations. The thesis of the weighted residual method is to approximate the solution with a finite-dimensional set of basis functions. All weighted residual methods have the advantage of being equipped with a vector space that allows for a formal definition of resolved and unresolved scales. A variety of weighted residual methods exist, including the Galerkin method, the Petrov-Galerkin method, the psuedo-spectral method, and the least-squares method. This work focuses on the Galerkin method.

The following sections outline the Galerkin method and discuss the modeling challenges that emerge from Galerkin discretizations. The majority of the discussion regarding the Galerkin method is standard and may be found in a variety of texts, such as [10, 11]. To the authors knowledge, however, the discussion of implicit filters inherent to the Galerkin method cannot be found outside of this dissertation. The notation for the "rough" finite element case is adopted from Ref [50].

2.5.1 Smooth Case (Spectral Methods)

The Galerkin discretization of Eq. 2.1 is formulated by casting Eq. 2.1 in its weak form. To derive the weak form, the test and trial functions have to be defined. Let $\mathcal{V}$ represent the trial space and $\mathcal{W}$ the test function space. In the Galerkin method, the test and trial space are set to be equivalent, i.e. $\mathcal{V} \equiv \mathcal{W}$. Let $\mathcal{V} \equiv L^2(\Omega)$ be the space of both the test and trial functions. The variational (weighted residual) problem is defined as follows: find $u \in \mathcal{V}$ such that $\forall w \in \mathcal{V}$,

$$(w, u_t + \mathcal{R}(u)) = (w, f), \tag{2.16}$$

where $(\cdot,\cdot)$ is the L^2 inner product,

$$(a,b) = \int_\Omega a(x)b(x)d\Omega. \tag{2.17}$$

In practice, we must approximate $\mathcal{V}$ with a finite dimensional space

$$\mathcal{V} \approx \tilde{\mathcal{V}}, u \approx \tilde{u}. \tag{2.18}$$

Under this approximation, the variational problem becomes: find $\tilde{u} \in \tilde{\mathcal{V}}$ such that $\forall \tilde{w} \in \tilde{\mathcal{V}}$,

$$(\tilde{w}, \tilde{u}_t + \mathcal{R}(\tilde{u})) = (\tilde{w}, f). \tag{2.19}$$

Equation 2.19 is the classic Galerkin method. Galerkin methods have been used successfully in a variety of settings, ranging from classic finite element methods to reduced order modeling using reduction techniques such as proper orthogonal decomposition.

Semi-Discrete Problem

The Galerkin method leads to a set of ordinary differential equations which can be solved via time marching (or space-time) techniques. Consider the expression of the state variable in the trial space,

$$\tilde{u}(x,t) = \sum_{j=0}^{N} w_j(x)a_j(t), \tag{2.20}$$

where $w_j(x)$ form a complete basis of $\tilde{\mathcal{V}}$ and where a_j are the modal coefficients to be solved for. Compactly,

$$\tilde{u}(x,t) = \tilde{\mathbf{w}}^T(x)\tilde{\mathbf{a}}(t). \tag{2.21}$$

In this discrete setting, Equation 2.19 can be written equivalently as,

$$\int_\Omega \tilde{\mathbf{w}}\tilde{\mathbf{w}}^T \frac{d\tilde{\mathbf{a}}}{dt} d\Omega + \int_\Omega \tilde{\mathbf{w}}\big(\mathcal{R}(\tilde{u}) - f\big)d\Omega = 0. \tag{2.22}$$

As the modal coefficients $\tilde{\mathbf{a}}$ are independent of space, the continuous problem becomes a coupled system of ordinary differential equations governed by

$$\frac{d\tilde{\mathbf{a}}}{dt} + \tilde{\mathbf{M}}^{-1} \int_{\Omega} \tilde{\mathbf{w}} \big(\mathcal{R}(\tilde{u}) - f \big) d\Omega = 0, \tag{2.23}$$

where $\mathbf{M}$ is the *mass matrix*,

$$\tilde{\mathbf{M}} = \int_{\Omega} \tilde{\mathbf{w}} \tilde{\mathbf{w}}^T d\Omega. \tag{2.24}$$

$\mathbf{L}^2$ Projection and Implicit Filtering

Galerkin methods (and all weighted residual methods) contain an implicit filter via projection. Consider the square error between the coarse-scale solution and the continuous state,

$$e^2 = \int_{\Omega} \big(u - \tilde{u} \big)^2 d\Omega. \tag{2.25}$$

The coarse-scale solution ($\tilde{u}$) that minimizes the square error can be found by differentiating Eq. 2.25 with respect to each modal coefficient and setting equal to zero,

$$\frac{de^2}{d\mathbf{a}} = \int_{\Omega} 2\tilde{\mathbf{w}} \big(u - \tilde{u} \big) d\Omega = 0. \tag{2.26}$$

Rearrangement results in,

$$\int_{\Omega} \tilde{\mathbf{w}} \tilde{u} d\Omega = \int_{\Omega} \tilde{\mathbf{w}} u d\Omega. \tag{2.27}$$

As $\tilde{u} = \tilde{\mathbf{w}}^T \tilde{\mathbf{a}}$,

$$\int_{\Omega} \tilde{\mathbf{w}} \tilde{\mathbf{w}}^T \tilde{\mathbf{a}} d\Omega = \int_{\Omega} \tilde{\mathbf{w}} u d\Omega. \tag{2.28}$$

The first term in Equation 2.28 is the mass matrix. Left multiplying by $\tilde{\mathbf{w}}^T \tilde{\mathbf{M}}^{-1}$ yields,

$$\tilde{u} = \tilde{\mathbf{w}}^T \tilde{\mathbf{M}}^{-1} \int_{\Omega} \tilde{\mathbf{w}} u d\Omega. \tag{2.29}$$

This can be written compactly in projection form,

$$\tilde{u} = \tilde{\Pi} u, \tag{2.30}$$

where $\tilde{\Pi}$ is the L^2 projection[5] onto $\tilde{\mathcal{V}}$. Note that $\tilde{\Pi}$ is formally a projector, i.e. $\tilde{\Pi}^2 = \tilde{\Pi}$. The L^2 projection acts as an implicit filter in the Galerkin method. This is seen by left multiplying Eq. 2.23 by $\tilde{\mathbf{w}}^T$,

$$\frac{\partial \tilde{u}}{\partial t} + \tilde{\mathbf{w}}^T \tilde{\mathbf{M}}^{-1} \int_\Omega \tilde{\mathbf{w}} \big(\mathcal{R}(\tilde{u}) - f\big) d\Omega = 0. \tag{2.31}$$

The reader should recognize that the second term is simply the L^2 projection of the residual. Equation 2.31 can be written equivalently as,

$$\tilde{\Pi} \frac{\partial \tilde{u}}{\partial t} + \tilde{\Pi} \big(\mathcal{R}(\tilde{u}) - f\big) = 0. \tag{2.32}$$

The L^2 projector acts in a similar fashion to filters in traditional large eddy simulation. The standard Galerkin method can be viewed as an implicitly filtered implicitly modeled LES approach.

2.5.2 Rough Case (Finite Element Methods)

Section 2.5.1 is only applicable to the case where the basis functions and states are globally smooth. This is appropriate for spectral methods (such as Fourier-Galerkin), but is not general enough for the complex geometries and multiscale problems commonly encountered in science and engineering. The finite element method (FEM) can be viewed as an extension of the spectral method and allows for more versatile numerical methods. The key difference between FEM and the globally smooth case is that the variational form must include boundary operators at element interfaces. We now detail this.

[5] For the error defined in Eq. 2.25, the L^2 projector is additionally the least squares projector, i.e. it minimizes the square error.

Define $\mathcal{T}_h$ to be the decomposition of the domain Ω into a set of non-overlapping elements, T, over Ω_k with boundaries Γ_k. We seek solutions in the finite element space,

$$\mathcal{V} = \{w \in L^2(\Omega) : v|_{\mathcal{T}} \in P^k(T), \forall T \in \mathcal{T}_h\}, \tag{2.33}$$

where P^k is the space of polynomials up to degree k. The above definition allows for discontinuities between elements. The continuous Galerkin formulation makes use of a subspace of Eq. 2.33 that enforces continuity between elements. Some notation is beneficial before proceeding. Define,

$$(\cdot,\cdot)_\Omega = \sum_{k\in\mathcal{T}} (\cdot,\cdot)_{\Omega_k}, \qquad (\cdot,\cdot)_\Gamma = \sum_{k\in\mathcal{T}} (\cdot,\cdot)_{\Gamma_k}. \tag{2.34}$$

The inner product $(\cdot,\cdot)_\Omega$ denotes a summation of volumetric inner products over the entire domain, while $(\cdot,\cdot)_\Gamma$ denotes a summation of surface inner products over the entire domain. The FEM weak formulation of Eq. 2.1 becomes

$$(w, u_t)_\Omega + (w, \mathcal{R}(u))_\Omega + (w, b(u))_\Gamma = (w, f)_\Omega \qquad \forall w \in \mathcal{V}, \tag{2.35}$$

where $b(u)$ is a boundary operator that arises via integration-by-parts. In the context of discontinuous Galerkin, one may consider Eq. 2.35 to be the "DG strong form". Similar to the globally smooth case, in practice one must approximate $\mathcal{V}$ with a finite dimensional space,

$$\mathcal{V} \approx \tilde{\mathcal{V}}, u \approx \tilde{u}. \tag{2.36}$$

As in the globally smooth case, the finite element method is equipped with an implicit filter.

2.6 The Variational Multiscale Method

It is well known that the standard continuous Galerkin method suffers from stability issues for a range of problems. This stems from the fact that the implicit

subgrid-scale model inherent to the Galerkin method is typically not dissipative. The variational multiscale method (VMS) of Hughes [48, 49, 50] is a slight modification of the classic Galerkin method and provides a convenient starting point for handling numerically unresolved phenomena. We proceed by outlining the variational multiscale method for the globally smooth case.

2.6.1 Globally Smooth Case (Spectral Methods)

The variational multiscale method utilizes a decomposition of the test/trial space into a coarse-scale resolved space $\tilde{\mathcal{V}} \subset \mathcal{V}$ and a fine-scale unresolved space $\mathcal{V}' \subset \mathcal{V}$. In VMS, the test/trial space is expressed as a sum decomposition,

$$\mathcal{V} = \tilde{\mathcal{V}} \oplus \mathcal{V}'. \tag{2.37}$$

Let $\tilde{\Pi}$ be the linear projector onto the coarse-scale space, $\tilde{\Pi} : \mathcal{V} \to \tilde{\mathcal{V}}$. Various choices exist for the projector $\tilde{\Pi}$ [52], and here we exclusively consider $\tilde{\Pi}$ to be the L^2 projector described previously,

$$(\tilde{w}, \tilde{\Pi} u) = (\tilde{w}, u) \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}, u \in \mathcal{V}. \tag{2.38}$$

The fine-scale space, $\mathcal{V}'$, becomes the orthogonal complement of $\tilde{\mathcal{V}}$ in $\mathcal{V}$ such that,

$$(\tilde{w}, \Pi' u) = 0 \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}, u \in \mathcal{V}, \tag{2.39}$$

where $\Pi' = \mathbf{I} - \tilde{\Pi}$. With this decomposition, the solution can be represented as

$$u = (\tilde{\Pi} + \Pi')u = \tilde{u} + u', \tag{2.40}$$

and the same for w.

Decomposing the test/trial space allows for Eq. 2.16 to be broken into two subproblems,

$$(\tilde{w}, \tilde{u}_t) + (\tilde{w}, \mathcal{R}(\tilde{u})) + (\tilde{w}, \mathcal{R}(u) - \mathcal{R}(\tilde{u})) = (\tilde{w}, f), \qquad \forall \tilde{w} \in \tilde{\mathcal{V}} \tag{2.41}$$

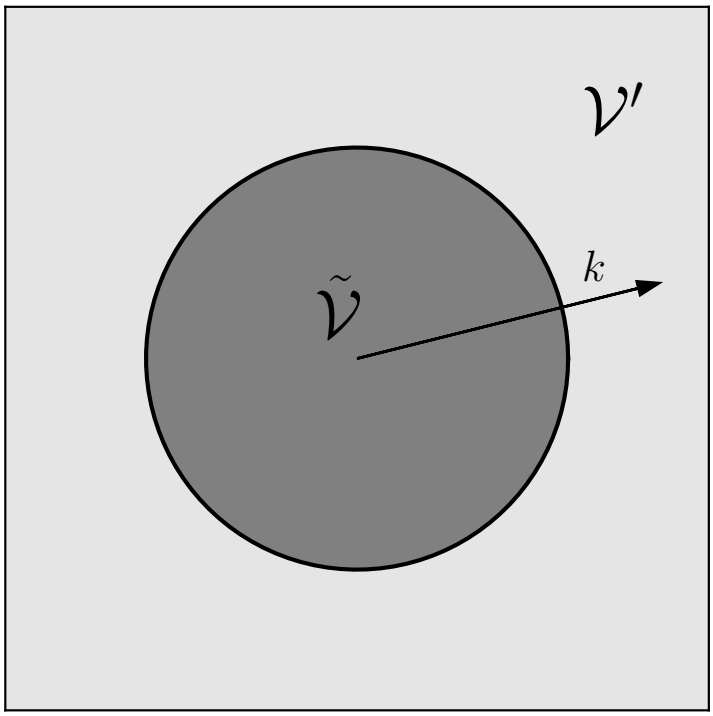


Figure 2.2: Graphical illustration of the decomposition of the test/trial space $\mathcal{V}$ into subspaces $\tilde{\mathcal{V}}$ and $\mathcal{V}'$ in the frequency domain (Figure concept adapted from [27]). The wavenumber is k. The subspace $\tilde{\mathcal{V}}$ corresponds to the low frequency, "coarse-scale" subspace and is resolved in a numerical method. The subspace $\mathcal{V}'$ is the high frequency "fine-scale" subspace and is not resolved in a numerical method. We consider a decomposition that obeys $\mathcal{V} = \tilde{\mathcal{V}} \oplus \mathcal{V}'$, with $\mathcal{V}'$ being L^2 orthogonal to $\tilde{\mathcal{V}}$.

$$(w', u_t') + (w', \mathcal{R}(\tilde{u})) + (w', R(u) - \mathcal{R}(\tilde{u})) = (w', f), \qquad \forall w' \in \mathcal{V}'. \tag{2.42}$$

Equation 2.41 is viewed as the large scale (resolved) equation, while Eq. 2.42 is the small scale (unresolved) equation. In the VMS approach, one generally tries to approximate the small scale equation to obtain an expression for u' and inject it into the large scale equation. The modeled equation can be written as,

$$(\tilde{w}, \tilde{u}_t) + (\tilde{w}, \mathcal{R}(\tilde{u})) + (\tilde{w}, \mathcal{M}(\tilde{u})) = (\tilde{w}, f), \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}. \tag{2.43}$$

The construction of the closure model $\mathcal{M}(\tilde{u})$ is the main challenge in the VMS approach.

Implicit Filtering in VMS

Equation 2.43 contains an implicit filter and is analogous to implicitly filtered explicitly modeled LES. Similar to Section 2.5.1, Eq. 2.43 can be written as

$$\tilde{\Pi}\frac{\partial u}{\partial t} + \tilde{\Pi}\mathcal{R}(\tilde{u}) + \tilde{\Pi}\Big[\mathcal{R}(u) - \mathcal{R}(\tilde{u})\Big] = \tilde{\Pi} f, \qquad \forall \tilde{w} \in \tilde{\mathcal{V}} \tag{2.44}$$

Equation 2.44 is in the same form as Eq. 2.6. The variational multiscale method is seen to be equivalent to large eddy simulation where the filter is the L^2 projection. The variational multiscale method is best viewed as an implicitly filtered explicitly modeled LES approach.

2.6.2 Rough Case (Finite Element Methods)

Similar to the Galerkin method, extension of the variational multiscale method to the rough FEM case requires the addition of boundary operators at the element interfaces. The VMS derivation for the rough case directly follows the VMS derivation of the smooth case and the Galerkin derivation for the rough case. The VMS

equations for the rough case are as follows,

$$(\tilde{w}, \tilde{u}_t)_\Omega + (\tilde{w}, \mathcal{R}(\tilde{u}))_\Omega + (\tilde{w}, \mathcal{R}(u) - \mathcal{R}(\tilde{u}))_\Omega + (\tilde{w}, b(\tilde{u}))_\Gamma + (\tilde{w}, b(u) - b(\tilde{u}))_\Gamma = (\tilde{w}, f)_\Omega \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}, \tag{2.45}$$

$$(w', u'_t)_\Omega + (w', \mathcal{R}(\tilde{u}))_\Omega + (w', \mathcal{R}(u) - \mathcal{R}(\tilde{u}))_\Omega + (w', b(\tilde{u}))_\Gamma + (w', b(u) - b(\tilde{u}))_\Gamma = (w', f)_\Omega \qquad \forall w' \in \mathcal{V}'. \tag{2.46}$$

Similar to the smooth case, the rough case contains an implicit filtering operation.

2.7 Subgrid-Scale Modeling

Subgrid-scale models are required in both explicitly modeled LES approaches and the variational multiscale method. While VMS and LES are analogous approaches (and, as was shown in Section 2.6, are equivalent in certain settings), the two methodologies were developed in different contexts and have led to different types of subgrid-scale models. The following sections outline classic subgrid-scale models and VMS-based subgrid-scale models.

2.7.1 Classic Subgrid-Scale Models

Classical subgrid-scale models used in LES have been developed within the fluid mechanics community specifically for the simulation of turbulent flows. These methods are typically introduced in the context of the incompressible Navier-Stokes equations,

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2.47}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_k \partial x_k}, \tag{2.48}$$

where u_i is the velocity component in the i_{th} direction, ρ is the fluid density, and ν the viscosity.

Application of a filter $\mathcal{G}(\mathbf{x}, \Delta)$ to the governing equations of motion yields,

$$\mathcal{G}\frac{\partial u_i}{\partial x_i} = 0, \tag{2.49}$$

$$\mathcal{G}\frac{\partial u_i}{\partial t} + \mathcal{G}\frac{\partial}{\partial x_j}(u_i u_j) = \mathcal{G}\bigg[-\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu\frac{\partial^2 u_i}{\partial x_k \partial x_k}\bigg]. \tag{2.50}$$

Defining $\mathcal{G}u = \tilde{u}$ and the same for pressure, the filtered Navier-Stokes equations are given as [101],

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0, \tag{2.51}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\tilde{u_i}\tilde{u_j}) = -\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\bigg[\nu\frac{\partial \tilde{u}_i}{\partial x_j} + \tau_{ij}\bigg], \tag{2.52}$$

where

$$\tau_{ij} = L_{ij} + C_{ij} + R_{ij}.$$

The term L_{ij} refers to the Leonard stress, C_{ij} refers to the cross stresses, and R_{ij} refers to the Reynolds stress. The above equations assume that the filters commute with the derivatives, which is - in general - not true. The term τ_{ij} is referred to as the *subgrid-stress* and requires modeling. The most prominent subgrid-scale models invoke a gradient-diffusion hypothesis analogous to molecular theory, where it is assumed that the subgrid-stresses are proportional to the strain rate,

$$\tau_{ij} = \nu_t\bigg(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\bigg). \tag{2.53}$$

This is often referred to as the *eddy viscosity* approach.

The Smagorinsky Model

The Smagorinsky model [88] is the first subgrid-scale model developed and is still one of the most popular models in use. The Smagorinsky model was developed in the context of geostrophic flows. The Smagorinsky model invokes the eddy viscosity concept to model the unresolved fluctuations with an approximate 3D Kolmogorov $k^{-5/3}$ cascade [55, 59]. In the Smagorinsky model, the eddy viscosity is taken to be

$$\nu_t = \big(C_s\Delta\big)^2\sqrt{S_{ij}S_{ij}}, \tag{2.54}$$

where C_s is the Smagorinsky constant and Δ is the effective filter size of the discretization. Lilly [62] showed that, by assuming the cutoff wavenumber is in the inertial sub-range of the Kolmogorov cascade, and $k_c = \pi/\Delta x$, the constant assumes a value,

$$C_s \approx \frac{1}{\pi}\big(\frac{3C_k}{2}\big)^{-3/4}, \tag{2.55}$$

where C_k is the Kolmogorov constant. Early measurements of the Kolmogorov constant in atmospheric turbulence suggested $C_k \approx 1.4$ [14], which leads to $C_s \approx 0.18$. More recent measurements in homogeneous turbulence measure the Kolmogorov constant to be closer to $C_k \approx 0.5$ [91], which leads to $C_s \approx 0.4$. In practice, however, most simulations use $C_s = 0.1$ [59], which leads to an eddy viscosity that is 16 times smaller in magnitude than theoretical predictions. The reason for this range in values of C_s is possibly due to the fact that, in practice, an LES rarely resolves the flow into the inertial subrange. Further, it is expected that C_s will depend on the flow regime, local resolution, and numerical method.

The Dynamic Smagorinsky Model

The variations in C_s motivated the development of the dynamic Smagorinsky model [41]. The dynamic Smagorinsky model utilizes the Germano identity [40] to

dynamically determine the constant, C_s. We leverage a similar idea later in this work, and as such present the derivation of the dynamic Smagorinsky model. The dynamic model determines C_s by applying a test filter to the subgrid-stress,

$$\overline{\tau}_{ij} = \overline{\widetilde{u_i u_j}} - \overline{\tilde{u}_i \tilde{u}_j}. \tag{2.56}$$

Defining the Leonard stress to be,

$$L_{ij} = \overline{\tilde{u}_i \tilde{u}_j} - \overline{\tilde{u}_i}\,\overline{\tilde{u}_j}, \tag{2.57}$$

one can express the Germano identity as,

$$T_{ij} = L_{ij} + \overline{\tau}_{ij}, \tag{2.58}$$

where $T_{ij} = \overline{\widetilde{u_i u_j}} - \overline{\tilde{u}_i}\,\overline{\tilde{u}_j}$. The Leonard stress, often referred to as the resolved stress, can be computed from the instantaneous filtered flow field. The tensor T_{ij} corresponds to the subgrid-stresses that arise due to both the test filter and the true filter. The Smagorinsky model is used to model both T_{ij} and $\overline{\tau}_{ij}$, allowing one to express Eq. 2.58 as,

$$-2\big(C_s\Delta_{k_L}\big)^2\big|\overline{\tilde{S}}\big|\overline{\tilde{S}}_{ij} = L_{ij} - 2\big(C_s\Delta_{k_c}\big)^2\overline{\big|\tilde{S}\big|\tilde{S}_{ij}}, \tag{2.59}$$

or more simply,

$$-2\big(C_s\Delta_{k_c}\big)^2\left(\alpha^2\big|\overline{\tilde{S}}\big|\overline{\tilde{S}_{ij}} - \overline{\big|\tilde{S}\big|\tilde{S}_{ij}}\right) = L_{ij}, \tag{2.60}$$

with $\alpha k_c = k_L$. Define,

$$M_{ij} = -2\big(\Delta_{k_c}\big)^2\left(\alpha^2\big|\overline{\tilde{S}}\big|\overline{\tilde{S}_{ij}} - \overline{\big|\tilde{S}\big|\tilde{S}_{ij}}\right). \tag{2.61}$$

The above is an over-constrained equation for k_c (there are six equations and only one unknown). The procedure developed by Lilly [63] minimizes the mean squared error, $\overline{\overline{(L_{ij} - C_s M_{ij})^2}}$, over some domain Ω_k. Mathematically,

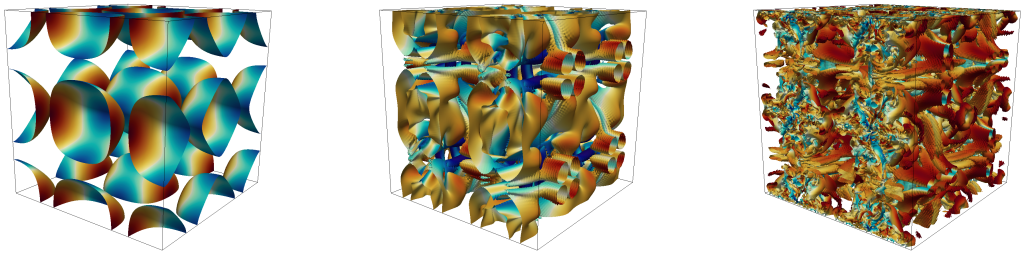
$$C_s^2 = \frac{\overline{\overline{L_{ij}M_{ij}}}}{\overline{\overline{M_{ij}M_{ij}}}}. \tag{2.62}$$

Additional Models

A large body of work regarding the development of subgrid-scale models exists outside of the static and dynamic Smagorinsky models. Several such models are Kraichnan's spectral eddy viscosity model [56], Vreman's model [99], the structure-function model of Métais and Lesiuer [67], and mixed models [103]. While many of these models are derived from an elegant treatment of turbulent structure and energy transfer, no model has been shown to significantly out-perform the Smagorinsky models in a general setting. As such, we will primarily consider the Smagorinsky models to be the baseline classic LES subgrid-scale closure in this work.

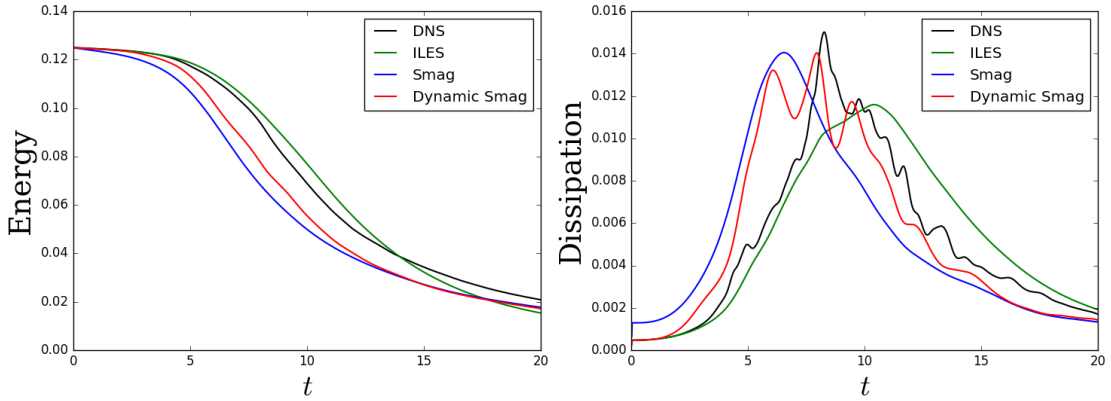
Breakdown of the Smagorinsky Models

The static and dynamic Smagorinsky models were developed in the context of homogeneous turbulent flows. The models, however, are known to perform poorly in many scenarios. To illustrate this, simulation results using 32^3 modes for the Taylor Green vortex at a Reynolds number of 1600 are presented. The Taylor Green vortex problem, which is further discussed in Chapter V, is a transitional flow problem that is characterized by a breakdown of initial coherent structures into turbulent flow. The evolution of this breakdown is depicted in Figure 2.3. Figure 2.4 shows the evolution of the total kinetic energy and dissipation in kinetic energy for LES simulations ran with no explicit subgrid-scale model as well as with the static and dynamic Smagorinsky models. All simulations were performed with the Galerkin spectral method and hence are implicitly filtered methods. In Figure 2.4, it is seen that explicit subgrid-scale models perform worse than the case with no explicit subgrid-scale model. This lack of accuracy is due to the fact that the Taylor Green vortex does not display canonical homogeneous turbulence, especially at early times. It instead is



(a) $t = 0$ (b) $t = 5$ (c) $t = 10$

Figure 2.3: Contours of q-criterion colored by the velocity magnitude for the Taylor Green vortex problem.



(a) Evolution of kinetic energy. (b) Dissipation of kinetic energy.

Figure 2.4: Temporal evolution of total kinetic energy and dissipation of kinetic energy for the Taylor Green vortex problem.

characterized by a complex transition from laminar to turbulent flow. The static and dynamic Smagorinsky models were not tuned for these types of flows. In practical scientific and engineering problems, this deviation from canonical conditions is commonplace. These limitations are further exposed when multiphysics problems, such as magnetohydrodynamics, are considered. The lack of accuracy of the Smagorinsky models for such problems motivates the development of new methods.

2.7.2 VMS-Based Subgrid-Scale Models

A variety of subgrid-scale models have emerged from the VMS community. The construction of subgrid-scale models in VMS originates from the work of Hughes [48].

In [48], Hughes considers the boundary-value problem,

$$\begin{aligned} Lu &= f \qquad \text{in} \qquad \Omega, \\ u &= 0 \qquad \text{on} \qquad \Gamma, \end{aligned} \tag{2.63}$$

where L is a linear differential operator and $\Omega \subset R^d$. Application of the VMS framework allows Eq. 2.63 to be separated into two sub-problems,

$$(\tilde{w}, L\tilde{u}) + (\tilde{w}, Lu') = (\tilde{w}, f), \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}, \tag{2.64}$$

$$(w', L\tilde{u}) + (w', Lu') = (w', f) \qquad \forall w' \in \mathcal{V}'. \tag{2.65}$$

Equations 2.64 and 2.65 may be equivalently written as,

$$(\tilde{w}, L\tilde{u}) + (L^*\tilde{w}, u') = (\tilde{w}, f), \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}, \tag{2.66}$$

$$(w', L\tilde{u}) + (w', Lu') = (w', f) \qquad \forall w' \in \mathcal{V}', \tag{2.67}$$

where we have used the integration-by-parts formula,

$$\big(\tilde{w}, Lu'\big) = \big(L^*\tilde{w}, u'\big), \qquad \forall \tilde{w} \in \tilde{V}, u' \in \mathcal{V}'. \tag{2.68}$$

The operator L^* is the adjoint of L. Note that the coarse-scale equation, Eq. 2.66, is unclosed. Closing the coarse-scale equation requires an expression for the fine scales. In the linear case, Hughes formulated an exact expression for the fine scales by solving the Green's function problem corresponding to the Euler-Lagrange equations of the fine-scale equation. The final expression for the fine-scale state is,

$$u'(y) = -\int_\Omega g'(x,y)\big(L\tilde{u} - f\big)(x) d\Omega_x, \tag{2.69}$$

where g' is the fine-scale Green's function satisfying,

$$\Pi' L^* g'(x,y) = \Pi' \delta(x-y) \qquad \forall x \in \Omega, \tag{2.70}$$

$$g'(x,y) = 0 \qquad \text{on} \qquad \Gamma. \tag{2.71}$$

The equation for the resolved, "coarse" scales can be obtained as,

$$(\tilde{w}, L\tilde{u}) + \big(L^*\tilde{w}, M'(L\tilde{u} - f)\big) = (\tilde{w}, f), \tag{2.72}$$

where,

$$\big(L^*\tilde{w}, M'(L\tilde{u} - f)\big) = -\int_\Omega (L^*\tilde{w})(y) \int_\Omega g'(x,y)(L\tilde{u} - f)(x) d\Omega_x d\Omega_y. \tag{2.73}$$

Residual-based Closures

For practical non-linear problems, the fine-scale Green's function is approximated. A common model assumes the fine-scale Green's function to be,

$$g'(x,y) \approx \tau(x)\delta(x-y), \tag{2.74}$$

where τ is a stabilization parameter. This approximation leads to the following expression for the fine-scale state,

$$u'(y) \approx -\tau\big(L\tilde{u} - f\big). \tag{2.75}$$

Equation 2.75 is referred to as a *residual-based method*. This name is derived from the fact that the fine scales are approximated by the coarse-scale residual, $L\tilde{u} - f$. The VMS coarse-scale equation with the model Eq. 2.75 appears as follows,

$$(\tilde{w}, L\tilde{u}\big) - \big(L^*\tilde{w}, \tau(L\tilde{u} - f)\big) = (\tilde{w}, f). \tag{2.76}$$

The approximation given by Eq. 2.76 is an adjoint-type stabilization technique and was proposed [39] before the VMS procedure. One of the significant accomplishments of the VMS framework is that it sheds light on the mathematical origins of such stabilization techniques.

Orthogonal Subscales

The concept of orthogonal subscales, developed by Codina [25, 26, 45, 5], leverages the fact that, with the L^2 projector, the fine-scale variable u' is orthogonal to the coarse scales, $\tilde{u}$. The orthogonal subscale approach approximates the fine-scale variable by

$$u' = -\tau\hat{\Pi}(L\tilde{u} - f),$$

where $\Pi' = I - \tilde{\Pi}$. With this selection, Codina enforces the approximation u' to be orthogonal to the resolved scales.

Series Perturbations

The VMS formulation provided in Hughes' pioneering work is not directly applicable to non-linear systems. In the case of non-linearity, a proposed methodology that is of relevance to this work is the perturbation series developed by Hughes [6]. The main idea is to express the fine scales, u', in a series expansion,

$$u' = \sum_{i=1}^{\infty} \epsilon^i {u'}_i, \tag{2.77}$$

where ϵ is the norm of the coarse-scale residual and is assumed to be small. This allows Eqns 2.41 and 2.42 to be linearized with respect to ϵ. The first order case is referred to as the VMS(ϵ) model [100],

$$u' = \epsilon^1 {u'}_1, \tag{2.78}$$

which allows for the linearization,

$$\mathcal{R}(\tilde{u} + u') = \mathcal{R}(\tilde{u}) + \frac{\partial \mathcal{R}}{\partial u'} u'. \tag{2.79}$$

Note that the linearization is taken about $u = \tilde{u}$. The linearization leads to the VMS equations being expressed as,

$$(\tilde{w}, \tilde{u}_t) + (\tilde{w}, \mathcal{R}(\tilde{u})) + (\tilde{w}, \frac{\partial \mathcal{R}}{\partial u'} u') = (\tilde{w}, f), \qquad \forall \tilde{w} \in \tilde{\mathcal{V}} \tag{2.80}$$

$$(w', u'_t) + (w', \mathcal{R}(\tilde{u})) + (w', \frac{\partial R}{\partial u'} u') = (w', f), \qquad \forall w' \in \mathcal{V}'. \tag{2.81}$$

As the fine-scale equation is now linear, the classic fine-scale Green's function procedure is now more robust. Approximating the fine-scale Green's function with the algebraic operator, τ, defined on the fine scales leads to the approximation,

$$u' \approx -\Pi' \tau (\mathcal{R}(\tilde{u}) - f).$$

The coarse-scale equation then becomes,

$$(\tilde{w}, \tilde{u}_t) + (\tilde{w}, \mathcal{R}(\tilde{u})) - (\tilde{w}, \Pi' \frac{\partial \mathcal{R}}{\partial u'} \tau (\mathcal{R}(\tilde{u}) - f)) = (\tilde{w}, f), \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}. \tag{2.82}$$

Equation 2.82 will be demonstrated to be equivalent to a class of Mori-Zwanzig-based models.

Scale-dependent Classic LES-Based Models

An additional modeling approach that has been introduced in the VMS community is based on a three-scale decomposition. In the three-scale decomposition, the resolved scales are decomposed into coarse-scale resolved scales and fine-scale resolved scales. Classic LES models, such as the Smagorinsky model, are then applied to the fine-scale resolved scales. These methods have been investigated in several contexts [27].

2.8 Outstanding Issues

Variational multiscale methods have been used successfully in a variety of contexts. The simulation of incompressible and compressible turbulent flows, magnetohydrodynamic flows, and solid mechanics are several such examples. A number of outstanding issues regarding variational multiscale models, however, remain. Several such issues are:

1. How can VMS methods be applied to non-linear problems? For non-linear problems, the Green's function procedure developed by Hughes' no longer holds. Typically, linearization arguments are used to extend VMS methods to non-linear equations.
2. How is the VMS procedure applied to unsteady problems, particularly at the semi-discrete level?

These issues motivate the use of the Mori-Zwanzig formalism and the development of a framework that combines Mori-Zwanzig with the variational multiscale method.

2.9 Chapter Summary

This chapter outlined the large eddy simulation technique. Important aspects of this chapter are:

- The discussion of implicitly/explicitly filtered and implicitly/explicitly modeled LES.
- Formulation of the Galerkin method and the variational multiscale method.
- Demonstration that Galerkin and variational multiscale methods contain implicit filters.
- Outline of classic LES and VMS-based subgrid-scale models.
- Discussion of the advantages and disadvantages of existing methods.

The next several chapters seek to address the deficiencies of existing methods by introducing the Mori-Zwanzig formalism as a modeling framework.

CHAPTER III

Development of the MZ-VMS Framework

As seen in the previous chapter, the variational multiscale method provides a formal decomposition of a solution into fine and coarse scales and is an attractive starting point for large eddy simulation. VMS-based subgrid-scale models are based on the notion of a fine-scale Green's function corresponding to the Euler-Lagrange equations of the fine-scale equation. This process is formulated at the *continuous level*. An alternative formulation can be developed by considering the solution to the fine-scale equation at the *semi-discrete level* and by making use of *Duhamel's principle*. The semi-discrete setting has the advantage that the governing equations have been reduced to a set of ordinary differential equations. Duhamel's principle can be conceptualized to be the "Green's function in time" and allows for a formal removal of the fine scales.

This chapter outlines how the Mori-Zwanzig formalism can be used to develop models for the variational multiscale method. The philosophy of Mori-Zwanzig will be demonstrated by first considering a linear initial value problem. After this, Chorin's formulation of Mori-Zwanzig will be outlined within the setting of the variational multiscale method. The memory term and orthogonal dynamics will be discussed.

3.1 Linear Initial Value Problem

Consider the linear initial value problem,

$$\frac{\partial u}{\partial t} + Lu = f \qquad x \in \Omega, t \in (0,T), \tag{3.1}$$

subject to the boundary and initial conditions,

$$u(x,t) = 0 \qquad x \in \Gamma, t \in (0,T), \tag{3.2}$$

$$u(x,0) = u_0 \qquad x \in \Omega. \tag{3.3}$$

Note that L is again a differential operator. Letting $\mathcal{V} \equiv H_0^1(\Omega)$ denote the test and trial space, the Galerkin formulation at the semi-discrete level leads to the weighted residual problem,

$$(w, u_t) + (w, Lu - f) = 0 \qquad \forall w \in \mathcal{V}. \tag{3.4}$$

Separating the test/trial space via the VMS sum decomposition leads to two sub-problems,

$$(\tilde{w}, \tilde{u}_t) + (\tilde{w}, u'_t) + (\tilde{w}, L\tilde{u}) + (\tilde{w}, Lu') = (\tilde{w}, f) \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}, \tag{3.5}$$

$$(w', \tilde{u}_t) + (w', u'_t) + (w', L\tilde{u}) + (w', Lu') = (w', f) \qquad \forall w' \in \mathcal{V}'. \tag{3.6}$$

For the L^2 projector considered in this work, the fine and coarse scales are orthogonal. Equations 3.5 and 3.6 simplify to,

$$(\tilde{w}, \tilde{u}_t) + (\tilde{w}, L\tilde{u}) + (\tilde{w}, Lu') = (\tilde{w}, f) \qquad \forall \tilde{w} \in \tilde{\mathcal{V}}, \tag{3.7}$$

$$(w', u'_t) + (w', L\tilde{u}) + (w', Lu') = (w', f) \qquad \forall w' \in \mathcal{V}'. \tag{3.8}$$

We assume $u_0 \in \tilde{\mathcal{V}}$ such that,

$$\tilde{u}(x,0) = u_0, \tag{3.9}$$

$$u'(x,0) = 0. \tag{3.10}$$

The objective of the multiscale formulation is to now obtain a parameterization for the fine scales u' in terms of the coarse scales. Hughes' achieves this through the use of Green's functions. A different approach is motivated by the following observations:

1. Equations 3.7 and 3.8 are unsteady.
2. Equations 3.7 and 3.8 can be equivalently written as ordinary differential equations.

To make use of these observations, we consider the set of discrete equations arising from Eqns. 3.5 and 3.6. Let the vector of trial/test functions be denoted by $\mathbf{w}$. The expression of the state variable in the trial space is,

$$u(x,t) = \sum_{j=0}^{\infty} w_j(x) a_j(t), \tag{3.11}$$

where a_j are the modal coefficients to be solved for. Compactly,

$$u(x,t) = \mathbf{w}^T(x)\mathbf{a}(t). \tag{3.12}$$

Equations 3.13 and 3.14 can be written equivalently as,

$$(\tilde{\mathbf{w}}, \tilde{u}_t) + (\tilde{\mathbf{w}}, L\tilde{u}) + (\tilde{\mathbf{w}}, Lu') = (\tilde{\mathbf{w}}, f), \tag{3.13}$$

$$(\mathbf{w}', u'_t) + (\mathbf{w}', L\tilde{u}) + (\mathbf{w}', Lu') = (\mathbf{w}', f). \tag{3.14}$$

The parameterization of the fine-scale state in terms of the coarse-scale state can be achieved through the use of integrating factors[1]. For simplicity, we restrict ourselves to the case of orthonormal basis functions, although no such simplification is necessary. The solution to the fine-scale equation can be written as,

$$u'(t) = -\int_0^t \mathbf{w'}^T e^{-s(\mathbf{w}', L\mathbf{w}')}(\mathbf{w}', L\tilde{u}(t-s) - f)ds,$$

[1]For the reader who is unfamiliar with integrating factors, a standard google search should prove sufficient.

where $e^{s(\mathbf{w}', L\mathbf{w}'^T)}$ is the matrix exponential. Injecting the fine-scale solution into the coarse-scale equation yields,

$$(3.15)\quad (\tilde{\mathbf{w}}, \tilde{u}_t) + \big(\tilde{\mathbf{w}}, L\tilde{u}\big) - \Big(\mathbf{w}', \int_0^t L\mathbf{w}'^T e^{-s(\mathbf{w}', L\mathbf{w}'^T)}(\mathbf{w}', L\tilde{u}(t-s) - f)ds\Big) = \big(\tilde{\mathbf{w}}, f\big).$$

The convolution integral is referred to as memory. Equation 3.15 can be written in a more transparent form by recognizing that the memory contains terms similar to a projection. Define,

$$\Pi'^{\mathcal{Q}}(t)g = \mathbf{w}'^T e^{-t(\mathbf{w}', L\mathbf{w}')}(\mathbf{w}', g).$$

With this notation,

$$(3.16)\qquad \Big(\tilde{\mathbf{w}}, \tilde{u}_t\Big) + \Big(\tilde{\mathbf{w}}, L\tilde{u}\Big) - \Big(\tilde{\mathbf{w}}, \int_0^t L\Pi'^{\mathcal{Q}}(s)\big[L\tilde{u}(t-s) - f\big]ds\Big) = \Big(\tilde{\mathbf{w}}, f\Big).$$

Remarks:

- Equation 3.16 is an exact, closed equation for the coarse scales.
- The formal removal of the fine scales leads to a coarse-scale problem that is non-local in time.
- Similar to the Green's function approach, the memory is driven by the residual of the coarse-scale equation. Additionally, the projection associated with Eq. 3.16 constrains the residual to be in the space of the fine scales. This is similar to the VMS approach, where the Green's function problem is defined on the fine scales.
- Even though Equation 3.16 is written for the coarse scales, no reduction in computational cost has been achieved as we have simply transformed the system into an integro-differential equation.
- Integrating factors are an instance of Duhamel's principle for ordinary dif-

ferential equations. Duhamel's principle will be used extensively in what follows.

- Equation 3.16 should be viewed as a starting point for model development in complex problems.

3.2 Nonlinear Problems: The MZ-VMS Framework

In the linear case, we showed that it is possible to obtain an analytic expression for the fine-scale state through the use of integrating factors. The method of integrating factors is, unfortunately, not generalizable to non-linear problems. These challenges can be addressed by using Chorin's formulation of the Mori-Zwanzig formalism.

Chorin's MZ procedure, which is often referred to as the optimal prediction framework, was developed by Chorin and co-workers in [20, 22, 42, 21]. The framework can be viewed as a model order reduction strategy for general ordinary differential equations. The framework centers around the use of projection operators that separate the phase space of an ordinary differential equation into resolved and unresolved subspaces. Using this decomposition, a high-dimensional non-linear Markovian dynamical system can be recast into an equivalent, lower-dimensional non-Markovian system. This formulation is mathematically exact. In the lower-dimensional system, which is commonly referred to as the generalized Langevin equation (GLE), the effect of the fine scales scales on the coarse scales scales is non-local in time and appears as a convolution integral. This term is typically referred to as memory. The memory term emerging from the MZ formalism is analogous to the Green's function emerging from the VMS procedure. While the memory integral is intractable in general non-linear problems, it serves as a convenient starting point for the construction of closure models.

The remainder of this section outlines Chorin's formulation of Mori-Zwanzig within the context of the variational multiscale method. Similar to the linear case, this leads to a formally closed coarse-scale equation that is non-local in time. This coarse-scale equation is viewed as a starting point for model development.

We consider the initial-value problem,

$$\frac{\partial u}{\partial t} + \mathcal{R}(u) = f \qquad x \in \Omega, t \in (0,T), \tag{3.17}$$

where $\mathcal{R}$ is a non-linear differential operator. The governing equations are subject to boundary and initial conditions,

$$u(x,t) = 0 \qquad x \in \Gamma, t \in (0,T), \tag{3.18}$$

$$u(x,0) = u_0 \qquad x \in \Omega. \tag{3.19}$$

With the trial and test space $\mathcal{V} \equiv H_0^1(\Omega)$, the discrete multiscale equations for the Galerkin method can be written as,

$$(\tilde{\mathbf{w}}, \tilde{u}_t) + (\tilde{\mathbf{w}}, \mathcal{R}(\tilde{u})) + (\tilde{\mathbf{w}}, \mathcal{R}(u) - \mathcal{R}(\tilde{u})) = (\tilde{\mathbf{w}}, f), \tag{3.20}$$

$$(\mathbf{w}', u'_t) + (\mathbf{w}', \mathcal{R}(\tilde{u})) + (\mathbf{w}', R(u) - \mathcal{R}(\tilde{u})) = (\mathbf{w}', f), \tag{3.21}$$

with $\mathbf{a}(t=0) = \mathbf{a}_0$. We again consider orthonormal basis functions for simplicity, although this is not required (eg. see Section 6.1). We denote the Hilbert space in which $\mathbf{a}_0$ resides as $\mathcal{H}$. The objective is again to analytically solve for the fine-scale state.

3.2.1 Transformation to Phase Space and the Liouville Equation

The Green's function used in the traditional VMS approach as well as the integrating factor approach described in the previous section relies on the principle of

superposition and is limited to linear systems. It was recognized by Chorin and co-workers, however, that *any* non-linear ordinary differential equation can be written as a *linear* partial differential equation that exists in phase space. This transformation is exact and allows for the use of superposition. Here, Eqns. 3.20 and 3.21 can be cast as the following linear partial differential equation [21],

$$\frac{\partial v}{\partial t} = \mathcal{L}v, \tag{3.22}$$

subject to initial conditions,

$$v(\mathbf{a}_0, 0) = g(\mathbf{a_0}). \tag{3.23}$$

The operator $\mathcal{L}$ is the Liouville operator and is given by,

$$\mathcal{L} = \sum_{j=0}^{\infty} (w_j, f - \mathcal{R}(u_0)) \frac{\partial}{\partial a_{0j}}. \tag{3.24}$$

Equation 3.22 is referred to as the Liouville equation and is an exact statement of the original dynamics. Chorin showed that the solution to Eq. 3.22 is given by [21],

$$v(\mathbf{a}_0, t) = g(\mathbf{a}(\mathbf{a_0}, t)). \tag{3.25}$$

Semi-group notation is now used, in which the solution to the Liouville equation may be expressed as,

$$v(\mathbf{a_0}, t) = e^{t\mathcal{L}} g(\mathbf{a}(\mathbf{a_0}, 0)). \tag{3.26}$$

The proper interpretation of Eq. 3.26 is that $v(\mathbf{a_0}, t)$, is given by the solution of the partial differential equation defined by the evolution operator $e^{t\mathcal{L}}$ (in this case Eq. 3.22), with initial conditions $v(\mathbf{a_0}, 0) = g(\mathbf{a}(\mathbf{a_0}, 0))$. The evolution operator $e^{t\mathcal{L}}$ has several interesting properties. Most notably, the operator can be moved inside of a function [107]. With this property, the solution to Eq. 3.22 may be written as,

$$v(\mathbf{a_0}, t) = e^{t\mathcal{L}} g(\mathbf{a}(\mathbf{a_0}, 0)) = g(e^{t\mathcal{L}} \mathbf{a}(\mathbf{a_0}, 0)). \tag{3.27}$$

The implications of $e^{t\mathcal{L}}$ are significant. It demonstrates that, given the trajectories $\mathbf{a}(\mathbf{a_0}, t)$, the solution v is known for any observable g. Noting that $\mathcal{L}$ and $e^{t\mathcal{L}}$ commute, Eq. 3.22 may be written in the semi-group notation as,

$$\frac{\partial v}{\partial t} = e^{t\mathcal{L}}\mathcal{L}v(\mathbf{a}_0, 0). \tag{3.28}$$

A set of equations for the resolved modes can be obtained by taking $g(\mathbf{a}_0) = \tilde{\mathbf{a}}_0$,

$$\frac{\partial}{\partial t}e^{t\mathcal{L}}\tilde{\mathbf{a}}_0 = e^{t\mathcal{L}}\mathcal{L}\tilde{\mathbf{a}}_0. \tag{3.29}$$

3.2.2 Projection Operators and the Generalized Langevin Equation

We proceed by decomposing the space of initial conditions, $\mathcal{H}$, into a resolved and unresolved subspace,

$$\mathcal{H} = \tilde{\mathcal{H}} \oplus \mathcal{H}'. \tag{3.30}$$

The associated projection operators are defined as $\mathcal{P} : \mathcal{H} \to \tilde{\mathcal{H}}$ and $\mathcal{Q} = I - \mathcal{P}$. We consider a projection operator that is appropriate for the deterministic initial conditions considered here (Eqns. 3.9 and 3.10),

$$\mathcal{P}f(\tilde{\mathbf{a}}_0, \mathbf{a'}_0) = \int_{\mathcal{H}} f(\tilde{\mathbf{a}}_0, \mathbf{a'}_0)\delta(\mathbf{a'}_0)d\mathbf{a'}_0, \tag{3.31}$$

which leads to

$$\mathcal{P}f(\tilde{\mathbf{a}}_0, \mathbf{a'}_0) = f(\tilde{\mathbf{a}}_0, 0). \tag{3.32}$$

Other projections, such as conditional expectations, are possible [20], but will not be pursued in the present work. It is important to emphasize that the projectors $\mathcal{P}$ and $\mathcal{Q}$ operate on $\mathcal{H}$ and are fundamentally different from the L^2 projectors $\tilde{\Pi}$ and Π' (see Figure 3.1). With the projection operators, the Liouville equation can be split as,

$$\frac{\partial}{\partial t}e^{t\mathcal{L}}\tilde{\mathbf{a}}_0 = e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\tilde{\mathbf{a}}_0 + e^{t\mathcal{L}}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0. \tag{3.33}$$

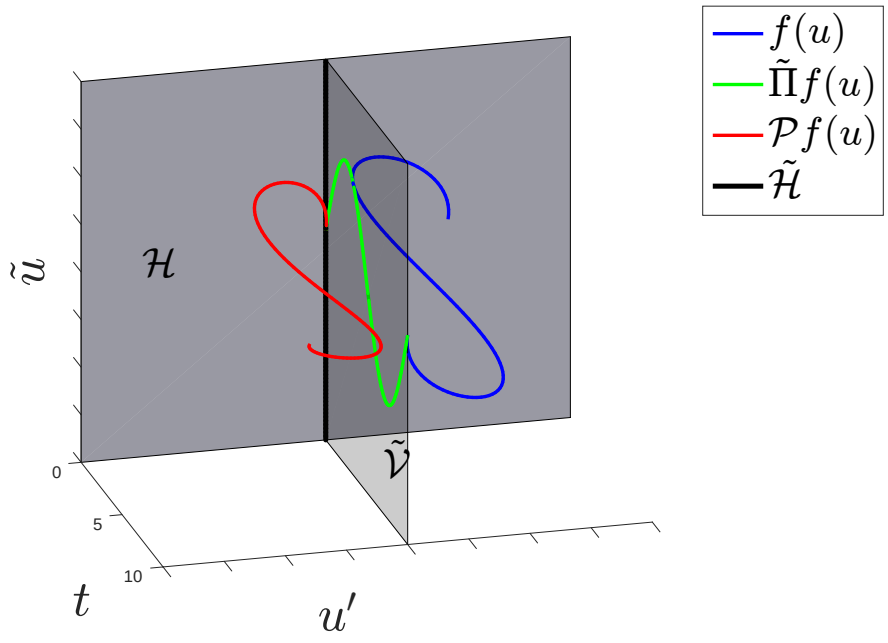


Figure 3.1: Graphical depiction of various projection operators acting on a function $f(u)$. $\tilde{\Pi}$ projects (in an L^2 sense) the signal onto the subspace $\tilde{\mathcal{V}}$, which in this case is denoted as the plane in which $u' = 0$. The projector $\mathcal{P}$ used in the Mori-Zwanzig formalism projects the *initial conditions* onto the subspace $\tilde{\mathcal{H}}$, which here is a vertical line at $t = 0$ and $u' = 0$. Note that terms projected by $\mathcal{P}$ do not necessarily evolve on the manifold defined by $\tilde{\mathcal{V}}$.

The objective now is to remove the dependence of the right hand side of Eq. 3.33 on the unresolved scales, $\mathbf{a}'$ (i.e. $\mathcal{QL}\tilde{\mathbf{a}}$). To demonstrate how this may be achieved, consider the partial differential operator governed by the semi-group $e^{t\mathcal{L}}$ written as,

$$\frac{\partial}{\partial t} - \mathcal{L} = 0. \tag{3.34}$$

We will refer to Eq. 3.34 as the homogeneous problem. Consider now the inhomogeneous problem with forcing $\mathcal{PL}$,

$$\frac{\partial}{\partial t} - \mathcal{L} = -\mathcal{PL}. \tag{3.35}$$

Making use of the identity $I = \mathcal{P} + \mathcal{Q}$, the inhomogeneous problem can be written as

$$\frac{\partial}{\partial t} - \mathcal{QL} = 0. \tag{3.36}$$

Eq. 3.36 is referred to in the literature as the orthogonal dynamics operator, and can be conceptualized as a Liouville operator with forcing. The evolution operator given by the orthogonal dynamics is $e^{t\mathcal{QL}}$. Here, we can leverage the linearity of the partial differential operators and make use of superposition. The solution to the orthogonal dynamics equation can be expressed in terms of solutions to the homogeneous Liouville equation through Duhamel's principle,

$$e^{t\mathcal{L}} = e^{t\mathcal{QL}} + \int_0^t e^{(t-s)\mathcal{L}}\mathcal{PL}e^{s\mathcal{QL}}ds. \tag{3.37}$$

Inserting Eq. 3.37 into Eq. 3.33, the generalized Langevin equation is obtained,

$$\frac{\partial}{\partial t}e^{t\mathcal{L}}\tilde{\mathbf{a}}_0 = \underbrace{e^{t\mathcal{L}}\mathcal{PL}\tilde{\mathbf{a}}_0}_{\text{Markovian}} + \underbrace{e^{t\mathcal{QL}}\mathcal{QL}\tilde{\mathbf{a}}_0}_{\text{Noise}} + \underbrace{\int_0^t e^{(t-s)\mathcal{L}}\mathcal{PL}e^{s\mathcal{QL}}\mathcal{QL}\tilde{\mathbf{a}}_0 ds}_{\text{Memory}}. \tag{3.38}$$

The system described in Eq. 3.38 is precise and not an approximation to the original ODE system. For notational purposes, define

$$F_j(\mathbf{a}_0, t) = e^{t\mathcal{QL}}\mathcal{QL}a_{0j}, \qquad K_j(\mathbf{a}_0, t) = \mathcal{PL}F_j(\mathbf{a}_0, t). \tag{3.39}$$

We refer to $K(\mathbf{a}_0, t)$ as the *memory kernel*. It can be shown that solutions to the orthogonal dynamics equation are in the null space of $\mathcal{P}$, meaning $\mathcal{P}F_j(\mathbf{a}_0, t) = 0$. By the definition of the initial conditions (Eq. 3.10), the noise-term is zero and we obtain,

$$\frac{\partial}{\partial t} e^{t\mathcal{L}} \tilde{\mathbf{a}}_0 = e^{t\mathcal{L}} \mathcal{P}\mathcal{L}\tilde{\mathbf{a}}_0 + \int_0^t e^{(t-s)\mathcal{L}} \mathcal{P}\mathcal{L} e^{s\mathcal{Q}\mathcal{L}} \mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 ds. \tag{3.40}$$

Equation 3.40 can be written in a more transparent form,

$$(\tilde{\mathbf{w}}, \tilde{u}_t) + (\tilde{\mathbf{w}}, \mathcal{R}(\tilde{u})) - \int_0^t K(\tilde{\mathbf{a}}(t-s), s) ds = (\tilde{\mathbf{w}}, f), \tag{3.41}$$

where $K_j(\mathbf{a}_0, t) = \mathcal{P}\mathcal{L}e^{t\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\mathbf{a}_0$. Note that the time derivative is represented as a partial derivative due to the Liouville operators embedded in the memory.

Remarks

- Equation 3.41 is precisely a Galerkin discretization of Eq. 3.17 with the addition of a memory term originating from the removal of the fine scales.
- Equation 3.41 is a non-local closed equation for the coarse scales.
- Evaluation of the memory term is not tractable as it involves the solution of the evolution operator, $e^{t\mathcal{Q}\mathcal{L}}$. This is referred to as the orthogonal dynamics and is discussed in the following section. Instead, Eq. 3.41 is viewed as a starting point for the construction of closure models.
- The formulation for the smooth case is limited to spectral methods on canonical domains. In Section 6.1, we extend the discussion to the case of finite elements.

3.3 The Memory Term

The MZ procedure itself does not provide a reduction in computational complexity as it has replaced the fine-scale state with a memory term. This memory term is given by,

$$\int_0^t K(\tilde{\mathbf{a}}(t-s),s)ds = \int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 ds. \tag{3.42}$$

The memory term can be equivalently written as a two-dimensional path integral (instead of a convolution),

$$\oint K(\tilde{\mathbf{a}}(t),s)d\tau = \frac{1}{\sqrt{2}}\oint e^{t\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}d\tau, \tag{3.43}$$

with τ being the line of constant $t-s$.

The memory term relies on solutions to the orthogonal dynamics equation, denoted by the evolution operator $e^{t\mathcal{Q}\mathcal{L}}$. The orthogonal dynamics equation is a high dimensional forced partial differential equation. While the evolution operator $e^{t\mathcal{L}}$ is a Koopman operator, no such result exists for $e^{t\mathcal{Q}\mathcal{L}}$ in the general non-linear case. As a consequence, evaluating terms evolved by $e^{t\mathcal{Q}\mathcal{L}}$ requires one to directly solve the orthogonal dynamics equation. Solutions of the orthogonal dynamics are intractable in the general case. Instead of reducing the computational complexity, the MZ procedure provides an exact representation of the fine-scale state in terms of the coarse scales. This is used as a starting point for model development. In this section, we expand our discussion of the orthogonal dynamics equation and explore the mechanics of the memory term.

3.3.1 The Kernel Surface

In the variational multiscale method, the fine-scale state is parameterized in terms of the coarse-scale state by virtue of a fine-scale Green's function. The Mori-Zwanzig

procedure instead uses Duhamel's principle to relate the solution of the orthogonal dynamics equation to the Liouville equation. This allows for the elimination of fine scales. The evolution operator of the orthogonal dynamics is given by $e^{t\mathcal{QL}}$. To help clarify the interaction of the memory term and the orthogonal dynamics, Figure 3.2 depicts the memory kernel in $s-t$ space. In Figure 3.2a, the evolution of the solution in time is denoted by the solid blue line at $s = 0$. To evaluate the memory, solutions to the orthogonal dynamics equation, $F(\tilde{\mathbf{a}}(t), s)$, must be evolved in pseudo-time s using initial conditions that depend on the solution at time t. This is depicted by the dashed red lines in Figure 3.2a. This leads to a three-dimensional surface in $s-t$ space, as seen in Figure 3.2b. Evaluation of the memory integral then requires a path integration backwards in time along the dashed-lines in Figure 3.2a, yielding the shaded yellow region in Figure 3.2b.

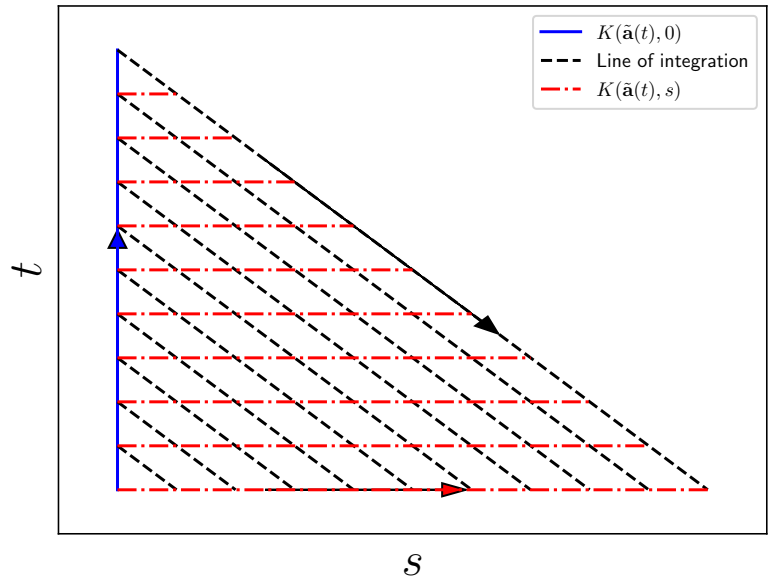
There are several important quantities of interest in the memory term. The most important quantity is the memory kernel evaluated at $s = 0$. This term, $K(\tilde{\mathbf{a}}(t), 0)$, drives the memory. In semi-group notation, the kernel at $s = 0$ is written as,

$$K(\tilde{\mathbf{a}}(t), 0) = e^{t\mathcal{L}}\mathcal{PLQL}a_{0j}. \tag{3.44}$$

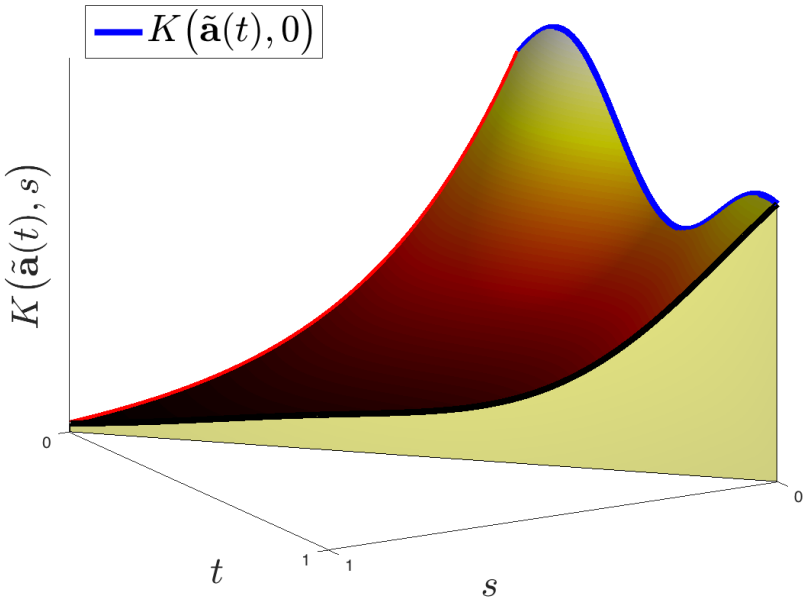
It is immediately observed that Eq. 3.44 has no dependence on the orthogonal dynamics. The temporal evolution of Eq. 3.44 is depicted by the blue line in Figure 3.2 at $s = 0$. It is next noted that the $\mathcal{PLQL}$ operators can be evaluated analytically. Evaluating the operators for the smooth case non-linear case presented in Section 3.2 yields (the step by step details are given in Appendix A),

$$K(\tilde{\mathbf{a}}(t), 0) = \int_\Omega \int_\Omega (\tilde{\mathbf{w}}\mathcal{R}')(x)\Pi'(x, y)(\mathcal{R}(\tilde{u}) - f)(y)d\Omega_y d\Omega_x, \tag{3.45}$$

where $\mathcal{R}' = \frac{\partial \mathcal{R}}{\partial \tilde{u}}$.



(a) Two-dimensional illustration in the $s - t$ plane.



(b) Three-dimensional memory surface. This illustrative surface has a decaying profile that is representative of finite memory effects.

Figure 3.2: Graphical depiction of the mechanics of the memory term.

Remarks

1. Equation 3.45 shows that the memory is driven by an orthogonal projection of the coarse-scale residual. If this residual is zero, no information is added to the memory. Further, if the coarse-scale equation is fully resolved, no information is added to the memory.
2. The finding that the memory term is based on the coarse-scale residual is significant. It demonstrates that MZ-based methods are, in fact, residual-based methods. This provides insight into the performance of MZ methods as it links them to a larger class of models. Subgrid-scale models derived from physical theory, such as the Smagorinsky model, are not residual-based methods.
3. The form of the memory at $s = 0$ displays many similarities to Eq. 2.73.
4. In Figure 3.2b, the memory surface is always at a maximum at $s = 0$. This is not neccesarily the case. For linear systems, the decay of the memory term can be linked to the eigenvalues of the fine-scale operator. For problems with positive fine-scale eigenvalues, the memory surface may grow with s. This will be discussed in the coming sections.

The second value of the kernel worth discussing is the derivative of the kernel with respect to s,

$$\frac{d}{ds}e^{t\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j} = e^{t\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}. \tag{3.46}$$

For a general s, Eq. 3.46 depends on the solution of the orthogonal dynamics. However, at $s = 0$, Eq. 3.46 has no dependence on the orthogonal dynamics,

$$\frac{\partial}{\partial s}K(\tilde{\mathbf{a}}(t), s)\bigg|_{s=0} = e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}. \tag{3.47}$$

Any order derivative of the kernel with respect to s at $s = 0$ can be expressed independently of the orthogonal dynamics. In general,

$$\frac{\partial^n}{\partial s^n} K(\tilde{\mathbf{a}}(t), s)\bigg|_{s=0} = e^{t\mathcal{L}} \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\big(\mathcal{Q}\mathcal{L}\big)^n a_{0j}. \tag{3.48}$$

3.3.2 The Orthogonal Dynamics

The previous subsection showed how the memory term can be viewed as a geometrical surface in the $s - t$ plane. Evaluating this surface requires solution to the orthogonal dynamics equation,

$$\frac{\partial}{\partial t} F_j(\mathbf{a}_0, t) = \mathcal{Q}\mathcal{L}F_j. \tag{3.49}$$

The orthogonal dynamics is a high dimensional partial differential equation. As previously mentioned, while the evolution operator $e^{t\mathcal{L}}$ is a Koopman operator, no such result exists for $e^{t\mathcal{Q}\mathcal{L}}$ in the general non-linear case. As a consequence, evaluating terms evolved by $e^{t\mathcal{Q}\mathcal{L}}$ requires one to directly solve the orthogonal dynamics equation. This is extremely difficult for non-linear problems. To the authors knowledge, no direct attempt has been made to solve the orthogonal dynamics equation. Extracting the memory kernel, even in an *a priori manner*, is intractable at this time.

The most general attempt to extract the memory term and orthogonal dynamics is presented in [20], where Hermite polynomials and Volterra integral equations are used to approximate the memory (and hence orthogonal dynamics). This procedure was shown to provide a reasonably accurate representation of the memory for a low-dimensional dynamical system. The procedure, however, is intractable for high-dimensional problems. This fact is exemplified in the work of Bernstein [7], where the methodology is applied to Burgers' equation and deemed intractable. Other attempts

to solve the orthogonal dynamics have been proposed in various contexts [58, 61]. To the authors understanding, none of these methods are tractable for the problems of interest to this dissertation.

3.3.3 The Orthogonal ODE

While solutions to the orthogonal dynamics are intractable for non-linear systems, progress can be made for linear systems. To demonstrate this, re-examine the multiscale equations for the linear initial value problem (Eqns. 3.13 and 3.14),

$$(\tilde{\mathbf{w}}, \tilde{u}_t) + (\tilde{\mathbf{w}}, L\tilde{u}) + (\tilde{\mathbf{w}}, Lu') = (\tilde{\mathbf{w}}, f), \tag{3.50}$$

$$(\mathbf{w}', u'_t) + (\mathbf{w}', L\tilde{u}) + (\mathbf{w}', Lu') = (\mathbf{w}', f). \tag{3.51}$$

As previously described, integrating factors can be used to analytically solve the fine scales to obtain,

$$(\tilde{\mathbf{w}}, \tilde{u}_t) + \big(\tilde{\mathbf{w}}, L\tilde{u}\big) - \Big(\mathbf{w}', \int_0^t L\mathbf{w}'^T e^{-s(\mathbf{w}', L\mathbf{w}'^T)}(\mathbf{w}', L\tilde{u}(t-s) - f)ds\Big) = \big(\tilde{\mathbf{w}}, f\big). \tag{3.52}$$

Compare this to the Mori-Zwanzig identity,

$$\frac{\partial}{\partial t} e^{t\mathcal{L}} a_{0j} = \underbrace{e^{tL}\mathcal{P}\mathcal{L}a_{0j}}_{\text{Markovian}} + \underbrace{\int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds}_{\text{Memory}} + \underbrace{e^{t\mathcal{Q}L}\mathcal{Q}\mathcal{L}a_{0j}}_{\text{Noise}}. \tag{3.53}$$

Notice that the evolution operator generated by the orthogonal dynamics is simply,

$$e^{t\mathcal{Q}\mathcal{L}} = e^{-t(\mathbf{w}', L\mathbf{w}'^T)},$$

and as such solutions to the orthogonal dynamics can be obtained by solving the ordinary differential equation

$$(\mathbf{w}', \frac{\partial}{\partial t} u'^{\mathcal{Q}}) + (\mathbf{w}', Lu'^{\mathcal{Q}}) = 0. \tag{3.54}$$

Equation 3.54 will be referred to as the orthogonal ODE. For linear systems, the orthogonal dynamics can be solved through a set of auxiliary ordinary differential equations.

The orthogonal ODE can be applied to non-linear systems through a linearization of the fine-scale equation. The multiscale non-linear equations were written as

$$(\tilde{\mathbf{w}}, \tilde{u}_t) + (\tilde{\mathbf{w}}, \mathcal{R}(u)) = (\tilde{\mathbf{w}}, f), \tag{3.55}$$

$$(\mathbf{w}', {u'}_t) + (\mathbf{w}', \mathcal{R}(u)) = (\mathbf{w}', f). \tag{3.56}$$

The fine-scale equation can be linearized about $u' = 0$ to obtain,

$$(\mathbf{w}', {u'}_t) + (\mathbf{w}', \mathcal{R}(\tilde{u})) + (\mathbf{w}', \mathcal{R}'u' + \mathcal{O}({u'}^2)) = (\mathbf{w}', f), \tag{3.57}$$

where again $\mathcal{R}' = \frac{\partial R}{\partial u}$. Dropping the higher order terms, the fine-scale equation becomes linear with respect to the fine scales and has the analytic solution,

$$u'(t) = -\int_0^t {\mathbf{w}'}^T e^{-s(\mathbf{w}', \mathcal{R}'\mathbf{w}')}(\mathbf{w}', \mathcal{R}(\tilde{u}(t-s)) - f)ds.$$

The linearization is accurate for small u'. The corresponding "orthogonal ODE" is,

$$(\mathbf{w}', \frac{\partial}{\partial t}{u'}^{\mathcal{Q}}) + (\mathbf{w}', \mathcal{R}'{u'}^{\mathcal{Q}}) = 0. \tag{3.58}$$

3.3.4 The Orthogonal ODE and its Procedure

In [43], Gouasmi et al. developed an orthogonal ODE in the context of composition operators and developed a numerical procedure to evaluate the memory term using the orthogonal ODE. The results using the orthogonal ODE presented in [43] can be shown to be equivalent for those obtained with Eq. 3.58 for systems with stable fixed points. Gouasmi recognized that the orthogonal ODE arises from assuming that $e^{t\mathcal{Q}\mathcal{L}}$ is a Koopman operator. Gouasmi's procedure uses the orthogonal ODE

to numerically evaluate the kernel using the trajectories of the resolved simulation. Despite the approximation in the orthogonal ODE, the reconstructed kernel has been shown to provide accurate representations of the memory for Burgers' equation and the Kuramoto-Sivashinsky equation [43].

3.4 Finite Memory of the Kernel

The ability to relate solutions of the orthogonal dynamics to a set of ordinary differential equations allows for great insight into the mechanics of the memory term. In particular, this provides insight into the idea of *finite memory*. The idea of finite memory has been suggested by various authors [93, 19] and poses the idea that the memory kernel has a finite temporal support,

$$\int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds \approx \int_{t-\tau}^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds. \tag{3.59}$$

This idea can be made concrete through the orthogonal ODE. By virtue of the orthogonal ODE, the fine scales were shown to have the solution,

$$u'(t) = -\int_0^t \mathbf{w}'^T e^{-s(\mathbf{w}', L\mathbf{w}'^T)}(\mathbf{w}', L\tilde{u}(t-s) - f)ds. \tag{3.60}$$

Equation 3.60 is exact in the linear case and is an approximation in the non-linear case. The linear case is considered for simplicity. Consider the eigen-decomposition of the fine-scale operator

$$(\mathbf{w}', L\mathbf{w}'^T) = \mathbf{S}\Lambda\mathbf{S}^{-1}, \tag{3.61}$$

where $\mathbf{S}$ and $\mathbf{\Lambda}$ are the eigenvectors and eigenvalues of the matrix $(\mathbf{w}', L\mathbf{w}'^T)$. With the eigen-decomposition, Eq. 3.60 can be written as

$$u'(t) = -\int_0^t \mathbf{w}'^T\mathbf{S}e^{-s\mathbf{\Lambda}}\mathbf{S}^{-1}(\mathbf{w}', \mathcal{R}(\tilde{u}(t-s)) - f)ds. \tag{3.62}$$

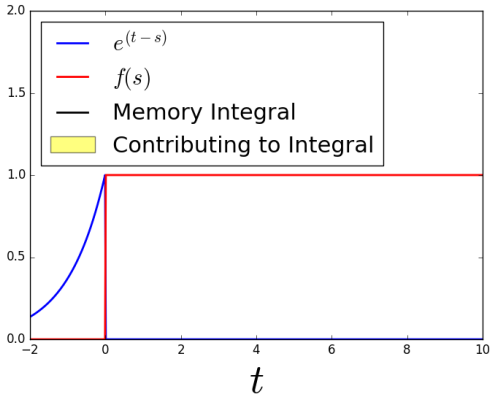
Equation 3.62 shows that, when the eigenvalues of the fine-scale operator are negative, the memory kernel has a finite support. The timescale of this support is proportional to the inverse of the eigenvalues. To explain this, a pictorial representation of the evaluation of a simple convolution integral is given in Figure 3.3. The figures show the graphical evaluation of the convolution integral $\int_0^t f(s)e^{\lambda(t-s)}ds$ with $f(t) = H(t)$[2] and $\lambda = -1$. It is seen that the exponential operator limits the support of the integrand. The time-scale of this support is related to the argument of the exponential operator, which is related to the eigenvalues of the fine-scale operator. It is worth noting that, while negative eigenvalues lead to a decaying memory term, positive eigenvalues lead to a growth in the memory term. Modeling the memory for this case is expected to be more challenging.

3.5 Summary

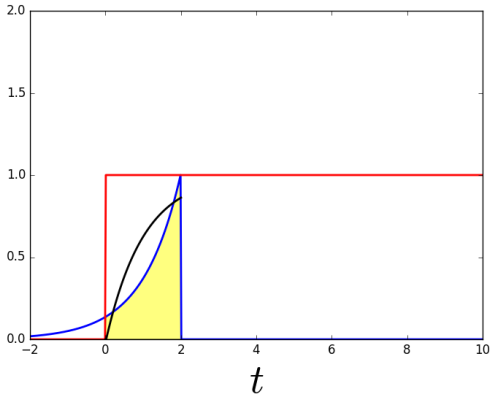
This chapter outlined Chorin's formulation of the Mori-Zwanzig formalism in the context of the variational multiscale method as an alternative to Hughes' fine-scale Green's function. This chapter demonstrated how model reduction leads to memory effects that are driven by a projection of the coarse-scale residual. For linear problems, this was achieved through the use of integrating factors. The nonlinear case achieved this through Chorin's formulation of Mori-Zwanzig. Important results of this chapter are:

- Formulation of Chorin's Mori-Zwanzig formulation for the variational multiscale method.
- Discovery that the memory is driven by an orthogonal projection of the coarse-scale residual. This displays similarities to many classic variational multiscale models. The fact that the memory is related to the coarse-scale residual is an as

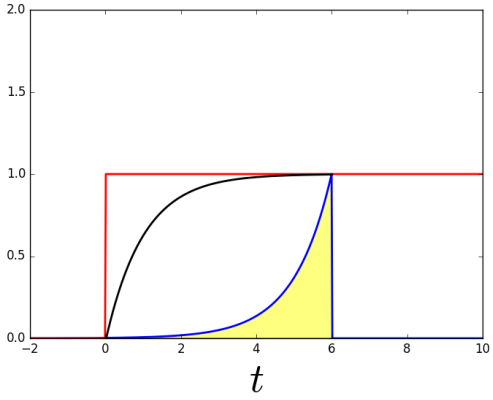
[2] $H(t)$ is the Heaviside function: $H(t) = 0\ \forall t < 0$, $H(t) = 1\ \forall t \geq 0$.



(a) $t = 0$.



(b) $t = 2$.



(c) $t = 6$.

Figure 3.3: Evolution of the convolution integral $\int_0^t f(s)e^{-(t-s)}ds$. To evaluate the convolution integral graphically, first reflect e^t, add a time offset, and then slide it along the t-axis. Then $f(t)$ is plotted as a function of t. The integral is the area under the curve of $e^{(t-s)}f(t)$.

of yet unappreciated fact. It demonstrates that MZ-based methods are, in fact, residual-based methods. It further provides great insight into the performance of MZ-based methods.

- Formulation of the orthogonal ODE. The orthogonal ODE allows for exact solutions to the orthogonal dynamics for linear problems and provides an approximation for non-linear problems.
- Discussion of the link between the memory length and the eigenvalues of the fine-scale operator.

A number of researchers have examined the application of Mori-Zwanzig-based models to the Navier-Stokes equation in a spectral Fourier-Galerkin setting [92, 15, 93, 95, 46]. The discussion to this point presents the general basis for these methods in the context of the variational multiscale method. The demonstration that the memory is driven by an orthogonal projection of the residual is a hitherto unappreciated fact that provides new insight into the mechanics of MZ-based models.

To this point, we have discussed how unresolved effects can be cast as a memory term that is a function of only the resolved scales. Directly evaluating this entire term, however, is intractable. The following chapter will discuss various strategies to model the memory.

CHAPTER IV

Modeling the Memory

The previous chapter outlined how the Mori-Zwanzig procedure can be used as an alternative to Hughes' fine-scale Green's function. The MZ-VMS procedure led to a coarse-scale equation that is non-local in time. The effect of the fine scales on the coarse scales was encoded in a memory term. Directly evaluating this memory term, however, is not practical. In the optimal scenario, solving the non-Markovian coarse-scale equation will have a similar cost to solving the original high-dimensional system. In a more realistic scenario, it is not clear if the memory term can even be evaluated. Similar to the Green's function, however, the memory term provides a starting point for the development of closure models.

The development of general Mori-Zwanzig closure models has been an area of interest in the research community[1]. Despite the complexity and minimal understanding of the orthogonal dynamics, various surrogate models for the memory exist. The t-model [19] is the first such MZ-based method. Since the introduction of the t-model, a number of researchers have examined more sophisticated models. Most notably, Stinis and coworkers [46, 92, 94, 93, 82] have developed several models for approximating the memory and have examined their performance on Burgers' equation and the Euler equations. More recently, Zhu, Dominy, and Venturi [105, 104]

[1] It is noted that none of these efforts have been performed within the scope of the variational multiscale method.

have examined additional approximations to the memory as well as error estimates for various MZ models.

This chapter outlines several prominent MZ-based methods. The energy stability of several models will be discussed.

4.1 The t-model and renormalized variants

The t-model devised by Chorin [20] approximates the memory using a zeroth order Taylor expansion of the memory about $s = 0$,

$$\int_0^t K(\tilde{\mathbf{a}}(t-s), s)ds \approx tK(\tilde{\mathbf{a}}(t), 0), \tag{4.1}$$

where $K(\tilde{\mathbf{a}}(t), 0) = e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$. The t-model may be alternatively derived via a rectangular quadrature rule starting from $s = 0$, or by approximating the orthogonal dynamics evolution operator with $e^{t\mathcal{Q}\mathcal{L}} \approx e^{t\mathcal{L}}$. The t-model has traditionally been viewed as a long memory model [18], and has been applied with varying degrees of success to Fourier-Galerkin solutions of Burgers' equation, the Euler equations, and incompressible Navier-Stokes equations.

More recently, it has been argued that the t-model requires an additional scaling to maintain accuracy. A class of renormalized models derived by Stinis [95] approximates the memory with a series expansion of the form,

$$\int_0^t K(\tilde{\mathbf{a}}(t-s), s)ds \approx \sum_{j=1}^{N} C_j(-1)^{j+1}\frac{t^j}{j!}\mathcal{P}e^{t\mathcal{L}}(\mathcal{P}\mathcal{L})^j\mathcal{Q}\mathcal{L}\mathbf{a}_{0k}. \tag{4.2}$$

Stinis [95] also presents a methodology to determine the coefficients C_j using certain norms of the full solution.

4.1.1 Energy Stability for Conservative Systems

In [46], a proof regarding the energy stability of the t-model was given. This proof is provided and elaborated on here for completeness.

Theorem IV.1. *Suppose the full order model satisfies*

$$\frac{dE}{dt} = 0, \tag{4.3}$$

where E is the energy as defined by,

$$E = (\tilde{\mathbf{a}}, \tilde{\mathbf{a}}) + (\mathbf{a}', \mathbf{a}'). \tag{4.4}$$

Then the total energy evolved by the t-model is given by,

$$\frac{dE^t}{dt} \leq 0. \tag{4.5}$$

Thus, if the t-model is constructed for an energy conserving system, the t-model will be dissipate energy. The same results extend to the τ and VMS(ϵ) models for $\tau > 0$.

Proof. The evolution equation for total energy is given by,

$$\frac{dE}{dt} = (\tilde{\mathbf{a}}, f(\tilde{\mathbf{a}}, \mathbf{a}')) + (\mathbf{a}', g(\tilde{\mathbf{a}}, \mathbf{a}')). \tag{4.6}$$

Setting $\mathbf{a}' = 0$ in Eq 4.6 gives

$$\big(\tilde{\mathbf{a}}, f(\tilde{\mathbf{a}}, 0)\big) = 0. \tag{4.7}$$

Next, let $\mathbf{a}' = \epsilon\mathbf{h}$,

$$\big(\tilde{\mathbf{a}}, f(\tilde{\mathbf{a}}, \epsilon\mathbf{h})\big) + \big(\epsilon\mathbf{h}, g(\tilde{\mathbf{a}}, \epsilon\mathbf{h})\big) = 0. \tag{4.8}$$

Expanding Eq. 4.8 in a Taylor series,

$$\big(\tilde{\mathbf{a}}, f(\tilde{\mathbf{a}}, 0)\big) + \epsilon\big(\tilde{\mathbf{a}}, \frac{\partial f}{\partial \mathbf{a}'}\big|_{\mathbf{a}'=\mathbf{0}}\mathbf{h}\big) + \epsilon\big(\mathbf{h}, g(\tilde{\mathbf{a}}, 0)\big) + \mathcal{O}(\epsilon^2) = 0. \tag{4.9}$$

Dividing by ϵ, letting $\epsilon \to 0$, inserting in Eq. 4.7, and letting $\mathbf{h} = g(\tilde{\mathbf{a}}, 0)$ yields,

$$\left(\tilde{\mathbf{a}}, \frac{\partial \mathbf{f}}{\partial \mathbf{a}'}\bigg|_{\mathbf{a}'=0} g(\tilde{\mathbf{a}}, 0)\right) = -\Big(g(\tilde{\mathbf{a}}, 0), g(\tilde{\mathbf{a}}, 0)\Big). \tag{4.10}$$

The t-model for this system can be written as

$$\frac{d\tilde{\mathbf{a}}}{dt} = f(\tilde{\mathbf{a}}, 0) + t\frac{\partial \mathbf{f}}{\partial \mathbf{a}'}\bigg|_{\mathbf{a}'=0} \mathbf{g}(\tilde{\mathbf{a}}, 0).$$

From Eq. 4.7 and Eq. 4.10,

$$\frac{dE}{dt} = -t\big(g(\tilde{\mathbf{a}}, 0), g(\tilde{\mathbf{a}}, 0)\big) \leq 0. \tag{4.11}$$

This completes the proof. □

4.1.2 Energy Stability for Dissipative Systems

Many numerical methods are not energy conserving. A relevant case is when viscosity is present. In this case, no proof exists showing that the t-model is dissipative. To illustrate this, assume the ODE system may be written as,

$$\frac{d\tilde{\mathbf{a}}}{dt} = f^i(\tilde{\mathbf{a}}, \mathbf{a}') + f^v(\tilde{\mathbf{a}}, \mathbf{a}'), \tag{4.12}$$

$$\frac{d\mathbf{a}'}{dt} = g^i(\tilde{\mathbf{a}}, \mathbf{a}') + g^v(\tilde{\mathbf{a}}, \mathbf{a}'), \tag{4.13}$$

where f^i and g^i are energy preserving functions while f^v and g^v dissipate energy. For example, f^v and g^v could be the viscous term. The equation for global energy is found by taking the inner product against $\mathbf{a}$,

$$\frac{dE}{dt} = \big(\tilde{\mathbf{a}}, f^i(\tilde{\mathbf{a}}, \mathbf{a}')\big) + \big(\mathbf{a}', g^i(\tilde{\mathbf{a}}, \mathbf{a}')\big) + \big(\tilde{\mathbf{a}}, f^v(\tilde{\mathbf{a}}, \mathbf{a}')\big) + \big(\mathbf{a}', g^v(\tilde{\mathbf{a}}, \mathbf{a}')\big). \tag{4.14}$$

The assumption that $\mathbf{f}^i$ and $\mathbf{g}^i$ are kinetic energy preserving implies,

$$\big(\tilde{\mathbf{a}}, f^i(\tilde{\mathbf{a}}, 0)\big) = 0, \tag{4.15}$$

$$\left(\tilde{\mathbf{a}}, \frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'}\bigg|_{\mathbf{a}'=0} \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right) = -\left(\mathbf{g}^i(\tilde{\mathbf{a}}, 0), \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right), \tag{4.16}$$

$$\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}^v(\tilde{\mathbf{a}}, 0)\right) = -\left(\mathbf{g}^v(\tilde{\mathbf{a}}, 0), \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right). \tag{4.17}$$

To determine if the t-model and τ-model are energy dissipative, consider first the energy budget for the truncated ODE system,

$$\frac{dE^{nm}}{dt} = \big(\tilde{\mathbf{a}}, f^i(\tilde{\mathbf{a}}, 0)\big) + \big(\tilde{\mathbf{a}}, f^v(\tilde{\mathbf{a}}, 0)\big). \tag{4.18}$$

Similarly, the energy budget for the truncated system with the t-model is given by

$$\frac{dE^t}{dt} = \big(\tilde{\mathbf{a}}, f^i(\tilde{\mathbf{a}}, 0)\big) + \big(\tilde{\mathbf{a}}, f^v(\tilde{\mathbf{a}}, 0)\big) + t\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}(\tilde{\mathbf{a}}, 0)\right), \tag{4.19}$$

where $\mathbf{f} = \mathbf{f}^i + \mathbf{f}^v$ and the same for $\mathbf{g}$. Subtracting Eq. 4.18 from Eq. 4.19 gives,

$$\frac{dE^t}{dt} - \frac{dE^{nm}}{dt} = t\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}(\tilde{\mathbf{a}}, 0)\right). \tag{4.20}$$

We can expand the right hand side,

$$\begin{aligned}\frac{dE^t}{dt} - \frac{dE^{nm}}{dt} = {} & t\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right) + t\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}^v(\tilde{\mathbf{a}}, 0)\right) + \\ & t\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}^v}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right) + t\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}^v}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}^v(\tilde{\mathbf{a}}, 0)\right).\end{aligned} \tag{4.21}$$

Using Eqns 4.16 and 4.17, we have,

$$\begin{aligned}\frac{dE^t}{dt} - \frac{dE^{nm}}{dt} = {} & -t\left(\mathbf{g}^i(\tilde{\mathbf{a}}, 0), \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right) - t\left(\mathbf{g}^v(\tilde{\mathbf{a}}, 0), \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right) + \\ & t\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}^v}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right) + t\left(\tilde{\mathbf{a}}, \left.\frac{\partial \mathbf{f}^v}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}^v(\tilde{\mathbf{a}}, 0)\right).\end{aligned} \tag{4.22}$$

There is no guarantee that Eq. 4.22 has a negative right hand side, and hence it is not possible to claim that the t and τ-models are purely dissipative. A special case worth considering is when f^v and g^v are only functions of $\tilde{\mathbf{a}}$ and $\mathbf{a}'$, respectively. Under these circumstances, the last three terms in Eq. 4.22 drop and we obtain

$$\frac{dE^t}{dt} - \frac{dE^{nm}}{dt} = -t\left(\mathbf{g}^i(\tilde{\mathbf{a}}, 0), \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\right). \tag{4.23}$$

This special case is encountered in Fourier-Galerkin spectral methods.

Remark

- The above proof considers conservation of "modal energy" at the discrete level. In later chapters, we will extend this proof to consider energy defined at the continuous level.

4.2 The τ-model

The τ-model [2] is motivated by the idea that memory has a finite support in time and makes the approximation,

$$\int_0^t K(\tilde{\mathbf{a}}(t-s), s)ds \approx \tau K(\tilde{\mathbf{a}}(t), 0), \tag{4.24}$$

where τ is the memory length. Approximating the memory with the τ-model gives rise to the following closed equations for the coarse scales.

$$(\tilde{\mathbf{w}}, \tilde{u}_t) + (\tilde{\mathbf{w}}, \mathcal{R}(\tilde{u})) - \tau\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t), 0) = (\tilde{\mathbf{w}}, f). \tag{4.25}$$

As the structure of the τ-model is equivalent to the structure of the t-model, the same proof regarding energy stability presented above applies to the τ-model.

4.2.1 Relationship to Adjoint-Stabilization and the VMS(ϵ) Model

Insight into the τ-model can be obtained by examining the linear case. Let $\mathcal{R}$ be a linear operator, $\mathcal{R} = L$. The steady-state solution achieved by the τ-model is

$$(\tilde{\mathbf{w}}, L\tilde{u}) - \tau\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t), 0) = (\tilde{\mathbf{w}}, f). \tag{4.26}$$

Integrating by parts, the memory at $s = 0$ may be expressed as

$$\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t), 0) = \int_\Omega \int_\Omega (L^*\tilde{\mathbf{w}})(x)\Pi'(x, y)(L\tilde{u} - f)(y)d\Omega_y d\Omega_x. \tag{4.27}$$

The memory term has the same form as that obtained by Hughes' in Ref. [48]. In the steady case, the orthogonal projection can be viewed as an approximation to the

fine-scale Green's function,

$$g'(x,y) \approx \tau(x,y)\Pi'(x,y). \tag{4.28}$$

The τ-model can be further viewed as a variant of the adjoint stabilization method, where the only difference is the inclusion of the orthogonal projector. Further, the τ-model has the same model form the VMS(ϵ) model in the case where the fine scales are defined to be orthogonal to the coarse scales.

4.3 The Dynamic-τ model

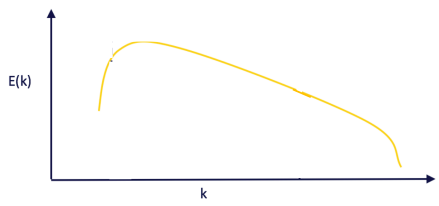
The dynamic-τ model, which is developed by the author in Ref. [74], is a parameter-free model that leverages the idea of finite memory, the Germano identity, and the value of the memory at $s = 0$. An advantage of the dynamic-τ model is that it only depends on the value of the kernel at $K(\tilde{\mathbf{a}}(t), 0)$ and does not require the computation of any high-order terms (i.e. $\mathcal{PLQLQLQL}a_{0j}$).

The derivation of the dynamic-τ model begins with the assumption that the memory has a finite support,

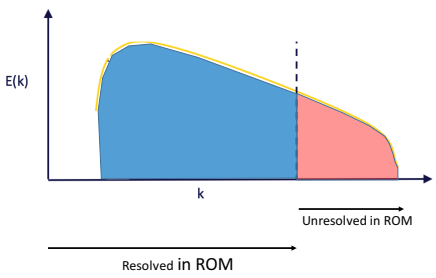
$$\int_0^t e^{(t-s)\mathcal{L}}\mathcal{PL}e^{s\mathcal{QL}}\mathcal{QL}\tilde{\mathbf{a}}_0 ds \approx \int_{t-\tau}^t e^{(t-s)\mathcal{L}}\mathcal{PL}e^{s\mathcal{QL}}\mathcal{QL}\tilde{\mathbf{a}}_0. \tag{4.29}$$

The numerical evidence for finite memory was presented in the previous chapter. There is also a theoretical argument for the presence of a finite memory. For problems that reach a statistically steady state (such as a turbulent channel flow), the memory should also reach a statistically steady state.

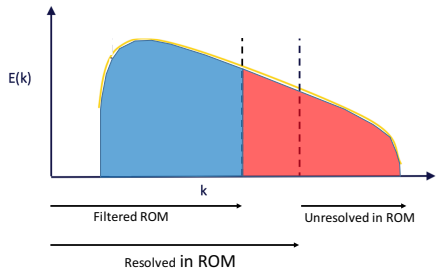
The derivation of the dynamic-τ model proceeds by using a single point quadrature rule starting from $s = 0$ to approximate the memory. The memory length is taken to be of an unknown length, $\tau_{\mathcal{P}}$. The finite memory assumption implies that $K(\tilde{\mathbf{a}}(t -$



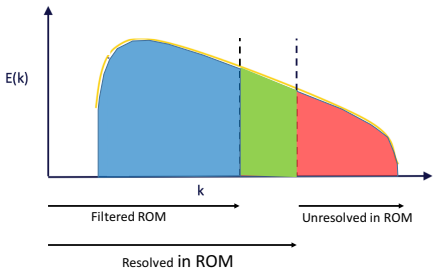
(a) Qualitative energy spectrum for a full order simulation.



(b) Decomposition into resolved and unresolved subspaces.



(c) The Germano identity performs an additional decomposition onto a "test" subspace.



(d) Information from the overlapped subspace is used to determine $\tau_{\mathcal{P}}$.

Figure 4.1: Graphical depiction of the Germano identity.

$\tau_{\mathcal{P}}, \tau_{\mathcal{P}})) = 0$. Under these assumptions the memory simplifies to,

$$\int_0^t e^{(t-s)\mathcal{L}} \mathcal{P}\mathcal{L} e^{s\mathcal{Q}\mathcal{L}} \mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 ds \approx C_q \tau_{\mathcal{P}}(t) e^{t\mathcal{L}} \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0, \tag{4.30}$$

where C_q is the constant associated with the quadrature rule. For example, a trapezoidal quadrature would take $C_q = 0.5$. One could alternatively assume the memory decays in an exponential fashion, sinusoidal fashion, etc. As in the t-model, the key simplification thus far is that we have eliminated the dependence on the orthogonal dynamics. Additionally, note that the above model can recover the t-model if $C_q \tau_{\mathcal{P}} = t$.

The model described in Eq. 4.30 requires the specification of the memory length $\tau_{\mathcal{P}}$. Here, a dynamic procedure using the Germano identity [40] is employed. A graphical depiction of the Germano procedure is provided in Figure 4.1.

To proceed, decompose the resolved variable $\tilde{\mathbf{a}}$ into two sets such that,

$$\tilde{\mathbf{a}} = \{\overline{\mathbf{a}}, \hat{\mathbf{a}}\}, \qquad \mathbf{a} = \{\overline{\mathbf{a}}, \hat{\mathbf{a}}, \mathbf{a}'\}. \tag{4.31}$$

Further, define the sharp cutoff filters $\tilde{\mathcal{G}}$ and $\overline{\mathcal{G}}$ that satisfy

$$\tilde{\mathcal{G}}\mathbf{a} = \tilde{\mathbf{a}} \qquad \overline{\mathcal{G}}\mathbf{a} = \overline{\mathbf{a}}.$$

Note that the filters do not commute with non-linear functions, $\tilde{\mathcal{G}}f(\mathbf{a}) \neq \tilde{f}(\tilde{\mathbf{a}})$; instead they act as a traditional sharp spectral cutoff filter. To derive an expression for $\tau_{\mathcal{P}}$, note the following identity for the zero-variance projection,

$$\mathcal{P}\int_0^t K_j(\tilde{\mathbf{a}}(s), t-s)ds = \tilde{\mathcal{G}}R_j(\mathbf{a}) - R_j(\tilde{\mathcal{G}}\mathbf{a}). \tag{4.32}$$

Applying the "test" filter $\overline{\mathcal{G}}$ to Eq. 4.32 and adding and subtracting $R_j(\overline{\mathcal{G}}\mathbf{a})$ yields,

$$\overline{\mathcal{G}}\mathcal{P}\int_0^t K_j(\tilde{\mathbf{a}}(s), t-s)ds = \left[\overline{\mathcal{G}}R_j(\mathbf{a}) - R_j(\overline{\mathbf{a}})\right] + \left[R_j(\overline{\mathbf{a}}) - \overline{\mathcal{G}}R_j(\tilde{\mathbf{a}})\right], \tag{4.33}$$

where we have used used the identities $\overline{\mathcal{G}}\tilde{\mathcal{G}}f(\mathbf{a}) = \overline{\mathcal{G}}f(\mathbf{a})$ and $\overline{\mathcal{G}}\mathbf{a} = \overline{\mathbf{a}}$. The first term on the RHS is simply the unclosed term that arises from filtering at a level $\overline{\mathcal{G}}$. Equation 4.33 is a statement of the Germano identity. The second bracketed term on the RHS can be computed from the under-resolved simulation. The memory length $\tau_{\mathcal{P}}$ can be approximated by using the τ-model to model the LHS and the first term on the RHS,

$$C_q\overline{\mathcal{G}}e^{t\mathcal{L}}\tau_{\mathcal{P}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j} = C_q e^{t\mathcal{L}}\tau_{\overline{\mathcal{P}}}\overline{\mathcal{P}}\mathcal{L}\overline{\mathcal{Q}}\mathcal{L}a_{0j} + \left[R_j(\overline{\mathbf{a}}) - \overline{\mathcal{G}}R_j(\tilde{\mathbf{a}})\right], \tag{4.34}$$

where $\overline{\mathcal{P}}$ and $\overline{\mathcal{Q}}$ are the corresponding MZ projection operators at the level $\overline{\mathcal{G}}$ (i.e. $\overline{\mathcal{P}}f(\mathbf{a}) = f(\overline{\mathbf{a}})$), $\tau_{\mathcal{P}}$ is the time-scale for coarse-graining at the level $\mathcal{P}$, and $\tau_{\overline{\mathcal{P}}}$ is the time-scale for coarse-graining at the level $\overline{\mathcal{P}}$.

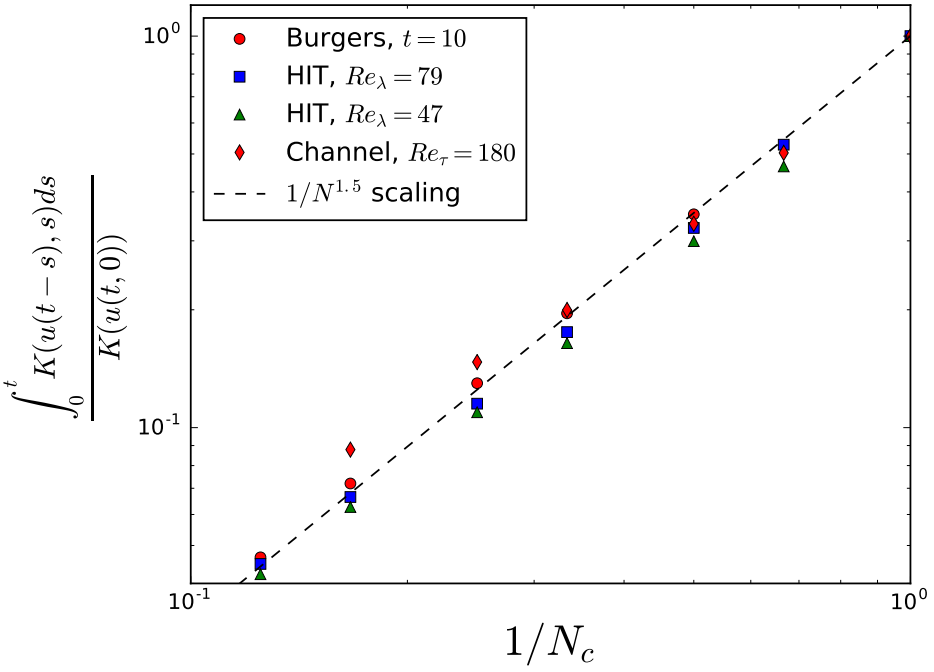


Figure 4.2: Ratio of the memory kernel integral (as extracted from the DNS) to the value of the kernel at $s = 0$ for Burgers equation, homogeneous turbulence, and channel flow. All terms are normalized. The x-axis is the ratio of the number of degrees of freedom in the LES to the number of degrees of freedom in the full-order simulation. Note that the integrated kernel can be computed from the full-order simulation as $e^{t\mathcal{L}}\mathcal{Q}\mathcal{L}u_{0j}$.

4.3.1 Scaling Laws for the Memory Length

To close Eq. 4.34, a constitutive relation between the time-scales $\tau_{\mathcal{P}}$ and $\tau_{\overline{\mathcal{P}}}$ needs to be established. To establish this relationship, the ratio between the integrated memory and memory at $s = 0$ is examined *a priori* for a variety of flows. Figure 4.2 shows the ratio of the integrated memory to that of the memory kernel at $s = 0$ for Burgers' equation, homogeneous turbulence at several Reynolds numbers, and channel flow. The memory-length is seen to approximately obey a $1/N^{1.5}$ scaling. As such, the relationship between $\tau_{\mathcal{P}}$ and $\tau_{\overline{\mathcal{P}}}$ is straightforward,

$$\tau_{\overline{\mathcal{P}}} = \left(\frac{\Delta_{\mathcal{P}}}{\Delta_{\overline{\mathcal{P}}}}\right)^{1.5} \tau_{\mathcal{P}}, \tag{4.35}$$

where Δ is the corresponding filter scale. The resulting equation for the memory length, $\tau_{\mathcal{P}}$, is

$$\tau_{\mathcal{P}} = \frac{1}{C_q} \frac{R_j(\overline{\mathcal{G}}\mathbf{a}) - \overline{\mathcal{G}} R_j(\hat{\mathcal{G}}\mathbf{a})}{\overline{\mathcal{G}} e^{t\mathcal{L}} \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L} a_{0j} - \left(\frac{\Delta_{\mathcal{P}}}{\Delta_{\overline{\mathcal{P}}}}\right)^{1.5} e^{t\mathcal{L}} \overline{\mathcal{P}}\mathcal{L}\overline{\mathcal{Q}}\mathcal{L} a_{0j}}. \tag{4.36}$$

It is noted that Eq. 4.36 still contains the unknown quadrature constant, C_q. This term, however, vanishes when Eq. 4.36 is injected into Eq. 4.30.

Figure 4.2 provides insight into the effect of resolution on memory effects. As the resolution level of a reduced-order model decreases, the time-scale of the memory term increases. This demonstrates how coarsening a numerical simulation leads to increased memory effects. It is noted, however, that these time-scales are a function of both the underlying numerics and physics.

4.3.2 Energy Transfer Constraint

Equation 4.34 is an over-determined system for $\tau_{\mathcal{P}}$. It is unlikely that one scalar $\tau_{\mathcal{P}}$ will exist that can satisfy Eq. 4.34 for each mode. As in the dynamic Smagorinsky model, Eq. 4.34 is understood to be valid in an average sense only. To constrain the equation for a single scalar $\tau_{\mathcal{P}}$, the coarse-graining time scale is estimated by equating the energy transferred by the subgrid model out of the modes $\overline{\mathbf{a}}$. To see how this is done, consider the system written discretely as,

$$\frac{da_j}{dt} = R_j(\mathbf{a}). \tag{4.37}$$

The evolution of the total energy is then given by,

$$\frac{dE}{dt} = \left(\mathbf{a}, \frac{d\mathbf{a}}{dt}\right) = \Big(\mathbf{a}, \mathbf{R}(\mathbf{a})\Big), \tag{4.38}$$

where $(\mathbf{a}, \mathbf{b})$ denotes an aptly defined inner product. For a Fourier Galerkin method, the inner product is simply $(\mathbf{a}, \mathbf{b}) = \sum_j a_j^* b_j$. The energy transfered out of the

resolved modes $\overline{\mathbf{a}}$ is then given by,

$$\frac{d\overline{E}}{dt} = \left(\overline{\mathbf{a}}, \mathbf{R}(\mathbf{a})\right). \tag{4.39}$$

Equating the energy transfer out of the modes in $\overline{\mathbf{a}}$ leads to the following equation for $\tau_{\mathcal{P}}$,

$$\tau_{\mathcal{P}} = \frac{1}{C_q} \frac{\mathbb{R}\left(\overline{\mathbf{a}}_j, R_j(\overline{\mathbf{a}}) - \overline{\mathcal{G}} R_j(\hat{\mathbf{a}})\right)}{\left(\overline{\mathbf{a}}_0, e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\mathbf{a}_0 - \left(\frac{\Delta_{\mathcal{P}}}{\Delta_{\overline{\mathcal{P}}}}\right)^{1.5} e^{t\mathcal{L}}\overline{\mathcal{P}}\mathcal{L}\overline{\mathcal{Q}}\mathcal{L}\mathbf{a}_0\right)}. \tag{4.40}$$

With Eqns. 4.30 and 4.40, the dynamic model for the memory integral is closed. The final expression for the closure model is

(4.41)

$$\mathcal{P}\int_0^t e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds \frac{\mathbb{R}\left(\overline{\mathbf{a}}_j, R_j(\overline{\mathbf{a}}) - \overline{\mathcal{G}} R_j(\hat{\mathbf{a}})\right)}{\left(\overline{\mathbf{a}}_0, e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\mathbf{a}_0 - \left(\frac{\Delta_{\mathcal{P}}}{\Delta_{\overline{\mathcal{P}}}}\right)^{1.5} e^{t\mathcal{L}}\overline{\mathcal{P}}\mathcal{L}\overline{\mathcal{Q}}\mathcal{L}\mathbf{a}_0\right)} e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}.$$

Note that C_q is canceled out in the dynamic procedure. The key assumptions in the dynamic-τ model are that the memory integral is well correlated to its value at $s = 0$ (Eq. 4.30) and that the scaling law (Eq. 4.35) holds. The stronger of these two assumptions is that the memory integral is well correlated to its value at $s = 0$. For decreasing resolution levels, the orthogonal dynamics are expected to grow in complexity and the Markovian approximation to the integral may not be sufficient.

4.4 Finite Memory Models

A hierarchy of models that again leverage the idea of finite memory was constructed by Stinis [46, 93]. The idea is to repeatedly differentiate the memory with respect to time to obtain a set of hierarchical equations, which are then truncated. As these models will be used later in this work, a thorough description is given.

Define,

$$w_j^{(m)}(\mathbf{a}_0,t) = \mathcal{P}\int_{a_m(t)}^{t} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\big(\mathcal{Q}\mathcal{L}\big)^{m+1}a_{0j}ds, \tag{4.42}$$

where $a_m(t) = t - \tau_m(t)$. For clarity of presentation, the dependence of $\mathbf{a}'$ on the initial conditions $\mathbf{a}_0$ and time t will be implicitly assumed throughout the rest of this section. Setting $m = 0$ (in which case Eq. 4.42 is simply the memory term) and differentiating Eq. 4.42 with respect to time yields,

$$\begin{aligned}\frac{d}{dt}w_j^{(0)}(\mathbf{a}_0,t) &= e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j} - \\ & e^{(t-\tau_0)\mathcal{L}}\mathcal{P}\mathcal{L}e^{\tau_0\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}a_0'(t) + \mathcal{P}\int_{a_0(t)}^{t} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}ds.\end{aligned} \tag{4.43}$$

Note that the first term on the right hand side does not require the solution of the orthogonal dynamics equation. The second term on the right hand side is dependent on the orthogonal dynamics. This dependence can be eliminated by using a discrete integration scheme to express the memory integral. In the case of the trapezoidal rule,

$$w_j^{(0)}(t) = \left[e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j} + e^{(t-\tau_0)\mathcal{L}}\mathcal{P}\mathcal{L}e^{\tau_0\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}\right]\frac{\tau_0(t)}{2} + \mathcal{O}(\tau_0^2). \tag{4.44}$$

In order to handle the case where the memory length τ_0 is not necessarily small, the memory integral in Eq. 4.42 is partitioned into N sub-intervals,

$$\begin{aligned}&\mathcal{P}\int_{t-\tau_0}^{t} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds = \mathcal{P}\int_{t-\Delta\tau_0}^{t} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds + \\ &\mathcal{P}\int_{t-2\Delta\tau_0}^{t-\Delta\tau_0} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds + \ldots + \mathcal{P}\int_{t-N\Delta\tau_0}^{t-(N-1)\Delta\tau_0} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds,\end{aligned} \tag{4.45}$$

where $\Delta\tau_0 = \tau_0/N$. Define,

$$w_j^{(m,n)} = \mathcal{P}\int_{t-n\Delta\tau_0}^{t-(n-1)\Delta\tau_0} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\big(\mathcal{Q}\mathcal{L}\big)^{m+1}a_{0j}ds,$$

for $n = 1, 2, ..., N$. Applying the trapezoidal integration scheme to each sub-interval yields the general form

$$\frac{d}{dt}w_j^{(0,n)}(a_0,t) = \left[\sum_{i=1}^{n-1}(-1)^{n+i+1}w_j^{(0,i)}\right]\frac{2}{\Delta\tau_0}\left(2-(2n-1)\Delta\tau_0'\right) - \frac{2}{\Delta\tau_0}w_j^{(0,n)}\left(1-n\Delta\tau_0'\right) + \left(2-(2n-1)\Delta\tau_0'\right)e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j} + \mathcal{P}\int_{t-n\Delta\tau_0}^{t-(n-1)\Delta\tau_0} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}ds + \mathcal{O}(\Delta\tau_0^2). \tag{4.46}$$

The right hand side of Eq. 4.46 is now closed with the exception of the memory term. Assume that the new memory term has a finite support from $t-\tau_1(t)$ to t. A differential equation for $w^{(1,n)}$ can be developed by again differentiating the convolution integral in Eq. 4.46 with respect to time and using the trapezoidal rule. The differentiation process can be continued to build an infinite hierarchy of Markovian equations. The general form obtained is

$$\frac{d}{dt}w_j^{(m,n)}(\mathbf{a}_0,t) = \left[\sum_{i=1}^{n-1}(-1)^{n+i+1}w_j^{(m,i)}\right]\frac{2}{\Delta\tau_m}\left(2-(2n-1)\Delta\tau_m'\right) - \frac{2}{\Delta\tau_m}w_j^{(m,n)}\left(1-n\Delta\tau_m'\right) + \left(2-(2n-1)\Delta\tau_m'\right)e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\left(\mathcal{Q}\mathcal{L}\right)^{m+1}a_{0j} + \mathcal{P}\int_{t-n\Delta\tau_m}^{t-(n-1)\Delta\tau_m} e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}(\mathcal{Q}\mathcal{L})^{m+2}a_{0j}ds + \mathcal{O}(\Delta\tau_m^2). \tag{4.47}$$

The infinite hierarchy of equations must be truncated at some point. This can be accomplished by modeling the effects of $w_j^{(m+1,n)}$ or, more simply, by neglecting it. Neglecting $w_j^{(m+1,n)}$ can be justified if the support or magnitude of the integrand decreases with the repeated application of $\mathcal{Q}\mathcal{L}$. The derivation above can be carried out using higher order quadrature [93]. In this work, models with a constant memory length and one sub-interval are considered. In this case, the models simplify to,

$$\frac{d}{dt}w_j^{(m)}(\mathbf{a}_0,t) = -\frac{2}{\tau_m}w_j^{(m)}(t) + 2e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\left(\mathcal{Q}\mathcal{L}\right)^{m+1}a_{0j} + w_j^{(m+1)}. \tag{4.48}$$

4.4.1 Alternative Setting for the Finite Memory Models

The first order class of finite memory models derived by Stinis can be obtained through an approximation to the orthogonal dynamics. A first order approximation to the orthogonal dynamics operator is,

$$e^{t\mathcal{Q}\mathcal{L}} \approx e^{-\frac{t}{\tau_0}}, \tag{4.49}$$

where τ_0 is some positive constant. Inserting this simplification into the memory kernel, we obtain,

$$\int_0^t e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds \approx \int_0^t e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{-\frac{t-s}{\tau_0}}\mathcal{Q}\mathcal{L}a_{0j}ds. \tag{4.50}$$

Differentiating the above with respect to time,

$$\frac{dw_j}{dt} = e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j} - \frac{1}{\tau_0}\int_0^t e^{s\mathcal{L}}\mathcal{P}\mathcal{L}e^{-\frac{t-s}{\tau_0}}\mathcal{Q}\mathcal{L}a_{0j}ds. \tag{4.51}$$

Noting that the last term on the right hand side is $1/\tau_0 w_j$,

$$\frac{dw_j}{dt} = -\frac{1}{\tau_0}w_j + e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}. \tag{4.52}$$

Equation 4.52 is almost equivalent to Stinis' first order finite memory model when one sub-integration is used. The first order finite memory model of Stinis' can be recovered with the approximation $e^{t\mathcal{Q}\mathcal{L}} \approx 2e^{-\frac{t}{\tau_0}}$, which yields,

$$\frac{dw_j}{dt} = -\frac{2}{\tau_0}w_j + 2e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}. \tag{4.53}$$

Equation 4.53 is equivalent to the first order finite memory models of Stinis. It is also important to note that the time-scale in the finite memory model is different than the timescale in the dynamic-τ model. The timescale in the dynamic-τ model is the timescale for which the integrand of the memory kernel decays to zero. The

timescale in Eq. 4.52 is the timescale at which the orthogonal dynamics decays. To illustrate the difference in these timescales, consider some toy memory term,

$$\int_0^t e^{A_{22}(t-s)} ds. \tag{4.54}$$

In this case, the orthogonal dynamics operator is $e^{t\mathcal{Q}\mathcal{L}} = e^{A_{22}t}$. The timescale in the dynamic-τ model is the timescale at which the integral decays. For example, the timescale could be defined as what memory length is necessary to recover 95% of the integral,

$$\int_{t-\tau}^t e^{A_{22}(t-s)} ds = 0.95 \int_0^t e^{A_{22}(t-s)} ds. \tag{4.55}$$

The above can be solved for τ,

$$\tau = \frac{1}{A_{22}} \log\left(0.95 e^{A_{22}t} + 0.05\right). \tag{4.56}$$

The timescale for the finite memory model, however, is simply the timescale at which the orthogonal dynamics decays,

$$\tau_0 = \frac{1}{-A_{22}}. \tag{4.57}$$

It is seen that, since the dynamic-τ model is an approximation of the integrand, it must predict the growing timescale of the integrand. The finite memory models, however, are an approximation to the orthogonal dynamics. A constant timescale in the finite memory models leads to an increasing timescale of the integrand.

4.5 Additional Models

A variety of additional MZ-based methods exist. For instance, a class of models has recently been developed by Price and Stinis [82] that assumes an almost commutativity of of $\mathcal{P}\mathcal{L}$ and $\mathcal{Q}\mathcal{L}$ to obtain an expansion of the memory term that is

demonstrated to be accurate for long time intervals. The second order expansion is given by

$$\int_0^t K(\tilde{\mathbf{a}}(t-s),s)ds \approx t\mathcal{P}e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 - \frac{t^2}{2}\mathcal{P}e^{t\mathcal{L}}\Big[\mathcal{P}\mathcal{L}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 - \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0\Big]. \tag{4.58}$$

Another class of models has been developed by Zhu and Venturi [105], who consider approximations to the memory that are based on a Faber series expansion of the orthogonal dynamics,

$$e^{t\mathcal{Q}\mathcal{L}} = \sum_{j=0}^{\infty} a_j(t)F_j(\mathcal{Q}\mathcal{L}), \tag{4.59}$$

where F_j is the j_{th} order Faber polynomial and a_j are the basis coefficients. In addition, [105] provides a cogent summary of several additional expansion techniques used to approximate the memory.

These additional models will not be examined in this work.

4.6 Numerical Evaluation of $e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}$

As evidenced in this chapter, most MZ-based models require the evaluation of $e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}$. This term can be computed by analytically evaluating the Liouville and projection operators (note that $\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}$ is applied to the initial conditions and hence the evaluation is simply an exercise in algebra; examples can be found in [21]). The naive analytic evaluation of such terms, however, is extremely tedious for general, complex systems of equations. The complexity of the derivations can be greatly simplified by recognizing that application of the Liouville operator to a vector $\mathbf{v}(\mathbf{a}_0)$ can be written as a matrix vector product [2],

$$\mathcal{L}\mathbf{v} = \frac{\partial \mathbf{v}}{\partial \mathbf{a_0}}\mathbf{R}(\mathbf{a_0}). \tag{4.60}$$

[2]The author thanks Ayoub Gouasmi for his work on MZ, which helped lead to this revelation.

Evaluating the Jacobian of $\mathbf{v}$ and matrix vector product in Eq. 4.60 can be costly and tedious. Fortunately, the only quantity of interest is the final matrix vector product, which is simply the Fréchet derivative of $\mathbf{v}$ in the direction of $\mathbf{R}$,

$$\mathcal{L}\mathbf{v} = \nabla_{R(\mathbf{a_0})}\mathbf{v} = \lim_{\epsilon \to 0} \frac{\mathbf{v}\big(\mathbf{a_0} + \epsilon \mathbf{R}(\mathbf{a_0})\big) - \mathbf{v}(\mathbf{a_0})}{\epsilon}. \tag{4.61}$$

Two techniques are considered to compute the Fréchet derivative for **general** ODE systems. The most straightforward approach is to approximate the derivative with a first order finite difference,

$$\mathcal{L}\mathbf{v}(\mathbf{a}_0) \approx \frac{\mathbf{v}\big(\mathbf{a_0} + \epsilon \mathbf{R}(\mathbf{a_0})\big) - \mathbf{v}(\mathbf{a_0})}{\epsilon}. \tag{4.62}$$

To evaluate $\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}$ using the first order finite difference approximation for the zero variance projection operators, define the vector field $\mathbf{v}$ to be

$$\mathbf{v}(\mathbf{a}_0) = \mathcal{Q}\mathcal{L}\mathbf{a}_0 = \mathbf{R}(\mathbf{a}_0) - \mathbf{R}(\tilde{\mathcal{G}}\mathbf{a}_0). \tag{4.63}$$

Applying the projection operator and the finite difference approximation of the Liouville operator to Eq. 4.63 yields

$$\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j} = \left[\frac{R_j\big(\tilde{\mathcal{G}}\mathbf{a_0} + \mathbf{R}(\tilde{\mathcal{G}}\mathbf{a_0})\big) - R_j\big(\tilde{G}\big[\tilde{\mathcal{G}}\mathbf{a_0} + \epsilon\mathbf{R}(\tilde{\mathcal{G}}\mathbf{a_0})\big]\big)}{\epsilon} - \frac{R_j\big(\tilde{\mathcal{G}}\mathbf{a_0}\big) - R_j\big(\tilde{\mathcal{G}}\mathbf{a_0}\big)}{\epsilon}\right]. \tag{4.64}$$

Noting that the last term on the right hand side is zero, one obtains

$$e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j} = \frac{R_j\big(\tilde{\mathcal{G}}\mathbf{a} + \epsilon\mathbf{R}(\tilde{\mathcal{G}}\mathbf{a})\big) - R_j\big(\tilde{G}\big[\tilde{\mathcal{G}}\mathbf{a} + \epsilon\mathbf{R}(\tilde{\mathcal{G}}\mathbf{a})\big]\big)}{\epsilon}. \tag{4.65}$$

We can further recognize that Eq. 4.65 is a perturbation about $\tilde{\mathcal{G}}\mathbf{a} + \epsilon\mathbf{R}(\tilde{\mathcal{G}}\mathbf{a})$, which itself is a perturbation about $\tilde{\mathcal{G}}\mathbf{a}$. In the limit of $\epsilon \to 0$, Eq. 4.65 is the same as an expansion about $\tilde{\mathbf{a}}$,

$$e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j} = \frac{R_j\big(\tilde{\mathbf{a}} + \epsilon(\mathbf{I} - \tilde{\mathcal{G}})\mathbf{R}(\tilde{\mathcal{G}}\mathbf{a})\big) - R_j\big(\tilde{\mathbf{a}}\big)}{\epsilon}. \tag{4.66}$$

The finite difference approximation to $\mathcal{PLQL}a_{0j}$ is attractive due to its simplicity. Further, the procedure can be used to evaluate higher order terms (i.e. $\mathcal{PL}(\mathcal{QL})^n a_{0j}$) through recursive differentiation of functions. This provides a potential pathway to the generation of automated MZ-based closures. Several issues, however, arise. First, the selection of the finite difference step size ϵ can be problematic since the magnitude of the direction imposed by $\mathbf{R}(\tilde{\mathbf{a}})$ may vary significantly. This can lead to both cancelation and truncation errors. While the present author has not found the finite difference approximation to be overly sensitive to ϵ for the problems considered, it is still a point of concern (especially for the evaluation of higher order terms). The second issue with the above approach is the computational cost. Evaluation of $e^{t\mathcal{L}}\mathcal{PLQL}a_{0j}$ requires an additional residual evaluation and must be computed in an enriched space. We do note that this cost can be reduced for certain problems due to symmetries (this will be seen later). Nonetheless, this approach increases the computational cost of a reduced model significantly.

The second approach considered to evaluate $\mathcal{PLQL}a_{0j}$ is to compute the exact Fréchet derivative. For notational purposes, consider the ODE system written as

$$\begin{aligned} \frac{d\tilde{\mathbf{a}}}{dt} &= f(\tilde{\mathbf{a}}, \mathbf{a}'), \\ \frac{d\mathbf{a}'}{dt} &= g(\tilde{\mathbf{a}}, \mathbf{a}'). \end{aligned} \tag{4.67}$$

The exact Fréchet derivative (and continuous counterpart of Eq. 4.66) is given by

$$e^{t\mathcal{L}}\mathcal{PLQL}\tilde{\mathbf{a}}_0 = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{a}'}\right|_{\mathbf{a}'=0} \mathbf{g}(\tilde{\mathbf{a}}, 0).$$

The exact derivative can be computed by linearizing the functional form produced by $\mathcal{QL}\mathbf{a_0}$, and then evaluating the resulting function at $R(\tilde{\mathcal{G}}\mathbf{a})$. While more tedious, this approach is attractive in that the exact functional form of the model is obtained and it is computationally more affordable.

4.7 Chapter Summary

This chapter outlined compact approximations to the memory. The t, τ, dynamic-τ, and finite memory models were presented. Several important developments in the chapter are:

- Development of the dynamic-τ model. The dynamic-τ model leverages finite memory effects and the Germano identity to dynamically compute a timescale which conserves the energy transfer across two levels of coarse-graining. The dynamic-τ model is the first parameter-free MZ model that does not explicitly contain a time term.
- The proof showing that, for systems that are kinetic energy conserving, the t and τ-models are globally dissipative.
- Demonstration that the τ and VMS(ϵ) models are equivalent in structure.
- The demonstration that the first order finite memory model derived by Stinis can alternatively be derived by the approximation $e^{t\mathcal{Q}\mathcal{L}} \approx 2e^{-\frac{t}{\tau}}$.

At this stage in the work, we have laid out the majority of the foundations of how Mori-Zwanzig and the variational multiscale method can be used to develop closure models for Galerkin discretizations. The remainder of this work will examine the performance of the models in practice. We will first examine spectral methods and then extend our scope to finite element methods.

CHAPTER V

MZ-VMS for Spectral Methods

This chapter will consider the application of MZ-VMS models to numerical simulations based on the Galerkin spectral method. In the spectral method, the state variables are expressed in a global basis that has support over the entire domain,

$$\mathbf{u}(\mathbf{x}, t) = \sum_{j=1}^{N} \mathbf{a}_j(t)\mathbf{w}_j(\mathbf{x}).$$

A key difference between spectral methods and finite element methods that is relevant to this work is that spectral methods are globally smooth, i.e. the functions are C^N continuous.

5.1 The Viscous Burgers' Equation

We first consider the viscous Burgers' equation (VBE). The VBE is a one-dimensional partial differential equation which displays (somewhat) similar characteristics to the equations of gas dynamics. The VBE is given by

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}. \tag{5.1}$$

We consider solutions to the VBE using a Fourier-Galerkin spectral method. In the Fourier-Galerkin spectral method, the test and trial functions are taken to be,

$$w_k(x) = e^{\imath k x}. \tag{5.2}$$

The viscous Burgers' equation in the frequency domain (k) is obtained through the weighted residual formulation previously described and is given by,

$$\frac{\partial a_k}{\partial t} + \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p,q \in F \cup G}} a_p a_q = -\nu k^2 a_k, \qquad k \in F \cup G, \tag{5.3}$$

with $a_k(0) = a_{0k}$. The Fourier modes $\mathbf{a} = \{\tilde{\mathbf{a}}, \mathbf{a}'\}$ are contained within the union of two sets, F and G. We consider large eddy simulations of Burgers' equation in which we seek to solve for the resolved modes, $\tilde{\mathbf{a}} \in F$. The unresolved fine-scale modes are $\mathbf{a}' \in G$. Partitioning Eq. 5.3 into the resolved and unresolved sets, the evolution equation for the resolved variables is written as,

$$\frac{\partial a_k}{\partial t} + \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p \in F, q \in F}} a_p a_q = -\nu k^2 a_k - \frac{\imath k}{2} \left(\sum_{\substack{p+q=k \\ p \in G, q \in G}} a_p a_q + \sum_{\substack{p+q=k \\ p \in F, q \in G}} a_p a_q + \sum_{\substack{p+q=k \\ p \in G, q \in F}} a_p a_q \right) \qquad k \in F. \tag{5.4}$$

The last three terms on the RHS of Eq. 5.4 contain the effect of the unresolved scales on the resolved scales and must be modeled. These unclosed terms show up in the form of cross stresses (terms that have wavenumbers in both F and G) and Reynolds stresses (terms that have wavenumbers only in G). In the forthcoming sections, large eddy simulations of the VBE will be performed using various models. The VBE is solved numerically using the spectral method. The convolution summations are evaluated via the fast Fourier transform (FFT). The FFT calculations are padded by the 3/2 rule. Jameson's explicit low storage 4th order Runge-Kutta method is used for time integration [79].

5.1.1 Kinetic Energy Properties of the Fourier-Galerkin Spectral Method

To assess the energy transfer properties of the MZ-VMS models, first consider the energy budget of Burgers' equation. First, consider the inviscid case ($\nu = 0$). The spectral approximation to the viscous Burgers' equation (Eq. 5.3) obeys the physical

space PDE [3],

$$\frac{\partial}{\partial t}\tilde{u} + \frac{1}{2}\frac{\partial}{\partial x}\big(\mathbb{P}_n\tilde{u}^2\big) = 0 \qquad x \in [x, 2\pi), \tag{5.5}$$

where $\mathbb{P}_n f(x)$ is the projection of $f(x)$ onto the Fourier basis. To derive an equation for kinetic energy [3], rewrite the above as,

$$\frac{\partial}{\partial t}\tilde{u} + \frac{1}{2}\frac{\partial}{\partial x}\tilde{u}^2 = \frac{1}{2}\frac{\partial}{\partial x}\big((I - \mathbb{P}_n)\tilde{u}^2\big). \tag{5.6}$$

Next, multiply Eq. 5.6 by $\tilde{u}$ and integrate over the domain,

$$\int \frac{1}{2}\frac{\partial}{\partial t}\tilde{u}^2 + \frac{1}{6}\frac{\partial}{\partial x}\tilde{u}^3 dx = \int \frac{\tilde{u}}{2}\frac{\partial}{\partial x}\big((I - \mathbb{P}_n)\tilde{u}^2\big)dx \tag{5.7}$$

The last term on the right hand side drops due to the orthogonality of the Fourier basis. One obtains,

$$\frac{1}{2}\int \frac{\partial}{\partial t}\tilde{u}^2 = 0. \tag{5.8}$$

Equation 5.8 shows that the Fourier-Galerkin spectral method is L^2 conservative (i.e. kinetic energy preserving) for the inviscid Burgers' equation. From the analysis in Section 4.1.1, it directly follows that the t and τ-model will dissipate kinetic energy at the rate,

$$\frac{1}{2}\int \frac{\partial}{\partial t}\tilde{u}^2 \propto -\int \left(\frac{\partial}{\partial x}\big((I - \mathbb{P}_n)\tilde{u}^2\big)\right)^2 dx. \tag{5.9}$$

In the case that viscosity is present, one will note that the viscous term shows up in Fourier space as

$$\mathcal{F}u_{xx} = -k^2 a_k.$$

The viscous term for the resolved modes is only a function of the resolved modes, and same for the unresolved modes. This falls into the special case discussed in Section 4.1.2 and it follows that the τ and t-model will again dissipate energy at the rate given in Eq. 5.9.

5.1.2 Analytic evaluation of $\mathcal{PLQL}a_{0j}$ for Burgers' Equation

The analytic form of $\mathcal{PLQL}a_{0j}$ can be computed by evaluating the exact Fréchet derivative of $\mathcal{QL}a_{0j}$ and then projecting it. This corresponds to a linearization. The right hand side of Burgers' equation at $t = 0$ in Fourier space is

$$\mathcal{L}a_{0k} = R_k(\mathbf{a}_0) = -\frac{\imath k}{2} \sum_{\substack{p+q=k \\ p,q\in F\cup G}} a_{p0}a_{q0} - \nu k^2 a_{k0}, \qquad k \in F \cup G. \tag{5.10}$$

Split the convolution in Eq. 5.10 into resolved and unresolved terms,

$$\begin{aligned}\mathcal{L}a_{0k} = &-\frac{\imath k}{2} \sum_{\substack{p+q=k \\ p,q\in F}} a_{p0}a_{q0} - \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in F,q\in G}} a_{p0}a_{q0} \\ &- \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in G,q\in F}} a_{p0}a_{q0} - \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p,q\in G}} a_{p0}a_{q0} - \nu k^2 a_{k0}, \qquad k \in F \cup G,\end{aligned} \tag{5.11}$$

where modes in F are resolved and in G are unresolved. Application of the complimentary projector eliminates the first term on the RHS,

$$\mathcal{QL}a_{0k} = -\frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in F,q\in G}} a_{p0}a_{q0} - \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in G,q\in F}} a_{p0}a_{q0} - \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p,q\in G}} a_{p0}a_{q0} - \underbrace{\nu k^2 a_{k0}}_{k\in G}, \qquad k \in F\cup G. \tag{5.12}$$

Now linearize Eq. 5.12 about $\mathbf{a}_0$,

$$\begin{aligned}\mathcal{QL}^{lin}a_{0k} = &-\frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in F,q\in G}} \left(a_{p0}a'_{q0} + a'_{p0}a_{q0}\right) - \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in G,q\in F}} \left(a_{p0}a'_{q0} + a'_{p0}a_{q0}\right) - \\ &\frac{\imath k}{2} \sum_{\substack{p+q=k \\ p,q\in G}} \left(a_{p0}a'_{q0} + a'_{p0}a_{q0}\right) - \underbrace{\nu k^2 a'_{k0}}_{k\in G}, \qquad k \in F \cup G.\end{aligned} \tag{5.13}$$

Application of the Liouville operator to Eq. 5.12 is simply Eq. 5.13 evaluated at $\mathbf{a_0}' = R(\mathbf{a}_0)$,

$$\mathcal{LQL}a_{0k} = -\frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in F, q\in G}} \big(a_{p0}R_q(\mathbf{a_0}) + R_p(\mathbf{a_0})a_{q0}\big) - \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in G, q\in F}} \big(a_{p0}R_q(\mathbf{a_0}) + R_p(\mathbf{a_0})a_{q0}\big) - \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p,q\in G}} \big(a_{p0}R_q(\mathbf{a_0}) + R_p(\mathbf{a_0})a_{q0}\big) - \underbrace{\nu k^2 R_{k0}(\mathbf{a_0})}_{k\in G}, \qquad k \in F \cup G. \tag{5.14}$$

Finally, project Eq. 5.14,

$$\mathcal{PLQL}a_{0k} = -\frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in F, q\in G}} a_{p0}R_q(\hat{\mathbf{a}}_\mathbf{0}) - \frac{\imath k}{2} \sum_{\substack{p+q=k \\ p\in G, q\in F}} R_p(\hat{\mathbf{a}}_\mathbf{0})a_{q0} - \underbrace{\nu k^2 R_{k0}(\hat{\mathbf{a}}_\mathbf{0})}_{k\in G}, \qquad k \in F \cup G. \tag{5.15}$$

Noting that the two separate terms on the RHS are identical and inserting in R_q, one obtains,

$$\mathcal{PLQL}a_{0k} = -\imath k \sum_{\substack{p+q=k \\ p\in F, q\in G}} a_{p0}\left[-\frac{\imath q}{2} \sum_{\substack{r+s=q \\ r,s\in F}} a_{r0}a_{s0}\right] - \underbrace{\nu k^2 R_{k0}(\hat{\mathbf{a}}_\mathbf{0})}_{k\in G}, \qquad k \in F \cup G. \tag{5.16}$$

The reader will note that the model form for the viscous Burgers' equation is the same for modes in F.

Remarks

- Due to the linearization present in the derivation of $\mathcal{PLQL}a_{0k}$, contributions from the Reynolds stress term are not present. MZ-VMS models that only include the term $\mathcal{PLQL}a_{0k}$ are incapable of modeling the Reynolds stresses.
- This limitation of the first order MZ-VMS models will manifest itself in the following sections when we apply the MZ-VMS models to turbulent flows

at a high Reynolds number.

5.1.3 Case 1: Standing Shockwave

The first case that will be considered uses the initial condition $u(x,0) = \sin(x)$ with a viscosity of $\nu = 0.005$. This leads to a standing shock after $t = 1$. The low viscosity allows for the solution to be mostly inviscid, but stops the PDE from becoming singular. The reduced model is taken to be of size $N = 16$ (corresponding to a cutoff frequency at $k = 8$), while the full order model takes $N = 1024$. The closure model must be capable of removing the energy transferred to the high frequency modes resulting from the shock. The simulation is evolved until $t = 15$.

Structure of the Memory and Shortcomings of the t-model

We first investigate the structure of the memory and demonstrate how the well-studied t-model will break down for this problem. As discussed in Chapter IV, the t-model [20] can be derived under the guise of various mathematical assumptions. For example, the t-model can be derived by expanding the memory integral in a Taylor series expansion about $s = 0$, or by approximating the orthogonal dynamics evolution operator ($e^{t\mathcal{QL}}$) with the evolution operator of the full dynamics ($e^{t\mathcal{L}}$). The most straightforward and physically intuitive derivation is to use a left hand side quadrature rule to approximate the memory integral using its value at $s = 0$,

$$\mathcal{P}\int_0^t e^{(t-s)\mathcal{L}}\mathcal{PL}e^{s\mathcal{QL}}\mathcal{QL}a_{0j}ds \approx te^{t\mathcal{L}}\mathcal{PLQL}a_{0j}. \tag{5.17}$$

With this approximation, one does not need to solve the troublesome orthogonal dynamics equation as the dependence on $e^{t\mathcal{QL}}$ is eliminated. The t-model has been applied with varying degrees of success to a number of problems, including Burgers' equation and the incompressible Navier-Stokes equations. Work by Stinis [94] has

argued that the t-model requires an additional scaling constant to maintain accuracy,

$$\mathcal{P}\int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}a_{0j}ds \approx C_t t e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}a_{0j}. \tag{5.18}$$

Stinis motivates the need for these renormalization constants with the hypothesis that the full-order model that the t-model is approximating is, itself, under-resolved.

Assessing the validity of the underlying assumptions of the t-model (as well as a renormalized t-model) is challenging as it requires one to directly evaluate the memory kernel. This involves solving the N-dimensional orthogonal dynamics equation, which is thus far intractable. Here, we use the orthogonal ODE and its procedure [43] presented in Section 3.3.3 to *approximate* solutions to the orthogonal dynamics in an attempt to directly assess the underlying assumptions of the t-model and Stinis' renormalized models. The purpose of the following discussion is to use the orthogonal ODE to approximate the memory kernel so as to understand how the fine-scale unresolved modes impact the coarse-scale resolved modes. Numerical details for the simulation considered are given in Table 5.1. The reconstruction of the memory kernel through the orthogonal ODE and its procedure is performed as follows [43]:

1. Run the full order simulation to obtain $\mathbf{a}(t_j)$, where j are the number of desired quadrature points for the eventual discrete reconstruction of the memory integral.
2. From the full order simulation data, compute $K(\mathbf{a}(t_j), 0) = e^{t_j\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$.
3. For each t_j, evolve the orthogonal ODE using the initial condition $K(\mathbf{a}(t_j), 0)$ to obtain $K(\tilde{\mathbf{a}}(t_j), s_i)$.
4. For each t_j discretely reconstruct the memory integrand, $\mathcal{M}(t) = \int_0^t K(\tilde{\mathbf{a}}(t-s), s)ds$, using any numerical quadrature rule.

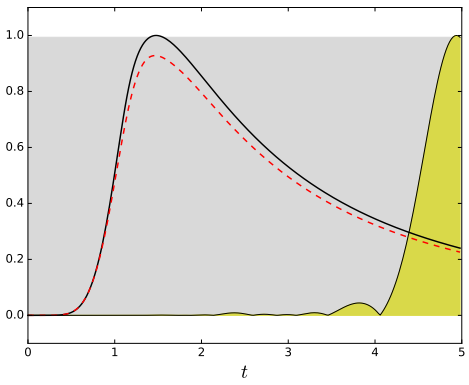
Figure 5.1 shows the reconstructed memory integrand at $t = 5$, as well as the total

L	ν	N	Δt	Quadrature Δt
2π	0.005	1028	0.001	0.005

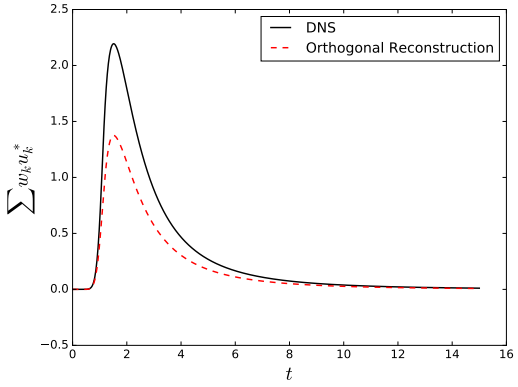
Table 5.1: Physical and numerical details for reconstruction of the kernel via the orthogonal ODE. N is the size of the full-order model.

subgrid energy transfer. The reconstructed terms are compared to those extracted from DNS data. The integrand assumed by the t-model is also shown for reference. The subgrid term given by MZ is the area of the shaded regions. The yellow shaded region is that predicted by the orthogonal ODE procedure, while the gray shaded region is that assumed by the t-model. It is seen that the comparison of the memory term approximated through the orthogonal ODE to the DNS is reasonable, although some error is notably present for the total subgrid energy transfer. Nonetheless, the important observation to make from Figure 5.1 is the presence of a decaying memory kernel. It is quite clear that the t-model will grossly overestimate the value of the integrand and that an additional scaling is required. The required scaling is the ratio of the true integral (area of the shaded yellow region) to the assumed t-model integral (area of the shaded gray region). Figure 5.2 shows these ratios computed for various reduced order model sizes (i.e. varying the cutoff wavenumber k_c) at $t = 1$. The size of the reduced order models are denoted by N_c. The ratios are compared to the renormalized t-model coefficients as computed by Stinis. The agreement is excellent.

This numerical evidence shows that, as suggested by Stinis, the t-model requires an additional scaling coefficient. The simple left hand side quadrature rule assumed by the t-model is inappropriate and unjustified. The decaying kernel must be accounted for to accurately model the memory.

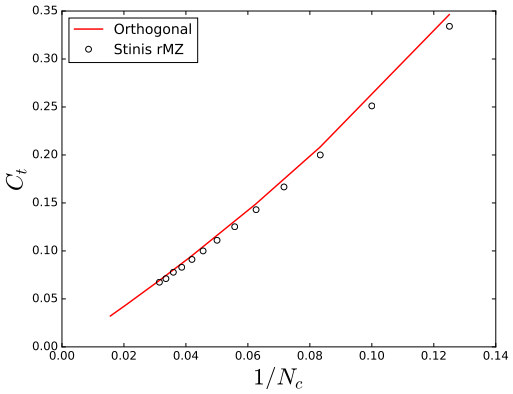


(a) Evaluation of the approximate (normalized) memory integrand for $k = 7$ at $t = 5.0$.

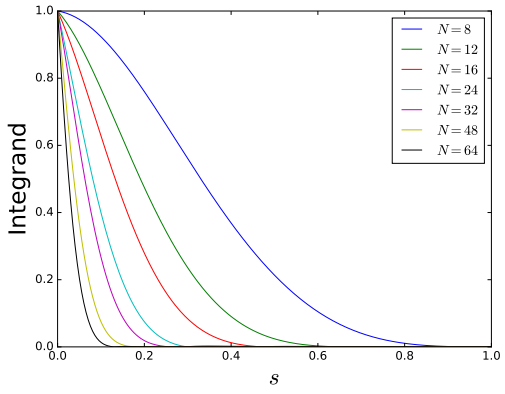


(b) The total subgrid energy transfer.

Figure 5.1: Reconstruction of the memory kernel via the orthogonal ODE.



(a) Ratio of the orthogonal dynamics integral to the assumed t-model integral for various N_c.



(b) Normalized integrands for various N_c.

Figure 5.2: Comparison of the ratio of the true memory integral to the t-model memory integral (i.e. the ratio of the area in the shaded yellow region in Fig. 5.1a to the area in the shaded gray region) versus the renormalization coefficients obtained by Stinis.

Predictive Results

Predictive results using the dynamic-τ model, the t-model, and VMS(ϵ) model are now examined. Per the previous discussion, we expect the τ and VMS(ϵ) models to outperform the t-model as they can account for a decaying kernel (if $\tau < t$). Note that all three of these models have the same form, but differ in the definition of the timescale. In the VMS(ϵ) model, the timescale is defined as [100],

$$\tau = \left[\frac{4}{h^2}u^2 + 3\pi\nu^2\left(\frac{4}{h^2}\right)^2\right]^{-1/2}. \tag{5.19}$$

The numerical results for total resolved kinetic energy and the mean magnitude of the subgrid-content are shown in Figure 5.3. In Figure 5.3 it is seen that all of the models accurately predict the decay of kinetic energy and the subgrid-content. Simulations run with no subgrid-scale model lead to inaccurate results. As expected, the t-model performs worse than the other models. The model slightly under-predicts the decay of kinetic energy and over predicts the subgrid-content. Figure 5.4 compares the memory timescale τ of the different models and compares it to the ratio of the memory at $s = 0$ over the total memory integral. The timescales of the dynamic-τ model and VMS(ϵ) model are seen to evolve similarly and compare well with the DNS data (i.e. the ratio of the true memory term to its value at $s = 0$). Figure 5.5 shows the $x-t$ diagrams of the numerical solutions for select cases in physical space. The trajectories of the simulation using the dynamic-τ model are seen to give the best comparison with the DNS data.

As previously discussed, both the t-model, dynamic-τ-model (and the VMS(ϵ) model) do not include contributions from the Reynolds stress. To demonstrate this, the subgrid-content at $t = 2.0$ is examined. Figure 5.6 shows the decomposition of the subgrid energy transfer into the cross stresses and the Reynolds stresses.

The stresses are compared to the total subgrid-stress as predicted by the dynamic-τ model. In Figure 5.6 it is seen that the cross stress is the dominant component of the subgrid-content for high wavenumbers. The dynamic-τ model is seen to offer an excellent qualitative depiction of the cross stress. The magnitude of the cross stresses are predicted reasonably well and, more importantly, its wavenumber dependence is well-characterized. The same, however, can not be said about the Reynolds stress. There is no qualitative resemblance between the subgrid-content as predicted by the MZ-VMS model and the DNS data. This is not surprising as the Reynolds stress makes no appearance in the MZ-VMS model.

5.1.4 Case 2: Burgers' Turbulence

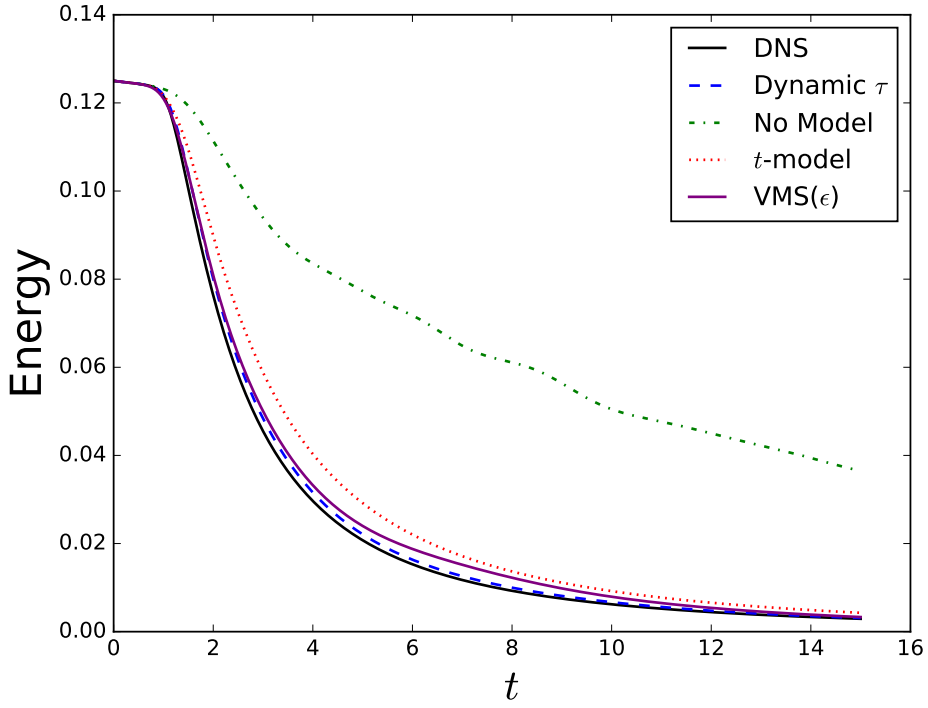
Numerical simulations of the VBE are now performed with the initial condition [60],

$$u(x) = U_0^* \sum_{i=1}^{k_c} \sqrt{2E(k_i)} \sin(k_i x + \beta_i), \tag{5.20}$$

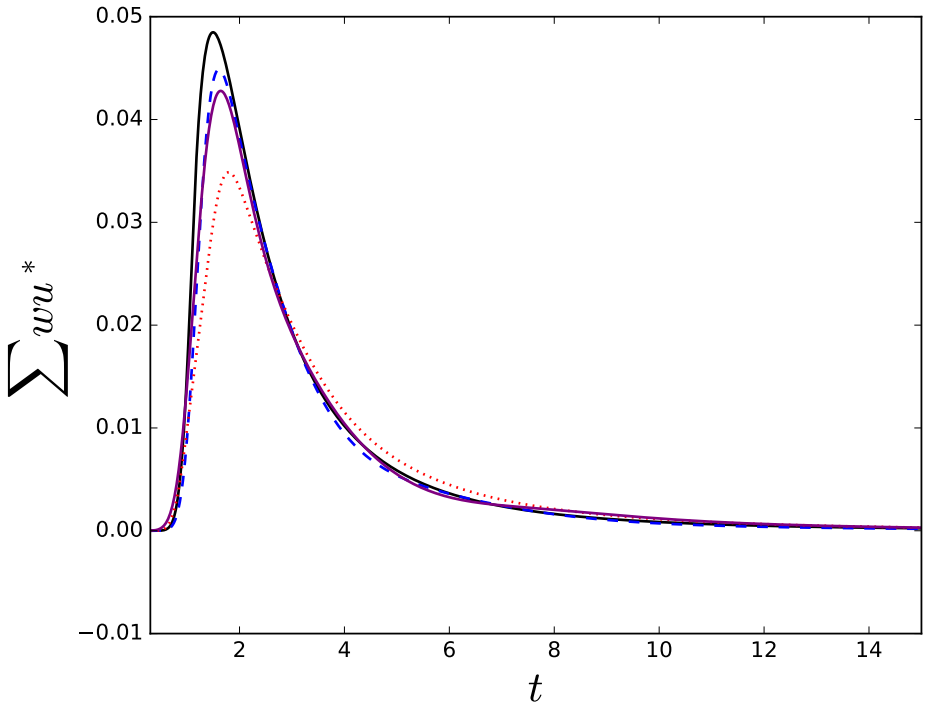
where $E(k) = 5^{-5/3}$ if $1 \leq k \leq 5$ and $E(k) = k^{-5/3}$ for $k > 5$. Eq. 5.20 initializes an energy spectrum with a -5/3 slope for $k > 5$. The phase angle β is a random number in the domain $[-\pi, \pi]$. A constant seed value is used in all of the simulations. No energy is added to the flow after a cutoff frequency k_c, such that all simulations are initially fully resolved. We consider simulations with the t-model, dynamic-τ model, VMS(ϵ) model, and Stinis' finite memory models. All of Stinis' higher order models use one quadrature point.

Selection of the Memory Length

While the dynamic-τ and t-model are autonomous, the finite memory models require a specification of the memory length. In Section 3.4 it was demonstrated that the memory length could be related to the eigenvalues of the fine-scale operator.



(a) Total kinetic energy in the resolved modes.



(b) Mean magnitude of the subgrid-content.

Figure 5.3: Evolution of integral quantities for Burgers' equation.

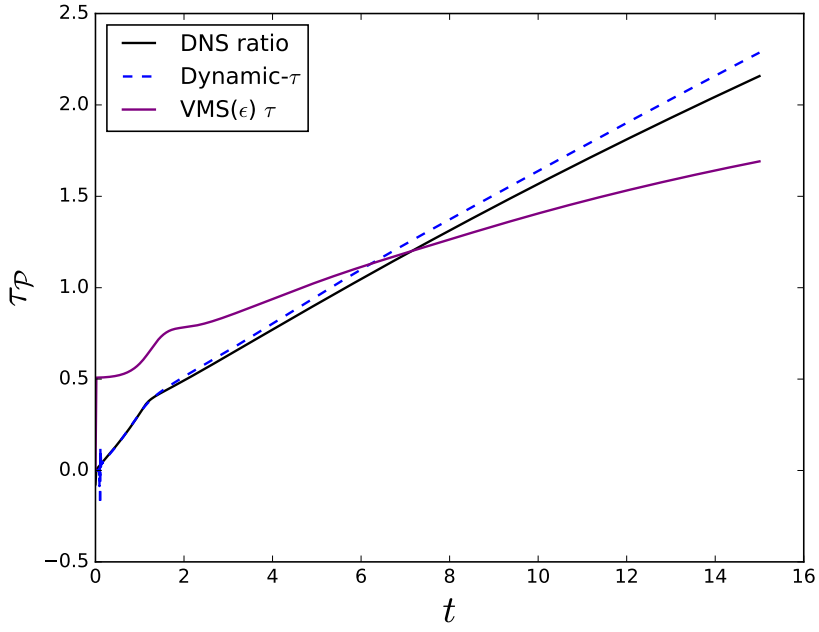
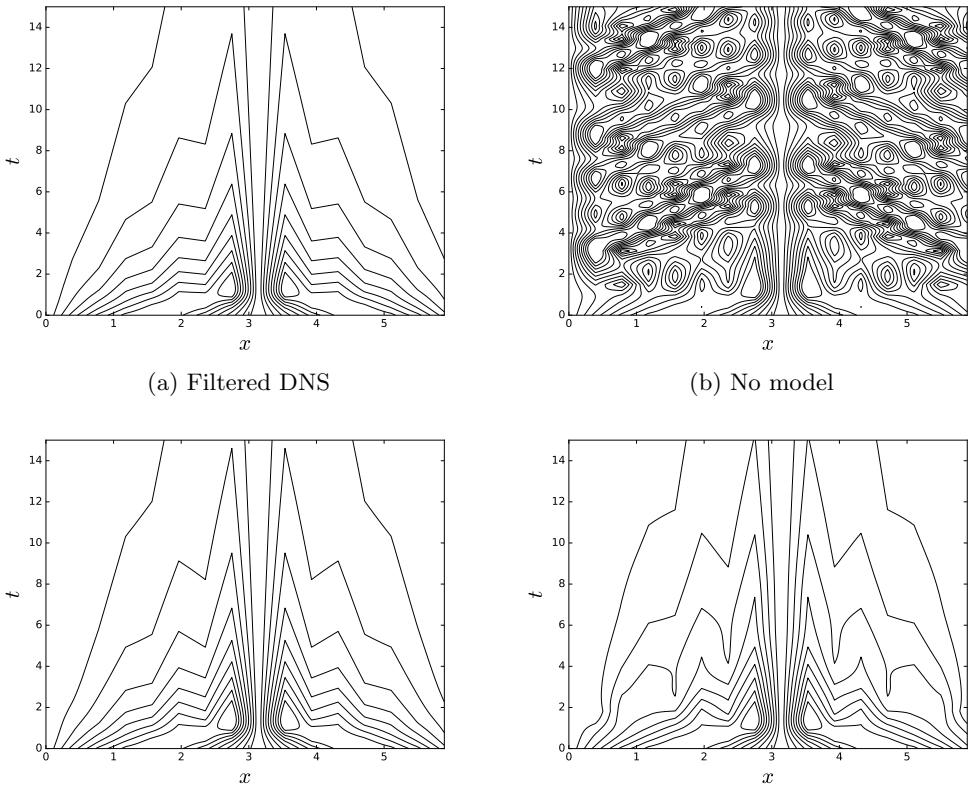


Figure 5.4: Evolution of the memory timescale, τ, for different models. Note that for the t-model $\tau = t$.

Since Burgers' equation does not exhibit scale separation, a logical hypothesis is that a mean timescale can be related to the spectral radius of the Jacobian of the resolved variables

$$\tau \propto 1/\rho\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right).$$

To provide evidence for this argument, a parametric study was performed involving 60 cases. The simulations were initialized with Eq. 5.20 and operated over a range of Reynolds numbers and resolutions. The cases considered were permutations of the following parameters: $\nu = [0.05, 0.01, 0.005, 0.001, 0.0005], k_c = [8, 16, 32], U_0^* = [1, 2, 5, 10]$. The DNS simulations were carried out using 4096 resolved modes. For each case, the time constant τ_0 in the first order finite memory model is found by solving an inverse problem. For simplicity, τ_0 is taken to be constant in time. The optimal time constant in the least squares sense was found by minimizing the differ-



(a) Filtered DNS

(b) No model

(c) Dynamic-τ model

(d) VMS(ϵ) model

Figure 5.5: $x - t$ diagrams for the filtered velocity field in physical space.

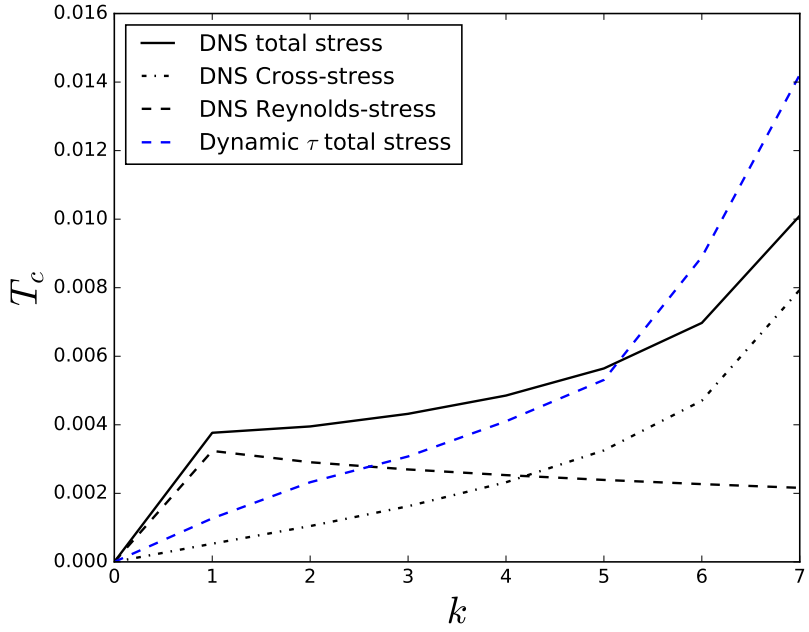


Figure 5.6: Decomposition of the subgrid-content into cross stresses and Reynolds stresses.

ence of the total kinetic energy dissipation rate between the large eddy simulation and a high resolution direct numerical simulation. The solution was minimized for $t \in [0, 2]$ using data at discrete time-steps spaced by intervals of $\Delta t = 0.01$. The discrete penalty function is given by,

$$\mathcal{J} = \sum_{n=1}^{N} \left(\left[\frac{dK_n}{dt} \right]_{MZ} - \left[\frac{dK_n}{dt} \right]_{DNS} \right)^2$$

where $N = 2/0.01 = 200$. The penalty function was minimized using SciPy's optimization suite. A downhill simplex algorithm was used. It is noted that the inferred results were similar to a penalty function that minimized the total kinetic energy. Figure 5.7 shows the inferred time constants plotted against the temporal mean of the spectral radius of the Jacobian of the resolved variables. The $k_c = 16$ and $k_c = 32$ cases are seen to collapse to the same line. The $k_c = 8$ cases also collapse to a linear line, but with a slightly greater slope. Given the wide range of cases considered, the collapse of the data is good. This result suggests that the properties of the Jacobian

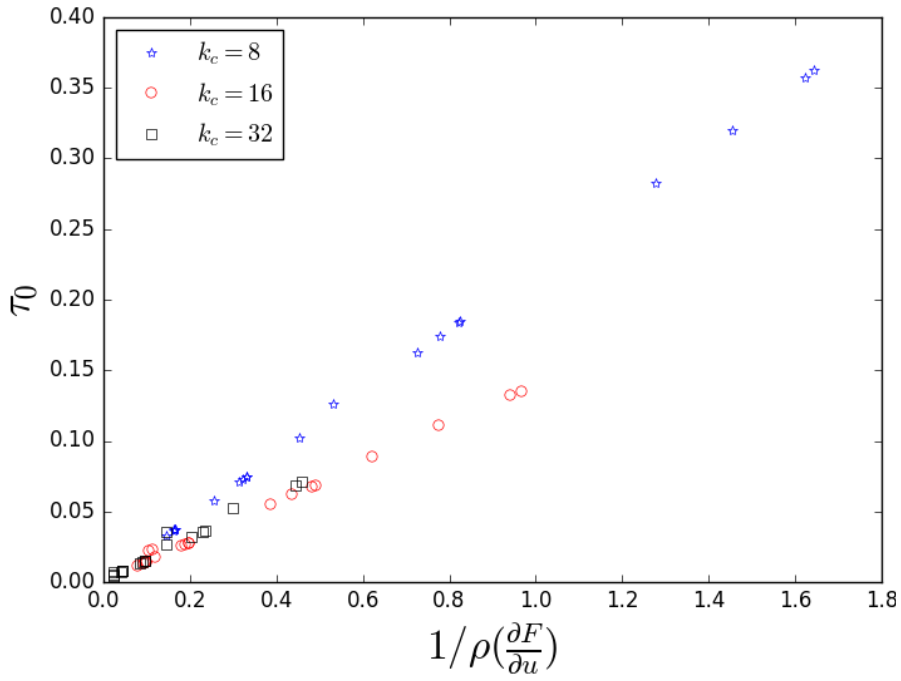


Figure 5.7: Inverse solution for the first order model time constant τ_0 plotted against the inverse of the spectral radius of the Jacobian of the resolved variables.

of the resolved variables can be used as a good indicator of the memory length. A fit of the above data yields $\tau_0 \approx 0.2/\rho\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}_0}\right)$. In the following section, this heuristic will be used to select the memory length. For simplicity, a constant value of τ_0 based on the spectral radius at $t = 0$ is used.

Predictive Results

Predictive simulations are now considered. Direct numerical simulations of the VBE were performed with 2048 resolved modes ($-1024 \leq k \leq 1023$) on a spatially periodic domain of length 2π for $t \in [0, 2]$. The initial condition given in Eq. 5.20 is used with $U_0^* = 1$ and $\nu = 0.01$. Large eddy simulations are performed with the closure models described in Chapter IV. The first, second, and third order finite memory models are considered. The large eddy simulations are initialized with the DNS solution for $k \leq k_c$ such that all simulations are fully resolved at $t = 0$.

	DNS	Smagorinsky	t-model	FM1	FM2	FM3
N	2048	32	32	32	32	32
Δt	1e-4	1e-3	1e-3	1e-3	1e-3	1e-3
Constants	NA	$Cs = 0.2$	$\tau_0 = t$	$\tau_0 = 0.135$	$\tau_{0,1} = 0.135, 0.07$	$\tau_{0,1,2} = 0.135, 0.07, 0.07$

Table 5.2: Summary of computational details for the numerical experiments of Burgers' equation.

The simulations were performed with 32 resolved modes, corresponding to a cutoff frequency at $k_c = 16$. The memory length τ_0 was selected by the procedure described in the previous section. A formal estimation procedure was not used for the memory lengths of Stinis' higher order models, which were simply chosen to be $\tau_1 = \tau_2 = 0.5\tau_0$. A summary of the relevant computational details is given in Table 5.2. Figures 5.8 and 5.9 compare the MZ-VMS based models to the filtered DNS data, the Smagorinsky model, and a simulation run on a 32 point mesh without any closure model. Figure 5.8a shows the temporal evolution of the total kinetic energy and rate of kinetic energy decay. Figure 5.8b shows the temporal evolution of the mean magnitude of $w^{(0)}$ (note $w^{(0)} = ik\tau^{sgs}$) and the energy spectrum at $t = 2.0$. Figure 5.9 shows the trajectories of the 8th (Fig. 5.9a) and 15th (Fig. 5.9b) modes of $\mathbf{a}$ and the memory term (denoted by $w^{(0)}$) in the complex plane. A brief discussion of the results of each simulation will now be provided.

The simulation performed without a closure model under-predicts the rate of energy decay at early time, leading to an over-prediction in total kinetic energy. As expected, the simulation under-predicts the dissipation rate. With no subgrid-mechanism present to remove energy from high wavenumbers, a pile-up of energy is seen for high k. This is evidenced in Figure 5.8b. This phenomena indicates that the simulation is under-resolved and a closure model is indeed required. The evolution of the individual modes of $\tilde{\mathbf{a}}$ contain significant error, particularly around the cutoff frequency. The evolution of the 15th mode, as shown in Figure 5.9a, is an excellent

example of the error present in high wavenumbers.

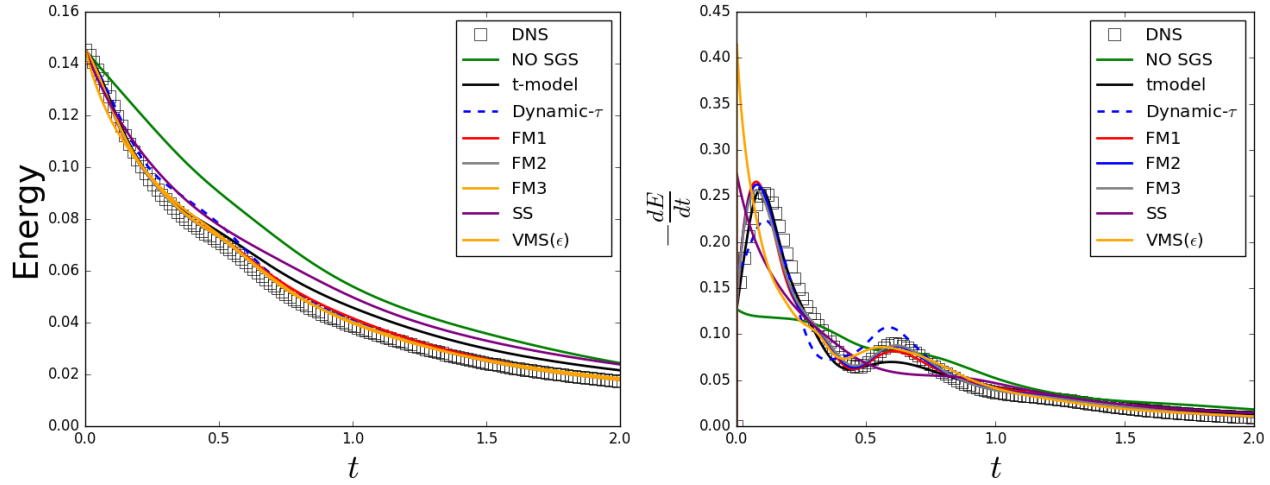
The Smagorinsky model offers improvements. The simulation utilizing the basic closure model provides decent predictions for both the total kinetic energy and the dissipation of kinetic energy. The energy spectrum at $t = 2.0$ and trajectories of the individual modes are additionally much improved. However, the Smagorinsky model is unable to differentiate the resolved scales from the unresolved scales. Despite being completely resolved at $t = 0$, the Smagorinsky model predicts that the subgrid-content is maximum at $t = 0$. The resulting predictions for $w^{(0)}$ are both quantitatively and qualitatively incorrect. In particular, the individual trajectories of $w^{(0)}$ show no similarity to that of the DNS. It is recognized that the Smagorinsky model was developed for homogeneous turbulent flows in the physical domain, so a critical evaluation of the model on Burgers' equation is not completely appropriate.

Simulations performed using the $t-$model provide improved predictions. The largest error is present in the prediction for $w^{(0)}$, where it is seen that the $t-$model slightly over predicts subgrid-content (especially for $t > 0.5$). The predictions, however, are still qualitatively correct. The model correctly predicts the initial peak in dE/dt and $w^{(0)}$ around $t = 0.15$ and qualitatively shows the presence of the second peak around $t = 0.6$. The trajectories of the individual modes in u and $w^{(0)}$ are improved, but become less accurate for late time. The prediction for the energy spectrum at $t = 2.0$ is not noticeably better than that of the Smagorinsky model. The explicit presence of t in the model leads to substantial error at large times. The performance of the t-model for the VBE shows the merit in the MZ-VMS-based models. To reiterate, the t-model contains no heuristics or coefficients. Work by Stinis [95] and the authors own numerical experiments show that renormalization of the t-model can lead to more accurate results.

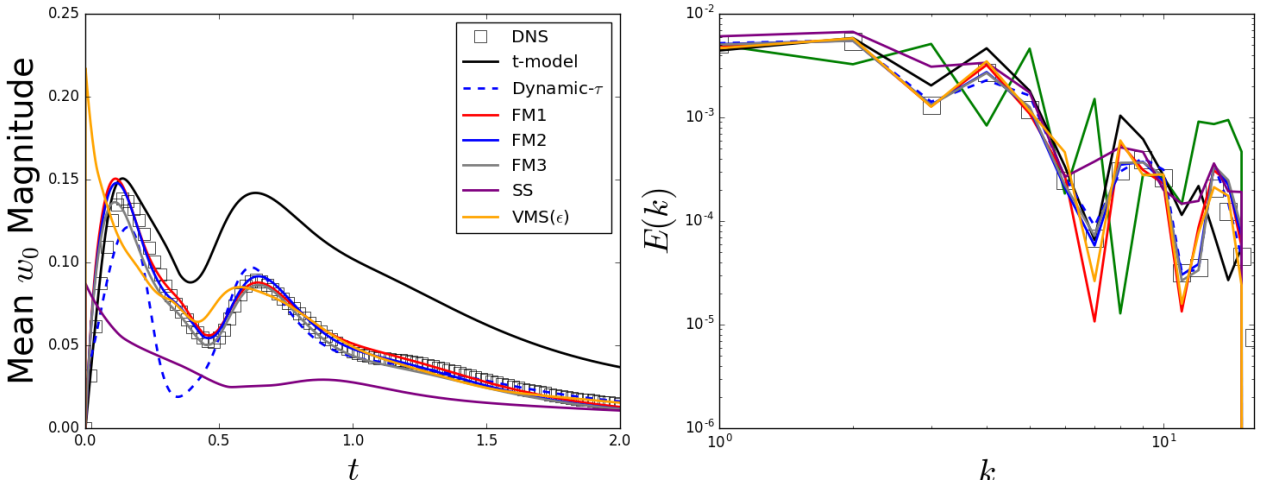
The dynamic-τ model does not perform as well as the t-model for early times, but outperforms it later in the simulation. It outperforms the Smagorinsky model as well as the no-model simulation. Due to the complex initial conditions and lack of scale similarity at initial time, the dynamic procedure to determine τ slightly suffers. Further, the initial conditions are such that the closure contribution is initially zero at $t = 0$. The dynamic-τ model is unable to predict this as it assumes scale similarity. As time increases and an energy cascade is established, the assumptions of the dynamic model improve and the model is able to accurately capture the impact of the unresolved scales. The VMS(ϵ) model performs qualitatively similar to the dynamic-τ model, although there is slightly more error in the phase space trajectories.

The finite memory models provide relatively accurate predictions for all quantities. The evolution of total kinetic energy, dissipation of kinetic energy, and mean subgrid-predictions are in good agreement with the DNS. The first order finite memory model accurately predicts the instantaneous energy spectrum at $t = 2.0$ for low wavenumbers, while the second and third order models provide accurate predictions for all wavenumbers. The trajectories of the individual modes are close to that of the DNS, as are the trajectories for the subgrid-terms. It does need to be noted, however, that the finite memory models were optimized via the selection of the memory length, while the t-model and dynamic-τ model have no such tuning parameters.

Results of the first order finite memory model, t-model, and dynamic-τ model are shown in Figure 5.10 for two additional cases. The first is run at a resolution of $k_c = 8$, a viscosity of $\nu = 0.01$, and a scaling of $U_0^* = 5$. The second case is run at a resolution of $k_c = 32$, a viscosity of $\nu = 5 \times 10^{-4}$, and a scaling of $U_0^* = 10$. The time constants for the finite memory model were again selected by the scaling of the

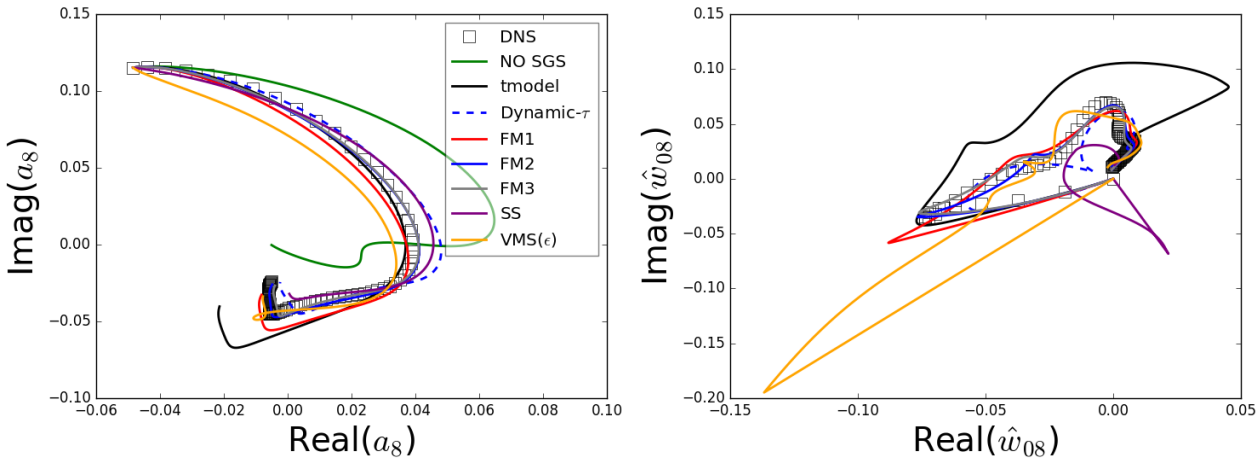


(a) Temporal evolution of total kinetic energy (left) and rate of decay of kinetic energy (right).

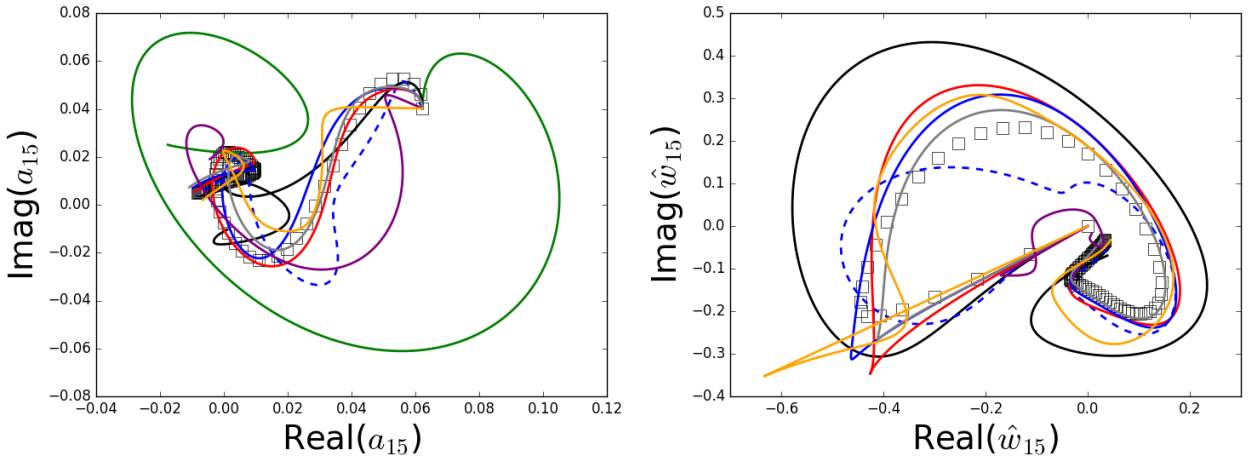


(b) Temporal evolution of the mean magnitude of $w^{(0)}$ (left) and energy spectrum at $t = 2.0$. Note that $w^{(0)} = ik\tau_{sgs}$.

Figure 5.8: A comparison of Large Eddy Simulations performed with 32 resolved modes to filtered DNS data obtained from a simulation performed with 2048 resolved modes.



(a) Evolution of the eighth mode of u (left) and w (right).



(b) Evolution of the 15th mode of u (left) and w (right).

Figure 5.9: Evolution of select modes of u and w in phase space. In phase space the DNS data (denoted by $\square$) is sparse around $t = 0$ and becomes clustered as $t \to 2$.

spectral radius of the Jacobian. Both of these cases are significantly under-resolved. The low viscosity case in particular has numerous shocks and required 4096 resolved modes for the DNS calculation.

5.2 Navier-Stokes Equations

Large eddy simulations of the incompressible Navier-Stokes equations using a Fourier-Galerkin spectral method are now considered. The incompressible Navier-Stokes equations are given by,

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{5.21}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial u_i}{\partial x_j}\right). \tag{5.22}$$

For triply periodic problems, Eqns. 5.21 and 5.22 can be Fourier-transformed in all directions. The pressure Poisson equation can be directly solved in Fourier space, allowing the governing equations to be written compactly as,

$$\left(\frac{\partial}{\partial t} + \nu k^2\right) a_i(\mathbf{k},t) + \left(\delta_{im} - \frac{k_i k_m}{k^2}\right) \imath k_j \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p},\mathbf{q}\in F\cup G}} a_j(\mathbf{p},t) a_m(\mathbf{q},t) = 0 \qquad \mathbf{k} \in F \cup G, \tag{5.23}$$

where the Fourier modes have been written to belong to the union of two sets (with F being the resolved set and G being the unresolved set). Separating the modes into resolved and unresolved sets yields the reduced system,

$$\left(\frac{\partial}{\partial t} + \nu k^2\right) a_i(\mathbf{k},t) + \left(\delta_{im} - \frac{k_i k_m}{k^2}\right) \imath k_j \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p},\mathbf{q}\in F}} a_j(\mathbf{p},t) a_m(\mathbf{q},t) = - \left(\delta_{im} - \frac{k_i k_m}{k^2}\right) \imath k_j \hat{\tau}_{jm}(\mathbf{k},t) \qquad k \in F, \tag{5.24}$$

where

$$\hat{\tau}_{jm}(\mathbf{k},t) = \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p},\mathbf{q}\in G}} a_j(\mathbf{p},t) a_m(\mathbf{q},t) + \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p}\in G,\mathbf{q}\in F}} a_j(\mathbf{p},t) a_m(\mathbf{q},t) + \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p}\in F,\mathbf{q}\in G}} a_j(\mathbf{p},t) a_m(\mathbf{q},t).$$

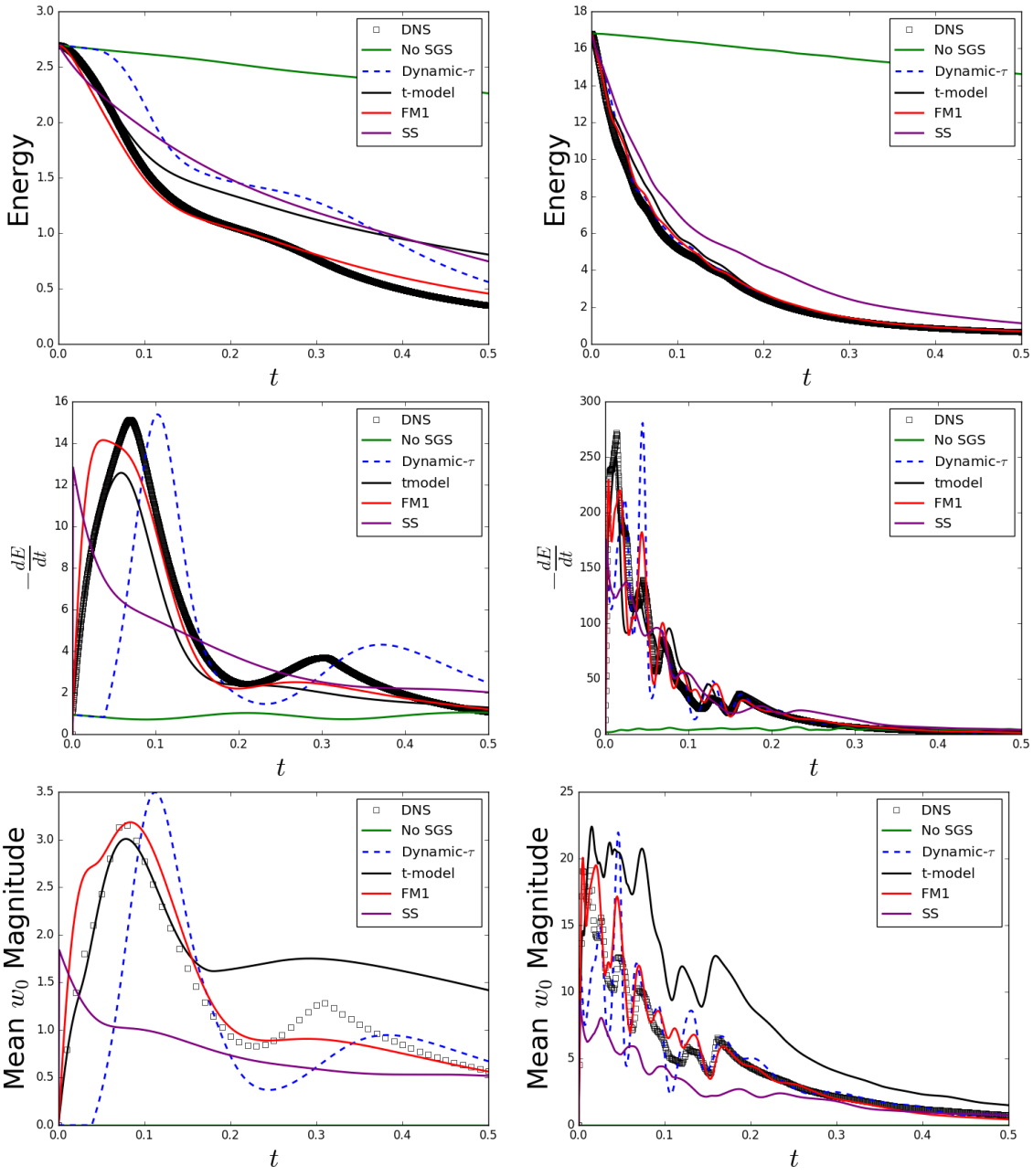


Figure 5.10: Evolution of select quantities of the additional simulations of the VBE. The conditions are $k_c = 8, \nu = 0.01, U_0^* = 5$ (left) and $k_c = 32, \nu = 5 \times 10^{-4}, U_0^* = 10$.

Note that, in Fourier space, the pressure term appears as a projection. This projection leads to additional non-linear interactions between the resolved and unresolved scales. As in Burgers' equation, the unclosed terms can be separated into the cross stresses and the Reynolds stresses. As previously seen, the first-order MZ-VMS models are incapable of modeling the Reynolds stresses. Through the use of the Mori-Zwanzig formalism, the RHS of Eq. 5.24 can be alternatively written as a convolution integral,

$$\left(\frac{\partial}{\partial t}+\nu k^2\right)a_i(\mathbf{k},t)+\left(\delta_{im}-\frac{k_i k_m}{k^2}\right)ik_j\sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k}\\ \mathbf{p},\mathbf{q}\in F}}a_j(\mathbf{p},t)a_m(\mathbf{q},t)= \mathcal{P}\int_0^t K_i(\tilde{\mathbf{a}}(t-s),s)ds \qquad k\in F. \tag{5.25}$$

5.2.1 Kinetic Energy Properties of the Fourier-Galerkin Spectral Method for the Incompressible Navier-Stokes Equations

The Fourier-Galerkin spectral approximation to the Euler equations can be shown to be kinetic energy conserving. The proof follows that which was previously performed for Burgers' equation, and we refer to the work of Bardos and Tadmor [3]. As in Burgers' equation, in the Navier-Stokes equations the impact of viscosity again shows up as a self interaction and hence we can still guarantee that the t-model and τ-model will dissipate energy. The guarantee of kinetic energy stability for the Euler equations demonstrates that the τ, t, and VMS(ϵ) models will be globally dissipative and stable for all Reynolds numbers and all resolutions.

5.2.2 MZ-VMS Model Forms

The MZ-VMS-based models described in Chapter 4 require computing terms of the form $\mathcal{PL}(\mathcal{QL})^n a_{0j}$. The analytic derivation of these terms is somewhat tedious,

but they do follow a general pattern. After much algebra it can be shown

$$\mathcal{PLQLA}_i(\mathbf{k},0) = \left(-\delta_{im} + \frac{k_i k_m}{k^2} \right) ik_j \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p}\in F, \mathbf{q}\in G}} a_j(\mathbf{p},0)\mathcal{PLA}_m(\mathbf{q},0) - \left(\delta_{im} + \frac{k_i k_m}{k^2} \right) ik_j \sum_{\substack{\mathbf{p}+\mathbf{q}=\mathbf{k} \\ \mathbf{p}\in F, \mathbf{q}\in G}} a_m(\mathbf{p},0)\mathcal{PLA}_j(\mathbf{q},0). \tag{5.26}$$

This is consistent with the results obtained by Stinis [92]. We note that viscosity does not change the above term for the modes in F. It will, however, change the result for the modes in G and must be considered for higher order models. We refer to Stinis [92] as a reference for the derivations for higher order models for the Euler equations, which are tedious and challenging but follow certain patterns. The addition of the viscous term adds little difficulty to the derivations. Symbolic MATLAB scripts are available to aid in the model derivations/validation.

5.2.3 VMS(ϵ) model

The VMS(ϵ) model has the same structural form as the t and τ-models but has a different definition of τ. The timescale is taken to be [100],

$$\tau = \left(\frac{4}{h^2}\mathbf{u}\cdot\mathbf{u} + 3\pi\nu^2 \left(\frac{4}{h^2}\right)^2 \right)^{-1/2}. \tag{5.27}$$

5.2.4 Numerical Implementation

A code named PySpectral was developed in this dissertation to assess the performance of the MZ-VMS models. PySpectral is a Python-based code that solves the three-dimensional incompressible Navier-Stokes equations with triply periodic boundary conditions in their primitive form. In addition to the standard Navier-Stokes equations, the code has been used to solve the Navier-Stokes equations with rotation as well as the equations of ideal magnetohydrodynamics. The code is parallelized through mpi4py and has been used to solve systems with over one billion

unknowns. The code implementation uses ideas developed in [70]. Details on the solver are presented in Appendix B

5.2.5 Taylor Green Vortex

The Taylor Green vortex [98] is a common model problem to study the accuracy of computational methods and subgrid models. The flow is characterized by a break down of initial coherent structures into turbulent flow. The flow is initialized on a periodic domain with $x_1, x_2, x_3 \in [-\pi, \pi]$ and,

$$\begin{aligned} u_1(x_1, x_2, x_3) &= \cos(x_1)\sin(x_2)\cos(x_3) \\ u_2(x_1, x_2, x_3) &= -\sin(x_1)\cos(x_2)\cos(x_3) \\ u_3(x_1, x_2, x_3) &= 0. \end{aligned} \tag{5.28}$$

Direct numerical simulations were performed for $Re = [400, 800, 1600]$. The simulations were considered to be resolved when the energy, dissipation, and spectra of the modes up to a cutoff frequency k_c remained constant after doubling the number of resolved modes in each direction and halving the time step. A breakdown of the computational details is provided in Table 5.3. Large eddy simulations were performed using various MZ-VMS and VMS models for each of these cases. In addition, simulations considering the static Smagorinsky models are considered. It is noted that, for the Taylor Green vortex, the static Smagorinsky model outperforms the dynamic Smagorinsky model. For the finite memory models, all cases were run with constants of $\tau_0 = 0.1$. The Smagorinsky model uses the default setup of $C_s = 0.16$. A detailed discussion for each case is now provided.

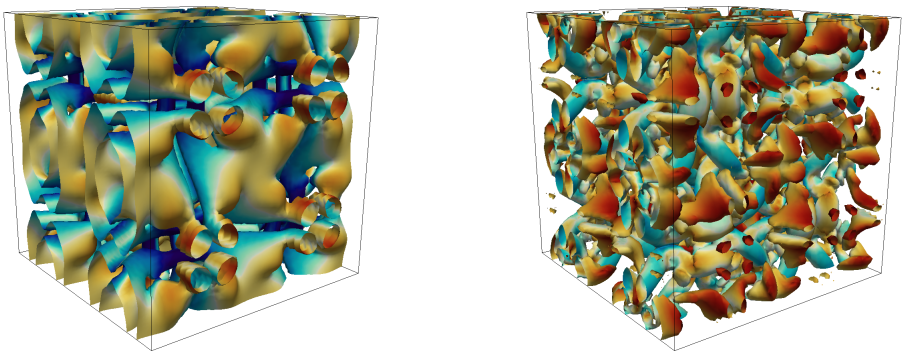
Case 1: Re = 400 on 32 modes

Large eddy simulations were run with 32 resolved modes in each direction using various closure models as well as with no closure model. At this resolution the sys-

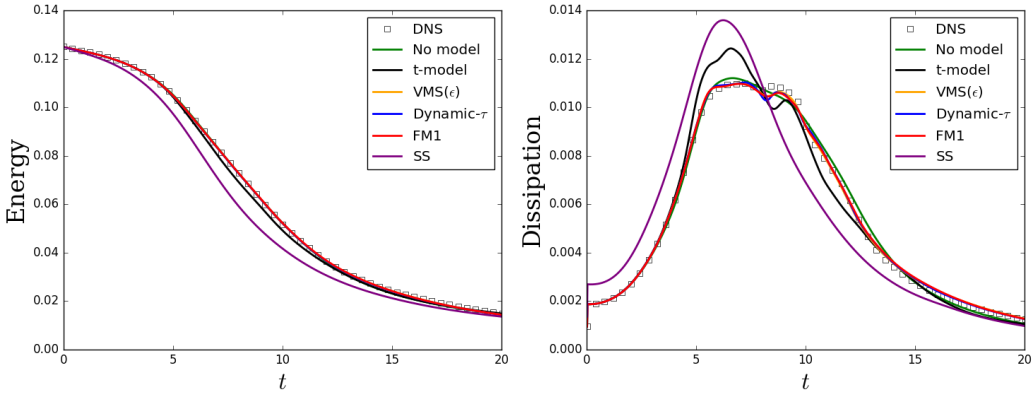
Case	Re	k_c	Grid	Time Step (s)
1	400	16	64^3	0.02
2	800	16	128^3	0.02
3	1600	16	256^3	0.005

Table 5.3: Computational details for DNS simulations of the Taylor Green Vortex.

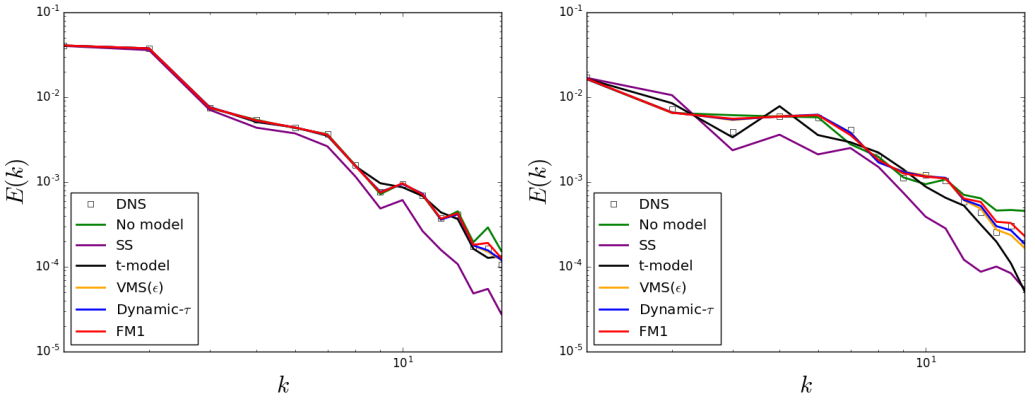
tem is only slightly under-resolved. The results are summarized in Figure 5.11. For physical insight, Figure 5.11a first shows the break down of the vortex via contours of q-criterion. Figure 5.11b shows the temporal evolution of the kinetic energy integrated over the whole domain as well as the temporal evolution of the kinetic energy dissipation rate (computed by $-\frac{dE}{dt}$). The finite memory, dynamic-τ, VMS(ϵ), and no model simulations all perform well. The performance of the t-model is significantly worse than the other methods. The model removes too much energy from the simulation, which leads to poor predictions for the dissipation and energy spectra. This error demonstrates the need for finite memory considerations. The Smagorinsky model is far too dissipative. Only the finite memory, dynamic-τ, and VMS(ϵ) models are able to accurately characterize the double peaked structure of the kinetic energy dissipation rate around $t = 8$. The energy spectra of the simulations at $t = 5$ and $t = 10$ are shown in Figure 5.11c. The Smagorinsky model again performs the worst, being too dissipative at low wavenumbers. In the case of the under-resolved simulation, the lack of a closure model leads to a slight buildup of energy at high wavenumbers. As time increases, this buildup of energy is felt by the lower wavenumbers and leads to small errors across the spectrum. The finite memory, dynamic-τ, and VMS(ϵ) models provide the best prediction. These models are able to mimic the cascade of energy from the resolved scales to the unresolved scales with a degree of accuracy.



(a) Contours of q-criterion from DNS simulation at $t = 5$ (left) and $t = 10$ (right).



(b) Evolution of integral quantities.



(c) Energy spectra at $t = 5, 10$, and 15..

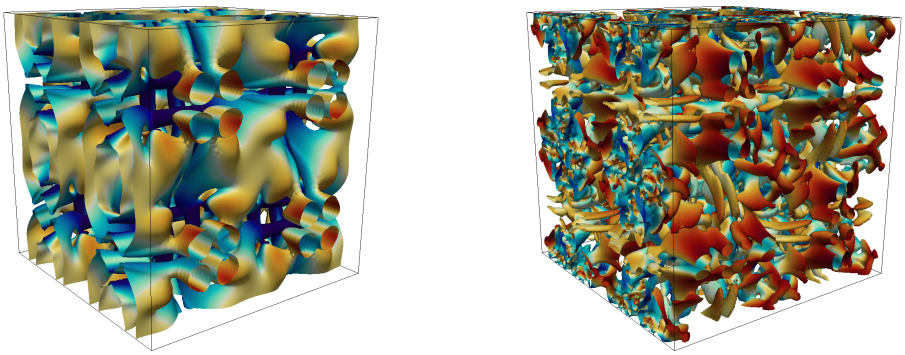
Figure 5.11: Results for numerical simulations of the Taylor Green Vortex at $Re = 400$. DNS quantities are obtained from filtered data obtained on a 64^3 grid. All other models are run on 32^3 grids.

Case 2: Re = 800 on 32 modes

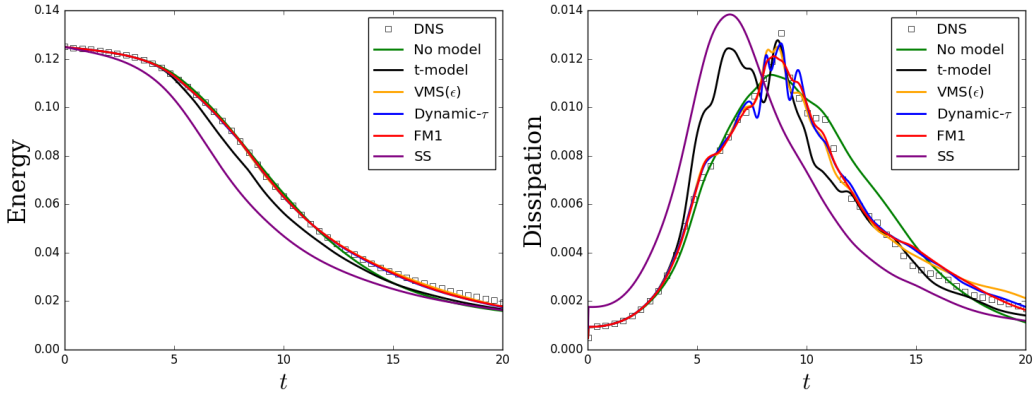
Simulations are again run with 32 resolved modes in each direction. At this resolution the system is under-resolved. The results are summarized in Figure 5.12. The conclusions are much the same as in the $Re = 400$ case. The finite memory, dynamic-τ, and VMS(ϵ) models provide better predictions than the Smagorinsky model and the no model simulation. The t-model removes too much energy and needs renormalization to maintain accuracy. The finite memory, dynamic-τ, and VMS(ϵ) models capture the qualitative structure of the peak in kinetic energy and the dissipation is well characterized (although more error is present than in the $Re = 400$ case). The under-resolved and Smagorinsky model show no indication of the complex peaked structure around $t = 10$. The energy spectra reveals an early buildup of energy for the under-resolved model, while the Smagorinsky model eliminates energy content from the low frequencies early on.

Case 3: Re = 1600 on 32 modes

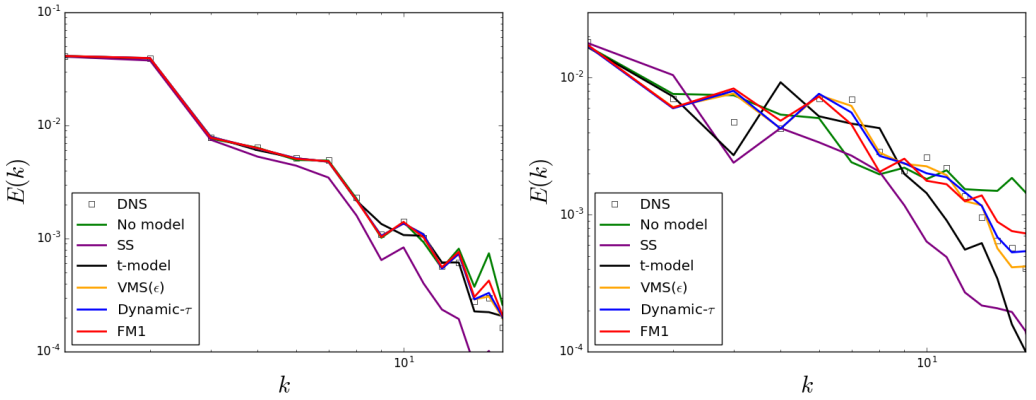
Simulations are again run with 32 resolved modes in each direction. The results are shown in Figure 5.13. Due to the increased Reynolds number the simulation is significantly under-resolved. The simulation run without a subgrid-scale model can not predict the initial peak in dissipation around $t = 8$ and has a buildup of energy at high wavenumbers. The Smagorinsky model is again too dissipative at early time, leading to incorrect predictions in total kinetic energy, dissipation, and spectra. The dynamic-τ, finite memory, and VMS(ϵ) models provide the best predictions. The predictions for kinetic energy and the dissipation of kinetic energy are excellent up to $t = 8$. For time $t > 10$, the MZ-VMS models are less accurate, but qualitative similarities are still present between the predictions and DNS quantities.



(a) Contours of q-criterion from DNS simulation at $t = 5$ (left) and $t = 10$ (right).

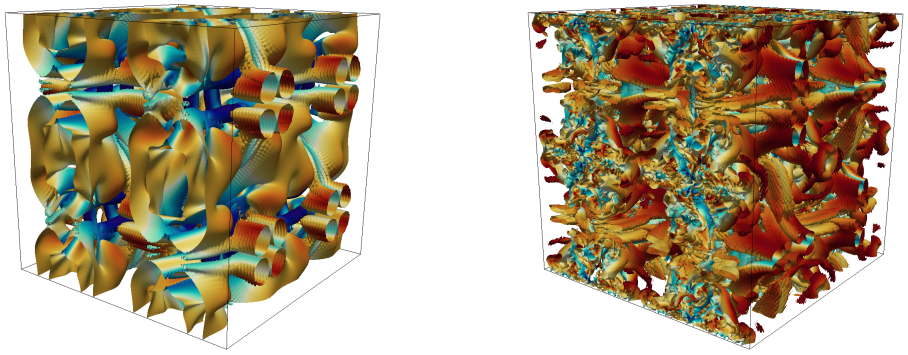


(b) Evolution of integral quantities.

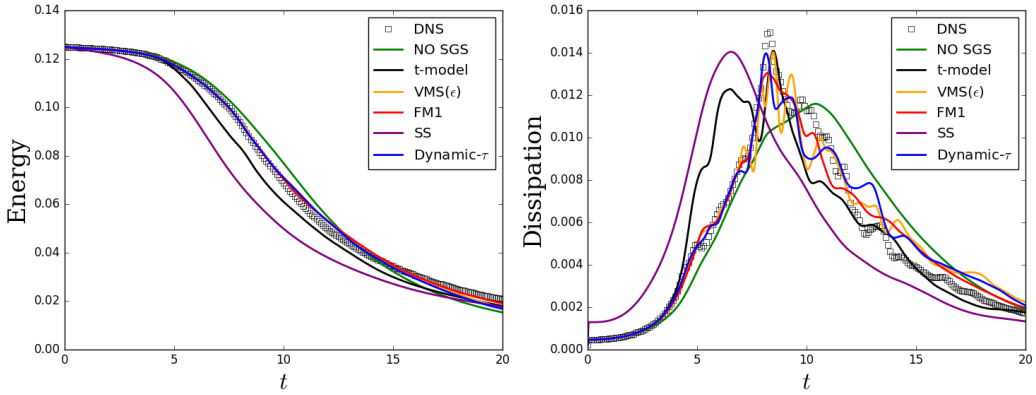


(c) Energy spectra at $t = 5, 10$, and 15..

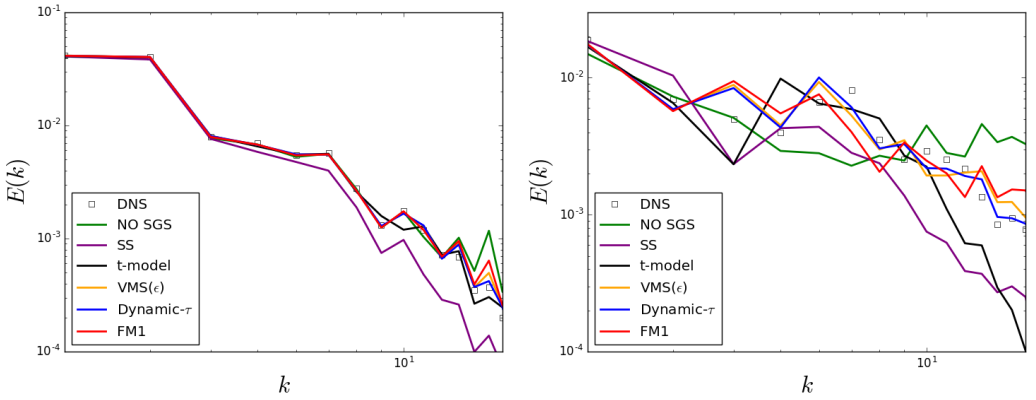
Figure 5.12: Results for numerical simulations of the Taylor Green Vortex at $Re = 800$. DNS quantities are obtained from filtered data obtained on a 128^3 grid. All other models are run on 32^3 grids.



(a) Contours of q-criterion from DNS simulation at $t = 5$ (left) and $t = 10$ (right).



(b) Evolution of integral quantities.



(c) Energy spectra at $t = 5$ (left) and $t = 10$ (right).

Figure 5.13: Results for numerical simulations of the Taylor Green Vortex at $Re = 1600$. DNS quantities are obtained from filtered data obtained on a 256^3 grid. All other models are run on 32^3 grids.

5.2.6 Analysis of MZ-VMS Models for the Taylor Green Vortex

To understand the mechanics of the MZ memory kernel, we consider approximations to the orthogonal dynamics via the orthogonal ODE. For Burgers' equation, the reader will recall that this procedure was used to extract the complete memory kernel. This involved performing an ensemble of orthogonal ODE simulations, each of which are the cost of a full order simulation. For the Navier-Stokes equations this is prohibitively expensive. Instead, we analyze the kernel profiles for a single time instance t in $s-t$ space. This is much more tractable and still provides valuable insight. With the orthogonal ODE approximation, the quantity we examine is

$$e^{t\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}u_{0j} \approx e^{t\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{L}\mathcal{Q}}\mathcal{Q}\mathcal{L}a_{0j}.$$

The profiles of this quantity are an excellent indicator of the memory length and the shape of the memory integrand. Figure 5.14 shows the evolution of $e^{t\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{L}\mathcal{Q}}\mathcal{Q}\mathcal{L}a_{0j}$ for $t = 10$ for various Reynolds numbers. The norm of this quantity is seen to display an exponential decay. The timescale of the decay increases with Reynolds number (or alternatively the ratio of the full order model to the large eddy simulation). However, this timescale is much greater than the memory lengths that were used in the simulations using the finite memory models as well as the memory length predicted by the dynamic procedure. This discrepancy becomes clear when one examines the trajectories of individual modes. In Figure 5.14 it is seen that the temporal trajectories of individual modes are complex, especially for higher Reynolds numbers. It is clear from Figure 5.14 that it is not adequate to model the trajectories of the orthogonal dynamics with a simple generator such as $e^{\tau^{-1}s}$ or with a basic quadrature.

Despite the complexity of the orthogonal dynamics trajectories, the quantity

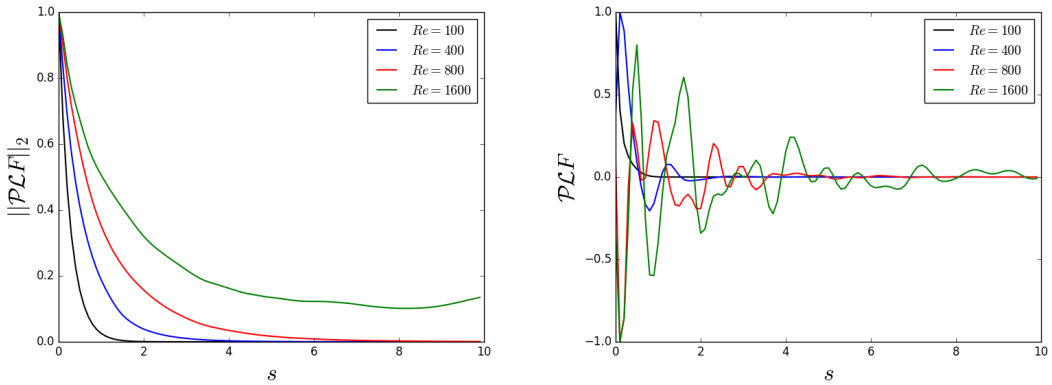


Figure 5.14: Temporal evolution of (normalized) quantities of $\mathcal{PLF}(\tilde{\mathbf{a}}_0(t=10), s)$ for the TGV at various Reynolds numbers. On the left we show a global measure of $\mathcal{PLF}$, while on the right we show individual trajectories for the $16, 16, 16$ a_1 mode.

$\mathcal{PLQLa}_0$ (i.e. $\mathcal{PLF}(\tilde{\mathbf{a}}_0, 0)$) can still be used to obtain a first order approximation to the memory. Table 5.4 shows the correlation coefficient between $\int \mathcal{PLF}(\tilde{\mathbf{a}}_0, s)ds$ and $\mathcal{PLF}(\tilde{\mathbf{a}}_0, 0)$ for various Reynolds numbers with 32^3 DOFs, the scaling ratio between the two, as well as the timescale τ as predicted by the dynamic-τ and VMS(ϵ) models at $t = 10$. It is seen that the correlation coefficient is reasonable for lower Reynolds numbers and degrades as the Reynolds number increases (with the resolution being held constant). This degradation occurs because, as the Reynolds number grows, the orthogonal dynamics become more complex and the integrand at $s = 0$ is less representative of the total integrated quantity. It is also seen that the scalings between the two quantities are similar to the timescale predicted by the dynamic-τ and VMS(ϵ) models. The timescale (or scaling ratio) τ is predicted very well for low Reynolds numbers. The accuracy of the prediction degrades somewhat for higher Reynolds numbers for similar reasons.

5.2.7 Decaying Homogeneous Isotropic Turbulence

Simulations of decaying Homogeneous Isotropic Turbulence are now considered. The Rogallo procedure is used to initialize the spectrum [85]. The initial field satisfies

	$Re = 100$	$Re = 400$	$Re = 800$	$Re = 1600$
Correlation	0.93	0.59	0.40	0.25
$\frac{\|\|\int \mathcal{PLF}(a_0,s)ds\|\|_2}{\|\|\mathcal{PLF}(a_0,0)\|\|_2}$	0.29	0.26	0.27	0.45
τ (Dynamic-τ model)	0.22	0.26	0.22	0.17
τ (VMS(ϵ) model)	0.25	0.30	0.28	0.20

Table 5.4: Correlation between $\int \mathcal{PLF}(\tilde{\mathbf{a}}_0, s)ds$ and $\mathcal{PLF}(\tilde{\mathbf{a}}_0, 0)$ for various Reynolds numbers. The timescale τ as predicted by the dynamic-τ and VMS(ϵ) models at $t = 10$ is also shown for reference. The resolved set of modes is 32^3.

continuity and a prescribed energy spectrum. The initial condition in the frequency domain is given by,

$$a_i(\mathbf{k}) = \alpha e_i^1 + \beta e_i^2, \tag{5.29}$$

where e_i^1 and e_i^2 are mutually orthogonal unit vectors in the plane orthogonal to the wave vector and,

$$\alpha = \left(\frac{E(k)}{4\pi k^2}\right)^{1/2} e^{i\theta_1}\cos(\phi), \qquad \beta = \left(\frac{E(k)}{4\pi k^2}\right)^{1/2} e^{i\theta_2}\sin(\phi),$$

where θ_1, θ_2, and ϕ are uniformly distributed random numbers on the interval $(0, 2\pi)$. The initial spectrum is taken to be,

$$E(k,0) = \frac{q^2}{2A}\frac{1}{k_p^{\sigma+1}}k^\sigma \exp\left(-\frac{\sigma}{2}(\frac{k}{k_p})^2\right),$$

where k_p is the wavenumber at which the energy spectra is maximum, σ is a parameter the slope of the energy spectra at low wavenumbers, and $A = \int_0^\infty k^\sigma \exp(-\sigma k^2/2)dk$. The velocity field is initialized by,

$$\mathbf{a} = \left(\frac{\alpha k k_2 + \beta k_1 k_3}{k(k_1^2+k_2^2)^{1/2}}\right)\hat{i} + \left(\frac{\beta k_2 k_3 - \alpha k k_1}{k(k_1^2+k_2^2)^{1/2}}\right)\hat{j} - \left(\frac{\beta(k_1^2+k_2^2)^{1/2}}{k}\right)\hat{k},$$

where $k = |\mathbf{k}|$. Note that this is the initialization used by Rogallo, but a sign error is present in his 1981 paper [85] for the velocity field in the $\hat{k}$ direction.

Decaying Homogeneous Turbulence at a Low Reynolds Number

Homogeneous isotropic turbulence with an initial Reynolds number of $Re_\lambda \approx 65$ is now considered. The Taylor microscale-based Reynolds number is defined by $Re_\lambda = E\sqrt{\frac{20}{3\nu\epsilon}}$, where ϵ is the dissipation rate. The parameters used are $\sigma = 4, k_p = 5$, and $q^2 = 3$. A direct computation is evolved on a 512^3 mesh until the Kolmogorov microscale starts to grow with time, indicating that realistic homogeneous turbulence is present. The truncated velocity field is then used for initial conditions for the large eddy simulations. For reference the initial spectrum is compared to the experiment of Comte Bellot and Corrsin (CB-C) [28]. Figure 5.15 shows the resulting spectrum and q-criterion. The initial Reynolds number of the simulations is $Re_\lambda \approx 65$. The DNS simulation using 512^3 degrees of freedom resolves up to $k\lambda_k \approx 1$, where $\lambda_k = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}}$ is the Kolmogorov lengthscale.

Figure 5.16 shows the evolution of the resolved kinetic energy and the energy spectra at various time instances. The dynamic-τ model, VMS(ϵ), and FM1 models are compared to the dynamic Smagorinsky model as well as a simulation with no subgrid model. The simulation with no subgrid model predicts the decay of energy reasonably, but the energy spectra are seen to exhibit significant error. The dynamic Smagorinsky model slightly over predicts the decay of energy, but offers improved predictions for the energy spectra. The dynamic-τ model, VMS(ϵ), and FM1 models are seen to provide excellent predictions for both the energy decay and spectra. Figure 5.17 shows the evolution of the energy transferred by the subgrid model. The dynamic Smagorinsky model is seen to remove too much energy from the low wavenumbers and too little from the high wavenumbers. The MZ-VMS models are seen to compare well to the subgrid energy transfer extracted from the DNS data.

An interesting feature to note about the MZ-VMS models is that they deposit a

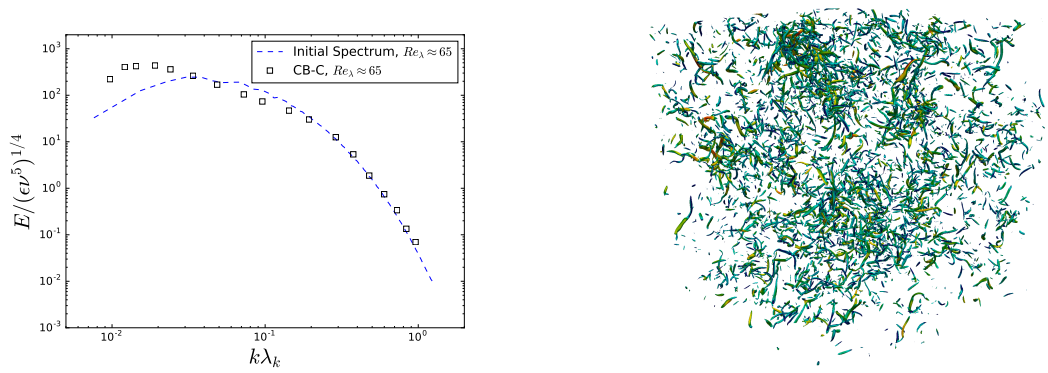


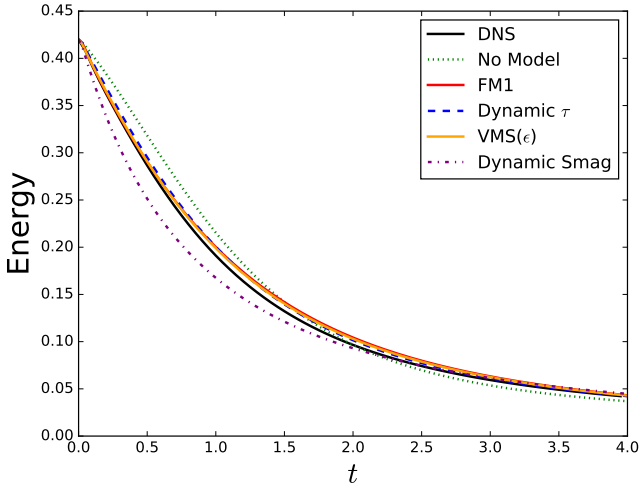
Figure 5.15: Energy spectra (left) and q-criterion (right) of the unfiltered DNS field used for the initialization of the low Reynolds number Large Eddy Simulations. The initial Reynolds number is $Re_\lambda \approx 65$.

small amount of energy into the low wavenumbers ($0 \leq k \leq 16$). This phenomena indicates that the MZ-VMS-based models are capable of predicting local backscatter. Indeed, at $t = 4.0$ (Figure 5.17c) it is clear that a small amount of backscatter is present in the DNS simulation at low wavenumbers. The MZ-VMS models are able to capture this phenomena. Note, however, that the model predicts backscatter at $t = 2.0$ (Figure 5.17b) while none is present in the DNS.

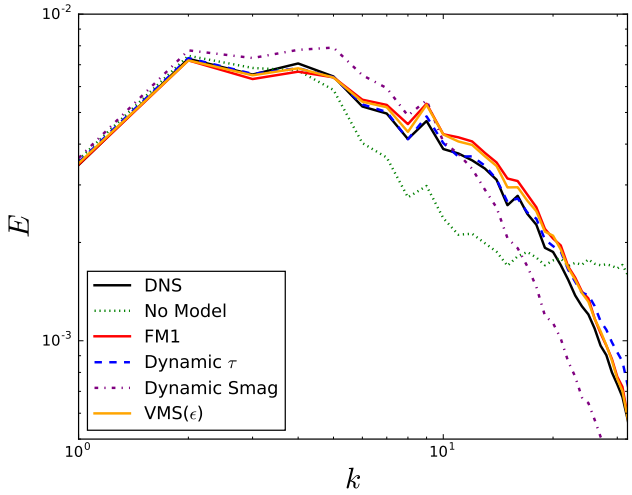
Decaying Homogeneous Turbulence at a Moderate Reynolds Number

The simulation of homogeneous turbulence at a moderate Reynolds number ($Re_\lambda \approx 75$) is now considered. The procedure described in the previous section is again used to initialize the flow field. Figure 5.18 shows the resulting initial spectrum and q-criterion. The initial condition is seen to contain more small-scale vortical structures than in the previous case. At this Reynolds number, the DNS simulation using 512^3 degrees of freedom resolves up to $k\lambda_k \approx 0.5$.

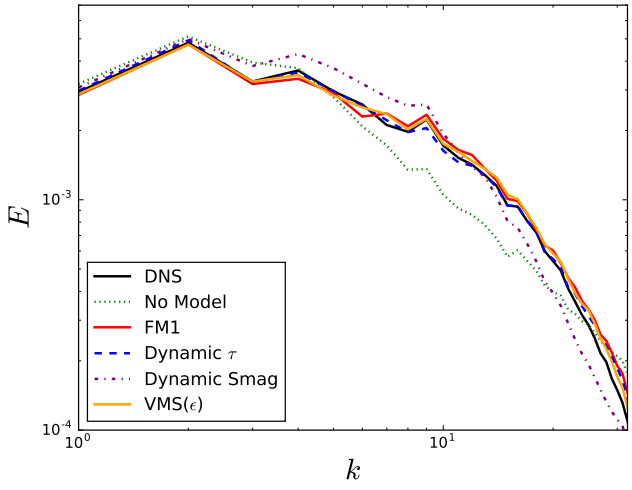
The MZ-VMS models do not perform as well at the increased Reynolds number. Figure 5.19 shows the evolution of the resolved kinetic energy and the energy spectra for multiple time instances. The dynamic-τ, VMS(ϵ), and FM1 models are seen to



(a) Evolution of the resolved kinetic energy.

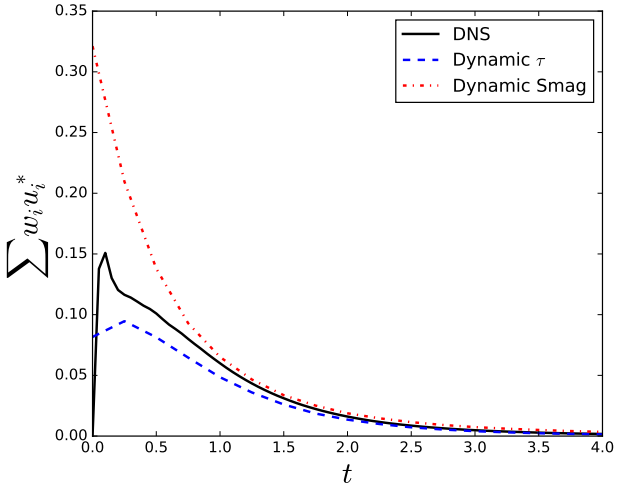


(b) Energy spectrum at $t = 2.0$.

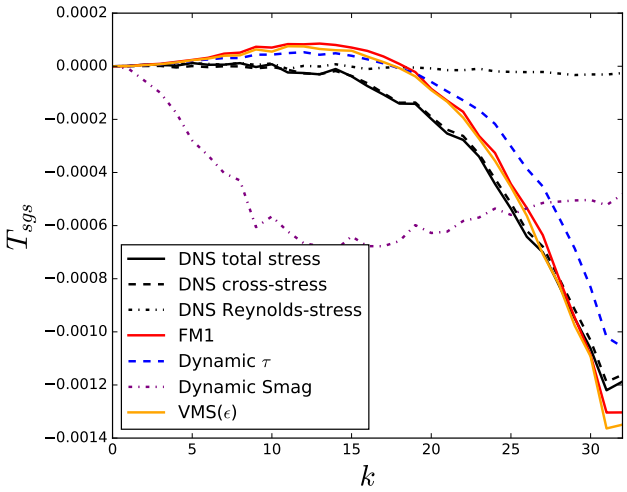


(c) Energy spectrum at $t = 4.0$.

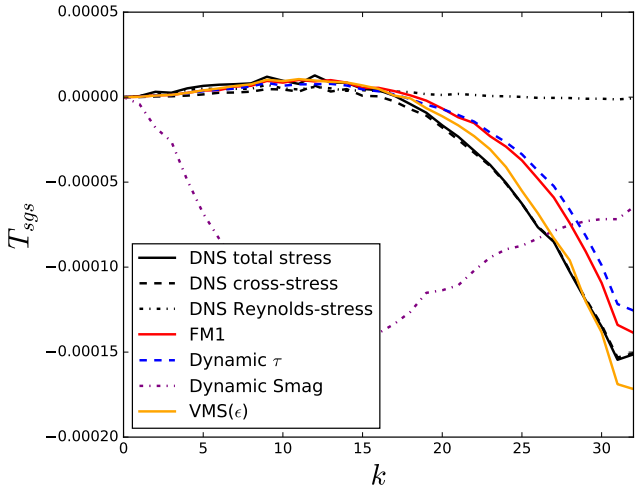
Figure 5.16: Evolution of the resolved kinetic energy and resolved spectra for the low Reynolds number case.



(a) Evolution S-G energy transfer.



(b) S-G energy transfer at $t = 2.0$.



(c) S-G energy transfer at $t = 4.0$.

Figure 5.17: Evolution of the subgrid (S-G) energy transfer for the low Reynolds number case. The subgrid energy transfer is computed by $T_{sgs_k} = w_k u_k{}^*$.

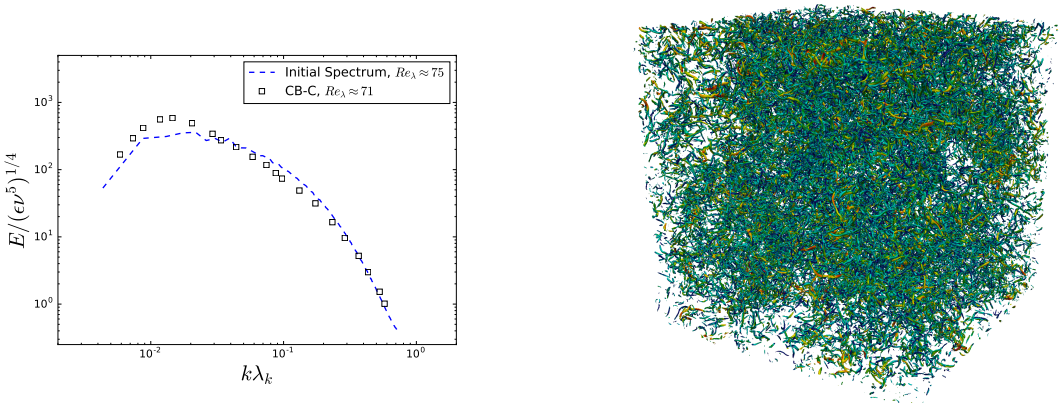
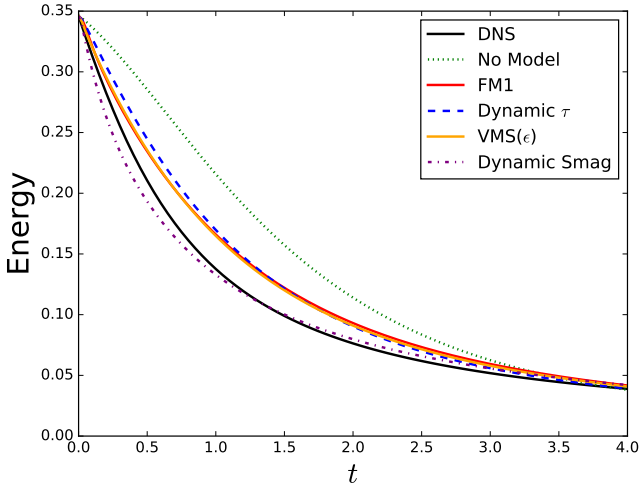


Figure 5.18: Energy spectra (left) and q-criterion (right) of the unfiltered DNS field used for the initialization of the moderate Reynolds number Large Eddy Simulations. The q-criterion is set at the same magnitude as in Fig. 5.15. The initial Reynolds numbers is $Re_\lambda \approx 75$.

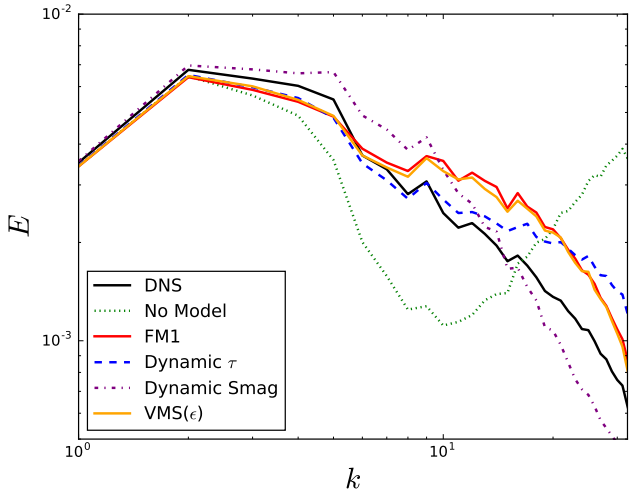
under predict the initial dissipation of energy. A slight build up of energy is seen for the mid wavenumbers at early times. As the simulation evolves, the spectra predicted by the MZ-VMS models improve and approach the DNS simulation. Figure 5.20 shows the evolution of the subgrid energy transfer and the subgrid energy transfer spectra at various time instances. The dynamic-τ, FM1, and VMS(ϵ) models are seen to under-predict the energy transfer for low to mid wavenumbers, while they over predict the energy transfer for higher wavenumbers. This leads to an under-prediction of the dissipation rate.

Decaying Homogeneous Turbulence at a High Reynolds Number

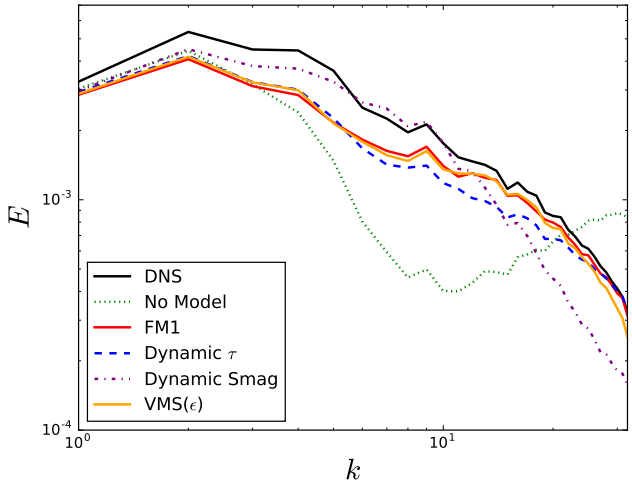
Homogeneous turbulence simulations at a high Reynolds number ($Re_\lambda \approx 164$) are now considered. At this Reynolds number, the DNS simulation requires 1024^3 degrees of freedom to resolve up to $k\lambda_k = 0.5$. We consider large eddy simulations with 64^3 and 128^3 degrees of freedom. These resolutions are 16 and 8 times coarser than the DNS, respectively. Figure 5.21 shows the evolution of resolved kinetic energy as well as the energy and subgrid transfer spectra at $t = 4.0$ for the 64^3 case. The same trends that were observed in the moderate Reynolds number case are again



(a) Evolution of the resolved kinetic energy.

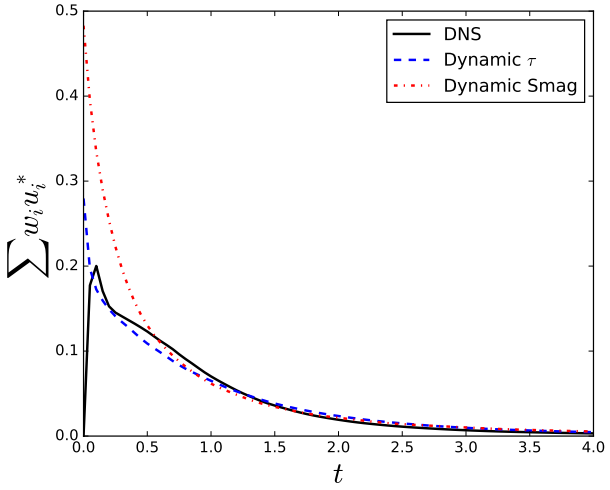


(b) Energy spectrum at $t = 2.0$.

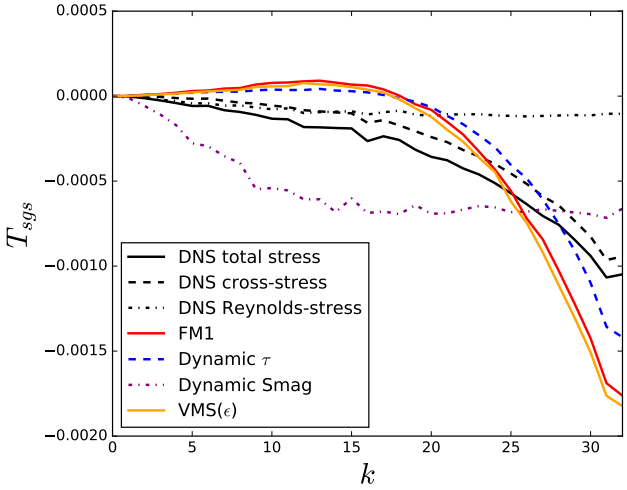


(c) Energy spectrum at $t = 4.0$.

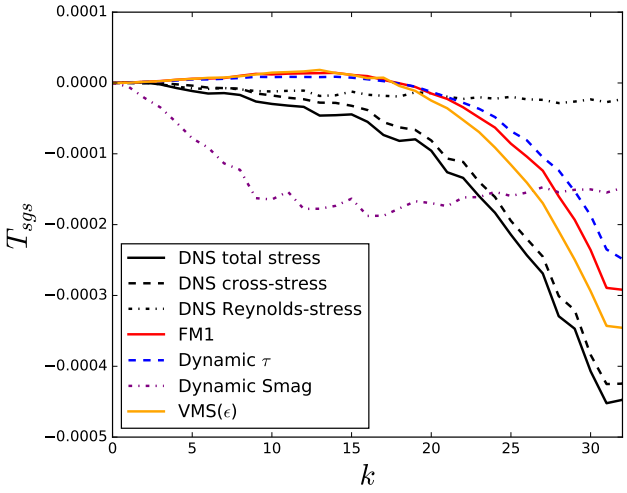
Figure 5.19: Evolution of the resolved kinetic energy for the moderate Reynolds number case.



(a) Evolution of S-G energy transfer.



(b) S-G energy transfer at $t = 2.0$.



(c) S-G energy transfer at $t = 4.0$.

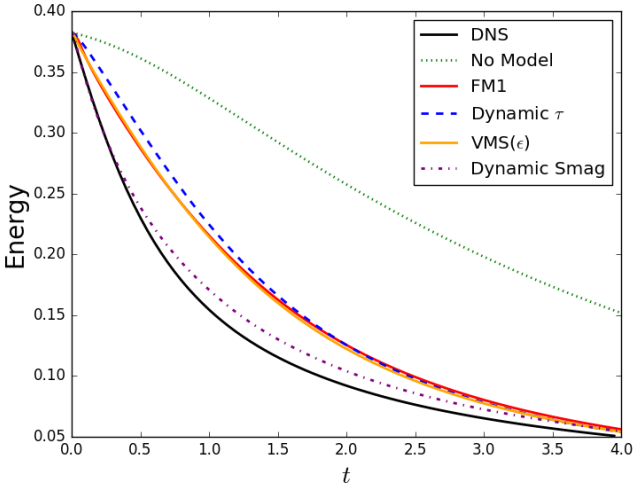
Figure 5.20: Evolution of the subgrid (S-G) energy transfer for the moderate Reynolds number case.

seen. The MZ-VMS models are slightly under dissipative. Not enough energy is removed from the low wavenumbers, while too much energy is removed from the high wavenumbers (as observed in Figure 5.21c). The dynamic Smagorinsky model outperforms the MZ-VMS models in this case. The MZ-VMS models still outperform the no model case. The degradation of the MZ-VMS model performance can be partially attributed to the increasing dominance of the Reynolds stress term. As seen in the subgrid energy transfer figures, the Reynolds stresses plays an increasing role as the Reynolds number grows. As discussed previously, the MZ-VMS models considered do not contain any terms that arise from the Reynolds stresses. Concurrently, the dynamic Smagorinsky model is seen to provide excellent qualitative (and quantitative) agreements between the modeled subgrid energy transfer and the Reynolds stress energy transfer. From this we can draw the conclusion that the Smagorinsky-type models perform well when the Reynolds stress dominates, while the MZ-VMS models perform well when cross stresses dominate.

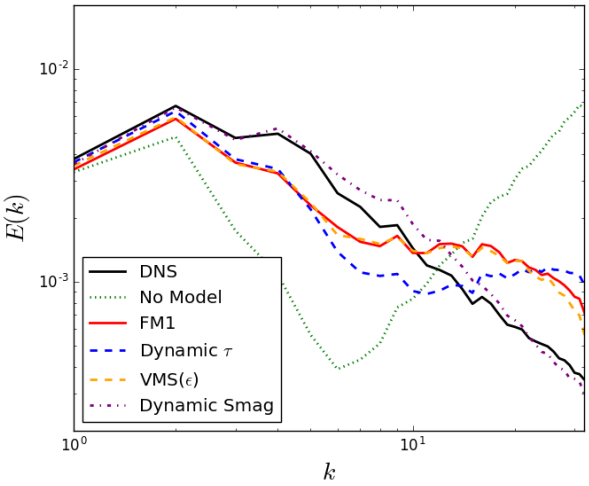
Figure 5.22 shows the same results for the 128^3 case. At this increased resolution, the performance of the MZ-VMS models improve while the performance of the dynamic Smagorinsky model is slightly worse than what it was in the 64^3 case. This is attributed to the fact that cross stresses are more dominant than the Reynolds stresses at this higher resolution.

Evolution of the memory timescale

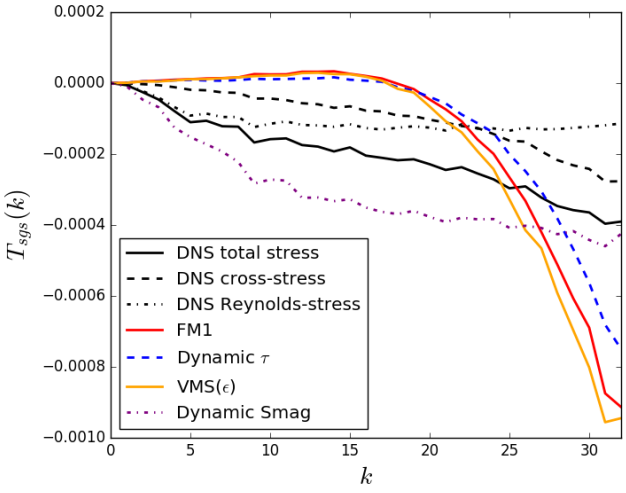
We now examine the evolution of the memory timescale predicted by the dynamic procedure and the VMS(ϵ) models. Figure 5.23 shows the evolution of τ as predicted by the dynamic procedure and VMS(ϵ) models for both the low Reynolds number case and the moderate Reynolds number case. As a reference, the predicted constant τ is additionally compared to the Kolmogorov timescale of the LES, $\tau_k = \sqrt{\frac{\nu}{\epsilon}}$. The



(a) Evolution of the resolved kinetic energy.

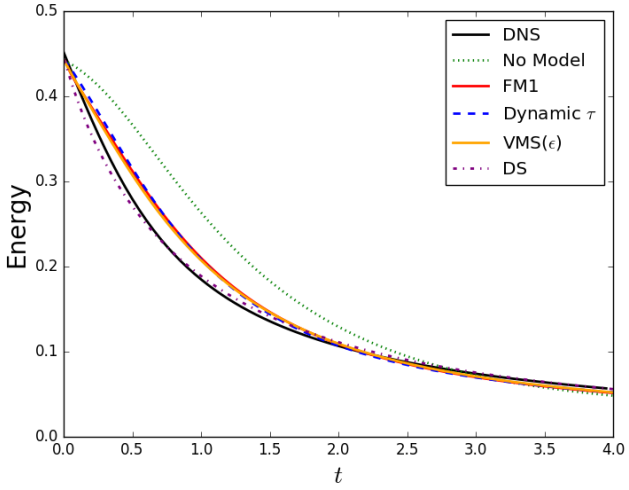


(b) Energy spectrum at $t = 4.0$.

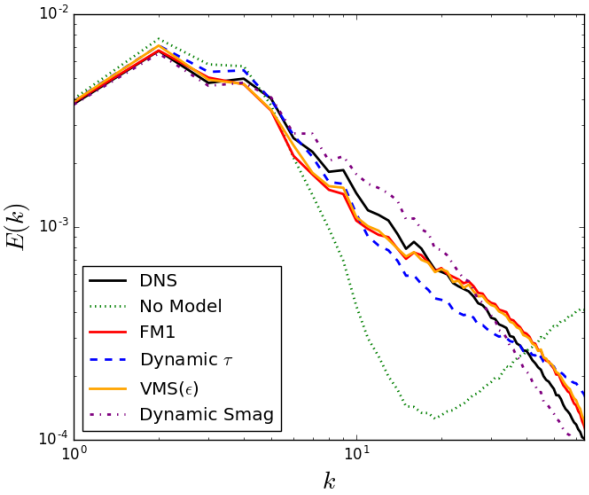


(c) S-G energy transfer at $t = 4.0$.

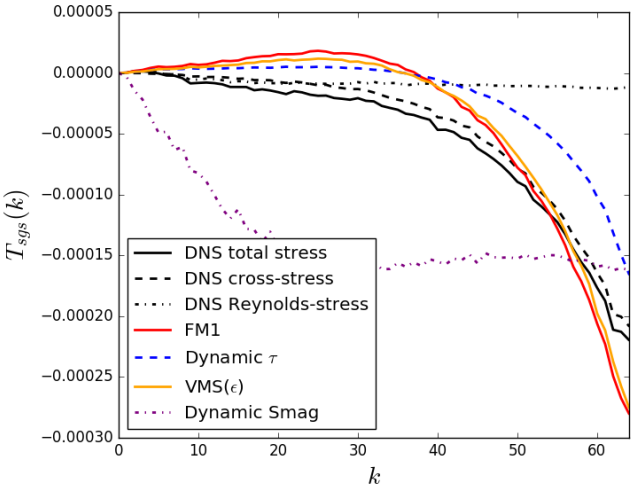
Figure 5.21: Summary of simulations of the high Reynolds number case using 64^3 DOFs.



(a) Evolution of the resolved kinetic energy.

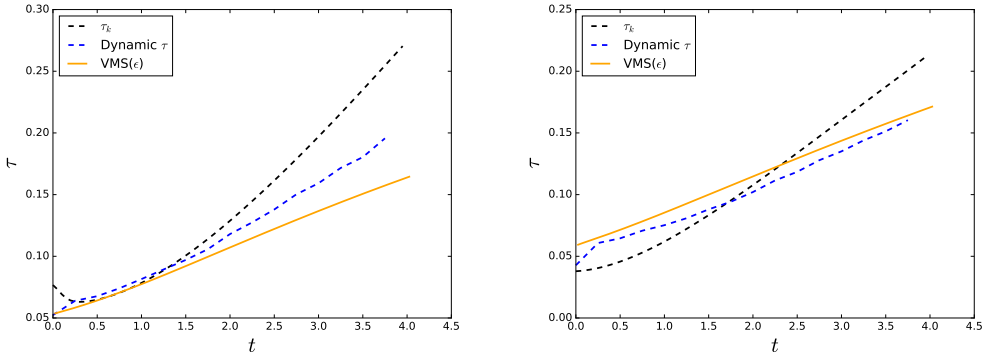


(b) Energy spectrum at $t = 4.0$.



(c) S-G energy transfer at $t = 4.0$.

Figure 5.22: Summary of simulations of the high Reynolds number case using 128^3 DOFs.



(a) Low Reynolds number case.

(b) Moderate Reynolds number case.

Figure 5.23: Evolution of the memory timescale as predicted by the dynamic procedure.

Kolmogorov timescale is the relevant physical quantity as it reflects the small-scale dynamics. The predicted timescales are seen to grow approximately linearly with respect to time. The linear growth of τ demonstrates that the dynamic-τ model and VMS(ϵ) models behave similarly to a renormalized t-model. Both models, however, predict the linear growth with respect to time while the renormalized t-model assumes it. It is further observed that τ compares well with the Kolmogorov timescale, demonstrating that there is a qualitative similarity between the timescales of the memory kernel and that of the small-scale dynamics.

Validity of model assumptions

In the previous sections, it was seen that the performance of the dynamic-τ model degrades slightly as the LES becomes increasingly coarse. This degradation is due to the breakdown of the assumption that the memory integral is well correlated to its value at $s = 0$. To demonstrate this, we compute the a priori correlation coefficient between the exact memory integral and its value at $s = 0$ as a function of the Reynolds number for several resolutions. The correlation coefficient between the memory integral and the term predicted by the Smagorinsky closure model is

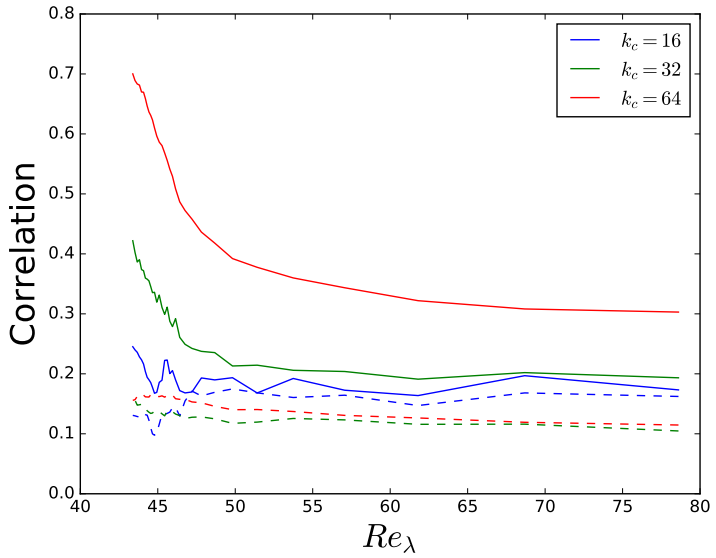


Figure 5.24: Correlation coefficient between the point-wise field of the approximated memory integral and the true memory integral. The solid lines are the dynamic-τ model, while the dashed lines are the dynamic Smagorinsky model.

additionally computed for reference. Figure 5.24 shows the resulting correlations. It is seen that, as the Reynolds number increases, the correlation coefficient drops. It is further seen that the correlation coefficient drops as the model becomes increasingly coarse. This drop in the correlation coefficient is due to the fact that the orthogonal dynamics becomes increasingly complex. It is noted, however, that the point-wise correlation between the field predicted by the dynamic-τ model is better than that predicted by the dynamic Smagorinsky model for all resolutions.

5.2.8 Two-Dimensional Homogeneous Turbulence

The case of two-dimensional turbulence is of interest from a modeling viewpoint due to the fact that the physical phenomena present in such systems are drastically different than their three dimensional counterparts. The notion of Richardson's energy cascade, for instance, does not exist in two-dimensions. We briefly consider the simulation of two-dimensional turbulence to demonstrate the robustness of MZ-

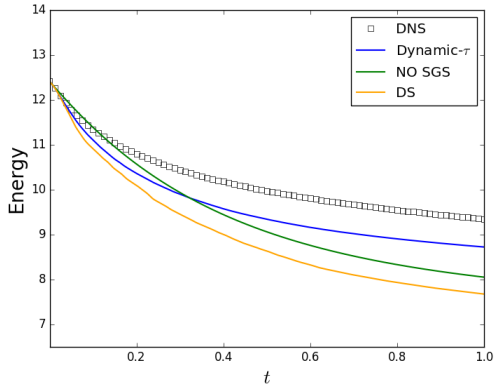
VMS models as compared to physics-based models.

Simulations are considered for initial conditions that assume an energy spectrum of the form [16],

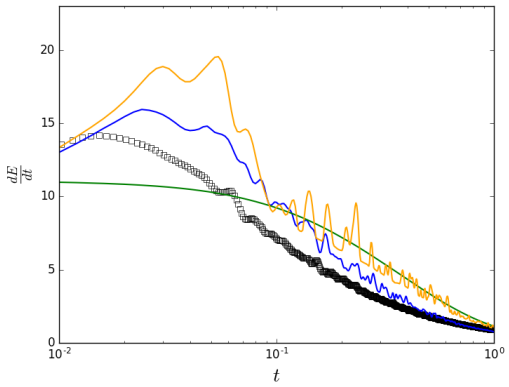
$$E(k,0) = \frac{1}{2} a_s u_0^2 k_p^{-1} \left(\frac{k}{k_p}\right)^{2s+1} \exp\left[-\left(s+\frac{1}{2}\right)\left(\frac{k}{k_p}\right)^2 \right], \tag{5.30}$$

with $s = 3, u_0 = 5$, and $k_p = 25$. Simulations are first considered at $\nu = 1/1600$. A DNS simulation is carried out using 512^2 DOFs, while the LES are carried out with 128^2 DOFs. Simulations are performed using the dynamic-τ model and dynamic Smagorinsky model, as well as the case with no closure model present. All the models considered are autonomous and require no tuning. Figure 5.25 shows the numerical results. All simulations are seen to under predict the energy content for $t > 0.2$. The dynamic-τ model and dynamic Smagorinsky models over-predict the dissipation rate at all times, while the simulation run with no model initially under-predicts the dissipation rate and then over-predicts it for later time. Of the models considered, the dynamic-τ model performs the best. The energy dissipation rate is reasonably characterized and the predicted energy spectra at $t = 1.0$ is excellent.

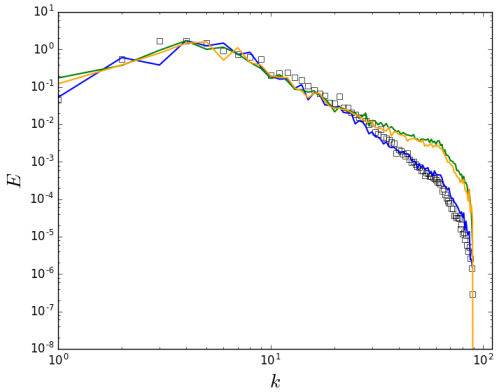
We next consider simulations with $\nu = 1/16000$. At this higher Reynolds number, viscosity drains minimal energy from the resolved system and the transfer of energy is largely due to the non-linear triadic interactions. The DNS simulation is run on a 1024^2 mesh, while the LES use a 128^2 mesh. Figure 5.26 shows the numerical results. At the increased viscosity, the simulation run with no closure model under-predicts the decay of kinetic energy, while both simulations run with closure models over-predict the kinetic energy decay. The energy spectra of both the simulation run with dynamic Smagorinsky and that run with no model show a buildup in energy for high wavenumbers. The dynamic-τ model again outperforms the Smagorinsky model as well as the LES run with no model. The decay in kinetic energy is reasonably



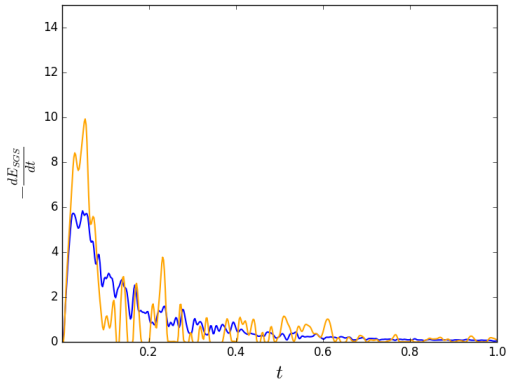
(a) Evolution of total kinetic energy



(b) Evolution of the kinetic energy dissipation

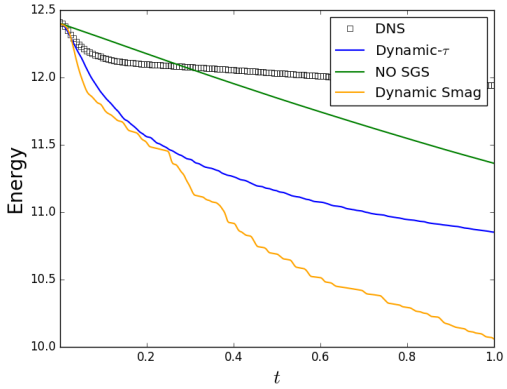


(c) Energy spectra at $t = 1.0$

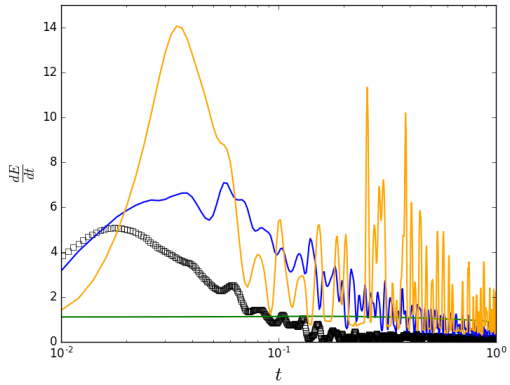


(d) Dissipation contributed from the subgrid-scale model

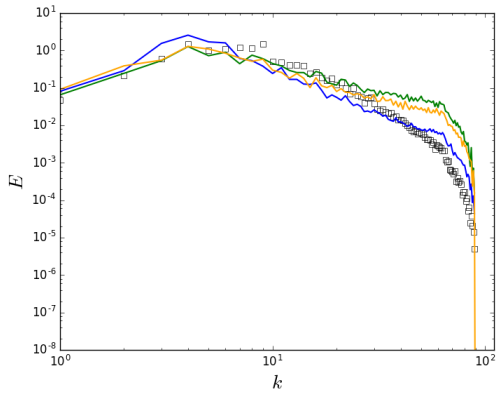
Figure 5.25: Results for two-dimensional homogeneous turbulence simulations at $\nu = 1/1600$.



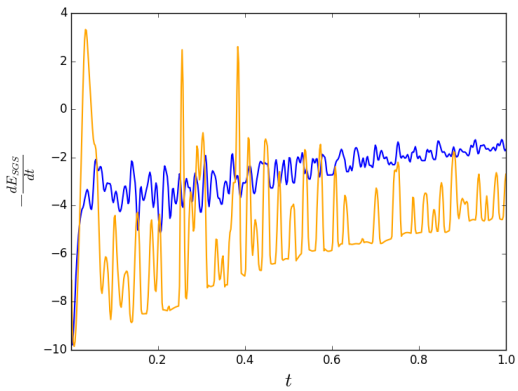
(a) Evolution of total kinetic energy



(b) Evolution of the kinetic energy dissipation



(c) Energy spectra at $t = 1.0$



(d) Dissipation contributed from the subgrid-scale model

Figure 5.26: Results for two-dimensional homogeneous turbulence simulations at $\nu = 1/16000$.

captured, as is the energy spectrum at $t = 1.0$.

5.2.9 Turbulent Channel flow

Thus far only decaying homogeneous problems have been considered. These problems are characterized by the decay of kinetic energy and the growth in memory length. The simulation of fully developed turbulent channel flow with a Fourier-Chebyshev pseudo-spectral method is now considered. There are several significant differences between the channel and the triply periodic Navier-Stokes equations that are worth pointing out. First, the channel is only streamwise and spanwise periodic. The flow is inhomogeneous in the wall-normal direction with no-slip boundary

conditions at the wall. Due to the inhomogeneity in the wall normal direction, no simple solution to the Poisson pressure equation exists for the channel (in contrast to Eq. 5.23, where pressure simply appears as a projection). This complicates the derivation of MZ-based models as pressure is determined by the solution of an elliptic equation. Pressure in incompressible flows is not a thermodynamic variable determined by an equation of state. In some sense, it can be viewed as a Lagrange multiplier that enforces the velocity field to be solenoidal at all times [44]. This viewpoint is adopted in this work. We choose to neglect the effects of pressure in the subgrid-scale model, and instead ensure that the continuity equation is satisfied in the solution space of the resolved variables.

For the channel flow, the Navier-Stokes equations are solved in skew-symmetric form. This form has been shown to minimize aliasing errors [11]. The skew-symmetric form is, however, significantly more complicated than the primitive form of the Navier-Stokes equations and the analytic derivation of the MZ models is considerably more challenging. As such, the MZ-based models are computed by numerically evaluating the Fréchet derivative, as discussed in Section 4.6. The dynamic model is constructed by coarse-graining in the periodic directions. A no-slip boundary condition is used at the wall. Note that, since we only coarse-grain in the periodic directions, the effect of coarse-graining on the boundary conditions is not addressed in this work. The Navier-Stokes equations are solved using a coupled semi-implicit Adams-Bashforth scheme for time integration, as in [68]. The continuity equation is directly enforced at each time-step, bypassing the need for pressure boundary conditions. The solver is a slight modification of the PySpectral solver. All FFT calculations (including the Chebyshev transforms) are de-aliased by the 3/2 rule. The solutions are compared to the dynamic Smagorinsky model. Simulations at Reynolds

Re_τ	L_x	L_y	L_z	N_x	N_y	N_z	Δt^+	Δt
180	4π	2	$4/3\pi$	32	64	32	0.24	0.02
395	2π	2	π	32	128	32	0.28	0.005
590	2π	2	π	32	192	32	0.31	0.0025

Table 5.5: Physical and numerical details for large eddy simulations of the channel flow.

numbers of $Re_\tau = \{180, 395, 590\}$ are considered. The relevant computational details are given in Table 5.5. The domain sizes were selected to match those in Ref. [71] and are designed to be long enough such that the periodicity constraint imposed by the Fourier ansatz is appropriate. Here we only consider simulations using the dynamic-τ model. Simulations utilizing the finite memory models, however, are presented by the author in [75].

Large eddy simulation solutions at $Re_\tau = 180$ are shown in Figure 5.27 and are compared to unfiltered DNS data from [71]. The filtering process will not affect the mean velocity profiles, but the filtered Reynolds stress profiles are expected to be slightly different. The mean velocity profile predicted by the dynamic-τ model is slightly larger in magnitude than that of the DNS, but it is much improved from the simulation run with no subgrid model. The model correctly damps the R_{22}, R_{33}, and R_{12} Reynolds stresses. A slight amplification of the R_{11} Reynolds stress is seen near the wall. The reason for this is not quite clear in the context of the dynamic-τ model. The mean non-dimensional memory timescale as predicted by the dynamic model is $\tau_{\mathcal{P}}^+ \approx 2.35$, where $\tau_{\mathcal{P}}^+ = \tau_{\mathcal{P}} u_\tau^2/\nu$. It is important to note that the dynamic procedure predicted a timescale that did not increase with time. A growing timescale, as was the case for the decaying problems, would be incorrect in this context.

Figure 5.28 shows the velocity profiles for LES solutions at $Re_\tau = 395$ and $Re_\tau = 590$. As was seen in the case of homogeneous turbulence, increasing the Reynolds number (while holding the resolution constant) leads to a slight decrease in model performance. The mean velocity profiles for both cases are slightly over-predicted.

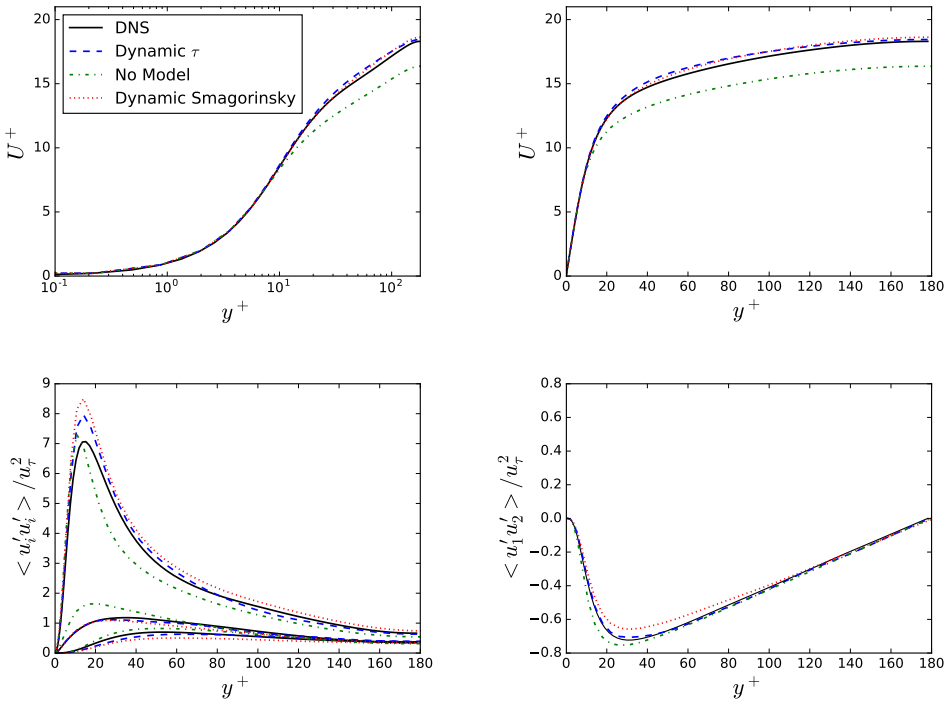


Figure 5.27: Statistical properties for fully-developed channel flow at $Re_\tau = 180$.

The mean memory timescales as predicted by the dynamic model are $\tau_{\mathcal{P}}^+ \approx \{2.2, 2.5\}$ for the 395 and 590 case, respectively. These non-dimensional timescales are similar to those observed in the $Re_\tau = 180$ case. It is also worth noting that the predicted timescale is roughly ten times the time step for all cases. This is due to the fact that the memory length scales with the spectral radius of the Jacobian [73], as does the largest stable time step. This indicates that the time-step can be used as an alternative indicator to the memory length.

5.3 Application to Magnetohydrodynamics

One of the appeals of the MZ-VMS framework is that it is a general mathematical formulation. This allows the framework to be applied to various multiphysics problems. To conclude this chapter, the MZ-VMS framework applied to ideal in-

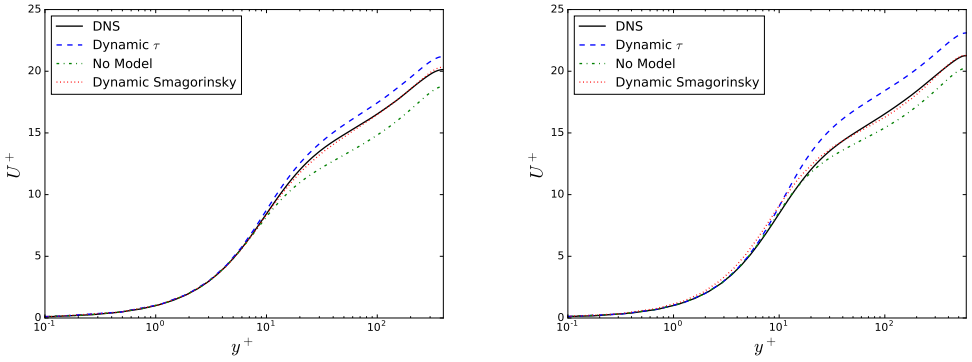


Figure 5.28: Velocity profiles for fully-developed channel flow at $Re_\tau = 395$ (left) and $Re_\tau = 590$ (right).

compressible magnetohydrodynamics (MHD) is considered. MHD describes the behavior of an electrically conducting fluid under the presence of a magnetic field, and is prevalent in areas such as fusion energy and astrophysics. The dynamics of ideal incompressible MHD are governed by a coupling of the Navier-Stokes equations and Maxwell equations. The equations of ideal MHD can be written to be,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u} - \mathbf{B} \otimes \mathbf{B}) = -\nabla \mathbf{p} + \nu \nabla^2 \mathbf{u}, \tag{5.31}$$

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{u} \times \mathbf{B}) = \lambda \nabla^2 \mathbf{B}, \tag{5.32}$$

subject to the divergence free constraints $\nabla \cdot \mathbf{B} = 0$, $\nabla \cdot \mathbf{u} = 0$.

5.3.1 Numerical Method

The ideal MHD equations are solved using the same Fourier-Galerkin spectral method solver previously described. The addition of the MHD equations leads to a slight modification of the Poisson pressure equation. Note that, in the Fourier-Galerkin method, a Lagrange multiplier is not required to enforce the divergence free condition in the magnetic field as the solenoidal constraints are automatically satisfied [89].

5.3.2 MZ-VMS Dynamic-τ Model

For the ideal MHD equations, we consider both the VMS(ϵ) model and the dynamic-τ model. The selection of the energy transfer constraint in the dynamic-τ model is of interest for MHD as energy naturally is transferred between the magnetic field and velocity field. As such, in the dynamic-τ model we allow for two timescales: one for the closure term arising in the magnetic equations and the other for the closure arising in the momentum equations. The timescales are determined through two constraints: the energy transferred out of the kinetic energy field and the energy transferred out of the magnetic field. Mathematically,

$$\tau_{\mathcal{P}}^{u} = \frac{1}{C_q} \frac{\mathbb{R}\left(\overline{\mathbf{a}}_j^u, R_j^u(\overline{\mathbf{a}}) - \overline{\mathcal{G}} R_j^u(\hat{\mathbf{a}}) \right)}{\left(\overline{\mathbf{a}}_0^u, e^{t\mathcal{L}} \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\mathbf{a}_0^u - \left(\frac{\Delta_{\mathcal{P}}}{\Delta_{\overline{\mathcal{P}}}}\right)^{1.5} e^{t\mathcal{L}} \overline{\mathcal{P}}\mathcal{L}\overline{\mathcal{Q}}\mathcal{L}\mathbf{a}_0^u \right)}, \tag{5.33}$$

$$\tau_{\mathcal{P}}^{B} = \frac{1}{C_q} \frac{\mathbb{R}\left(\overline{\mathbf{a}}_j^B, R_j^b(\overline{\mathbf{a}}) - \overline{\mathcal{G}} R_j^B(\hat{\mathbf{a}}) \right)}{\left(\overline{\mathbf{a}}_0^B, e^{t\mathcal{L}} \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\mathbf{a}_0^B - \left(\frac{\Delta_{\mathcal{P}}}{\Delta_{\overline{\mathcal{P}}}}\right)^{1.5} e^{t\mathcal{L}} \overline{\mathcal{P}}\mathcal{L}\overline{\mathcal{Q}}\mathcal{L}\mathbf{a}_0^B \right)}, \tag{5.34}$$

where $\tilde{\mathbf{a}}^u$ corresponds to the velocity modes and $\tilde{\mathbf{a}}^B$ to the magnetic modes (and the same for R).

5.3.3 Taylor Green Vortex

The Taylor Green vortex at a mild Reynolds number is first considered. The viscosity is taken to be $\nu = \lambda = 2.5 \times 10^{-4}$, which corresponds to a Reynolds number of $Re = R_m = 5.1 \times 10^3$. Note that the magnetic Reynolds number and hydrodynamic Reynolds number are the same. The DNS simulation requires 512^3

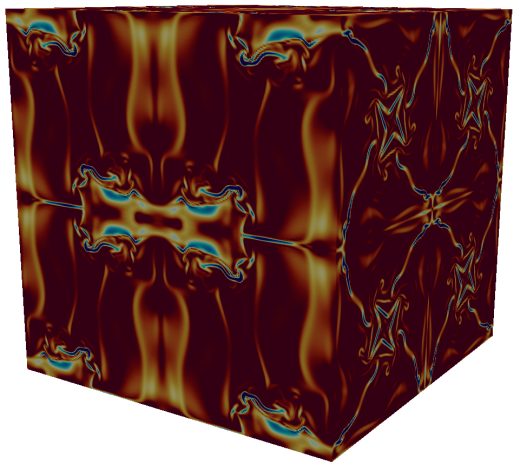


Figure 5.29: Contours of the current field, $\nabla \times \mathbf{B}$.

modes. The initial conditions are [90],

$$\mathbf{u}(x,y,z) = \left\{ \begin{array}{c} \sin(x)\cos(y)\cos(z) \\ -\cos(x)\sin(y)\cos(z) \\ 0 \end{array} \right\}, \qquad \mathbf{B}(\mathbf{x}) = B_0 \left\{ \begin{array}{c} \cos(x)\sin(y)\sin(z) \\ \sin(x)\cos(y)\sin(z) \\ -2\sin(x)\sin(y)\cos(z) \end{array} \right\}.$$

The simulations are initialized such that the initial energy in the magnetic field is equal to the kinetic energy. This corresponds to $B_0 = 1/\sqrt{3}$. The energy is broken into kinetic and magnetic energy,

$$E_{ke} = \int \frac{1}{2}(u_1^2 + u_2^2 + u_3^2)d\Omega, \qquad E_I = \int \frac{1}{2}(B_1^2 + B_2^2 + B_3^2)d\Omega.$$

Figure 5.29 shows the current field for the problem at $t = 8$.

Large eddy simulations are performed on 32^3 grids. Figure 5.30 shows the performance of the dynamic-τ and VMS(ϵ) models. The models are compared to a simulation not utilizing any subgrid model. The timescales used in the VMS(ϵ) model are given by [90],

$$\tau_{\mathcal{P}}^u = \left[\frac{4}{h^2}(\mathbf{u}\cdot\mathbf{u} + \mathbf{B}\cdot\mathbf{B}) + 3\pi\nu^2\left(\frac{4}{h^2}\right)^2 \right]^{-1/2}, \tag{5.35}$$

$$\tau_{\mathcal{P}}^{B} = \left[\frac{4}{h^2}\mathbf{B}\cdot\mathbf{B} + 3\pi\nu^2\big(\frac{4}{h^2}\big)^2\right]^{-1/2}. \tag{5.36}$$

Note that in [90], VMS-based models were shown to perform significantly better than physically inspired eddy viscosity models, thus eddy viscosity models are not considered here. In Figure 5.30 the dynamic-τ model is seen to lead to significant improvements over the simulation run with no subgrid model. This demonstrates the importance of accounting for unresolved physics. The dynamic-τ model is shown to slightly outperform the VMS(ϵ) model. The energy spectra of both models is seen to compare well with the DNS simulation, although some buildup of energy is seen for the kinetic energy spectra.

Figure 5.31 shows the same results but for a 64^3 simulation. At this resolution, the simulation is moderately resolved. The dynamic-τ and VMS(ϵ) models are seen to behave similarly and both offer improvements over the no model case.

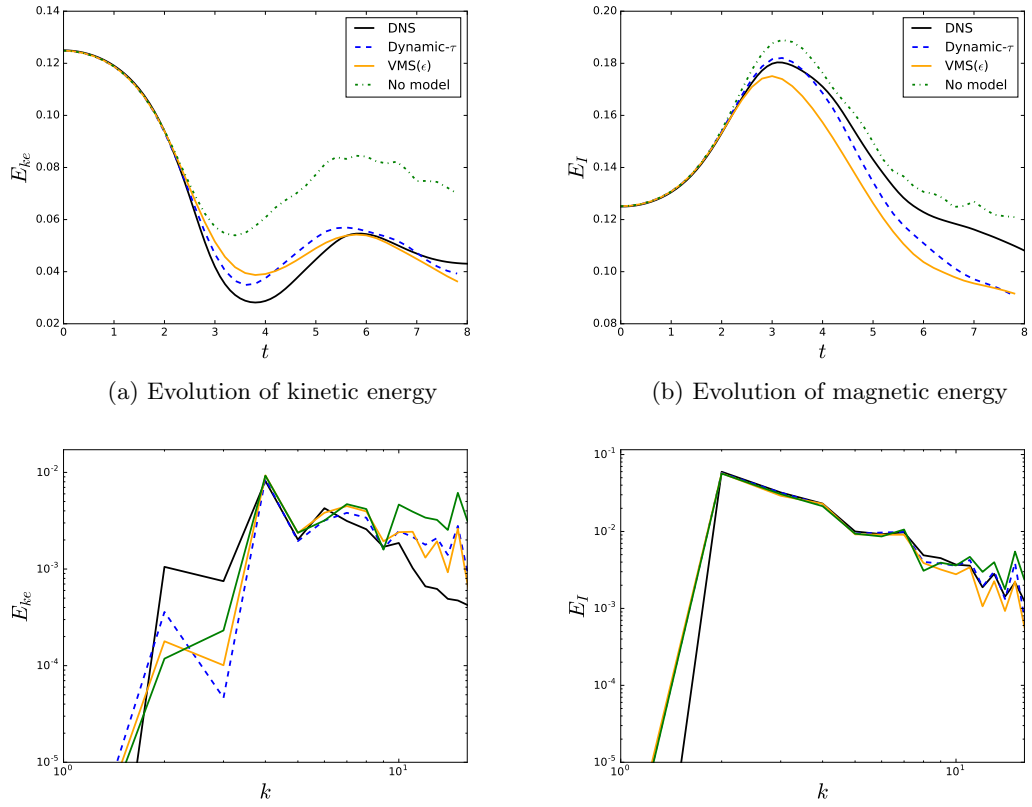
5.3.4 Forced Homogeneous Turbulence

The next case considered is that of homogeneous turbulence using a Taylor Green stirring forcing [1]. The forcing is imposed only on the momentum equations and is given by,

$$\mathbf{F} = f_0\big[\sin(k_f x)\cos(k_f y)\cos(k_f z)\hat{\mathbf{i}} - \cos(k_f x)\sin(k_f y)\cos(k_f z)\hat{\mathbf{j}}\big], \tag{5.37}$$

where $k_f = 2$ and $f_0 = 0.25$. Direct numerical simulation data for this case at a resolution of 1024^3 is available from the Johns Hopkins Turbulence Database [1]. Large eddy simulations are considered at a resolution of 64^3.

Figure 5.32 shows the steady state energy spectra for kinetic and magnetic energy. Simulations run with no subgrid-scale model are seen to display a significant buildup



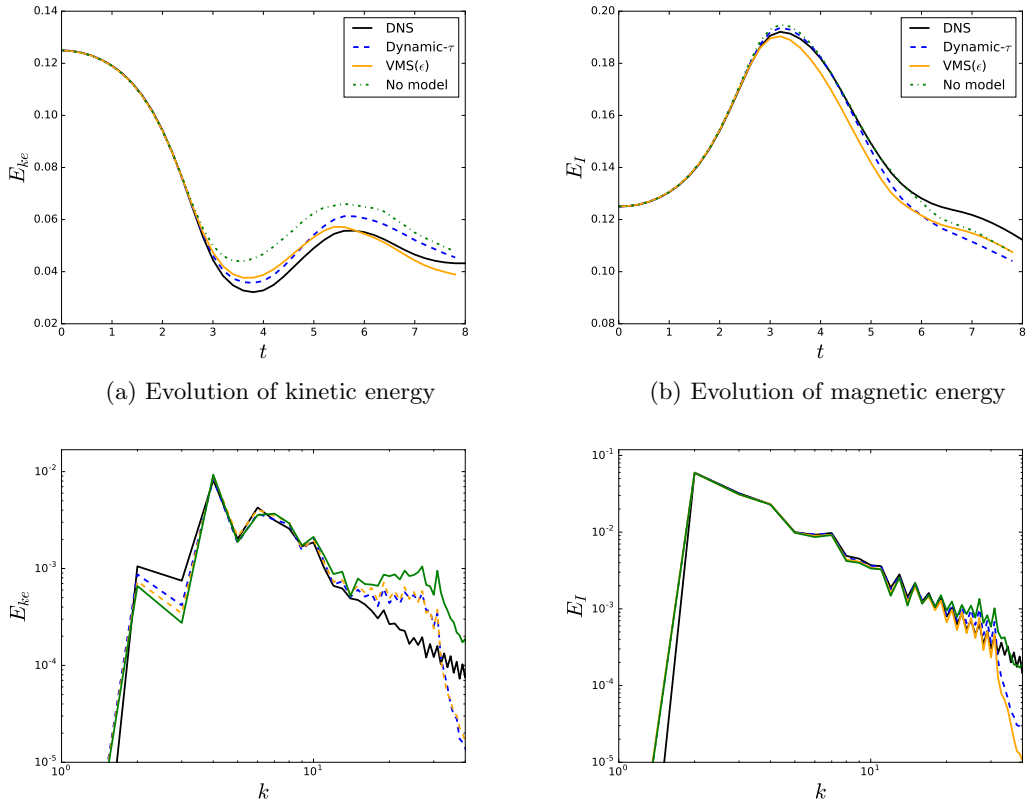
(a) Evolution of kinetic energy

(b) Evolution of magnetic energy

(c) Kinetic energy spectra at $t = 4$.

(d) Magnetic energy spectra at $t = 4$.

Figure 5.30: Numerical results for MHD Taylor Green vortex problem at 32^3. Note that the DNS is filtered.



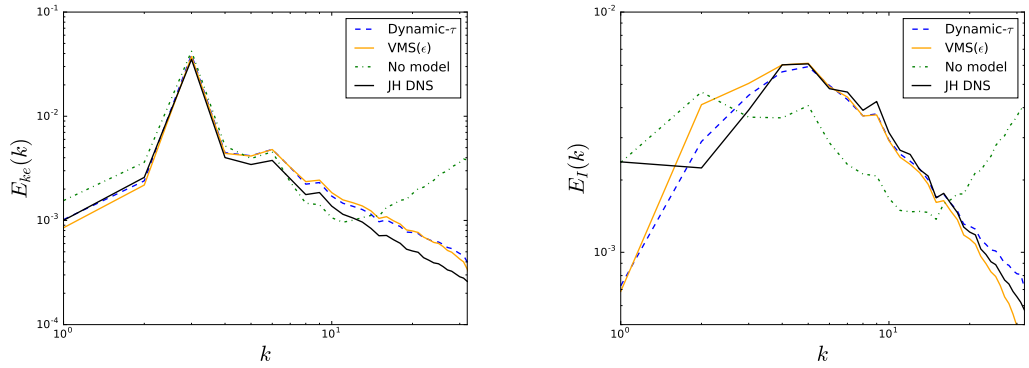
(a) Evolution of kinetic energy

(b) Evolution of magnetic energy

(c) Kinetic energy spectra at $t = 4$.

(d) Magnetic energy spectra at $t = 4$.

Figure 5.31: Numerical results for MHD Taylor Green vortex problem at 64^3. Note that the DNS is filtered.



(a) Kinetic energy spectra

(b) Magnetic energy spectra

Figure 5.32: Steady-state energy spectra for isotropic magnetohydrodynamic turbulence.

of kinetic and magnetic energy at high frequencies. This leads to poor predictions across all wavenumbers. Simulations run with the dynamic-τ model lead to significant improvements over the baseline simulations. Both the kinetic and magnetic energy spectrum are well captured.

5.4 Spectral Method Summary

In this chapter we have explored the use of MZ-VMS methods as closure models for a variety of systems in the spectral setting. We demonstrated that the MZ-VMS-based methods lead to improved answers for Burgers' equation, two and three-dimensional Navier-Stokes equations, and for the equations of magnetohydrodynamics. Important aspects of this chapter are:

1. The demonstration that finite memory effects lead to errors in the well studied t-model.

2. Numerical results for the viscous Burgers' equation show that the dynamic-τ model, "tuned" finite memory models, and VMS(ϵ) model all outperform classic LES models.

3. When applied to the Navier-Stokes equations, the dynamic-τ, "tuned" finite

memory model, and VMS(ϵ) model outperform the static and dynamic Smagorinsky models at low to mild Reynolds numbers. At higher Reynolds numbers the Smagorinsky models outperform the MZ and VMS models.

4. The degradation of the MZ and VMS models at high Reynolds numbers was shown to be partially attributed to their inability to model the Reynolds stress terms. It was shown, however, that the methods are able to accurately model the cross stress terms.

5. The robustness of MZ and VMS methods was demonstrated through applications to two dimensional turbulence and magnetohydrodynamics.

The results presented here are specific to the spectral method. The following chapter will examine the application of MZ-VMS to the more robust finite element method.

CHAPTER VI

MZ-VMS for Finite Elements

In the previous chapters we demonstrated that the Mori-Zwanzig formalism is a powerful tool to be used for the construction of subgrid-scale closure models. The discussion to this point, however, has been limited by the assumption of global smoothness. These assumptions limit the application of the developed techniques to methods such as Fourier-Galerkin spectral methods. These methods are limited to canonical domains and are not sufficiently robust for the complex flows and domains commonly encountered in science and engineering. In this chapter, we relax the assumption of global smoothness and develop the Mori-Zwanzig formalism for finite element methods. The discontinuous Galerkin method will be primarily considered.

6.1 MZ-VMS Formulation for the Finite Element Method

In this section, we develop the formulation of the MZ-VMS framework for finite element methods. The following discussion will encompass both continuous and discontinuous Galerkin methods. As before, denote the test and trial space as $\mathcal{V} = \tilde{\mathcal{V}} \oplus \mathcal{V}'$, where $\mathcal{V} \equiv L^2(\Omega)$. Define $\mathcal{T}_h$ to be the decomposition of the domain Ω into a set of non-overlapping elements, T, over Ω_k with boundaries Γ_k. We seek solutions for the coarse scales in the finite element space

$$\tilde{\mathcal{V}} = \{w \in L^2(\Omega) : v|_{\mathcal{T}} \in P^k(T), \forall T \in \mathcal{T}_h\}, \tag{6.1}$$

where P^k is the space of polynomials up to degree k. The above definition allows for discontinuities between elements. The continuous Galerkin formulation makes use of a subspace of Eq. 6.1 that enforces continuity between elements. Some notation is beneficial before proceeding. Define,

$$(\cdot,\cdot)_\Omega = \sum_{k\in\mathcal{T}} (\cdot,\cdot)_{\Omega_k}, \qquad (\cdot,\cdot)_\Gamma = \sum_{k\in\mathcal{T}} (\cdot,\cdot)_{\Gamma_k}. \tag{6.2}$$

The FEM weak formulation of Eq. 3.17 becomes,

$$(w,u_t)_\Omega + (w,\mathcal{R}(u))_\Omega + (w,b(u))_\Gamma = (w,f)_\Omega \qquad \forall w\in\mathcal{V}, \tag{6.3}$$

where $b(u)$ is a boundary operator that arises via integration-by-parts. In the context of discontinuous Galerkin, one may consider Eq. 6.3 to be the "DG strong form". The finite element method proceeds to approximate the solution u in a finite dimensional subspace, $\tilde{\mathcal{V}}\subset\mathcal{V}$. With $\mathcal{V}'\perp\tilde{\mathcal{V}}$, we obtain the multiscale weak formulation,

$$\begin{aligned}(\tilde{w},\tilde{u}_t)_\Omega + (\tilde{w},\mathcal{R}(\tilde{u}))_\Omega + (\tilde{w},\mathcal{R}(u)-\mathcal{R}(\tilde{u}))_\Omega + \\ (\tilde{w},b(\tilde{u}))_\Gamma + (\tilde{w},b(u)-b(\tilde{u}))_\Gamma = (\tilde{w},f)_\Omega \qquad \forall\tilde{w}\in\tilde{\mathcal{V}},\end{aligned} \tag{6.4}$$

$$\begin{aligned}(w',\tilde{u}_t)_\Omega + (w',\mathcal{R}(\tilde{u}))_\Omega + (w',\mathcal{R}(u)-\mathcal{R}(\tilde{u}))_\Omega + \\ (w',b(\tilde{u}))_\Gamma + (w',b(u)-b(\tilde{u}))_\Gamma = (w',f)_\Omega \qquad \forall w'\in\mathcal{V}'.\end{aligned} \tag{6.5}$$

To illustrate the Mori-Zwanzig procedure, consider the discrete system resulting from Eqns. 6.4 and 6.5. As the basis functions in FEM are typically not orthonormal, we retain the mass matrices in what follows. Equations 6.4 and 6.5 become,

$$\frac{d\tilde{\mathbf{a}}}{dt} = \tilde{\mathbf{M}}^{-1}\big[-(\tilde{\mathbf{w}},\mathcal{R}(\tilde{u}))_\Omega - (\tilde{\mathbf{w}},\mathcal{R}(u)-\mathcal{R}(\tilde{u}))_\Omega + (\tilde{\mathbf{w}},f)_\Omega - (\tilde{\mathbf{w}},b(\tilde{u}))_\Gamma - (\tilde{\mathbf{w}},b(u)-b(\tilde{u}))_\Gamma\big], \tag{6.6}$$

$$\frac{d\mathbf{a}'}{dt} = \mathbf{M}'^{-1}\big[-(\mathbf{w}',\mathcal{R}(\tilde{u}))_\Omega - (\mathbf{w}',\mathcal{R}(u)-\mathcal{R}(\tilde{u}))_\Omega + (\mathbf{w}',f)_\Omega - (\mathbf{w}',b(\tilde{u}))_\Gamma - (\mathbf{w}',b(u)-b(\tilde{u}))_\Gamma\big], \tag{6.7}$$

where the mass matrices are,

$$\tilde{\mathbf{M}} = (\tilde{\mathbf{w}}, \tilde{\mathbf{w}}^T)_\Omega, \qquad \mathbf{M}' = (\mathbf{w}', \mathbf{w}'^T)_\Omega. \tag{6.8}$$

Note that there is no coupling between the mass matrices as a result of L^2 orthogonality of the coarse and fine scales. Compactly, we can express the entire system as,

$$\frac{d\mathbf{a}}{dt} = \mathbf{M}^{-1}\big[-(\mathbf{w}, \mathcal{R}(u))_\Omega - (\mathbf{w}, b(u))_\Gamma + (\mathbf{w}, f)_\Omega \big], \tag{6.9}$$

with $\mathbf{M} = (\mathbf{w}, \mathbf{w}^T)_\Omega$. Through the Mori-Zwanzig procedure, we can integrate out the fine-scale variable and express Eq. 6.6 as,

$$(\tilde{\mathbf{w}}, \tilde{u}_t)_\Omega + (\tilde{\mathbf{w}}, \mathcal{R}(\tilde{u}))_\Omega + (\tilde{\mathbf{w}}, b(\tilde{u}))_\Gamma = (\tilde{\mathbf{w}}, f)_\Omega + \tilde{\mathbf{M}} \int_0^t K(\tilde{\mathbf{a}}(t-s), s) ds. \tag{6.10}$$

The memory kernel is given by,

$$K_j(\mathbf{a}_0, t) = \mathcal{P}\mathcal{L} e^{t\mathcal{Q}\mathcal{L}} \mathcal{Q}\mathcal{L} a_{0j}, \tag{6.11}$$

where,

$$\mathcal{L} \equiv \sum {M_{ij}}^{-1} \big[-(w_j, \mathcal{R}(u_0))_\Omega - (w_j, b(u_0))_\Gamma + (w_j, f)_\Omega \big] \frac{\partial}{\partial a_{0i}}. \tag{6.12}$$

The summation is over all global unknowns. The memory kernel at $s = 0$ is again of interest. The derivation closely follows that of the smooth case and one finds,

$$\begin{aligned} \tilde{\mathbf{M}} K(\tilde{\mathbf{a}}(t), 0) = & \int_\Omega \int_\Omega (\tilde{\mathbf{w}} \mathcal{R}')(x) \Pi'(x, y) (\mathcal{R}(\tilde{u}) - f)(y) d\Omega_y d\Omega_x \\ & + \int_\Omega \int_\Gamma (\tilde{\mathbf{w}} \mathcal{R}')(x) \Pi'(x, y) b(\tilde{u}(y)) d\Gamma_y d\Omega_x \\ & + \int_\Gamma \int_\Omega (\tilde{\mathbf{w}} b')(x) \Pi'(x, y) (\mathcal{R}(\tilde{u}) - f)(y) d\Gamma_y d\Omega_x \\ & + \int_\Gamma \int_\Gamma (\tilde{\mathbf{w}} b')(x) \Pi'(x, y) b(\tilde{u}(y)) d\Gamma_y d\Gamma_x, \end{aligned} \tag{6.13}$$

where again,

$$\mathcal{R}' = \frac{\partial \mathcal{R}}{\partial \tilde{u}}, \qquad b' = \frac{\partial b}{\partial \tilde{u}}. \tag{6.14}$$

Remarks

- Compared to the smooth case, the finite element method gives rise to additional interactions between the coarse and fine scales.
- The coarse-scale equation, Eq. 6.10, is again non-local in time.
- The memory kernel at $s = 0$ contains both volumetric and surface integrals. The form of this term is again similar to that obtained by Hughes [48] for the "rough" FEM case.

6.2 The Discontinuous Galerkin Approach

In the remainder of this chapter, we apply MZ-VMS models to numerical methods that utilize the discontinuous Galerkin (DG) method. Consider the system of conservation laws,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = 0 \tag{6.15}$$

with $\mathbf{x} \in \Omega$. In the discontinuous Galerkin approach, the domain Ω is subdivided into individual elements, Ω_k. The state variable $\mathbf{u}$ is then approximated by basis functions over Ω,

$$\mathbf{u}(\mathbf{x}, t) = \sum_{k=0}^{N-1} \sum_{j=0}^{p} \mathbf{a}_{j,k}(t) \mathbf{w}_{k,j}(\mathbf{x}),$$

where $\mathbf{a}$ are the basis coefficients and $\mathbf{w}$ are the basis functions. Note that in the DG approach, the basis functions have local support and are hence discontinuous between the elements.

The DG approach proceeds by multiplying Eq. 6.15 by the basis functions and integrating over the support of each basis function,

$$\int_{\Omega_k} \mathbf{w}_{k,j} \left[\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} \right] d\Omega = 0. \tag{6.16}$$

The flux term is then integrated by parts. This integration by parts couples all the

elements and establishes the finite-volume aspect of the DG approach. One has [38],

$$\int_{\Omega_k} \mathbf{w}_{k,j} \frac{\partial \mathbf{u}}{\partial t} d\Omega - \int_{\Omega_k} \nabla \mathbf{w}_{k,j} \cdot \mathbf{F} d\Omega + \int_{d\Omega_k} \mathbf{w}^+_{k,j} \hat{\mathbf{F}}(\mathbf{u}^+, \mathbf{u}^-, \hat{n}) dl = 0, \tag{6.17}$$

where + and − refers to the element interior and exterior, respectively. The term $\hat{\mathbf{F}}$ is the numerical flux between the faces.

Equation 6.17 is the fundamental equation that we will solve.

6.2.1 Legendre Basis Functions

Any finite element approach requires the selection of basis functions. The Lagrange basis functions are convenient as the basis coefficients (which are the unknowns) are simply the nodal values of the Lagrange interpolating polynomial. In the context of MZ-VMS, however, Lagrange basis functions introduce additional complications. To illustrate this, consider a second order Lagrange interpolating polynomial for $u(\zeta, t = 0)$ for $\zeta \in (-1, 1)$,

$$u(\zeta, 0) = a_0 \frac{\zeta}{2}(\zeta - 1) - a_1(\zeta + 1)(\zeta - 1) + a_2 \frac{\zeta}{2}(\zeta + 1). \tag{6.18}$$

The representation of u in Eq. 6.18 was constructed by interpolating $u(-1), u(0)$, and $u(1)$. The modal coefficients a_i in this case are the nodal values of u,

$$u(\zeta, 0) = u(-1) \frac{\zeta}{2}(\zeta - 1) - u(0)(\zeta + 1)(\zeta - 1) + u(1) \frac{\zeta}{2}(\zeta + 1). \tag{6.19}$$

A simple truncation projector that eliminates the highest mode would give,

$$\mathcal{P}u(\zeta, 0) = u(-1) \frac{\zeta}{2}(\zeta - 1) - u(0)(\zeta + 1)(\zeta - 1). \tag{6.20}$$

Equation 6.20 is no longer a consistent representation of u, as seen in Figure 6.1. As previously mentioned, we require a formulation in which the fine scales are orthogonal to the coarse scales. The Legendre polynomials offer one such route.

The Legendre polynomials are a set of orthogonal polynomials in $\zeta = [-1, 1]$. The first few polynomials are given by,

$$w_0 = 1, \qquad w_1 = \zeta, \qquad w_2 = \frac{1}{2}\big(3\zeta^2 - 1\big), \qquad w_3 = \frac{1}{2}\big(5\zeta^3 - 3\zeta\big), \tag{6.21}$$

and the solution is represented by,

$$u(\zeta, 0) = \sum_i a_i w_i. \tag{6.22}$$

Due to orthogonality, the Legendre polynomials can be shown to obey the following relation,

$$\int_{-1}^{1} w_i(\zeta) w_j(\zeta) d\zeta = \frac{2}{2j+1}\delta_{ij}. \tag{6.23}$$

As such, the coefficients a_j are found by multiplying by w_i and integrating,

$$\int_{-1}^{1} w_i(\zeta) \sum_{j=0}^{N} a_j w_j(\zeta) d\zeta = \int_{-1}^{1} u(\zeta) w_i(\zeta) d\zeta, \tag{6.24}$$

$$a_j = \frac{2j+1}{2} \int_{-1}^{1} u(\zeta) w_j(\zeta) d\zeta. \tag{6.25}$$

The Legendre polynomials can serve as a hierarchical basis. To demonstrate this, consider again a second order Legendre polynomial approximating $u(\zeta, t = 0)$,

$$u(\zeta, 0) = a_0 + a_1\zeta + a_2\frac{1}{2}\big(3\zeta^2 - 1\big). \tag{6.26}$$

Application of an MZ truncation projector that eliminates the highest mode gives,

$$\mathcal{P}u(\zeta, 0) = a_0 + a_1\zeta. \tag{6.27}$$

The projected representation of u in Eq 6.27 is still a first order approximation to u (see again Figure 6.1), and hence it is seen that the basis is hierarchical. As such, we can safely truncate high-order modes and still expect a consistent numerical method. The MZ-VMS closure model will then account for the eliminated high-order modes. In the rest of this work, we will be working exclusively with the Legendre basis.

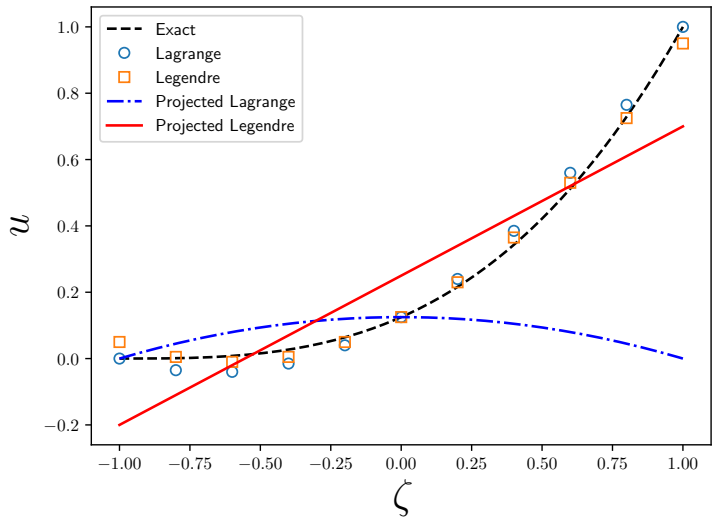


Figure 6.1: Illustration of the hierarchical property of the Legendre polynomials. Shown are quadratic approximations to $u = (\zeta + 1)^3/8$ using a Lagrange basis and a Legendre basis. Projecting out the a_2 coefficient in the Lagrange basis (Eq. 6.18) destroys the solution approximation. Projecting out the a_2 coefficient in the Legendre basis leads to a linear approximation of u.

Remarks

- Note that it is possible to formulate MZ-VMS for the Lagrange basis. The methodology, however, will have to contain a linear transformation to a hierarchical orthogonal basis to eliminate high frequency modes, as well as the corresponding transformation back to the Lagrange basis. Using a Legendre basis simplifies the formulation and provides a more intuitive framework.

6.3 Numerical Properties of MZ-VMS models for DG

Here we outline some important properties of the t, τ, and VMS(ϵ) models applied to the DG method.

6.3.1 Residual-Based Artificial Viscosity

Consider the DG discretization of 6.15,

$$\int_\Omega \mathbf{w}\mathbf{u}_t d\Omega + \int_\Omega \mathbf{w}\nabla\cdot\mathbf{F}(\mathbf{u})d\Omega + \int_\Gamma \mathbf{w}\mathbf{b}(\mathbf{u},\mathbf{n})d\Gamma = 0, \tag{6.28}$$

where again $\mathbf{b}$ is a boundary operator, and $\mathbf{n}$ is the normal vector at element interfaces. Application of the MZ-VMS procedure leads to,

$$\int_\Omega \tilde{\mathbf{w}}\mathbf{u}_t d\Omega + \int_\Omega \tilde{\mathbf{w}}\nabla\cdot\mathbf{F}(\tilde{\mathbf{u}})d\Omega + \int_\Gamma \tilde{\mathbf{w}}\mathbf{b}(\tilde{\mathbf{u}},\mathbf{n})d\Gamma = \tilde{\mathbf{M}}\int_0^t K(\tilde{\mathbf{a}}(t-s),s)ds. \tag{6.29}$$

The value of the memory at $s=0$ can be expressed as,

$$\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t),0) = \int_\Omega \tilde{\mathbf{w}}\nabla\cdot\mathbf{F}'(\mathbf{q})d\Omega + \int_\Gamma \tilde{\mathbf{w}}\mathbf{b}'(\mathbf{q},\mathbf{n})d\Gamma, \tag{6.30}$$

where $\mathbf{q}$ is given by,

$$\int_\Omega \mathbf{w}'\mathbf{q}d\Omega = \int_\Omega \mathbf{w}'\nabla\cdot\mathbf{F}(\tilde{\mathbf{u}})d\Omega + \int_\Gamma \mathbf{w}'\mathbf{b}(\tilde{\mathbf{u}},\mathbf{n})d\Gamma. \tag{6.31}$$

The term $\mathbf{F}' = \frac{\partial\mathbf{F}}{\partial\tilde{\mathbf{u}}}$ is the flux Jacobian and $\mathbf{b}'$ is the numerical flux function linearized about $\tilde{\mathbf{u}}$. The resulting coarse-scale equation for the τ-model is

$$\int_\Omega \tilde{\mathbf{w}}\mathbf{u}_t d\Omega + \int_\Omega \tilde{\mathbf{w}}\nabla\cdot\big(\mathbf{F}(\tilde{\mathbf{u}}) - \tau\mathbf{F}'(\mathbf{q})\big)d\Omega + \int_\Gamma \tilde{\mathbf{w}}\big(\mathbf{b}(\tilde{\mathbf{u}},\mathbf{n}) - \tau\mathbf{b}'(\mathbf{q},\mathbf{n})\big)d\Gamma = 0. \tag{6.32}$$

Consider now Eq. 6.15 augmented with an artificial viscosity term that is proportional to the orthogonal projection of the divergence of the flux,

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot\mathbf{F} = \tau\nabla\cdot\mathbf{F}'(\Pi'\nabla\cdot\mathbf{F}). \tag{6.33}$$

A standard discretization technique for this second order equation is to split it into two first order equations [4],

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot\big(\mathbf{F}(\mathbf{u}) - \mathbf{F}'(\mathbf{q})\big) = 0, \tag{6.34}$$

with,

$$\mathbf{q} = \Pi' \nabla \cdot \mathbf{F}(\mathbf{u}). \tag{6.35}$$

Assuming that the boundary operators are handled analogously, the discretization of Eq. 6.34 and Eq. 6.35 through finite elements leads to precisely Eqns. 6.31 and 6.32.

Remarks

1. For a hyperbolic conservation law, the memory is driven by a non-linear term that acts as a type of non-linear artificial viscosity.
2. The magnitude of the artificial dissipation is proportional to the projection of the flux onto the fine scales. If the flux term is fully resolved, no information is added to the memory.
3. Due to the appearance of the orthogonal projector, it is difficult to comment on the sign of the artificial viscosity. While proofs exist showing that the term $e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$ is globally dissipative in certain settings [46], no such result is readily apparent in the general case.

6.3.2 Conservation Properties

Consider the spatially discretized set of conservation laws, Eq. 6.17, written in terms of the modal basis coefficients $\mathbf{a}$,

$$\frac{d\tilde{\mathbf{a}}^i}{dt} = \mathbf{f}^i(\tilde{\mathbf{a}}, \mathbf{a}'), \tag{6.36}$$

$$\frac{d\mathbf{a}'^i}{dt} = \mathbf{g}^i(\tilde{\mathbf{a}}, \mathbf{a}'), \tag{6.37}$$

for $i = \{0, 1, ..., L\}$ where $\mathbf{a} = \{\tilde{\mathbf{a}}, \mathbf{a}'\} = \{\tilde{\mathbf{a}}_0, \tilde{\mathbf{a}}_1, ..., \mathbf{a}'_0, \mathbf{a}'_1, ...\}$. The i_{th} component refers to the i_{th} conservation equation. For example, in the Euler equations $\mathbf{a}^1$

would consist of the modal coefficients for ρU. In Section 4.1.1 we presented a proof originating from Stinis showing that for systems that conserve discrete energy, defined by $E = \sum a_i a_i^*$, the t-model, τ-model, and VMS(ϵ) models dissipate energy. Here we extend Stinis' proof to make it applicable to DG and consider mass and energy conservation properties of the MZ-VMS models.

Theorem VI.1. *Suppose the full order model satisfies*

$$\frac{d\mathcal{M}^i}{dt} = 0, \tag{6.38}$$

where $\mathcal{M}_i$ is the mass of the i^{th} conserved variable,

$$\mathcal{M}^i = \sum_k \int_{\Omega_k} u^i d\Omega = 0. \tag{6.39}$$

Then the mass of the i^{th} conserved variable evolved by the τ-model is given by,

$$\frac{d\mathcal{M}^{i,\tau}}{dt} = 0. \tag{6.40}$$

Thus, if the τ-model is constructed from a method that is conservative, the τ-model will be conservative. The same results extend to the t and VMS(ϵ) models.

Proof. The total mass of the i_{th} variable is defined by,

$$\mathcal{M}^i = \sum_k \int_{\Omega_k} u^i d\Omega = \sum_k \int_{\Omega_k} \sum_j \mathbf{a}^i_{jk} w_{jk}(\mathbf{x}) d\Omega. \tag{6.41}$$

By definition of the Legendre polynomials,

$$a_0 = \frac{1}{2} \int_{-1}^{1} u(\zeta) w_0(\zeta) d\zeta,$$

where $w_0(\zeta) = 1$. We can use this property to write Eq. 6.41 as

$$\mathcal{M}^i = \sum_k \int_{\Omega_k} u^i d\Omega = 2 \sum_k |J_k| \mathbf{a}^i_{0k}, \tag{6.42}$$

where $|J_k|$ is the determinant of the Jacobian that provides the map between the global coordinates $\mathbf{x}$ and local coordinates η. The evolution of the rate of change of total mass is obtained by multiplying the equations for $\tilde{a}^i_{0k}$ by $|J_k|$ and summing,

$$\frac{d}{dt}\mathcal{M}^i = 2\sum_k |J_k| f^i_{0k}(\tilde{\mathbf{a}}, \mathbf{a}') = 0. \tag{6.43}$$

As Eq. 6.43 holds for any $\mathbf{a}$, set $\mathbf{a}' = 0$,

$$\sum_k |J_k| f^i_{0k}(\tilde{\mathbf{a}}, 0) = 0. \tag{6.44}$$

Next, set $\mathbf{a}' = \epsilon\mathbf{h}$,

$$\sum_k |J_k| f^i_{0k}(\tilde{\mathbf{a}}, \epsilon\mathbf{h}) = 0. \tag{6.45}$$

Proceed by expanding in a Taylor series,

$$\sum_k |J_k| f^i_{ok}(\tilde{\mathbf{a}}, 0) + \epsilon\sum_k |J_k| \frac{\partial f^i_{0k}}{\partial \mathbf{a}'}\mathbf{h} + \mathcal{O}(\epsilon^2) = 0. \tag{6.46}$$

Noting that the first term on the left hand side is zero by Eq. 6.44,

$$\epsilon\sum_k |J_k| \frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'}\mathbf{h} + \mathcal{O}(\epsilon^2) = 0. \tag{6.47}$$

Proceed by setting $\mathbf{h} = \mathbf{g}(\tilde{\mathbf{a}}, 0)$, dividing by ϵ, and letting $\epsilon \to 0$ to obtain

$$\sum_k |J_k| \frac{\partial f^i_{0k}}{\partial \mathbf{a}'}\mathbf{g}(\tilde{\mathbf{a}}, 0) = 0. \tag{6.48}$$

The τ-model for the i^{th} variable in ODE system given by Eqns. 6.36 and 6.37 is written as

$$\frac{d\tilde{\mathbf{a}}^i}{\partial t} = \mathbf{f}^i(\tilde{\mathbf{a}}, 0) + \tau\frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'}g(\tilde{\mathbf{a}}, 0). \tag{6.49}$$

The evolution of the total mass evolved by the τ-model is given by,

$$\frac{d\mathcal{M}^{i,\tau}}{dt} = 2\sum_k |J_k| f^i_{0k}(\tilde{\mathbf{a}}, 0) + 2\tau\sum_k |J_k| \frac{\partial f^i_{0k}}{\partial \mathbf{a}'}\mathbf{g}(\tilde{\mathbf{a}}, 0). \tag{6.50}$$

By Eqns. 6.44 and 6.48 the terms on the right hand side drop and we have

$$\frac{d\mathcal{M}^{i,\tau}}{dt} = 0. \tag{6.51}$$

This result shows that for a discretization that is conservative for the i_{th} variable, the τ-model, t-model, and VMS(ϵ) models will be conservative as well. □

6.3.3 Energy Dissipation

The same style of proof can be used to examine the dissipative nature of the t, τ, and VMS(ϵ) models.

Theorem VI.2. *Suppose the full order model satisfies,*

$$\frac{d\mathcal{J}^{i}}{dt} = 0, \tag{6.52}$$

where $\mathcal{J}^i$ is the energy in the i_{th} variable as defined by the L^2 norm,

$$\mathcal{J}^i = \frac{1}{2}\sum_k \int_{\Omega_k} (u^i)^2 d\Omega.$$

For L^2 orthogonal polynomials, the energy of the i^{th} conserved variable evolved by the τ-model is then given by,

$$\frac{d\mathcal{J}^{i,\tau}}{dt} \leq 0. \tag{6.53}$$

Thus, if the τ-model is constructed from a method that is energy conservative, the τ-model will be dissipative. The same results extend to the t and VMS(ϵ) models.

Proof. The energy in the i_{th} variable is,

$$\mathcal{J} = \frac{1}{2}\sum_k \int_{\Omega_k} (u^i)^2 d\Omega = \frac{1}{2}\sum_k \int_{\Omega_k} \big(\sum_j \mathbf{a}_{jk} w_{jk}(\mathbf{x})\big)^2 d\Omega.$$

Define the inner product,

$$(\mathbf{a}, \mathbf{b}) = \sum_k \int_{\Omega_k} \big(\sum_j a_j w_{jk}(\mathbf{x})\big)\big(\sum_j b_j w_{jk}(\mathbf{x})\big) d\Omega. \tag{6.54}$$

For L^2 orthogonal polynomials,

$$\int w_{jk}(\mathbf{x}) w_{ik}(\mathbf{x}) d\mathbf{x} = c_{jk}\delta_{ij}. \tag{6.55}$$

Making use of Eq. 6.55, Eq. 6.54 can be written as

$$(\mathbf{a}, \mathbf{b}) = \sum_k \int_{\Omega_k} \big(\sum_j a_j w_{jk}(\mathbf{x}) \big) \big(\sum_j b_j w_{jk}(\mathbf{x}) \big) d\Omega = \sum_k \sum_j c_{jk} a_{jk} b_{jk}. \tag{6.56}$$

This shows,

$$(\mathbf{a} + \mathbf{b}, \mathbf{c} + \mathbf{d}) = (\mathbf{a}, \mathbf{c}) + (\mathbf{b}, \mathbf{d}). \tag{6.57}$$

Using Eq. 6.57, the evolution of the total energy in the i_{th} variables is given by,

$$\frac{d\mathcal{J}}{dt} = \big(\mathbf{a}^i, \frac{d\mathbf{a}^i}{dt}\big) = \big(\tilde{\mathbf{a}}^i, \mathbf{f}^i(\tilde{\mathbf{a}}, \mathbf{a}')\big) + \big(\mathbf{a}'^i, \mathbf{g}^i(\tilde{\mathbf{a}}, \mathbf{a}')\big) = 0. \tag{6.58}$$

Setting the unresolved variables to zero shows,

$$\big(\tilde{\mathbf{a}}^i, \mathbf{f}^i(\tilde{\mathbf{a}}, 0)\big) = 0. \tag{6.59}$$

Now taking $\mathbf{a}' = \epsilon \mathbf{h}$,

$$\big(\tilde{\mathbf{a}}^i, \mathbf{f}^i(\tilde{\mathbf{a}}, \epsilon\mathbf{h})\big) + \big(\epsilon\mathbf{h}^i, \mathbf{g}^i(\tilde{\mathbf{a}}, \epsilon\mathbf{h})\big) = 0. \tag{6.60}$$

Expanding in a Taylor series gives,

$$\big(\tilde{\mathbf{a}}^i, \mathbf{f}^i(\tilde{\mathbf{a}}, 0)\big) + \epsilon\big(\tilde{\mathbf{a}}^i, \frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'}\mathbf{h}\big) + \epsilon\big(\mathbf{h}^i, \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\big) + \mathcal{O}(\epsilon^2) = 0. \tag{6.61}$$

Dividing by ϵ, letting $\epsilon \to 0$, and setting $\mathbf{h} = \mathbf{g}(\tilde{\mathbf{a}}, 0)$ gives,

$$\big(\tilde{\mathbf{a}}^i, \frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'}\mathbf{g}(\tilde{\mathbf{a}}, 0)\big) = -\big(\mathbf{g}^i(\tilde{\mathbf{a}}, 0), \mathbf{g}^i(\tilde{\mathbf{a}}, 0)\big) \tag{6.62}$$

The τ for the i_{th} semi-discrete equation is written as,

$$\frac{d\tilde{\mathbf{a}}^i}{\partial t} = \mathbf{f}^i(\tilde{\mathbf{a}}, 0) + \tau \frac{\partial \mathbf{f}^i}{\partial \mathbf{a}'} g(\tilde{\mathbf{a}}, 0). \tag{6.63}$$

Taking the inner product of Eq. 6.63 with $\tilde{\mathbf{a}}^i$ and inserting Eq. 6.62,

$$\frac{d\tilde{\mathcal{J}}}{\partial t} = -\tau\big(\mathbf{g}^i(\tilde{\mathbf{a}},0),\mathbf{g}^i(\tilde{\mathbf{a}},0)\big) \leq 0. \tag{6.64}$$

Equation 6.64 demonstrates that, for a numerical method conserving energy for the i_{th} variable, the MZ-VMS model will dissipate energy for that variable. □

6.4 The MZ-VMS DG Flux Function

Recall that in the FEM case the memory is driven by,

$$\begin{aligned}\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t),0) = &\int_\Omega\int_\Omega(\tilde{\mathbf{w}}\mathcal{R}')(x)\Pi'(x,y)(\mathcal{R}(\tilde{u})-f)(y)d\Omega_y d\Omega_x\\ &+\int_\Omega\int_\Gamma(\tilde{\mathbf{w}}\mathcal{R}')(x)\Pi'(x,y)b(\tilde{u}(y))d\Gamma_y d\Omega_x\\ &+\int_\Gamma\int_\Omega(\tilde{\mathbf{w}}b')(x)\Pi'(x,y)(\mathcal{R}(\tilde{u})-f)(y)d\Gamma_y d\Omega_x\\ &+\int_\Gamma\int_\Gamma(\tilde{\mathbf{w}}b')(x)\Pi'(x,y)b(\tilde{u}(y))d\Gamma_y d\Gamma_x,\end{aligned} \tag{6.65}$$

where again

$$\mathcal{R}' = \frac{\partial\mathcal{R}}{\partial\tilde{u}}, \qquad b' = \frac{\partial b}{\partial\tilde{u}}. \tag{6.66}$$

The final term in the above expression is in the form of a flux function. Models such as the t and τ-model will naturally include a flux function. Here we investigate the form of this function.

6.4.1 One Dimensional Case

The flux contribution of the memory term for the discontinuous Galerkin discretization of a hyperbolic conservation equation in one-dimension is considered. The derivation is pursued using Legendre polynomials for the reasons previously discussed. The hyperbolic conservation equation we consider is given by,

$$\frac{\partial\mathbf{u}}{\partial t} + \frac{\partial\mathbf{f}}{\partial x} = 0, \tag{6.67}$$

The discontinuous Galerkin discretization of Eq. 6.67 leads to the following weak formulation,

$$\int_{\Omega_k} w\frac{\partial \mathbf{u}}{\partial t}d\Omega - \int_{\Omega_k}\frac{\partial w}{\partial x}\mathbf{f}(\mathbf{u})d\Omega = -\int_{\Gamma_k} w\mathbf{f}^*(\mathbf{u},\mathbf{n})d\Gamma, \tag{6.68}$$

where $\mathbf{f}^*$ is a numerical flux function that provides coupling between the elements. Separating scales, we can express Eq. 6.68 by,

$$\begin{aligned}\int_{\Omega_k} \tilde{w}\frac{\partial \tilde{\mathbf{u}}}{\partial t}d\Omega - \int_{\Omega_k}\frac{\partial \tilde{w}}{\partial x}f(\tilde{\mathbf{u}})d\Omega - \int_{\Omega_k}\frac{\partial \tilde{w}}{\partial x}\big(\mathbf{f}(\mathbf{u}) - f(\tilde{\mathbf{u}})\big)d\Omega = \\ -\int_{\Gamma_k}\tilde{w}f^*(\tilde{\mathbf{u}},\mathbf{n})d\Gamma - \int_{\Gamma_k}\tilde{w}\big(f^*(\mathbf{u},\mathbf{n}) - f^*(\tilde{\mathbf{u}},\mathbf{n})\big)d\Gamma.\end{aligned} \tag{6.69}$$

$$\begin{aligned}\int_{\Omega_k} w'\frac{\partial \tilde{u}}{\partial t}d\Omega - \int_{\Omega_k}\frac{\partial w'}{\partial x}f(\tilde{u})d\Omega - \int_{\Omega_k}\frac{\partial w'}{\partial x}\big(f(u) - f(\tilde{u})\big)d\Omega = \\ -\int_{\Gamma_k}w'f^*(\tilde{u},\mathbf{n})d\Gamma - \int_{\Gamma_k}w'\big(f^*(u,\mathbf{n}) - f^*(\tilde{u},\mathbf{n})\big)d\Gamma.\end{aligned} \tag{6.70}$$

Note that we have expressed the fine-scale equation, Eq. 6.70, in the DG strong form. Through the MZ-VMS framework, we can express the coarse-scale equation as,

$$\int_{\Omega_k}\tilde{w}\frac{\partial \tilde{u}}{\partial t}d\Omega - \int_{\Omega_k}\frac{\partial \tilde{w}}{\partial x}f(\tilde{u})d\Omega = -\int_{\Gamma_k}\tilde{w}f^*(\tilde{u},n)d\Gamma + \tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t),0), \tag{6.71}$$

where $\tilde{\mathbf{M}}$ is the coarse-scale mass matrix. We are interested in the flux portion of the memory term, which is given by,

$$\tilde{\mathbf{M}}K_f(\tilde{\mathbf{a}}(t),0) = \int_\Gamma\int_\Gamma(\frac{\partial b^*}{\partial \tilde{u}}\tilde{\mathbf{w}})(x)\Pi'(x,y)b(\tilde{u}(y))d\Gamma_y d\Gamma_x, \tag{6.72}$$

where b is the boundary operator. The boundary operators on element k correspond to the flux functions,

$$\int_{\Gamma_k} b_k(u)d\Gamma = -\big[f(u_k^R) - f^*(u_k^R, u_{k+1}^L)\big] + \big[f(u_k^L) - f^*(u_{k-1}^R, u_k^L)\big] \tag{6.73}$$

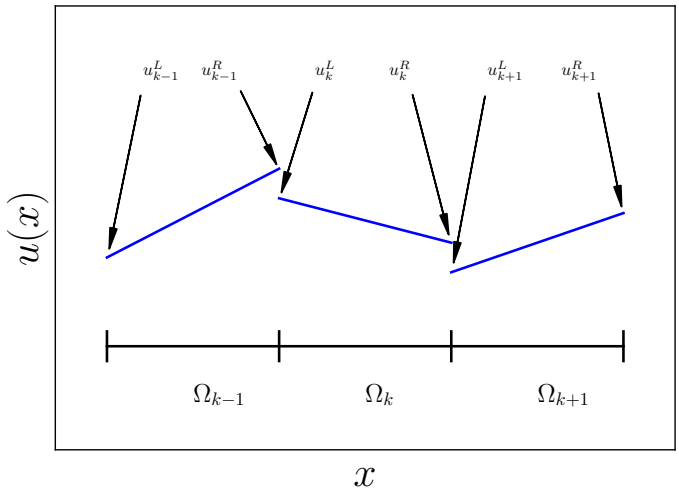


Figure 6.2: Diagram of linear discontinuous solution to demonstrate notation associated with the boundary operators.

where f^* is a numerical flux function and u_k^L and u_k^R correspond to the value of u on the left and right boundary of the k_{th} element (see Figure 6.2). Similarly we have,

$$\int_{\Gamma_k} b_k^*(u)d\Gamma = f^*(u_k^R, u_{k+1}^L) - f^*(u_{k-1}^R, u_k^L). \tag{6.74}$$

For notational purposes we denote,

$$\Delta f_k^R = f(u_k^R) - f^*(u_k^R, u_{k+1}^L), \qquad \Delta f_k^L = f(u_k^L) - f^*(u_{k-1}^R, u_k^L). \tag{6.75}$$

Evaluating the integral over Γ_y,

$$\tilde{\mathbf{M}}K_k(\tilde{\mathbf{a}}(t), 0) = -\int_{\Gamma_k} (\frac{\partial b_k^*}{\partial \tilde{u}}\tilde{\mathbf{w}}_k)(x)\mathbf{w'}_k(x)^T\mathbf{M'}^{-1}\left[\mathbf{w'}_k^R\Delta f_k^R - \mathbf{w'}_k^L\Delta f_k^L\right]d\Gamma_x. \tag{6.76}$$

Evaluation of the integral of Γ_x then yields the expression,

$$\begin{aligned}(6.77)\qquad \tilde{\mathbf{M}}K_k(\tilde{\mathbf{a}}(t),0) = &-\tilde{\mathbf{w}}_k^R\frac{\partial f^*}{\partial \tilde{u}}\Big(\mathbf{w'}_k^{R^T}\mathbf{M'}^{-1}\big[\mathbf{w'}_k^R\Delta f_k^R - \mathbf{w'}_k^L\Delta f_k^L\big],\\ &\qquad \mathbf{w'}_{k+1}^{L^T}\mathbf{M'}^{-1}\big[\mathbf{w'}_{k+1}^R\Delta f_{k+1}^R - \mathbf{w'}_{k+1}^L\Delta f_{k+1}^L\Big)\\ &+\tilde{\mathbf{w}}_k^L\frac{\partial f^*}{\partial \tilde{u}}\Big(\mathbf{w'}_{k-1}^{R^T}\mathbf{M'}^{-1}\big[\mathbf{w'}_{k-1}^R\Delta f_{k-1}^R - \mathbf{w'}_{k-1}^L\Delta f_{k-1}^L\big],\\ &\qquad \mathbf{w'}_k^{L^T}\mathbf{M'}^{-1}\big[\mathbf{w'}_k^R\Delta f_k^R - \mathbf{w'}_k^L\Delta f_k^L\big]\Big).\end{aligned}$$

This can be expressed compactly as,

$$\begin{aligned}(6.78)\qquad \tilde{\mathbf{M}}K_k(\tilde{\mathbf{a}}(t),0) = &-\tilde{\mathbf{w}}_k^R\frac{\partial f^*}{\partial \tilde{u}}\Big(S_1\Delta f_k^R - S_2\Delta f_k^L, S_2\Delta f_{k+1}^R - S_1\Delta f_{k+1}^L\Big)\\ &+\tilde{\mathbf{w}}_k^L\frac{\partial f^*}{\partial \tilde{u}}\Big(S_1\Delta f_{k-1}^R - S_2\Delta f_{k-1}^L, S_2\Delta f_k^R - S_1\Delta f_k^L\Big),\end{aligned}$$

where S_1 and S_2 are scalars given by,

$$(6.79)\qquad S_1 = \mathbf{w'}^{R^T}\mathbf{M'}^{-1}\mathbf{w'}^R, \qquad S_2 = \mathbf{w'}^{R^T}\mathbf{M'}^{-1}\mathbf{w'}^L.$$

Note that for the Legendre polynomials we have,

$$(6.80)\qquad \mathbf{w'}^{L^T}\mathbf{M'}^{-1}\mathbf{w'}^L = \mathbf{w'}^{R^T}\mathbf{M'}^{-1}\mathbf{w'}^R = S_1.$$

Further, for Legendre polynomials one will note that the ratio $\frac{S_2}{S_1} \approx \frac{1}{N}$, where N is the number of basis functions. In the limit that the fine scales have infinite support ($N \to \infty$), we may drop the terms involving S_2 to have,

(6.81)

$$\tilde{\mathbf{M}}K_k(\tilde{\mathbf{a}}(t),0) = -\tilde{\mathbf{w}}_k^R\frac{\partial f^*}{\partial \tilde{u}}\Big(S_1\Delta f_k^R, -S_1\Delta f_{k+1}^L\Big) + \tilde{\mathbf{w}}_k^L\frac{\partial f^*}{\partial \tilde{u}}\Big(S_1\Delta f_{k-1}^R, -S_1\Delta f_k^L\Big).$$

It is seen that the memory is dependent on the numerical flux function. The choice of numerical flux functions will hence impact the final form of the MZ model. As flux functions are often specific to certain physical problems and are themselves often

implicit subgrid models, only the symmetric central flux is examined here. Problem specific fluxes will be examined in more detail later in this chapter.

The central flux is given by an arithmetic average of the left and right states,

$$f^*(u_k^R, u_{k+1}^L) = \frac{1}{2}(f_k^R + f_{k+1}^L). \tag{6.82}$$

Note that for the central flux we have

$$\Delta f_k^R = \frac{1}{2}(f_k^R - f_{k+1}^L), \qquad \Delta f_k^L = \frac{1}{2}(f_k^L - f_{k-1}^R). \tag{6.83}$$

Further,

$$\frac{\partial f^*}{\partial \tilde{u}}(\Delta f_k^R, -\Delta f_{k+1}^L) = \frac{\mathbf{A}_k^R + \mathbf{A}_{k+1}^L}{4}(f_k^R - f_{k+1}^L), \tag{6.84}$$

$$\frac{\partial f^*}{\partial \tilde{u}}(\Delta f_{k-1}^R, -\Delta f_k^L) = \frac{\mathbf{A}_{k-1}^R + \mathbf{A}_k^L}{4}(f_{k-1}^R - f_k^L). \tag{6.85}$$

We see that Eq. 6.81 reduces to

(6.86)

$$\tilde{\mathbf{M}} K_k(\tilde{\mathbf{a}}(t), 0) = -S_1 \tilde{\mathbf{w}}_k^R \frac{\mathbf{A}_k^R + \mathbf{A}_{k+1}^L}{4}(f_k^R - f_{k+1}^L) + S_1 \tilde{\mathbf{w}}_k^L \frac{\mathbf{A}_{k-1}^R + \mathbf{A}_k^L}{4}(f_{k-1}^R - f_k^L).$$

In models such as the t-model and τ model, this term leads to adding a numerical flux correction at an interface of the form

$$\mathbf{f}_{MZ}(\mathbf{u}_k^R, \mathbf{u}_{k+1}^L) = \tau^* \frac{\mathbf{A}_k^R + \mathbf{A}_{k+1}^L}{4}(\mathbf{f}_k^R - \mathbf{f}_{k+1}^L), \tag{6.87}$$

with $\tau^* = \tau S_1$. Throughout the remainder of this chapter flux functions for a variety of systems will be considered.

6.5 Application to Linear Hyperbolic Systems

We first consider the case where Eq. 6.67 is a linear hyperbolic equation that can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{u}}{\partial x} = 0, \tag{6.88}$$

where $\mathbf{A}$ is the flux Jacobian, $\mathbf{f_u}$, and is constant in space. In the linear case, the first three terms of Eq. 6.65 will drop and the only the MZ-VMS flux function remains. In this case the numerical flux function that emerges is,

$$\mathbf{f}_{MZ}(\mathbf{u}_k^R, \mathbf{u}_{k+1}^L) = \tau^* \frac{\mathbf{A^2}}{2}(\mathbf{u}_k^R - \mathbf{u}_{k+1}^L). \tag{6.89}$$

Note we have used the identity $\mathbf{f} = \mathbf{Au}$ to obtain Eq. 6.89. A natural selection for the timescale is,

$$\tau^* = \mathbf{S}|\mathbf{\Lambda}^{-1}|\mathbf{S}^{-1},$$

where $\mathbf{S}$ and Λ are the eigenvectors and eigenvalues arising from an eigendecomposition of $\mathbf{A}$. This choice leads to the numerical flux function,

$$\mathbf{f}_{MZ}(\mathbf{u}_k^R, \mathbf{u}_{k+1}^L) = \frac{|\mathbf{A}|}{2}(\mathbf{u}_k^R - \mathbf{u}_{k+1}^L), \tag{6.90}$$

which is precisely flux correction present for an upwind solver. A large body of work exists on upwind flux schemes for linear hyperbolic equations, and we conclude our discussion for such systems here.

6.5.1 Fine-scale Eigenvalues and Finite Memory

In Section 3.3.3, it was discussed how the eigenvalues of the fine-scale operator can be related to the memory length. The eigendecomposition of the fine-scale operator of the advection diffusion equation is performed to demonstrate this. The advection diffusion equation is given by,

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}, \tag{6.91}$$

on a periodic domain of $x \in [0, 2\pi)$. Figure 6.3 shows the *minimum negative eigenvalue* for the fine-scale operator of the advection diffusion equation for a variety of Reynolds numbers. The inverse of the eigenvalue will correspond to the longest

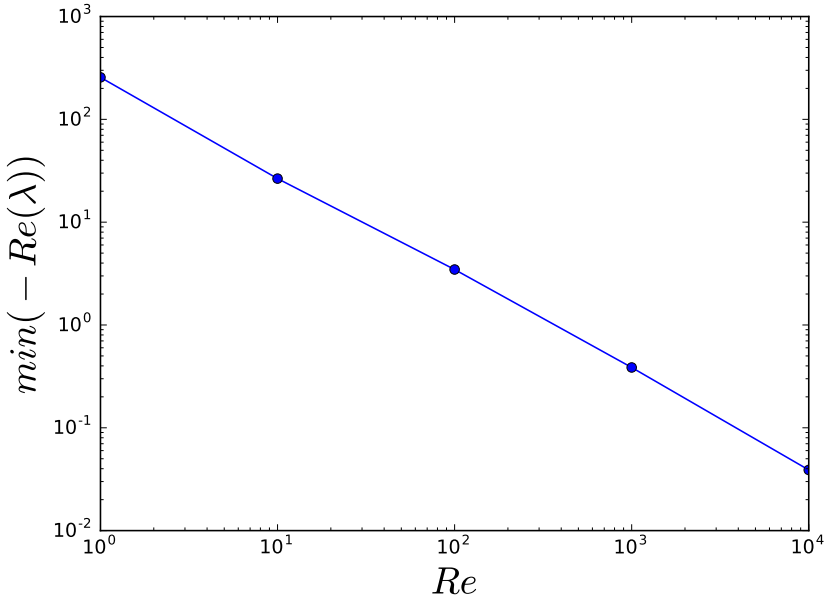


Figure 6.3: Impact of Reynolds number on the fine-scale eigenvalues.

timescale. The eigenvalues are computed for a coarse-scale discretization using $N = 8$ elements and $p = 1$ polynomials. The fine scales use a discretization with $N = 8$ elements and $p = 32$ polynomials. In Figure 6.3, the minimum eigenvalue is seen to obey a power law with respect to the Reynolds number. As the Reynolds number increases, the minimum eigenvalue of the fine-scale operator decreases. This corresponds to an increase in memory length.

The impact of Reynolds number on the memory length can be further demonstrated by directly extracting the memory kernel. Note that the advection diffusion equation is linear, and hence the fine scales can be exactly removed by virtue of integrating factors (eg. see Chapter III). We consider an initial condition $u(x, 0) = \sin(x)$ on a domain with $N = 8$ elements at polynomial order $p = 1$. The fine scales were computed using an eighth order discretization. Figure 6.4 shows the memory integrand for the a_0 mode on the first element in the domain at $t = 2$ for various

Reynolds numbers. The integrand, denoted by the shaded yellow region, is seen to have a longer support in pseudo-time as the Reynolds number increases. This increase in memory length approximately follows the scaling seen in Figure 6.3.

6.6 Application to Burgers' Equation

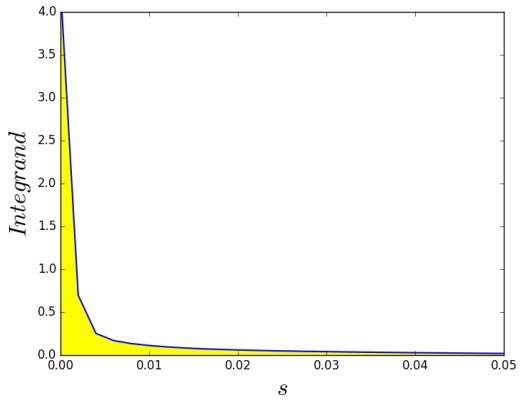
Burgers' equation is now considered. Burgers' equation is significantly different from linear advection in that a smooth solution can develop discontinuities. This can lead to challenges in the construction of a stable reduced model. Burgers' equation displays many of the fundamental issues that will be encountered in the Navier-Stokes. In particular, this section will pay attention to stability for non-linear problems.

6.6.1 Kinetic Energy and Entropy

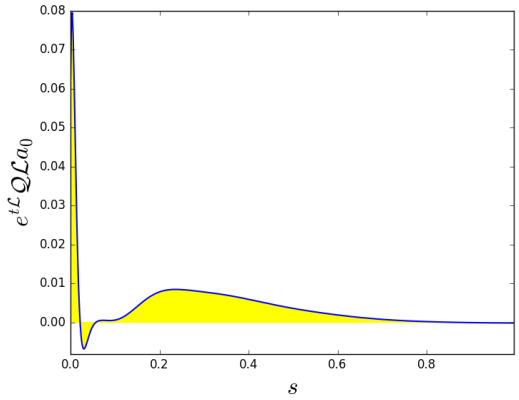
To construct a stable, accurate MZ-VMS model for Burgers' equation, first consider the relationship between kinetic energy and entropy for Burgers' equation. Recall that the Burgers' equation is given by

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0. \tag{6.92}$$

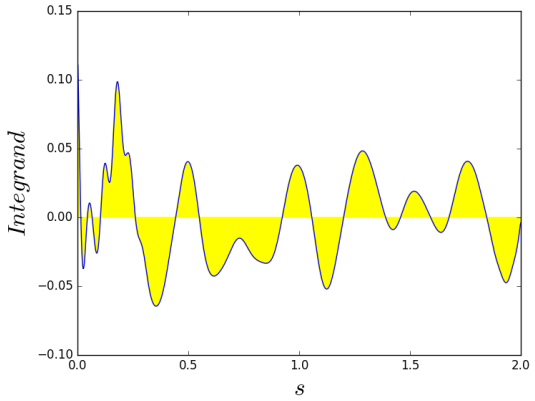
For Burgers' equation, we can guarantee that the t, τ, and VMS(ϵ) models will be dissipative if the original discretization is kinetic energy conserving. The kinetic energy can further be shown to be an entropy variable for Burgers' equation. A numerical entropy variable can be defined by a scalar convex function [96, 97] U such that a system of hyperbolic conservation equations can be mapped into a single scalar conservation equation for entropy. For Burgers' equation, the function $U = u^2$ meets these requirements. For smooth solutions, the entropy throughout the domain



(a) Re=1



(b) Re=10



(c) Re=100

Figure 6.4: Plot of the memory integrand vs the pseudo-time in the convolution, s for various Reynolds numbers.

remains constant, i.e.

$$\frac{d}{dt}\int_{\Omega} u^2 d\Omega = 0.$$

For square entropy functions, such as in the case of Burgers' equation, discontinuous Galerkin methods satisfy cell entropy inequalities for scalar and symmetric hyperbolic systems [17]. Provided an entropy conservative flux scheme is used, DG methods will be entropy conservative for these types of systems. With the numerical entropy being twice the kinetic energy for Burgers' equation, it directly follows that for an entropy conservative scheme the t, τ, and VMS(ϵ) models will be entropy stable. For a scheme that dissipates entropy (and is hence entropy stable), no such conclusion can be drawn. It is important to note that, while for Burgers' equation the kinetic energy is an entropy function, this will not be the case for the Navier-Stokes equations.

6.6.2 Flux Functions for Burgers' Equation

Three flux functions are considered for Burgers' equation. The first is a simple central flux,

$$\hat{F}(u^+, u^-, \hat{n}) = \frac{1}{4}\big(u^{+2} + u^{-2}\big). \tag{6.93}$$

The central flux can be shown to be an entropy stable flux for Burgers' equation. In addition to the central flux, an entropy conservative flux is considered [53],

$$\hat{F}(u^+, u^-, \hat{n}) = \frac{1}{6}\big(u^{+2} + u^+u^- + u^{-2}\big). \tag{6.94}$$

We can guarantee stability of the t, τ, and VMS(ϵ) models for the entropy conservative flux due to the analysis in Section 4.1.1. The last flux considered is the standard upwind flux, which is given by

$$\hat{F}(u^+, u^-, \hat{n}) = \frac{1}{4}\big(u^{+2} + u^{-2}\big) - \frac{1}{2}|u^*|\big(u^+ - u^-\big), \tag{6.95}$$

where u^* is the average state. Of the fluxes considered, only the entropy conservative flux is guaranteed to produce stable MZ-VMS models.

6.6.3 MZ-VMS Flux Functions for the Burgers' Equation

Each numerical flux scheme will have a corresponding MZ-VMS flux correction. For simple fluxes in one dimension, these MZ-VMS flux functions can be derived. This is shown for a central flux and for the entropy stable flux.

Central Flux

For the central flux, we can directly use the form derived in Eq. 6.87. For Burgers' equation, the flux Jacobian (which in this case is given by u) is non-constant in space, and the general form of the MZ-VMS flux is,

$$f_{MZ}(f_k^R, f_{k+1}^L) = \tau^* \frac{u_k^R + u_{k+1}^L}{4}(f_k^R - f_{k+1}^L). \tag{6.96}$$

A natural selection for the timescale in the τ model is the arithmetic average of the wave speed,

$$\tau = \frac{1}{S_1 \lambda_{\text{Roe}}} = \frac{2}{S_1 |u_k^R + u_{k+1}^L|}.$$

This selection leads to the Roe flux correction,

$$f_{MZ}(f_k^R, f_{k+1}^L) = \frac{|u_k^R + u_{k+1}^L|}{2}(u_k^R - u_{k+1}^L). \tag{6.97}$$

Entropy Stable Flux

For Burgers' equation, another interesting case to examine is where the original flux is the entropy conserving flux (Eq. 6.94). In this case, the MZ-VMS flux function becomes,

$$f_{MZ}(u_k^R, u_{k+1}^L) = \frac{\tau^*}{18}(5u_L^2 + 8u_L u_R + 5u_R^2). \tag{6.98}$$

The corresponding flux function is

$$f(u_k^R, u_{k+1}^L) = \frac{1}{6}(u_L^2 + u_L u_R + u_R^2) - \frac{\tau^*}{18}(5u_L^2 + 8u_L u_R + 5u_R^2). \tag{6.99}$$

For all $\tau > 0$, Eq. 6.99 is an entropy stable flux function. The choice of τ controls the amount of dissipation added to the system.

6.6.4 Numerical Applications

We consider the solution of the inviscid Burgers' equation with MZ-VMS models constructed on the above fluxes. Burgers' equation is solved on a periodic domain with $x \in [0, 2\pi)$.

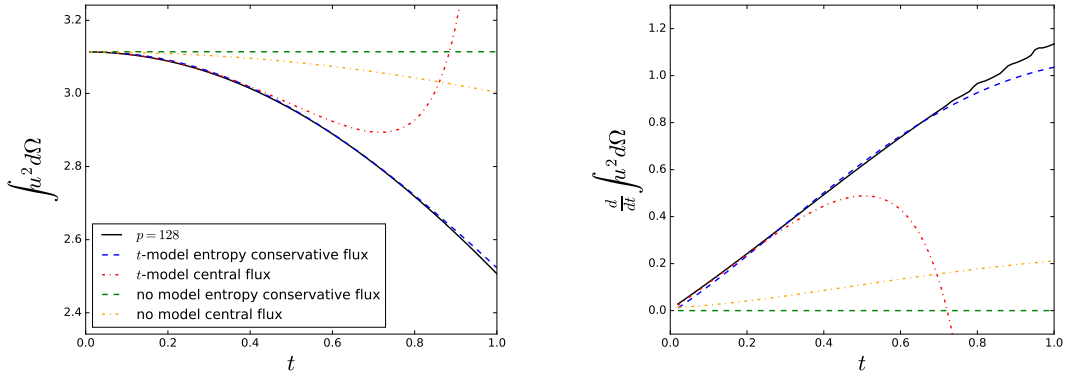
Effect of the numerical flux function

We first examine the impact of the numerical flux function on the resulting MZ-VMS model. Here only the t-model is considered, as the functional form of the model is simple and it has no constants. We first examine the initial condition

$$u(x, 0) = \sin(x),$$

which leads to a standing shock at $t = 1$.

Figure 6.5 shows results of simulations using one element and a third order polynomial (four total DOFs) for $t \leq 1$. The results are compared to a projected solution that was obtained using $p = 127$. Note that the $p = 127$ solution starts to display oscillations close to $t = 1$. Before the shock occurs, the total entropy remains constant. The entropy in the resolved variables, however, does not remain constant. In Figure 6.5, one can see that simulations run using an entropy conservative flux and central flux under-predict the dissipation in resolved entropy. The entropy conservative flux leads to no net decrease in entropy, while the central flux leads to a slight



(a) Temporal evolution of resolved entropy

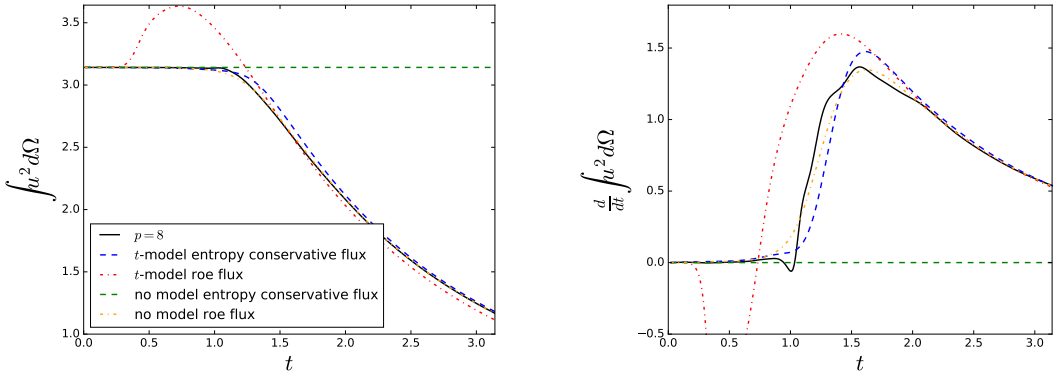
(b) Temporal evolution of the rate of change of resolved entropy

Figure 6.5: Numeric solutions to the Burgers' equation using 1 element with $p = 3$.

decrease in entropy. The results of the Roe scheme are not shown in Figure 6.5 as they are comparable to the central flux.

Figure 6.5 shows the importance of the flux function for the MZ-VMS models. It is seen that the t-model constructed from the central flux rapidly goes unstable. This instability is not surprising since the central flux does not guarantee entropy conservation and hence it is possible for the MZ-VMS model to add entropy to the system. The t-model constructed from an entropy conservative flux provides a stable and accurate solution.

Next, we consider the simulations integrated through the shock wave. The shock wave can not be fully resolved and leads to a decay in entropy. Figure 6.6 shows the results of the simulations for $N = 16$ elements with $p = 1$ polynomials. The solutions are compared to a high-order solution using $N = 16$ elements and $p = 7$ polynomials. In this case we compare t-models constructed from the Roe flux and the entropy stable flux. It is again seen that the t-model constructed from the entropy stable flux provides a stable and accurate result. The t-model constructed from the Roe flux performs poorly. Although the t-model constructed from the Roe flux does



(a) Temporal evolution of resolved entropy

(b) Temporal evolution of the rate of change of resolved entropy

Figure 6.6: Numeric solutions to the Burgers' equation using 16 elements with $p = 1$.

remain stable in this case, this can not be guaranteed for the general case.

Model Results

Having established the importance of constructing MZ-VMS models using the entropy stable flux formulation, we now further investigate the performance of the t-model as well as the dynamic-τ model. All MZ results use the entropy conservative flux. As formulated in Section 4.3, the dynamic-τ model determines a scaling of the MZ term by relating energy transfer across two scales. The dynamic model can be directly extended to the discontinuous-Galerkin formulation by using the inner product defined in Eq. 6.54, which corresponds to a physical space volume integral of the two vectors.

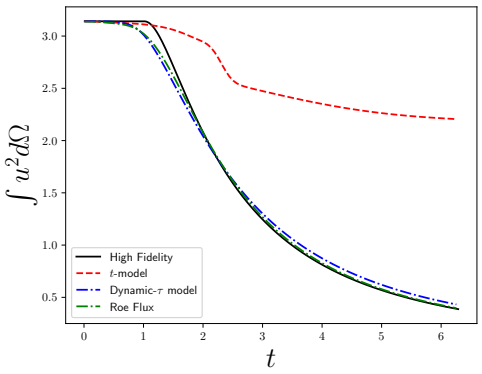
We examine the performance of the MZ models for the standing shockwave case in two different regimes. First, we examine the performance of the model in the case that we have $N = 7$ elements with a linear $p = 1$ polynomial representation in each element. This case is intended to be representative of under-resolved effects in the case of a low order discretization. Figure 6.7 shows the simulation results for this setting. It is seen that in this regime, the t-model performs poorly. Both the

evolution of total entropy and the physical space profiles are poor. The dynamic-τ model offers significant improvements. The evolution of entropy is well characterized. The physical space solution, however, still displays some unphysical features. The simulation run with the Roe flux provides the best results in this setting. With multiple inter-element fluxes, the numerical flux function is able to dissipate an appropriate amount of entropy.

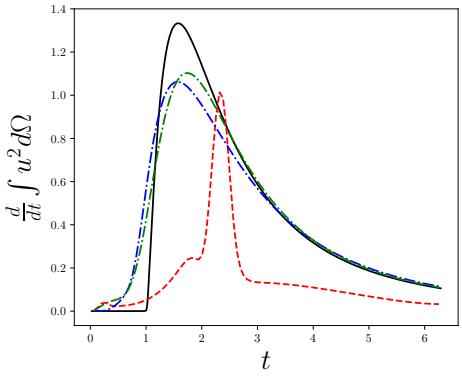
Next we examine the performance of the models in the case where we have $N = 1$ element with a high-order $p = 15$ polynomial representation in the element. This case is representative of under-resolved features in a high-order discretization, where degrees of freedom have been packed into p instead of h. Figure 6.8 shows the results of the numerical simulations. The results in this regime are drastically different than they were in the lower order counter-part. We first observe that the Roe flux performs poorly. This is not surprising as traditional numerical flux functions were designed in the context of low order finite volume schemes. The simulation performed using the t-model produces much improved results, allowing for a reasonable characterization of evolution of entropy as well as the physical space solution. The dynamic-τ model further improves upon the t-model.

6.6.5 Burgers' Equation Summary

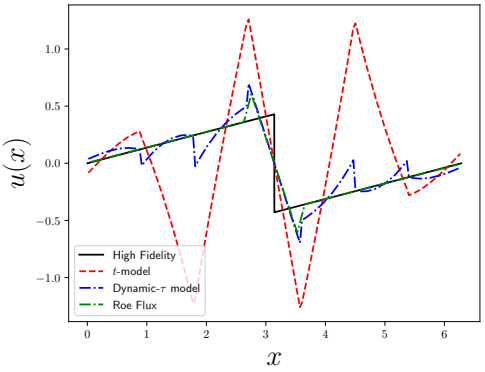
In the application to Burgers' equation, the importance of the numerical flux functions and kinetic energy stability (which is an entropy function for the Burgers' equation) was established. Numerical simulations showed that formulations of the t-model based on non-entropy stable fluxes lead to both poor performance and stability issues. The use of a kinetic energy conservative formulation, however, led to accurate MZ-VMS models. For more complex systems, such as the Navier-Stokes equations, guarantees of entropy stability will be critical. This will be shown in the next section.



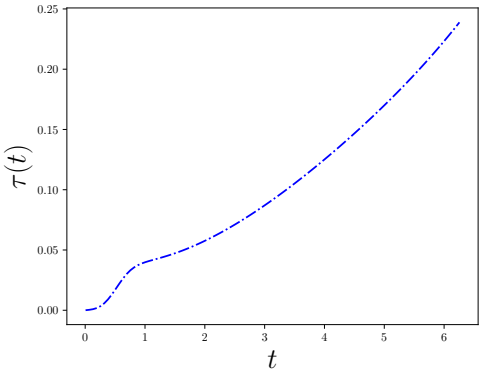
(a) Temporal evolution of resolved entropy



(b) Rate of change of resolved entropy

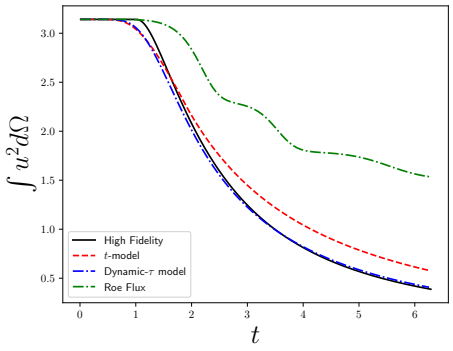


(c) Physical space solution

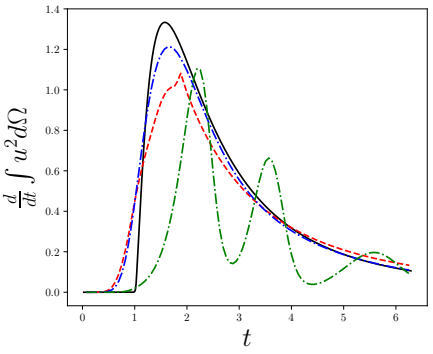


(d) Time constant predicted by the dynamic model

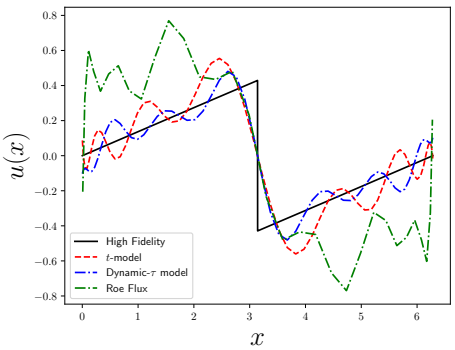
Figure 6.7: Numeric solutions to the Burgers' equation using 7 elements with $p = 1$.



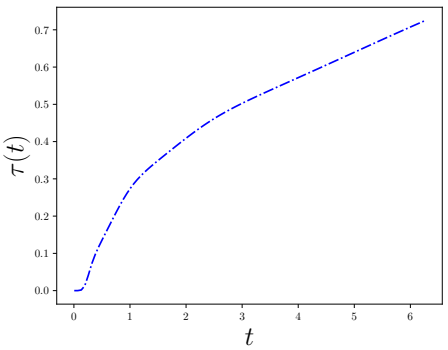
(a) Temporal evolution of resolved entropy



(b) Rate of change of resolved entropy



(c) Physical space solution



(d) Time constant predicted by the dynamic model

Figure 6.8: Numeric solutions to the Burgers' equation using 1 element at $p = 16$.

6.7 The Compressible Navier-Stokes Equations

We now consider the Navier-Stokes equations of gas dynamics. The Navier-Stokes equations consist of a non-linear system of differential equations that govern the conservation of mass, momentum, and energy, and are given by,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \big(\mathbf{F}(\mathbf{u}) - \mathbf{F}_v(\mathbf{u}, \nabla \mathbf{u})\big) = 0, \tag{6.100}$$

where $\mathbf{F}$ and $\mathbf{F}_v$ are the inviscid and viscous fluxes, respectively. For a three dimensional flow the vectors are,

(6.101)

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{Bmatrix}, \qquad \mathbf{F}_1 = \begin{Bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ \rho u_1 u_3 \\ u_1(E+p) \end{Bmatrix}, \qquad \mathbf{F}_2 = \begin{Bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ \rho u_2 u_3 \\ u_2(E+p) \end{Bmatrix}, \qquad \mathbf{F}_3 = \begin{Bmatrix} \rho u_3 \\ \rho u_1 u_3 \\ \rho u_2 u_3 + p \\ \rho u_3^2 \\ u_3(E+p) \end{Bmatrix},$$

$$\mathbf{F}_{v_1} = \begin{Bmatrix} 0 \\ \tau_{11} \\ \tau_{12} \\ \tau_{13} \\ u_j \tau_{j1} + c_p \frac{\mu}{\mathrm{Pr}} \frac{\partial T}{\partial x_1} \end{Bmatrix}, \qquad \mathbf{F}_{v_2} = \begin{Bmatrix} 0 \\ \tau_{21} \\ \tau_{22} \\ \tau_{23} \\ u_j \tau_{j2} + c_p \frac{\mu}{\mathrm{Pr}} \frac{\partial T}{\partial x_2} \end{Bmatrix}, \qquad \mathbf{F}_{v_3} = \begin{Bmatrix} 0 \\ \tau_{31} \\ \tau_{32} \\ \tau_{33} \\ u_j \tau_{j3} + c_p \frac{\mu}{\mathrm{Pr}} \frac{\partial T}{\partial x_3} \end{Bmatrix}.$$

For a Newtonian fluid, the viscous stress tensor is of the form,

$$\tau_{ij} = 2\mu S_{ij}, \tag{6.102}$$

where

$$S_{ij} = \frac{1}{2}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big) - \frac{1}{3}\frac{\partial u_k}{\partial x_i}\delta_{ij}. \tag{6.103}$$

The Navier-Stokes equations are closed with constitutive relationships for a calorically perfect gas,

$$p = \rho RT; \qquad E = e + \frac{1}{2}\rho(u_1^2 + u_2^2 + u_3^2).$$

6.7.1 Numerical Discretization Strategies

There are various ways in which Eq. 6.100 may be discretized within the discontinuous Galerkin framework. The first choice lies in what variables to discretize. The most straightforward process is to discretize the conservative variables,

$$\mathbf{u}(\mathbf{x}, t) = \sum_{k=0}^{N-1} \sum_{j=0}^{p} \mathbf{a}_{jk}(t)\mathbf{w}_{jk}(\mathbf{x}). \tag{6.104}$$

The conservative variable approach is widely used. However, such an approach suffers from non-linear stability issues [32]. An alternative formulation which has been shown to be more robust is to discretize in entropy variables. While a full discussion on entropy variables and entropy stability is outside the scope of this work, we briefly discuss it here.

For inviscid flows, the Navier-Stokes equations simplify to the Euler equations,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0. \tag{6.105}$$

The Euler equations are known to obey an auxiliary conservation law for a quantity referred to as entropy,

$$\frac{\partial U}{\partial t} + \nabla \cdot f(U) = 0. \tag{6.106}$$

What are referred to as entropy variables are defined as,

$$\mathbf{v} = \left(\frac{\partial U}{\partial \mathbf{u}}\right)^T. \tag{6.107}$$

For the scope of this work, it suffices to say that Eq. 6.106 is obtained by,

$$\mathbf{v}^T \frac{\partial \mathbf{u}}{\partial t} + \mathbf{v}^T \nabla \cdot \mathbf{F}(\mathbf{u}) = 0. \tag{6.108}$$

At the continuous level, any solution to Eq. 6.105 will satisfy the entropy conservation law. At the discrete level, however, no such guarantee exists. This has resulted in a wide body of research into what are known as entropy conservative and entropy stable numerical methods. For discontinuous Galerkin methods, entropy conservative schemes can be obtained through the use of entropy conservative fluxes *and* by discretizing in the entropy variables, i.e.

$$\frac{\partial}{\partial t}\mathbf{u}(\mathbf{v}) + \nabla \cdot \mathbf{F}\big(\mathbf{u}(\mathbf{v})\big) = 0. \tag{6.109}$$

where

$$\mathbf{v}(\mathbf{x}, t) = \sum_{k=0}^{N-1} \sum_{j=0}^{p} \mathbf{a}_{j,k}(t)\mathbf{w}_{k,j}(\mathbf{x}).$$

Entropy conservative and entropy stable formulations have been shown to greatly improve nonlinear stability properties of numerical methods [32]. For the Navier-Stokes equations, the entropy variables are,

$$\mathbf{v} = \begin{bmatrix} -\frac{s}{\gamma-1} + \frac{\gamma+1}{\gamma-1} + \frac{\rho E}{p} \\ \frac{\rho u}{p} \\ \frac{\rho v}{p} \\ \frac{\rho w}{p} \\ -\frac{\rho}{p} \end{bmatrix}, \tag{6.110}$$

where $s = \ln(p/\rho^{\gamma})$. In the following sections, both the conservative and entropy formulations will be considered

6.7.2 Numerical Fluxes

The discontinuous Galerkin approach requires the selection of a numerical flux scheme. The majority of flux functions add numerical dissipation to a system and act as implicit subgrid-scale models. Various flux functions are considered in this

work. We examine the Roe flux [84], Rusanov flux [86], Roe's entropy conservative flux [54], and a central flux. These are briefly outlined.

The Roe flux is given by

$$\hat{\mathbf{F}}(\mathbf{u}_L, \mathbf{u}_R) = \underbrace{\frac{1}{2}(\mathbf{F_L} + \mathbf{F_R})}_{\text{central}} - \underbrace{\frac{1}{2}\left|\mathbf{A}(\mathbf{u}^*)\right|(\mathbf{u}_R - \mathbf{u}_L)}_{\text{implicit SGS-model}}, \tag{6.111}$$

where $\mathbf{u}^*$ is the Roe averaged state. In writing Eqn. 6.111, we have expressed the flux as a non-dissipative symmetric central flux with the addition of a asymmetric dissipative flux. In the context of under-resolved LES, the asymmetric component of the flux can be viewed as an implicit subgrid-scale model [32].

The Rusanov flux (or local Lax Friedrichs) is a simpler version of the Roe flux and is given by

$$\hat{\mathbf{F}}(\mathbf{u}_L, \mathbf{u}_R) = \underbrace{\frac{1}{2}(\mathbf{F_L} + \mathbf{F_R})}_{\text{central}} - \underbrace{\frac{1}{2}\rho\Big(\mathbf{A}(\mathbf{u}^*)\Big)(\mathbf{u}_R - \mathbf{u}_L)}_{\text{implicit SGS-model}}, \tag{6.112}$$

with $\rho()$ being the spectral radius. Using the maximum eigenvalue upwinds all waves in the system, but leads to the scheme being overly dissipative for systems with multiple wave speeds [38].

Roe's entropy conservative flux was designed to conserve entropy for finite volume-type discretizations. The flux scheme is given by,

$$\hat{\mathbf{F}}(\mathbf{u}_L, \mathbf{u}_R) = \mathbf{F}(\hat{\mathbf{u}}), \tag{6.113}$$

where $\hat{\mathbf{u}}$ is an averaged state consisting of arithmetic and logarithmic averages [54]. The entropy conservative flux is a non-dissipative flux and is strong candidate for MZ modeling. Note that, in the context of discontinuous Galerkin methods, entropy conservative schemes require that one discretizes with entropy variables and uses an entropy conservative flux.

Lastly, the central flux is considered,

$$\hat{\mathbf{F}}(\mathbf{u}_L, \mathbf{u}_R) = \frac{1}{2}(\mathbf{F_L} + \mathbf{F_R}). \tag{6.114}$$

Like Roe's entropy conservative flux, the central flux is a non-dissipative flux. It, however, does not guarantee entropy stability.

6.7.3 MZ-VMS Flux Functions

As has been previously discussed, each numerical flux scheme has a corresponding MZ-VMS flux function. For the Navier-Stokes equations in three dimensions, however, the algebra required to derive the flux schemes becomes overly complex. Further, even in the one-dimensional case, the MZ-VMS flux functions for sophisticated fluxes become overly complex. As a result, only the flux scheme derived for the central flux in one dimension is presented.

Central Flux

For the central flux, the MZ-VMS flux function can not be simplified in any coherent way and is written as

$$\mathbf{f}_{MZ}(\mathbf{u}_k^R, \mathbf{u}_{k+1}^L) = \tau^* \frac{\mathbf{A}_k^R + \mathbf{A}_{k+1}^L}{4}(\mathbf{f}_k^R - \mathbf{f}_{k+1}^L). \tag{6.115}$$

Some insight into the flux function can be drawn, however, with the identity $\mathbf{f} = \mathbf{A}\mathbf{u}$. This allows for the manipulation,

$$\mathbf{f}_{MZ}(\mathbf{u}_k^R, \mathbf{u}_{k+1}^L) = \tau^* \frac{\mathbf{A}_k^R + \mathbf{A}_{k+1}^L}{4}(\mathbf{A}_k^R\mathbf{u}_k^R - \mathbf{A}_{k+1}^L\mathbf{u}_{k+1}^L). \tag{6.116}$$

We make several important remarks:

1. It is seen that the multiplication of the flux Jacobians will lead to a type of upwinding scheme.
2. The flux function given in Eq. 6.115 does not satisfy the conditions established in [84] that were deemed necessary for a flux to satisfy the Rankine Hugoniot conditions. As such, the MZ-VMS flux function constructed on a central flux does not satisfy the Rankine Hugoniot conditions.

Entropy Conserving Flux

Another interesting flux function to consider is Roe's entropy conservative flux. Due to the logarithmic averages and non-linear expressions, however, the flux scheme becomes incoherent and we do not pursue an analytic expression here.

6.7.4 Entropy Stability

The entropy stability of the t, τ, and VMS(ϵ) models is now considered. For simplicity, the following derivation will neglect boundary operators. In the derivation, the first step is to assume that the original numerical method conserves entropy. As such, the results are applicable only to entropy conservative formulations.

Consider the governing equations in weighted residual form,

$$(\mathbf{w}, \mathbf{u}_t) + (\mathbf{w}, \mathcal{R}(\mathbf{u})) = 0. \tag{6.117}$$

We consider numerical methods that conserve entropy. To derive an evolution equation for entropy, first set $\mathbf{w} = \mathbf{v}^T$,

$$(\mathbf{v}^T, \mathbf{u}_t) = -(\mathbf{v}^T, \mathcal{R}(\mathbf{u})) = 0. \tag{6.118}$$

Note that it is only possible to set $\mathbf{w} = \mathbf{v}^T$ for formulations where $\mathbf{v}$ is in the subspace spanned by the basis functions. This is the case when one discretizes in

entropy variables, but not when one discretizes in conservative variables. Proceed by splitting $\mathbf{v}$ into coarse and fine scales,

$$(\tilde{\mathbf{v}}^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}} + \mathbf{v}'))) + (\mathbf{v}'^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}} + \mathbf{v}'))) = 0. \tag{6.119}$$

Setting $\mathbf{v}' = 0$, it is seen that,

$$(\tilde{\mathbf{v}}^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}}))) = 0. \tag{6.120}$$

Now set $\mathbf{v}' = \epsilon\mathbf{h}$,

$$(\tilde{\mathbf{v}}^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}} + \epsilon\mathbf{h}))) + (\epsilon\mathbf{h}^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}} + \epsilon\mathbf{h})) = 0. \tag{6.121}$$

Expanding in a Taylor series and using the chain rule,

$$(\tilde{\mathbf{v}}^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}})) + \epsilon(\mathbf{h}^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}})) + (\tilde{\mathbf{v}}^T, \frac{\partial\mathcal{R}}{\partial\mathbf{u}}\frac{\partial\mathbf{u}}{\partial\mathbf{v}}\epsilon\mathbf{h})) + \epsilon^2(\mathbf{h}^T, \frac{\partial\mathcal{R}}{\partial\mathbf{u}}\frac{\partial\mathbf{u}}{\partial\mathbf{v}}\epsilon\mathbf{h}) = 0. \tag{6.122}$$

Dividing by ϵ, letting $\epsilon \to 0$, and noting the first term drops,

$$(\mathbf{h}^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}})) + (\tilde{\mathbf{v}}^T, \frac{\partial\mathcal{R}}{\partial\mathbf{u}}\frac{\partial\mathbf{u}}{\partial\mathbf{v}}\mathbf{h})) = 0. \tag{6.123}$$

Setting,

$$\mathbf{h} = \Pi'_{\mathbf{v}}\mathcal{R}(\mathbf{u}(\tilde{\mathbf{v}})),$$

where $\Pi'_{\mathbf{v}}$ is the projection onto the fine scales (i.e. the fine-scale mass matrix) as defined by the entropy variables,

$$\Pi'_{\mathbf{v}} f = \mathbf{w}'^T \left[\int \mathbf{w}' \frac{\partial\mathbf{u}}{\partial\mathbf{v}} \mathbf{w}'^T d\Omega \right]^{-1} (\mathbf{w}', f). \tag{6.124}$$

gives

$$(\mathcal{R}(\mathbf{u}(\tilde{\mathbf{v}}))^T \Pi'^T_{\mathbf{v}}, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}})) + (\tilde{\mathbf{v}}^T, \frac{\partial\mathcal{R}}{\partial\mathbf{u}}\frac{\partial\mathbf{u}}{\partial\mathbf{v}}\Pi'_{\mathbf{v}}\mathcal{R}(\mathbf{u}(\tilde{\mathbf{v}})))) = 0. \tag{6.125}$$

For entropy variables the projection matrix is positive definite which implies,

$$(\mathcal{R}(\mathbf{u}(\tilde{\mathbf{v}}))^T \Pi'^T_{\mathbf{v}}, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}})) > 0. \tag{6.126}$$

The following result emerges,

$$(\tilde{\mathbf{v}}^T, \frac{\partial \mathcal{R}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Pi'_{\mathbf{v}} \mathcal{R}(\mathbf{u}(\tilde{\mathbf{v}})))) \leq 0. \tag{6.127}$$

The entropy evolution for the τ and VMS(ϵ) models is given by,

(6.128)

$$\left(\tilde{\mathbf{v}}^T, \frac{\partial}{\partial t}\mathbf{u}(\tilde{\mathbf{v}})\right) + \left(\tilde{\mathbf{v}}^T, \mathcal{R}\big(\mathbf{u}(\tilde{\mathbf{v}})\big)\right) + \left(\tilde{\mathbf{v}}^T, \tau \frac{\partial \mathcal{R}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Pi'_{\mathbf{v}} \mathcal{R}(\mathbf{u}(\tilde{\mathbf{v}}))\right) <= 0 \qquad \forall \tau > 0.$$

It is seen that, in an entropy conservative formulation, the t, τ, and VMS(ϵ) models dissipate entropy. Hence, the schemes are entropy stable.

6.7.5 Numerical Applications

We now consider the application of the MZ-VMS models to the compressible Navier-Stokes equations. Note that only the Euler fluxes are considered in the construction of the MZ-VMS models. A high-order DG code utilizing Legendre basis functions, termed PyDG, was developed for this purpose. The viscous fluxes are handled through the interior penalty formulation [47]. All numerical integration is performed using $2p+1$ quadrature points. The code is a structured code utilizing Legendre basis functions to have arbitrary order of accuracy. The code has matrix-free implicit time marching and spacetime capabilities. Appendix C provides additional details about the code.

Numerical Strategy for MZ-VMS Models

The following sections will consider the first order MZ-VMS models, which require the evaluation of $e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\mathbf{a}$. While obtaining the analytic form of the volumetric term is straightforward, obtaining the analytic form of the multi-dimensional flux functions is not. Two strategies are pursued to evaluate the terms:

1. Finite difference-enriched space method: For cases where a conservative formulation is used on an orthogonal grid, the first order models can be computed

through a call to an enriched space. This technique, which is discussed in Chapter 3, is straightforward to implement but costly. In the following examples, the enriched space is taken to be twice the order of the solution space. To accelerate the evaluation of the term, a collocated quadrature is used for the enriched space. It is important to emphasize that this formulation should only be used with a numerical method that has a non-dissipative flux scheme (such as central flux) as the MZ-VMS model will implicitly contain a dissipative flux term.

2. Volumetric method: For non-orthogonal grids and entropy stable formulations, the computation of the first order models through the finite difference-enriched space method is made challenging due to requirements on the orthogonality between spaces. Computation of the volumetric term, however, remains straightforward. In the volumetric method, only the volumetric term is added to the discontinuous Galerkin scheme. The volumetric term requires the computation of the strong form residual, which here is achieved by differentiating the discrete unknowns and using chain rule. In addition to tractability, this technique is advantageous as it is independent of the numerical flux function and allows the MZ-VMS models to be *augmented* to existing flux schemes that have numerical dissipation. Philosophically, the approach is using existing flux functions to model the "flux" portion of the memory and is adding in an extra volumetric term.

6.7.6 Conservative Formulation

The Taylor Green Vortex problem using a conservative variable formulation is first considered. In this example, simulations are performed using a central flux with the τ model. Note that simulations using this setup with no subgrid-scale model are unstable. The MZ-VMS model form is computed using the finite difference-enriched

space method described above. As we can not provide any theoretical guarantees about stability, this is first addressed. In particular, the sensitivity of the first order MZ-VMS models to τ is examined.

Taylor Green at Infinite Reynolds Number

We first examine the Taylor Green vortex at an infinite Reynolds number. No viscous dissipation is present in this case and the numerical method must be capable of providing adequate numerical dissipation. Numerical simulations are performed using the τ model at 4_{th} order with eight elements. Crank Nicolson time integration at $CFL < 1$ is used. A simulation using the Roe flux with no MZ-VMS model is additionally presented. Figure 6.9 shows the evolution of the time rate of change for total entropy. Simulations run with the Roe flux are seen to dissipate entropy for all time, indicating that the simulation is stable (which does not necessarily mean accurate). The stability of simulations run using the τ-model and a central flux are seen to depend on the selection of τ. In Figure 6.9 it is seen that if τ grows too large, the rate of change of entropy becomes positive and the simulation diverges. It is also observed that, if τ grows too small, the simulation again diverges.

The fact that the stability of the MZ-VMS central-flux τ-model formulation hinges on τ taking on an appropriate intermediate value is unattractive and not robust. This issue pushes the idea of pursuing MZ-VMS methods within an entropy stable formulation. Before doing this, however, the simulation of several canonical flows using the MZ-VMS central-flux τ-model is considered, using an appropriate value of τ. Note that the τ emerging from the VMS(ϵ) formulation leads to stability issues. Based on numerical experiments and CFL considerations, the remainder of

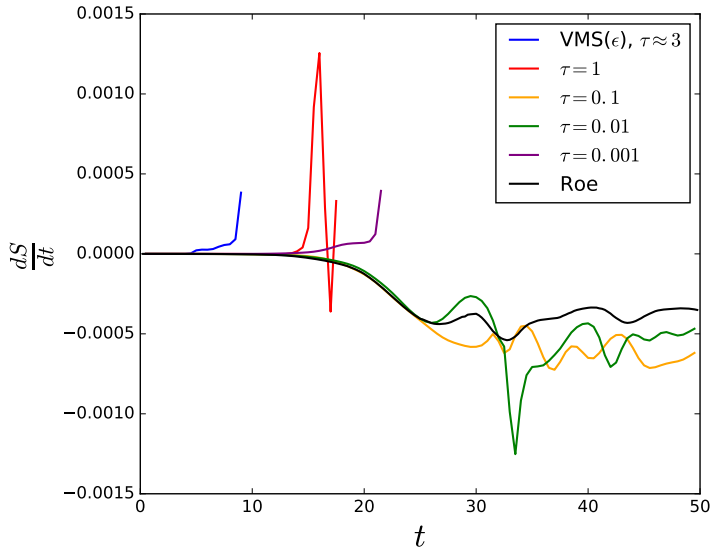


Figure 6.9: Evolution of the rate of change of entropy

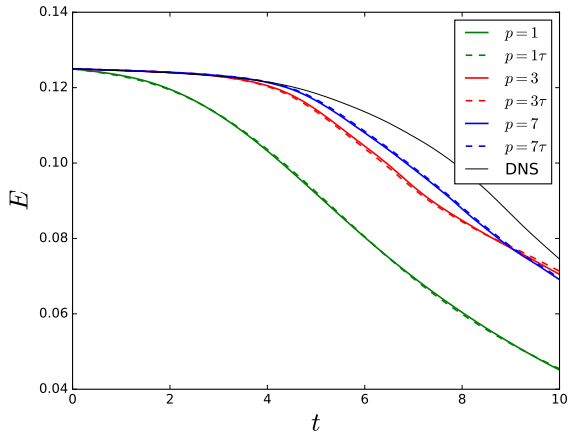
this section will examine solutions where the value of τ is chosen to be,

$$\tau = \tau * \frac{a_\infty \Delta x}{p^2}, \tag{6.129}$$

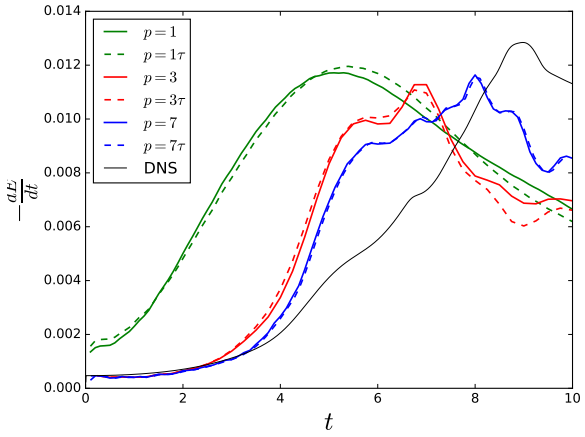
where $\tau^* = 1$, p is the maximum polynomial order of the enriched space, and a_∞ the speed of sound.

Taylor Green Vortex at $Re = 1600$

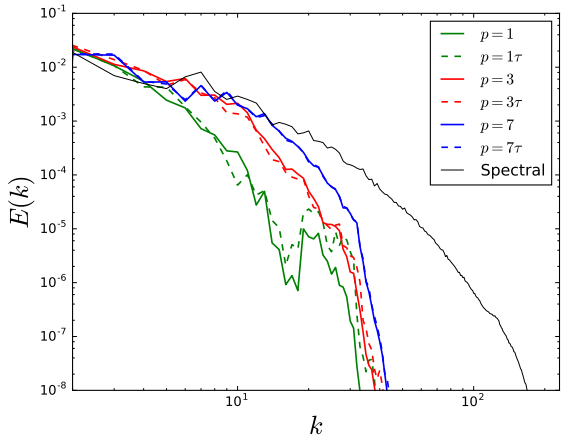
The Taylor Green vortex at a finite Reynolds number of $Re = 1600$ is now considered. Simulations are performed at $M = 0.2$ using the central flux τ-model (with τ as defined in Eq. 6.129) and the Roe flux with no subgrid model. Figure 6.10 shows results for various orders of accuracy at 32^3 DOFs, while Figure 6.11 shows the same results for 64^3 DOFs. For both cases, the central flux τ-model (with the appropriate selection of τ) leads to a result that is remarkably similar to the simulation with the Roe flux. These similarities demonstrate how flux schemes, such as the Roe flux, act as implicit subgrid-scale models. It is quite surprising to the author that, in this complex three dimensional case, the two methods produce such similar results.



(a) Evolution of kinetic energy.

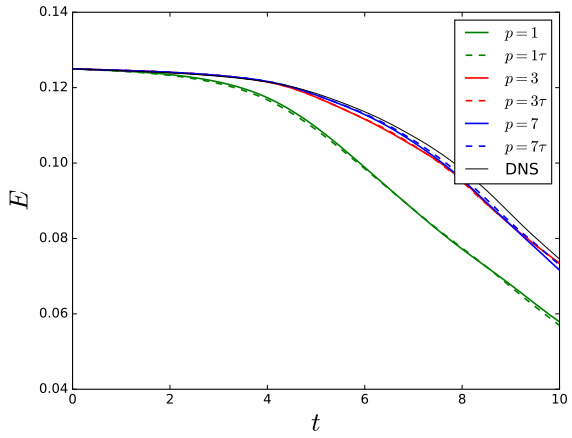


(b) Dissipation of kinetic energy, dE/dt.

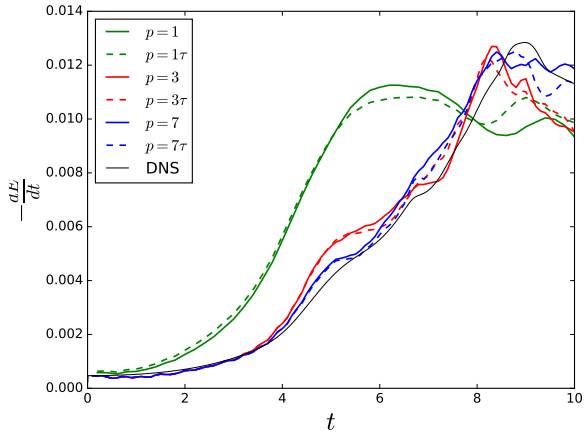


(c) Energy spectra at $t = 10.0$

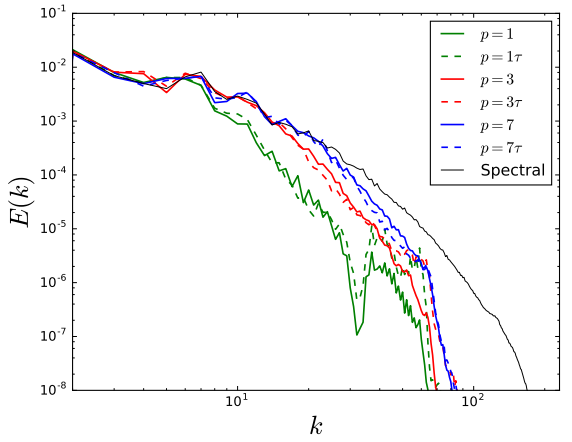
Figure 6.10: Quantities of interest for the Taylor Green Vortex using the τ models at 32^3 DOFs. Note that the DNS shown is unfiltered, so comparison of the total kinetic energy and dissipation is qualitative.



(a) Evolution of kinetic energy.



(b) Dissipation of kinetic energy, dE/dt.



(c) Energy spectra at $t = 10.0$

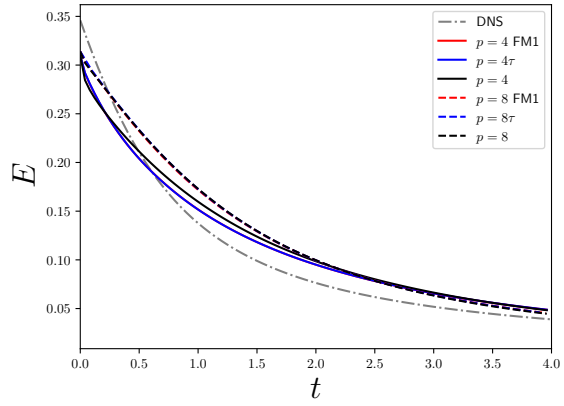
Figure 6.11: Quantities of interest for the Taylor Green Vortex using the τ models at 64^3 DOFs. Note that the DNS shown is unfiltered, so comparison of the total kinetic energy and dissipation is qualitative.

Homogeneous Turbulence

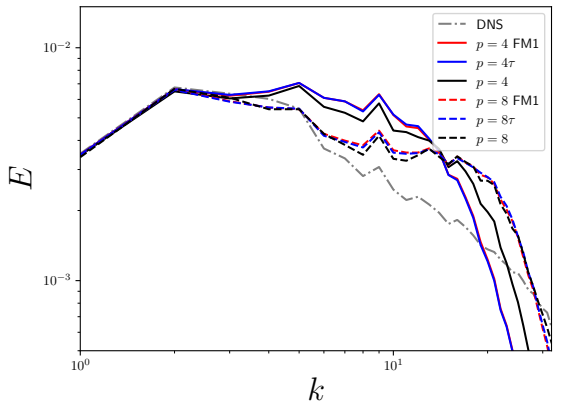
Homogeneous turbulence at $Re_\lambda \approx 80$ is next examined. Both the static-τ model and the finite memory models are considered. The results are shown in Figure 6.12 for simulations at 64 DOFs using both a 4th order and an 8th order discretization. Unfiltered DNS from a spectral code is additionally shown. It is again seen that adding the MZ-VMS model to the central flux leads to a scheme that performs comparably to a Roe scheme. Additionally, it is seen that the finite memory models and static-τ model produce essentially identical results. To further investigate and understand the similarities between the MZ-VMS models and existing flux schemes, the implicit SGS-model inherent to the Roe scheme (i.e. the upwinding mechanism) is compared to the τ-model. The Roe scheme can be written as,

$$\mathbf{F}^*(\mathbf{u}_L, \mathbf{u}_L) = \underbrace{\frac{1}{2}(\mathbf{F_L} + \mathbf{F_R})}_{\text{central}} - \underbrace{\frac{1}{2}\left|\mathbf{A}(\mathbf{u}^*)\right|(\mathbf{u}_R - \mathbf{u}_L)}_{\text{implicit SGS-model}}. \tag{6.130}$$

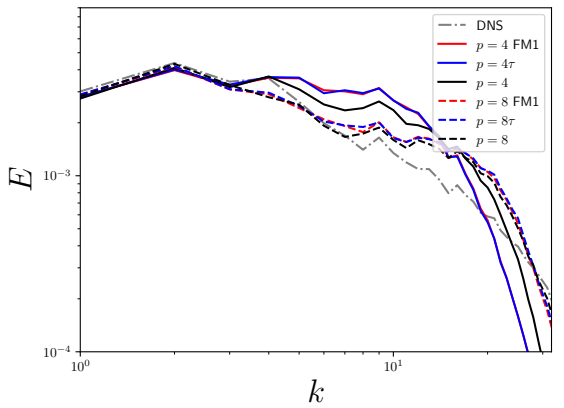
The implicit SGS-model inherent to the Roe scheme is the second term on the right hand side of Eq. 6.130. To compare the methods, the sub-grid transfer spectra of the SGS-models are examined. Figure 6.13 shows the computed sub-grid transfer spectra for the HIT simulation at $t = 2.0$. The Roe scheme, τ-model, and Rusanov scheme are shown. The transfer spectra of the τ-model are seen to behave in a qualitatively similar fashion as the Roe (and Rusanov) scheme, but the magnitudes are slightly different. This discrepancy is attributed to the selection of τ, which was determined heuristically. Figure 6.14 compares the spectra for the "optimal" τ^* (i.e. the τ^* that most closely matches the Roe scheme). The τ-model compares almost perfectly to the Roe scheme, provided the correct τ is selected.



(a) Evolution of kinetic energy.

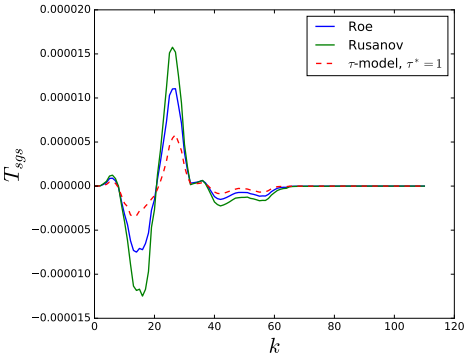


(b) Energy spectra at $t = 2.0$

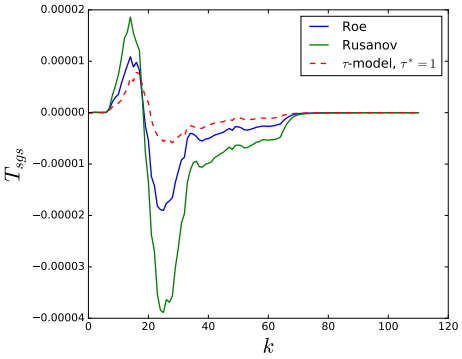


(c) Energy spectra at $t = 4.0$

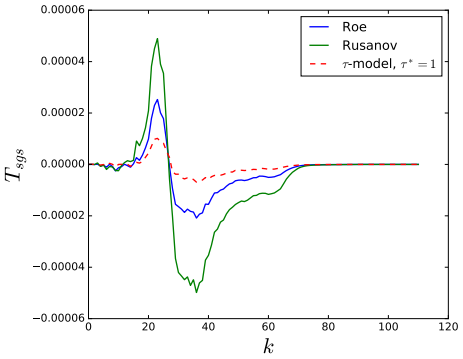
Figure 6.12: Quantities of interest for the homogeneous turbulence using MZ models at 64^3 DOFs. Note that the DNS shown is unfiltered, so comparison of the total kinetic energy is qualitative. All black lines are obtained with the Roe flux.



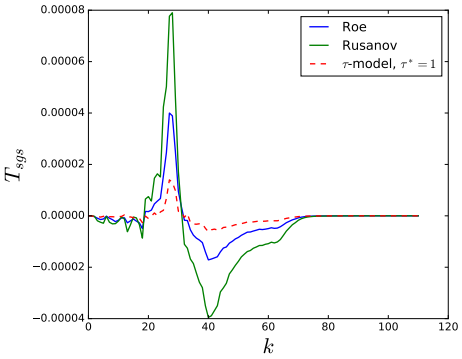
(a) $p = 2$ polynomial with 32 elements.



(b) $p = 4$ polynomial with 16 elements.

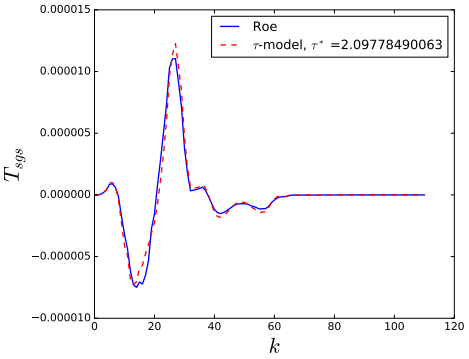


(c) $p = 8$ polynomial with 8 elements.

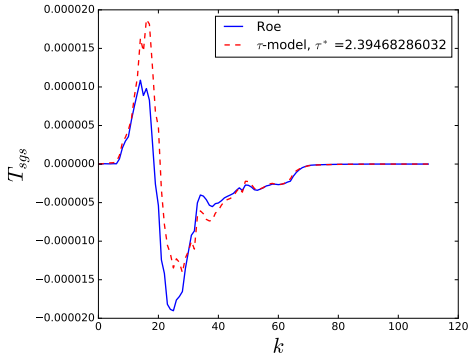


(d) $p = 16$ polynomial with 4 elements.

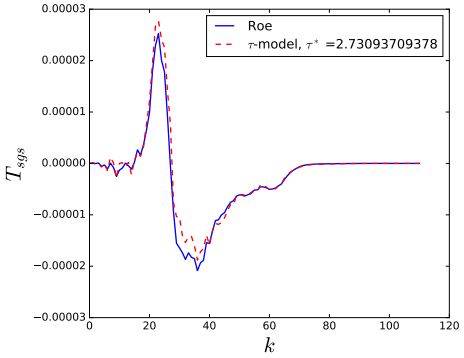
Figure 6.13: Transfer spectra for the sub-grid energy spectra at $t = 2.0$ for various combinations of p and h.



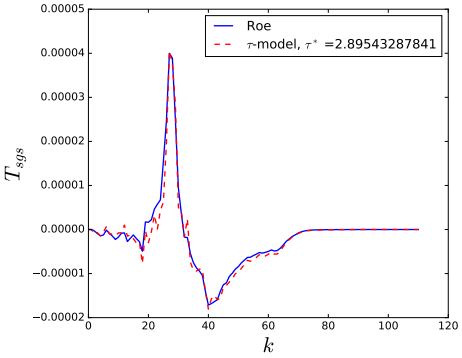
(a) $p = 2$ polynomial with 32 elements.



(b) $p = 4$ polynomial with 16 elements.



(c) $p = 8$ polynomial with 8 elements.



(d) $p = 16$ polynomial with 4 elements.

Figure 6.14: Transfer spectra for the sub-grid energy spectra at $t = 2.0$ for various combinations of p and h with an "optimal" τ^*.

Stability

A significant issue that arrises in the examples presented above is that of stability. For the MZ-VMS models constructed using a conservative variable formulation, there are no theoretical guarantees of stability. Further, as the MZ-VMS models are constructed from a non-dissipative flux function, the models must add the dissipation required for stability. In the context of the τ-model, the selection of a τ that is too low leads to numerical blowup due to insufficient numerical dissipation. Selection of a τ that is too high additionally leads to numerical blowup. It was due to this sensitivity that τ was selected in an a priori fashion via Eq. 6.129. Due to these stability issues, the utility of the MZ-VMS models outlined in this dissertation for conservative formulations is challenging. In practice, using MZ-VMS closures as a method to stabilize an otherwise unstable calculation (without having any theoretical stability guarantees), is not a robust or desirable technique.

We now turn our attention to the application of MZ-VMS models within an entropy conservative formulation. As discussed earlier, the entropy conservative formulation allows for theoretical guarantees in entropy stability for both the baseline discontinuous Galerkin simulations as well as for simulations using the first order MZ-VMS models.

6.7.7 Entropy Conservative Formulation

Simulations using an entropy conservative formulation are now considered. The MZ-VMS models are evaluated using the volumetric method described above, which does not require any calls to an enriched space. Simulations using Roe's entropy conservative flux (Roe EC), the local Lax Friedrichs (Rusanov) flux, the Roe EC flux with the VMS(ϵ) model, and the local Lax Friedrichs flux with the VMS(ϵ)

model. The standard Roe flux is not entropy stable and as such is not used in this section. It is noted that, while not pursued here, the Roe flux can be modified to maintain entropy stability[54]. For the VMS(ϵ) model, which the reader will recall is equivalent to the τ-model, the timescale requires a modification to account for the polynomial order. In this work we take,

$$\tau = \left[\frac{4}{h_p^2}u^2 + 3\pi\nu^2\left(\frac{4}{h_p^2}\right)^2\right]^{-1/2}, \tag{6.131}$$

where $h_p = h/p$. Note that in the previous section, simulations using the VMS(ϵ) model could not be performed as the value of τ predicted through the method led to numerical blowup.

Taylor Green Vortex at $Re = 1600$

The Taylor Green vortex problem using an entropy stable formulation is now considered. Simulations are first performed at $Re = 1600$ using 64^3 degrees of freedom. Simulations are considered using four methods:

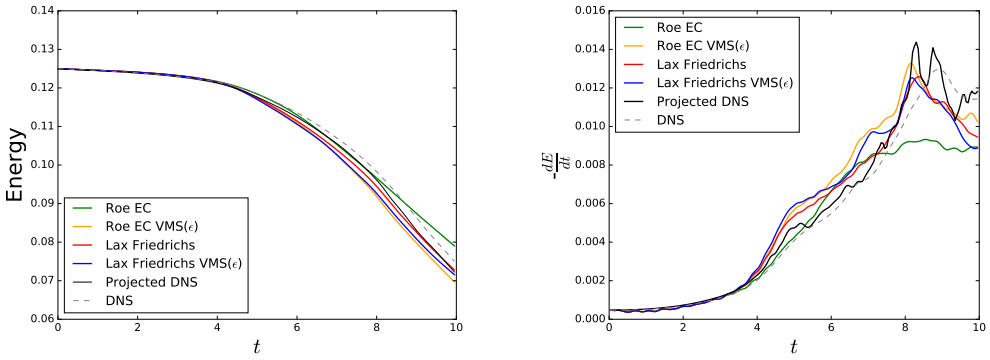
1. Entropy conservative formulation with Roe's entropy conserving flux (Roe EC). This case can be considered the "no model" case as it does not add any numerical dissipation to the system.
2. Entropy conservative formulation with Roe EC and the volumetric VMS(ϵ) model. This setup is entropy stable and will dissipate entropy in time. In this case dissipation is implicitly introduced through a volumetric term, the form of which is derived from the MZ-VMS framework.
3. Entropy conservative formulation with the Rusanov (local Lax-Friedrichs) flux. This setup is entropy stable and will dissipate entropy in time. In this case, dissipation is being introduced through the Rusanov flux.

4. Entropy conservative formulation with the Rusanov flux and volumetric VMS(ϵ) model. This setup is entropy stable and will dissipate entropy in time. Dissipation is introduced through a volumetric term derived from MZ-VMS as well as the Rusanov flux.

Figure 6.15 shows results for fourth order DG simulations using 16^3 elements. The baseline simulation using Roe's entropy conservative flux is observed to not be sufficiently dissipative. This is analogous to the results obtained in Chapter V. The addition of the VMS(ϵ) model is seen to introduce extra diffusion to the system and leads to an improved prediction, although the model is seen to add slightly too much diffusion. Simulations utilizing the Lax Friedrichs flux are seen to offer more dissipation than Roe's EC flux, but not as much diffusion as the VMS(ϵ) model. Simulations that use the Lax Friedrichs flux and the VMS(ϵ) model are seen to perform comparably to the Roe EC VMS(ϵ) simulations. Figure 6.16 shows results for the same case, but at eighth order and 8^3 elements. For the higher order simulation, the solution space is seen to be richer than the lower order counterpart. The baseline simulations using Roe's entropy conservative flux are improved from lower order case. Simulations that utilize the VMS(ϵ) model are again seen to add diffusion to the system and are again slightly over dissipative.

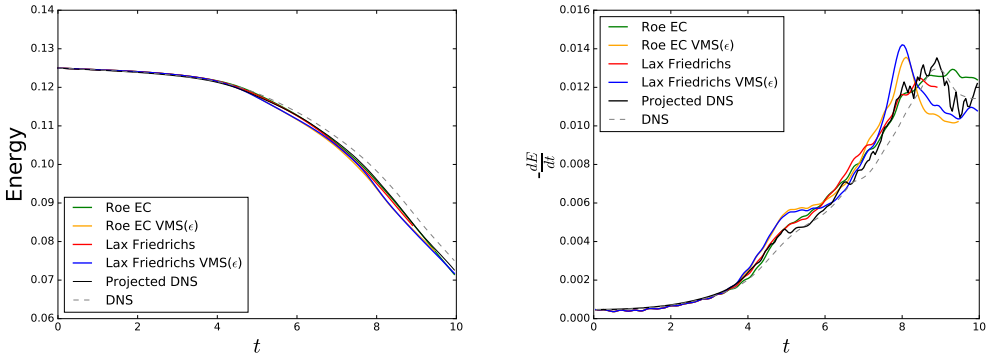
Taylor Green Vortex at Infinite Reynolds Number

To further distinguish between simulations with and without numerical dissipation, the Taylor Green vortex at an infinite Reynolds number is now considered. In this case, explicit numerical dissipation is required to remove energy from the system. Figure 6.17 shows the evolution of total kinetic energy and the rate of dissipation of kinetic energy for all methods. The simulation run with the Roe EC flux, which has no numerical dissipation, is seen to have minimal to no change in total kinetic



(a) Evolution of total kinetic energy. (b) Dissipation of kinetic energy

Figure 6.15: Numerical results for the Taylor Green vortex problem at $Re = 1600$ at fourth order with 16^3 elements.



(a) Evolution of total kinetic energy. (b) Dissipation of kinetic energy

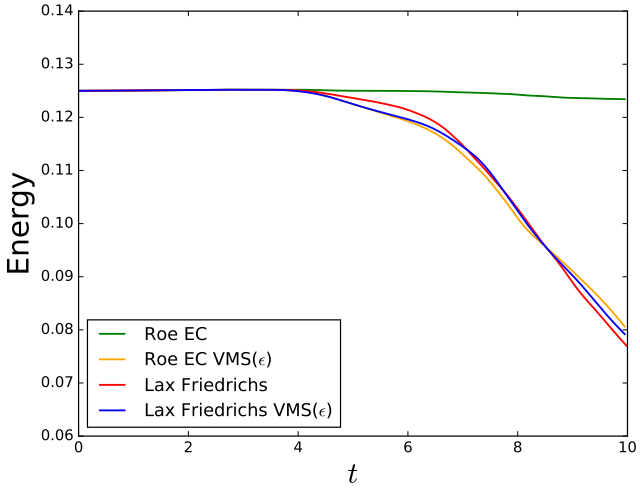
Figure 6.16: Numerical results for the Taylor Green vortex problem at $Re = 1600$ at eighth order with 8^3 elements.

energy. This lack of dissipation leads to a build up in kinetic energy at the fine scales, as seen in the energy spectra. It is worth noting that, although the simulation is inviscid, the total kinetic energy in the *resolved* scales should still decrease in time due to the turbulent cascade. The addition of the VMS(ϵ) model to the Roe EC flux is seen to lead to a monotonic decay in energy and predicts a qualitatively accurate energy spectra. Simulations that are evolved using the local Lax Friedrichs flux behave qualitatively similar to the Roe EC VMS(ϵ) simulation. This is especially surprising considering the fundamental differences between the two methods and their disparity with the Roe EC simulation. The energy spectrum from the Lax Friedrichs simulation, however, is seen to display a slight buildup of kinetic energy at high wave-numbers. The addition of the VMS(ϵ) model to the Lax Friedrichs flux removes kinetic energy at high wave numbers and leads to predictions that are comparable to the Roe EC VMS(ϵ) method.

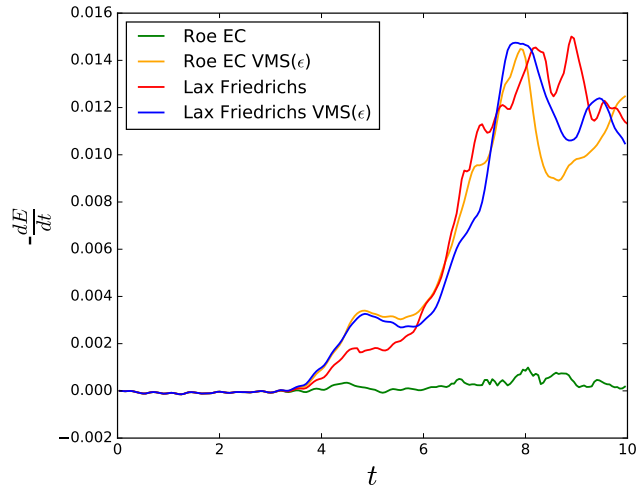
6.8 MZ-VMS Inspired Filtering

Thus far, we have analyzed MZ-VMS as a closure modeling strategy. We demonstrated that the MZ-VMS formulation gives rise to both volumetric and flux terms. For conservative formulations, we showed that the net contribution of these terms provides dissipation in a form very similar to stabilization methods such as a Roe scheme. In predictive simulations, these closure terms were shown to provide an effect similar to the Roe flux. The stability of the formulation for conservative methods, however, was seen to be dependent on the selection of the timescale τ. We proceeded to show how MZ-VMS methods can lead to improvements in entropy stable formulations. This offers a promising path forward for MZ-VMS models.

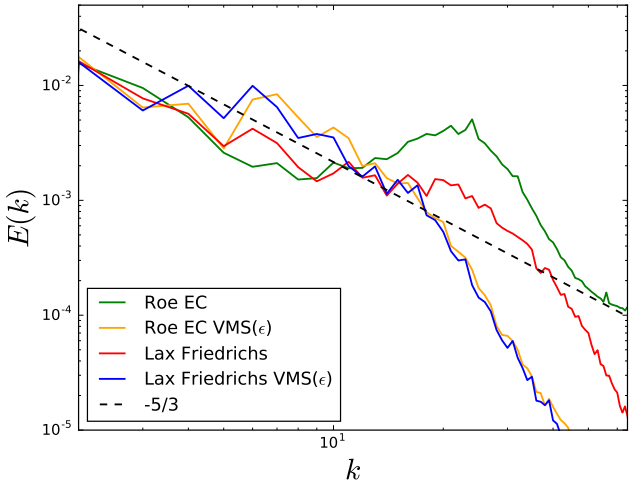
The majority of discontinuous Galerkin codes, unfortunately, are not formulated



(a) Evolution of total kinetic energy.



(b) Dissipation of kinetic energy



(c) Kinetic energy spectra at $t = 10.0$

Figure 6.17: Numerical results for the Taylor Green vortex problem at $Re = \infty$ at eighth order with 8^3 elements.

using entropy variables. The ability to improve the solution accuracy of discontinuous Galerkin methods using conservative variables by virtue of existing MZ-VMS methods is thus made challenging. While advancements in accuracy and stability are perhaps possible by closely examining the balance of volumetric and surface terms, such a path is not clear at this time. Fortunately, the concepts developed in this chapter can still be used to improve the predictive capabilities of standard high-order discontinuous Galerkin methods. To provide this context, we conclude this chapter by demonstrating how MZ-VMS methods can be used as resolution indicators. This information can, in turn, be used to improve the predictive capabilities of a numerical method.

High-order methods are advantageous for the simulation of under-resolved flows due to their low amounts of numerical dissipation and dispersion. Further, they are a strong candidate for future computing architectures to to their high flop count per degree of freedom and limited communication/compute ratio. High-order methods, however, may struggle in the presence of shocks and/or strong gradients. Under-resolution in theses regimes often leads to unphysical oscillations and, at times, failure of a numerical method. The most common remedy for these issues is to employ shock capturing methods. Traditionally, shock capturing methods have focused on the addition of artificial viscosity [77, 78] and limiting [23, 57]. In the latter case, troubled cells are typically identified and then limited (eg. as in [57]).

The MZ-VMS formulation is an excellent candidate as an under-resolution indicator. Examining the memory at $s = 0$ provides information about the level of resolution of a numerical method and can be used to indicate when a simulation is under-resolved. We demonstrate now how the memory at $s = 0$ can be utilized for limiting.

6.8.1 Hierarchical Filtering

We consider again the application of the MZ-VMS framework to the conservation equation given in Eq. 6.15. Through the MZ-VMS framework, the evolution equation for the coarse scales can be exactly written as,

$$\int_\Omega \tilde{\mathbf{w}}\mathbf{u}_t d\Omega + \int_\Omega \tilde{\mathbf{w}}\nabla\cdot\mathbf{F}(\tilde{\mathbf{u}})d\Omega + \int_\Gamma \tilde{\mathbf{w}}\mathbf{b}(\tilde{\mathbf{u}},\mathbf{n})d\Gamma = \tilde{\mathbf{M}}\int_0^t K(\tilde{\mathbf{a}}(t-s),s)ds. \tag{6.132}$$

As should be evident by now, computing the entire memory term is intractable. The value at $s = 0$, however, is computable and shows up in the form of Eq. 6.65. As we have discussed, Eq. 6.65 contains terms that correspond to volumetric terms and surface terms. Here, we use the volumetric term as an indicator for under-resolution,

$$r_{pk}(\tilde{\mathbf{a}}) = \int_\Omega\int_\Omega (\tilde{w}_{pk}\mathcal{R}')(x)\Pi'(x,y)(\mathcal{R}(\tilde{u})-f)(y)d\Omega_y d\Omega_x. \tag{6.133}$$

The volumetric term alone is utilized as it is local to an element and its computation is straightforward. We use this indicator to assess the resolution level and then, if required, filter the solution. Various filtering procedures are possible, and here a simple hierarchical projection-based filter is used. Recall that in one-dimension, the representation of our polynomial on the k_{th} element is given by

$$\tilde{u}_k = \sum_{j=0}^{p} w_{jk}(x)a_{jk}(t). \tag{6.134}$$

In our hierarchical limiting procedure, we set $a_{jk} = 0$ if the resolution indicator exceeds a certain threshold. We take this threshold to be a percentage of the right hand side budget. The selection of this threshold determines the sensitivity of the filtering procedure. The algorithm is shown below.

for $k \in \mathcal{T}$ **do**

  for $j = 0, p$ **do**

    compute $r_{pk} = \int_\Omega \int_\Omega (\tilde{w}_{pk}\mathcal{R}')(x)\Pi'(x,y)(\mathcal{R}(\tilde{u}) - f)(y)d\Omega_y d\Omega_x$;

    if $r_{jk}(\tilde{\mathbf{a}}) > \epsilon\big(\int_\Omega \tilde{w}_{jk}\nabla\cdot\mathbf{F}(\tilde{\mathbf{u}})d\Omega + \int_\Gamma \tilde{w}_{jk}\mathbf{b}(\tilde{\mathbf{u}},\mathbf{n})d\Gamma\big)$ **then**

      $a_{qk} = 0$ for all $q \geq j$;

    end

  end

end

Algorithm 1: Memory-based hierarchical limiting algorithm.

Remarks

- Note that $r_{pk} = 0$ for $p = 0$. As such the filtering procedure will, at most, reduce the scheme to first order.
- It is observed that, in general, $r_{(p+1),k} > r_{p,k}$.

6.8.2 Application to the Isentropic Vortex

Before assessing the performance of the filtering algorithm on problems with strong gradients where filtering is required for stability, the performance of the algorithm is assessed in smooth regimes. In such regimes, the filtering algorithm should have a minimal impact on the solution and not effect the asymptotic order of convergence. Here the isentropic vortex problem is examined [102], which is initialized

by,

$$\begin{Bmatrix} \rho \\ u \\ v \\ T \end{Bmatrix} = \begin{Bmatrix} T^{\frac{1}{\gamma-1}} \\ u_\infty - \frac{\beta}{2\pi} e^{\frac{1}{2}(1-r^2)}(y-y_0) \\ v_\infty + \frac{\beta}{2\pi} e^{\frac{1}{2}(1-r^2)}(x-x_0) \\ 1-(\gamma-1)\frac{\beta^2}{8\gamma\pi^2} e^{1-r^2} \end{Bmatrix}, \tag{6.135}$$

where,

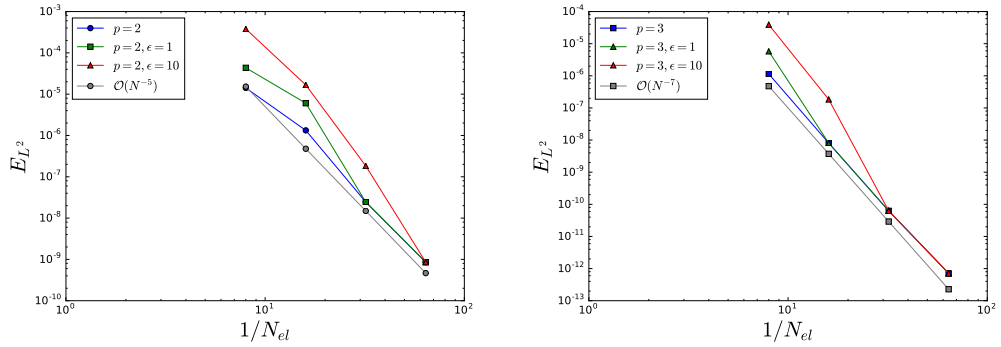
$$r = \sqrt{(x-x_0)^2 + (y-y_0)^2}.$$

The parameters selected are $x_0, y_0, \beta = 5$ and $u_\infty, v_\infty = 1$. The periodic domain has dimensions $L \in [0,10] \times [0,10]$. The exact solution corresponds to advection of the vortex at the free-stream velocity.

Figure 6.18 shows the convergence in the L^2 state error after the vortex propagates once through the domain for various polynomial errors and mesh sizes. The L^2 state error is defined as,

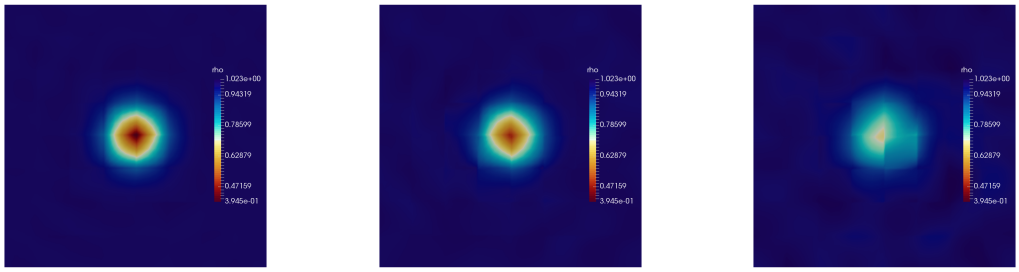
$$E_{L^2} = \frac{1}{\Omega} \int (\mathbf{u} - \mathbf{u}_{exact})^2 d\Omega.$$

In Figure 6.18, it is seen that the filtering process only modifies the convergence properties of the DG scheme when the solution is under-resolved. For all cases considered, simulations utilizing the filtering algorithm retain the asymptotic rate of convergence. Figure 6.18 also shows the sensitivity of the filtering procedure to the parameter, ϵ. For simulations run with a higher value of ϵ, the filtering scheme is more active. Figure 6.19 shows contours of density after the vortex has advected once through the domain for the case with $p = 2$ and 8 elements in each direction. This is the coarsest case considered. It is seen that the solution run with no filtering process does a reasonable job capturing the advection of the vortex. Due to the low resolution, however, jumps at the element interfaces can be seen. For more complex problems, these jumps can lead to stability issues. Solutions run with the filtering



(a) $p = 2$ (b) $p = 3$

Figure 6.18: Convergence profiles for the isentropic vortex case at various polynomial orders.



(a) $\epsilon = 0$ (b) $\epsilon = 1$ (c) $\epsilon = 10$

Figure 6.19: Physical space density profiles for $p = 2, N = 8$ case.

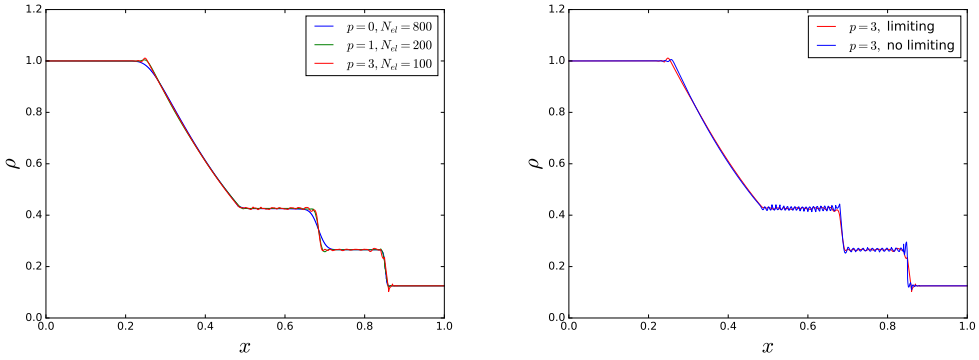
algorithm are seen to diffuse the vortex. This leads to a decrease in accuracy but an increase in stability.

6.8.3 Application to the Sod Shocktube

The Sod shocktube is a classic shock-capturing problem that presents many challenges for high-order methods. The problem setup is given by,

$$\rho = \begin{cases} 1 & x \leq 0.5 \\ 0.125 & x > 0.5 \end{cases}, \qquad p = \begin{cases} 1 & x \leq 0.5 \\ 0.1 & x > 0.5 \end{cases}, \qquad u = \begin{cases} 0 & x \leq 0.5 \\ 0 & x > 0.5 \end{cases}. \tag{6.136}$$

Figure 6.20 shows numerical results for simulations at 400 DOFs. The filtering parameter is taken to be $\epsilon = 1$. To illustrate the effectiveness of higher-order methods,



(a) Solution to the Sod shocktube problem at $t = 0.2$ for various orders of accuracy

(b) Comparison of limited and non-limited solutions

Figure 6.20: Numerical results for the Sod shocktube problem.

the $p = 0$ solution is run using 800 DOFs. The filtering procedure is seen to effectively remove spurious oscillations and allows for high-order methods to be effectively applied to shock-dominated problems.

6.8.4 Shu-Osher Problem

We next consider the Shu-Osher problem. The Shu-Osher problem consists of a Mach 3 shock wave impacting a fluctuating density wave and is a canonical problem for shock-turbulence interactions. The numerical method must be able to both capture the fine-scale density fluctuations and the strong shock wave. The problem setup is given by,

$$\rho = \begin{cases} 3.8571 & x \leq -4 \\ 1 + 0.2\sin(5x) & x > -4 \end{cases}, \qquad p = \begin{cases} 10.3333 & x \leq -4 \\ 1 & x > -4 \end{cases}, \tag{6.137}$$

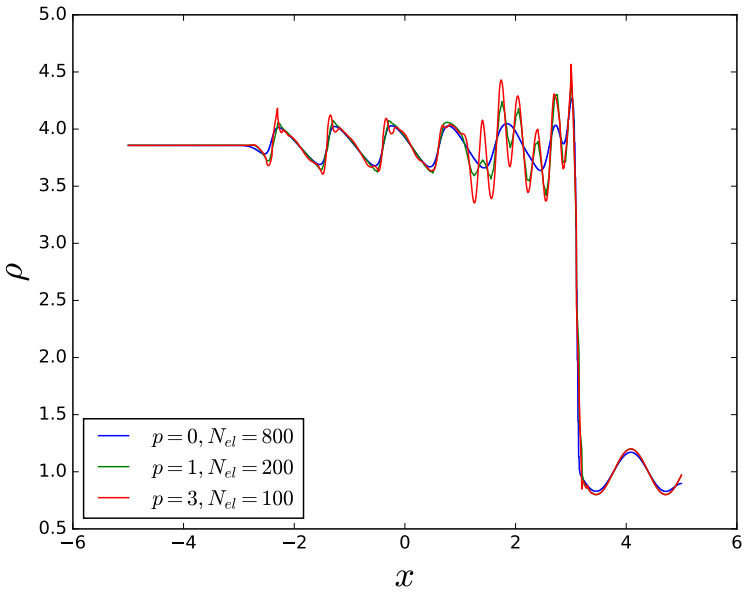
$$u = \begin{cases} 2.6293 & x \leq -4 \\ 0 & x > 4 \end{cases}.$$

Figure 6.21 shows numerical results for the Shu-Osher problem for polynomial orders $p = 1$ and $p = 3$ on grid sizes corresponding to a total of 400 and 800 DOFs. As be-

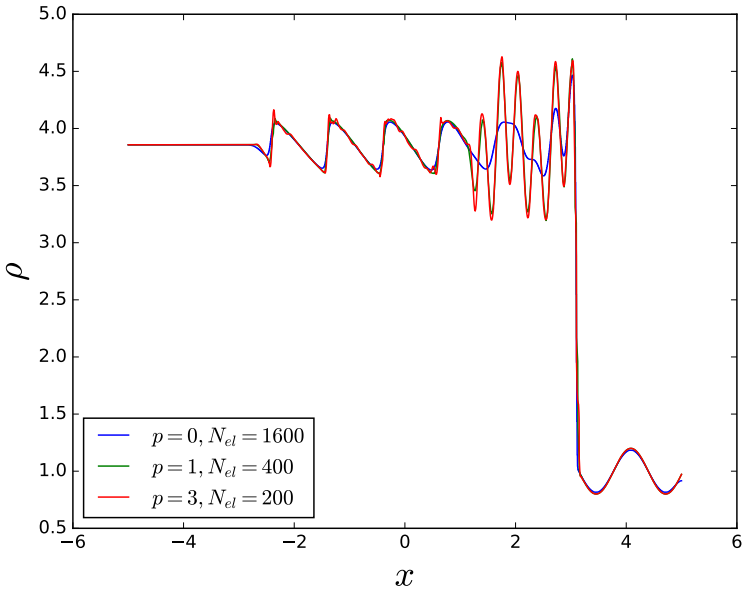
fore, the solutions are compared to a $p = 0$ solution run using twice the total degrees of freedom. For the Shu-Osher problem, all DG simulations with $p >= 1$ using the Roe flux went unstable. The inclusion of the MZ-VMS filtering procedure stabilizes the simulations and allows for the use of high-order methods. In Figure 6.21, the high-order methods (with filtering) are seen to out-perform the lower order methods for capturing the fluctuating density wave behind the shock. This is particularly relevant in the 400 DOF simulations. The high-order methods, however, do have oscillations around the shock fronts.

6.8.5 Viscous Sod shocktube in two dimensions

As a final example, we consider the application of the filtering procedure to a multidimensional calculation. Note that the algorithm is directly applicable to multidimensional calculations (this is often not the case for limiting procedures). The problem considered is the sod shocktube on the unit square with a viscosity of $\mu = 1e-5$. Simulations are performed on a uniform mesh at a resolution of $N_{el} = 400 \times 400$ with $p = 3$ polynomials in each direction. Isothermal boundary conditions are used on the top and bottom walls, while reflecting boundary conditions are used on the left and right walls. The simulation results at $t = 1.0$ and $t = 2.0$ are shown in Figure 6.22. It is noted that the baseline DG simulation went unstable at this resolution. The inclusion of the MZ-VMS filtering procedure is seen to stabilize the simulation, while still allowing for complex features of the flow to be captured. In particular, the simulation is able to capture the lambda shock pattern arising from the shock-boundary layer interactions as well as the corner vortices caused by the shock reflections.

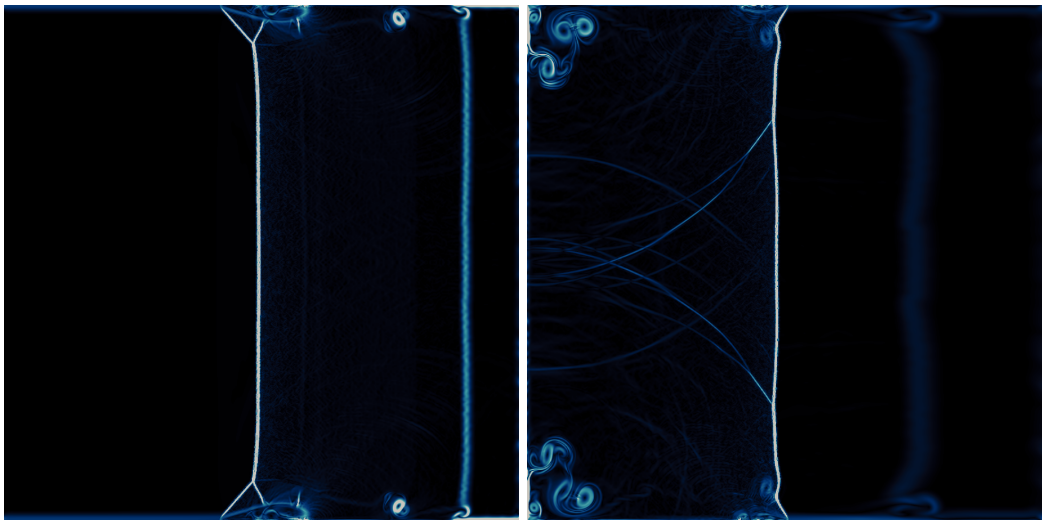


(a) Solution to the Shu-Osher problem at $t = 2.0$ for various orders of accuracy.



(b) Solution to the Shu-Osher problem at $t = 2.0$ for various orders of accuracy on refined mesh.

Figure 6.21: Numerical results for the Shu-Osher problem.



(a) Solution at $t = 1.0$ (b) Solution at $t = 2.0$

Figure 6.22: Contours of the density gradient for the two-dimensional Sod shocktube problem.

6.9 Chapter Summary

In this chapter we outlined the application of the MZ-VMS framework for finite elements and considered the application of MZ-VMS methods within the context of discontinuous Galerkin. The important findings from this chapter are:

1. The finding that the MZ-VMS memory appears in the form of volumetric and flux terms. The flux terms display many commonalities with classic upwinding flux functions. For Burgers' equation and linear advection in one dimension, the τ-model (with an appropriate selection for τ) constructed on a central flux was shown to be formally equivalent to the upwind flux correction.

2. The demonstration that the MZ-VMS framework can lead to closure terms that improve the solution predictions for high-order simulations of Burgers' equation. For lower-order methods, however, the flux schemes dominate the memory term and significant improvements over baseline calculations using the Roe flux were not observed.

3. The construction of MZ-VMS methods for discontinuous Galerkin methods using a conservative variable formulation led to stability issues. The simulations were observed to diverge if the timescale τ was chosen to be either too high or too low. For an appropriately chosen τ, the MZ-VMS methods were able to stabilize a central flux and led to results similar to those obtained with traditional flux schemes.

4. The development of MZ-VMS methods within an entropy conservative formulation led to entropy stable closure models. These closure models were shown to improve the predictive capabilities of the numerical simulations.

5. The MZ-VMS framework was shown to be useful as an error indicator. We demonstrated how the memory can be used as an indicator for troubled cells and outlined a filtering algorithm that filters high-order polynomials when a simulation is under-resolved. The applicability of this algorithm was demonstrated for shock-dominated calculations in one and two-dimensions. It was shown that the filtering algorithm was shown to stabilize high-order calculations without compromising the accuracy of the method in smooth regimes.

This chapter concludes this dissertations investigation of MZ-VMS methods. The following chapter will provide conclusions and perspectives.

CHAPTER VII

Perspectives

7.1 Conclusions

The simulation of multiscale systems continues to be a prominent challenge in computational physics. Various strategies exist for handling multiscale physics. The variational multiscale method is one such promising approach. The VMS method, which was developed in the context of linear steady-state problems, provides a formal strategy for the numerical simulation of multiscale systems. The extension of classic VMS approaches to non-linear and unsteady problems has, however, proved challenging. This work advances the state-of-the-art in VMS modeling by unifying the variational multiscale method with the Mori-Zwanzig formalism. The MZ-formalism, a tool from statistical mechanics, has long been viewed as a powerful model reduction strategy. The promise of the method, however, has yet to be realized. In developing the MZ-formalism within the VMS framework, this work demonstrates that the MZ-formalism can be used as a practical modeling tool. The MZ-VMS formulation shows, for the first time, that memory effects are closely linked to classic stabilization methods that have existed in the community for some time. The MZ-VMS framework further shows that several existing subgrid-scale/stabilization methods can be viewed as a subset of a much broader class of models, thus offering the promise of

improved predictive capabilities for multiscale systems.

The remainder of this chapter will outline the specific contributions of this dissertation as well as recommended future work.

7.2 Contributions

The major contributions of this work are:

1. **A formulation of of Chorin's optimal prediction framework for variational multiscale-type discretizations of partial differential equations.** This formulation was pursued for a specific projection operator. Prior to this work, the formulation of MZ-based methods for practical discretization techniques has remained unclear.

2. **A rigorous investigation of the performance of existing MZ models for Fourier-Galerkin simulations of turbulent flows.** Prior to this work, the t-model had only been briefly considered as a closure model for LES in the work of Chandy and Frankel [15]. This work provided a much deeper investigation into the performance of the models for canonical turbulent flows at a variety of resolutions. We showed that for simulations where the cross-stresses dominate, which is the case when a simulation is mildly under-resolved, MZ models outperform the dynamic Smagorinsky model. MZ models were shown to be capable of predicting backscatter at several wavenumbers, while still being globally dissipative. It was then shown that when the Reynolds stresses dominate, which is the case when a simulation is severely under-resolved, the MZ models considered do not perform as well as the Smagorinsky model. This shortcoming is attributed to the fact that the MZ methods examined in this work are unable to model the Reynolds stresses.

3. **An establishment of connections between MZ-based methods and residual-based closures.** By using MZ-based methods within the VMS framework, these connections emerged from the analytic derivation of the model forms. In particular, it was shown that the t and τ-model have a structural form that is equivalent to a class of VMS models. This provides a great deal of insight into the mechanics of MZ-based methods and helps bridge a gap that has existed between two communities.

4. **Application of MZ-based methods to the discontinuous Galerkin method and an establishment of connections between MZ-based methods, upwinding, and artificial viscosity. This work marks the first time that MZ-based approaches have been applied to finite element methods.** To the authors knowledge, this is additionally the first application of residual-based VMS models to discontinuous Galerkin simulations of turbulent flows.

5. **A demonstration that, in discontinuous Galerkin simulations, unclosed effects lead to a memory term that is driven by the coarse-scale residual and inter-element jumps.** The inter-element jumps lead to memory terms that resemble upwind fluxes and approximate Riemann solvers. This dissertation work further showed how VMS and MZ-VMS methods applied to discretizations of the Euler equations using conservative variables lead to stability issues. This was demonstrated through both theoretical and numerical analyses. It was shown that, when applied to discretizations of the Euler equations using entropy variables, entropy stability can be guaranteed. This was again demonstrated through both theoretical and numerical analyses. When applied to entropy stable formulations, the MZ-VMS models were seen to provide improved solutions at high Reynolds numbers.

6. **The development of an autonomous dynamic MZ-VMS model, coined the dynamic-τ model.** The dynamic-τ model is constructed by assuming similarities between two levels of coarse-graining via the Germano identity. The model was shown to perform well for several flows. For spectral methods it led to improved predictions over the existing VMS(ϵ) model and, in general, performed comparable to tuned versions of Stinis' finite memory models. The developed model offers a significant advancement in MZ-based models in that it does not explicitly contain a time term (unlike the t-model) and requires minimal to no tuning.

7. **A formulation of an auxiliary ODE system that can be used to approximate the orthogonal dynamics.** This "orthogonal ODE" provides the first tractable approximation to the orthogonal dynamics for high dimensional system that the author is aware of. The author wishes to take this point to acknowledge Ayoub Gouasmi for his contributions to the development of the orthogonal ODE and its procedure [43]. Without these contributions, some of the theoretical origins of the orthogonal ODE and how it could be actually be used to compute the memory would be unclear.

8. **Investigation of MZ-VMS methods for limiting.** It was shown how MZ-VMS methods can be used as resolution indicators. These resolution indicators can be used to drive limiting and/or adaptation. An MZ-VMS-based filtering algorithm was devised to stabilize high-order methods. The algorithm was shown to maintain asymptotic convergence for smooth problems and stabilize otherwise unsteady calculations for shock-dominated problems.

9. **The development of an open-source spectral method solver for the triply periodic Navier-Stokes equations and an open-source discon-**

tinuous Galerkin solver for the compressible Navier-Stokes equations. These codes are both distributed-memory Python-based codes and have displayed strong scaling to over 500 cores. The codes are freely available on GitHub.

7.3 Future Work

This dissertation work highlighted several important areas where research is required to further advance MZ-VMS-based methods. Several areas for future work are:

1. The development of MZ-models that allow for increasingly accurate approximations to the memory kernel. These models should maintain a low cost. We showed in this work that several existing MZ models (Stinis' finite memory models and the dynamic-τ model) are incapable of dissipating enough energy for significantly under-resolved, high Reynolds number simulations using a Fourier-Galerkin method. The same shortcomings were observed with VMS methods. With Fourier-Galerkin methods being the simplest numerical method on which to test MZ-VMS models, future efforts in model development should address this issue.

2. When applied to discontinuous Galerkin simulations of the compressible Navier-Stokes equations using conservative variables, MZ-VMS models offered no improvements over the implicit subgrid-scale models inherent to physics-based flux schemes and led to stability challenges. In this work, we showed that formulating the MZ-VMS models within an entropy conservative method remedied this issue. In practice, however, the majority of numerical codes make use of the conservative variable formulation. The question of whether MZ-VMS methods can be used as a practical tool for such numerical methods remains unclear at

this time and should be a topic of future work.

3. Application of the dynamic-τ model to discontinuous Galerkin simulations of the compressible Navier-Stokes equations. In this work, only the VMS(ϵ) model was considered. In general, it was observed that the VMS(ϵ) model was slightly over dissipative. The dynamic model offers the promise to remedy this issue, allowing for more accurate numerical simulations of under-resolved problems.

4. Application of MZ-VMS methods to multiphysics problems. A primary advantage of the MZ-VMS formulation is its mathematical foundations. This allows it to be applied to complex flows in a straightforward manner. This was demonstrated in this dissertation in the context of magnetohydrodynamics. Future work should continue to explore multiphysics applications.

As a whole, this work offers a significant advancement in the development of MZ models and their application to the equations of fluid dynamics. We have provided a unique perspective into under-resolution in the context of multiscale problems and have helped to advance the understanding of the construction of closure models through a unifying mathematical framework.

APPENDICES

APPENDIX A

Derivation of the memory at $s = 0$ (Spectral Method Case)

A variety of Mori-Zwanzig-based models require evaluating the memory term at $s = 0$,

$$K(\tilde{\mathbf{a}}(t), 0) = e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0.$$

Evaluation of $\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$ is an exercise in algebra, and we provide the step-by-step process for the smooth case where jump terms between the elements are not considered. We consider the non-linear equation,

$$\frac{\partial u}{\partial t} + \mathcal{R}(u) = f.$$

The evaluation of $\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$ is as follows:

1. First compute $e^{t\mathcal{L}}\mathcal{L}\tilde{\mathbf{a}}_0$. This is simply the right hand side of the coarse-scale equation,

   $$e^{t\mathcal{L}}\mathcal{L}\tilde{\mathbf{a}}_0 = (\tilde{\mathbf{w}}, f - \mathcal{R}(\tilde{u})) - (\tilde{\mathbf{w}}, \mathcal{R}(u) - \mathcal{R}(\tilde{u})). \tag{A.1}$$

2. Next compute $e^{t\mathcal{L}}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$. By definition of how the right hand side is written,

   $$e^{t\mathcal{L}}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 = -(\tilde{\mathbf{w}}, \mathcal{R}(u) - \mathcal{R}(\tilde{u})), \tag{A.2}$$

   or in terms of the basis coefficients

   $$e^{t\mathcal{L}}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 = -\Big(\tilde{\mathbf{w}}, \mathcal{R}(\mathbf{w}^T\mathbf{a}) - \mathcal{R}(\tilde{\mathbf{w}}^T\tilde{\mathbf{a}})\Big). \tag{A.3}$$

3. Compute $e^{t\mathcal{L}}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$. This is simplified by recognizing that the Liouville operator is the Fréchet derivative in the direction of the right hand side. For some function scalar function g,

$$\mathcal{L}g(u(\mathbf{a}_0)) = \frac{\partial g}{\partial \mathbf{a}_0}\mathcal{L}\tilde{\mathbf{a}}_0. \tag{A.4}$$

The linearization in the Fréchet derivative is with respect to the modal variables, $\mathbf{a}_0$. For the mapping from $u = \mathbf{w}^T\mathbf{a}$, the chain rule gives

$$\frac{\partial g}{\partial \mathbf{a}_0}\mathcal{L}\tilde{\mathbf{a}}_0 = \frac{\partial g}{\partial u_0}\mathbf{w}^T\mathcal{L}\tilde{\mathbf{a}}_0. \tag{A.5}$$

The Liouville operator can be applied to $\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$ by linearizing $\mathcal{R}$ with respect to u, and evaluating the resulting model at $\mathcal{R}(u)$,

$$\begin{aligned} e^{t\mathcal{L}}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 = -\Big(&\tilde{\mathbf{w}}, \mathcal{R}'\big(\mathbf{w}^T\big[(\tilde{\mathbf{w}}, f - \mathcal{R}(\tilde{u})) - (\tilde{\mathbf{w}}, \mathcal{R}(u) - \mathcal{R}(\tilde{u}))\big]\big) \\ &- \mathcal{R}'\big(\tilde{\mathbf{w}}^T\big[(\tilde{\mathbf{w}}, f - \mathcal{R}(\tilde{u})) - (\tilde{\mathbf{w}}, \mathcal{R}(u) - \mathcal{R}(\tilde{u}))\big]\Big)\Big). \end{aligned} \tag{A.6}$$

4. Compute $e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0$. This sets all terms involving $\mathbf{a}'$ to zero and we obtain

$$e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 = -\Big(\tilde{\mathbf{w}}, \mathcal{R}'\big(\mathbf{w}^T(\mathbf{w}, f - \mathcal{R}'(\tilde{u}))\big) - \mathcal{R}'\big(\tilde{\mathbf{w}}^T(\tilde{\mathbf{w}}, f - \mathcal{R}(\tilde{u}))\big)\Big). \tag{A.7}$$

5. Equation A.7 may be simplified by recognizing that terms appear as projections,

$$e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 = -\Big(\tilde{\mathbf{w}}, \mathcal{R}'\big(\Pi(f - \mathcal{R}(\tilde{u}))\big) - R'\big(\tilde{\Pi}(f - \mathcal{R}(\tilde{u}))\big)\Big). \tag{A.8}$$

6. With $\Pi' = \Pi - \tilde{\Pi}$ and noting that R' is linear,

$$e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_0 = -\Big(\tilde{\mathbf{w}}, \mathcal{R}'\big(\Pi'(f - \mathcal{R}(\tilde{u}))\big)\Big). \tag{A.9}$$

APPENDIX B

Spectral Method Solver: PySpectral

PySpectral solves the triply periodic spectral Navier-Stokes equations in their primitive form using a Galerkin pseudo-spectral method. The code is parallelized through mpi4py and has been used to solve systems with over one billion unknowns. The code implementation uses ideas developed in [70]. Several of the important highlights of the code are detailed in the following section.

B.1 The Psuedo-Spectral Method

The incompressible Navier-Stokes equations are given by

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{B.1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial u_i}{\partial x_j}\right). \tag{B.2}$$

For incompressible flows, application of the divergence free condition allows one to express the pressure as a Poisson equation,

$$\frac{-1}{\rho}\frac{\partial^2 p}{\partial x_i^2} = \frac{\partial u_j}{\partial x_i}\frac{\partial u_i}{\partial x_j}. \tag{B.3}$$

The Galerkin spectral method using Fourier basis functions can be obtained either through a weighted residual formulation, which is equivalently the Fourier transform

of the governing equations. The pressure Poisson equation can be directly solved in Fourier space, allowing the governing equations to be written compactly as

$$\text{(B.4)}\quad \left(\frac{\partial}{\partial t}+\nu k^2\right)a_i(\mathbf{k},t)+\left(\delta_{im}-\frac{k_i k_m}{k^2}\right)\imath k_j \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} a_j(\mathbf{p},t)a_m(\mathbf{q},t)=0 \qquad \mathbf{k}\in F,$$

where a_i is the Fourier transform of u_i. Equation B.4 makes up a set of ordinary differential equations for the modes in F. The challenge with directly solving Eq. B.4 is the expensive convolution term. Instead of directly evaluating the convolution, the psuedo-spectral method evaluates the convolution by first performing the inverse Fourier transform of the velocities (i.e. we compute the inverse Fourier transform for each a_i), computing the non-linear multiplication in physical space, and then taking the Fourier transform of the resulting non-linear term. Listing B.1 shows the implementation of the psuedo-spectral method on PySpectral.

```
    # perform inverse fourier transform of state variables
    myFFT.myifft3D(main.uhat,main.u)
    myFFT.myifft3D(main.vhat,main.v)
    myFFT.myifft3D(main.what,main.w)
    # compute non linear terms and then take the Fourier transform
    myFFT.myfft3D(main.u*main.u,main.NL[0])
    myFFT.myfft3D(main.v*main.v,main.NL[1])
    myFFT.myfft3D(main.w*main.w,main.NL[2])
    myFFT.myfft3D(main.u*main.v,main.NL[3])
    myFFT.myfft3D(main.u*main.w,main.NL[4])
    myFFT.myfft3D(main.v*main.w,main.NL[5])
```

Listing B.1: Python implementation of psuedo-spectral right hand side computation.

B.2 Parallelization and distributed memory Fourier transforms

PySpectral is parallelized through mpi4py [29]. The mpi4py package is a robust module and is a wrapper for the majority of MPI routines. The primary challenge associated with the parallelization of a pseudo-spectral solver is performing the three dimensional fast Fourier transforms in parallel. Several packages exist in Python which allow for the calculation of a three-dimensional Fourier-Transform, including numpy and scipy. Neither of these modules, however, directly have a parallel FFT implementation. To circumvent this issue, PySpectral utilizes the slab decomposition

discussed in [70]. The slab decomposition distributes the mesh along a single index, and can be viewed as parallelizing the code in one-dimension. A three-dimensional parallel FFT is performed by performing local two-dimensional FFTs on each CPU, a global communication, and then a final one-dimensional local FFT on each CPU. Listing B.2 shows the Python code associated with this process.

```
    ## Code for the inverse fourier transform
    def myifft3D(uhat,u):
      self.Uc_hat[:,:,:] = np.fft.ifft(uhat,axis=1)
      self.U_mpi[:] = np.rollaxis(self.Uc_hat.reshape(Npx, num_processes, Npy, N3
    /2+1) ,1)
      comm.Alltoall(self.U_mpi,self.Uc_hatT)
      u[:] = np.fft.irfft2(self.Uc_hatT,axes=(0,2) )
      return u
    ## Code for the fourier transform
    def myfft3D(u,uhat):
      self.Uc_hatT[:,:,:] = np.fft.rfft2(u,axes=(0,2) )
      comm.Alltoall(self.Uc_hatT, self.U_mpi )
      self.Uc_hat[:,:,:] = np.rollaxis(self.U_mpi,1).reshape(self.Uc_hat.shape)
      uhat[:] = np.fft.fft(self.Uc_hat,axis=1)
      return uhat
```

Listing B.2: Python implementation of parallel FFTs.

B.3 Vectorization and broadcasting in Python

One of the primary costs in the pseudo-spectral solver is the element-wise multiplication, addition, and division of arrays. For efficient evaluation of these terms in a high-level language such as Python, array vectorization is critical. A further important consideration relevant to vectorized multiplication in Python is broadcasting. Broadcasting refers to how numpy deals with operations on multiple arrays of different shapes. To illustrate these concepts, we consider the algorithm for the x-derivative of a three-dimensional field $\hat{u}$ in the frequency domain, which is compactly defined as $\hat{u}_x = \imath k_1 \hat{u}$. Listing B.3 shows the implementation of this calculation in python. The reader will note that computation is performed with no loops, thus allowing numpy to efficiently interface with C routines to compute the element-wise multiplications. The reader will further note that k_1 is a one-dimensional vector, which is then broadcast into a three-dimensional vector to perform the element-wise

multiplication. This broadcasting allows for memory savings in storing the k_1 array (i.e. we do not need to store a 3D field).

```
def compute_grad_x(uhat_x,k1,uhat):
    uhat_x[:,:,:] = k1[:,None,None]*uhat[:,:,:]
```

Listing B.3: Python implementation of the x-derivative.

B.4 De-aliasing

Psuedo-spectral methods require some type of de-aliasing to maintain accuracy. De-aliasing is required due to quadratic non-linearities present in the Navier-Stokes equations and there is in fact a close relationship between de-aliasing and closure modeling. PySpectral uses standard 2/3 dealiasing, where the energy content in the highest 2/3 wave-numbers is removed. For the LES calculations, most of which are performed in a slightly older version of PySpectral, 2x padding is used for the FFTs. This allows for minimal bookkeeping in the MZ-VMS models, but is not a requirement.

B.5 Miscellaneous Details

All results presented in the following sections leverage Jameson's low storage RK4 time integration techniques. For visualization, PySpectral utilizes a VTK package, PyEVTK, to write data into a format readable by ParaView. PySpectral is accessible on GitHub.

APPENDIX C

Discontinuous Galerkin Solver: PyDG

PyDG is a high-order discontinuous Galerkin solver developed in this dissertation. The code is developed in a pure Python environment and interfaces with mpi4py and numpy to provide a scalable code that can be used for high fidelity simulations. This appendix outlines several important aspects of the code.

C.1 Tensor Product Legendre Basis Functions and Order of Accuracy

PyDG is built on tensor products using Legendre basis functions defined at the Gauss quadrature points. PyDG interfaces with numpy's polynomial class to construct these polynomials to an arbitrary order of accuracy. Using this functionality, PyDG has been run for up to $p = 63$ order polynomials.

C.2 Parallelization

Similar to PySpectral, PyDG is parallelized through mpi4py. The solver utilizes a pencil decomposition, where the computational domain is decomposed in the x_1 and x_2 directions. PyDG has been run in a cluster environment on up to 512 CPUs.

C.3 Efficient Implementation of Volumetric Integrations

One of the primary costs in a high order discontinuous Galerkin solver is the evaluation of the volumetric integrals. In a spacetime formulation, these integrals are

four dimensional. To develop an efficient Python code, these volumetric integrations must be performed in a loop free manner. To illustrate how this is performed in PyDG, consider the volumetric integration of a source termed f tested against the weighting functions w,

$$\mathbf{R} = \int \mathbf{w}(x,y,z,t) f(x,y,z,t) d\Omega.$$

For a tensor product basis, the weighting functions can be described as

$$\mathbf{w}(x,y,z,t) = w_0(x) \otimes w_1(y) \otimes w_2(z) \otimes w_3(t).$$

The tensor product decomposition allows the volumetric integration to be performed as a series of four one dimensional integrations. Listing C.1 shows an efficient Python implementation of this integration using numpy's tensordot function. It is noted that the function f is a nine-dimensional array (# conserved variables $\times$ number of quadrature points in $(x,y,z,t)\times$ number of elements in (x,y,z,t)). The code shown in Listing C.1 provides a concise and efficient four-dimensional volumetric integration of f tested against the weighting function w in five lines of Python code.

```
def volIntegrateGlob_tensordot(main,f,w0,w1,w2,w3):
  tmp = np.rollaxis(np.tensordot(w0*main.weights0[None,:],f,axes=([1],[1])),0,9)
  tmp = np.rollaxis(np.tensordot(w1*main.weights1[None,:],tmp,axes=([1],[1])) , 0 ,
    9)
  tmp = np.rollaxis(np.tensordot(w2*main.weights2[None,:],tmp,axes=([1],[1])) , 0 ,
    9)
  tmp = np.rollaxis(np.tensordot(w3*main.weights3[None,:],tmp,axes=([1],[1])) , 0 ,
    9)
  return np.rollaxis( np.rollaxis( np.rollaxis( np.rollaxis( tmp , -4 , 1) , -3 , 2)
    , -2, 3), -1, 4)
```

Listing C.1: Python implementation of the volumetric right hand side integral.

C.4 Matrix-Free Newton Krylov Solver

Spacetime and implicit discretization techniques lead to a nonlinear set of equations to be solved at each time-step,

$$\mathbf{R}^*(\mathbf{u}^N) = 0.$$

Solving this non-linear set of equations is made challenging by the high-dimensionality of the spacetime system and the memory requirements of the associated Jacobians. PyDG solves the non-linear set of equations using a matrix-free Newton Krylov method. Matrix-free methods leverage the fact that Krylov subspace methods only require the action of a Jacobian on a vector. This action can be performed by computing the Fréchet derivative of the residual in the direction of the vector $\mathbf{v}$,

$$\mathbf{J}\mathbf{v} = \lim_{\epsilon \to 0} \frac{\mathbf{R}^*(\mathbf{u}^N + \epsilon \mathbf{v}) - \mathbf{R}^*(\mathbf{u}^N)}{\epsilon},$$

where the Jacobian $\mathbf{J} = \mathbf{R}^*{}_{\mathbf{u}^N}$. This Fréchet derivative can be either be approximated via finite difference, or it can be computed exactly by computing the linearization of $\mathbf{R}^*$. PyDG primarily makes use of the finite difference approximation, although several exact linearizations are present. The simulation results presented in this work use the generalized minimum residual method (GMRES) as the preferred linear solver.

C.4.1 Preconditioning

Converge of the implicit solvers is accelerated by using the mass-matrix as a preconditioner. Note that, while for cuboid grids the mass-matrix resulting from the Legendre polynomials is diagonal, it is still an efficient preconditioner. The standard mass matrix is computed by,

$$\mathbf{M} = \int \mathbf{w}\mathbf{w}^T d\Omega.$$

Solution of the non-linear system is accelerated by left multiplying by the inverse mass matrix,

$$\mathbf{M}^{-1}\mathbf{R}^*(\mathbf{u}^N) = 0.$$

For formulations using entropy variables, the solver is accelerated by using the "entropy" mass matrix,

$$\mathbf{M}_s = \int \mathbf{w} \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \mathbf{w}^T d\Omega.$$

The nonlinear solvers are then accelerated by

$$\mathbf{M}_s^{-1} \mathbf{R}^*(\mathbf{u}^N) = 0.$$

It must be noted that the entropy mass matrix changes in time and must be recomputed as $\mathbf{u}$ and $\mathbf{v}$ change. In the context of preconditioning, however, the exact mass matrix is not required. In PyDG, the entropy mass matrix is instead computed at set intervals throughout the calculation.

C.5 Miscellaneous Details

PyDG utilizes the same VTK package as PySpectral, PyEVTK, to write data into a format readable by ParaView. PyDG is available on GitHub.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] H. Aluie. Hydrodynamic and Magnetohydrodynamic Turbulence: Invariants, Cascades, and Locality. *Ph.D. Thesis*, The Johns Hopkins University, Baltimore, 2009.

[2] J. L. Barber. *Application of Optimal Prediction to Molecular Dynamics.* PhD thesis, University of California, Berkeley, CA, 2004.

[3] Claude Bardos and Eitan Tadmor. Stability and spectral convergence of Fourier method for nonlinear problems: on the shortcomings of the 2/3 de-aliasing method. *Numerische Mathematik*, 129(4):749–782, 2015.

[4] F. Bassi and S. Rebay. A High Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations. *Journal of Computational Physics*, 131:267–279, 1997.

[5] Camilo Bayona, Joan Baiges, and Ramon Codina. Variational multi-scale approximation of the one-dimensional forced Burgers equation: the role of orthogonal sub-grid scales in turbulence modeling. *International Journal for Numerical Methods in Fluids*, 2017.

[6] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 197(1-4), 2007.

[7] David Bernstein. Optimal Prediction of Burgers's Equation. *Multiscale Model. Simul.*, 6(1):27–52, 2007.

[8] J.P. Boris, F.F. Grinstein, E.S. Oran, and R.L. Kolbe. New insights into large eddy simulation. *Fluid Dynamics Research*, 10:199–228, 1992.

[9] Sanjeeb Bose, Parviz Moin, and Donghyun You. Grid-independent large-eddy simulation using explicit filtering. *Physics of Fluids*, 22(10), 2010.

[10] John P. Boyd. *Chebyshev and Fourier Spectral Methods.* Dover Publications Inc., 2000.

[11] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods in Fluid Dynamics.* Springer-Verlag, 1988.

[12] Daniele Carati and Alan A. Wray. Time filtering in large eddy simulations. *Center for Turbulence Research, Proceedings of the Summer Program*, 2000.

[13] Pietro Catalano and Marcello Amato. An evaluation of RANS turbulence modelling for aerodynamic applications. *Aerospace Science and Technology*, 7(7):493–509, 2003.

[14] F.H. Champagne, C.A. Friehe, J.C. LaRue, and J.C. Wyngaard. Flux measurements, flux estimation techniques and fine-scale turbulence measurements in the unstable surface layer over land. *Journal of Atmospheric Science*, 34:515–530, 1977.

[15] Abhilash J. Chandy and Steven H. Frankel. The t-model as a Large Eddy Simulation model for the Navier-Stokes Equations. *Multiscale Model. Simul.*, 8(2):445–462, 2009.

[16] J. R. Chasnov. On the decay of two-dimensional homogeneous turbulence. *Phys. Fluids*, 9(1):171–180, 1996.

[17] Tianheng Chen and Chi-Wang Shu. Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation equations. *Journal of Computational Physics*, 345(15):427–261, 2017.

[18] A. Chorin and O. Hald. *Stochastic Tools in Mathematics and Science*. Springer-Verlag, 2005.

[19] A. Chorin, O. Hald, and R. Kupferman. Optimal prediction with memory. *Phys. D*, pages 239–257, 2002.

[20] Alexandre J. Chorin, O. Hald, and R. Kupferman. Optimal prediction and the Mori-Zwanzig representation of irreversible processes. *Proc. Natl Acad. Sci.*, 97((doi:10.1073/pnas.97.7.2968)):2968–2973, 2000.

[21] Alexandre J. Chorin and Ole H. Hald. *Stochastic Tools for Mathematics and Science*. Springer, 2013.

[22] Alexandre J. Chorin, Ole H. Hald, and Raz Kupferman. Prediction from Partial Data, Renormalization, and Averaging. *Journal of Scientific Computing*, 28(2-3):245–261, September 2006.

[23] Bernardo Cockburn and Chi-Wang Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[24] R Codina, S Badia, J Baiges, and J Principe. Variational multiscale methods in computational fluid dynamics. *Encyclopedia of computational mechanics, E. Stein, R. de Borst, and TJ Hughes, Eds., Wiley Online Library*, 2017.

[25] Ramon Codina. Stabilization of incompressibility and convection through orthogonal subscales in finite element methods. *Computer methods in applied mechanics and engineering*, 190(1579-1599), 2000.

[26] Ramon Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer methods in applied mechanics and engineering*, 191(4295-4321), 2002.

[27] SS. Collis. Monitoring unresolved scales in multiscale turbulence modeling. *Physics of Fluids*, 13(6):1800–1806, 2001.

[28] G. Comte-Bellot and S. Corrsin. Simple Eulerian time correlation of full and narrow band velocity signals in isotropic turbulence. *Journal of Fluid Mechanics*, 48:273, 1971.

[29] Lisandro D. Dalcin, Rodrigo R. Paz, Pablo A. Kler, and Alejandro Cosimo. Parallel distributed computing using Python. *Advances in Water Resources*, 34:1124–1139, 2011.

[30] Corentin Carton de Wiart, Koen Hillewaert, Phillippe Geuzaine, Robbi Luccioni, Laurent Bricteux, Gregory Coussement, and Gregoire Winckelmans. Assessment of LES modeling within a high order discontinuous Galerkin solver. In *9th International Ercoftac symposium on Engineering Turbulence Modeling and Measurements*, Thessaloniki, Greece, June 2012.

[31] JW Deardoff. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics*, 41(2):453–480, 1970.

[32] Laslo T. Diosady and Scott M. Murman. High-Order Methods for Compressible Turbulent Flows Using Entropy Variables. *53rd AIAA Aerospace Sciences Meeting*, 2015.

[33] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence Modeling in the Age of Data. *arXiv:1804.00183v1*, 2018.

[34] Karthik Duraisamy, Philippe R. Spalart, and Christopher L. Rumsey. Status, Emerging Ideas and Future Directions of Turbulence Modeling Research in Aeronautics. Technical Memorandum 219682, NASA, November 2017.

[35] Tarek Echekki and Epaminondas Mastorakos. *Turbulent Combustion Modeling*. Springer-Verlag, 1988.

[36] P. Fernandez, N.C. Nguyen, X. Roca, and J. Peraire. Implicit large-eddy simulations of compressible flows using the Interior Embedded Discontinuous Galerkin method. *arXiv:1605.01020v1*, 2016.

[37] R. Feynman, R. B. Leighton, and M. Sands. The Feynman lectures on physics. In *Addison-Wesley*, Boston, MA, 1964.

[38] Krzysztof Fidkowski. *Advanced Computational Fluid Dynamics*. Aero 623 Course Notes, February 2017.

[39] L.P. Franca, S.L. Frey, and T.J.R. Hughes. Stabilized finite element methods: 1. Application to the advective-diffusive model. *Computer methods in applied mechanics and engineering*, 95(2):253–276, 1992.

[40] M. Germano. Turbulence: the filtering approach. *Journal of Fluid Mechanics*, (238):325–336, 1992.

[41] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids*, A(3):1760–1765, 1991.

[42] Dror Givon, Ole H. Hald, and Raz Kupferman. Existence Proof for Orthogonal Dynamics and the Mori-Zwanzig Formalism. *Israel Journal of Mathematics*, 145(1):221–241, 2005.

[43] Ayoub Gouasmi, Eric J. Parish, and Karthik Duraisamy. A priori estimation of memory effects in reduced-order models of nonlinear systems using the Mori-Zwanzig formalism. *Proceedings of The Royal Society A*, 473(20170385), 2017.

[44] Philip M. Gresho and Robert L. Sani. On pressure boundary conditions for the incompressible Navier-Stokes Equations. *International Journal for Numerical Methods in Fluids*, 7:1111–1145, 1987.

[45] Oriol Guasch and Ramon Codina. Statistical behavior of the orthogonal subgrid scale stabilization terms in the finite element large eddy simulation of turbulent flows. *Computer methods in applied mechanics and engineering*, 261-262(154-166), 2013.

[46] Ole 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(16):6527–6532, 2007.

[47] Ralf Hartmann and Paul Houston. Symmetric Interior Penalty DG Methods for the Compressible Navier-Stokes Equations I: Method Formulation. *International Journal o fNumerical Analysis and Modeling*, (3):1–20, 2006.

[48] Thomas J.R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgridscale models, bubbles and the origins of stabilized methods. *Computer methods in applied mechanics and engineering*, 127:387–401, 1995.

[49] Thomas J.R. Hughes, GR Feijoo, L Mazzei, and JB Qunicy. The variational multiscale method - a paradigm for computational mechanics. *Computer methods in applied mechanics and engineering*, 166:173–189, 1998.

[50] Thomas J.R. Hughes, Luca Mazzei, and Kenneth E. Jansen. Large Eddy Simulation and the variational multiscale method. *Computing and Visualization in Science*, 2000.

[51] Thomas JR Hughes, Guglielmo Scovazzi, and Leopoldo P Franca. Multiscale and stabilized methods. *Encyclopedia of computational mechanics*, 2004.

[52] T.J.R. Hughes and G. Sangalli. Variational Multiscale Analysis: The Fine-Scale Green's Function, Projection, Optimization, Localization, and Stabilized Methods. *SIAM Journal on Numerical Analysis*, 45(2):539–557, 2007.

[53] Farzad Ismail. *Toward a reliable prediction of shocks in hypersonic flow: resolving carbuncles with entropy and vorticity control*. PhD thesis, University of Michigan, 2006.

[54] Farzad Ismail and Philip L. Roe. Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks. *Journal of Computational Physics*, 228:5410–5436, 2009.

[55] A.N Kolmogorov. Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Number, 1941.

[56] R.H. Kraichnan. Eddy Viscosity in two and three dimensions. *Journal of Atmospheric Science*, 33:1521–1536, 1976.

[57] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007.

[58] Hee Sun Lee, Surl-Hee Ahn, and Eric F. Darve. Building a Coarse-Grained Model Based on the Mori-Zwanzig Formalism. *MRS Proceedings*, 1753, 2015.

[59] Marcel Lesieur and Olivier Metais. New Trends in Large-Eddy Simulations of Turbulence. *Annu. Rev. Fluid. Mech.*, (28):45–82, 1996.

[60] Yanan Li and Z.J. Wang. A Priori and A Posteriori Evaluations of Subgrid Stress Models with the Burgers' Equation. In *53rd AIAA Aerospace Sciences Meeting*, Kissimmee, Florida, 5-9 January 2015.

[61] Zhen Li, Xin Bian, Xiantao Li, and George Em Karniadakis. Incorporation of memory effects in coarse-grained modeling via the Mori-Zwanzig formalism. *The Journal of Chemical Physics*, 143(24), 2015.

[62] D. K. Lilly. On the Application of the Eddy Viscosity Concept in the Inertial Sub-range of Turbulence. *NCAR Manuscript*, (123), 1966.

[63] D. K. Lilly. A proposed modification of the Germano subgrid-scale closure method. *Physics of Fluids*, 4:633–636, 1992.

[64] T.S. Lund. The Use of Explicit Filters in Large Eddy Simulation. *Computers and Mathematics with Applications*, 46:603–616, 2003.

[65] Dimitri Mavriplis, Elizabeth Lurie, William Gropp, David Darmofal, Juan Alonso, Abdollah Khodadoust, and Jeffrey Slotnick. CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences. Technical Report CR-2014-218178, NASA, 2013.

[66] Charles Meneveau and Joseph Katz. Grid-independent large-eddy simulation using explicit filtering. *Annu. Rev. Fluid Mech.*, 32:1–32, 2000.

[67] O. Metais and M. Lesieur. Spectral large-eddy simulations of isotropic and stably-stratified turbulence. *Journal of Fluid Mechanics*, 239:157–194, 1992.

[68] P. Moin and J. Kim. On the Numerical Solution of Time-Dependent Viscous Incompressible Fluid Flows Involving Solid Boundaries. *Journal of Computational Physics*, (35):381–392, 1980.

[69] H. Mori. Transport, collective motion, and Brownian motion. *Prog. Theoret. Phys*, 33(3):423–455, 1965.

[70] Mikael Mortensen and Hans Petter Langtangen. High performance Python for direct numerical simulations of turbulent flows. *Computer Physics Communications*, 203:53–65, 2016.

[71] R. Moser, J. Kim, and N. Mansour. Direct numerical simulation of channel flow up to Re_τ = 590. *Physics of Fluids*, 11(4):943–945, 1999.

[72] Eric Parish and Karthik Duraisamy. A Dynamic Subgrid Scale Model for LES based on the Mori-Zwanzig Formalism. *Proceedings of the Summer Program, Center for Turbulence Research, Stanford University*, 2016.

[73] Eric Parish and Karthik Duraisamy. Reduced Order Modeling of Turbulent Flows Using Statistical Coarse-graining. In *46th AIAA Fluid Dynamics Conference, AIAA AVIATION Forum*, Washington, D.C., June 2016.

[74] Eric J. Parish and Karthik Duraisamy. A dynamic subgrid scale model for Large Eddy Simulations based on the Mori-Zwanzig formalism. *Journal of Computational Physics*, 349:154–175, 2017.

[75] Eric J. Parish and Karthik Duraisamy. Non-Markovian Closure Models for Large Eddy Simulation using the Mori-Zwanzig Formalism. *Physical Review Fluids*, 2(1), 2017.

[76] Eric J. Parish and Karthik Duraisamy. A Unified Framework for Multiscale Modeling Using Mori-Zwanzig and the Variational Multiscale Method. *CMAME*, (Submitted), 2018.

[77] Per-Olof Person. Shock Capturing for High-Order Discontinuous Galerkin Simulation of Transient Flow Problems. In *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 2013.

[78] Per-Olof Persson and Jaime Peraire. Sub-Cell Shock Capturing for Discontinuous Galerkin Methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2006.

[79] J. Pike and P. L. Roe. Accelerated Convergence of Jameson's Finite-Volume Euler Scheme Using Van Der Houwen Integrators. *Computers and Fluids*, 13(2):223–236, 1985.

[80] Jonathn Poggie. Large-Scale Structures in Implicit Large-Eddy Simulation of Compressible Turbulent Flow. In *AIAA Aviation Forum*, Atlanta, Georgia, June 2014.

[81] Stephen B. Pope. Ten questions concerning the large-eddy simulation of turbulent flows. *New Journal of Physics*, 2004.

[82] Jacob Price and Panos Stinis. Renormalized Reduced Order Models with Memory for Long Time Prediction. *arXiv:1707.01955v1*, 2017.

[83] C. D. Pruett. Temporal Large-Eddy Simulation. Final Report NAG1-02033, NASA, July 2004.

[84] P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[85] Robert S. Rogallo. Numerical Experiments in Homogeneous Turbulence. Technical Memorandum 81315, NASA, September 1981.

[86] VV Rusanov. On Difference Schemes of Third Order Accuracy for Nonlinear Hyperbolic Systems. *Journal of Computational Physics*, 5:507–516, 1970.

[87] U. Schumann. Subgrid-scale model for finite difference simulation of turbulent flows in plane channels and annuli. *Journal of Computational Physics*, 85(3):429–438, 1975.

[88] Joseph Smagorinsky. General Circulation Experiments with the Primitive Equations. *Monthly Weather Review*, 91(3):99–164, March 1963.

[89] D. Sondak, J.N. Shadid, A.A. Oberai, R.P. Pawlowski, and E.C. Cyr. A new class of finite element variational multiscale turbulence models for incompressible magnetohydrodynamics. *Journal of Computational Physics*, 295(15):596–616, 2015.

[90] David Sondak and Assad A. Oberai. Large eddy simulation models for incompressible magnetohydrodynamics derived from the variational multiscale formulation. *Physics of Plasmas*, 19(10):102308, 2012.

[91] Katepalli R. Sreenivasan. On the universality of the Kolmogorov constant. *Physics of Fluids*, 7(11), 1995.

[92] Panagiotis Stinis. Higher order Mori-Zwanzig models for the Euler equations. *Multiscale Model. Simul.*, 6(3):741–760, 2007.

[93] Panos Stinis. Mori-Zwanzig reduced models for uncertainty quantification I: Parametric uncertainty. *arXiv:1211.4285*, 2012.

[94] Panos Stinis. Renormalized reduced models for singular PDEs. *Commun. Appl. Math. Comput. Sci.*, 8(1):39–66, 2013.

[95] Panos Stinis. Renormalized Mori-Zwanzig-reduced models for systems without scale separation. *Proceedings of The Royal Society A*, 471(20140446), 2015.

[96] E. Tadmor. Entropy functions for symmetric systems of conservation laws. *Journal of Mathematical Analysis and Applications*, 122(2):355–359, 1987.

[97] E. Tadmor. The numerical viscosity of entropy stable schemes for conservation laws. I. *Mathematics of Computation*, 49(91-103), 1987.

[98] G. I. Taylor and A. E. Green. Mechanism of the Production of Small Eddies from Large Ones. *Proc. R. Soc. Lond. A*, 158:499–521, 1937.

[99] AW Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids*, 16(10), 2004.

[100] Z. Wang and A.A. Oberai. Spectral analysis of the dissipation of the residual-based variational multiscale method. *Comput. Methods Appl. Mech. Engrg.*, 199(13-16), 2010.

[101] David C Wilcox. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA, 1998.

[102] H-C Yee, N. Sandham, and M. Djomehri. Low dissipative high order shock-capturing methods using characteristic based filters. *Journal of Computational Physics*, 150(1):199–238, 1999.

[103] Yan Zhang, Robert L. Street, and Jeffrey R. Koseff. A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows. *Physics of Fluids*, 5(3186), 1993.

[104] Yuanran Zhu, Jason M. Dominy, and Daniele Venturi. Rigorous error estimates for the memory integral in the Mori-Zwanzig formulation. *arXiv:1708.02235v2*, 2017.

[105] Yuanran Zhu and Daniele Venturi. Faber approximation to the Mori-Zwanzig equation. *arXiv:1708.03806v2*, 2017.

[106] R. Zwanzig. Memory Effects in Irreversible Thermodynamics. *Phys. Rev.*, 124(4):983, 1961.

[107] R. Zwanzig. *Nonequilibrium Statistical Mechanics*. Oxford University Press, 2001.