Mathematical Formalisms and Data-Driven Approaches for Coarse-Graining of Multi-Scale Finite Element Discretizations

by

Aniruddhe Pradhan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2022

Doctoral Committee:

Professor Karthik Duraisamy, Chair
Professor Krzysztof Fidkowski
Professor Krishna Garikipati
Associate Professor Eric Johnsen

Aniruddhe Pradhan

anipra@umich.edu

ORCID iD: 0000-0002-1905-184X

For my parents.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Prof. Karthik Duraisamy who allowed me to work at the Computational Aerosciences Lab (CASLAB). He has been an integral part of my journey and has helped me in every possible way, both technically and outside of research. I am incredibly grateful to him for giving me all the flexibility to work on the topics of my interest while ensuring that it fits the bigger project goals. He has been very patient with me in reviewing all my conference presentations and research papers and helping me to improve them by orders of magnitude. Prof. Karthik's excellent academic speaking abilities have always inspired me to become a better presenter. I am also thankful to Prof. Karthik and my previous lab members for bringing so much exposure to CASLAB. As a result, I was offered two internship opportunities even without formally applying to them.

I want to thank my committee members, Prof. Fidkowski, Prof. Garikipati, and Prof. Eric Johnsen, for all their insightful comments and help in preparing this document. I thank Prof. Fidkowski for his technical insights and valuable comments during the bi-weekly project meetings and for being such a great teacher. I would also like to thank Prof. Eric Johnsen for taking my first class at the University of Michigan in Fall 2016 and making it a memorable experience.

I have been fortunate to have the company of friends who have made my stay in Michigan exciting and stress-free. I want to thank my cool first-year roommates, Arjun and Sagardip, who made my transition into a new country seamless. I cannot thank Sagardip enough for the countless free-destressing car rides to random locations in and around Michigan. I

want to thank Nandan, Arun, Subramaniam, and Anubhav for all the discussions on CFD and fluid dynamics. Special thanks to Nandan for keeping me stress-free by sending funny memes frequently. Thanks to Bhargav for all the trips to the Pontiac trail and the "Curry on" Wednesdays. Thanks to Prashant for all his advice on job applications and interviews. Thanks to Jophin for being such a great roommate and for introducing me to Kerala's cuisine. I cannot thank Aaditya enough for trying out my experimental food dishes. Thanks to Aditya Sunder for doing all the spicy food challenges with me. Thanks to Rishav and Jishnu for the road trips and the innumerable memories since high school. Thanks to Souvik for all the distressing discussions. Finally, I thank all the CASLAB members for making this journey memorable. Thanks to Shaowu and David for trying to teach me some basic Chinese. Special thanks to Shaowu for accompanying me to Cardamom and Evergreen for lunches and listening to my ideas patiently. I thank Vishal and Rajarshi for all the discussions on CFD, adjoints, life, and beyond.

Lastly, I would like to thank my parents, who have been my constant pillar of support. Thank you for supporting all my decisions and believing in me. Thank you for being so patient and considerate for not visiting you for four years. Thank you for fueling my interest in science and all the sacrifices you made for me.

This research was funded by NASA under the project "Scale-resolving turbulence simulations through adaptive high-order discretizations and data-enabled model refinements", grant number 80NSSC18M0149 (Technical monitor: Dr. Gary Coleman), and by the AFOSR under the project "LES Modeling of Non-local effects using Statistical Coarse-graining", grant number FA9550-16-1-0309.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Numerical simulations of multi-scale problems remain challenging in many applications due to complex interactions between the resolved and unresolved scales. Effective computations of these problems require coarse-grained models that approximate the impact of the fine scales in terms of the coarse scales. This work develops coarse-grained modeling strategies that leverage the structure of the underlying partial differential equations and formal projections of the available high-dimensional data to discover closures and augment existing model forms.

First, a coarse-grained modeling approach for Galerkin discretizations is developed by combining the Variational Multi-scale decomposition and the Mori–Zwanzig (M–Z) formalism. An appeal of this approach is that – akin to Green's functions for linear problems – the impact of unresolved dynamics on resolved scales can be formally represented as a convolution (or memory) integral in a non-linear setting. A parameter-free dynamic version of the MZ-VMS model is then developed for the continuous Galerkin method and assessed in detail in coarse-grained simulations of a range of problems from the one-dimensional Burgers equation to incompressible turbulence.

Second, the VMS sub-scale model forms discussed in the first part are rewritten in a non-dimensionalized form to generate a neural network (N-N) model form and a set of generalizable local non-dimensional input and output features. These features, along with the model structure, are embedded into a special N-N called the variational super-resolution N-N (VSRNN), providing a general framework for the data-driven discovery of closures for various Galerkin discretizations. It is further demonstrated that for linear

problems, our formulation reduces the problem of learning the sub-scales to one of learning the basis coefficients of the projected element Green's function. By training the VSRNN network on a sequence of L_2-projected data, and using the super-resolved state to compute the discontinuous Galerkin fluxes, improvement in the optimality and the accuracy of the method is obtained for both the linear advection problem and turbulent channel flow problem. The model is also shown to extrapolate to out-of-sample initial conditions and Reynolds numbers.

Finally, closure model discovery through formal projections of the high-dimensional data proposed in the second part is extended to the near-wall region. This resulted in the development of a unified framework that can be used as a lens to quantitatively assess and augment a wide range of coarse-grained models of turbulence, viz. large eddy simulations (LES), hybrid Reynolds-averaged/LES methods, and wall-modeled (WM)LES. Taking a turbulent channel flow as an example, optimality is assessed in the wall-resolved limit, the hybrid RANS/LES limit, and the WMLES limit via projections at different resolutions suitable for these approaches. These optimal a priori estimates are shown to have similar characteristics to existing a posteriori solutions reported in the literature. Consistent accuracy metrics are developed for scale-resolving methods using the optimal solution as a reference, and evaluations are performed.We further characterize the slip velocity (a form of sub-scale) in WMLES in terms of the near-wall under-resolution, and develop a universal scaling relationship for the slip wall model coefficient, which is used to augment existing slip wall models. Various a posteriori tests reveal superior performance over the dynamic slip wall model.

Overall, this dissertation develops mathematical formalisms and data-driven tools that enable the development of generalizable coarse-grained models for a wide range of multi-scale problems and allows for an objective assessment and augmentation of existing closures.

CHAPTER I

Introduction

1.1 The multi-scale phenomenon.

The multi-scale phenomenon is ubiquitous in science and engineering and is characterized by disparate ranges of spatio-temporal scales and complex interactions therein. If the separation between the smallest and the largest scales is sufficiently large, the number of degrees of freedom required to describe the system can become computationally intractable. Hence, to reduce computational complexity, multi-scale modeling is required. The applications of multi-scale modeling range from protein conformational dynamics to astrophysics. One problem where the multi-scale phenomenon is strongly observed is fluid flow. Figure 1.1 shows the hierarchy of multi-scale models used to model fluid-flow problems. The most straightforward approach is to track the movement of each molecule, i.e., molecular dynamics. This approach becomes computationally intractable for large systems. In most applications, the position of each molecule is not important, and the quantities of interest are the average quantities such as the velocity, the pressure, or the distribution functions. Multi-scale modeling aims to reduce computational complexity by deriving new governing equations for the quantities of interest without losing considerable predictive accuracy. This process is called *coarse-graining*, and the resulting model is called a *coarse-grained model*.

Figure 1.1 also shows the various equations used to describe fluid flows at different levels of approximation. Moving from from right to left, the computational cost increases, and

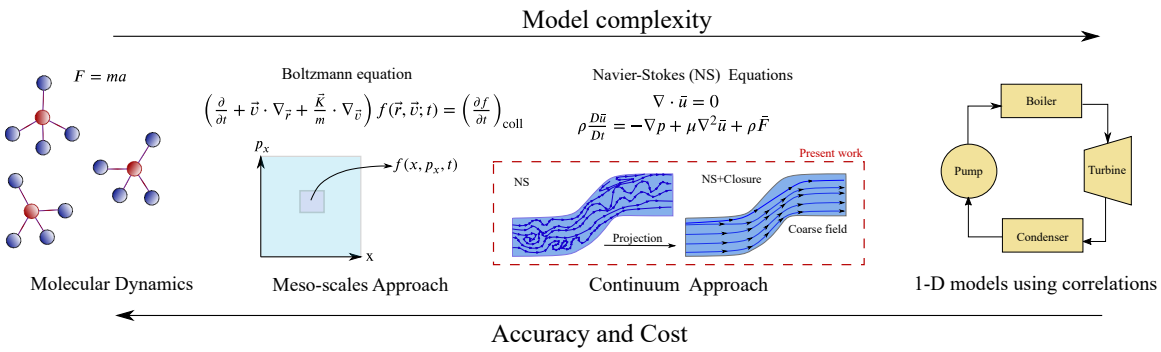


Figure 1.1: Descriptions of fluid-flow at different levels of multi-scale approximations.

the governing equations become simpler yet computationally more complex and have more physics embedded. Moving from left to right, i.e., coarse-graining, is not easy. As will be seen later in this thesis, whenever coarse-graining is performed, the effect of the unresolved physics has to be represented in terms of the coarse-graining variables only. This is called the *closure* problem. For example, the collision term in the Boltzmann equation, which models the effect of collisions between molecules on the particle distribution functions, is a form of closure approximated in terms of the distribution functions. Similarly, the constitutive relationship that links the viscous fluxes to the velocity gradients is another form of closure.

Once a closure is developed, the coarse-grained equations govern the evolution of the coarse-grained variables as long as the coarse-graining assumptions are valid and the closure is accurate. For example, the Navier-stokes equations are valid under the assumption that the molecular mean free path λ_m is much smaller than the length scale of the fluid flow problem, and the constitutive relations for the viscous fluxes are valid.

However, the multi-scale phenomenon is not just restricted to the hierarchy shown in Figure 1.1. Nature has cursed us with more hierarchy levels within the solutions to the coarse-grained equations without violating any of the modeling assumptions. For example, the solutions to the Navier-Stokes equation under the conditions of high Reynolds numbers show desperate ranges of spatial and temporal scales while still satisfying the assumptions of continuum mechanics. This multi-scale structure, in these cases, is purely a product

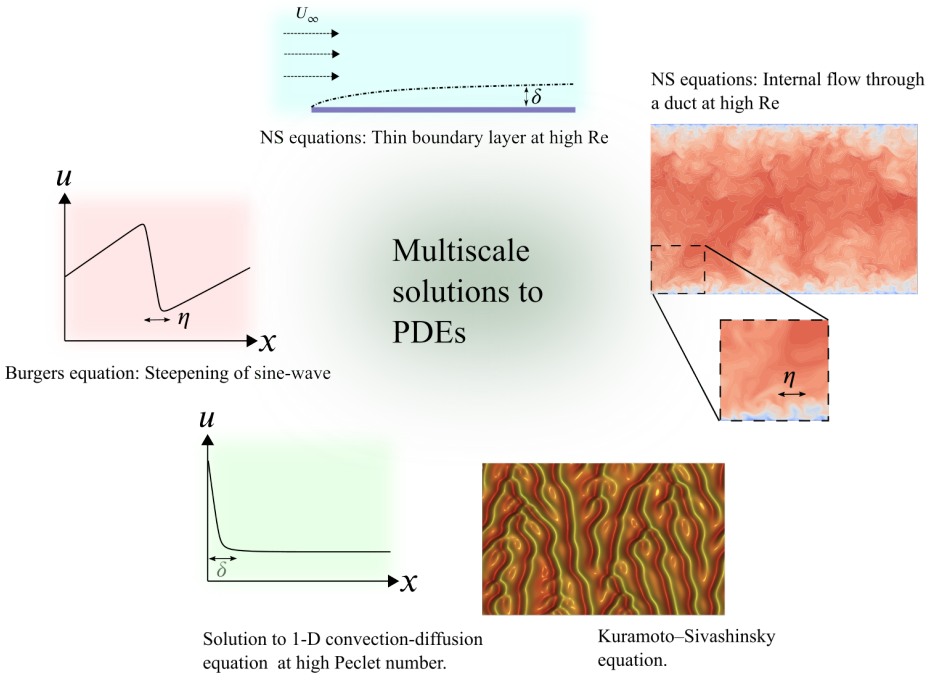


Figure 1.2: Multi-scale solutions to PDEs.

of the PDEs and not the modeling assumption used to obtain the PDE. In this thesis, the aim is to develop coarse models for similar PDEs to ensure efficient computation of their solutions.

1.2 Multi-scale nature of PDEs.

In the previous section, a discussion was presented on how a multi-scale hierarchy could be present in the solution of the PDE itself. This multi-scale hierarchy present in the solution of PDE is similar to the outer hierarchy shown in figure 1.1, of which the PDE is originally a part. However, the physics governing this multi-scale behavior is self-contained in the PDE without violating any modeling assumptions used to arrive at the PDE. Figure 1.2 shows the solution for some of the PDEs that exhibit multi-scale nature. It can also be observed from Figure 2 that the multi-scale behavior is not limited to non-linear PDEs, suggesting that linear PDEs also require coarse-grained modeling. In all these PDEs, the multi-scale behavior is controlled by parameters such as the Reynolds number or the Peclet number.

When the Peclet number or Reynolds number is sufficiently high, the smallest length or time scales present in the flow are much smaller than the problem's geometrical length and time scales. As a result, an extremely fine grid and time-step are required to resolve these flows. However, in most of these cases, only specific quantities, such as the mean solution, the motion of the large structures, or the solution outside the boundary layer, are of interest. Hence, for all practical purposes, the multi-scale modeling of PDEs is essential.

1.2.1 Turbulence.

Turbulence is a multi-scale phenomenon exhibited by the solutions to the Navier-Stokes equations under high Reynolds number conditions. Although a precise definition does not exist, the flow is considered to be turbulent if:

1. The flow is three-dimensional, non-linear, irregular, and chaotic.
2. A range of scales is present. This is driven by the balance between inertial forces and viscous forces.
3. Continuous cascade of energy from large-scale structures to small-scale structures where it is dissipated as heat through the action of viscosity.
4. Increased mixing of mass, momentum, and energy.
5. They are rotational, i.e., there is a presence of non-zero vorticity.

These properties make turbulence an interesting but extremely hard problem to understand or solve computationally. Richard Feynman called turbulence "the most important unsolved problem of classical physics.". Historical paintings like "The Starry Night" (1889) by Van Gogh and "The Great Wave off Kanagawa" by Hokusai (1830) have depicted turbulence and its features through art.

1.3 Resolution requirement for multi-scale PDEs.

The previous section discussed how specific parameters such as the Peclet number or Reynolds numbers trigger multi-scale behavior in PDEs. In this section, the resolution requirement for the Navier-Stokes equations will be quantified. Similar estimates can also be obtained for other multi-scale PDEs. The resolution requirement depends on two different factors mainly:

1. The accuracy of the numerical method. An accurate numerical scheme would require a lesser number of degrees of freedom.

2. The size of the smallest length or the smallest time scale dictated by the physics of the problem.

First, the effect of the numerical method is discussed. In any numerical method, the solution is sampled at a fixed number of points, nodes, etc. The distance between the points or nodes along a direction i, is an estimate of the grid size Δx_i in that direction. In general, the physics of the problem dictates the the size of the smallest scales (λ_i) present in the solution along a particular direction i. Estimates of the number of grid points *Kawai and Larsson* (2012) required to resolve these length scales are given by:

$$\lambda_i \geq \beta \Delta x_i \tag{1.1}$$

where β is the grid points per wavelength. The value of β depends on the numerical scheme used. A lower β implies a better numerical method. However, β cannot be so small that it violates the Nyquist theorem.

Second, an estimate of λ_i for turbulent flows is obtained using Kolmogorov's hypothesis. As shown in figure 1.3 and discussed previously, in turbulence, a continuous cascade of energy from large-scale structures (low-wavenumber) to small-scale structures (high-wavenumber) is present. The amount of energy cascaded is governed by the size of the

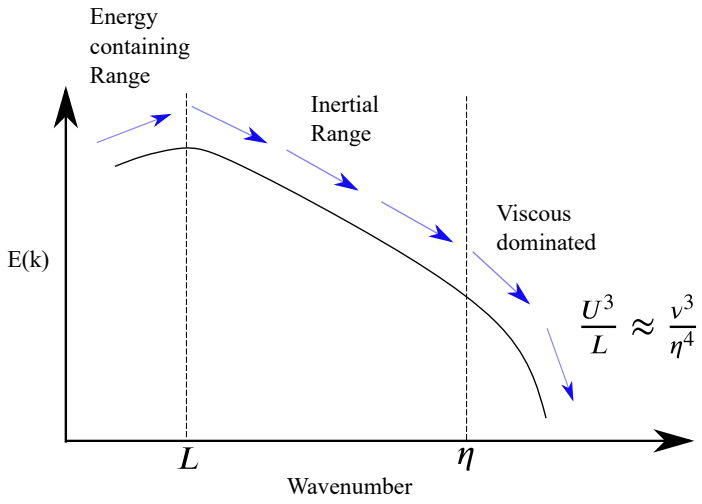


Figure 1.3: Energy cascade in turbulence.

large-scale structures that span the geometry L and the velocity scale U of the problem. By equating the amount of energy that is cascaded from the large scales to that dissipated as heat at the smallest length scale λ_i through viscous dissipation, an estimate of λ_i can be obtained as follows:

$$\frac{\lambda_i}{L} \approx \frac{\eta}{L} \approx \frac{1}{Re^{3/4}} \tag{1.2}$$

where Re is based on the size of the geometry L and the velocity scale U i.e. $Re = \frac{UL}{\nu}$. The estimated size of small scales, i.e., λ_i is also called the Kolmogorov's length scale and is denoted by η in the literature.

Finally, by combining equations 1.1 and 1.2 the following grid point estimate is obtained:

$$N_i \approx \frac{L}{\Delta x_i} \approx \beta Re^{3/4}. \tag{1.3}$$

The above estimate is for a single direction. Turbulence being a 3-D phenomenon, the actual number of spatial degrees of freedom can be obtained as follows:

$$N_{dof} = N_i^3 \approx \beta^3 Re^{9/4}. \tag{1.4}$$

Setting the CFL number C constant to reduce time-stepping errors and ensure stability (for explicit methods), the following estimate of the number of time steps required to perform the time integration for one flow through the time scale is obtained:

$$N_t = \frac{L}{U\Delta t} = \frac{L}{C\Delta x_i} \approx \frac{\beta Re^{3/4}}{C} \tag{1.5}$$

The approximate number of operations N_{ops} required to solve the problem is given by

$$N_{ops} \approx N_{dof} N_t \approx \frac{\beta^4 Re^3}{C} \tag{1.6}$$

Hence, the accuracy of the numerical method can only play a small role when the Reynolds number of the problem becomes excessively large. Taking the case of a commercial jet ($Re \approx 10^8$), fixing the CFL number to 1, and assuming that an extremely accurate spatial scheme with $\beta \approx 2$ is used, a grid with approximately 2000000^3 grid points is required and 2000000 time-steps are needed to be taken to obtain the solution. Combined with the multi-physics nature of these problems, these problems are intractable on any existing supercomputer.

1.4 Modeling multi-scale PDEs.

In the previous section, it was shown that it is impossible to solve the PDEs numerically for many of the problems of our interest. In this section, various coarse-grain modeling techniques are discussed that are used to reduce their computational complexity.

1.4.1 General Idea.

The general idea in all these coarse-graining techniques is to start with the PDE; for example

$$\frac{\partial u}{\partial t} + R(u) = 0, \tag{1.7}$$

and applying a coarse-graining operator $\mathcal{G}$ to obtain the following:

$$\mathcal{G}\frac{\partial u}{\partial t} + \mathcal{G}R(u) = 0. \tag{1.8}$$

Assuming time invariance of the operator and defining $\mathcal{G}u$ as u_h the following equation is obtained:

$$\frac{\partial u_h}{\partial t} + R(u_h) + [\mathcal{G}R(u) - R(u_h)] = 0 \tag{1.9}$$

In general, u_h can be represented on a much coarser mesh than u. Our goal is to obtain a PDE that governs the evolution of the coarse-grained variable u_h. Consequently, the new modified PDE can be resolved well on the coarse grid. The first part of the equation resembles the original PDE given in equation (1.7) except it has been replaced with u_h. The computation of the additional term $[\mathcal{G}R(u) - R(u_h)]$ still requires the access to u. The final step is to approximate approximates $[\mathcal{G}R(u) - R(u_h)]$ in terms of u_h only i.e.

$$\frac{\partial u_h}{\partial t} + R(u_h) + C(u_h) = 0 \tag{1.10}$$

This step is called *closure modelling* and $C(u_h)$ is called a *closure*. Equation 1.10 represents the coarse-grained model in the strong form. As will be seen later in the thesis, based on what coarse-graining method has been employed, the coarse-grained equation can also exist in a weak form. Based on what the coarse-graining operator $\mathcal{G}$ represents, equation 1.10 can represent the governing equations for different coarse-grained methods. For example, in the case of fluid flow, if the action of $\mathcal{G}$ on the flow variables represents their time average for a statistically stationary flow or their ensemble average for a case where repeatable transient simulations can be performed, then $C(u_h)$ represents a Reynolds-Averaged Navier-Stokes (RANS) closure.

Similarly, if the effect of $\mathcal{G}$ is the same as that of a filter with a filter width that is approximately the size of the grid, then $C(u_h)$ represents a sub-grid closure. Consequently,

the effect of smaller scales than the grid needs to be modeled; hence, the term "sub-grid" is used. In this dissertation, the primary focus will be on sub-grid scale modeling. The sub-grid models for fluid-flow problems are also known as large eddy simulation (LES) models. However, as will be later seen in this thesis, the actual closure might be a mix and match of both the RANS and the LES models.

1.4.2 Sub-grid modeling.

As discussed previously, sub-grid modeling refers to a particular type of coarse-grained modeling approach where only the scales that can be well represented on the grid (coarse-scales) are resolved, whereas the effect of the high frequency (wavenumber) oscillations on the coarse-scales are modeled. As a consequence, the cost of simulating the PDE is reduced because the coarse-grained solution can now be solved on a much coarser grid. However, this does not imply that the grid can be as coarse as possible. This section will discuss this limit in the context of large eddy simulations for turbulent flows.

1.4.2.1 Large eddy simulation.

Large eddy simulation (LES) is an essential tool for performing scale-resolving simulations of the atmospheric boundary layers, combustion, acoustics, plasma, aerodynamics, and many other significant problems. The word "large eddy" is derived from the large-scale flow structures (eddies) which are resolved in these simulations. Although LES is cheaper in comparison to DNS, it remains an expensive tool for most problems. However, the recent increase in the computational power of most computers and the emergence of high-performance clusters (HPCs) have led to the wider adoption of LES, both by the industry and academia. By 2030, NASA plans to perform wall-modeled LES (WMLES) of an entire aircraft configuration *Slotnick et al.* (2014).

To demonstrate the action of LES, consider the energy spectra shown in figure 1.4. In RANS, the large-scale mean flow structures are resolved. As a result, the impact of all the

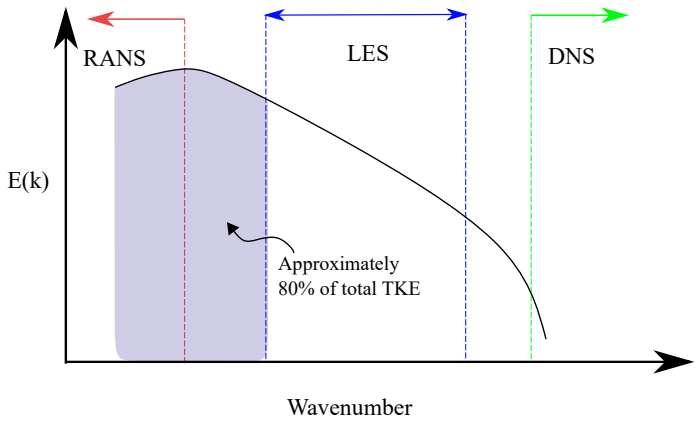


Figure 1.4: Distinction between turbulence modeling approaches: DNS, LES and RANS.

scales (entire spectrum) needs to be modeled. In LES, scales until the inertial range of the turbulence spectrum are resolved and the rest small residual scales that are responsible for the dissipation of energy into heat are modeled. In DNS, all the scales are resolved, including the small dissipative scales. In the case of high-Reynolds turbulence, the inertial spectrum can be quite large, i.e., the wavenumber can vary across orders of magnitude. Hence, to reduce the computational complexity, the filter width for LES should be placed in the low-wavenumber range of the inertial spectrum. A general rule of thumb to obtain accurate LES results is to choose the filter width such that it resolves 80 % of the total turbulent kinetic energy (TKE). However, this puts an upper limit on the computational cost reduction that LES can provide.

1.5 Ideal properties of a sub-grid model.

Various sub-grid models have been successfully applied to a wide range of turbulent flow problems. As discussed by *Parish* (2018), traditional sub-grid approaches can be broadly categorized into:

- Implicitly-filtered implicit LES.
- Implicitly-filtered explicit LES.

- Explicitly-filtered implicit LES.
- Explicitly-filtered explicit LES.

Readers are encouraged to read *Parish* (2018) for a detailed overview of these methods. In addition to traditional LES methods, methods such as the variational multiscale method (VMS) and the approximate deconvolution method, have also been used successfully. For a detailed description of the VMS methods, readers are encouraged to jump to Chapter 2 directly. For each of these sub-grid models, there exists both advantages and disadvantages that make them suitable for a problem. However, there is no consensus within the sub-grid modeling community about which approach works best for a general class of problems. There are several parameters to judge a sub-grid model. In the next part, a discussion on some properties that are desired in a sub-grid model is provided and the existing sub-grid models are rated based on each point.

1.5.1 Formalism-driven approach.

The Navier-Stokes equations were derived around the 1800s. A century later, the first conceptual framework for turbulence was given by Kolmogorov in 1941. Twenty years later, the first sub-grid model based on the Kolmogorov theory was proposed by Smagorinsky to simulate atmospheric air current (1963). After nearly 30 years, the dynamic versions were introduced (*Germano et al.*, 1991; *Meneveau et al.*, 1996). As history has shown, it takes a lot of time and effort to understand the physics of a PDE system and develop a sub-grid model based on it. Although the physics-based techniques provide many insights and have been successfully tested on many fluid flow problems, they are non-existent or extremely hard to derive for many multi-physics problems (e.g., combustion) where the Navier-Stokes equations are solved along with other PDEs. Given the multitude of such multi-physics problems, it is impossible to understand the physics of every problem and propose a physics-based sub-grid model. Hence, a *formal* strategy is required, which

allows the generation of the sub-grid model directly from the PDE without requiring any phenomenological assumption.

Different formal techniques have been successfully applied to derive subgrid models. One such formal approach is the approximate deconvolution approach by *Stolz and Adams* (1999); *Stolz et al.* (2001), where the closure is directly computed using the de-convoluted variables. To compute the de-convoluted variables, an approximate form of the filter inverse does not require the knowledge of any turbulence theory. Implicit LES also qualifies as a formal approach since it relies on the numerical dissipation of the method, and explicit sub-grid modeling is not required. Another class of formal approaches is based on the variational multi-scale (VMS) (*Hughes et al.*, 1998b). In Chapter 3, a novel model reduction strategy for non-linear problems is derived that combines the VMS method with the Mori-Zwanzig formalism.

1.5.2 Perfect scale decomposition.

In the traditional LES approach (also known as implicitly-filtered explicit LES), to obtain the coarse-grained PDE, a filtering/coarse-graining operation is performed on the original PDE as shown in equation (1.9). The sub-grid term i.e. $[\mathcal{G}R(u) - R(u_h)]$ is then modeled using a explicit sub-grid closure (for e.g. the Smagorinsky model). However, the definition of $\mathcal{G}$ is not required in this derivation. It is assumed that the action of the explicit sub-grid closure would ensure that the sub-grid fine scales are not present and the coarse-grained variables are implicitly filtered.

The sub-grid model controls the size of the smallest flow structures that are resolved, which are approximately equal to the size of the grid. As a result, the discretization errors in LES can be roughly the same order of magnitude as the sub-grid modeling errors. The different treatment of equal-in-size sub-grid modeling errors (sub-grid model) and discretization errors (numerical scheme) in implicitly-filtered explicit LES makes it challenging to define what the LES solution represents. This is because the true coarse-

grained PDE that is solved is modified due to the numerical method and, consequently, the coarse-grained solution as well. Hence, G is not known a priori and is dependent on both the numerical method and the sub-grid closure. Additionally, the sub-grid model depends on the grid size. As a result, when the mesh size is reduced, the PDE system is changed, and consequently, the coarse-graining operator G also changes, which makes it hard to define convergence and assess the performance of the LES model.

Alternatively, an another filter $\mathcal{G}_2$ can be derived and applied to the non-linear term at every time step as follows:

$$\frac{\partial u_h}{\partial t} + \mathcal{G}_2 R(u_h) + [\mathcal{G}R(u) - R(u_h)] = 0. \tag{1.11}$$

By filtering the non-linear term, the spectral content of the advective term is brought down to the $\mathcal{G}_2$ level *Lund* (2003). Consequently, the spectral content of all the terms is brought down to the $\mathcal{G}_2$ level. If the size of $\mathcal{G}_2$ is larger than the grid size, the small scales that are difficult to be represented on the grid can be damped, and the truncation error can be reduced. Hence, the coarse-scale solution approximately satisfies the relationship $\mathcal{G}_2 u \approx u_h$. Another approach where the distinction between the coarse and fine-scale is exact is the variational multiscale methods (VMS). As will be later seen in Chapter 2, the solution and weighting spaces in VMS is decomposed as follows:

$$\mathcal{V} = \mathcal{V}_h \oplus \mathcal{V}', \tag{1.12}$$

The next step is to choose a projection operator which gives the best representation of u on $\mathcal{V}_h$ based on some optimality. Hence, it is known before solving the coarse-grained system what part of the true solution u is being resolved and not resolved in the simulation. As a consequence, there is no distinction between the sub-grid modeling errors and the discretization errors, and their effect jointly is modeled through a single closure. In Chapter 3 and Chapter 4, based on this idea of perfect scale decomposition, VMS closures will be

developed using mathematical formalism and data, respectively. The idea of perfect scale decomposition will be again exploited in Chapter 5 to obtain optimal estimates of u_h for various coarse-grained methods for wall-bounded flows.

1.5.3 Accurate closure development.

The goal of closure modeling is to approximate the subgrid term as accurately as possible i.e.

$$\mathcal{G}R(u) - R(u_h) \approx C(u_h) \tag{1.13}$$

For transient PDEs, the true fine scales are both functions of the coarse space variables and their time-history *Stinis* (2007); *Parish and Duraisamy* (2017a,b,c); *Pradhan and Duraisamy* (2020). Similarly, for non-linear transient PDEs, the closure not only depends on the coarse-grained variables' time history but also on the solution of an extremely high-dimensional PDE called the orthogonal dynamics equation *Parish and Duraisamy* (2017a,b); *Gouasmi et al.* (2017); *Parish and Duraisamy* (2017c). Hence, even if an analytical form of closure exists, the cost of computing it might be comparable to or more expensive than solving the original system. Hence, an approximate form of closure is required for all practical purposes. If the approximate sub-grid model accurately captures the actual sub-grid stresses, it automatically qualifies as an excellent sub-grid model.

It is rare in turbulent fluid flow problems that the sub-grid model is accurate in capturing the actual sub-grid stresses. This is because the fine scales and the sub-grid stresses are strongly dependent on the current state and the entire time history. However, most sub-grid model forms depend entirely on the coarse-field at time t to predict the sub-grid stresses at time t. It is often argued how close the modeled sub-grid stress should be to the actual sub-grid stress. As described in *Meneveau and Katz* (2000), most of the time, the interest is not so much in the detail that the DNS solution provides but only in the ability of LES to generate the correct flow statistics, such as the mean and r.m.s. of the flow variables. Hence, it is acceptable that the actual sub-grid stresses are different from their modeled

form as long as the first and second-order statistics of the flow are accurate.

For the accurate computation of the first and second-order statistics, one of the quantities the sub-grid model should accurately model is the mean dissipation rate of the resolved kinetic energy. By modeling this rate correctly, sub-grid models such as the Smagorinksky model work excellently on many problems without showing a high correlation coefficient between true and the modeled sub-grid stresses *Meneveau and Katz* (2000). Another interpretation of an accurate sub-grid model is based on the optimal LES framework by *Langford and Moser* (1999a). In optimal LES, it is possible to construct an abstract sub-grid model that can obtain correct single-time, multi-point statistics and generates a minimal error in the instantaneous dynamics. Here, the LES model is written as a conditional average, an average over all the instantaneous fields that correspond to the same LES solution when filtered. As a result, the modeled sub-grid stresses do not necessarily have to show a high spatial correlation with the actual sub-grid stresses; however, it still preserves predictive accuracy.

1.5.4 Problem adaptivity.

A common problem in most of the sub-grid closures is the presence of free parameters. The optimal value of these free parameters depends on the problem and the numerical method. The classic example is the constant coefficient Smagorinky model. Many different values have been reported for the Smagorinsky coefficient C_s ranging from 0.1 *Deardorff* (1970), 0.15 (*Pope and Pope*, 2000), 0.17 (*McMillan and Ferziger*, 1979), to 0.23 (*Lilly*, 1966) for various problems. In addition to C_s being problem dependent, its value can also vary in space. For example, in a wall-bounded flow, the value of C_s goes to zero at the wall. Better model forms *Vreman* (2004); *Nicoud and Ducros* (1999); *Nicoud et al.* (2011) have also been proposed, which only require the specification of a single global coefficient that is constant in space. Sub-grid models derived using the MZ formalism and fixed memory model also require the specification of a memory length τ. Hence, a mechanism is

required to dynamically adjust these coefficients in both space and time without specifying these parameters explicitly. This approach is called dynamic sub-grid modeling. The first dynamic version of the Smagorinsky model was introduced by *Germano et al.* (1991) and extended to complex geometry by *Meneveau et al.* (1996). A global dynamic modeling approach for the VREMEN model was also introduced by *You and Moin* (2007b), which enabled automatic adjustment of the VREMEN model coefficient in time. Most of these approaches are based on Germano's identity, which exploits the relationship between the sub-grid stresses and the non-linear terms at two different filter levels to obtain the model parameter. In Chapter 3, a dynamic version of an MZ-VMS sub-grid closure will be introduced for the continuous Galerkin finite element method that uses the Germano's identity to obtain the memory length dynamically.

1.5.5 Stability.

The sub-grid model should ensure the stability of the coarse-grained simulation. Most sub-grid models are dissipative, i.e., they take out energy from the system and prevent its buildup at the high wave-number modes. For example, the Smagorinsky model adds a positive non-linear viscosity to the original system. Hence, the effect of the Smagorinsky model is mostly stabilizing. However, this is not always the case. The absolute deconvolution method *Stolz and Adams* (1999); *Stolz et al.* (2001) which is based on the idea of using the filter inverse to compute the non-linear terms, is not stable without a dissipative relaxation term. Another non-dissipative closure is the scale-similarity model. The sub-grid stresses obtained using the scale-similarity model show a high correlation with the actual sub-grid stresses; however, it fails to keep the coarse-grained simulation stable. The scale-similarity model is always used in its mixed form by blending it with the Smagorinsky model *Bardina et al.* (1980); *Liu et al.* (1994, 1995); *Vreman et al.* (1994); *Xie et al.* (2019b) to ensure stability. In addition to the scale-similarity model, the dynamic Smagorinsky model is susceptible to instability if the predicted model coefficient is negative. This is true near the

wall, where back-scattering can happen, and the model coefficient can become negative. This problem can be fixed by capping the value to zero to prevent the model coefficient from becoming negative. Sub-grid models trained using data can also suffer from numerical instability. This is because the stability of the final closure is generally not guaranteed during the training process. This is the case in Chapter 5, where our data-driven closure was unstable and needed to be under-relaxed with a known stable method.

1.5.6 Computational efficiency.

The cost of a coarse-grained simulation largely depends on the complexity of the closure and the numerical method used. The cost of computing the closure can account for a significant fraction of the computational cost. The most cost-effective methods are based on the implicit LES approach. In this approach, the numerical dissipation present in the scheme is assumed to model turbulence, and no ad-hoc treatment in the form of a sub-grid model is required. In addition to the implicit LES approach, the variational multiscale method also does not require any ad-hoc treatment for turbulence since it does not distinguish between the effect of the small-scale structures in turbulence and the unresolved numerical fine-scales it attempts to model. Among explicit LES models, the constant coefficient-based models *Smagorinsky* (1963); *Nicoud and Ducros* (1999); *Nicoud et al.* (2011); *Vreman* (2004) are the easiest to implement and most cost-effective. The dynamic version of these models is generally known to be more accurate. However, they are more expensive due to the test-filtering operation, which is required to obtain the model coefficient dynamically.

Additionally, the model coefficient obtained through the dynamic procedure varies rapidly and requires additional averaging in space/time, which adds to the computational cost. The averaging procedure is convenient when there are homogeneous directions, as in the case of channel flow (stream-wise and span-wise) or flow over a 2D airfoil (span-wise). For complex geometries with no homogeneous directions, the averaging procedure is challenging and requires the specification of local regions where the averaging needs to

be performed. A robust way of dynamic modeling in the case of complex geometries is to perform the averaging over the trajectory of the fluid particle *Meneveau et al.* (1996). However, this requires the solution of two additional transport equations, further adding to the computational cost. In review, the cost of sub-grid modeling depends on the case's complexity, the sub-grid model's accuracy, and the user's expertise in specifying model parameters.

1.5.7 Model consistency.

In the age of data, machine learning has become an important tool for learning closures directly from data. In a data-driven approach the goal is to learn a function $C_{ML}(u_h)$ that approximates the sub-grid terms $\mathcal{G}R(u) - R(u_h)$ i.e.

$$\mathcal{G}R(u) - R(u_h) \approx C_{ML}(u_h). \tag{1.14}$$

This approach was restricted in the past due to the lack of availability of high-resolution data and computational resources. One approach is to compute the sub-grid term $\mathcal{G}R(u) - R(u_h)$ and learn it as a functions of u_h and its derivatives. However, it is not possible to learn the closure precisely and some modeling error is always present i.e.

$$\mathcal{G}R(u) - R(u_h) = C_{ML}(u_h) + \epsilon. \tag{1.15}$$

When this model is used online as a sub-grid model, the effect of the modeling error ϵ on the quantities of interest is not guaranteed to be favorable. Hence, it is essential to train the closure to minimize the difference between the actual value and predicted values of the quantities of interest rather than between the predicted and the actual sub-grid stresses. This is called a *model consistent* approach where the model is consistent with the solver in which it will be used. Model consistency is an important property to have in a data-driven sub-grid model. However, there are many challenges.

The first problem is regarding the model consistency of sub-grid models in the scale-resolving simulations of chaotic problems. For these cases, if the sensitivity of the model weights is taken with respect to the solution at a far enough time in the future, there is a chance that these sensitivities might blow up. Hence, gradient-based optimization approaches cannot be applied to train the models. The second problem is regarding the sensitivity of these scale-resolving subgrid models to the numerical method. If a sub-grid model is developed in a *model consistent* setting, the model might get over-fit to the numerical method used in the solver. This problem can be solved by keeping multiple solvers with different numerical methods in the training loop or switching to the VMS formalism, where the sub-grid model is tied to the numerical method. In this dissertation, the model-consistent approach is not considered. However, a discussion on model consistency is provided at the end of Chapter 5.

1.6 Coarse-graining wall-bounded flows.

Although LES provides huge computational advantages in the case of many canonical problems, such as homogeneous isotropic turbulence, it fails to provide a considerable computational advantage when applied to wall-bounded flows. The lack of computational benefits in the case of wall-bounded flows results from the size of the energy-containing scales near the wall that scale with the viscous length scale $O(\delta_v)$, which eventually, in the log layer region, increases approximately linearly with the distance from the wall. A schematic of the energy-containing eddies is shown in figure 1.5. Further, if the Reynolds number of the flow is extremely high, the viscous length scale δ_v is much smaller than the boundary layer thickness δ. As shown in figure 1.5, to satisfy the 80 % criteria, the wall-resolved LES mesh needs to have a grid size comparable to δ_v near the wall and can increase to δ near the edge of the boundary layer. As a result, many grid points are required in the inner layer leading to the cost of wall-resolved LES (WRLES) being comparable to DNS. The term "wall-resolved" is used because the energy-containing structures near

the wall are also resolved. Based on the latest estimate by *Yang and Griffin* (2021), for external aerodynamics, the grid requirement for performing DNS, wall-resolved (WR)LES, and wall modeled (WM)LES are $O(Re^{2.05})$, $O(Re^{1.86})$ and $O(Re)$ respectively. For this case, the cost of DNS is not very different from LES. Among the three approaches, the WMLES method has the least complexity. The WMLES approach and similar cost-effective approaches for simulating wall-bounded turbulence will be explained next.

A wide range of approaches have been proposed to perform cost-effective computations of wall-bounded flows at high Reynolds numbers. The two most commonly used approaches are the wall-stress/slip-based wall-modeled LES (WMLES) approach and the hybrid RANS-LES approach. These methods are based on the idea of performing traditional LES far away from the wall and transitioning to a RANS-based description near the wall. As shown in figure 1.5, the grid sizes (or the filter sizes) for the hybrid RANS-LES case in the span-wise and stream-wise directions are comparable to the semi-channel height or the boundary layer thickness δ ($\approx 0.1\delta$). Figure 1.5 also shows the representative energy spectra at two different wall-normal locations. The spectrum near the wall is shifted to the right because more energy is present in the higher frequency wave numbers near the wall than in the center of the channel. As a result, the grid size required to capture 80% of the energy near the wall requires a finer mesh.

If a filter size of approximately 0.1δ (hybrid RANS-LES) is used in the span-wise and stream-wise direction, the grid at the center of the channel can capture a good fraction of the total turbulent kinetic energy to qualify as an LES mesh. However, the exact resolution fails to capture the required amount of energy near the wall. Consequently, the effect of almost the entire spectrum must be modeled in the near-wall region, especially at high Reynolds numbers. Hence, the RANS method is more suitable for the near-wall region in this case. As discussed earlier, the RANS approach is not required for the wall-resolved case because the mesh size is smaller near the wall to ensure 80% is captured throughout, and traditional sub-grid models can be used.

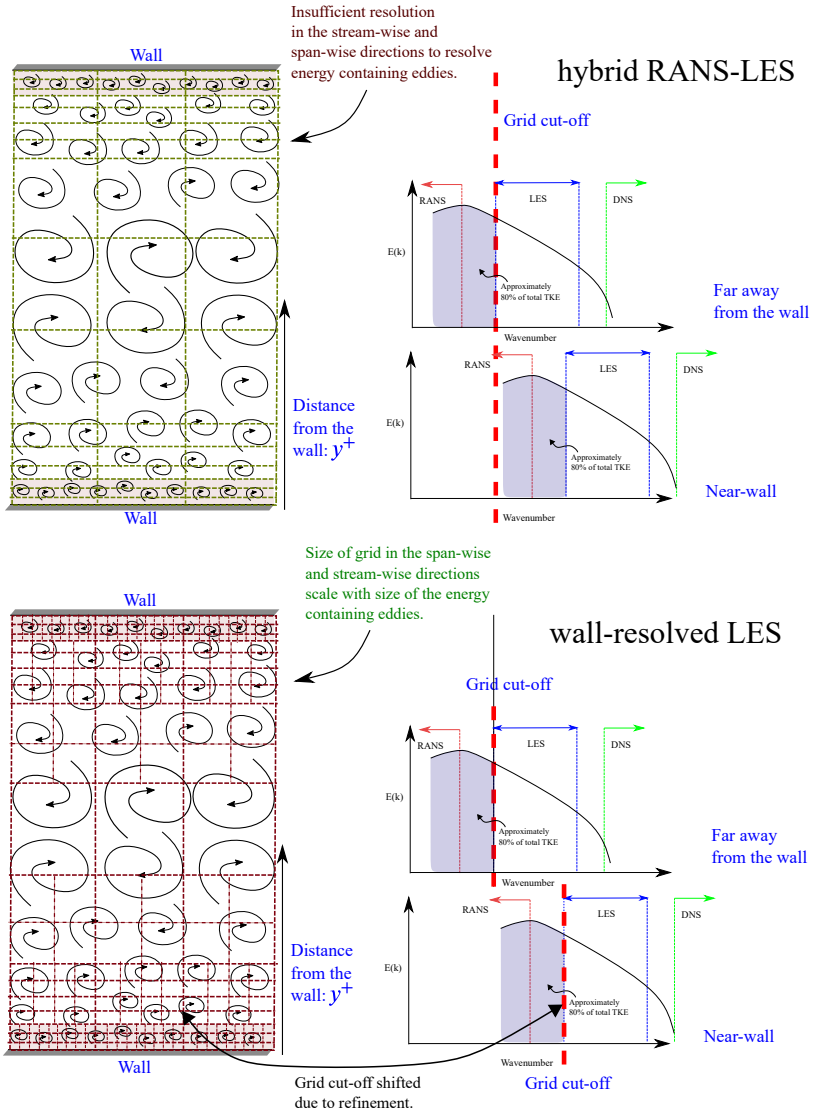


Figure 1.5: The size of the energy containing eddies in a wall bounded flow increase with the distance from the wall. The use of filter sizes bigger than the size of eddies that contribute significantly to the TKE necessitates RANS-like closure modelling approaches.

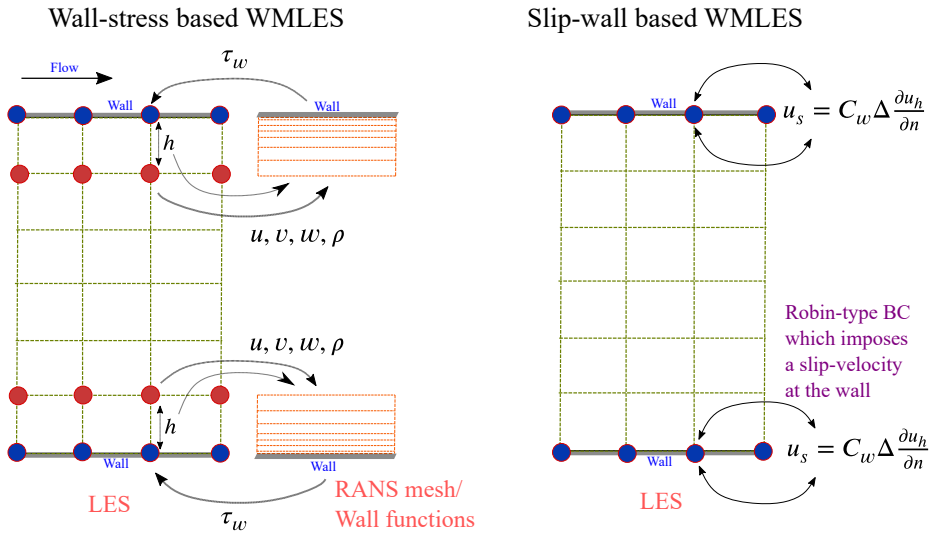


Figure 1.6: The traditional wall-stress and slip-wall based WMLES approaches.

An alternate approach to the hybrid RANS-LES approach is wall-modeled LES (WMLES) approach. As shown in figure 1.6, there are mainly two types of WMLES approaches. The first approach is the wall-stress-based WMLES approach, where the RANS and LES components are solved on two separate meshes. The RANS mesh exists primarily near the wall and receives velocity information from the off-wall grid points on the LES mesh. RANS is then performed on this mesh, and the wall-shear stress is computed. The wall-shear stress is then passed onto the LES grid as a Neumann boundary condition. Alternatively, the RANS mesh can be replaced with a wall function that describes the solution in the inner layer under equilibrium conditions. The second WMLES approach is the slip-wall-based WMLES approach, where a Robin boundary condition is specified at the wall. This approach does not require the sampling of the velocity fields at the off-wall grid points and works by imposing a slip velocity. As will be shown in Chapter 5, the exact slip velocity needed to be imposed depends on the solution of the RANS equations and the numerical method used to perform the LES.

1.6.1 The need for a special treatment for wall-bounded flows.

Ideally, no a priori knowledge of near-wall physics should be required to develop models using the formalism-based approaches. However, irrespective of whether the model is developed using phenomenological assumptions or formal techniques, the accuracy of the final closure largely depends on the final modeling assumptions. For example, the Smagorinsky model assumes that the modeled rate of sub-grid production for the turbulence kinetic energy transfer from large to small scale balances dissipation, which is wrong when the filter size used is huge near the wall. Similarly, as will be seen later in Chapter 3, the finite memory M-Z model, which is used to model the memory term in the Mori-Zwanzig identity, assumes that the memory integral is highly correlated with the value of the integral at $s = 0$. These modeling assumptions might fail again when a large filter size is used near the wall. In fact, the finite-memory-based M-Z closure for the spectral Fourier Galerkin method can be proved to be globally dissipative. This property, although stabilizing, is not representative of the small-scale dynamics when a large filter size is used near the wall. In summary, given the current state of memory modeling for the M-Z formalism, an ad-hoc modeling approach for the near-wall region might still be required. This is the motivation behind the wall-modeling work presented in Chapter 5, where the poor performance of some of the existing wall models has been investigated, and better model forms have been suggested.

1.7 Presence of a unified structure.

In the previous sections, different types coarse-grained modeling techniques were introduced. In this section, some of the existing coarse-grained modeling approaches are compared, and commonalities are drawn between them. To this end, consider one of the

simplest PDEs (i.e., the linear-advection equation) as follows:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0. \tag{1.16}$$

It is well understood that the discretization of the convective term with a central scheme makes it susceptible to instability for linear advection. The semi-discrete form for this case is given by

$$\frac{\partial u}{\partial t}|_k + a\frac{u_{k+1} - 2u_k + u_{k-1}}{2\Delta x} + O(\Delta x^2) = 0 \tag{1.17}$$

where u_k denotes the solution at k^{th} grid point. A more stable formulation is obtained by using points in the upwind direction to compute the convective term as follows:

$$\frac{\partial u}{\partial t}|_k + a\frac{u_k - u_{k-1}}{\Delta x} = 0, \tag{1.18}$$

for $a > 0$. Expanding u_{k-1} using the Taylor series, one can obtain:

$$\frac{\partial u}{\partial t}|_k + a\frac{\partial u}{\partial x}|_k = \frac{a\Delta x}{2}\frac{\partial^2 u}{\partial x^2}|_k + O(\Delta x^2). \tag{1.19}$$

The additional term $\frac{a\Delta x}{2}\frac{\partial^2 u}{\partial x^2}|_k$ can be considered to be a sub-grid model for equation (1.16) to improve its stability. A Galerkin discretization of equation (1.16) results in the following:

$$\left(\frac{\partial u_h}{\partial t} + a\frac{\partial u_h}{\partial x}, w_h\right)_{\Omega'} = 0 \quad \forall w_h \in \mathcal{V}_h. \tag{1.20}$$

Equation (1.20) is also susceptible to instability without a proper closure. A stream-wise upwind Petrov Galerkin (SUPG) based closure for equation (1.20) is given by:

$$\left(\frac{\partial u_h}{\partial t} + a\frac{\partial u_h}{\partial x}, w_h\right)_{\Omega'} + \sum_K \int_K \tau a^2 \frac{\partial w_h}{\partial x}\frac{\partial u_h}{\partial x} d\Omega' = 0. \quad \forall w_h \in \mathcal{V}_h \tag{1.21}$$

An expression for τ for the SUPG method is given by:

$$\tau = \frac{\Delta x}{2a}. \tag{1.22}$$

Substituting back in equation (1.21) the following equation is obtained:

$$\left(\frac{\partial u_h}{\partial t} + a\frac{\partial u_h}{\partial x}, w_h\right)_{\Omega'} + \sum_K \int_K \frac{a\Delta x}{2}\frac{\partial w_h}{\partial x}\frac{\partial u_h}{\partial x} d\Omega' = 0. \quad \forall w_h \in \mathcal{V}_h \tag{1.23}$$

Equation (1.23) can also be obtained directly by the Galerkin discretization of the modified PDE given by equation (1.19). Further comparing equation (1.23) and (1.19), the following relationship between τ and the equivalent artificial viscosity ν_e that is added to the original PDE, is obtained:

$$\nu_e = \tau a^2. \tag{1.24}$$

Hence, the effect of the stabilization terms present in the VMS method is identical to adding numerical dissipation to stabilize finite difference methods for advection-dominated flows. The VMS-ϵ approach by *Wang and Oberai* (2010a) and the constant-τ based MZ method also exhibit an identical model form irrespective of the fact that they were derived independently using two different approaches. We will also show in Chapter 5 that there is an equivalence between the traditional wall-stress-based wall model and the slip-wall-based wall model. The similarity between these methods suggests that a universal structure generally exists in all these coarse-grained models independent of the method.

1.8 Dissertation setting

The outline of the dissertation is as follows: We introduce the relevant mathematical formalisms and data-driven techniques in Chapter 2. Chapter 3 introduces the MZ-VMS framework and derives continuous Galerkin (CG) MZ-VMS closures for non-linear multi-

scale PDEs. Chapter 4 develops a data-driven framework that can be used for closure modeling of both the continuous and discontinuous Galerkin (CG and DG) discretizations of multi-scale PDEs. This framework is used to discover high-order CG closures for the 1-D convection-diffusion problem and DG closures for both the linear-advection and the turbulent channel flow problems. Chapter 5 uses optimal finite element projections of the DNS data obtained from turbulent channel flow simulations at different Reynolds numbers to develop a unified understanding of scale-resolving simulations and perform the near-wall modeling of turbulent flows.

Overall, the goal of this dissertation is to advance the state-of-art in coarse-grained modeling of multi-scale PDEs by proposing novel formalism-driven and data-driven frameworks that lend themselves to generalization and also improve our understanding of the under-resolved simulation of wall-bounded flows. This is made possible by answering the following questions:

1. How can ideas from the Mori-Zwanzig method and the variational multiscale (VMS) method be combined to derive continuous Galerkin finite element closures directly from the structure of the PDE?

2. Is there a link between the existing super-resolution work and the fine-scale solution from the variational multiscale method?

3. Can data from high-resolution simulations be used efficiently to derive generalizable closures? If yes, what role does non-dimensionalisation play in their generalisability?

4. Can projection onto finite element spaces give new insights into under-resolved turbulence simulations of wall-bounded flows?

5. Can the no-slip boundary conditions be fully satisfied for under-resolved turbulent flow simulations?

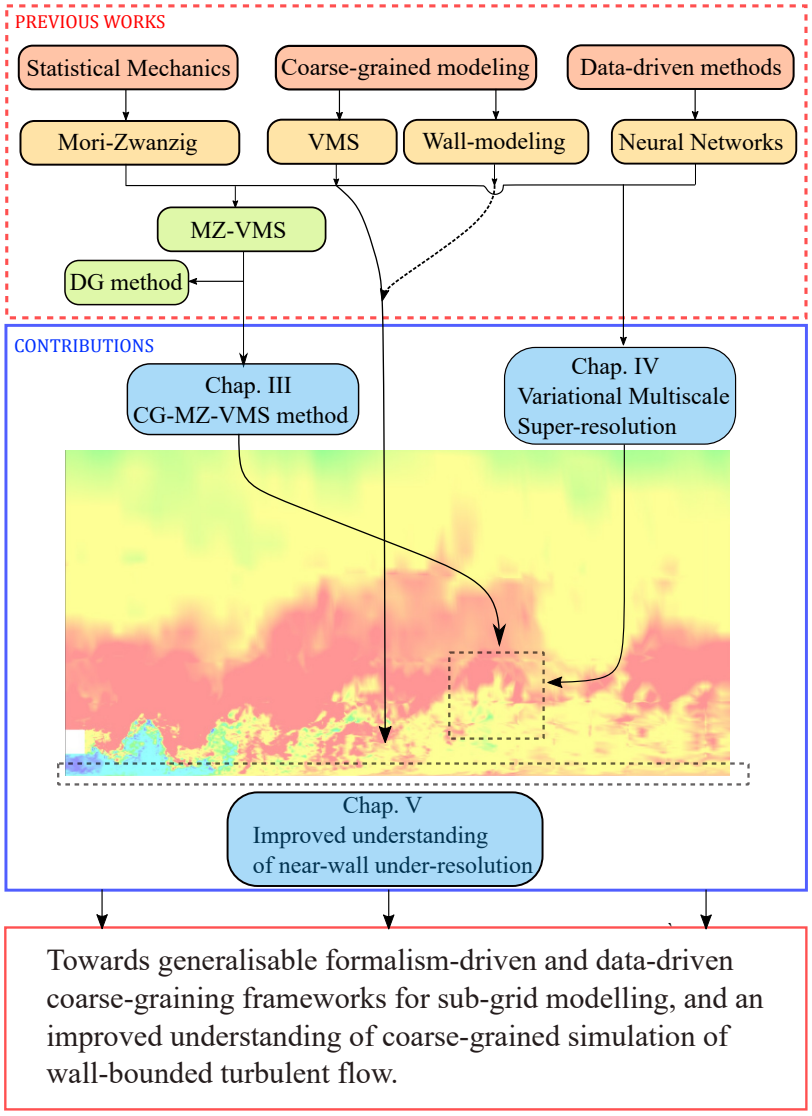


Figure 1.7: Sketch of contributions in this dissertation.

1.9 Contributions

To address the aforementioned questions, the key contributions (fig. 1.7) of this dissertation are as follows:

1. A non-linear model reduction strategy for the continuous Galerkin (CG) finite element method based on the MZ-VMS framework proposed by *Parish and Duraisamy* (2017c).

2. A dynamic-τ based MZ-VMS closure that adapts to the instantaneous level of resolution to avoid the imposition of an ad-hoc memory length for the MZ-VMS-CG framework was developed.

3. A link between the operator that super-resolves the coarse-grained solution of PDEs and the fine-scale Green's operator in the VMS method was established.

4. A unique neural network structure - the variational super-resolution N-N (VSRNN) - was developed to construct a super-resolved model consistent with existing analytical VMS closures.

5. Generalizable data-driven VMS closures were developed for both the continuous and the discontinuous Galerkin methods. This work marks the first time a generalizable closure for finite elements has been directly learned from data.

6. An optimal finite element projections technique was developed that acts as a lens to quantitatively assess and augment a wide range of coarse-grained turbulence models, viz. large eddy simulations (LES), hybrid Reynolds-averaged/LES methods, and wall-modeled (WM)LES.

7. A link between near-wall under-resolution and the slip velocity obtained in WMLES was established, and new slip-wall model forms were proposed.

8. A new slip-wall model form was proposed that generalises to different resolutions and Reynolds numbers with minimal variation in the model coefficient for the channel flow problem.

These contributions are also presented in the form of the following publications:

1. Pradhan, Aniruddhe, and Karthik Duraisamy. "Variational multiscale closures for finite element discretizations using the Mori–Zwanzig approach." Computer Methods in Applied Mechanics and Engineering 368 (2020): 113152. *Pradhan and Duraisamy* (2020)

2. Xu, Jiayang, Aniruddhe Pradhan, and Karthikeyan Duraisamy. "Conditionally parameterized, discretization-aware neural networks for mesh-based modeling of physical systems." In Advances in Neural Information Processing Systems. 2021. *Xu et al.* (2021)

3. Pradhan, Aniruddhe, and Karthik Duraisamy. "Variational Multi-scale Super-resolution: A data-driven approach for reconstruction and predictive modeling of unresolved physics." arXiv preprint arXiv:2101.09839 (2021). (Submitted to International Journal for Numerical Methods in Engineering) *Pradhan and Duraisamy* (2021)

4. Pradhan, Aniruddhe, and Karthik Duraisamy. "A unified understanding of scale-resolving simulations of turbulent flows using optimal projections." arXiv preprint arXiv:2207.13060 (2022). (Under revision, JFM) *Pradhan and Duraisamy* (2022)

CHAPTER II

Mathematical Formalisms and Data-Driven Techniques

2.1 Introduction

As discussed in Chapter 1, a considerable amount of time and effort is required to understand the physics of a PDE system and transfer this knowledge to develop accurate coarse-grained models. We further discussed how the different multi-physics problems arising in different aspects of fluid dynamics necessitate formal strategies for coarse-grained model development. In this chapter, we will discuss two of these formal tools: (i.) the Mori-Zwanzig (M-Z) formalism; (ii.) the variational multiscale (VMS). These tools are called *mathematical formalisms* because they are a set of formal mathematical instructions that require minimal knowledge of the physics of the problem. The physics knowledge required by these formalisms is limited to the governing PDE or an ODE system obtained after the semi-discretization of the PDE. This chapter aims to familiarize the readers with some of these formal tools that will be extensively used in Chapters 3,4, and 5.

In addition to mathematical formalisms, another class of tools leverage data. The availability of data from experiments and high-resolution numerical simulations has sparked new interest in these tools. However, these tools need to be efficient to ensure that they can learn the complex nonlinear relationships between the closure terms and the resolved variables. Additionally, these tools should be flexible enough to balance model performance and complexity easily. The neural network is a function approximation that has demon-

strated excellent performance on complex problems and allows seamless variation of model complexity through hyperparameters turning. Hence, neural networks are extensively used in Chapter 4 to learn various closures.

As we will see later in this dissertation, it is hard to segregate formal techniques from data-driven methods precisely. For example, in Chapter 3, although the model form is developed by combining the M-Z formalism with the VMS method, the coarse-graining exponent required to implement the dynamic model is obtained by fitting data. In Chapter 4, the sub-scale model is a neural network model. However, the input and output features of the model are derived by taking inspiration from existing VMS closures. Similarly, in Chapter 5, the model form for the wall sub-scale (slip), derived formally by assuming a differential filter, is assessed and improved using data.

Although a mix-and-match approach works best for these tools, we have separately discussed them in this chapter for convenience. To this end, the outline of the chapter is as follows: In the first half of the chapter, the M-Z formalism and the VMS method are introduced. The second half of this chapter introduces data-driven techniques, specifically neural networks.

2.2 Mathematical formalisms.

2.2.1 The Mori-Zwanzig (M-Z) formalism

2.2.1.1 Linear Dynamical System - An Example

In this section, we introduce the general principles of the M-Z formalism *Chorin et al.* (2002). The concept of M-Z was first introduced in the context of statistical mechanics *Mori* (1965); *Zwanzig* (1980) but was later extended by Chorin *Chorin et al.* (2002) to more generalized systems. To demonstrate the basic idea of M-Z, we introduce it in a simple linear dynamical system with two degrees of freedom. Following this, we present the generalization of this concept to non-linear systems via the Generalized Langevin

Equations (GLEs).

2.2.1.2 Linear Dynamical System - An Example

Consider a dynamical system with two degrees of freedom given by

$$\frac{dx}{dt} = A_{11}x + A_{12}y \tag{2.1}$$

$$\frac{dy}{dt} = A_{21}x + A_{22}y, \tag{2.2}$$

where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ are the state space, and time $t \in (0, T]$ with initial conditions: $x(0)$ and $y(0)$ provided. Our aim is to write an exact evolution equation for just one variables, say x, i.e.

$$\frac{dx}{dt} = A_{11}x + F(x). \tag{2.3}$$

Using the appropriate integration factor and integrating Equation (2.2) we have the following equation in x:

$$\frac{dx}{dt} = A_{11}x + A_{12}\int_0^t e^{A_{22}s}A_{21}x(t-s)ds + A_{12}e^{A_{22}t}y(0). \tag{2.4}$$

Equation (2.4) has three terms: (i.) the first term represents the Markovian term containing the resolved variable; (ii.) the second term is the memory integral; and (iii.) the third term represents the dependence on the initial condition of y on x. Although Equation (2.4) represents the evolution of x without any dependence on the second variable y, the flow of x at any point of time not only depends on the current values of x but also on its history weighted by an exponential factor. The two variable Markovian system is now converted to a one variable non-Markovian system without loss of accuracy. For coarse-grained model development for x, Equation (2.4) requires the inclusion of closure for the memory integral

term.

2.2.1.3 The Generalized Langevin Equation

Although the discussion in the previous section was limited to a linear system, the M-Z formalism can be generalized to non-linear problems as well. To this end, consider the spatial discretization of a space-time problem leading to the following set of N coupled ODEs in time:

$$\frac{d\phi}{dt} = R(\phi), \tag{2.5}$$

where $\phi = \{\hat{\phi}, \tilde{\phi}\}$, $\hat{\phi} \in \mathbb{R}^M$ and $\tilde{\phi} \in \mathbb{R}^{N-M}$ are the modes we want to resolve and model respectively. The choice of spatial discretization can be of non-tailored basis such as spectral methods *Parish and Duraisamy* (2017a,b), continuous, and the discontinuous Finite Element (FE) basis functions *Parish and Duraisamy* (2017c) or tailored basis obtained from purely data driven techniques such as the proper orthogonal decomposition (POD) *Parish et al.* (2018). By assuming the initial condition of the problem to be ϕ_0, we aim to solve for $\hat{\phi}$ without solving for the un-resolved modes $\tilde{\phi}$ to reduce the computational cost. However, non-linearity restricts us from using the integration factor approach previously used for linear system of ODEs. The Mori-Zwanzig approach *Chorin et al.* (2002); *Chorin and Hald* (2009); *Chorin et al.* (2000) circumvents this problem by allowing us to cast the above non-linear problem (Equation (2.5)) in the form of a linear PDE in the space of initial condition variables ϕ_0 and time t as follows

$$\frac{\partial}{\partial t} u(\phi_0, t) = \mathcal{L} u(\phi_0, t), \tag{2.6}$$

where the Liouville operator $\mathcal{L}$ corresponding to Equation (2.5) is defined as

$$\mathcal{L} = \sum_{k=1}^{N} R_k(\phi_0) \frac{\partial}{\partial \phi_{0k}}, \tag{2.7}$$

and initial conditions $u(\phi_0, 0) = g(\phi(\phi_0, 0))$, where $g : \mathbb{R}^N \rightarrow \mathbb{R}$ is a scalar valued observable. Equation (2.6) can be shown to have the following solution *Chorin et al.* (2002); *Chorin and Hald* (2009):

$$u(\phi_0, t) = g(\phi(\phi_0, t)). \tag{2.8}$$

The above equation can be re-written using the semi-group notation as follows

$$u(\phi_0, t) = e^{t\mathcal{L}} g(\phi(\phi_0, 0)) = g(\phi(\phi_0, t)), \tag{2.9}$$

where $e^{t\mathcal{L}}$ is called the Koopman operator, which is an infinite dimensional linear operator which when applied to an observable $g(\phi_0, 0)$ evolves it in time t. As a special case, the observables are chosen to be the same as the initial states $g(\phi_0) = \phi_{0j}$. As a consequence, Equation (2.6) along with Equation (2.9) results in the following

$$\frac{\partial}{\partial t} e^{t\mathcal{L}} \phi_{0j} = \mathcal{L} e^{t\mathcal{L}} \phi_{0j} = e^{t\mathcal{L}} \mathcal{L} \phi_{0j}, \tag{2.10}$$

where the last equality is a result of commutative property *Chorin and Hald* (2009) between $\mathcal{L}$ and $e^{t\mathcal{L}}$. Next, we decompose the right hand side of Equation (2.10) into spaces of resolved initial conditions and un-resolved initial conditions as follows:

$$\frac{\partial}{\partial t} e^{t\mathcal{L}} \phi_{0j} = e^{t\mathcal{L}} \mathcal{P} \mathcal{L} \phi_{0j} + e^{t\mathcal{L}} \mathcal{Q} \mathcal{L} \phi_{0j}, \tag{2.11}$$

where $\mathcal{P} : L^2 \rightarrow \hat{L}^2$ is the projection operator, where the spaces formed by all initial conditions and resolved initial conditions are denoted by L^2 and $\hat{L}^2$ respectively and operator $\mathcal{Q}$ is defined as $\mathcal{Q} = I - \mathcal{P}$. Different forms of projectors $\mathcal{P}$ can be used *Parish and Duraisamy* (2017a,b); *Gouasmi et al.* (2017). In the present work, we use a truncation projector *Chorin and Hald* (2009); *Parish and Duraisamy* (2017c) which when applied to

the function $f(\hat{\phi}_0, \tilde{\phi}_0)$ results in the truncation of the unresolved initial conditions $f(\hat{\phi}_0, 0)$. The projector $\mathcal{P}$ acts on the space formed by initial conditions and should not be confused with the L^2-projectors commonly used to project onto finite dimensional spaces. By applying Duhamel's formula *Chorin and Hald* (2009),

$$e^{t\mathcal{L}} = e^{t\mathcal{Q}\mathcal{L}} + \int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}ds, \tag{2.12}$$

in Equation (2.11), which is equivalent to the integration factor approach for linear ODEs, we obtain the generalized Langevin equation (GLE) *Chorin et al.* (2002); *Chorin and Hald* (2009); *Chorin et al.* (2000) also known as the Mori-Zwanzig identity:

$$\frac{\partial}{\partial t}e^{t\mathcal{L}}\phi_{0j} = e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\phi_{0j} + e^{t\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\phi_{0j} + \int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\phi_{0j}ds. \tag{2.13}$$

An important observation is that Equation (2.13) has a similar structure to Equation (2.4). The first term is the Markovian term, the second term is the noise due to uncertainty in the initial condition, and the last term is called the memory integral. The noise term which is given by $\mathcal{F}_j(\phi_0, t) = e^{t\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\phi_{0j}$ satisfies the orthogonal dynamics *Chorin et al.* (2002); *Zhu and Venturi* (2018) equation given by

$$\frac{\partial}{\partial t}\mathcal{F}_j(\phi_0, t) = \mathcal{Q}\mathcal{L}\mathcal{F}_j(\phi_0, t). \tag{2.14}$$

It can also be shown *Chorin and Hald* (2009) that the noise $\mathcal{F}_j(\phi_0, t)$ lies in the null space of the projector $\mathcal{P}$ i.e. $\mathcal{P}\mathcal{F}_j(\phi_0, t) = 0$. As a result, application of $\mathcal{P}$ on Equation (2.13) results in the following simplification

$$\frac{\partial}{\partial t}e^{t\mathcal{L}}\hat{\phi}_{0j} = e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\hat{\phi}_{0j} + \int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}e^{s\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\hat{\phi}_{0j}ds. \tag{2.15}$$

Equation (2.15) is exact and governs the evolution of the resolved modes $\hat{\phi}$ without any dependence on the unresolved modes. However, it does not lead to reduction in the overall computational cost as the evaluation of the memory term requires the solution to the orthogonal dynamics equation (Equation (2.14)) which is a high-dimensional PDE and is intractable to solve. However, this marks the starting point for deriving coarse-grained models based on different approximations *Chorin and Hald* (2009); *Chorin et al.* (2000); *Stinis* (2015, 2012); *Parish and Duraisamy* (2017b); *Zhu and Venturi* (2018) to the memory term. Examples on application of this formalism for non-linear model reduction of toy problems can be found in *Chorin et al.* (2002).

2.2.2 The Variational Multiscale Method

We now present a brief overview of the variational multiscale (VMS) method, which was originally formalized by Hughes et al. *Hughes et al.* (1998a) . Consider the following PDE on an open and bounded domain $\Omega \subset \mathbb{R}^d$, where $d \geq 1$ is the dimension of the problem, with a smooth boundary $\Gamma = \partial\Omega$:

$$\frac{\partial u}{\partial t} + \mathcal{R}(u) - f = 0, \tag{2.16}$$

where the operator $\mathcal{R} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ can be both linear or non-linear, the function $f : \Omega \rightarrow \mathbb{R}$ and the time varying from $t \in (0, T]$. Let, $\mathcal{V} \equiv \mathcal{H}^1(\Omega)$ denote the Sobolev space containing square integral functions with square integral derivatives. We define the variational problem as follows:

$$(\frac{\partial u}{\partial t}, w) + (\mathcal{R}(u), w) = (f, w), \tag{2.17}$$

find $u \in \mathcal{V}$ for all $w \in \mathcal{V}$, where $(\cdot, \cdot)$ denotes the L_2 inner product. The solution and weighting space are decomposed as follows:

$$\mathcal{V} = \mathcal{V}_h \oplus \mathcal{V}', \tag{2.18}$$

where $\oplus$ represents a direct sum of $\mathcal{V}_h$ and $\mathcal{V}'$. From the perspective of a numerical method, $\mathcal{V}_h$ is the resolved finite dimensional space and $\mathcal{V}'$ represents the space of functions which is not resolved. This leads to a decomposition for u and w:

$$u = u_h + u', \quad \text{and} \quad w = w_h + w', \tag{2.19}$$

where $u_h, w_h \in \mathcal{V}_h$ and $u', w' \in \mathcal{V}'$. By substituting Equation (2.19) into the variational problem given by Equation (2.17) the following is obtained,

$$(\frac{\partial(u_h + u')}{\partial t}, w_h + w') + (\mathcal{R}(u_h + u'), w_h + w') = (f, w_h + w'). \tag{2.20}$$

Due to the linear independency of w' and w_h, equation (2.20) is separated into the coarse scale and fine-scale equations, respectively:

$$(\frac{\partial(u_h + u')}{\partial t}, w_h) + (\mathcal{R}(u_h + u'), w_h) = (f, w_h), \tag{2.21}$$

$$(\frac{\partial(u_h + u')}{\partial t}, w') + (\mathcal{R}(u_h + u'), w') = (f, w'). \tag{2.22}$$

When the coarse-scale Equation (2.21) is rearranged, the following is obtained,

$$(\frac{\partial u_h}{\partial t}, w_h) + (\mathcal{R}(u_h), w_h) - (f, w_h) = -(\frac{\partial u'}{\partial t}, w_h) - (\mathcal{R}(u_h + u') - \mathcal{R}(u_h), w_h). \tag{2.23}$$

The LHS of Equation (2.23) contains terms present in the standard Galerkin procedure. However, it is also depends on the solution to the fine-scale equation which can be considered as the error in the coarse scale approximation. The goal of VMS sub-grid modelling is to approximate the fine-scale solution using Equation (2.22) and substitute it in Equation (2.21). Different closures can be obtained for the fine-scales depending on the type of approximation, especially when the problem is non-linear *Codina et al.* (2007); *Codina*

(2002); *Bazilevs et al.* (2007); *Wang and Oberai* (2010a); *Gravemeier et al.* (2010); *Masud and Calderer* (2011b). However, for linear problems Hughes et al. *Hughes et al.* (1998a) demonstrated that the fine scale solution u' is related to the Green's function $g'(x, y)$ of the adjoint operator $\mathcal{R}^*$ and the coarse scale residual as follows:

$$u'(y) = -\int_{\Omega} g'(x, y)(\mathcal{R}(u_h) - f)(x) d\Omega_x. \tag{2.24}$$

The simplest approximation to which is given by

$$u' = -\tau(\mathcal{R}(u_h) - f). \tag{2.25}$$

Although, Equation (2.25) and Equation (2.24) have been derived for linear problems, the idea that the coarse-scale residual can be linked to the fine-scale solution, remains the basis for developing non-linear VMS models *Codina et al.* (2007); *Codina* (2002); *Bazilevs et al.* (2007); *Wang and Oberai* (2010a); *Gravemeier et al.* (2010); *Masud and Calderer* (2011b) as well.

2.3 Data-driven techniques.

2.3.1 Neural networks.

Neural networks (NNs) are computational algorithms that mimic the brain's functions by learning existing patterns in data. They have been used extensively for regression, clustering, and classification tasks. A few types of commonly used NNs are: (a) fully connected neural network (FNN), (b) convolutional neural network (CNN), (c) recurrent neural networks (RNN), and (d) generative adversarial network (GAN).

In this dissertation, we will be using neural networks primarily for super-resolution. The CNN architecture has been extensively used for performing super-resolution in the literature. For the same number of trainable parameters, CNNs achieve better performance

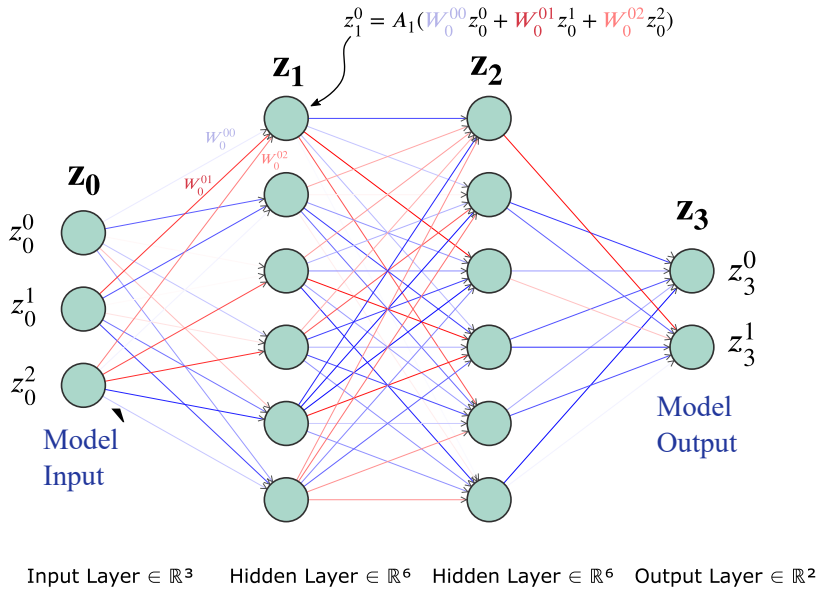


Figure 2.1: A Sketch of a sample FNN with 3 inputs, 2 output, and 2 hidden layers with 6 neurons each. The arrows are colored using the value of weights corresponsing to each connection.

than FNNs by exploiting the spatial correlations in the data. However, CNNs are only ideal when the model input or output data is present on a structured grid. As we will see in Chapter 4, the coarse and fine solutions in the present work are present as basis coefficients and not as an image. Hence, in this work, we will restrict ourselves to using variations of the FNNs only.

As shown in figure 2.1, an FNN is a mapping between the input layer $\mathbf{z}_0 \in R^{n_0}$ and the output layer $\mathbf{z}_N \in \mathbb{R}^{n_N}$. There are multiple layers between $\mathbf{z}_0$ and $\mathbf{z}_N$, which are known as hidden layers. Ideally, if the number of hidden layers, i.e., $N-1$ exceeds one, the FNN network is called a deep neural network (DNN). However, the true power of a DNN can only be harnessed when there are sufficiently many hidden layers. Each layer in an FNN is connected to the previous layer through the following relation:

$$\mathbf{z}_i = A_i(\mathbf{W}_{i-1}\mathbf{z}_{i-1} + \mathbf{b}_i) \quad for \quad i = 1, 2, 3, ..., N \tag{2.26}$$

where $\mathbf{W}_{i-1}$ is a matrix of weights of size $\mathbb{R}^{n_{i-1} \times n_i}$, $\mathbf{b}_i$ is a column vector of biases of size

$\mathbb{R}^{n_{i-1}\times 1}$ and A_i is the activation function. The activation function operates individually on the elements of $\mathbf{W}_{i-1}\mathbf{z}_{i-1} + \mathbf{b}_{i-1}$. There exist various choices for the activation function, such as the sigmoids, rectified linear units (ReLUs), leaky ReLU, exponential linear units (ELUs), and Swish. In Chapter 4, we will use the ELU) activation function with $\alpha = 1.0$ for all the layers except the last layer, where no activation function is used. This ensures that the network's output can assume any values and is not just capped to positive values. As a part of the training process, the values of the weights and biases present in equation (2.26) are optimized to ensure that the output of the network minimizes this loss function. This is done using the Adam optimizer *Kingma and Ba* (2014) which is part of the Keras *Chollet et al.* (2015) library. For the loss function, the mean squared error (MSE) $\mathcal{L}_{mse}$ is chosen, which is defined as:

$$\mathcal{L}_{mse} = \frac{1}{N_{dat}} \sum_{i=1}^{N_{dat}} \frac{||\mathbf{z}_{N,i,true} - \mathbf{z}_{N,i,pred}||_2^2}{n_N} \tag{2.27}$$

where $\mathbf{z}_{N,i,true}$ is the truth, $\mathbf{z}_{N,i,pred}$ is the prediction by the neural network, n_{dat} is the size of the output vector and N_{dat} is the number of data points over which the NN is trained. The Adam optimizer requires the gradient of the loss function with respect to the weights and biases, which is obtained through *backpropagation*.

CHAPTER III

Closure Development for Finite Element Discretizations Using the MZ-VMS Framework

3.1 Background and Motivation

Numerical simulation of multi-scale phenomena requires the development of coarse-grained models, which attempts to resolve a sub-set of the scales while providing a model for unresolved features. As discussed previously in Chapter 1, a popular approach employed in the simulation of turbulent flows is large-eddy simulation *Smagorinsky* (1963); *Vreman* (2004); *Nicoud and Ducros* (1999); *Germano et al.* (1991); *Meneveau et al.* (1996); *You and Moin* (2007a), which filters the flow field to resolve the largest energy containing scales and providing a model for the scales smaller than the filter length referred to as the sub-grid scales (SGS). The success of these sub-scale models largely depends on the validity of assumptions at simulated flow conditions. For example, the Smagorinsky model *Smagorinsky* (1963), which is one of the most commonly employed SGS models, is based on the assumption that modeled rate for turbulence kinetic energy transfer from large to small scale balances dissipation *Pope and Pope* (2000). This assumption is clearly not valid for all turbulent flows. An alternate approach to SGS modeling without employing phenomenological assumption, which we pursue, is to derive sub-grid models directly from the structure of the PDE and the numerical discretization.

In addition to physics-based sub-grid scale models *Smagorinsky* (1963); *Nicoud and Ducros* (1999); *Vreman* (2004); *Nicoud et al.* (2011) presented above, models based on the variational multiscale method have proven to be quite successful for both linear *Hughes et al.* (1998a, 1989); *Brooks and Hughes* (1982); *Codina* (2000); *Hughes et al.* (1986) and non-linear problems *Codina et al.* (2007); *Codina* (2002); *Bazilevs et al.* (2007); *Wang and Oberai* (2010a); *Gravemeier et al.* (2010); *Masud and Calderer* (2011b).

As we discussed in section 2.2.2, the variational multiscale method is based on a similar idea of decomposing the flow field into resolved (coarse-scale) and un-resolved (fine-scale) variables. The fine-scales are then approximated using simple algebraic operators acting on the residual of the coarse scales which give rise to additional stabilization terms in the standard Galerkin procedure *Hughes et al.* (1989); *Brooks and Hughes* (1982); *Codina* (2000). A link between the stabilization terms and implicit sub-grid models *Hughes et al.* (1998a) has been established. Different stabilization techniques such as the Galerkin least square (GLS) *Hughes et al.* (1989), the streamwise upwind Petrov Galerkin (SUPG) *Brooks and Hughes* (1982); *Hughes et al.* (1986), the adjoint-stabilized methods *Codina* (2000); *Franca et al.* (1992); *Codina* (2002), and the orthogonal sub-scale stabilization *Codina* (2002) can be derived based on the type of algebraic model for the fine-scales.

In the early days of the development of the VMS techniques, these methods were formulated for linear problems, and their application to the non-linear multiscale problems such as the Navier-Stokes equations though successful, depended on constructs such as transformations to linear problems such as the Oseen equations at every non-linear iteration. Recently, however, several attempts have been extended to develop non-linear VMS closure models by Codina et al. *Codina et al.* (2007); *Codina* (2002), Bazilevs et al. *Bazilevs et al.* (2007) and many others *Wang and Oberai* (2010a); *Gravemeier et al.* (2010); *Masud and Calderer* (2011b). These methods utilize a stabilization parameter τ which is an approximation to the inverse of the differential operator of the governing equation. This model parameter τ is typically defined in terms of a local length-scale, elemental Reynolds

and Courant numbers *Franca et al.* (1992) or derived from the Fourier analysis of the fine scale equation *Codina* (2002). Further, it is chosen to aid optimal convergence and stability of the method.

In this work, we aim to develop a general coarse-graining approach in the context of the continuous Galerkin method, that is: (i) built using a *non-linear* model reduction strategy akin to Greens functions for linear problems; (ii) capable of generating a fine-scale description directly from the structure of the PDE and the underlying numerics; and (iii) model parameters are adaptive to the resolution, and are dynamically determined.

The VMS decomposition of a PDE leads to a set of coupled equations which govern the coarse-scales (resolved) and the fine-scales (un-resolved) respectively. However, the fine-scale closure problem still persists. In our approach, the dependence of the fine-scale variables on the coarse-scale variable is removed by using the optimal prediction framework developed by Chorin *Chorin et al.* (2002); *Chorin and Hald* (2009). This framework, originally developed in the context of non-equilibrium statistical mechanics, enables the higher dimensional non-linear Markovian dynamical system to be written into an exactly equivalent lower dimensional non-Markovian dynamical system *Parish and Duraisamy* (2017a,b); *Gouasmi et al.* (2017). The advantage is that the evolution of any observable in time can be represented solely in terms of the resolved variables. The cost of evaluating the resulting closure term, however, is enormous. The possible simplifications will be discussed later in this chapter. Similar ideas have been put forward by Stinis *Stinis* (2007), Parish and Duraisamy *Parish and Duraisamy* (2017a,b) in context of spectral methods and discontinuous Galerkin (DG) *Parish and Duraisamy* (2017c) methods.

The MZ-VMS philosophy presented herein closely follows the approach presented in *Parish and Duraisamy* (2017c); *Parish* (2018). However, the derivation of the closure terms in a continuous Galerkin setting requires specific approximations. As will be discussed in this chapter, the contributing term to the final closure model in CG is different from that in DG. The main contribution of this work is to extend these dynamic closure models to

the continuous Galerkin (CG) method previously not explored by Parish and Duraisamy *Parish and Duraisamy* (2017c); *Parish* (2018) and testing them on canonical turbulent flow problems.

The outline of this chapter is as follows: In section 3.2, we develop the VMS-MZ method in the context of a continuous Galerkin (CG) discretization. In section 3.3, we derive a dynamic model for the estimation of the memory length of the convolution integral to provide a parameter free closure to the model. In the section 3.4, we discuss results for the burgers equation. We discuss results for canonical turbulence cases in section 3.5. Finally, we conclude our work in section 3.6.

3.2 The CG-MZ-VMS Framework

We now present the derivation of the CG-MZ-VMS framework. We begin with the governing equation in the domain $\Omega \subset \mathbb{R}^d$ with the boundary $\Gamma = \partial\Omega$, where $d \geq 1$ is the dimension of the problem as follows,

$$\frac{\partial u}{\partial t} + \mathcal{R}(u) - f = 0, \tag{3.1}$$

where $u = g$ at the boundary Γ and time $t \in (0, T]$. The weak form of the above PDE, obtained after integration by parts, can be written as follows,

$$\left(\frac{\partial u}{\partial t}, w\right)_\Omega + (R(u), w)_\Omega + (b(u), w)_\Gamma = (f, w)_\Omega \quad \forall w \in \mathcal{V}, \tag{3.2}$$

where $u \in \mathcal{V}$. The Sobolev space of functions $\mathcal{V} \equiv \mathcal{H}^1(\Omega)$ and first derivatives are square integrable. The functional space $\mathcal{V}$ is an infinite dimensional and must be approximated by a finite dimensional approximation $\tilde{\mathcal{V}}$. We consider the tessellation of Ω into non-overlapping finite elements. The domain and boundary of an element marked by Ω_e and Γ_e respectively. Also consider the following notations:

$$\Omega' = \bigcup_{i=1}^{n_{el}} \Omega^e \quad \text{and} \quad \Gamma' = \bigcup_{i=1}^{n_{el}} \Gamma^e. \tag{3.3}$$

where Ω', Γ' denote the interior and boundaries of all the elements respectively. Let $\tilde{\mathcal{V}} \subset C^0 \cap \mathcal{H}^1(\Omega)$ denote our finite dimensional FE approximation space containing basis functions having C^0 continuity everywhere including element boundaries. Approximating w by $\tilde{w}$ and u by $\tilde{u}$ in Equation (3.2), leads to the standard Galerkin procedure given by

$$\left(\frac{\partial \tilde{u}}{\partial t}, \tilde{w}\right)_{\Omega'} + (R(\tilde{u}), \tilde{w})_{\Omega'} + (b(\tilde{u}), \tilde{w})_{\Gamma'} = (f, \tilde{w})_{\Omega'} \quad \forall \tilde{w} \in \tilde{\mathcal{V}}, \tag{3.4}$$

where $\tilde{u} \in \tilde{\mathcal{V}}$. The above method, although directly applicable to diffusion dominated problems, encounters stability issues when applied to convection dominated problems. The VMS procedure provides a solution to this problem by elegantly accounting for the sub-grid scale effects. By splitting the space of the solution $u = \tilde{u} + u'$ and the weighting function $w = \tilde{w} + w'$, and substituting it into Equation (3.4), we obtain the following integral equations for the coarse and fine scales, respectively,

$$\left(\frac{\partial \tilde{u}}{\partial t}, \tilde{\mathbf{w}}\right)_{\Omega'} + (R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} + (R(u) - R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} + (b(\tilde{u}), \tilde{\mathbf{w}})_{\Gamma'} + (b(u) - b(\tilde{u}), \tilde{\mathbf{w}})_{\Gamma'} = (f, \tilde{\mathbf{w}})_{\Omega'}, \tag{3.5}$$

$$\left(\frac{\partial u'}{\partial t}, \mathbf{w}'\right)_{\Omega'} + (R(\tilde{u}), \mathbf{w}')_{\Omega'} + (R(u) - R(\tilde{u}), \mathbf{w}'))_{\Omega'} + (b(\tilde{u}), \mathbf{w}')_{\Gamma'} + (b(u) - b(\tilde{u}), \mathbf{w}')_{\Gamma'} = 0, \tag{3.6}$$

where u' and w' lie in a space orthogonal to $\tilde{u}$ and $\tilde{w}$ i.e. $u', w' \in \mathcal{V}'$ and $f \in \tilde{\mathcal{V}}$. This idea of decomposing the full space into orthogonal spaces has previously been pursued, for instance the Orthogonal Sub-Scale (OSS) method by Codina *Codina* (2002) and Parish and Duraisamy *Parish* (2018); *Parish and Duraisamy* (2017b). By substituting the resolved and un-resolved variables in terms of their modal coefficients and their basis function as

follows,

$$\tilde{u} = \tilde{\mathbf{w}}^{\mathbf{T}}\tilde{\mathbf{a}} \quad \text{and} \quad u' = \mathbf{w'}^{\mathbf{T}}\mathbf{a'}, \tag{3.7}$$

we get the following ODE systems for modal coefficients of the coarse and fine scales,

$$\frac{d\tilde{\mathbf{a}}}{dt} = \tilde{\mathbf{M}}^{-\mathbf{1}}(-(R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} - (R(u) - R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} - (b(\tilde{u}), \tilde{\mathbf{w}})_{\Gamma'} - (b(u) - b(\tilde{u}), \tilde{\mathbf{w}})_{\Gamma'} + (f, \tilde{\mathbf{w}})_{\Omega'}), \tag{3.8}$$

$$\frac{d\mathbf{a'}}{dt} = \mathbf{M'}^{-\mathbf{1}}(-(R(\tilde{u}), \mathbf{w'})_{\Omega'} - (R(u) - R(\tilde{u}), \mathbf{w'}))_{\Omega'} - (b(\tilde{u}), \mathbf{w'})_{\Gamma'} - (b(u) - b(\tilde{u}), \mathbf{w'})_{\Gamma'}), \tag{3.9}$$

where the mass matrices for resolved scales and un-resolved orthogonal scales can be written as

$$\tilde{\mathbf{M}} = (\tilde{\mathbf{w}}, \tilde{\mathbf{w}}^{\mathbf{T}}) \quad \text{and} \quad \mathbf{M'} = (\mathbf{w'}, \mathbf{w'}^{\mathbf{T}}). \tag{3.10}$$

By utilizing the Mori-Zwanzig procedure to integrate out variables in Equation (3.9) from (3.8), we get the following system:

$$\left(\frac{\partial \tilde{u}}{\partial t}, \tilde{\mathbf{w}}\right)_{\Omega'} + (R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} = \tilde{\mathbf{M}} \int_0^t K(\tilde{\mathbf{a}}(t-s), s)ds, \tag{3.11}$$

where the additional term to the RHS is due to the memory effects. In the present formulation, we will use the finite memory model i.e. $\int_0^t K(\tilde{\mathbf{a}}(t-s), s)ds \approx \tau K(\tilde{\mathbf{a}}(t), 0)$ which results in the following simplification:

$$\left(\frac{\partial \tilde{u}}{\partial t}, \tilde{\mathbf{w}}\right)_{\Omega'} + (R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} = \tau \tilde{\mathbf{M}} e^{\mathcal{L}t} \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_{\mathbf{0}}, \tag{3.12}$$

where τ is the memory length. The memory kernel $\tilde{\mathbf{M}}e^{\mathcal{L}t}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_{\mathbf{0}}$ can be further simplified as follows

$$\begin{aligned}\tilde{\mathbf{M}}e^{\mathcal{L}t}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_{\mathbf{0}} = &\int_{\Omega'}\int_{\Omega'} \tilde{\mathbf{w}}R'(\Pi'(x,y)(R(\tilde{u})-f))d\Omega'_y d\Omega'_x \\ &+\int_{\Omega'}\int_{\Gamma'} \tilde{\mathbf{w}}R'(\Pi'(x,y)(b(\tilde{u})))d\Gamma'_y d\Omega'_x \\ &+\int_{\Gamma'}\int_{\Omega'} \tilde{\mathbf{w}}b'(\Pi'(x,y)(R(\tilde{u})-f))d\Omega'_y d\Gamma'_x \\ &+\int_{\Gamma'}\int_{\Gamma'} \tilde{\mathbf{w}}b'(\Pi'(x,y)(b(\tilde{u})))d\Gamma'_y d\Gamma'_x,\end{aligned} \tag{3.13}$$

where Π' is the orthogonal projector onto the space of the the fine scales i.e,

$$\Pi'(x,y) = \mathbf{w'}^T(x)\mathbf{M'^{-1}}\mathbf{w'}(y). \tag{3.14}$$

Detailed derivation of Equation (3.13) can be found in Appendix A. An important aspect of the sub-scales u' is that it is dynamic *Codina* (2002); *Codina et al.* (2007) in nature. When these sub-scales are approximated with the inverse of the spatial operator, the resulting VMS models are non-Markovian *Codina* (2002); *Codina et al.* (2007) and the sub-scales u' have to be tracked in time. The M-Z formalism on the other hand, precisely integrates out the time dependency of these sub-scales, making the final formulation Markovian on the resolved variables only. To make this formulation computationally tractable, we assume that the memory integral is correlated to its integrand at $s = 0$ and has finite support in time. This is the main reason an approximation to the inverse of the differential operator - as is popularly used to derive VMS models - is not required here. This marks the end of the derivation of the MZ-VMS framework that was originally developed by *Parish* (2018); *Parish and Duraisamy* (2017c). This framework has been successfully applied to the DG method *Parish* (2018); *Parish and Duraisamy* (2017c). In the rest of this chapter, we will extend it to the CG method and validate its performance.

As shown by Parish and Duraisamy *Parish and Duraisamy* (2017c); *Parish* (2018), the main contributing term in Equation (3.13) to the final closure model in DG is Term 4. However, in this formulation, we assume that the fine-scales vanish at the element boundaries analogous to the concept of bubble functions *Masud and Calderer* (2011a); *Franca and Farhat* (1995); *Brezzi et al.* (1992); *Hughes* (1995). This approximation has also been used in the OSS model *Codina* (2002); *Codina et al.* (2007). By using this approximation, Term 2 and Term 4 in Equation (3.13) are neglected, resulting in the following equation:

$$\begin{aligned}\tilde{\mathbf{M}} e^{\mathcal{L}t} \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}_0} = &\int_{\Omega'}\int_{\Omega'} \tilde{\mathbf{w}} R'(\Pi'(x,y)(R(\tilde{u}) - f)) d\Omega'_y d\Omega'_x \\ &+ \int_{\Gamma'}\int_{\Omega'} \tilde{\mathbf{w}} b'(\Pi'(x,y)(R(\tilde{u}) - f)) d\Omega'_y d\Gamma'_x.\end{aligned} \tag{3.15}$$

The scale separation by projection of the residual on the fine scale can be computed as follows,

$$\int_{\Omega'_y} \Pi'(x,y)(R(\tilde{u}(y)) - f) d\Omega'_y = (R(\tilde{u}(x)) - f) - \tilde{\Pi}(R(\tilde{u}(x)) - f), \tag{3.16}$$

where $\tilde{\Pi}$ is again the L_2 projector on the finite dimensional space spanned by $\tilde{\mathbf{w}}$. This concludes the derivation of CG-MZ-VMS framework for the fixed memory type model.

3.3 Dynamic Memory Estimation

While the constant memory length model provides a closure to the memory term in the M-Z expression, the parameter τ should adapt to the evolving resolution and not necessarily remain constant. Another approach is to allow the parameter τ to dynamically vary in time to attempt to represent the variations of the effects of the fine-scale quantities on the coarse scales. To facilitate the stabilization of our method with fewer parameters and

account for the temporal variations of the memory length, we seek a dynamic memory length model using the variational counterpart of the Germano's identity *Germano et al.* (1991); *Germano* (1992); *Oberai and Wanderer* (2005); *Akkerman et al.* (2010). A similar dynamic procedure has been previously used by Oberai et al. *Oberai and Wanderer* (2005) and Akkerman et al. *Akkerman et al.* (2010) to estimate model coefficients. We begin by applying a zero-variance phase space projector with a fully resolved initial condition with the large-scale equation (Eqn (3.5)) to obtain an exact solution to the closure problem as following:

$$\tilde{\mathbf{M}} \int_0^t K(\tilde{\mathbf{a}}(t-s), s) ds = (R(\tilde{u}) - R(u), \tilde{\mathbf{w}})_{\Omega'} + (b(\tilde{u}) - b(u), \tilde{\mathbf{w}})_{\Gamma'}. \tag{3.17}$$

By assuming that the memory term has a finite support we obtain

$$\tau_1 \tilde{\mathbf{M}} K(\tilde{\mathbf{a}}(t), 0) = (R(\tilde{u}) - R(u), \tilde{\mathbf{w}})_{\Omega'} + (b(\tilde{u}) - b(u), \tilde{\mathbf{w}})_{\Gamma'}. \tag{3.18}$$

Similarly, for a separate coarser mesh with weighting function $\hat{\mathbf{w}} \in \hat{\mathcal{V}}$, where $\hat{\cdot}$ signifies a coarser mesh than $\tilde{\cdot}$, the memory terms can be written as

$$\tau_2 \hat{\mathbf{M}} K(\hat{\mathbf{a}}(t), 0) = (R(\hat{u}) - R(u), \hat{\mathbf{w}})_{\Omega'} + (b(\hat{u}) - b(u), \hat{\mathbf{w}})_{\Gamma'}. \tag{3.19}$$

We choose $\tilde{\mathbf{w}}$ such that it spans the weighting function on the coarser mesh $\hat{\mathbf{w}}$ i.e. $\hat{\mathcal{V}} \subset \tilde{\mathcal{V}}$, which results in the following equation:

$$\tau_1 \mathbf{G} \tilde{\mathbf{M}} K(\tilde{\mathbf{a}}(t), 0) = (R(\tilde{u}) - R(u), \hat{\mathbf{w}})_{\Omega'} + (b(\tilde{u}) - b(u), \hat{\mathbf{w}})_{\Gamma'} \tag{3.20}$$

where $\mathbf{G}$ is a matrix which transforms $\tilde{\mathbf{w}}$ to $\hat{\mathbf{w}}$ given by

$$\mathbf{G}\tilde{\mathbf{w}} = \hat{\mathbf{w}} \tag{3.21}$$

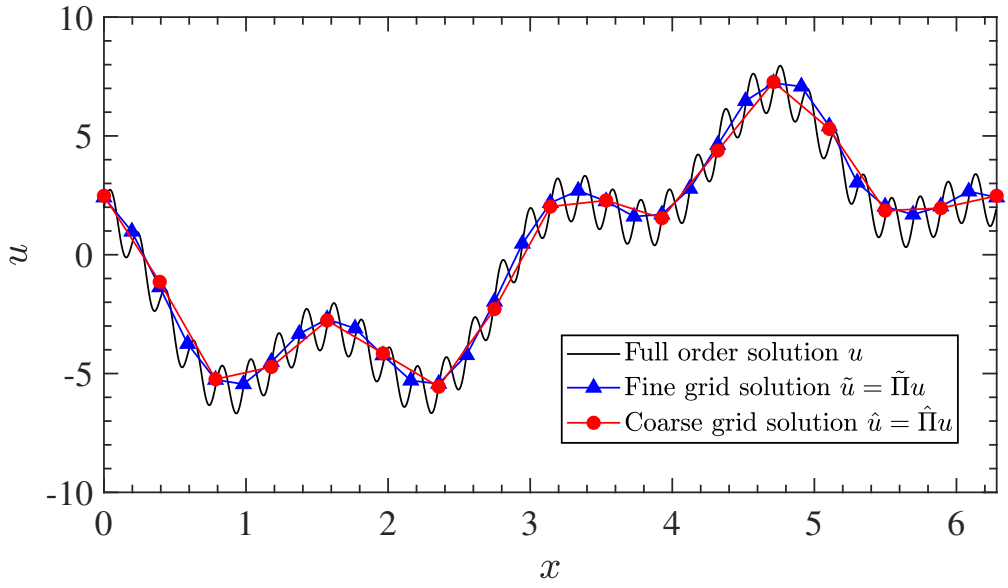


Figure 3.1: L_2 projection of an example full order solution on two meshes with element sizes h and 2h respectively i.e $\tilde{u}$ and $\hat{u}$.

If the finer grid is obtained by element wise refinement of the coarse grid, the fine grid $\tilde{\mathbf{w}}$ basis functions span all the weighting functions on the coarser mesh $\hat{\mathbf{w}}$. By subtracting Equations (3.19) and (3.20) we obtain

$$\tau_1 \mathbf{G}\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t),0) - \tau_2 \hat{\mathbf{M}}K(\hat{a}(t),0) = (R(\tilde{u}) - R(\hat{u}), \hat{\mathbf{w}})_{\Omega'} + (b(\tilde{u}) - b(\hat{u}), \hat{\mathbf{w}})_{\Gamma'}. \quad (3.22)$$

To obtain $\hat{u}$ project the $\tilde{u}$

$$\hat{u} = \hat{\Pi}\tilde{u}, \quad (3.23)$$

where $\hat{\Pi} : \tilde{L}_2 \rightarrow \hat{L}_2$ is the L_2 projector on the coarse grid. Figure 3.1 shows the projection of a fully-resolved simulation onto $\tilde{u}$ and $\hat{u}$. This is similar to test filtering in the dynamic Smagorinsky model *Germano et al.* (1991) employed in LES. Here, we assume a scaling law similar to the one proposed by Parish and Duraisamy *Parish and Duraisamy* (2017b) relating the memory lengths τ at two different levels of coarsening as following

$$\frac{\tau_1}{\tau_2} = \left[\frac{\Delta_1}{\Delta_2}\right]^{1.5}, \quad (3.24)$$

where Δ_1 and Δ_1 denote the element sizes at the fine and coarse mesh. An important observation is Equation (3.13) cannot be satisfied for all $\hat{\mathbf{w}}$ with a single value of τ, but is true only in the average sense. To satisfy this condition, three different possibilities are considered here:

1. Dynamic-τ-AVG: Scale the modes with their respective modal values

$$\tau_1 \hat{\mathbf{a}}^{\mathbf{T}} \mathbf{G} \tilde{\mathbf{M}} K(\tilde{\mathbf{a}}(t), 0) - \tau_2 \hat{\mathbf{a}}^{\mathbf{T}} \hat{\mathbf{M}} K(\hat{\mathbf{a}}(t), 0) = (R(\tilde{u}) - R(\hat{u}), \hat{u})_{\Omega'} + (b(\tilde{u}) - b(\hat{u}), \hat{u})_{\Gamma'}, \tag{3.25}$$

   which gives the following final form for the dynamic memory length,

$$\tau = \frac{(R(\tilde{u}) - R(\hat{u}), \hat{u})_{\Omega'} + (b(\tilde{u}) - b(\hat{u}), \hat{u})_{\Gamma'}}{\hat{\mathbf{a}}^{\mathbf{T}} \mathbf{G} \tilde{\mathbf{M}} K(\tilde{\mathbf{a}}(t), 0) - \left(\frac{\Delta_2}{\Delta_1}\right)^{1.5} \hat{\mathbf{a}}^{\mathbf{T}} \hat{\mathbf{M}} K(\hat{\mathbf{a}}(t), 0)}, \tag{3.26}$$

2. Dynamic-τ-LS: Solve the overdetermined system based on some optimality condition,

$$\mathbf{L} = \mathbf{R}\tau, \tag{3.27}$$

   where $\mathbf{L}$ and $\mathbf{R}$ are given by

$$\mathbf{L} = (R(\tilde{u}) - R(\hat{u}), \hat{\mathbf{w}})_{\Omega'} + (b(\tilde{u}) - b(\hat{u}), \hat{\mathbf{w}})_{\Gamma'}, \tag{3.28}$$

$$\mathbf{R} = \mathbf{G} \tilde{\mathbf{M}} K(\tilde{\mathbf{a}}(t), 0) - \left[\frac{\Delta_2}{\Delta_1}\right]^{1.5} \hat{\mathbf{M}} K(\hat{\mathbf{a}}(t), 0). \tag{3.29}$$

   The above system can be solved using the least-squares approach which is commonly used with the DSM *Germano et al.* (1991) LES model, resulting in the following expression:

$$\tau = \frac{\mathbf{L}^{\mathbf{T}} \mathbf{R}}{\mathbf{R}^{\mathbf{T}} \mathbf{R}}. \tag{3.30}$$

3. Dynamic-τ-l_2: Approximate τ based on the following equation:

$$\tau = \frac{||\mathbf{L}||}{||\mathbf{R}||}, \tag{3.31}$$

where $||\cdot||$ denotes any kind of norm. In the present work, we have used l_2 or Euclidean norm for all our calculations. By using this averaging procedure, we obtain a value of τ that is (i) always positive; and (ii) free from division errors.

Although the steps involved in derivation of the above formulation closely follow that of the DSM *Germano et al.* (1991); *Meneveau et al.* (1996), our approach is valid for general PDEs in that the functional form of the model is not chosen based on the underlying physical phenomena. Unlike other traditional LES SGS models such as DSM *Germano et al.* (1991); *You and Moin* (2007a); *Meneveau et al.* (1996), WALE *Nicoud and Ducros* (1999), VREMEN *Vreman* (2004), and Sigma *Nicoud et al.* (2011) which are derived exclusively for the Navier-Stokes equation or other scalar transport equations, our model is not equation specific.

3.4 One-dimensional viscous Burgers equation

As a first step towards deriving coarse-grained models for the Navier-Stokes equation, we apply our framework to a 1-D non-linear PDE exhibiting multiscale features. To this end, let $\mathcal{V} \equiv \mathcal{H}^1(\Omega)$ denote the Sobolev space where our solution u and weighting functions w exist. The viscous Burgers equation in the domain $\Omega \subset \mathbb{R}$ is given by the following equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \tag{3.32}$$

with periodic boundary conditions and the time varying from $t \in (0, T]$. The weak form of Equation (3.32) translates into a problem of finding $u \in \mathcal{V}$ such that

$$\left(\frac{\partial u}{\partial t}, w\right)_\Omega + \left(u\frac{\partial u}{\partial x}, w\right)_\Omega + \nu\left(\frac{\partial u}{\partial x}, \frac{\partial w}{\partial x}\right)_\Omega = 0 \quad \forall w \in \mathcal{V}. \tag{3.33}$$

Using integration by parts we obtain *Tezduyar* (2001),

$$\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2}, w\right)_\Omega + (J(u), w)_\Gamma = 0, \tag{3.34}$$

where, $(a, b)_{\Gamma'} = \sum_k \int_{\Gamma'_k} ab\, d\Gamma'$ and $J(u) = \nu n_1.\nabla u_1 + \nu n_2.\nabla u_2$ (where subscripts 1 and 2 denote adjacent elements sharing a boundary). By utilizing the present coarse graining procedure to Equation 3.33 we get

$$\left(\frac{\partial \tilde{u}}{\partial t} + \tilde{u}\frac{\partial \tilde{u}}{\partial x} - \nu\frac{\partial^2 \tilde{u}}{\partial x^2}, \tilde{w}\right)_{\Omega'} + (J(\tilde{u}), \tilde{w})_{\Gamma'} = \tau\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t), 0), \tag{3.35}$$

where the memory term $\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t), 0)$ is given by,

$$\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t), 0) = \int_{\Omega'}\int_{\Omega'} \tilde{\mathbf{w}}R'(\Pi'(x, y)(R(\tilde{u})))d\Omega'_y d\Omega'_x + \int_{\Gamma'}\int_{\Omega'} \tilde{\mathbf{w}}J(\Pi'(x, y)(R(\tilde{u})))d\Omega'_y d\Gamma'_x \tag{3.36}$$

and R' denotes the linearization of the non-linear operator about $\tilde{u}$. Using integration by parts and neglecting the sub-scale contributions at the elemental boundaries we have

$$\tilde{\mathbf{M}}K(\tilde{\mathbf{a}}(t), 0) = \int_{\Omega'} R^*(\tilde{\mathbf{w}}(x))[\int_{\Omega'} \Pi'(x, y)R(\tilde{u})d\Omega'_y]d\Omega'_x, \tag{3.37}$$

where R^* is the adjoint of the linearized operator R'. The integrand is computed as follows:

$$\int_{\Omega'_y} \Pi'(x, y)R(\tilde{u}(y))d\Omega'_y = R(\tilde{u}(x)) - \tilde{\Pi}(R(\tilde{u}(x))). \tag{3.38}$$

The resulting closure is very similar to the adjoint stabilization method *Codina* (2000); *Hughes et al.* (1998a); *Franca et al.* (1992) except $R(\tilde{u}(x)) - \tilde{\Pi}(R(\tilde{u}(x)))$ is present instead of $R(\tilde{u}(x))$. Here, R^* denotes the adjoint of the linearized operator R'. The adjoint operator R^* is given by

$$R^*(\tilde{w}) = -\tilde{u}\frac{\partial \tilde{w}}{\partial x} - \nu\frac{\partial^2 \tilde{w}}{\partial x^2} \tag{3.39}$$

Substitution of Equation (3.37) into Equation (3.35) results in the following problem for the coarse scales $\tilde{u} \in \tilde{\mathcal{V}}$:

$$\left(\frac{\partial \tilde{u}}{\partial t}+\tilde{u}\frac{\partial \tilde{u}}{\partial x}, \tilde{w}\right)_{\Omega'} +\nu\left(\frac{\partial \tilde{u}}{\partial x}, \frac{\partial \tilde{w}}{\partial x}\right)_{\Omega'} = \tau \sum_K \int_K R^*(\tilde{w})[R(\tilde{u}(x))-\tilde{\Pi}(R(\tilde{u}(x)))]d\Omega' \quad \forall \tilde{w} \in \tilde{\mathcal{V}}. \tag{3.40}$$

Similarly the following coarse grained model can be derived for the linear advection-diffusion equation:

$$\begin{aligned} &\left(\frac{\partial \tilde{u}}{\partial t} + a\frac{\partial \tilde{u}}{\partial x}, \tilde{w}\right)_{\Omega'} + \nu\left(\frac{\partial \tilde{u}}{\partial x}, \frac{\partial \tilde{w}}{\partial x}\right)_{\Omega'} \\ &= \tau \sum_K \int_K \left(- a\frac{\partial \tilde{w}}{\partial x} - \nu\frac{\partial^2 \tilde{w}}{\partial x^2}\right)[R(\tilde{u}(x)) - \tilde{\Pi}(R(\tilde{u}(x)))]d\Omega' \quad \forall \tilde{w} \in \tilde{\mathcal{V}}. \end{aligned} \tag{3.41}$$

Equations (3.40) and (3.41) are first discretized in time using the θ family of methods *Donea and Huerta* (2003). Equation (3.40) is then linearized using the standard Picard algorithm.

3.4.1 Steepening of sine wave

To benchmark our coarse-grained model, the solution to the viscous Burgers equation is computed at $T = 3.0$ for an initial sine profile *Wang and Oberai* (2010a); *Parish* (2018) on a periodic domain of length 2π. To discretize in time, we use the θ family of methods *Donea*

and Huerta (2003) with $\theta = 0.5$ (Crank-Nicolson). The simulation parameters for DNS and coarse grained simulations are summarized in Table 3.1. When sufficient resolution is available (i.e. DNS limit), the viscosity is responsible for dissipating energy at the shock. However, when the resolution is insufficient, sub-grid models are responsible for dissipating the energy. To ensure that the viscous dissipation due to large scales is negligible, viscosity ν has been set to a small value of 10^{-4}. As a consequence, the primary contribution to the total dissipation comes from the sub-grid model.

It can be observed in Figure 3.2, the coarse-grained model solution approaches the projected DNS. Figures 3.3a and 3.3b show the time evolution of resolved KE and its rate of dissipation. These results indicate that the performance of all the models are comparable except the case when no sub-grid model was used or a fixed $\tau = 0.01$ was used. A comparison between the solutions on the space-time diagram obtained using our dynamic-τ model, OSS, no-model and projected DNS has been presented in Figure 3.5. Among our Finite Memory (FM) models, the case with $\tau = 0.01$ performs the worst, as can be seen in Figures 3.2, 3.3a and 3.3b. The solution at $T = 3$ improves when τ is increased to 0.11 and becomes worse when further increased to $\tau = 0.23$, which suggests the existence of an optimum τ value. The uncertainty in choosing the value of τ close to its optimum value can be reduced with a dynamic model. To this end, methods described in Section 3.3 are used to compute τ dynamically in Figure 3.4. Results indicate that the Dynamic-τ-AVG, the Dynamic-τ-LS and the Dynamic-τ-l_2 models predict a similar magnitude of τ for the period of time considered. However, the Dynamic-τ-l_2 model, which ensures positivity of the τ, was found to be most stable and was used for all the following calculations in this chapter. Although it is possible to use Method 1 and Method 2 by clipping τ above zero, they were found to be unstable for the TGV problem which will be discussed later in Section 3.5.2.

At $T = 3$, the Dynamic-τ-l_2 predicts $\tau \approx 0.05$, which supports our argument that an optimum τ exists in the range of 0.01 and 0.23. The t-model which assumes $\tau = t$ predicts

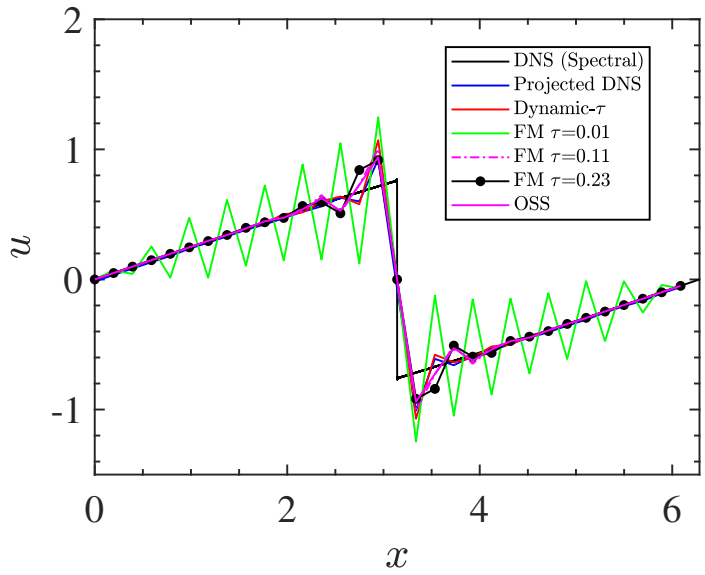


Figure 3.2: Solution to the Burgers Equation at $T = 3.0$ computed using different methods compared to projected DNS for the sine wave problem.

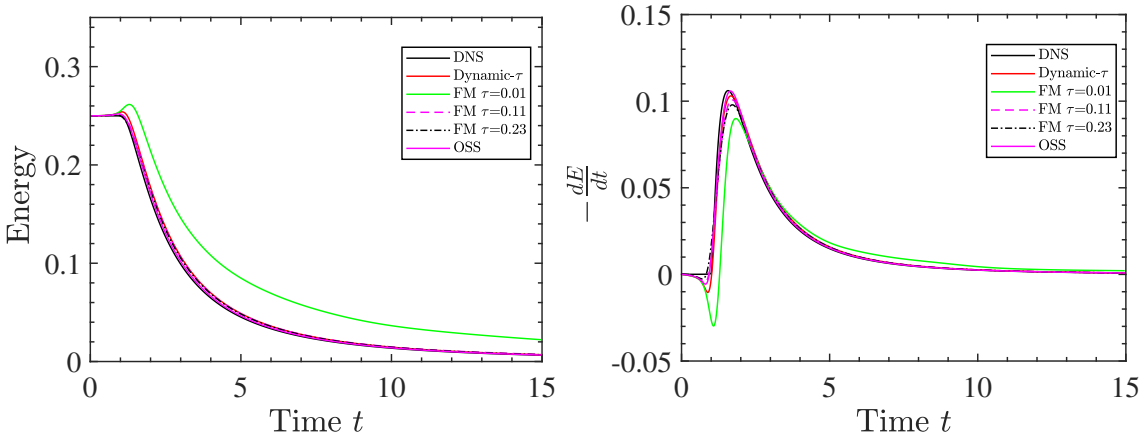
a τ which does not perform well in this case and becomes unstable. However, as noted by Stinis *Stinis* (2013), the t-model needs to be re-normalized with a coefficient for the correct prediction of the memory length i.e. $\tau = C_N t$. When renormalization is used, $\tau = 0.014t$ is the correct representative of the memory length with $C_N = 0.014$ as shown in Figure 3.4.

Case	Domain Size L	Degrees of Freedom N	Grid Size dx	Time Step dt	Viscosity ν	Memory Length τ
DNS (Spectral)	2π	8192 modes	7.67×10^{-4}	1.92×10^{-4}	10^{-4}	-
Dynamic τ	2π	32 elements	1.96×10^{-1}	1.96×10^{-2}	10^{-4}	Dynamic
FM τ=0.01	2π	32 elements	1.96×10^{-1}	1.96×10^{-2}	10^{-4}	0.01
FM τ=0.11	2π	32 elements	1.96×10^{-1}	1.96×10^{-2}	10^{-4}	0.11
FM τ=0.23	2π	32 elements	1.96×10^{-1}	1.96×10^{-2}	10^{-4}	0.23
OSS *Codina* (2002)	2π	32 elements	1.96×10^{-1}	1.96×10^{-2}	10^{-4}	-
No-Model	2π	32 elements	1.96×10^{-1}	1.96×10^{-2}	10^{-4}	0

Table 3.1: Simulation parameters for DNS and LES of the Burgers Equation for an initial sine profile.

3.4.2 Burgers turbulence

To further assess the performance of our coarse-grained model for turbulence, we use it to study the Burgers turbulence problem *Parish and Duraisamy* (2017a); *Gouasmi et al.* (2017). The solution to the Burgers equations exhibits some similarities to realistic turbulence, both having an inertial and a dissipation range *Wang and Oberai* (2010a). However, the solution



(a) Time evolution of kinetic energy.

(b) Rate of kinetic energy decay.

Figure 3.3: Time evolution of (a) kinetic energy and (b) rate of kinetic energy decay computed using different methods compared to DNS for the sine wave problem .

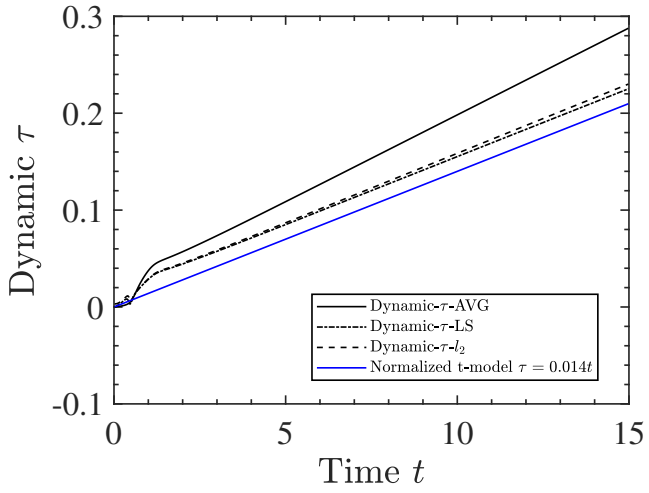
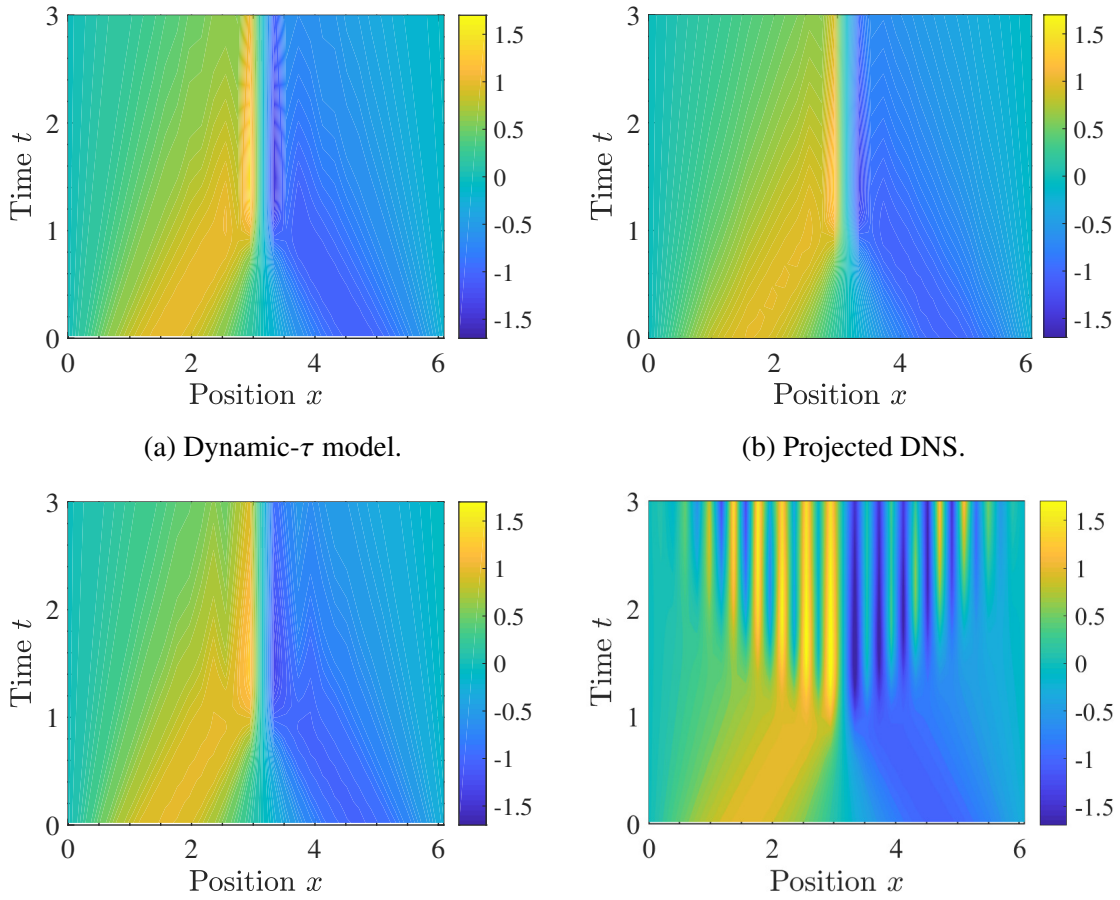


Figure 3.4: Evolution of memory length τ predicted using different dynamic models for the sine wave problem.



(a) Dynamic-τ model.

(b) Projected DNS.

(c) OSS model *Codina* (2002).

(d) No-Model.

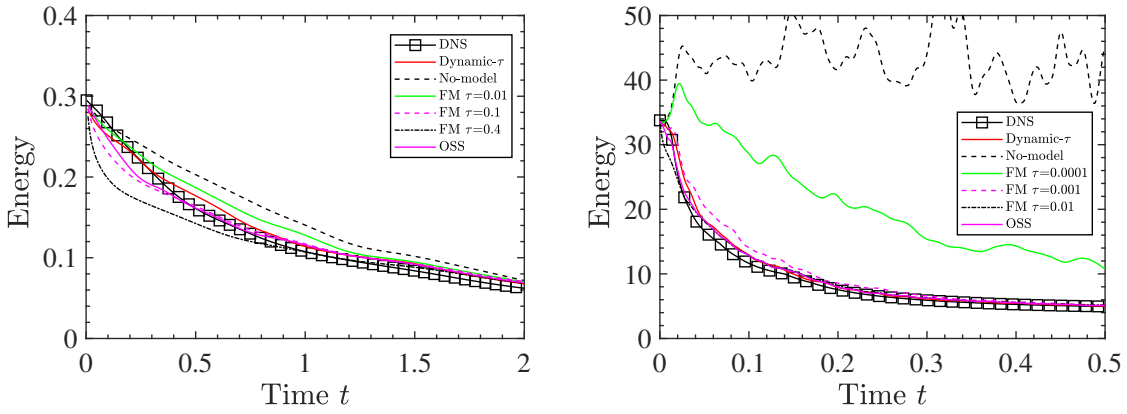
Figure 3.5: Comparison of the wave system obtained using the dynamic-τ model, projected DNS, OSS and No-model on the x-t diagram.

to Burgers turbulence is non-chaotic, unlike physically realistic turbulence obtained though the Navier-Stokes equations. To obtain the initial flow field satisfying a given energy spectrum, the following initial condition has been used

$$U(x,0) = \sum_{k=1}^{K_c} U^* \sqrt{2E(k)} sin(kx + \beta), \tag{3.42}$$

where, the phase β is randomly set from $[-\pi, \pi]$ and the energy spectra $E(k)$ is set to $5^{-5/3}$ for $k = 1$ to 5 and $k^{-5/3}$ thereafter. Two different test cases are considered here: (i.) a high viscosity case A with $U^* = 1$, $K_c = 8$ and $\nu = 0.01$, and (ii.) a low viscosity case B with $U^* = 10$, $K_c = 32$ and $\nu = 0.0005$. Simulation parameters are summarized in Table 3.2. The two cases are considered to demonstrate the effect of the sub-grid model on moderately and highly under-resolved simulations respectively.

In the first case, when no sub-grid model is employed, the time variation of resolved kinetic energy is close to the DNS solution. However, for the low viscosity case, a sub-grid model becomes necessary. For comparison, DNS using the Fourier-Galerkin method is performed using 1024 and 4096 modes for case A and case B, respectively. The de-aliasing of the non-linear terms for the Fourier-Galerkin method is conducted by zero-padding (3/2-rule). The LES is conducted using the present coarse grained model with just 32 and 64 linear elements, respectively. For each case, results from the FM model and dynamic-τ are compared to results obtained without using a sub-grid model, the OSS model and the DNS. For the FM model, different values of τ are considered in the range where our simulations are stable. Figures 3.6 and 3.7 show the time evolution of resolved KE and its rate of dissipation for all these cases. Both these figures indicate that both the dynamic-τ model and OSS model accurately predict the time-evolution of the resolved kinetic energy in comparison DNS. Figure 3.8 shows the variation of τ obtained from our dynamic model which for both case A and B, predict a large variation of τ in time, suggesting the importance of the adaptive selection of τ. Figure 3.9 also shows the energy spectra at the final time. It



(a) Case A (b) Case B

Figure 3.6: Evolution of resolved KE using different methods compared to DNS for the Burgers Turbulence problem.

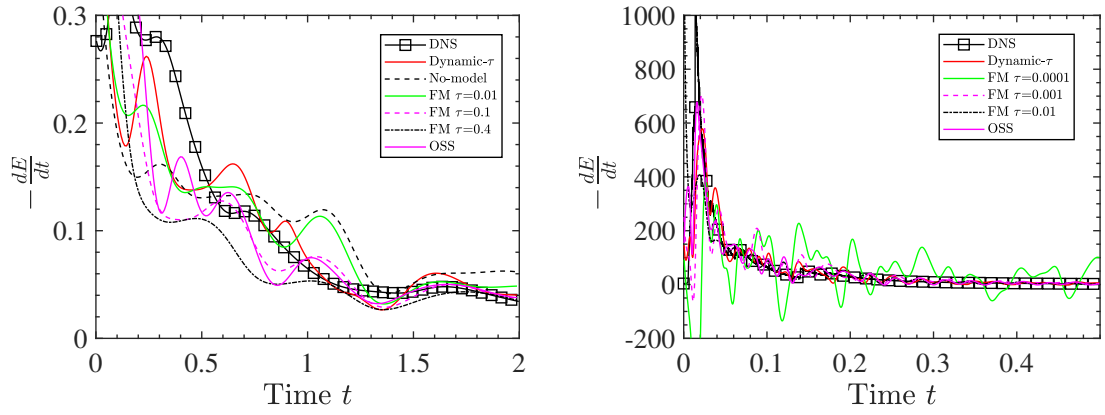
can be observed for Case A, that all the models perform similarly and the resolved modes are able to capture most of energy. For Case B, the present dynamic-τ model performs better than other models especially at lower wavenumbers (large scales).

Case	L	N	dx	dt	ν	U^*	K_c	FM τ's
Case A (DNS)	2π	1024 modes	3.06×10^{-3}	2.33×10^{-4}	10^{-2}	1	8	-
Case A (LES)	2π	32 elements	1.96×10^{-1}	8.5×10^{-3}	10^{-2}	1	8	0.01, 0.1 and 0.4
Case B (DNS)	2π	4096 modes	7.66×10^{-4}	3.41×10^{-6}	5×10^{-4}	10	32	-
Case B (LES)	2π	64 elements	9.81×10^{-2}	4.67×10^{-4}	5×10^{-4}	10	32	0.0001, 0.001 and 0.01

Table 3.2: Simulation parameters for DNS and LES of the Burgers Equation for the Burger turbulence problem.

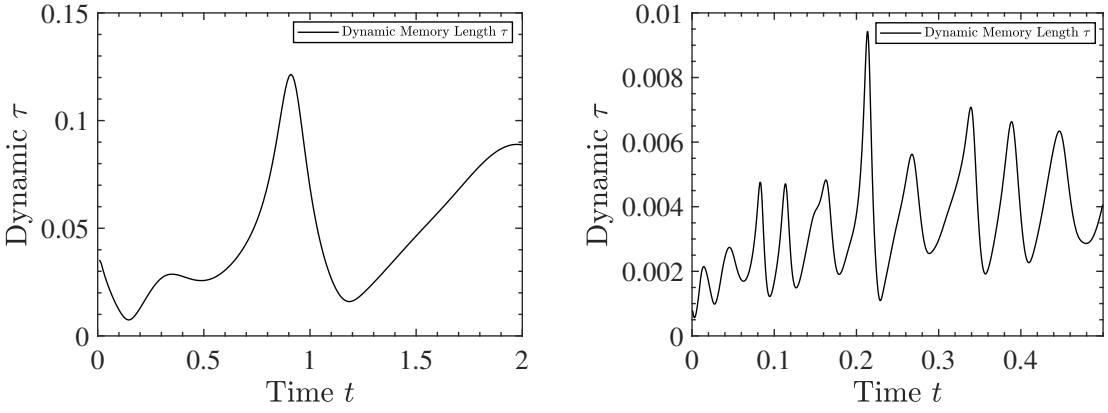
3.5 Application to the Navier-Stokes Equations

In this section, the coarse grained model is extended to the incompressible Navier-Stokes equations. Let $\mathcal{V}_d \equiv (\mathcal{H}^1(\Omega))^d$ and $\mathcal{K} \equiv L^2(\Omega)$ denote the Sobolev and Lebesque spaces where our solution and weighting functions exist and $d \geq 2$ in general. The weak form of the Navier-Stokes equations consists of finding $\mathbf{u} : [0, T] \rightarrow \mathcal{V}_d$, $p : [0, T] \rightarrow \mathcal{K}$ such that



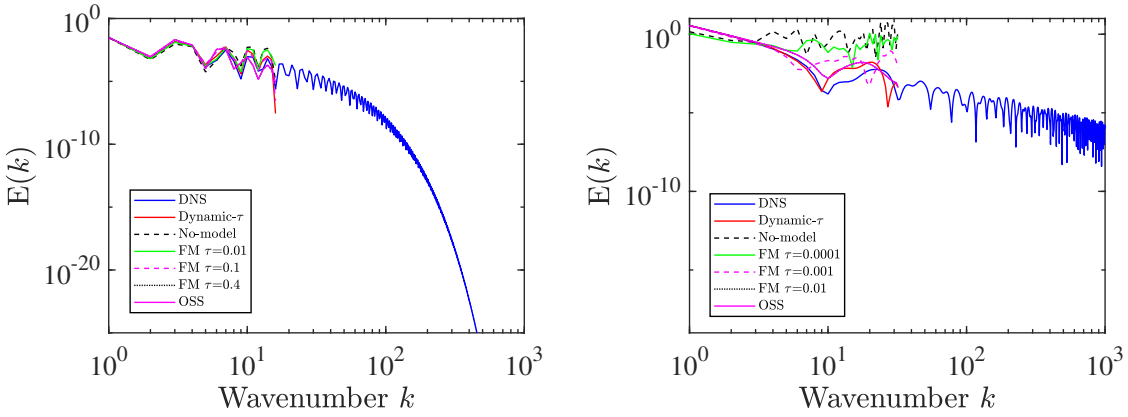
(a) Case A (b) Case B

Figure 3.7: Rate of energy decay due to dissipation by the sub-grid model and viscous dissipation by large scales using different methods compared to DNS for the Burgers Turbulence problem.



(a) Case A (b) Case B

Figure 3.8: Evolution of memory length τ predicted using our dynamic model for the Burgers Turbulence problem.



(a) Case A at time $T = 2.0$ (b) Case B at time $T = 0.5$

Figure 3.9: Energy spectra obtained using different methods compared to DNS for the Burgers Turbulence problem.

$$(\partial_t \mathbf{u}, \mathbf{w}) + \nu(\nabla \mathbf{u}, \nabla \mathbf{w}) + (\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{w}) - (p, \nabla \cdot \mathbf{w}) = (\mathbf{f}, \mathbf{w}) \quad \text{and} \quad (k, \nabla . \mathbf{u}) = 0, \tag{3.43}$$

for all $[\mathbf{w}, k] \in \mathcal{V}_d \times \mathcal{K}$.

We start by assessing how our coarse-grained model stabilizes the forced viscous Burgers equation in higher dimensions. The weak form for the Burgers equations in higher dimensions can be written as

$$\left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{w}\right)_\Omega + (\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{w})_\Omega + \nu(\nabla \mathbf{u}, \nabla \mathbf{w})_\Omega = (\mathbf{f}, \mathbf{w}) \quad \forall \mathbf{w} \in \mathcal{V}_d. \tag{3.44}$$

By applying integration by parts to the viscous term we get

$$\left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{w}\right)_\Omega + (\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{w})_\Omega - \nu(\nabla^2 \mathbf{u}, \mathbf{w})_\Omega + (\mathbf{J}(\mathbf{u}), \mathbf{w})_\Gamma = (\mathbf{f}, \mathbf{w}) \quad \forall \mathbf{w} \in \mathcal{V}_d, \tag{3.45}$$

where $\mathbf{J}(\mathbf{u}) = \nu \mathbf{n}_1 \cdot \nabla \mathbf{u}_1 + \nu \mathbf{n}_2 \cdot \nabla \mathbf{u}_2$ is the diffusive flux from adjacent elements sharing a boundary. Equations (3.44) and (3.45) can be equivalently written as,

$$\left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{w}\right)_{\Omega} + (\mathbf{R}(\mathbf{u}), \mathbf{w})_{\Omega} + (\mathbf{J}(\mathbf{u}), \mathbf{w})_{\Gamma} = (\mathbf{f}, \mathbf{w}) \quad \forall \mathbf{w} \in \mathcal{V}_d. \tag{3.46}$$

By decomposing the spaces into $\mathcal{V}_d = \tilde{\mathcal{V}}_d \oplus \mathcal{V}_d'$ and $\mathcal{K} = \tilde{\mathcal{K}} \oplus \mathcal{K}'$, and applying our finite memory based framework leads to the following formulation for the coarse scales $\tilde{\mathbf{u}} \in \tilde{\mathcal{V}}_d$:

$$\left(\frac{\partial \tilde{\mathbf{u}}}{\partial t}, \tilde{\mathbf{w}}\right)_{\Omega'} + (\mathbf{R}(\tilde{\mathbf{u}}), \tilde{\mathbf{w}})_{\Omega'} + (\mathbf{J}(\tilde{\mathbf{u}}), \tilde{\mathbf{w}})_{\Gamma'} = (\mathbf{f}, \tilde{\mathbf{w}}) + \tau(\mathbf{R}'(\mathbf{q}), \tilde{\mathbf{w}})_{\Omega'} + \tau(\mathbf{J}'(\mathbf{q}), \tilde{\mathbf{w}})_{\Gamma'} \quad \forall \tilde{\mathbf{w}} \in \tilde{\mathcal{V}}_d, \tag{3.47}$$

where $\tilde{\mathcal{V}}_d$ is our FE approximation space, and $\mathbf{R}'$ and $\mathbf{b}'$ represent the linearizations of $\mathbf{R}$ and $\mathbf{b}$ with respect to $\tilde{\mathbf{u}}$. The fine-scale variable $\mathbf{q} \in \mathcal{V}_d{}'$ involving projection of the residuals on the fine-scales is given by

$$(\mathbf{q}, \mathbf{w}')_{\Omega'} = (\mathbf{R}(\tilde{\mathbf{u}}) - \mathbf{f}, \mathbf{w}')_{\Omega'} + (\mathbf{J}(\tilde{\mathbf{u}}), \mathbf{w}')_{\Gamma'} \quad \forall \mathbf{w}' \in \mathcal{V}_d{}'. \tag{3.48}$$

In the above equation, we assume that the fine-scales vanish at elemental boundaries *Masud and Calderer* (2011a); *Franca and Farhat* (1995); *Brezzi et al.* (1992); *Hughes* (1995), a thus neglect the second term. The quantity $\mathbf{q}$ is approximated as follows

$$\mathbf{q} = \Pi'(\mathbf{R}(\tilde{\mathbf{u}}) - \mathbf{f}) = (\mathbf{R}(\tilde{\mathbf{u}}) - \mathbf{f}) - \tilde{\Pi}((\mathbf{R}(\tilde{\mathbf{u}}) - \mathbf{f})). \tag{3.49}$$

By further simplifying the memory term we obtain the following:

$$\tau(\mathbf{R}'(\mathbf{q}), \tilde{\mathbf{w}})_{\Omega'} + \tau(\mathbf{J}'(\mathbf{q}), \tilde{\mathbf{w}})_{\Gamma'} = \tau[(\mathbf{q} \cdot \nabla \tilde{\mathbf{u}}, \tilde{\mathbf{w}})_{\Omega'} + (\tilde{\mathbf{u}} \cdot \nabla \mathbf{q}, \tilde{\mathbf{w}})_{\Omega'} - \nu(\mathbf{q}, \nabla^2 \tilde{\mathbf{w}})_{\Omega'}]. \tag{3.50}$$

Where the second term is simplified using the Green's identity and calculated as follows

$$(\tilde{\mathbf{u}}.\nabla \mathbf{q}, \tilde{\mathbf{w}})_{\Omega'} = -(\mathbf{q}, \tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{w}}) - (\nabla \cdot \tilde{\mathbf{u}}, \mathbf{q} \cdot \tilde{\mathbf{w}}). \tag{3.51}$$

From an implementation perspective, all the above terms are computed using numerical integration at the quadrature points. This results in the following problem for the coarse scales $\tilde{\mathbf{u}} \in \tilde{\mathcal{V}}_d$:

$$\left(\frac{\partial \tilde{\mathbf{u}}}{\partial t}, \tilde{\mathbf{w}}\right)_{\Omega'} + (\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}, \tilde{\mathbf{w}})_{\Omega'} + \nu(\nabla \tilde{\mathbf{u}}, \nabla \tilde{\mathbf{w}})_{\Omega'} = (\mathbf{f}, \tilde{\mathbf{w}}) + \tau[(\mathbf{q} \cdot \nabla \tilde{\mathbf{u}}, \tilde{\mathbf{w}})_{\Omega'} + (\tilde{\mathbf{u}} \cdot \nabla \mathbf{q}, \tilde{\mathbf{w}})_{\Omega'} - \nu(\mathbf{q}, \nabla^2 \tilde{\mathbf{w}})_{\Omega'}] \quad \forall \tilde{\mathbf{w}} \in \tilde{\mathcal{V}}_d, \tag{3.52}$$

$$\mathbf{q} = (\mathbf{R}(\tilde{\mathbf{u}}) - \mathbf{f}) - \tilde{\Pi}((\mathbf{R}(\tilde{\mathbf{u}}) - \mathbf{f})). \tag{3.53}$$

The role of pressure herein is to impose the divergence free condition on the velocity field. In this formulation, only the velocity sub-scales have been accounted for, and the pressure terms arising from standard Galerkin procedure are retained and treated like a forcing function. This leads to additional stabilization terms to the standard Galerkin procedure given by,

$$\begin{aligned} &\left(\frac{\partial \tilde{\mathbf{u}}}{\partial t}, \tilde{\mathbf{w}}\right)_{\Omega'} + (\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}, \tilde{\mathbf{w}})_{\Omega'} + \nu(\nabla \tilde{\mathbf{u}}, \nabla \tilde{\mathbf{w}})_{\Omega'} - (\tilde{p}, \nabla \cdot \tilde{\mathbf{w}}) = (\mathbf{f}, \tilde{\mathbf{w}}) \\ &+\tau[(\mathbf{q} \cdot \nabla \tilde{\mathbf{u}}, \tilde{\mathbf{w}})_{\Omega'} + (\tilde{\mathbf{u}} \cdot \nabla \mathbf{q}, \tilde{\mathbf{w}})_{\Omega'} - \nu(\mathbf{q}, \nabla^2 \tilde{\mathbf{w}})_{\Omega'}] \qquad \forall \tilde{\mathbf{w}} \in \tilde{\mathcal{V}}_d, \end{aligned} \tag{3.54}$$

$$\mathbf{q} = (\mathbf{R}(\tilde{\mathbf{u}}) + \nabla \tilde{p} - \mathbf{f}) - \tilde{\Pi}((\mathbf{R}(\tilde{\mathbf{u}}) + \nabla \tilde{p} - \mathbf{f})). \tag{3.55}$$

Although closure terms were obtained for the momentum equations in Equation (3.54), the effect of the velocity sub-scales on the continuity equation should also be accounted for. Hence, an approximate form of the velocity sub-scales is required. To this end, consider Equation (3.47) in a re-arranged form:

$$\left(\frac{\partial \tilde{\mathbf{u}}}{\partial t}, \tilde{\mathbf{w}}\right)_{\Omega'} + (\mathbf{R}(\tilde{\mathbf{u}}) - \tau \mathbf{R}'(\mathbf{q}), \tilde{\mathbf{w}})_{\Omega'} + (\mathbf{J}(\tilde{\mathbf{u}} - \tau \mathbf{q}), \tilde{\mathbf{w}})_{\Gamma'} = (\mathbf{f}, \tilde{\mathbf{w}}) \quad \forall \tilde{\mathbf{w}} \in \tilde{\mathcal{V}}_d, \tag{3.56}$$

where the operators $\mathbf{R}$ and $\mathbf{J}$ are non-linear and linear respectively and $\mathbf{R}'$ is the linearization of $\mathbf{R}$ about $\tilde{\mathbf{u}}$. For small sub-scale $\mathbf{u}'$ *Bazilevs et al.* (2007) approximation, we have,

$$\mathbf{R}(\tilde{\mathbf{u}} + \mathbf{u}') \approx \mathbf{R}(\tilde{\mathbf{u}}) + \mathbf{R}'(\mathbf{u}'). \tag{3.57}$$

Consequently, we can express velocity sub-scales approximately as

$$\mathbf{u}' \approx -\tau \mathbf{q}. \tag{3.58}$$

A similar form of sub-scales was also obtained by Wang et al. *Wang and Oberai* (2010a) by writing an asymptotic series in terms of residual *Bazilevs et al.* (2007). Finally, the effect of the sub-scales on the continuity equation is taken into consideration as follows:

$$(\nabla \cdot (\tilde{\mathbf{u}} + \mathbf{u}'), \tilde{k})_{\Omega'} = 0 \quad \forall \tilde{k} \in \tilde{\mathcal{K}}, \tag{3.59}$$

$$(\nabla \cdot (\tilde{\mathbf{u}} - \tau \mathbf{q}), \tilde{k})_{\Omega'} = 0 \quad \forall \tilde{k} \in \tilde{\mathcal{K}}. \tag{3.60}$$

By applying integration by parts and using the fact that sub-scales vanish at the elemental boundaries, we have the following formulation for the continuity equation:

$$(\nabla . \tilde{\mathbf{u}}, \tilde{k})_{\Omega'} + \tau(\mathbf{q}, \nabla \tilde{k})_{\Omega'} = 0 \quad \forall \tilde{k} \in \tilde{\mathcal{K}}, \tag{3.61}$$

The next step is to discretize the above equation in time using the θ family of methods. This resulting variational problem at each time step is to find $\tilde{\mathbf{u}}^{n+\theta} \in \tilde{\mathcal{V}}_d$ and $\tilde{p}^{n+\theta} \in \tilde{\mathcal{K}}$ such that

$$\left(\frac{\tilde{\mathbf{u}}^{n+1}-\tilde{\mathbf{u}}^{n}}{\Delta t},\tilde{\mathbf{w}}\right)_{\Omega'}+(\tilde{\mathbf{u}}^{n+\theta}\cdot\nabla\tilde{\mathbf{u}}^{n+\theta},\tilde{\mathbf{w}})_{\Omega'}+\nu(\nabla\tilde{\mathbf{u}}^{n+\theta},\nabla\tilde{\mathbf{w}})_{\Omega'}-(\tilde{p}^{n+\theta},\nabla\cdot\tilde{\mathbf{w}})=$$
$$(\mathbf{f}^{n+\theta},\tilde{\mathbf{w}})+\tau[(\mathbf{q}^{n+\theta}\cdot\nabla\tilde{\mathbf{u}}^{n+\theta},\tilde{\mathbf{w}})_{\Omega'}+(\tilde{\mathbf{u}}^{n+\theta}\cdot\nabla\mathbf{q}^{n+\theta},\tilde{\mathbf{w}})_{\Omega'}-\nu(\mathbf{q}^{n+\theta},\nabla^2\tilde{\mathbf{w}})_{\Omega'}]\quad\forall\tilde{\mathbf{w}},\in\tilde{\mathcal{V}}_d \tag{3.62}$$

$$\mathbf{q}^{n+\theta}=(\mathbf{R}(\tilde{\mathbf{u}}^{n+\theta})+\nabla\tilde{p}^{n+\theta}-\mathbf{f}^{n+\theta})-\tilde{\Pi}((\mathbf{R}(\tilde{\mathbf{u}}^{n+\theta})+\nabla\tilde{p}^{n+\theta}-\mathbf{f}^{n+\theta})), \tag{3.63}$$

$$(\nabla.\tilde{\mathbf{u}}^{n+\theta},\tilde{k})_{\Omega'}+\tau(\mathbf{q}^{n+\theta},\nabla\tilde{k})_{\Omega'}=0\quad\forall\tilde{k}\in\tilde{\mathcal{K}}. \tag{3.64}$$

One way to linearize the above set of non-linear equations is by using Picard iteration based technique given by

$$\left(\frac{\tilde{\mathbf{u}}^{n+\theta,i+1}-\tilde{\mathbf{u}}^{n}}{\theta\Delta t},\tilde{\mathbf{w}}\right)_{\Omega'}+(\tilde{\mathbf{u}}^{n+\theta,i}\cdot\nabla\tilde{\mathbf{u}}^{n+\theta,i+1},\tilde{\mathbf{w}})_{\Omega'}+\nu(\nabla\tilde{\mathbf{u}}^{n+\theta,i+1},\nabla\tilde{\mathbf{w}})_{\Omega'}-(\tilde{p}^{n+\theta,i+1},\nabla\cdot\tilde{\mathbf{w}})=$$
$$(\mathbf{f}^{n+\theta,i},\tilde{\mathbf{w}})+\tau[(\mathbf{q}^{n+\theta,i}\cdot\nabla\tilde{\mathbf{u}}^{n+\theta,i+1},\tilde{\mathbf{w}})_{\Omega'}+(\tilde{\mathbf{u}}^{n+\theta,i}\cdot\nabla\mathbf{q}^{a},\tilde{\mathbf{w}})_{\Omega'}-\nu(\mathbf{q}^{a},\nabla^2\tilde{\mathbf{w}})_{\Omega'}]\quad\forall\tilde{\mathbf{w}}\in\tilde{\mathcal{V}}_d, \tag{3.65}$$

where $i+1$ and i denote the present and previous iteration respectively. It can be noted that $\mathbf{q}^{a}$ and $\mathbf{q}^{n+\theta,i}$ are defined differently. This has been done so that $(\tilde{\mathbf{u}}^{n+\theta,i}\cdot\nabla\tilde{\mathbf{u}}^{n+\theta,i+1},\tilde{\mathbf{w}})_{\Omega'}$ and $\tau(\mathbf{q}^{n+\theta,i}\cdot\nabla\tilde{\mathbf{u}}^{n+\theta,i+1},\tilde{\mathbf{w}})_{\Omega'}$ can be merged together. This is possible because $\mathbf{q}^{n+\theta,i}$ is calculated from previous iteration variables. This is similar to Codina's procedure *Codina* (2002) of adding sub-scales to the convective velocity, and a direct consequence of retaining the non-linearity in the VMS formulation *Codina et al.* (2007). Defining $\mathbf{q}^{a}$ in this manner allows for an implicit calculation of the memory terms which is similar to the stabilization

term in *Codina* (2002); *Brooks and Hughes* (1982); *Hughes et al.* (1989, 1986) as follows:

$$\begin{aligned}\mathbf{q}^{a} = (\tilde{\mathbf{u}}^{n+\theta,i} \cdot \nabla\tilde{\mathbf{u}}^{n+\theta,i+1} - \nu\nabla^2\tilde{\mathbf{u}}^{n+\theta,i+1} + \nabla\tilde{p}^{n+\theta,i+1} - \mathbf{f}^{n+\theta,i}) \\ -\tilde{\Pi}(\tilde{\mathbf{u}}^{n+\theta,i} \cdot \nabla\tilde{\mathbf{u}}^{n+\theta,i} - \nu\nabla^2\tilde{\mathbf{u}}^{n+\theta,i} + \nabla\tilde{p}^{n+\theta,i} - \mathbf{f}^{n+\theta,i}),\end{aligned} \tag{3.66}$$

$$\begin{aligned}\mathbf{q}^{n+\theta,i} = (\tilde{\mathbf{u}}^{n+\theta,i} \cdot \nabla\tilde{\mathbf{u}}^{n+\theta,i} - \nu\nabla^2\tilde{\mathbf{u}}^{n+\theta,i} + \nabla\tilde{p}^{n+\theta,i} - \mathbf{f}^{n+\theta,i}) \\ -\tilde{\Pi}(\tilde{\mathbf{u}}^{n+\theta,i} \cdot \nabla\tilde{\mathbf{u}}^{n+\theta,i} - \nu\nabla^2\tilde{\mathbf{u}}^{n+\theta,i} + \nabla\tilde{p}^{n+\theta,i} - \mathbf{f}^{n+\theta,i}),\end{aligned} \tag{3.67}$$

Equations (3.64), (3.65), (3.66) and (3.67) are iterated until convergence of the relative norm of the solution vector between two consecutive iterations is achieved.

3.5.1 Homogeneous Isotropic Turbulence (HIT)

In this section, we present results for decaying homogeneous isotropic turbulence (HIT) and compare it to DNS. We choose the OSS model as a reference, as it has been shown to be a good VMS closure for turbulence *Colomés et al.* (2015). The HIT problem has been extensively studied in literature both numerically *Mansour and Wray* (1994); *Orszag and Patterson Jr* (1972); *Ishida et al.* (2006) and experimentally *Comte-Bellot and Corrsin* (1971). This problem is well defined in a 3-D periodic box and the initialization of the initial velocity field for DNS is done using Rogallo's procedure *Rogallo* (1981) which assumes the following energy spectrum at initial time:

$$E(k, t = 0) = \frac{q^2}{2A}\frac{1}{k_p^{\sigma+1}}k^p exp(\frac{-\sigma}{2}(\frac{k}{k_p})^2), \tag{3.68}$$

where k_p is the wavenumber at which the energy spectra peaks and A is defined as $\int_0^\infty k^\sigma exp(-\sigma k^2/2)dk$. The velocity in spectral space is given by

$$\mathbf{a}(\mathbf{k}) = (\frac{\alpha k k_2 + \beta k k_1}{k(k_1^2 + k_2^2)^{1/2}})\hat{i} + (\frac{\beta k_2 k_3 - \alpha k_1 k_3}{k(k_1^2 + k_2^2)^{1/2}})\hat{j} - (\frac{\beta(k_1^2 + k_2^2)^{1/2}}{k})\hat{k}, \tag{3.69}$$

where k denotes the magnitude of the wavenumber vector and α and β are defined as follows:

$$\alpha = (\frac{E(k)}{4\pi k^2})^{1/2} e^{i\theta_1} cos(\phi), \quad \beta = (\frac{E(k)}{4\pi k^2})^{1/2} e^{i\theta_2} sin(\phi), \tag{3.70}$$

where ϕ, θ_1, θ_2 are uniformly distributed random numbers from 0 to 2π. In all the simulations, $k_p = 3$, $q^2 = 3$ and $\sigma = 4$. Although, the initial velocity field satisfies the divergence free condition, it does not represent a physical homogeneous isotropic turbulent velocity field. To achieve this state, the field is allowed to decay to a lower Re_λ where the field will resemble a more realistic velocity field due to redistribution of energy *Mansour and Wray* (1994). Three different initial Re_λ have been considered here: $Re_\lambda \approx 65$, 75 and 164 where Re_λ is defined as the Reynolds number based on the Taylor microscale λ as follows:

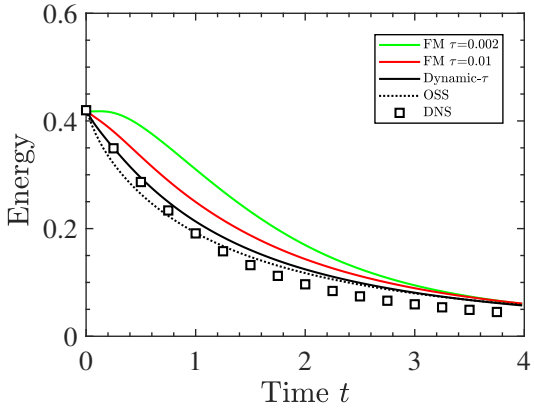
$$Re_\lambda = \frac{u'\lambda}{\nu}, \tag{3.71}$$

where u' is the velocity fluctuation/root mean square (rms) of the velocity field defined as $\sqrt{2k/3}$. The initial conditions for all these cases are generated from DNS simulations by starting at a higher Re_λ and allowing it to reach our target Re_λ of 65, 75 and 164 respectively. The kinematic viscosity for the three different cases are set to $\nu = 0.001$, 0.0005 and 0.0001 respectively. The LES simulations utilize 64^3 linear elements for all the three Re_λ cases, the results of which are presented in Figures 3.10,3.11 and 3.12 respectively. This allows us to study the effects of increasing Re_λ by retaining the same resolution.

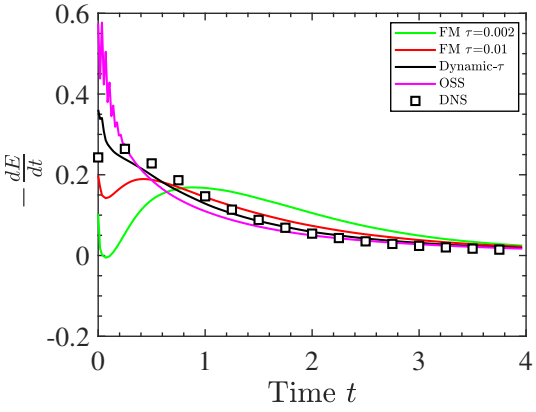
At a relatively lower Reynolds number of $Re_\lambda \approx 65$, all the models perform fairly well in predicting the time history of the resolved kinetic energy except the fixed memory models where an arbitrary choice of τ is used. As can be seen from the kinetic energy decay plots in Figures 3.10b, 3.11b and 3.12b, for all the three different Reynolds numbers, the OSS model is not stable and can be seen to oscillate initially. This indicates that for the initial time

period, the OSS model incorrectly forces the turbulence. We compare the energy spectra at two different times of T=2.0 and T=4.0 in Figure 3.11c and 3.11d respectively and observe that all models perform very well at the lower wavenumber modes. However, the OSS is clearly more dissipative at higher wave-numbers where it predicts a lower energy content compared to the present model and DNS. At the higher Re_λ = 75 case, we find that both the Dynamic-τ and the OSS model predict reasonably the evolution of kinetic energy and rate of kinetic energy decay. If we compare the energy spectra at T=2.0 in Figure 3.11c, all the models predict a higher energy content across different wave-numbers compared to the DNS solution with the dynamic-τ again performing better at higher wave-numbers again. However, at T=4.0, when it decays to a lower Re_λ, the performance of all the models improve, as can be seen in Figure 3.11d. At the highest Re_λ case, Re_λ = 174, we find that the performance of all the model becomes worse compared to the DNS results. The energy spectra for this case in presented is Figure 3.12c where only the lower wavenumber modes are resolved accurately in comparison to DNS. One possible reason for the deterioration of performance at high Re_λ cases is that the current VMS models are efficient in modeling the cross-stress terms and not the Reynolds stress terms *Wang and Oberai* (2010a) which dominate at higher Re_λ values.

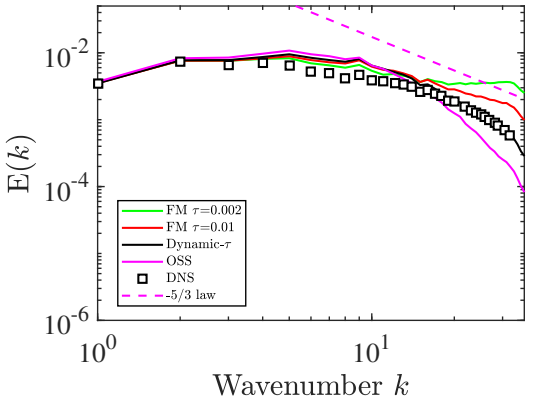
The time variation of the predicted dynamic memory length for all the three cases is shown in Figure 3.13. From the plots, it can be observed that the memory length increases almost linearly with time similar to the viscous Burgers equation. This is consistent with Stinis *Stinis* (2013) of re-normalizing the t-model for stability and accuracy. Also, the predicted value for τ by our dynamic model is higher in comparison to the randomly chosen τ values for our fixed memory model. The differences in the temporal evolution between the dynamically-selected τ and the imposed τ is a further indicate that the dynamic model is necessary for calculating the memory length.



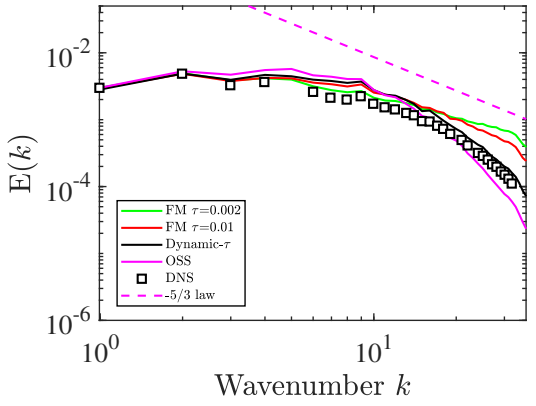
(a) Time evolution of kinetic energy.



(b) Rate of kinetic energy decay.

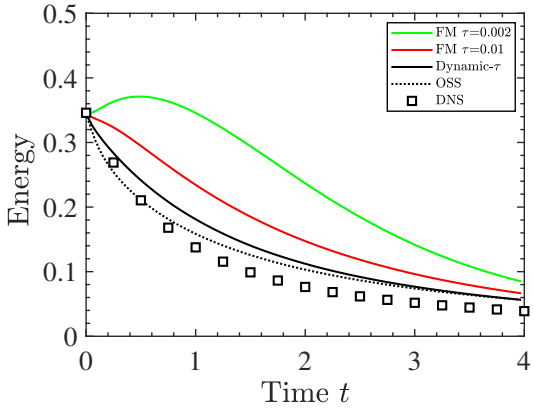


(c) Energy Spectra at T=2.0.

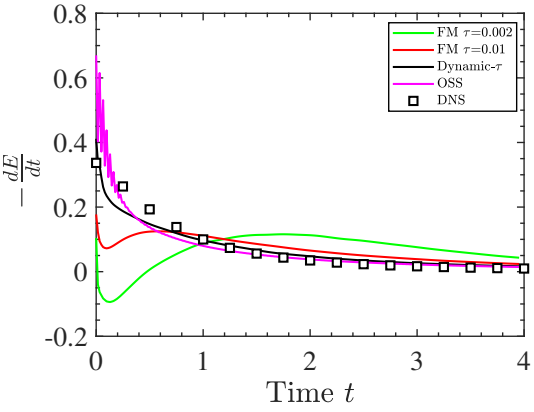


(d) Energy Spectra at T=4.0.

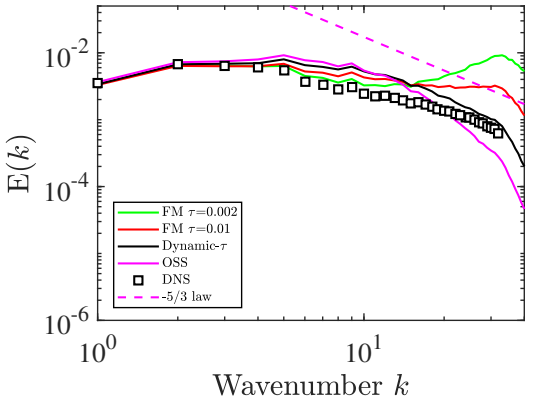
Figure 3.10: (a) Kinetic energy, (b) dissipation, (c) energy spectra at T = 2 and (d) energy spectra at T=4 for homogeneous isotropic turbulence at initial $Re_\lambda \approx 65$.



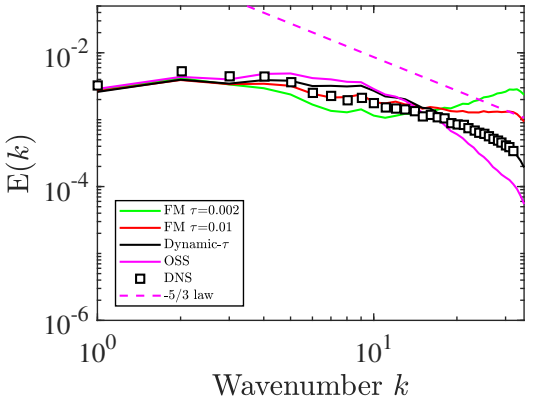
(a) Time evolution of kinetic energy.



(b) Rate of kinetic energy decay.

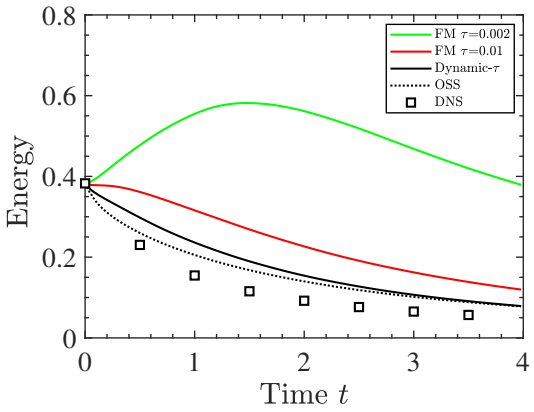


(c) Energy Spectra at T=2.0.

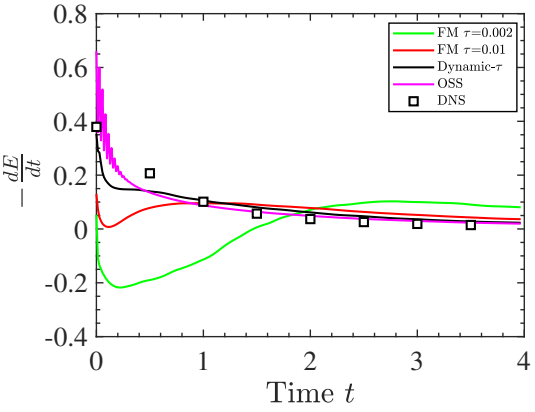


(d) Energy Spectra at T=4.0.

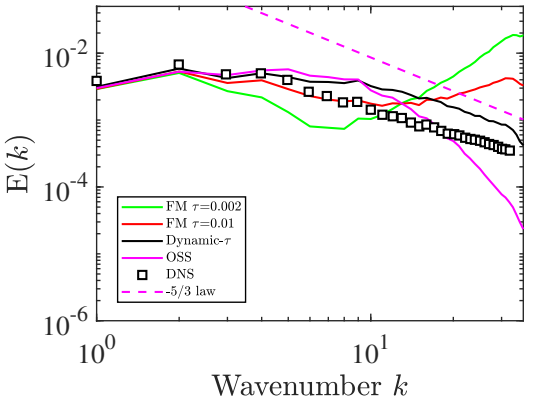
Figure 3.11: (a) Kinetic energy, (b) dissipation, (c) energy spectra at T = 2 and (d) energy spectra at T=4 for homogeneous isotropic turbulence at initial $Re_\lambda \approx 75$.



(a) Time evolution of kinetic energy.

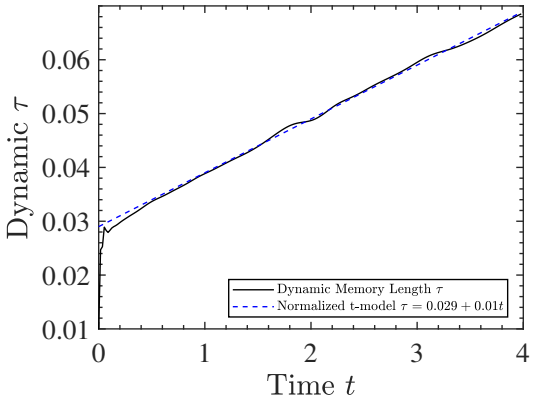


(b) Rate of kinetic energy decay.

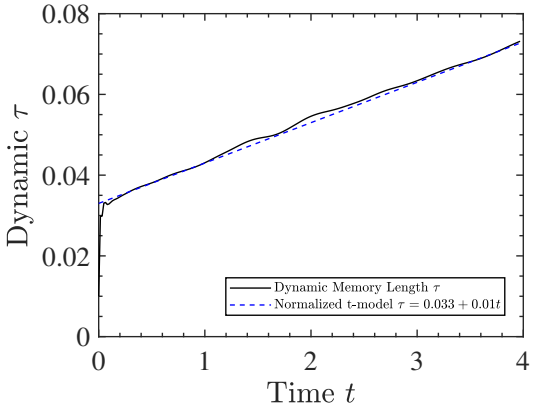


(c) Energy Spectra at T=4.0.

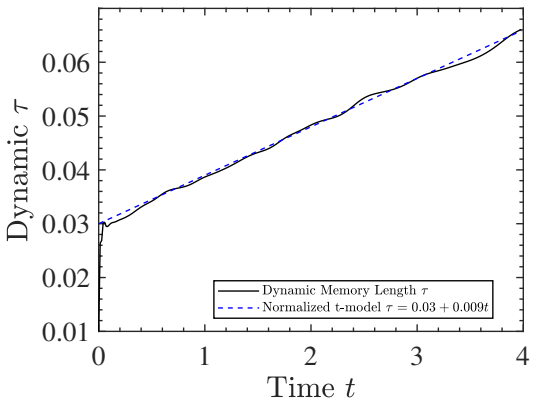
Figure 3.12: (a) Kinetic energy, (b) dissipation and (c) energy spectra at T=4 for homogeneous isotropic turbulence at initial $Re_\lambda \approx 164$.



(a) Initial $Re_\lambda \approx 65$.



(b) Initial $Re_\lambda \approx 75$.



(c) Initial $Re_\lambda \approx 164$.

Figure 3.13: Evolution of memory length τ predicted using our dynamic model for homogeneous isotropic turbulence for different initial Re_τ.

3.5.2 Taylor Green Vortex (TGV)

The next step in understanding the applicability of the proposed method is to employ the model on a turbulent flow that undergoes complex dynamics such the Taylor-Green vortex. This problem involves transition to turbulence-like flow, as well as decay. Models such as the Smagorinsky which have been derived based on assumptions of homogeneity, isotropy and balance between sub-grid production and dissipation *Pope and Pope* (2000) might not optimally preform in such flows where there is non-homogenity and transition to turbulence. Similar to HIT, this problem is well-defined on a 3-D periodic box with smooth initial conditions which are given as follows:

$$u = U_o cos(x) sin(y) cos(z), \quad v = -U_o sin(x) cos(y) cos(z), \quad w = 0 \tag{3.72}$$

where u, v, w denote the velocity in x, y and z directions respectively and $x, y, z \in [-\pi L, \pi L]$. To study this problem, three different Reynolds numbers are considered: $Re = \frac{U_o L}{\nu} = 400$, 800 and 1600. The values for L and U_o are unity and the Re is changed solely by varying the kinematic viscosity ν. The initial conditions for the velocity field are kept the same for all the cases.

The profiles for the resolved kinetic energy for $Re = 400$, 800 and 1600 are shown in Figures 3.14a, 3.15a and 3.16a respectively. The evolution of the kinetic energy indicates that both the OSS and Dynamic-τ perform well with only 32^3 degrees of freedom. Similar trends are also observed for the rate of KE energy decay for $Re = 400$, 800 and 1600 in Figures 3.14b,3.15b and 3.16b respectively, where the fixed τ models fail to accurately predict the correct results in comparison to DNS. When 48^3 and 64^3 degrees of freedoms are used for $Re = 800$ and 1600 respectively, there is an overall improvement in the results for the fixed memory model. The present dynamic model and the OSS model perform well at finer resolutions.

Figures 3.14c, 3.15c and 3.16c, and Figures 3.14d, 3.15d and 3.16d show the energy

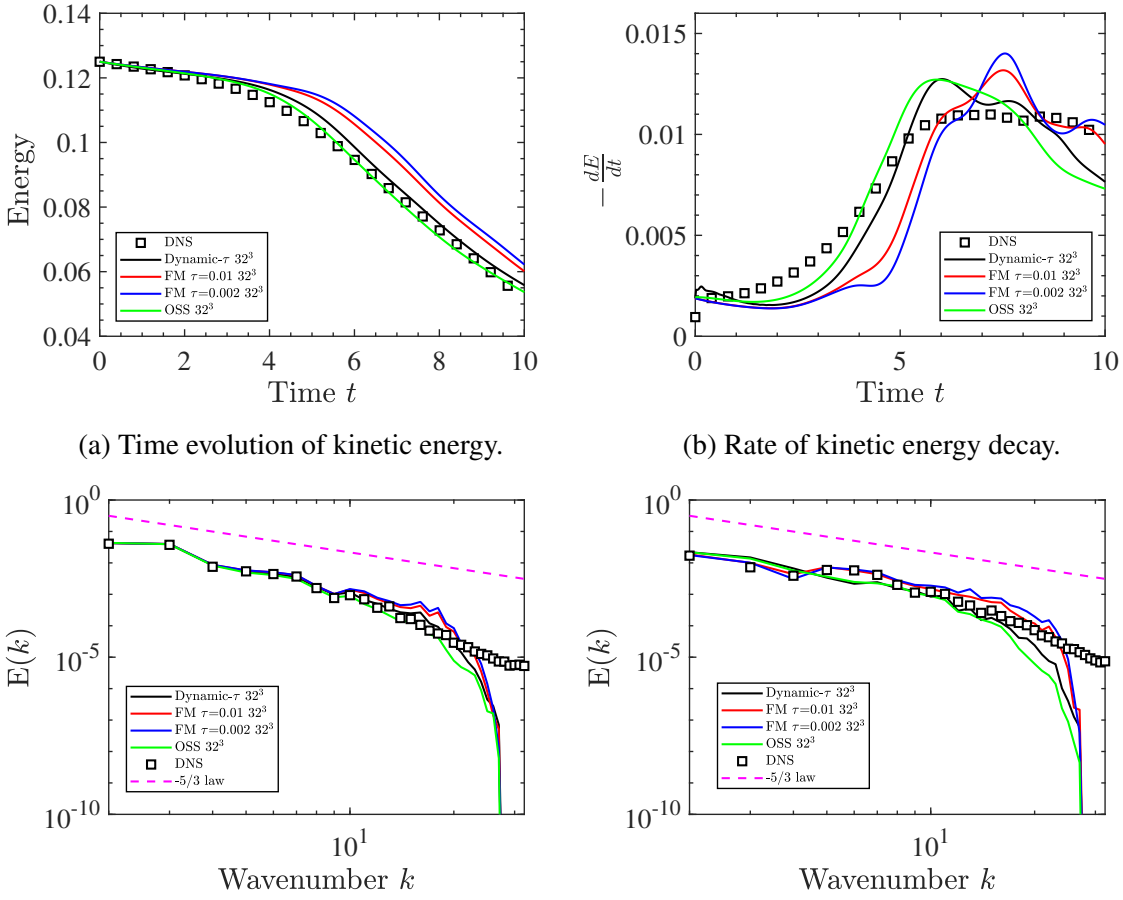
spectra of the resolved velocity fields at two time instants $T = 5.0$ and $T = 10.0$ respectively. At $T = 5.0$, all the models are in agreement with DNS at the low wavenumber modes even with just 32^3 degrees of freedom. However, the constant τ models produces a build-up of energy which grows with Re at high-wavenumber modes. This suggests that either an incorrect value of τ is used for the FM models or the assumption of constant memory length throughout the simulation is not very accurate. As a result, model with constant memory length is not capable of producing enough dissipation and the energy increases at the high wavenumber modes. At a later time $T = 10.0$, a similar trend is also observed with the constant τ model where energy increases at high-wave numbers. On the other hand, the dynamic τ-model and OSS do not result in energy increase at high wavenumbers. Although our dynamic model and OSS model perform closely for the 32^3 cases, at higher resolutions OSS is clearly more dissipative wherein lower energy is present at high wavenumber especially for the high Reynolds number case. In spite of the OSS model and the dynamic-τ model performing closely, the stabilization parameter in OSS and the memory length in dynamic-τ is computed differently.

Case	$DOFs$	dx	dt	ν	U_0	L	FM τ's
DNS-400	64^3	9.81×10^{-2}	2×10^{-2}	2.5×10^{-3}	1	1	-
LES-FM-400	32^3	1.96×10^{-1}	1.96×10^{-2}	2.5×10^{-3}	1	1	0.01 and 0.002
LES-DY-400	32^3	1.96×10^{-1}	1.96×10^{-2}	2.5×10^{-3}	1	1	Dynamic
DNS-800	128^3	4.90×10^{-2}	2×10^{-2}	1.25×10^{-3}	1	1	-
LES-FM-800	$32^3, 48^3$	$1.96 \times 10^{-1}, 1.31 \times 10^{-1}$	1.96×10^{-2}	1.25×10^{-3}	1	1	0.01 and 0.002
LES-DY-800	$32^3, 48^3$	$1.96 \times 10^{-1}, 1.31 \times 10^{-1}$	1.96×10^{-2}	1.25×10^{-3}	1	1	Dynamic
DNS-1600	256^3	2.45×10^{-2}	5×10^{-3}	6.25×10^{-4}	1	1	-
LES-FM-1600	$32^3, 64^3$	$1.96 \times 10^{-1}, 9.81 \times 10^{-2}$	$1.96 \times 10^{-2}, 10^{-2}$	6.25×10^{-4}	1	1	0.01 and 0.002
LES-DY-1600	$32^3, 64^3$	$1.96 \times 10^{-1}, 9.81 \times 10^{-2}$	10^{-2}	6.25×10^{-4}	1	1	Dynamic

Table 3.3: Simulation parameters for DNS and LES of the Taylor Green Vortex problem.

3.6 Conclusion

The Variational Multiscale method and the Mori-Zwanzig formalism are combined within the Continuous Galerkin method to develop coarse grained models for multiscale PDEs. This approach utilizes the Variational Multiscale method to separate scales with the



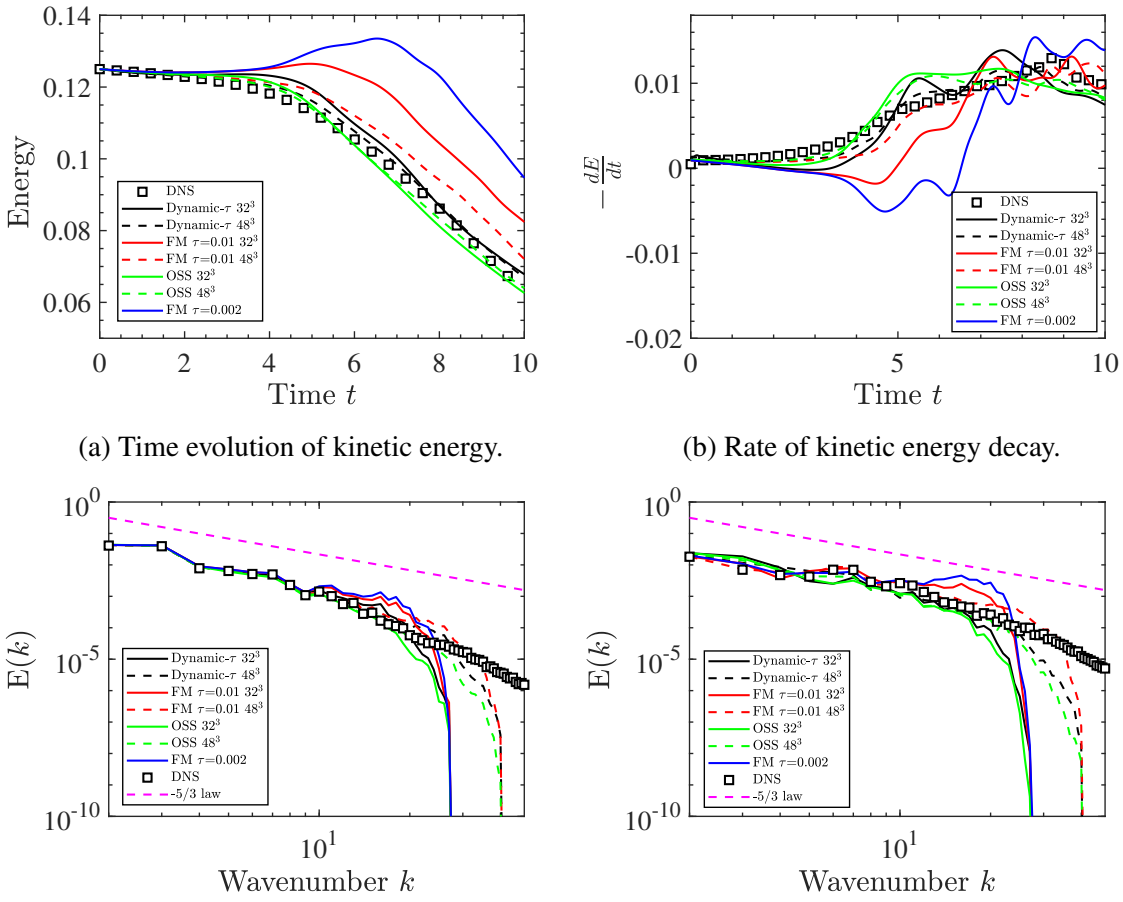
(a) Time evolution of kinetic energy.

(b) Rate of kinetic energy decay.

(c) Energy Spectra at T=5.0.

(d) Energy Spectra at T=10.0.

Figure 3.14: (a) Kinetic energy, (b) dissipation, (c) energy spectra at T = 5 and (d) energy spectra at T=10 for Taylor Green vortex at Re=400 using different coarse graining methods.



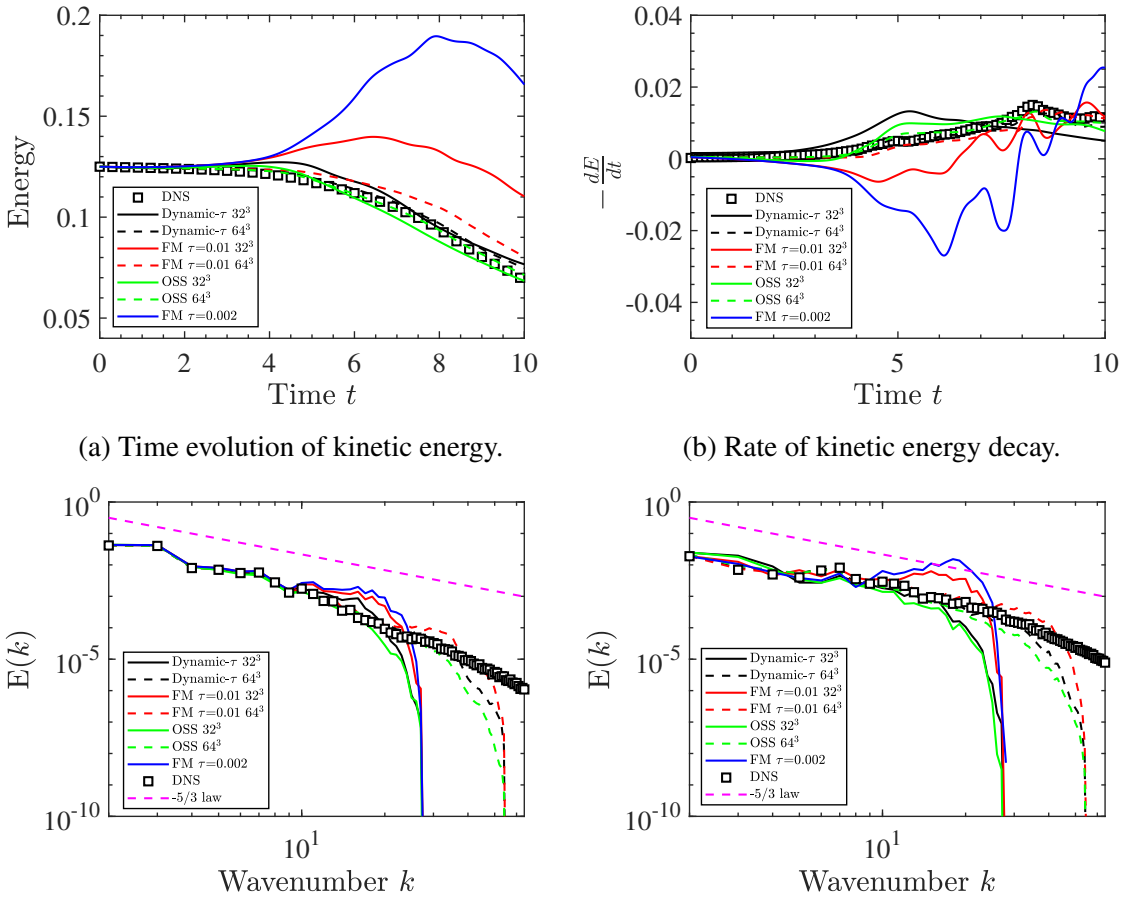
(a) Time evolution of kinetic energy.

(b) Rate of kinetic energy decay.

(c) Energy Spectra at T=5.0.

(d) Energy Spectra at T=10.0.

Figure 3.15: (a) Kinetic energy, (b) dissipation, (c) energy spectra at T = 5 and (d) energy spectra at T=10 for Taylor Green vortex at Re=800 using different coarse graining methods.



(a) Time evolution of kinetic energy.

(b) Rate of kinetic energy decay.

(c) Energy Spectra at T=5.0.

(d) Energy Spectra at T=10.0.

Figure 3.16: (a) Kinetic energy, (b) dissipation, (c) energy spectra at T = 5 and (d) energy spectra at T=10 for Taylor Green vortex at Re=1600 using different coarse graining methods.

capability of the Mori-Zwanzig formalism to represent the impact of unresolved dynamics on the resolved dynamics. This approach - in a similar spirit to existing non-linear VMS models *Codina et al.* (2007); *Codina* (2002); *Bazilevs et al.* (2007); *Wang and Oberai* (2010a); *Gravemeier et al.* (2010); *Masud and Calderer* (2011b) - is developed to provide sub-grid scale/stabilization models without phenomenological assumptions. This procedure is generalizable and can be applied to arbitrarily complex non-linear multiscale PDEs. In context of turbulent flows, this approach provides a general framework for large-eddy simulation that eschews assumptions such as those based on energy balance between scale. Sub-grid scale models are developed for the Burgers equation and the Navier-Stokes using the proposed approach. The sub-grid scale models include a parameter called the memory length, τ, which represents the the time correlation of unresolved dynamics, and controls the stabilization. We impose different memory lengths τ and observe that there is an optimum memory length τ which provides results comparable to the full order solution *Parish* (2018). To avoid the imposition of an adhoc memory length, and recognizing that the model should adapt to the instantaneous level of resolution, we derived a dynamic-τ model and found that it can accurately predict the temporal evolution of τ. The predicted value of τ was observed to be generally linear in time, as conceptualized by the renormalized t-model *Stinis* (2013). In general, for the range of problems that were investigated, the proposed technique performs favorably in comparison to existing counterparts.

CHAPTER IV

Data-Driven Reconstruction and Predictive Modeling of Unresolved Physics

4.1 Background and Motivation

The availability of high resolution data from numerical simulations and experiments in the past decade has led to an interest in data-based modeling. Applications of machine learning augmentations have been used in RANS (e.g. *Singh and Duraisamy* (2016); *Parish and Duraisamy* (2016); *Ling et al.* (2016); *Wang et al.* (2017) and LES (e.g. *Sarghini et al.* (2003); *Gamahara and Hattori* (2017); *Xie et al.* (2019a); *Beck et al.* (2019)). In the LES front, Maulik et al. (*Maulik et al.* (2019)) used machine learning to classify and blend different LES models to select the most accurate model at run-time. Yang. et al *Yang et al.* (2019) used physics-informed features to improve the performance of equilibrium LES wall models in non-equilibrium cases. Similarly, many other notable attempts to improve LES models using data have also been made by Maulik et al. *Maulik and San* (2017); *Maulik et al.* (2018, 2019), Beck et al. *Beck et al.* (2019), Sarghini et al. *Sarghini et al.* (2003), Ghamara and Hattori *Gamahara and Hattori* (2017), Wang et. al *Wang et al.* (2018) and many more *Xie et al.* (2019a,b, 2020). These data-driven techniques have also found application in developing closures for reduced-order models (ROMs) *Mou et al.* (2021); *Xie et al.* (2018a); *Mohebujjaman et al.* (2019); *Wang et al.* (2020b).

Very recently, super-resolution of turbulent-flow fields has been pursued using neural networks *Xie et al.* (2018b); *Deng et al.* (2019); *Liu et al.* (2020); *Fukami et al.* (2019); *Kim et al.* (2020); *Fukami et al.* (2021). Xie et al. *Xie et al.* (2018b) and Fukami et al. *Fukami et al.* (2019) appear to be the first to introduce this idea in fluid dynamic applications. These were followed by Deng et al. *Deng et al.* (2019) who improved traditional GAN performance by augmenting the model architecture. Improvements in flow field reconstruction were shown by Liu et al. *Liu et al.* (2020), using both spatial and temporal information. Fukami et al. *Fukami et al.* (2021) performed super-resolution and in-betweening to reconstruct a highly-resolved space-time solution using two low-resolution snapshots taken at the start t and the end $t + \Delta t$ of an interval. These models have demonstrated an ability to reconstruct fine scales from highly coarse-grained data for either the same or similar data-set on which they have been trained both in a supervised and an unsupervised setting *Kim et al.* (2020). However, applying the trained models to super-resolve coarse flow-fields at different Reynolds numbers or another part of the flow, is relatively unexplored.

The idea of using the super-resolved field to compute the closure terms is similar to the ADM approach. However, compared to the approximately deconvolved solution that lies on the same mesh as the filtered solution, the super-resolved solution lies on a higher resolution mesh. A super-resolution model capable of reconstructing fine-space data for a case where the fine-space data already exists and the same information is used for training has no use in a predictive setting. This work attempts to improve predictive capabilities by bringing in generalizable model forms and features. In addition to the generalizability of these models, the definition of a coarse-space solution is ambiguous. This ambiguity is because a variety of low-fidelity data, including LES solution, obtained using finite difference method (FDM), finite volume method (FVM), spectrally filtered DNS solution, and stabilized finite element (FE) solution on a coarse grid qualify as coarse-solutions. The nodal or modal values in each of these methods represent different quantities. For

example, Fukami et al. *Fukami et al.* (2019) used the max-pooling operation to obtain coarse data. Consequently, the trained model is dependent on the type of method or filter used to generate data, and the mapping learned by the network has no formal basis. To resolve this ambiguity, we define both our coarse space and our fine space in terms of the L_2 projection of the DNS solution on low and high order polynomial basis functions in a similar spirit to the Variational Multiscale Method. As a result, the trained model will approximate the function that maps the L_2 projection of the DNS data on the two sub-spaces. Additionally, the model should be preferably compact and applied patch-wise rather than on the entire flow. This is because there is no guarantee that the coarse data that needs to be super-resolved has the same size as that of the input layer, and interpolating it back to the network size defeats the purpose of super-resolution. In this work, we develop N-N closures that are: (i.) capable of extrapolating to unseen flow conditions and resolutions; (ii.) use non-dimensional features rather than dimensional features for better generalizability; and (iii.) can be applied patch-wise rather than on the entire field.

The outline of this chapter is as follows: We re-introduce the VMS methodology in section 4.2. In section 4.3, we derive VMS consistent features. In section 4.4, we propose a model form and a new network architecture for learning it. We describe the procedure of generating training data in section 4.5. In section 4.6, we apply our approach to the linear advection problem both in an online (numerical method) and offline (super-resolution) setting. In section 4.7, we evaluate the performance of a model approach to the turbulent channel flow. In section 4.8, we apply our approach to the 1-D convection diffusion equation to discover high-order CG schemes. Finally, we summarize this chapter in section 4.9.

4.2 The Variational Multiscale (VMS) Method

This section re-summarizes the Variational Multiscale Method (VMS) *Hughes et al.* (1998a), however, in the context of super-resolution. As discussed previously, this method has been extensively used for developing closures for both linear *Hughes et al.* (1998a,

1989); *Brooks and Hughes* (1982); *Codina* (2000); *Hughes et al.* (1986) and non-linear PDEs *Codina et al.* (2007); *Codina* (2002); *Bazilevs et al.* (2007); *Wang and Oberai* (2010a); *Gravemeier et al.* (2010); *Masud and Calderer* (2011b). This section will only discuss it in the context of a linear problem and use it as a guiding principle for feature selection and in shaping the network architecture. As discussed by Hughes et al. *Hughes et al.* (1998a), the development of VMS closure can be broadly categorized into two different subsections based on the type of basis functions used, which are detailed below, along with a context for super-resolution.

4.2.1 Smooth Case

In the 'smooth case', the basis functions are sufficiently smooth so that the distributional effects may be ignored *Hughes et al.* (1998a). Both the Fourier basis and the Chebyshev spectral basis qualify as a smooth basis. For this case, consider the following PDE on an open and bounded domain $\Omega \subset \mathbb{R}^d$, where $d \geq 1$ is the dimension of the problem, with a smooth boundary $\Gamma = \partial\Omega$:

$$\mathcal{L}(u) = f \quad in \quad \Omega, \;\; u = g \quad on \quad \Gamma \tag{4.1}$$

where the operator $\mathcal{L}$ can be linear or non-linear, the functions $f : \Omega \to \mathbb{R}$ and $g : \Gamma \to \mathbb{R}$ are given. The variational form of the above PDE is given by

$$(\mathcal{L}(u), w) = (f, w), \tag{4.2}$$

such that $u \in \mathcal{V}$ for all $w \in \mathcal{V}$, where $(\cdot, \cdot)$ denotes the $L_2(\Omega)$ inner product, and $\mathcal{V} \equiv \mathcal{H}^1(\Omega)$ is the Sobolev space. The solution and weighting space are decomposed as follows:

$$\mathcal{V} = \mathcal{V}_h \oplus \mathcal{V}', \tag{4.3}$$

where $\oplus$ represents a direct sum of $\mathcal{V}_h$ and $\mathcal{V}'$. Applying the VMS operation, we have

$$(\mathcal{L}(u_h + u'), w_h) + (\mathcal{L}(u_h + u'), w') = (f, w_h) + (f, w'). \tag{4.4}$$

While the above equation is valid for both non-linear and linear equations, further simplifications can be made if the differential operator is assumed to be linear. To this end, using the linear independence of w_h and w', and taking the differential operator to be linear, we obtain the coarse and fine equations :

$$(\mathcal{L}(u_h), w_h) + (\mathcal{L}(u'), w_h) = (f, w_h) \tag{4.5}$$

$$(\mathcal{L}(u'), w') = -(\mathcal{L}u_h - f, w'). \tag{4.6}$$

The coarse and fine scale equations can be re-written as:

$$(\mathcal{L}(u_h) - f, w_h) = -(\mathcal{L}(u'), w_h) \tag{4.7}$$

$$\Pi'\mathcal{L}(u') = -\Pi'(\mathcal{L}u_h - f), \tag{4.8}$$

where Π' is the L_2-projector on the fine-scale basis functions. The Green's function corresponding to the adjoint of the fine-scale problem is found by solving the following equations

$$\Pi'\mathcal{L}^*(g'(x, y)) = \Pi'(\delta(x - y)) \quad \forall x \in \Omega \;\; ; \;\; g'(x, y) = 0 \quad \forall x \in \Gamma. \tag{4.9}$$

The fine-scale can be obtained by super-position as follows:

$$u'(y) = -\int_{\Omega} g'(x, y)(\mathcal{L}u_h - f)(x) d\Omega_x. \tag{4.10}$$

In the current super-resolution approach, we do not seek u'. Rather, we seek u'_f, which is the optimal projection of u' on the finer-space w_f. The space of functions in w_f is finer in-comparison to w_h or represents a different kind of space. To this end, the optimal projection of u' on w_f is given by:

$$u'_f(y) = \Pi_f u'(y) = -\int_{\Omega} \Pi_f(g'(x,y))(\mathcal{L}u_h - f)(x)d\Omega_x, \tag{4.11}$$

where $\Pi_f(g'(x,y))$ is L_2-projection of $g'(x,y)$ on $w_f(y)$. For this case, the fine space can be constructed using higher wavenumber Fourier modes or higher-order Chebyshev polynomials.

4.2.2 Rough Case

In the 'rough case', the lack of continuity of derivatives at element interfaces requires us to account for the distributional effects explicitly *Hughes et al.* (1998a). This case is typical of the finite element method, where piece-wise continuous polynomial functions are used within each element. Similar to the 'smooth case', Hughes et al. *Hughes et al.* (1998a) showed that the exact form of sub-scales in the case of finite elements is given by:

$$u'(y) = -\sum_e \left(\int_{\Omega_e} g'(x,y)(\mathcal{L}u_h - f)(x)d\Omega_x + \int_{\Gamma_e} g'(x,y)(bu_h)(x)d\Gamma_x \right). \tag{4.12}$$

It has to be mentioned that the 'smooth case' can be considered as a limiting case of the 'rough case' when a single element is used and element interfaces are not present. The sub-scale solution depends on the coarse-scale inside the element and its neighbors. Applying

the projection operator on both sides of equation (4.12) we obtain

$$u'_f(y) = \Pi_f u'(y) = -\sum_e \Big(\int_{\Omega_e} \Pi_f(g'(x,y))(\mathcal{L}u_h - f)(x) d\Omega_x + \int_{\Gamma_e} \Pi_f(g'(x,y))(bu_h)(x) d\Gamma_x \Big). \tag{4.13}$$

An approximation to the above equation is given in the form of compact bubble functions which vanish at the element boundaries *Masud and Calderer* (2011a); *Franca and Farhat* (1995); *Brezzi et al.* (1992); *Hughes* (1995). For 1-D linear problems, solving the element Green's function leads to the coarse-scale solution being the endpoint interpolant of the actual solution *Hughes et al.* (1998a). Assuming that the coarse-scale is given in the form of the endpoint interpolant of the true solution, the fine-scale solution is given by

$$u'_f(y) = \Pi_f u'(y) = -\int_{\Omega_e} \Pi_f(g'(x,y))(\mathcal{L}u_h - f)(x) d\Omega_x. \tag{4.14}$$

A point to note is that application of the projection operator $\Pi_f(y)$ on the element Green's function $g'(x,y)$ leads to the reduction of dimension only in y, i.e.

$$g'_f(x,y) = \Pi_f g'(x,y) = \sum_i g'_{x,i}(x)\psi_i(y/h), \tag{4.15}$$

where the basis coefficients $g'_{x,i}(x)$ are functions of x, which are not necessarily polynomials. However, if the polynomial order p_c of the coarse-scale is given, the coarse-scale residual for (e.g.) the convection-diffusion equation will be of polynomial order $p_C - 1$. Thus, projecting $\sum g'_{x,i}(x)$ onto the space of polynomials with order $p_C - 1$ and representing $g'(x,y)$ onto tensor-product basis functions in x and y is sufficient. The polynomial order of y is determined by the polynomial order of fine-scales, whereas the polynomial order of x is decided by the maximum polynomial order arising in the coarse-scale residual. For the

convection-diffusion problem, the element Green's function is given by:

$$g(x,y) = \begin{cases} C_1(y)\left(1 - e^{-2\alpha x/h}\right), & \text{if } x \le y \\ C_2(y)\left(e^{-2\alpha x/h} - e^{-2\alpha}\right), & x \ge y \end{cases}$$

where α is the cell Peclet number $\alpha = \frac{ah}{2\kappa}$ and the functions $C_1(y)$ and $C_2(y)$ are defined as

$$C_1(y) = \frac{1 - \mathrm{e}^{-2\alpha(1-(y/h))}}{a\left(1 - \mathrm{e}^{-2\alpha}\right)}, \quad C_2(y) = \frac{e^{2\alpha(y/h)} - 1}{a\left(1 - e^{-2\alpha}\right)}. \tag{4.16}$$

The element's Green's function approximated using different order tensor-product basis functions in x and y is shown in Figure 4.1. The basis functions used here to approximate the sub-scale do not necessarily vanish at the element boundary. However, one can also select them to ensure that the sub-scales vanish at the element boundary, similar to a bubble function, as shown in Figure 4.2. Hence, when the input and output order is fixed, the Green's function for the fine-scales can be represented in a finite number of dimensions. This makes it easier to learn the mapping between the coarse-scale and fine-scale solutions. On further inspection, we find that $ag'(x,y)$ is a function of $\alpha, x/h$ and y/h. Consequently, $ag'_f(x,y)$ can be written as follows:

$$ag'_f(x,y) = \sum_{i,j} g^a_{ij}(\alpha)\phi_i(y/h)\psi_j(x/h), \tag{4.17}$$

where $\phi_i(y/h)$ and $\psi_j(x/h)$ are 1-D basis functions constituting the tensor-product basis functions. Substituting back in equation (4.14), we obtain:

$$u'_f(y) = -\sum_i \phi_i(y/h) \sum_j \int_{\Omega_e} g^a_{ij}(\alpha)\psi_j(x/h)\left(\frac{du_h}{dx} - \frac{\kappa}{a}\frac{d^2u_h}{dx^2}\right)d\Omega_x. \tag{4.18}$$

Next we introduce a constant u_m in the coarse-scale residual calculation as follows:

$$u'_f(y) = -\sum_i \phi_i(y/h) \sum_j \int_{\Omega_e} g^a_{ij}(\alpha)\psi_j(x/h)\left(\frac{d(u_h - u_m)}{dx} - \frac{\kappa}{a}\frac{d^2(u_h - u_m)}{dx^2}\right)d\Omega_x. \quad (4.19)$$

It can be observed that subtracting u_m from u_c does not introduce any error in equation (4.19) because we are taking derivatives of a constant. Writing the coarse-scale solution and the constant u_m in terms of the nodal Lagrange basis functions $u_h = \sum_k u_{h,k} w_{h,k}(x/h)$ and $u_m = \sum_k u_m w_{h,k}(x/h)$, and substituting in equation (4.19) we get

$$u'_f(y) = -\sum_i \phi_i(y/h) \sum_j \int_{\Omega_e} g^a_{ij}(\alpha)\psi_j(x/h) \sum_k (u_{h,k} - u_m)\left(w'_{h,k}(x/h) - \frac{1}{\alpha} w''_{h,k}(x/h)\right)\frac{d\Omega_x}{h}. \quad (4.20)$$

Dividing both sides with a normalizing parameter u_{rms} (which will be defined later) and re-arranging we obtain:

$$\frac{u'_f(y)}{u_{rms}} = -\sum_i \phi_i(y/h) \sum_k \frac{u_{h,k} - u_m}{u_{rms}} \int_{\Omega_e} \left(w'_{h,k}(x/h) - \frac{1}{\alpha} w''_{h,k}(x/h)\right) \sum_j \left(g^a_{ij}(\alpha)\psi_j(x/h)\right)\frac{d\Omega_x}{h}, \quad (4.21)$$

where the integral $\int_{\Omega_e} \left(w'_{h,k}(x/h) - \frac{1}{\alpha} w''_{h,k}(x/h)\right) \sum_j \left(g^a_{ij}(\alpha)\psi_j(x/h)\right) \frac{d\Omega_x}{h}$ is a function of α which can be written as linear combinations of $g^a_{ij}(\alpha)$ and $\frac{g^a_{ij}(\alpha)}{\alpha}$.

With this normalization, the problem of learning the sub-scales reduces to learning the projected element Green's function basis coefficients $g^a_{ij}(\alpha)$ which define the shapes of the surfaces plotted in Figure 4.1. Equation (4.21) also suggests that the normalised sub-scale basis coefficients can be written as sum of the products of normalized coarse scales basis coefficients and functions of α. Further, the number of such functions of α needed to be learnt are finite. Equation (4.21) also suggests for a linear problem that the sub-scales depend linearly on the coarse-scale basis coefficients. The dependence on α,

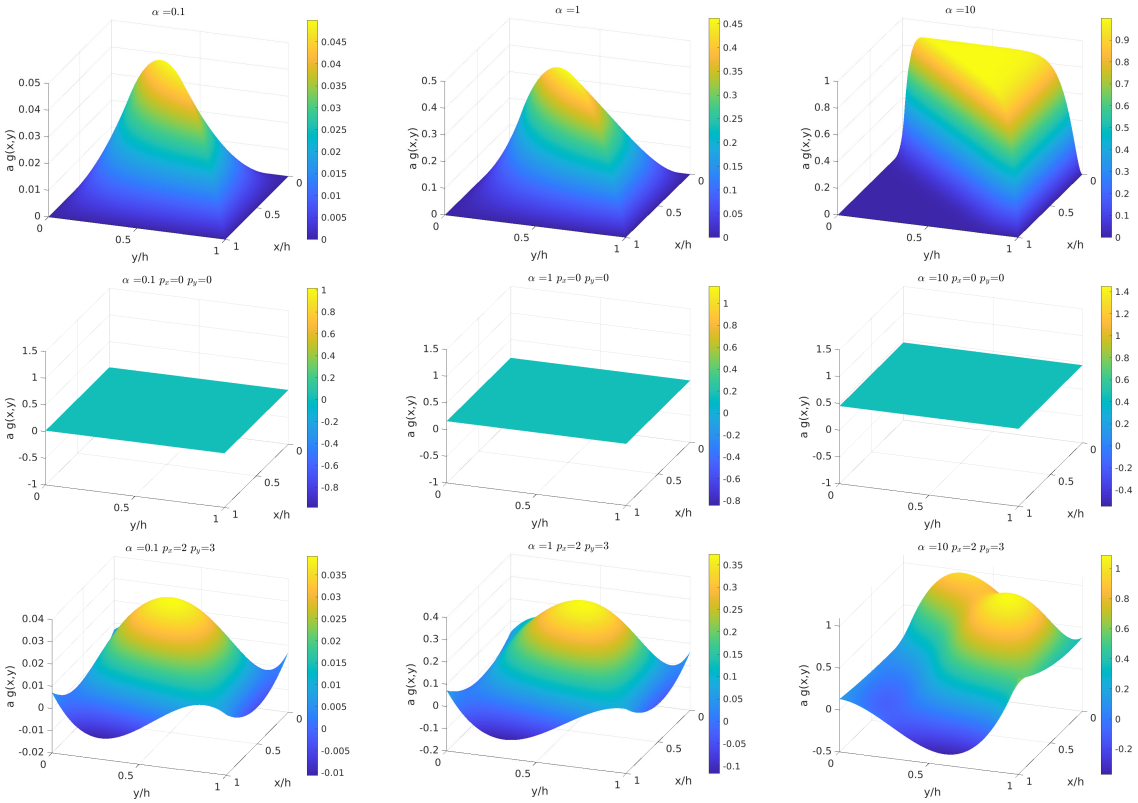


Figure 4.1: L_2 optimal approximation of the fine-scale Green's function on various tensor-product polynomial basis function g' for different cell Peclet number α.

however, can be non-linear. These insights will be used later in sections 4.3 and 4.4.

4.3 VMS-inspired feature selection

In this section, we will derive an appropriate feature set and the network architecture for our model. To demonstrate the action of the super-resolution operator, we assume the coarse space to be composed of piece-wise linear polynomials and the governing PDE to be the linear convection-diffusion equation given by the following differential operator and boundary condition

$$\mathcal{L} \triangleq a\frac{d}{dx} - \kappa\frac{d^2}{dx^2} \quad in \quad \Omega = [0, L], \;\; u = 0 \quad on \quad \Gamma = \{0, L\}. \tag{4.22}$$

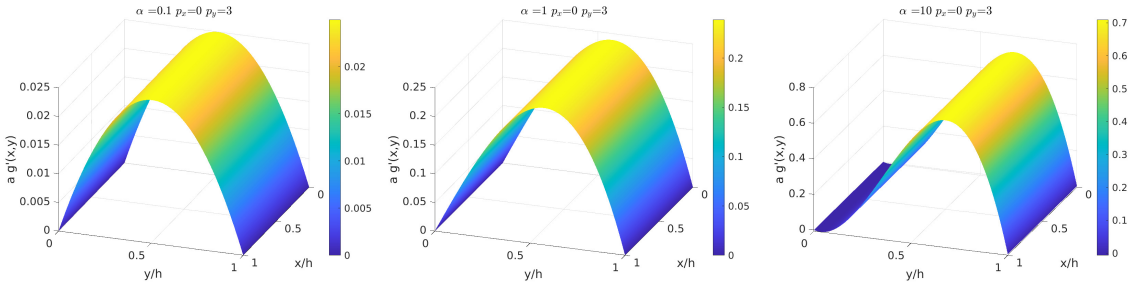


Figure 4.2: L_2-optimal approximation of the fine-scale Green's function on $p_x = 0$ and $p_y = 3$ basis such that $u'_f(y)$ is zero on the boundaries.

For a linear 1-D element of size h existing between $0 \leq x \leq h$ in its local co-ordinates, the coarse solution u_h in terms of the end-point values $u(0)$ and $u(h)$ is given by:

$$u_h(x) = u(0)(1 - x/h) + u(h)(x/h). \tag{4.23}$$

The space in which the fine scales are approximated can be a discontinuous finite element space or a bubble function space. The L_2-optimal approximation of the fine scales u' on any polynomial space existing inside an element is given by the projection Π_f :

$$u'_f(y) = \Pi_f u'(y) = -\int_{\Omega_e} \Pi_f(g'(x,y)) a \left(\frac{u(h) - u(0)}{h}\right) d\Omega_x. \tag{4.24}$$

The coarse solution considered here is an endpoint interpolant of the true solution. In that case, one can determine the exact form of the sub-scale inside the element by assuming the endpoint values as Dirichlet boundary conditions and by solving the equation inside the element:

$$u'(y) = (u(h) - u(0)) \left(\frac{1 - e^{2\alpha \frac{y}{h}}}{1 - e^{2\alpha}} - \frac{y}{h}\right). \tag{4.25}$$

The simplest approximation of the bubble function is obtained by projecting it on a p_0 discontinuous space inside the element i.e.

$$u'_0(y) = \int_0^h (u(h) - u(0)) \left(\frac{1 - e^{2\alpha \frac{y}{h}}}{1 - e^{2\alpha}} - \frac{y}{h} \right) dy = -a \frac{(u(h) - u(0))}{h} \frac{h}{2a} \left(coth(\alpha) - \frac{1}{\alpha} \right), \tag{4.26}$$

where the first part is the residual when linear basis functions are used, and the second part is the form of the stabilization parameter τ commonly used in stabilized methods. For the linear CG finite element method, Equation (4.26) represents the closure for obtaining the nodally exact solution*Hughes et al.* (1998a). Equation (4.26) can also be obtained by first projecting the elements Green's function on $p = 0$ discontinuous space i.e.

$$g'_{f,0}(x, y) = \frac{1}{h^2} \int_{\Omega_e} \int_{\Omega_e} g'(x, y) d\Omega_x d\Omega_y = \frac{1}{2a} \left(coth(\alpha) - \frac{1}{\alpha} \right), \tag{4.27}$$

and using this result in equation (4.24) to evaluate the sub-scale as follows:

$$\begin{aligned} u'_0(y) = \Pi_{f,0} u'(y) = -\int_{\Omega_e} \frac{1}{2a} \left(coth(\alpha) - \frac{1}{\alpha} \right) a \frac{(u(h) - u(0))}{h} d\Omega_x = \\ -a \frac{(u(h) - u(0))}{h} \frac{h}{2a} \left(coth(\alpha) - \frac{1}{\alpha} \right). \end{aligned} \tag{4.28}$$

Next, we define the mean and r.m.s quantities of the coarse solution in an element as follows:

$$u_m = \frac{\int_0^h u_h d\Omega}{h} \;\; ; \;\; u_{rms} = \sqrt{\frac{\int_0^h (u_h - u_m)^2 d\Omega}{h}}. \tag{4.29}$$

An important observation is that the solution is independent of the mean u_m:

$$u'(y) = ((u(h) - u_m) - (u(0) - u_m)) \left(\frac{1 - e^{2\alpha \frac{y}{h}}}{1 - e^{2\alpha}} - \frac{y}{h} \right). \tag{4.30}$$

If our approximation space is linear, then u_m and u_{rms} are given by:

$$u_{m,1} \triangleq (u(0) + u(h))/2 \quad u_{rms,1} \triangleq \frac{|u(h) - u(0)|}{\sqrt{12}}. \tag{4.31}$$

Re-arranging this form we get:

$$\frac{u_0'(y)}{u_{rms,1}} = \sqrt{3}(\frac{1}{\alpha} - coth(\alpha))sgn(u(h) - u(0)), \tag{4.32}$$

where sgn denotes the sign function. The above equation can be simplified as follows:

$$\frac{u_0'(y)}{u_{rms,1}} = \begin{cases} \sqrt{3}(\frac{1}{\alpha} - coth(\alpha)), & \frac{u(h)-u(0)}{|u(h)-u(0)|} \geq 0 \\ -\sqrt{3}(\frac{1}{\alpha} - coth(\alpha)). & \frac{u(h)-u(0)}{|u(h)-u(0)|} \leq 0 \end{cases} \tag{4.33}$$

If we compute the the mean-subtracted basis-coefficients of the coarse solution u_h and re-scale them with the r.m.s $u_{rms,1}$ we get:

$$\frac{u(0) - u_m}{u_{rms,1}} = \sqrt{3}\frac{u(0) - u(h)}{|u(h) - u(0)|}, \;\; ; \;\; \frac{u(h) - u_m}{u_{rms,1}} = \sqrt{3}\frac{u(h) - u(0)}{|u(h) - u(0)|}. \tag{4.34}$$

These parameters determine the sign of the sub-scale in equation (4.33). Hence, the magnitude of the sub-scales is fully determined by the physics-informed parameter α and its sign (phase) is determined by these parameters. Consequently, an appropriate choice of the feature will be:

$$\frac{u_0'(y)}{u_{rms,1}} = f\left(\alpha, \frac{u(0) - u_m}{u_{rms}}, \frac{u(h) - u_m}{u_{rms}}\right). \tag{4.35}$$

The two extra parameters, in this case, are redundant because they are the same in magnitude and opposite in sign. Hence, only one parameter can be used:

$$\frac{u_0'(y)}{u_{rms,1}} = f\left(\alpha, \frac{u(h) - u(0)}{u_{rms}}\right). \tag{4.36}$$

Although equations (4.26) and (4.36) are identical, the generalisibilty of a neural-network model is considerably affected by the choice of the feature set and the normalization process. For example, the model form

$$u'_0(y) = g\left(a, \kappa, h, u(h), u(0)\right), \tag{4.37}$$

does not utilize the idea that only specific combinations of a, κ, h i.e. cell Peclet number α effect the sub-scale solution distinctively. To train this model, a big data-set with a large stretch in the values of the input parameters is required. This equation is also not invariant to the units or the scaling used for the input parameters. The proposed model form in equation (4.36) tries to address some of these challenges. An important point to note is that the functional form presented in equation (4.36) is derived for the advection-diffusion problem. However, this structure will serve as the inspiration for applying this method to other non-linear problems, as detailed in the following section.

4.4 Learning VMS-consistent subscales

In this section, we will attempt to derive a general model structure that can be used to learn sub-scales arising in a wide range of multi-scale problems. In case of super-resolution, the sub-scales contain the fine-scale information that is absent in the lower-resolution image. When used as the closure, they are responsible for modeling the effect of unresolved fine-scales on the coarse-scales. Irrespective of the mode of application, the model structure should not be different. As a first step, we learn the mapping presented in equation (4.36) and compare it to the analytical solution. Data is first obtained by solving the equation at different Peclet numbers Pe on very fine meshes (we refer to this as the DNS) for training the network. Similarly, we can also generate data by dividing a single high Pe DNS case into multiple cases with different element sizes, i.e., different cell Peclet numbers. The coarse-scale is obtained in the form of end-point interpolant, and the approximation to the

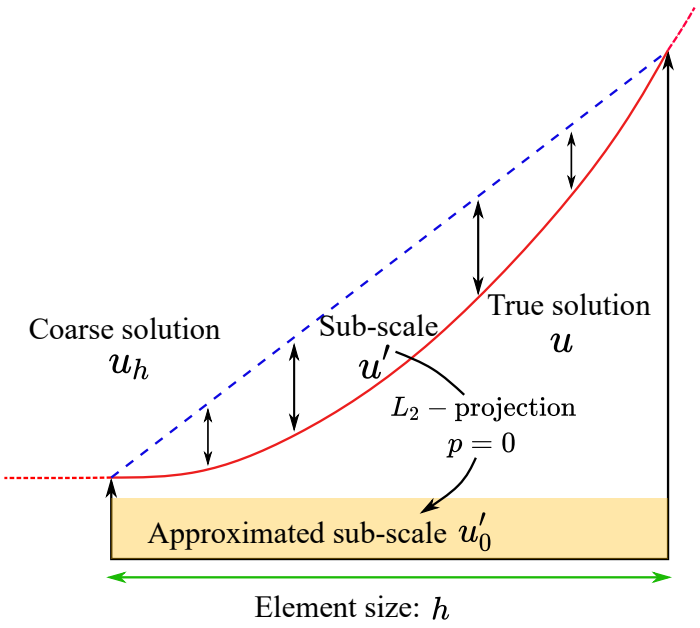


Figure 4.3: Computation of u'_0 by element-wise L_2-projection of the sub-scale u' on the $p = 0$ polynomial space.

sub-scale is then computed for each such element numerically as shown in figure 4.3. A small network with a size of $3 \times 8 \times 8 \times 8 \times 1$ is then trained using this data. Figure 4.4 shows the comparison between the sub-scale obtained analytically vs. that learned purely from data using neural network (N-N). It can be observed that the small network could learn the analytical solution accurately. The discussion in this section was mainly focused on learning the sub-scales. In section 4.8, we demonstrate how these sub-scales can be used as closures for the CG finite element method and further extend them to high order discretizations.

One way of obtaining the exact form of the sub-scale is by keeping our approximating space of the coarse-scale the same (linear) and increasing the order of the discontinuous space p in which u' is approximated. In limit, $p \to \infty$ the approximate sub-scale should approach $u'(y)$. However, the order of the polynomial p required to learn the function increases when Pe is increased. By limiting the output order p to the order of super-resolution, we are reducing the complexity and size of the network, as discussed in the last section. This is because, when $\alpha \to \infty$, $f(\alpha) = \frac{|u'|}{u_{rms}} = \sqrt{3}(coth(\alpha) - \frac{1}{\alpha})$ is well-behaved,

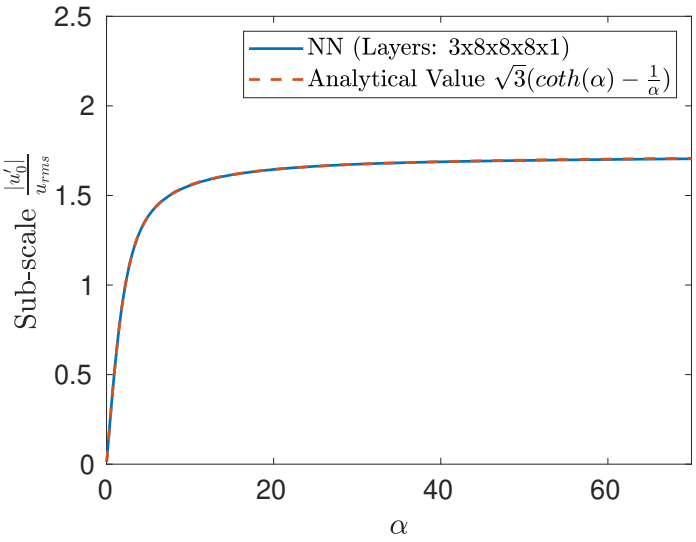


Figure 4.4: Comparison of sub-scales magnitude as a function of cell Peclet number α obtained analytically vs. that learnt from data using a N-N.

whereas if $u'(y)$ given by the equation (4.25) is learnt directly as a function of y and α, the function becomes steep at $y = h$ for $\alpha \to \infty$. A solution to this problem is to use features such as $\frac{1-e^{2\alpha\frac{y}{h}}}{1-e^{2\alpha}}$ as inputs. However, this restricts the method to one kind of problem. Similarly, the optimal form of the sub-scale on discontinuous $p = 1$ basis functions is given by:

$$\frac{u_1'(y)}{u_{rms,1}} = C_1'(\alpha)\psi_1(y/h) + C_1'(\alpha)\psi_2(y/h). \tag{4.38}$$

The form of the function to be learned for this case is:

$$\left[C_1', C_2'\right] = \mathbf{f}(\alpha). \tag{4.39}$$

The above analysis was performed for the continuous Galerkin (CG) method, but these ideas can be extended to the discontinuous Galerkin (DG). Retaining the same structure, we extend the technique to non-linear/linear problems for both CG/DG types of basis functions

as follows:

$$[C'_{1,p_s}, C'_{2,p_s}, .., C'_{(p_s+1)^d,p_s}] = \mathbf{f}\Big(\alpha, \left[\tilde{C}_{1,p_c}, \tilde{C}_{2,p_c}, .., \tilde{C}_{(p_c+1)^d,p_c}\right], .. \\ , \left[\tilde{C}_{1,p_c}, \tilde{C}_{2,p_c}, .., \tilde{C}_{(p_c+1)^d,p_c}\right]_m, ..\Big), \tag{4.40}$$

where p_s and p_c are the polynomial orders of the spaces in which the sub-scale and coarse-scale are optimally represented, and d denotes the dimension of the problem. This function, apart from α (equivalent to cell Re/ cell Pe) also contains the basis coefficients of the current element and its neighbors. The term $\left[\tilde{C}_{1,p_c}, \tilde{C}_{2,p_c}, .., \tilde{C}_{(p_c+1)^d,p_c}\right]_m$ with subscript m denotes the mean subtracted normalized basis coefficient of m^{th} neighbour. The neighbour information is critical when discontinuous basis is used, or when bubble function approximation are not employed in CG, or non-local transfer of information happens from outside the element. These coefficients are first subtracted with the coarse scale mean u_m and then normalised with the coarse scale R.M.S. u_{rms} as done previously. The output of the function $\left[C'_{1,p_s}, C'_{2,p_s}, .., C'_{(p_s+1)^d,p_s}\right]$ denotes the basis coefficients of the sub-scale that has been normalised with u_{rms} only i.e.

$$\left[\tilde{C}_{1,p_c}, \tilde{C}_{2,p_c}, .., \tilde{C}_{(p_c+1)^d,p_c}\right]_m = \left[C_{1,p_c} - u_m, C_{2,p_c} - u_m, .., C_{(p_c+1)^d,p_c} - u_m\right]_m / u_{rms}, \tag{4.41}$$

where $C_{i,j}$ denotes the actual basis coefficients. The quantities used for shifting and non-dimensionalizing the input parameters,i.e., u_m and u_{rms} respectively, and non-dimensionalizing the output parameters u_{rms} are calculated using the coarse-scale solution in the center element only. As will be seen later in this chapter, the non-dimensionalization process is critical for the N-N model to generalize. The output coefficients are finally re-scaled with

u_{rms} and added to the coarse-scale to obtain the super-resolved solution as follows:

$$u_{sr} = u_{p_c} + u'_{p_s} = u_{p_c} + u_{rms} \sum_{i}^{(p_s+1)^d} C'_{i,p_s} \psi_{i,p_s}, \tag{4.42}$$

where, ψ_{i,p_s} denotes basis function corresponding to the i^{th} node or mode. Division by u_{rms} in equation (4.41) is a problem when u_{rms} is precisely equal to zero. However, adding a small positive number ϵ to u_{rms} while dividing was sufficient for all the cases presented below.

4.4.1 The Variational Super-resolution Network architecture

In addition to the model features, the model architecture can be made consistent with the VMS formulation. As proposed in equation (4.40), the input to the model are the physics-informed parameters such as the cell Péclet number α, along with the normalized mean-subtracted coarse-scale basis coefficients of an element and its neighbor. The output to the network are the normalized sub-scale basis coefficients in that particular element. The physics-informed parameter can also be other non-dimensional numbers such as the CFL number or the cell Reynolds number Re_Δ specific to the problem. Given these sets of input and output features, many possible ways of embedding them into the model exist. Figure 4.5 shows two different kinds of network architectures to achieve this.

The traditional approach is based on the idea of training one single fully connected N-N with both the normalized coarse-scale basis coefficients and the physics informed-parameter as inputs. If the normalized sub-scale basis coefficients are denoted by $\mathbf{u}'$, the normalized input coarse basis coefficients of the element and its neighbors as $\mathbf{u}_c$ and the physics-informed parameter α, then the traditional model is given by:

$$\mathbf{u}' = \mathbf{f}^{FNN}\left(\alpha, \mathbf{u}_c\right), \tag{4.43}$$

where $\mathbf{f}^{FNN}$ denotes a fully-connected neural network (FNN). Another approach also called

the variational super-resolution N-N (VSRNN), is based on a multiplicative strategy in which the fine-scales are approximated by a sum of products of individual functions of the parameters and the coarse-scales. The model form can be summarized as follows:

$$\mathbf{u}' = \mathbf{f}^{FNN}(\mathbf{g}_\alpha \odot \mathbf{g}_u), \;\; ; \;\; \mathbf{g}_\alpha = \mathbf{h}^{FNN}(\alpha), \;\; ; \;\; \mathbf{g}_u = \mathbf{k}^{FNN}(\mathbf{u}_c), \tag{4.44}$$

where $\mathbf{f}^{FNN}$, $\mathbf{h}^{FNN}$ and $\mathbf{k}^{FNN}$ denote three different FNNs. The symbol $\odot$ denotes element-wise multiplication between two vectors of the same size. This architecture is inspired by equation (4.21) and the analytical solution of the sub-scale provided in equations (4.32). In this case, $\mathbf{g}_\alpha$ (Part B) learns the dependence of α i.e. $\sqrt{3}\left(\frac{1}{\alpha} - coth(\alpha)\right)$ and $\mathbf{g}_u$ (Part A) learns the dependence of the normalized coarse-scale basis coefficients, i.e., $sgn(u(h) - u(0))$. In sections 4.2-4.4 and section 4.8, we tried to develop insights into the working of the VSRNN and demonstrated its application as a closures for the CG method. In sections 4.5-4.7, we use the VSRNN to perform super-resolution and sub-grid modelling for the DG method. This does not imply that the exact sub-scales used for the CG method in sections 4.2-4.4 are re-used in sections 4.5-4.7. Only the model form has been assumed to be the same which is finally trained on the correct sub-scale that is consistent with the DG approach. Details about the scale-decomposition used for the DG method are detailed in section 4.5.

4.5 Data Generation

The generation of proper training and testing data is as critical as the model architecture and features. For example, low-resolution data can be obtained from a variety of high fidelity sources (simulations and experiment) and can be coarse-grained. There is no guarantee that a model trained to perform super-resolution of the filtered solution will be useful unless the filtering operation is consistent with the underlying numerics. To this end, consider the

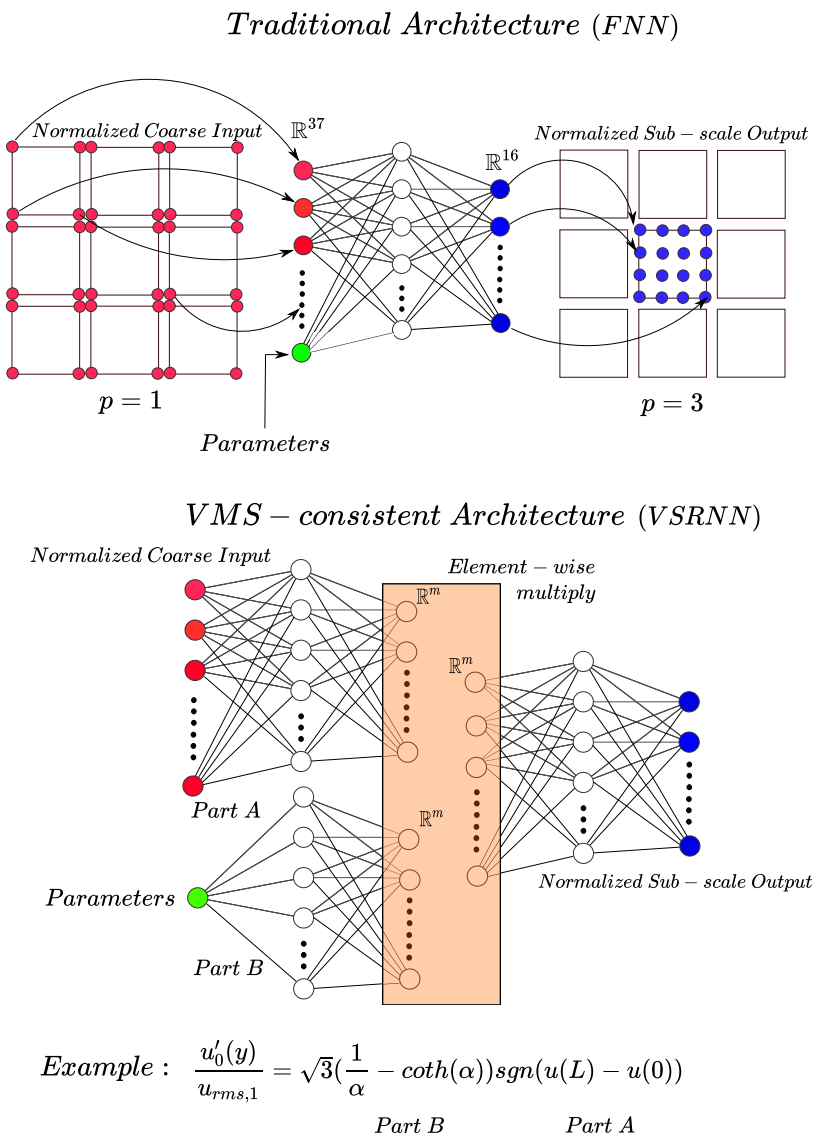


Figure 4.5: VMS-consistent architecture and features are used for learning the mapping in VSRNN.

VMS decomposition of the full-order solution u as follows:

$$u = u_h + u', \tag{4.45}$$

where $u_h \in \mathcal{V}_h$ and $u' \in \mathcal{V}'$. The vector space of functions $\mathcal{V} \equiv \mathcal{H}^1(\Omega)$ is a Sobolev space where the functions and their derivative are square-integrable. This space is now decomposed as follows:

$$\mathcal{V} = \mathcal{V}_h \oplus \mathcal{V}', \tag{4.46}$$

where $\oplus$ represents a direct sum of $\mathcal{V}_h$ and $\mathcal{V}'$. Let us also define $\mathcal{T}_h$ to be a tessellation of domain Ω into a set of non-overlapping elements, K, each having a sub-domain Ω_k and boundary Γ_k. $\mathcal{V}_h$ is now defined as:

$$\mathcal{V}_h \triangleq \left\{ u \in L_2(\Omega) : u|_{\mathcal{T}} \in P^k(T), T \in \mathcal{T}_h \right\}, \tag{4.47}$$

where the space of polynomials up to degree k is denoted as P^k. Defining $\mathcal{V}_h$ in this manner allows discontinuity in the solution across element boundaries. Given u from the high-fidelity simulation, our goal is to find the optimal representation of u in the coarse sub-space $\mathcal{V}_h$. In our case, we will use the L^2 projection to obtain u_h which minimises the value of $||u - u_h||_2^2$. This problem is equivalent to the problem of finding $u_h \in \mathcal{V}_h$ such that

$$(u, w_h) = (u_h, w_h) \quad \forall w_h \in \mathcal{V}_h. \tag{4.48}$$

For example, to generate training data for section 4.7, we use direct simulation (DNS) results for a channel flow at $Re_\tau \approx 950$ *Lozano-Durán and Jiménez* (2014). The 3-D data is sliced into many 2-D planes at different y^+ locations, and projection is performed in 2-D for simplicity. The fine space and coarse space's polynomial orders are chosen to be 3 and 1, respectively, i.e., we are super-resolving $p = 1$ results to $p = 3$ as shown in figure 4.6.

The computation of terms on the RHS of the equation (4.48) requires special care.

Although u has been assumed to exist in an infinitely high dimensional space, in reality, it is not. For example, the Kolmogorov scale η dictates the size of the smallest size eddy and the size of u. Although the dimension of u is finite, it is enormous when compared to u_h because the size of our finite element grid h is much greater than η. Hence, to accurately compute these terms, the DNS data is first interpolated using cubic-splines to a much finer-grid $O(\eta)$ and then the inner products with the coarse finite element basis function (having dimension $O(h)$) are computed on these fine-meshes using the Simpson's Rule. Interpolation of DNS was done to ensure that the projected solution u_h did not change significantly due to the numerical integration scheme. Sample L_2-projected snapshots of the DNS data on elements of different sizes and orders are shown in figure 4.6.

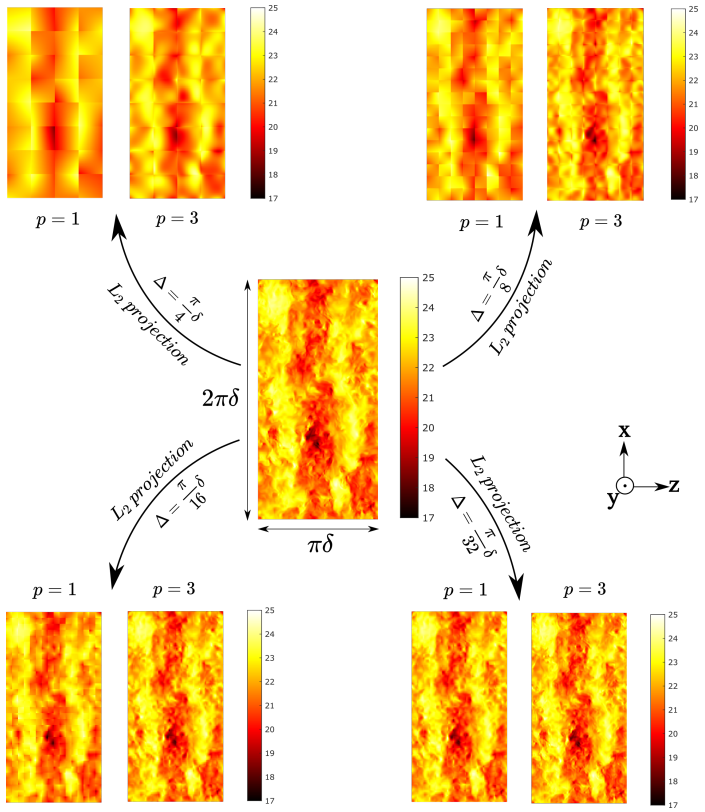


Figure 4.6: Example L_2-projected snapshots of the DNS data for channel flow at $Re_\tau \approx 950$ and wall normal distance of $y^+ \approx 910$.

N_x	N_t	Δx	Δt	CFL	$\|u_2 - u_1\|_2$	$\|u_{2,NN} - u_1\|_2$	$\|u_{2,NN} - u_2\|_2$	$\frac{\|u_2-u_1\|_2}{\|u_2\|_2}$	$\frac{\|u_{2,NN}-u_1\|_2}{\|u_2\|_2}$	$\frac{\|u_{2,NN}-u_2\|_2}{\|u_2\|_2}$
16	32	0.3927	0.19635	0.5	0.41684	0.41684	0.0018975	0.066344	0.066345	0.00030201
16	16	0.3927	0.3927	1	0.56984	0.57053	0.0044046	0.090699	0.090809	0.00070106
16	8	0.3927	0.7854	2	1.4709	1.4681	0.0066213	0.23456	0.23411	0.0010559
16	4	0.3927	1.5708	4	3.3222	3.3223	0.016548	0.56392	0.56394	0.0028089
16	2	0.3927	3.1416	8	3.115	2.9975	1.7061	0.75135	0.723	0.41152
32	64	0.19635	0.098175	0.5	0.1075	0.10755	0.00075342	0.01711	0.017117	0.00011991
32	32	0.19635	0.19635	1	0.14736	0.14738	0.0020468	0.023453	0.023457	0.00032576
32	16	0.19635	0.3927	2	0.41684	0.4157	0.005024	0.066344	0.066163	0.00079962
32	8	0.19635	0.7854	4	1.4235	1.4245	0.009266	0.22699	0.22716	0.0014776
32	4	0.19635	1.5708	8	3.3146	2.723	1.3326	0.5626	0.46219	0.2262

Table 4.1: Reconstruction error when super-resolved from $p = 1$ to $p = 2$ for the linear advection problem with an unseen initial condition at different CFL numbers.

4.6 Application to Linear Advection

In this section, we apply our super-resolution methodology to the linear advection equation in the domain $\Omega \subset \mathbb{R}$ with the boundary $\Gamma = \partial\Omega$ as follows

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{4.49}$$

with time $t \in (0, T]$, and spatially periodic boundary conditions on Γ. The parameter a denotes the advection velocity. The required training data is generated by L_2-projecting the true solution on coarse and fine spaces. Unlike traditional methods, in which only the spatial term in the PDE is discretized using finite elements, we will consider 2-D space-time finite elements spanned by $p = 1, 2$ degree tensor-product Lagrange basis functions in space and time. The goal is to investigate the application of our super-resolution method in two different settings. First, as a model to super-resolve coarse low-order finite element data to high-order finer finite element data. Second, as a method to improve the existing finite element method for this problem in a predictive setting on a problem with a very different initial condition in comparison to the training data.

4.6.1 Super-resolution

To train the super-resolution model, we will generate the true solution on a fine grid. The initial condition for this case is $sin(x) + sin(2x) + sin(4x)$, and the size of the grid

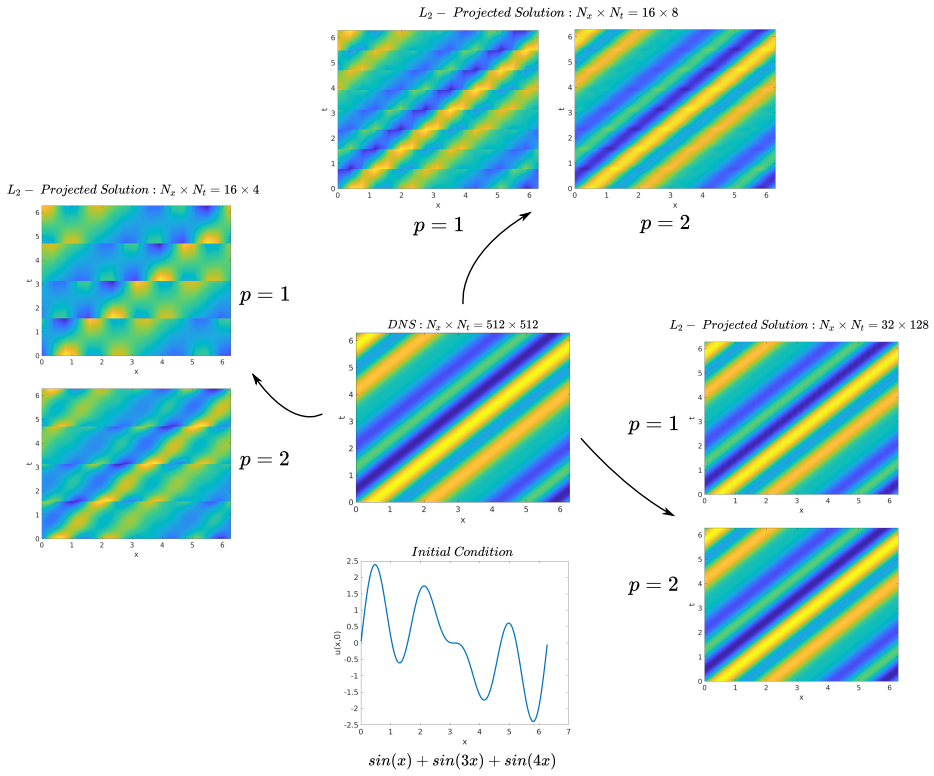


Figure 4.7: High-resolution linear advection solution on a 512x512 mesh is L_2-projected on different finite element meshes for $p = 1, 2$ to generate data for training the model.

is taken to be $N_x \times N_t : 512 \times 512$. This high-resolution mesh is to ensure that the true solution remains highly resolved on this grid. For simplicity, periodic boundary conditions are also applied in the x direction. The true solution is then evaluated on all the grid points to obtain the DNS solution. The next step is to obtain the coarse $p = 1$ and fine $p = 2$ L_2-projected solution for different meshes having spatial and temporal element sizes Δx and Δt respectively as shown in figure 4.7. A non-dimensional parameter naturally arising in this case is the $CFL = \frac{a\Delta t}{\Delta x}$ number which is similar to Peclet number in the 1-D convection-diffusion problem. This solution is then projected on grids of various sizes corresponding to CFL numbers of 0.25,0.5,1.0,2 and 4. These CFL numbers correspond to three sets of grid-sizes: (i.) $N_x \times N_t$: 32×128, 32×64, 32×32, 32×16 and 32×8; (ii.) $N_x \times N_t$: 16×64, 16×32, 16×16, 16×8 and 16×4; and (iii.) $N_x \times N_t$: 24×96, 24×48, 24×24, 24×12 and 24×6. For each element in these grids, the basis coefficients corresponding to the coarse-space is extracted along with its neighbors, excluding those which are part of the future time step,

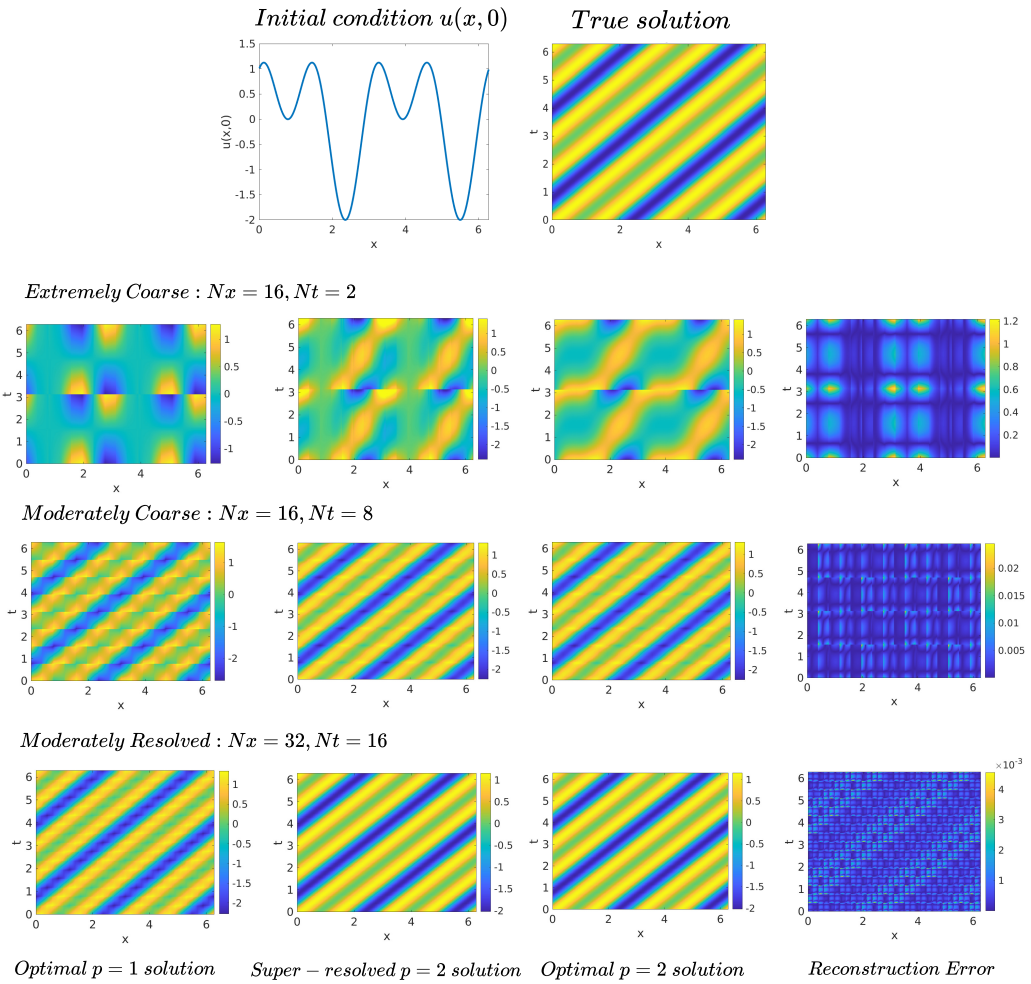


Figure 4.8: Comparison of the true L_2-projected $p = 2$ solution and that obtained by super-resolution of L_2-projected $p = 1$ solution on the space-time plane.

i.e., only space-time elements present in the south, south-east, south-west, east, and west of the central coarse element are extracted. The corresponding fine-space basis coefficients are also extracted for the central element. As a first step towards normalization, a mean value u_m is first computed inside the element as follows:

$$u_m = \frac{\iint_{\Omega_e} u_1(x,t)\, dx\, dt}{\iint_{\Omega_e} dx\, dt}. \tag{4.50}$$

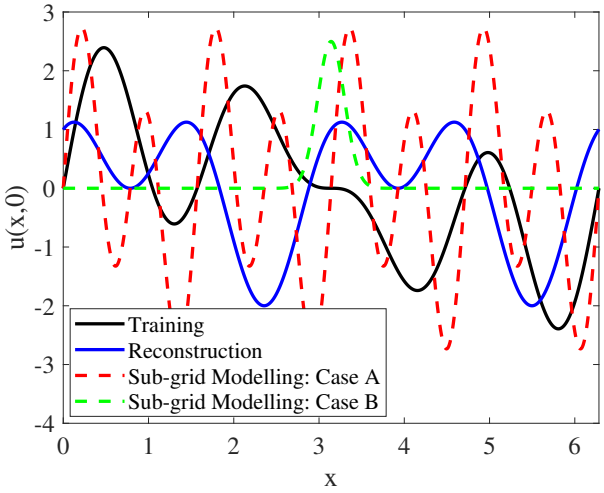


Figure 4.9: Different initial conditions are used for training, offline reconstruction and online evaluation of the model.

N_x	N_t	Δx	Δt	CFL	$\|u_3 - u_1\|_2$	$\|u_{3,NN} - u_1\|_2$	$\|u_{3,NN} - u_3\|_2$	$\frac{\|u_3-u_1\|_2}{\|u_3\|_2}$	$\frac{\|u_{3,NN}-u_1\|_2}{\|u_3\|_2}$	$\frac{\|u_{3,NN}-u_3\|_2}{\|u_3\|_2}$
16	64	0.3927	0.098175	0.25	0.40799	0.40956	0.029322	0.064934	0.065183	0.0046667
16	32	0.3927	0.19635	0.5	0.42022	0.42039	0.004307	0.066879	0.066907	0.00068548
16	16	0.3927	0.3927	1	0.57471	0.57558	0.0043912	0.091468	0.091606	0.00069888
16	8	0.3927	0.7854	2	1.5204	1.5228	0.0087391	0.242	0.24238	0.001391
16	4	0.3927	1.5708	4	3.8684	3.8747	0.018305	0.62236	0.62338	0.0029449
16	2	0.3927	3.1416	8	3.8513	3.6725	3.079	0.81524	0.77739	0.65176
32	128	0.19635	0.049087	0.25	0.108	0.10626	0.0096799	0.017188	0.016912	0.0015406
32	64	0.19635	0.098175	0.5	0.10772	0.10788	0.0011823	0.017145	0.017169	0.00018816
32	32	0.19635	0.19635	1	0.14764	0.14791	0.0015759	0.023498	0.02354	0.00025081
32	16	0.19635	0.3927	2	0.42022	0.42123	0.0037693	0.066879	0.06704	0.0005999
32	8	0.19635	0.7854	4	1.4737	1.4799	0.011139	0.23456	0.23554	0.0017729
32	4	0.19635	1.5708	8	3.8616	3.0788	1.266	0.62126	0.49532	0.20368

Table 4.2: Reconstruction error when super-resolved from $p = 1$ to $p = 3$ for the linear advection problem with an unseen initial condition at different CFL numbers.

Similarly, an R.M.S value is also computed inside the element:

$$u_{rms} = \sqrt{\frac{\iint_{\Omega_e} (u_1(x,t) - u_m)^2 \, dx \, dt}{\iint_{\Omega_e} dx \, dt}}. \tag{4.51}$$

A model is then sought in the following form

$$\left[C'_{1,p_s}, C'_{2,p_s}, .., C'_{p_s+1,p_s}\right] = \mathbf{f}\Big(log(CFL), \left[\tilde{C}_{1,p_c}, \tilde{C}_{2,p_c}, .., \tilde{C}_{p_c+1,p_c}\right], .. \\ , \left[\tilde{C}_{1,p_c}, \tilde{C}_{2,p_c}, .., \tilde{C}_{p_c+1,p_c}\right]_m, ..\Big), \tag{4.52}$$

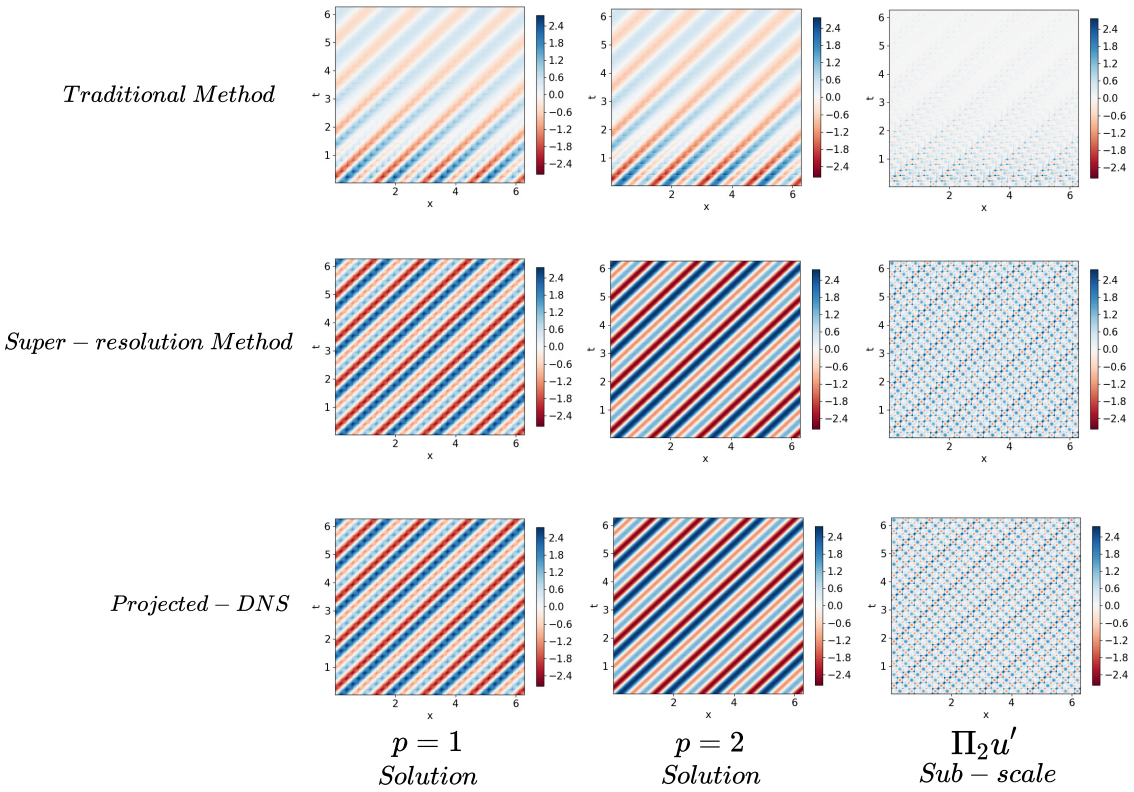


Figure 4.10: Comparison of solution obtained using the traditional space-time method and the super-resolved method to the projected DNS solution for initial condition $u(x,0) = sin(4x) + 2sin(8x)$.

where $C'_{i,j}$ and $\tilde{C}_{i,j}$ are defined in terms of u_m and u_{rms} similar to equation (4.41) . For learning this function, the VSRNN architecture is adopted with size 24×12 for part A, 1×6×12 for part B, and 12×9 for the post-multiplication part, respectively. The network is first trained on data obtained by projecting a highly resolved DNS solution, as shown in figure 4.7.

In the next step, the super-resolution model is evaluated on an unseen coarse solution obtained by projecting the DNS solution for a different set of initial conditions, as shown in figure 4.8. It can be observed that unless an extremely coarse model was used, the model could efficiently super-resolve unseen coarse solution to its fine-solution with minimal reconstruction error. This reconstruction error for the cases is reported in table 4.1. Except for the case when the CFL number was as high as 8, the error in reconstruction $\|u_{2,NN} - u_2\|_2$

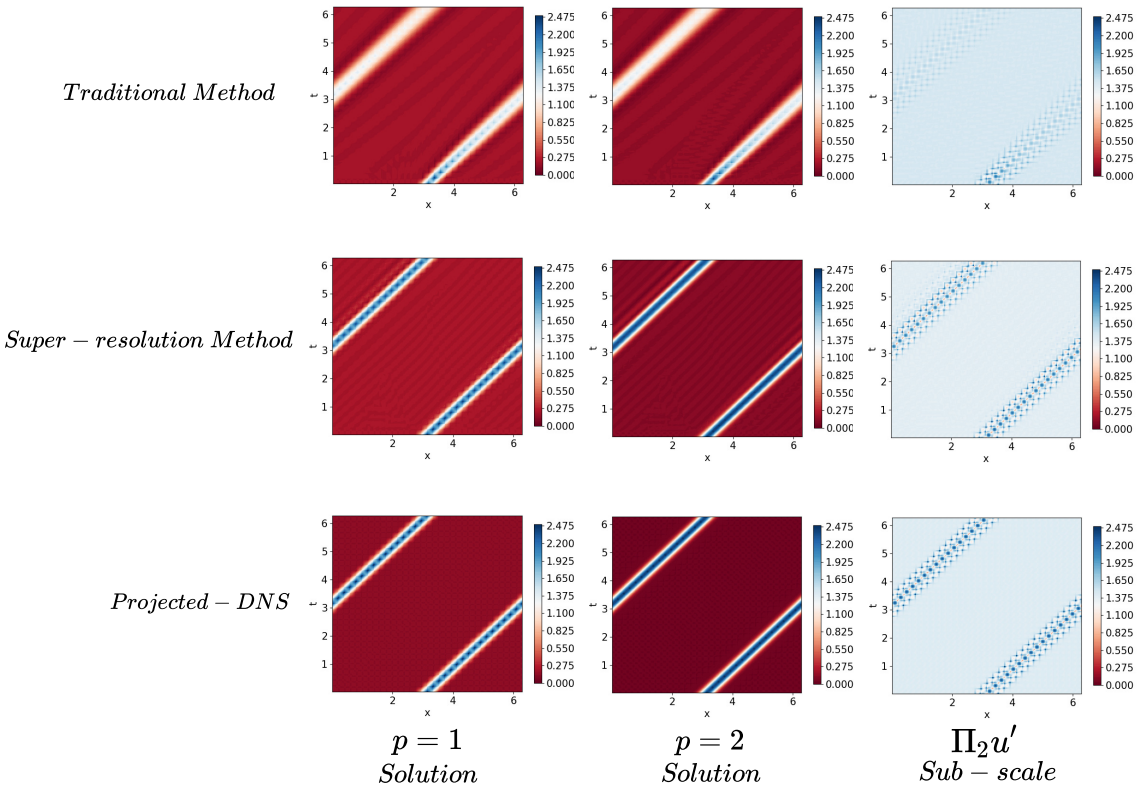


Figure 4.11: Comparison of solution obtained using the traditional space-time method and the super-resolved method to the projected DNS solution for initial condition $u(x,0) = 2.5e^{-20(x-\pi)^2}$.

is orders of magnitude smaller in comparison to the magnitude of the sub-scale $\|u_2 - u_1\|_2$. Hence, the super-resolution model is very efficient in reconstructing the fine-solution as long as the CFL is not large. A separate model for super-resolving $p = 1$ solution to $p = 3$ solution is also trained by repeating the steps above. As reported in table 4.2, a similar trend is again observed for this case where efficient reconstruction was obtained at lower CFL values. Irrespective of the large reconstruction error at CFL values of 8 and above, the magnitude of $\|u_3 - u_1\|_2$ is still very close to $\|u_{3,NN} - u_1\|_2$ for all CFL numbers. This indicates that the super-resolution model can also act as an efficient error indicator that can be used for mesh adaption.

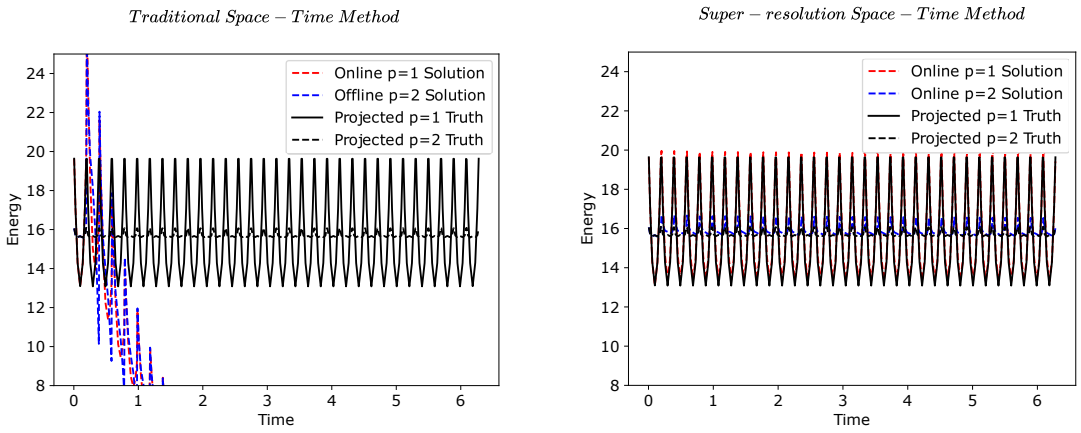


Figure 4.12: Evolution of energy $E(t) = \int u(x,t)^2 d\Omega_x$ as a function of time for traditional space-time method vs. super-resolution method for initial condition $u(x,0) = sin(4x) + 2sin(8x)$.

4.6.2 Sub-grid Modelling

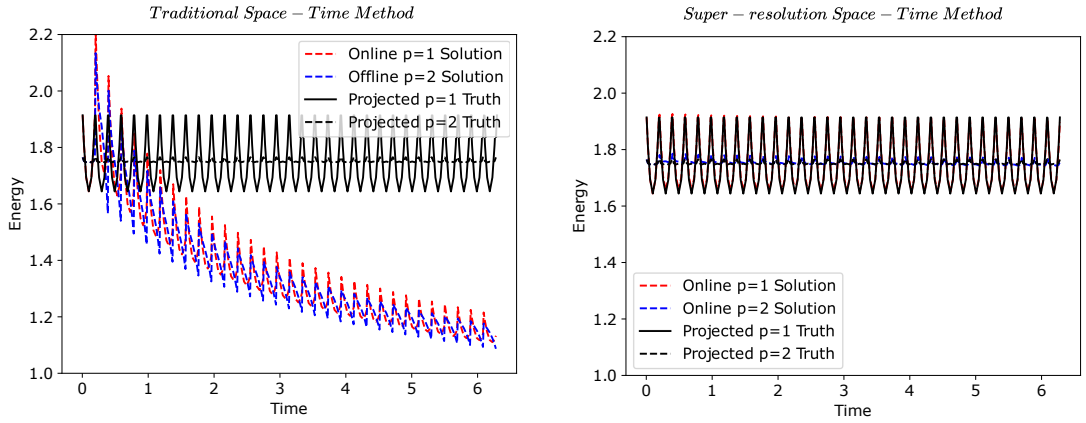
In the previous section, we showed that the neural network could efficiently predict the sub-scales as long as the grid is not highly under-resolved. In this section, we will use the trained model from the previous section to improve the existing space-time based numerical method. To this end, we start with the linear advection equation in the domain $\Omega \subset \mathbb{R}$ with the boundary $\Gamma = \partial\Omega$ as follows

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{4.53}$$

with a periodic boundary condition at the boundary Γ and time $t \in (0, T]$. The weak form of the above equation is obtained by multiplying it with a test function w and integrating it over the space-time element as follows

$$\int_{\Omega_e} \left(\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} \right) w d\Omega = 0, \tag{4.54}$$

such that $u \in \mathcal{V}$ for all $w \in \mathcal{V}$. Let us also define $\mathcal{T}_h$ as a tessellation of the domain Ω into a set of non-overlapping elements, K, each having a sub-domain Ω_e and boundary Γ_e. The



(a) (b)

Figure 4.13: Evolution of energy $E(t) = \int u(x,t)^2 d\Omega_x$ as a function of time for traditional space-time method vs. super-resolution method for initial condition $u(x,0) = 2.5e^{-20(x-\pi)^2}$.

vector space of functions $\mathcal{V} \equiv \mathcal{H}^1(\mathcal{T}_h)$ is a Sobolev space where the functions and their derivatives are square-integrable *inside each element*. Simplifying the previous equation, we obtain the following:

$$\int_{\Omega_e} \left(\frac{\partial uw}{\partial t} + \frac{\partial auw}{\partial x} \right) d\Omega - \int_{\Omega_e} \left(u \frac{\partial w}{\partial t} + au \frac{\partial w}{\partial x} \right) d\Omega = 0. \tag{4.55}$$

Application of the divergence theorem leads to the following

$$\int_{\Gamma_e} (auw\hat{\mathbf{i}} + uw\hat{\mathbf{j}}).(n_x\hat{\mathbf{i}} + n_t\hat{\mathbf{j}}) d\Gamma - \int_{\Omega_e} \left(u \frac{\partial w}{\partial t} + au \frac{\partial w}{\partial x} \right) d\Omega = 0, \tag{4.56}$$

where n_x and n_t denote the components of the outward normal on the surface of the element along the space and time axis, respectively. The first term in the DG method is replaced with a numerical flux as follows:

$$\int_{\Gamma_e} (F_x^*(au, au^-)\hat{\mathbf{i}} + F_t^*(u, u^-)\hat{\mathbf{j}}).(n_x\hat{\mathbf{i}} + n_t\hat{\mathbf{j}}) w d\Gamma - \int_{\Omega_e} \left(u \frac{\partial w}{\partial t} + au \frac{\partial w}{\partial x} \right) d\Omega = 0. \tag{4.57}$$

The traditional space-time DG method can be obtained by applying the Galerkin approximation to the previous equation as follows:

$$\int_{\Gamma_e} (F_x^*\hat{\mathbf{i}} + F_t^*\hat{\mathbf{j}}).(n_x\hat{\mathbf{i}} + n_t\hat{\mathbf{j}})w_h d\Gamma - \int_{\Omega_e} \left(u_h\frac{\partial w_h}{\partial t} + au_h\frac{\partial u_h}{\partial x}\right) d\Omega = 0, \tag{4.58}$$

where $F_x^* = au_h{}^-$ when $an_x < 0$ and au_h when $an_x > 0$. The temporal flux on the bottom face is based on previous space-time slab i.e. $F_t^* = u_h{}^-$. The effect of the numerical fluxes is similar to that of a closure, which is dissipative in action due to the jump term, ensuring the stability of the method. The numerical fluxes were originally developed for solving the exact 1-D problem at the interface, and application to the DG method is generally made by applying it along the normal direction of the interface. However, this might not be the most optimal choice for the flux. To this end, we revisit the strong form of the DG i.e., equation (4.57) through the VMS approach. The coarse-scale equation corresponding to equation (4.57) is given by:

$$\begin{aligned}\int_{\Gamma_e} \left(F_x^*(a(u_h+u'), a(u_h^- + u'^-))\hat{\mathbf{i}} + F_t^*(u_h+u', u_h^- + u'^-)\hat{\mathbf{j}}\right).(n_x\hat{\mathbf{i}} + n_t\hat{\mathbf{j}})w_h d\Gamma - \\ \int_{\Omega_e} \left((u_h+u')\frac{\partial w_h}{\partial t} + a(u_h+u')\frac{\partial w_h}{\partial x}\right) d\Omega = 0.\end{aligned} \tag{4.59}$$

If $\mathcal{V}_h \perp \mathcal{V}'$, then the effect of the sub-scale on the interior flux is negligible i.e.

$$\begin{aligned}\int_{\Gamma_e} \left(F_x^*(a(u_h+u'), a(u_h^- + u'^-))\hat{\mathbf{i}} + F_t^*(u_h+u', u_h^- + u'^-)\hat{\mathbf{j}}\right).(n_x\hat{\mathbf{i}} + n_t\hat{\mathbf{j}})w_h d\Gamma - \\ \int_{\Omega_e} \left(u_h\frac{\partial w_h}{\partial t} + au_h\frac{\partial w_h}{\partial x}\right) d\Omega = 0,\end{aligned} \tag{4.60}$$

and the effect of un-resolved sub-scales is only through the flux. The true solution $u_h + u'$ is infinite-dimensional. However, only a few of its moments are required in the form of inner-products with low-order basis functions on element faces. In this limit, we

assume that the following approximation can be made: $u_s \approx u_h + u'$, where u_s denotes the super-resolved solution of u_h i.e.

$$\begin{aligned}\int_{\Gamma_e} \left(F_x^*(a(u_h+u'), a(u_h^- + u'^-))\hat{\mathbf{i}} + F_t^*(u_h+u', u_h^- + u'^-)\hat{\mathbf{j}}\right).(n_x\hat{\mathbf{i}} + n_t\hat{\mathbf{j}})w_h d\Gamma \\ \approx \int_{\Gamma_e} \left(F_x^*(au_s, au_s^-)\hat{\mathbf{i}} + F_t^*(u_s, u_s^-)\hat{\mathbf{j}}\right).(n_x\hat{\mathbf{i}} + n_t\hat{\mathbf{j}})w_h d\Gamma,\end{aligned} \tag{4.61}$$

where F_x^* and F_t^* are traditional up-wind numerical fluxes but computed using the super-resolved state u_s instead of u_h. In this work, we choose the spaces of u_h and u_s to be $p = 1$ and $p = 2$ respectively.

To obtain the super-resolved state, we will re-use the super-resolution network trained in the previous subsection. Two different initial conditions are chosen:(i.) Case A with an initial condition $sin(4x) + 2sin(8x)$ (ii.) Case B with an initial condition $2.5e^{-20(x-\pi)^2}$. These initial conditions are different from those used for training and testing. A comparison of different initial conditions used in training, reconstruction, and online evaluation is summarised in figure 4.9. The space-time slab is then discretized into 32 elements in the spatial direction, and the CFL value is taken to be 1.0. Figure 4.10 and 4.11 shows results obtained for the super-resolution model and the traditional model compared to the optimal solution obtained by L_2-projection of the DNS solution for the two different initial conditions, respectively. The super-resolution model is far more accurate than the traditional method, where the sub-scales were recovered with a high level of accuracy for both cases. In the case of the traditional method, extrema can be seen decreasing with time considerably in comparison to the super-resolution space-time method both in figures 4.10 and 4.11. This shows a higher dissipation characteristic of the traditional numerical method over the super-resolution method. This can also be quantitatively seen in both the figures 4.12(a) and 4.13(a), where the red line denoting time evolution of energy i.e. $E(t) = \int_{\Omega_x} u_h(x,t)^2 d\Omega_x$ decreases in time for the traditional approach in comparison to the optimal $p = 1$ solution which is oscillatory but energy conservative. Hence, the optimal solution oscillates about

a fixed value because the true solution itself conserves energy. The reason $E(t)$ oscillates for the optimal solution is because it is computed by integrating $u_h(x,t)^2$ over only space x and can still vary in time across the space-time slab.

In the next step, we obtain the space-time solution by first solving the problem using the traditional approach and applying our super-resolution model on this coarse solution. In this case, the super-resolution network has no contribution to the numerical simulation stage. Rather, it is used offline when the solution is made available. As shown in figure 4.12(a) and 4.13(a), application of the super-resolution in an offline stage does not improve the results. On the other hand, when the super-resolved states were used to compute the flux in the numerical method online, a high level of L_2-optimality was also obtained in the coarse solution as shown in figures 4.12(b) and 4.13(b). Consequently, the corresponding super-resolved solution was also accurate and close to the $p = 2$ optimal solution.

As observed previously in this section, the super-resolution did not improve the result when it was used on the data obtained using the traditional DG method. However, when used in an online setting, it improved the performance of the numerical method. This can be explained by figure 4.14, which shows the evolution of the DNS solution (red line) in an infinite-dimensional space. When one trains the super-resolution model, the mapping from a point on the green line (optimal LES) to its corresponding point on the red line (DNS) is learnt. However, when running an online numerical simulation using a traditional approach, the trajectory (blue line) taken by the solution (LES) is entirely different. The model encounters input parameters that it has not encountered during training and outputs an incorrect super-resolved solution in the evaluation stage. The blue line represents another optimal representation of DNS solution on the coarse solution space and not the L_2-optimal solution for which the method has been trained. However, when the super-resolved state is used to compute the fluxes, it forces the coarse resolution towards L_2-optimality because the closure has been formulated using the VMS method, where the coarse and fine spaces are formally defined. As shown in the second part of figure 4.14, consistent numerical

methods are required for the super-resolution models to work correctly. The VMS method is an ideal candidate to help us in achieving this consistency.

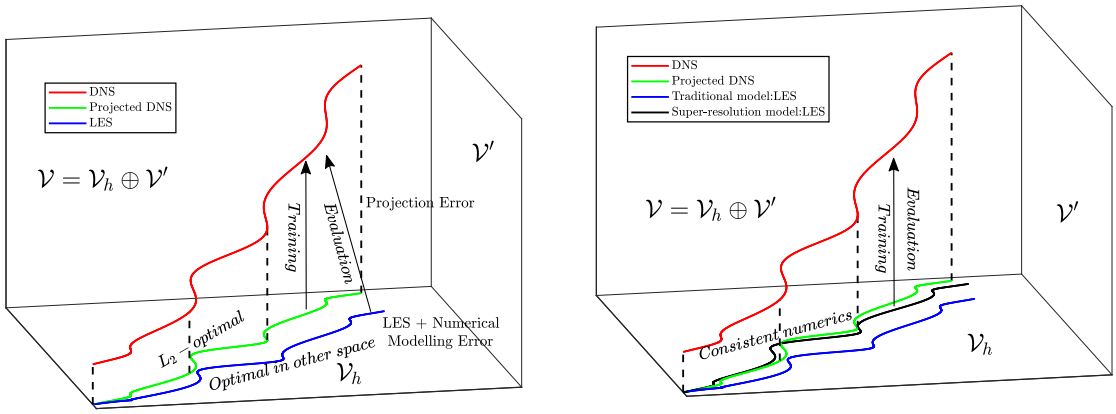


Figure 4.14: Sources of errors in offline and online super-resolution.

4.7 *p*-Super-resolution of turbulent channel flow.

The assessment of super-resolution models for turbulent flows poses a stern challenge. This is because, given limited measurements from a severely under-resolved coarse-grained solution such as LES, there are infinitely many possible solutions for the fine-scales. This problem is especially true for filtering using the sharp spectral filter, where the fine-scale solution is lost after filtering, and it is impossible to recover the original field from filtered data. The exact fine-scales are both functions of the coarse space and their time-history *Stinis* (2007); *Parish and Duraisamy* (2017a,b,c); *Pradhan and Duraisamy* (2020). As shown by Langford and Moser *Langford and Moser* (1999b) in their work on optimal LES, to compute the correct single-time multi-point statistical quantity of the large-scale field exact fine-scales might not be required. The Smagorinsky and the VMS models *Hughes et al.* (1998a); *Codina* (2002); *Codina et al.* (2007); *Bazilevs et al.* (2007); *Wang and Oberai* (2010a); *Masud and Calderer* (2011b); *Pradhan and Duraisamy* (2020), which perform well online, are well-known to perform poorly in an a priori setting. Similarly, the N-N generated super-resolved field is not point-wise exact. Rather it is an optimal representation

of the fine-space generating correct single-time multi-point statistics.

Our model being compact, the L_2 error is computed only locally in a single element. The training data consists of data from several elements, part of a 2-D DNS slice having homogeneity in stream-wise and span-wise directions. During the optimization, these errors from each element are averaged. As a result, the model output, in some sense, is an optimal representation of the fine-scales for all possible realizations. To this end, we will be using one-dimensional energy spectra that have been averaged over homogeneous directions as a measure for model accuracy in-place of the L_2 norm for the full field. The contours of the reconstructed fine-space solution will only be presented for qualitative purposes.

The first step is to generate data for training the model. As described in section 4.5, a single 2-D DNS snapshot (x-z plane) is extracted at a wall-normal height of $y^+ \approx 850$ and is L_2-projected on discontinuous polynomial spaces spanned by order $p = 1, 3$ tensor-product Lagrange basis functions on meshes of different sizes. In this case, we project the DNS solution on meshes with elements $N_x \times N_z$: 8×4, 16×8, 32×16, 64×32 in the x (stream-wise) and z (span-wise) directions respectively. Once the $p = 1, 3$ projected solutions are obtained, for each element present in these meshes, $p = 1$ coarse-scale basis coefficients are extracted for both the element and its immediate neighbors along with the $p = 3$ fine-scale basis coefficients. The next step is to evaluate the normalising parameter for each element i.e. $u_{rms} = \sqrt{\frac{\int_{\Omega_e}(u_1 - u_m)^2 d\Omega_e}{\int_{\Omega_e} d\Omega_e}}$, where the mean velocity u_m inside an element is given by $u_m = \frac{\int_{\Omega_e} u_1 d\Omega_e}{\int_{\Omega_e} d\Omega_e}$. Finally, a functional form similar to equation (4.40) is assumed except the parameter α is replaced with the logarithm of cell Reynolds number i.e. $log(Re_\Delta)$. The physics-informed feature $log(Re_\Delta)$ ensures that different orders of magnitude of Re_Δ in training data is accounted for. Finally, the normalized input and output basis coefficient data and the logarithmic cell Reynolds number $log(Re_\Delta)$ for each element are assembled into a single table as a training data-set.

A VSRNN architecture for the N-N model is then assumed with sizes: 37×32×32×32 for the part A, 16×32×32 for the part B, and a 32×64×64×16 sized post-multiplication

part, respectively. Finally, the model performance is evaluated in figures 4.16 and 4.15 by comparing the stream-wise and span-wise energy spectra obtained for the super-resolved $p = 3$ solution and the $p = 3$ L_2-optimal solution at different wall-normal heights of $y^+ \approx 500, 800, 850$. To compute the energy spectra, the solution in first obtained on a uniform mesh with size $(p+1)N_x \times (p+1)N_z$ where the factor $p+1$ accounts for the effective grid-size at higher orders. This also prevents the over-sampling of the data.

As can be observed in figures 4.16 and 4.15, the network can successfully recreate the correct energy distribution across different wave-numbers both in the stream-wise and the span-wise directions. This is true for both the cases: the plane at $y^+ \approx 850$, which was used for training, and the unseen planes at $y^+ \approx 500$ and $y^+ \approx 800$. However, the energy at the high wave-number modes for all these cases was slightly higher than the L_2-projected $p = 3$ optimal solution suggesting that a small amount of de-aliasing would be helpful. A qualitative plot of the coarse $p = 1$ solution, the super-resolved $p = 3$ solution and the L_2-optimally projected $p = 3$ solution for different grid sizes at a wall-normal distance of $y^+ \approx 500$ is shown in figure 4.17. It can be observed that the super-resolved solution, similar to the optimal $p = 3$ projected solution, has finer structures when compared to the coarse $p = 1$ solution at different mesh resolutions.

The generalizability of the model trained using a single snapshot of DNS data stems from the fact that when the DNS image is projected on several finite element meshes with different element sizes, the average value of the cell Reynolds number changes. As a result, the training data contains an extensive range of cell Reynolds numbers. When the trained model is evaluated at different wall-normal distances while retaining the same the grid size, the cell Reynolds number changes due to changes in u_{rms}. However, this new cell Reynolds number can also be obtained at a previous wall-normal height by changing the grid-size alone. This can also be observed in equation (4.33) for the normalized sub-scales. The normalized sub-scales only depend on the cell Peclet number α and the non-dimensionalized inputs rather than the grid size or the diffusion coefficient separately.

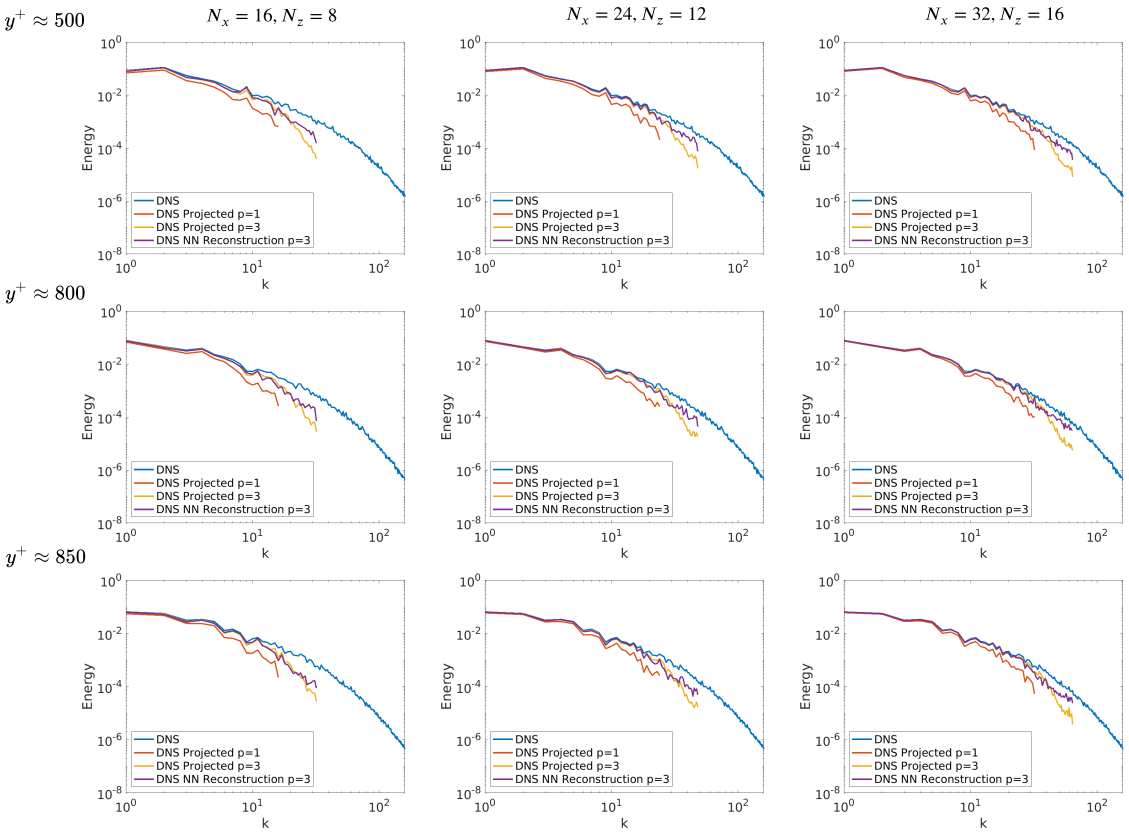


Figure 4.15: Stream-wise energy spectra obtained for the L_2-projected stream-wise velocity solution on $p = 1$, L_2-projected stream-wise velocity solution on $p = 3$, N-N super-resolved $p = 3$ solution and DNS at different wall normal height y^+ and mesh resolutions.

4.7.1 Sub-grid Modelling

The previous subsection showed that the super-resolution model accurately reconstructs the high-order optimal solution from the low-order optimal solution. We consider LES of the compressible Navier–Stokes equation, but for simplicity of presentation, we only detail the development for the inviscid terms. The domain $\Omega \subset \mathbb{R}^d$ is used with the boundary $\Gamma = \partial\Omega$, where $d \geq 1$ is the dimension of the problem as follows,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0, \tag{4.62}$$

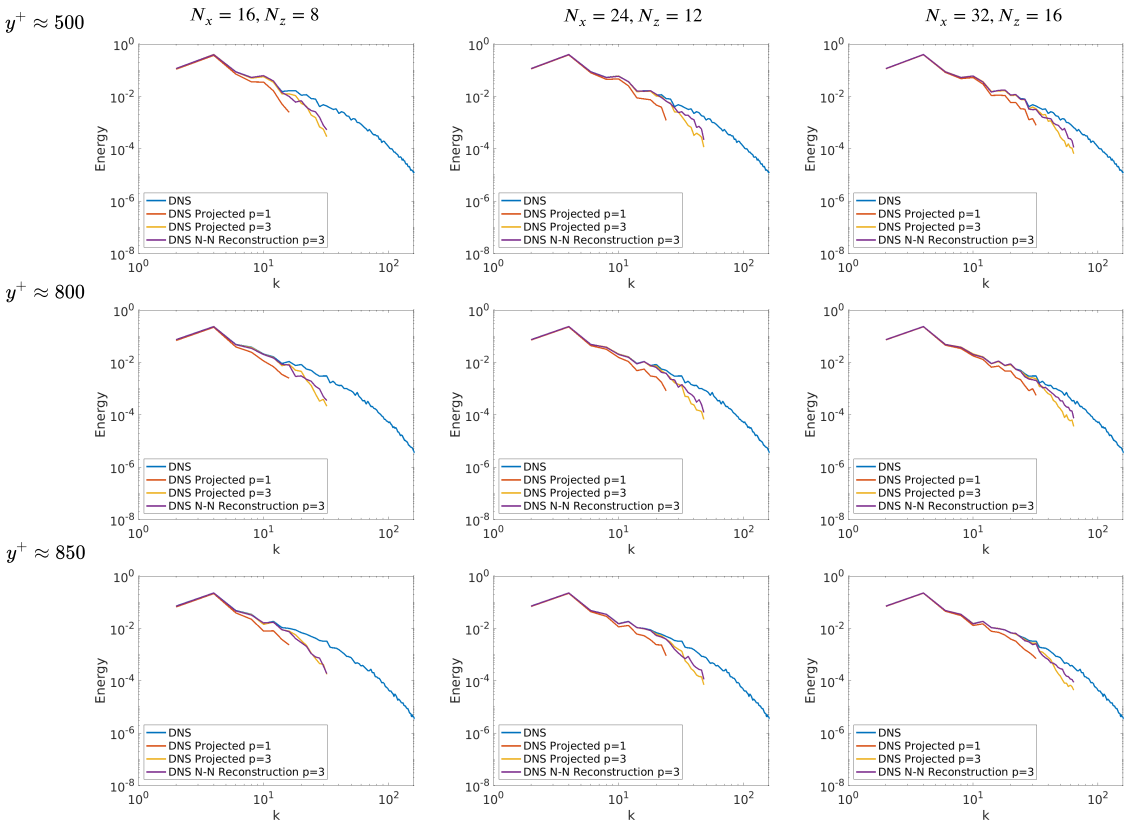


Figure 4.16: Span-wise energy spectra obtained for the L_2-projected stream-wise velocity solution on $p = 1$, L_2-projected stream-wise velocity solution on $p = 3$, N-N super-resolved $p = 3$ solution and DNS at different wall normal height y^+ and mesh resolutions.

with appropriate boundary conditions on Γ and time $t \in (0, T]$. The state vector $\mathbf{u}$ is given by:

$$\mathbf{u} = [\rho, \rho u, \rho v, \rho w, \rho E]^T, \tag{4.63}$$

and the matrix $\mathbf{F}(\mathbf{u})$ corresponding to the flux is given by:

$$\mathbf{F}(\mathbf{u}) = \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p & \rho uv & \rho uw \\ \rho vu & \rho v^2 + p & \rho vw \\ \rho wu & \rho wv & \rho w^2 + p \\ \rho uH & \rho vH & \rho wH \end{bmatrix}. \tag{4.64}$$

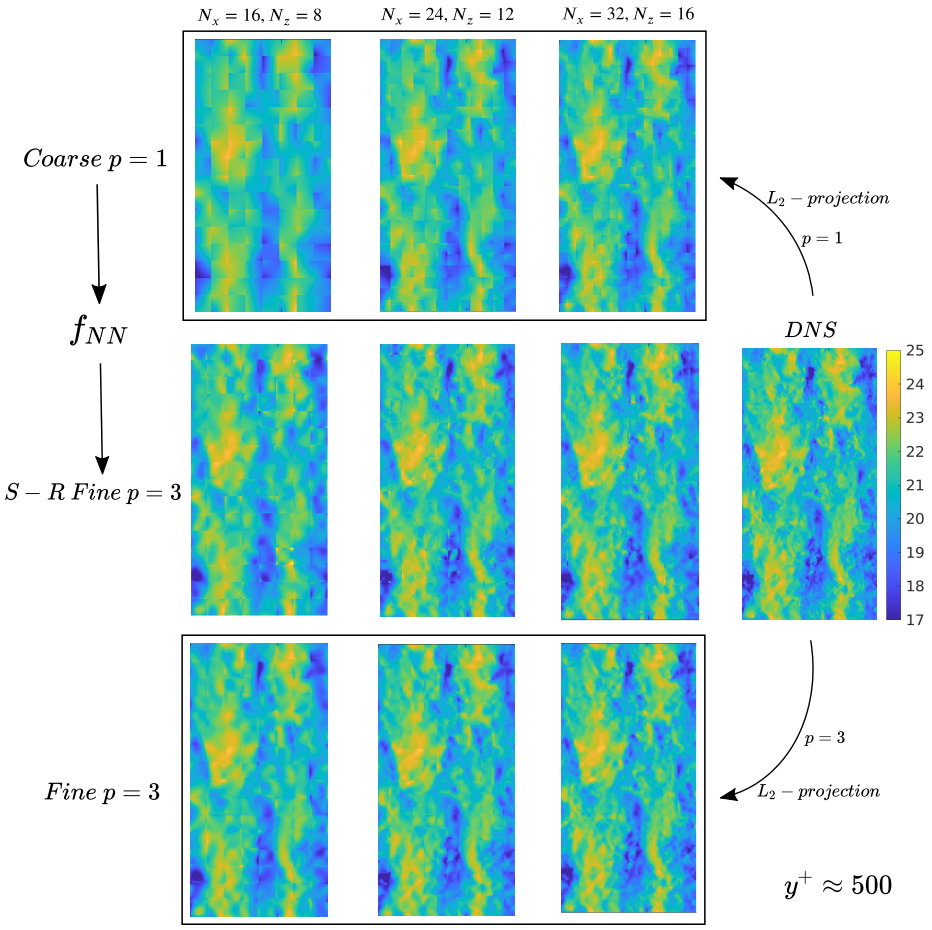


Figure 4.17: Stream-wise Velocity contours of the coarse solution, super-resolved fine solution and the corresponding optimal fine solution. The y-axis is aligned with the wall-normal direction.

Let us also define $\mathcal{T}_h$ as a tessellation of the domain Ω into a set of non-overlapping elements, K, each having a sub-domain Ω_k and boundary $\partial\Omega_k$. The DG weak form is then obtained by multiplying with weighting functions $\mathbf{w}$ and performing integration by parts on an element:

$$\int_{\Omega_k} \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) \right) \cdot \mathbf{w} d\Omega = 0. \tag{4.65}$$

After application of integration by parts we obtain:

$$\int_{\Omega_k} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{w} d\Omega - \int_{\Omega_k} \nabla \mathbf{w} : \mathbf{F}(\mathbf{u}) d\Omega + \int_{\partial\Omega_k} \mathbf{w} \cdot \mathbf{F}^*(\mathbf{u}, \mathbf{u}^-) d\Gamma = 0. \tag{4.66}$$

Using the Galerkin approximation we have,

$$\int_{\Omega_k} \frac{\partial \mathbf{u_h}}{\partial t} \cdot \mathbf{w_h} d\Omega - \int_{\Omega_k} \nabla \mathbf{w_h} : \mathbf{F}(\mathbf{u_h}) d\Omega + \int_{\partial\Omega_k} \mathbf{w_h} \cdot \mathbf{F}^*(\mathbf{u_h}, \mathbf{u_h^-}) d\Gamma = 0, \tag{4.67}$$

where components of $\mathbf{u_h}$, i.e., $u_{h,i} \in \mathcal{V}_{h,3}$ and $\mathcal{V}_{h,3}$ is the space of $p = 3$ tensor-product polynomial basis functions as follows:

$$\mathcal{V}_{h,3} \triangleq \left\{ u \in L_2(\Omega) : u|_{\mathcal{T}} \in P^3(T), T \in \mathcal{T}_h \right\}. \tag{4.68}$$

The numerical flux $\mathbf{F}^*$ is assumed to be the Roe flux, and an under-resolved model results in a sub-optimal solution $\mathbf{u_h}$, i.e.,

$$\mathbf{u_h} \neq \mathbf{u_3} = \Pi_3 \mathbf{u}, \tag{4.69}$$

where Π_3 denotes L_2-projection on $\mathcal{V}_{h,3}$. Similarly, the large-scales in $\mathbf{u_h}$ will also be inconsistent i.e.

$$\mathbf{u_{h,1}} = \Pi_1 \mathbf{u_h} \neq \mathbf{u_1} = \Pi_1 \mathbf{u_3} = \Pi_1 \mathbf{u}, \tag{4.70}$$

where Π_1 is a L_2-projects onto a coarse-space formed by tensor-product of $p = 1$ basis functions in the stream-wise and the span-wise directions and $p = 3$ basis function in the wall-normal direction. This results in coarsening in the stream-wise and the span-wise directions only. The coarse part of the solution $\mathbf{u_h}$ obtained after projection i.e. $\mathbf{u_{h,1}} = \Pi_1 \mathbf{u_h}$ is better resolved in the coarse space $\mathcal{V}_{h,1}$ in comparison to $\mathbf{u_h}$ in the fine space $\mathcal{V}_{h,3}$. This is because the numerical dissipation due to the standard numerical flux is more likely to corrupt the smaller scales in comparison to the larger, resolved scales. We can now use our model to super-resolve $\mathbf{u_{h,1}}$ back to $\mathbf{u_s} \in \mathcal{V}_{h,3}$. as follows

$$\mathbf{u_s} = f_{NN}(\mathbf{u_{h,1}}, ...), \tag{4.71}$$

where $\mathbf{u_s}$ is the super-resolved state in the element when the coarse-scale solution in the element and its neighbours are given by $\{\mathbf{u_{h,1}}, ...\}$. The super-resolution of each state is performed independently on 2-D planes (wall-parallel planes). The size of the network used for super-resolution is reduced to 36×32×16×16 to ensure computational efficiency. Finally, in a similar approach to section 4.6.2, the super-resolved state is used to compute the flux terms in the DG formulation as follows:

$$\int_{\Omega_k} \frac{\partial \mathbf{u_h}}{\partial t} \cdot \mathbf{w_h} d\Omega - \int_{\Omega_k} \nabla \mathbf{w_h} : \mathbf{F}(\mathbf{u_h}) d\Omega + \int_{\partial\Omega_k} \mathbf{w_h} \cdot \mathbf{F}^*(\mathbf{u_s}, \mathbf{u_s}^-) d\Gamma = 0. \tag{4.72}$$

The application of the super-resolved state $\mathbf{u_s}$ directly in the boundary flux term makes it unstable when used with the explicit R-K type time-stepping methods. To stabilize this approach (SR-LES), the super-resolution process is relaxed as follows:

$$\int_{\Omega_k} \frac{\partial \mathbf{u_h}}{\partial t} \cdot \mathbf{w_h} d\Omega - \int_{\Omega_k} \nabla \mathbf{w_h} : \mathbf{F}(\mathbf{u_h}) d\Omega \tag{4.73}$$

$$+ \int_{\partial\Omega_k} \mathbf{w_h} \cdot \mathbf{F}^* \left((1-\lambda)\mathbf{u_h} + \lambda \mathbf{u_s}, (1-\lambda)\mathbf{u_h}^- + \lambda \mathbf{u_s}^-\right) d\Gamma = 0. \tag{4.74}$$

A value $\lambda = 0.1 - 0.2$ is chosen for the following numerical simulations. Although a higher value of the relaxation factor λ is desirable, stability generally demands $\lambda \leq 0.2$. The discretization of the viscous terms in the compressible Navier-Stokes equation is performed using the second form of Bassi and Rebay *Bassi and Rebay* (2000b) scheme. The boundary terms arising due to the viscous fluxes are also evaluated using the under-relaxed super-resolved state similar to the inviscid fluxes.

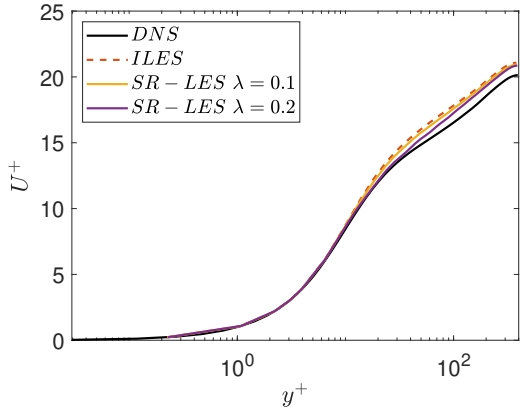
To compare the performance of different models, we perform LES of channel flow at $Re_\tau \approx 395$. The number of elements in stream-wise (x), wall-normal (y) and span-wise (z) directions are $N_x = 24$, $N_y = 24$ and $N_z = 12$ respectively. Similarly, the size of the domain in these directions is taken to be $[L_X, L_y, L_z] : [2\pi\delta, 2\delta, \pi\delta]$, respectively. The

stream-wise and span-wise element sizes in wall-units are $\Delta x^+ \approx 103.41$ and $\Delta z^+ \approx 103.41$ respectively. The element sizes in wall-normal direction vary from $\Delta y^+_{min} \approx 3.37$ near the wall to $\Delta y^+_{max} \approx 51.55$ at the center of the channel. Since, high-order basis functions are used, the effective grid size can be approximated by $\Delta_{eff} \approx \frac{\Delta}{p}$, where Δ is the element size and p is the order of the polynomial i.e. $p = 3$. Time marching was performed using the explicit RK3-TVD scheme for all the cases. Figure 4.18 shows the velocity statistics obtained for the channel flow problem using ILES, SR-LES and DNS. The performance improves as λ is increased to 0.2. This is observed both in the mean velocity profile, and the stream-wise root mean square (RMS) velocity profile. The RMS peak of the stream-wise velocity obtained for both $\lambda = 0.1$ and $\lambda = 0.2$ is closer to DNS and lower than ILES. The span-wise and wall-normal RMS velocity statistics and the turbulent shear-stresses obtained are comparable for all three LES cases.

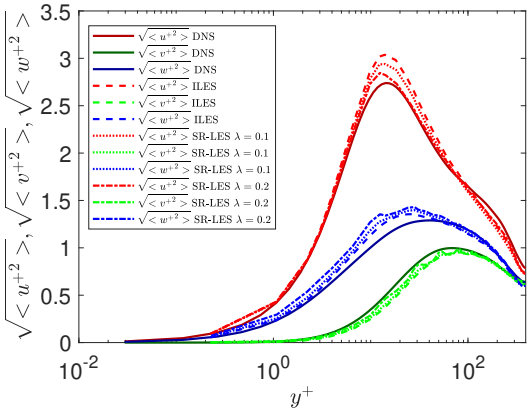
A maximum relaxation factor of $\lambda = 0.2$ also suggests that additional stabilization is required for the model to work at higher values of λ where the model is expected to work better. One of the reasons for the constraint in λ is due to the explicit time-stepping scheme was used. No such factor was needed for the linear advection case in the previous section, where an implicit space-time method was employed. As discussed further in the perspectives section (Chapter 6), additional challenges have to be addressed to ensure success of super-resolution networks for predictive modeling of turbulent flows.

4.8 'Nodally exact' high-order CG schemes for 1-D Convection-Diffusion

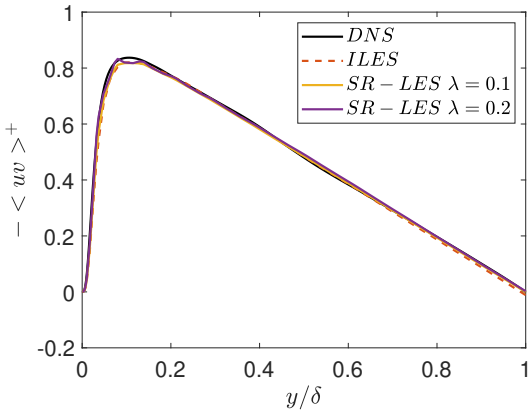
To demonstrate the action of sub-scales in 4.3, we assumed that the coarse space to be composed of piece-wise linear polynomials. However, this approach can be extended to higher-order polynomials as well. In this section, we will use the VSRNN architecture to learn closures for high order CG discretizations where the coarse-scale is 'nodally exact'. The governing PDE is again taken to be the linear convection-diffusion equation as follows:



(a) Stream-wise mean velocity profile U^+ vs. channel wall normal height y^+ in wall units.



(b) Root mean square of velocity components $\sqrt{< u^{+2} >}$, $\sqrt{< v^{+2} >}$,$\sqrt{< w^{+2} >}$ vs. channel wall normal height y^+ in wall units.



(c) Resolved turbulent shear-stress -$< uv >^+$ vs. channel wall normal height y/δ normalised with the semi-channel height.

Figure 4.18: Velocity statistics for channel flow using ILES and SR-LES at $Re_\tau \approx 395$. The x-axis, y-axis and z-axis are aligned with the stream-wise, wall-normal and the span-wise directions, respectively.

$$\mathcal{L} \triangleq a\frac{d}{dx} - \kappa\frac{d^2}{dx^2} \quad in \quad \Omega = [0, L] \tag{4.75}$$

with Dirichlet boundary conditions: $u(0) = u_0$ and $u(L) = u_L$. To derive VSRNN closures for 'nodally exact' coarse-scales, we start with the variational form:

$$(a\frac{du}{dx} - \nu\frac{d^2u}{dx^2}, w) = 0 \tag{4.76}$$

The weak form after integration by parts is obtained as follows:

$$\left(a\frac{du}{dx}, w\right) + \kappa\left(\frac{du}{dx}, \frac{dw}{dx}\right) = 0 \tag{4.77}$$

The next step is to apply the VMS decomposition such that the coarse-scale is exactly the interpolate of the true solution at nodal points i.e.

$$(a\frac{du_h}{dx}, w_h) + \kappa(\frac{du_h}{dx}, \frac{dw_h}{dx}) + (a\frac{du'}{dx}, w_h) + \kappa(\frac{du'}{dx}, \frac{dw_h}{dx}) = 0 \tag{4.78}$$

Since the coarse-scale is the true interpolant of the solution, the sub-scales should vanish at the nodal points. Hence, integration by parts can be performed as follows:

$$(a\frac{du_h}{dx}, w_h) + \kappa(\frac{du_h}{dx}, \frac{dw_h}{dx}) + \sum_e \int_{\Omega_e} u'(-a\frac{dw_h}{dx} - \kappa\frac{d^2w_h}{dx^2})d\Omega = 0 \tag{4.79}$$

It can be recognized that the sub-scale lies in an infinite dimensional space. However, for each element only its inner-product with $-a\frac{dw_h}{dx} - \kappa\frac{d^2w_h}{dx^2}$ needs to be computed inside each element. Hence, if p is the order of the polynomial used to describe the coarse-scales, L_2-projecting u' in a discontinuous polynomial space inside the element consisting of polynomials up to order $p - 1$ is sufficient. To learn these projected sub-scales, the

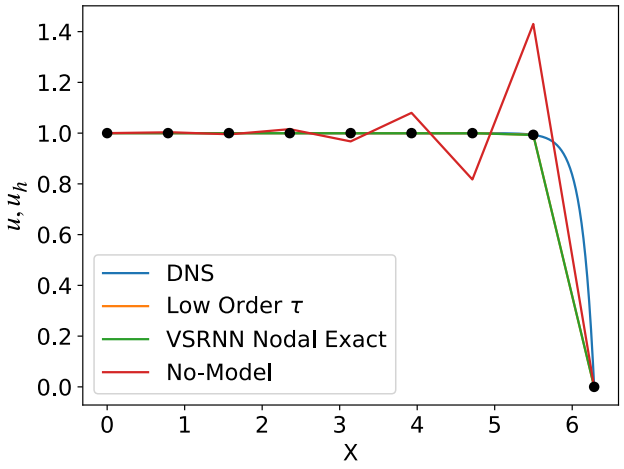
VSRNN is used as follows:

$$\left[C'_{1,p-1}, C'_{2,p-1}, .., C'_{p,p-1}\right] = \mathbf{f}\left(log(\alpha), \left[\tilde{C}_{1,p}, \tilde{C}_{2,p}, .., \tilde{C}_{p+1,p}\right]\right),$$

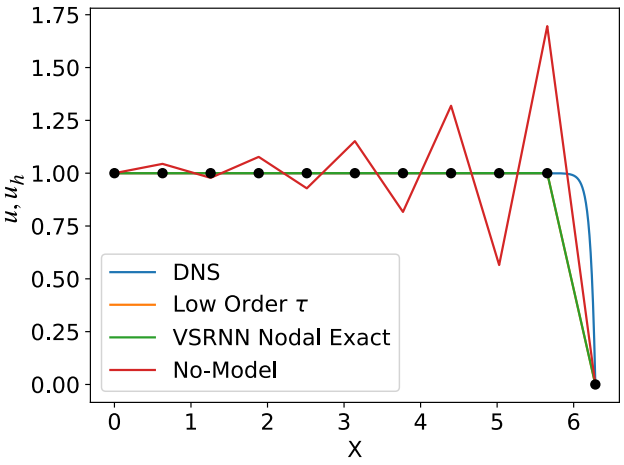
where, $\left[C'_{1,p-1}, C'_{2,p-1}, .., C'_{p,p-1}\right]$ represents the sub-scale basis coefficients normalised by coarse-scale R.M.S u_{rms}, $[\tilde{C}_{1,p}, \tilde{C}_{2,p}, .., \tilde{C}_{p+1,p}]$ represents the mean u_{mean} subtracted and u_{rms} normalised coarse-scale basis coefficients, and α is the Peclet number.

It is noted that although u' is infinite dimensional, $\left[C'_{1,p-1}, C'_{2,p-1}, .., C'_{p,p-1}\right]$ is not. It is sufficient to learn these projected sub-scales to precisely compute the required inner-products. For example, when the coarse solution is linear ($p = 1$), the sub-scale can be represented by $p = 0$ constant functions which corresponds to commonly used $\tau = \frac{h}{2a}(coth(\alpha) - \frac{1}{\alpha})$. The online evaluation of the $p = 1$ closure presented in Figure 4.4 is shown in figure 4.19. As expected, the VSRNN is able to precisely recreate the results obtained using the analytical expression for τ.

The data generation procedure used here for deriving closures for higher order polynomials is same as that used in section 4.4 for the $p = 1$ case. The model is trained and applied to equation (4.79). Two different cases with global Peclet numbers $Pe_g = \frac{aL}{\kappa}$ =20 and 40 are considered here. For each case, two CG elements with different polynomial orders $p = 3, 4, 7$ and 8 are used to discretize the domain. Figure 4.20 and 4.21 shows the comparison of the present VSRNN closure to existing closures and no-model for global Peclet numbers of 20 and 40 respectively. The VSRNN model in both the cases accurately learns the sub-scales and ensures that the coarse-scale is the interpolate of the true solution. As expected, $\tau = \frac{h}{2a}(coth(\alpha) - \frac{1}{\alpha})$ based on low-order discretization is not accurate at high-orders and the no-model discretization (Galerkin) is oscillatory when resolution is not sufficient. The no-model performance increases at high-order because effective resolution increases with p.

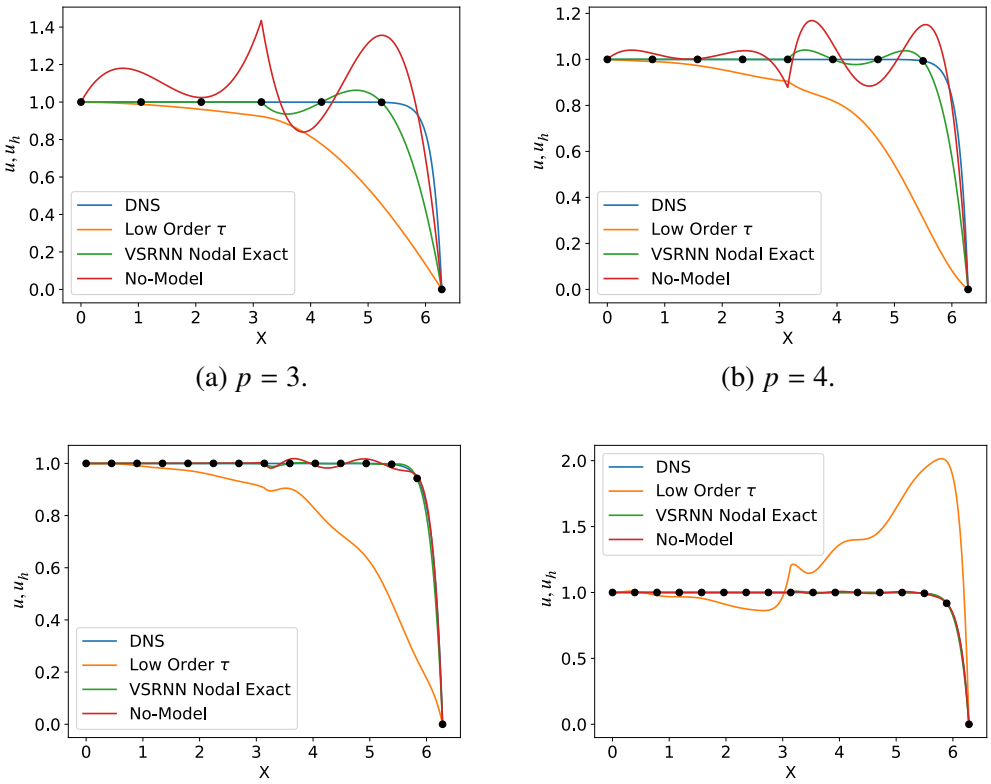


(a) $Pe_g = 20$.



(b) $Pe_g = 40$.

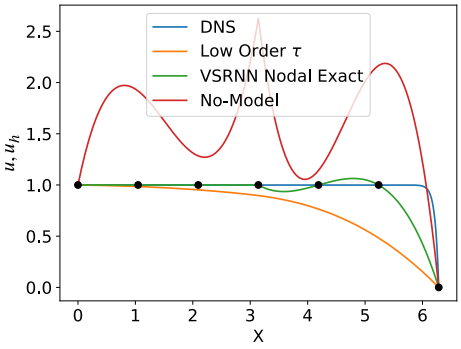
Figure 4.19: Comparison of VSRNN closure to existing closure for $p = 1$ CG finite elements at different Peclet numbers.



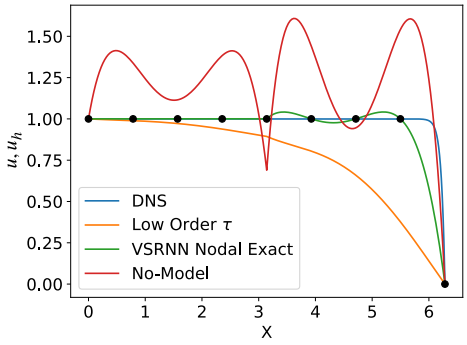
(a) $p = 3$. (b) $p = 4$.

(c) $p = 7$. (d) $p = 8$.

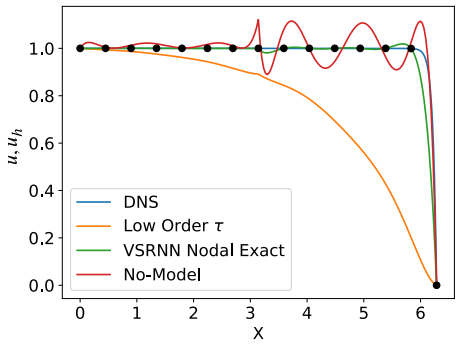
Figure 4.20: Discretizations of the 1-D Convection-Diffusion equation at $Pe_g = 20$ using two CG elements $N_{ele} = 2$ and different polynomial orders $p = 3, 4, 7, 8$.



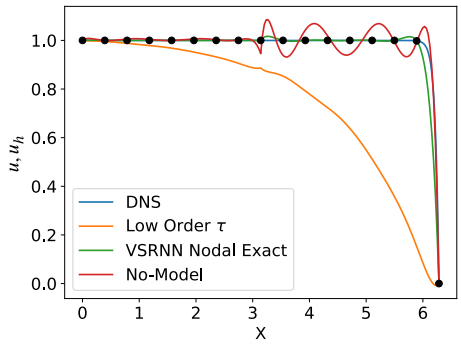
(a) $p = 3$.



(b) $p = 4$.



(c) $p = 7$.



(d) $p = 8$.

Figure 4.21: Discretizations of the 1-D Convection-Diffusion equation at $Pe_g = 40$ using two CG elements $N_{ele} = 2$ and different polynomial orders $p = 3, 4, 7, 8$.

4.9 Conclusions

We proposed a strategy for multi-scale modeling in which the coarse and the fine scales are defined in terms of projection onto their respective finite element spaces, and segregated using a variational multiscale formulation. Existing variational multiscale formulations provide guiding principles for the construction of consistent features and network architecture to define a super-resolution model of the fine scales. Particularly, we define an architecture - called the Variational super-resolution neural network (VSRNN) - which approximates the sub-scales as a sum of products of individual functions of coarse-scales and the physics-informed parameters. This model form and network structure is inspired by analytical expression for the sub-scales as given by the convection-diffusion equation. It is emphasized that traditional architectures - such as a fully connected neural network - are not ideal for this purpose because they combine heterogeneous quantities (e.g. coarse-scale basis coefficients and physics informed-parameters) as inputs. The input features and output quantities are obtained by appropriately non-dimensionalizing the coarse-scales and the sub-scale basis coefficients. By applying the super-resolved state to compute the Discontinuous Galerkin (DG) fluxes, we ensure that the online coarse-scale solution is forced towards its L_2-optimal state.

We verify that when the present approach is applied to the convection-diffusion problem, it can learn the analytical solution to a high degree of accuracy. Similarly, for the 1-D linear-advection space-time problem, the model could accurately super-resolve low-order coarse solutions to high-order fine-solution. The network could also reproduce super-resolved velocity fields with the proper energy distribution across different wave-numbers in the stream-wise and the span-wise direction for the turbulent channel case.

Next, we assessed the predictive capability of these models. Super-resolution was used to determine the DG fluxes for the linear-advection problem, and shown to result in higher accuracy and optimality of the method over traditional space-time methods for the same number of degrees of freedom. When applied to LES of turbulent channel flow, this

approach led to a more modest performance improvement. This improvement stems from the fact that the present model has been trained using L_2-optimal fine and coarse solutions, leading to sub-grid models that are consistent with the type of optimality sought. The present method was found to generalize to out-of-sample initial conditions and Reynolds numbers for both the linear advection and the turbulent channel flow cases. Similarly, when the fine scales from the VSRNN model were used to compute the sub-grid terms arising in the high-order continuous Galerkin discretization of the convection-diffusion equation, highly accurate numerical schemes were obtained.

In addition to reconstruction and sub-grid modeling, the super-resolution model can be used as an error indicator for adaptive grid refinement : Regions in which the high magnitude of the sub-scale values can be used as a measure for under-resolution. Finally, the authors would like to point out that effective implementation of this approach solvers requires the development of efficient non-linear solvers and preconditioners to handle the additional non-linearity and stiffness due to the model.

CHAPTER V

Optimal Finite Element Projections of Scale-Resolving Turbulent Flow Simulations

5.1 Background and Motivation

As discussed in the previous chapters, the simulation of turbulent flows remains a challenge in most practical flows because of the disparate range of spatial and temporal scales that need to be resolved (*Pope*, 2000). An alternative to directly solving the Navier-Stokes equations is to solve its reduced complexity versions. Reynolds averaged Navier-Stokes (RANS) models solve for the ensemble average or time average of the true solution. Large Eddy Simulations (LES) (*Germano et al.*, 1991; *You and Moin*, 2007a; *Meneveau et al.*, 1996; *Nicoud et al.*, 2011; *Vreman*, 2004; *Nicoud and Ducros*, 1999; *Codina*, 2002; *Codina et al.*, 2007; *Bazilevs et al.*, 2007; *Wang and Oberai*, 2010a; *Gravemeier et al.*, 2010; *Masud and Calderer*, 2011b; *Parish and Duraisamy*, 2017c) resolve the spatio-temporal dynamics of the large scales. The cost of LES is, however, still prohibitive near the wall. To alleviate the need of mesh refinement near the wall, boundary conditions are imposed weakly in a wall-modeled LES (WMLES) (*Piomelli and Balaras*, 2002; *Bose and Moin*, 2014; *Bae et al.*, 2019). Alternate approaches to WMLES are hybrid RANS-LES techniques like DES (*Spalart*, 2009) and IDDES (*Shur et al.*, 1999), where the inner-layer is solved using RANS and the rest using LES. Consequently, the cost associated with resolving the near-

wall structures in the stream-wise and span-wise directions is no longer present. However, the cost associated with resolving the wall-normal gradient in hybrid RANS-LES is still present.

Over the past few decades, various contributions have been made in the development and application of these methods to highly complex problems (e.g. *Goc et al.* (2020); *Lozano-Duran et al.* (2020); *Park and Moin* (2016); *Iyer and Malik* (2020); *Goc et al.* (2021); *Kiris et al.* (2022)). Our view is that, since all of the scale-resolving methods are coarse-grained from the Navier–Stokes equations, there must exist a unified view. The Partially-averaged Navier Stokes (PANS) approach brings together several turbulence closures of various modeled-to-resolved scale ratios ranging from Reynolds-averaged Navier Stokes (RANS) to Navier-Stokes (direct numerical simulations (DNS)) into one formulation (*Girimaji and Abdol-Hamid*, 2005). The behavior of the PANS equations can be varied smoothly from the RANS equations to the Navier-Stokes (DNS) equations by changing the filter-width control parameters. The unified RANS-LES approach *Heinz* (2007); *Gopalan et al.* (2013) is an optimal hybrid RANS–LES framework that uses different time-scales to switch between the RANS and LES approaches. In pursuit of similar unified models and in an effort to augment existing frameworks, we propose a filtering technique using optimal finite element projections which: (i) offers a unifying perspective through a common coarse-graining strategy; (ii) provides optimal solutions for the existing coarse-grained methods to improve upon.

The use of filtered DNS data to perform a priori analysis of closure models for RANS and LES is indeed not new. LES models such as the scale-similarity or Smagorinsky models have also been frequently evaluated against sub-grid stresses obtained from filtered DNS data (*Vreman et al.*, 1995; *Bou-Zeid et al.*, 2008; *Meneveau and Katz*, 2000; *Girimaji and Abdol-Hamid*, 2005). In most of the prior studies, filtering is either performed in the Fourier space using the sharp spectral cutoff or Gaussian filters when the problem has periodic directions or the box-filter in more complex problems. In case of filters that are

applied in spectral space, the filter width remains the same along the periodic directions in which it is applied. However, as observed in most coarse grained simulations, the filter width can vary considerably. In fact, filter sizes define these methods. For example, in case of a LES of channel flow that is performed on a structured grid, the filter size in the span-wise and stream-wise directions can be a constant. However, the filter width in the wall-normal direction can vary from a few wall units near the wall to 0.1δ at the center of the channel or the edge of the boundary layer. In WMLES, the filter width is approximately of the order of 0.1δ throughout in all directions. For hybrid RANS-LES approaches (such as DES and IDDES), the filter width is of the order of δ in the span-wise and stream-wise directions, and similar to LES in the wall-normal direction. The non-uniform filtering requirement in LES and RANS-LES methods stems from the fact that in both the cases, the wall-stress is resolved which requires a near-wall grid that scales with wall units. In addition to the filter size, the type of filter can also change the nature of the solution. For example, both the box and spectrally (sharp) filtered DNS both qualify as synthetic LES solutions. In the case of finite element projections, the quality of the filter is linked to the order of polynomial used to filter the solution. In this work, we aim to address some of these issues by using finite element projections which allow for variation in the filter width in the domain and also provide the required flexibility to change the quality of the filter by changing the order of the polynomial.

The idea of projection is at the core of the variational multiscale method (VMS) (*Hughes et al.*, 1998b; *Codina*, 2002; *Codina et al.*, 2007; *Bazilevs et al.*, 2007; *Wang and Oberai*, 2010a; *Gravemeier et al.*, 2010; *Masud and Calderer*, 2011b; *Parish and Duraisamy*, 2017c). In VMS, projections are used to formally distinguish the coarse-scales from the fine-scales. The coarse-scale (filtered) solution that is obtained after the projection operation represents the 'best' coarse-grained solution u on the coarse-space based on some optimality condition, for example, the L_2-optimality condition. In this work, we perform L_2-projections on finite element basis functions. It is pertinent to mention that the current

idea of optimal projections should not be confused with optimal LES (*Langford and Moser*, 1999b). Our work optimally represents the DNS solution u on a finite dimensional coarse-space, whereas, optimal LES is an ideal LES model that computes accurately single-time multi-point statistics of the coarse-solution.

Projections of the DNS data on finite element basis have been previously used to both improve existing finite element methods (*Pradhan and Duraisamy*, 2021) and turbulence models (*Vreman et al.*, 1995; *Bou-Zeid et al.*, 2008; *Meneveau and Katz*, 2000; *Girimaji and Abdol-Hamid*, 2005). In this chapter, however, we employ them as an alternate to traditional filters for scale-separation. As discussed previously, the present approach has an advantage in cases where the filter length is anisotropic, varies rapidly, or when non-homogeneous directions are present, as in wall-bounded flows. The filtering strategy that has some similarities to the present approach is the differential filter (*Germano*, 1986; *Najafi-Yazdi et al.*, 2015), which consists of a filtering length scale l_p. This length scale l_p can be varied along the domain to have a similar effect.

In the past few years, several efforts have been made to train sub-grid models using machine learning approaches both in an offline and model-consistent setting (*Maulik and San*, 2017; *Maulik et al.*, 2018, 2019; *Beck et al.*, 2019; *Sarghini et al.*, 2003; *Gamahara and Hattori*, 2017; *Wang et al.*, 2018; *Xie et al.*, 2019a,b, 2020). Such data-driven LES models require a filtered form of the DNS, which in turn will depend on the filter size. Similarly, in WMLES, the model cannot be trained using the mean solution in the first few grid points where the influence of the slip condition will be observed. By applying projections and obtaining statistics from the optimal solution, one can obtain more reasonable targets to training the model (*Beck et al.*, 2019; *Duraisamy et al.*, 2019; *Chung and Freund*, 2022) on more complex problems. Note that this is a first step towards addressing model consistency *Duraisamy* (2021). Finite element projections are not restricted to simple geometries (Appendix D) and can be applied to more complex flows, and provide the additional flexibility of choosing polynomial orders for the geometry and the solution

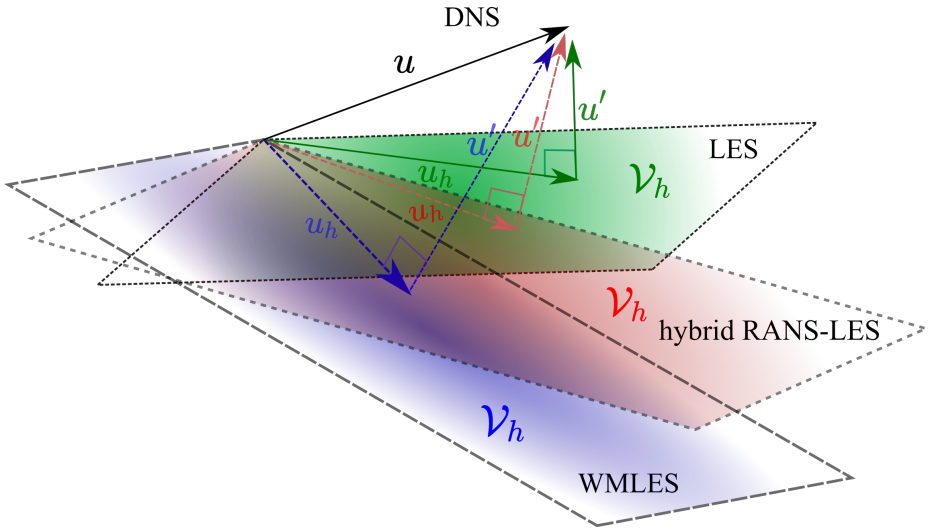


Figure 5.1: Schematic of the projection of the DNS solution u on coarse finite element spaces $\mathcal{V}_h$ to obtain the L_2-optimal LES, WMLES or hybrid-RANS solution u_h.

independently.

The main objective of the present work is to develop a unified framework that can be used as a lens to quantitatively assess, augment and calibrate a wide range of coarse-grained models. Particular attention is paid to the behavior of various models in the proximity of the wall, and to ascertain whether scaling relationships exist.

In section 5.2 of this chapter, we describe the procedure of performing L_2-projection and provide a discussion on the choice of coarse basis functions that will be used. In section 5.3, we compute filtered solutions for the channel flow problem in the LES, WMLES and hybrid RANS-LES limit, and compute the coarse-scale statistics in each of these cases. In section 5.4, we show that the slip velocity in case of WMLES is a natural consequence of under-resolution in the wall-normal directions and guiding principles for improved slip wall models are proposed. In section 5.5, we propose a new slip-wall model and assess it online performance against the traditional method. Finally, we summarize our work in section 5.6.

5.2 Finite element projection

In this section, we will describe the DNS dataset and the procedure of computing the filtered/coarse-scale solution. For our purpose, we use the channel flow data at friction Reynolds numbers of $Re_\tau \approx 1000$ and $Re_\tau \approx 5200$ from the Johns Hopkins Turbulence Database (JHTDB) (*Li et al.*, 2008; *Perlman et al.*, 2007; *Lee and Moser*, 2015) and a smaller channel $Re_\tau \approx 950$ case form the Texas turbulence file server (*Hoyas and Jiménez*, 2006; *Del Alamo et al.*, 2004). To this end, consider the decomposition of the full-order (DNS) solution u into coarse and fine scales as follows:

$$u = u_h + u', \tag{5.1}$$

where $u_h \in \mathcal{V}_h$ and $u' \in \mathcal{V}'$. The vector space of functions $\mathcal{V} \equiv L^2(\Omega)$ is the space of square-integrable functions. This space is decomposed as follows:

$$\mathcal{V} = \mathcal{V}_h \oplus \mathcal{V}', \tag{5.2}$$

where $\oplus$ represents a direct sum of $\mathcal{V}_h$ and $\mathcal{V}'$. Let us also define $\mathcal{T}_h$ to be a tessellation of domain Ω into a set of non-overlapping elements, K, each having a sub-domain Ω_k and boundary Γ_k. The functional space $\mathcal{V}$ is infinite dimensional and must be approximated by a finite dimensional approximation $\mathcal{V}_h$. The domain and boundary of an element marked by Ω_e and Γ_e respectively. Also consider the following notations:

$$\Omega' = \bigcup_{i=1}^{n_{el}} \Omega^e \quad \text{and} \quad \Gamma' = \bigcup_{i=1}^{n_{el}} \Gamma^e, \tag{5.3}$$

where Ω', Γ' denote the interior and boundaries of the elements, respectively. In case of the continuous Galerkin (CG) method, the coarse space basis functions $\mathcal{V}_h \subset C^0 \cap L^2(\Omega)$ have C^0 continuity everywhere including element boundaries. In the case of discontinuous

Galerkin (DG) methods, the coarse space $\mathcal{V}_h$ is defined as:

$$\mathcal{V}_h \triangleq \left\{ u \in L_2(\Omega) : u|_T \in P^k(T), T \in \mathcal{T}_h \right\}, \tag{5.4}$$

where the space of polynomials up to degree k is denoted as P^k. Defining $\mathcal{V}_h$ in this manner allows for discontinuities in the solution across element boundaries. The DG space is a more richer space compared to CG space if the number of element and polynomial order is kept fixed. Irrespective of the choice of basis functions used (CG or DG), given u from the high-fidelity simulation, our goal is to find the optimal representation of u in the coarse sub-space $\mathcal{V}_h$. In our case, we will use the L^2-projection to obtain u_h which minimises the value of $||u - u_h||_2^2$. This problem is equivalent to the problem of finding $u_h \in \mathcal{V}_h$ such that

$$(u, w_h) = (u_h, w_h) \quad \forall w_h \in \mathcal{V}_h. \tag{5.5}$$

In the case of CG basis functions, the mass matrix is global and a large matrix needs to be inverted to obtain the final filtered solution. The DG mass matrix on the other hand is local to the element and lends itself to easy parallelization. In section 5.3, the a priori analysis for LES, WMLES and hybrid RANS-LES is performed using CG basis functions using $Re_\tau \approx 1000$ data. Depending on the application, projection on DG basis functions might be more efficient sometimes. For example, in section 5.4, we will use DG basis to obtain slip velocity estimates for WMLES for both $Re_\tau \approx 1000$ and $Re_\tau \approx 5200$. The slip velocity is obtained by evaluating the coarse-scale solution at the wall. As a result, for DG, the projection operation need to be only performed on those elements which share a face with the wall, leading to computational efficiency. Similar estimates can also be obtained using CG basis at a higher computational cost. This is challenging especially for the $Re_\tau \approx 5200$ case. The computation of 3-D projections even for the $Re_\tau \approx 1000$ case is expensive and has been approximated by performing a sequence of 1-D projections along the wall-normal, stream-wise and span-wise directions as shown in figure 5.2.

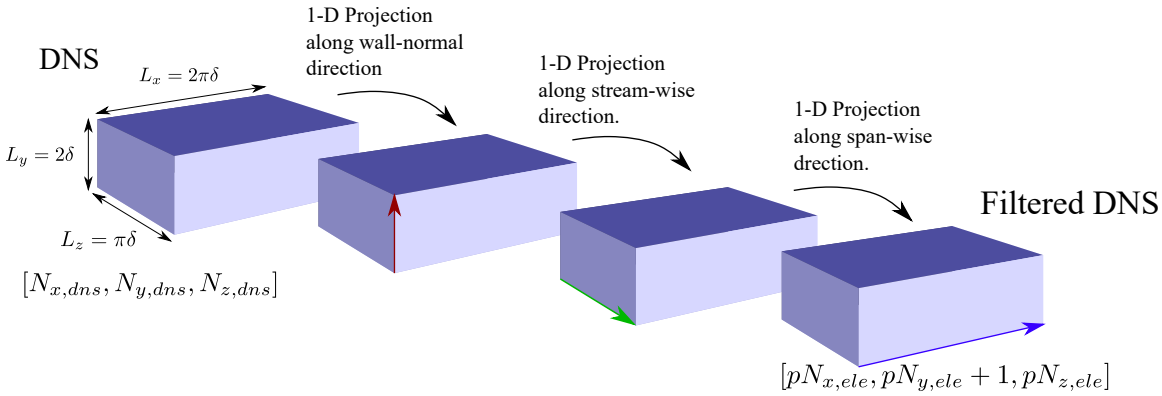


Figure 5.2: The projected solution is obtained by performing a sequence of 1-D projections along the wall-normal, stream-wise and span-wise directions.

The next step is the computation of inner products in equation 5.5. The right hand part of the equation is easy to compute because both u_h and w_h are finite dimensional and can be computed precisely using quadrature rules. The left hand side of equation 5.5 requires the computation of inner products of u and the coarse basis functions w_h. The computation of this integral needs special care because u is extremely high dimensional in comparison to w_h. The high-dimensionality of u is restricted by the size of DNS which exists on a very fine mesh capable of resolving the Kolmogorov scales O(η). To compute this term precisely, we interpolate the coarse-scale basis functions and the DNS solution on a very fine mesh of the size of O(η) and apply numerical integration to compute the inner products. The size of the numerical integration mesh is adjusted till the final projected solution is independent of the numerical integration mesh size. Additional details on the procedure to compute the L_2-procedure is given in Appendix C.

The final comment is on the imposition of the near-wall behavior of the coarse-space. There are two choices: (i.) project on a space which strongly satisfies the boundary condition at the nodal points, or (ii.) keep the boundary DOFs free and make no such assumptions. The second choice appears more reasonable because when the solution is coarse-grained in the wall-normal direction, the solution might no longer satisfy the boundary conditions strongly. This is especially true for WMLES where the coarse-solution no longer satisfies the boundary condition and slip is observed. However, as the grid is refined near the wall,

the no-slip boundary condition is naturally satisfied.

5.3 Application to channel flow.

5.3.1 Effect of projection

As a first step towards obtaining the projected DNS solution for the channel flow problem, we discuss the effect of the choice of the coarse-space basis functions on the coarse-scale solution obtained after the projection operation. After projection, the coarse space can only represent an optimal version of the DNS solution. However, depending on the coarse-space basis, the projected solution can be either a low-dimensional compressed representation of the original solution or a spatially filtered version of it. The low-dimensional compressed representation is obtained when the coarse basis is tailored using data (*Rowley et al.*, 2004; *Carlberg et al.*, 2017) or existing analytical solutions (*Krank and Wall*, 2016). To ensure that the projection step leads to a more general spatial filtering approach, non-tailored basis functions commonly used in the finite element method are used. The projection operation onto these coarse finite element grids will lead to filtering.

The resulting coarse solution after filtering might be considerably different from the DNS solution due to truncation of the high-frequency components present originally in the DNS solution. In the near-wall region, the effect of projection can vary with the size of the filter in each direction. One manifestation of under-resolution in the wall-normal direction is the occurrence of a slip velocity. This slip-velocity can - in fact - be tracked down to the mean-profile itself. To resolve the mean solution, a near-wall grid spacing of $\Delta y^+ \approx 1$ is required in the wall-normal direction. However, if a grid size of $\Delta y \approx 0.1\delta$ is used, even the mean solution can no longer be resolved and a slip velocity at the wall will be observed. This is true unless the solution is artificially forced to go from a large value to zero over just one grid point. A solution to make the coarse-scale solution satisfy the no-slip boundary condition is to enrich the coarse-space with a tailored basis (*Krank and Wall*, 2016). As a

consequence, the tailored basis mimics the mean profile between the wall and the first grid point and ensures that the no-slip is satisfied collectively by the coarse non-tailored basis and the enriched tailored basis.

5.3.2 Selection of the Coarse space.

The next step is to define the coarse space onto which the DNS will be projected. To define this coarse space, we first construct a finite element mesh and chose the polynomial order of the basis functions. The idea here is that by selecting the grid and the polynomial order of the basis functions, we are enforcing our desired filter size distribution. A variety of coarse spaces have been generated as shown in table 5.1. The 'A'-type grids are the DNS grids on which the high-resolution solution u exists. Two different DNS solutions at friction Reynolds numbers of $Re_\tau \approx 950$ and $Re_\tau \approx 1000$ are used and their corresponsing grids are marked as $A1$ and $A2$, respectively. The $Re_\tau \approx 950$ solution (*Hoyas and Jiménez*, 2006; *Del Alamo et al.*, 2004) is obtained from a relatively smaller domain having a stream-wise size of $L_x \approx 2\pi\delta$ and a span-wise sizes of $L_z \approx \pi\delta$, whereas, the $Re_\tau \approx 1000$ solution (*Li et al.*, 2008; *Perlman et al.*, 2007; *Lee and Moser*, 2015) is obtained as a cutout from a simulation performed on a larger domain.

The goal here is to not to make any distinctions between the different coarse-grained simulation methods (i.e. WRLES, WMLES, hybrid RANS-LES) and show that L_2- optimal solutions for all these approaches can be obtained by projecting the same DNS solution on different meshes appropriate for these methods. However, for the sake of convenience of the readers, we have divided these grids into four different types: 'B','C','D' and 'E'-type grids suitable for different types of coarse-grained methods. A representative grid for each type is shown in figure 5.3. The distinguishing characteristic between the various grids are the mesh sizes that are used in each direction and their relative scaling with respect to the inner and outer length scales.

As discussed earlier, the grids marked by the letter 'A' represent the DNS grid. These

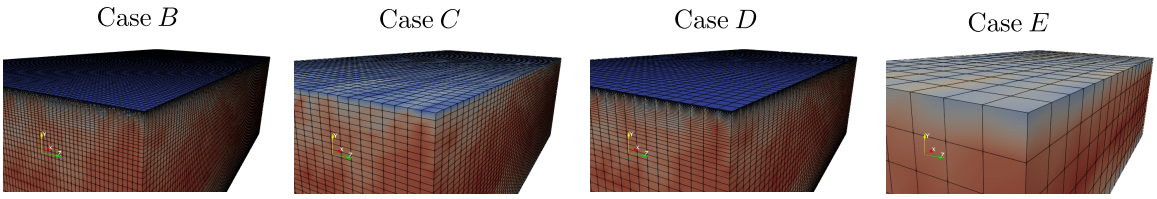


Figure 5.3: Near-wall grids used for cases B,C,D and E.

grids have a mesh size that scales with the wall units throughout the domain i.e. $\Delta x^+ \approx 10$, $\Delta y^+ \approx 0.1 - 10$ and $\Delta z^+ \approx 5$ (*Lee and Moser*, 2015). The 'B'-type grids, on the other hand, are tailored for performing wall-resolved LES, where, the near-wall mesh resolution still scales with the order of the wall units i.e $\Delta y^+ \approx 0.1 - 1$ and $\Delta x^+, \Delta z^+ \approx 20 - 50$. As a result of the size of the largest energy containing eddies scaling with the distance from the wall (*Yang and Griffin*, 2021) outside the viscous sub-layer, a mesh resolution of $\Delta y \approx 0.1\delta - 0.25\delta$ is used for the type-'B' grids at the center of the channel. Similarly, the 'C'-type grids are tailored for performing WMLES simulations using the wall-stress or the slip-wall based approaches. A mesh resolution of approximately $\Delta y \approx 0.1\delta$ is ideally required for this approach. The type-'D' grids are more suitable for assessing the WMLES branch of the hybrid RANS-LES methods (*Shur et al.*, 2008). For type-'D', the resolution in the stream-wise and the span-wise direction is similar to the 'C'-type grid. However, in the wall-normal direction, a grid spacing similar to type-'B' grid has been assumed i.e. $\Delta y^+ \approx 0.1 - 1$ in the near wall region and $\Delta y \approx 0.1\delta - 0.25\delta$ at the center of the channel. The type-'E' grid is an extremely coarse grid with resolutions of $\Delta x \approx 0.35\delta$,$\Delta y \approx 0.334\delta$ and $\Delta z \approx 0.35\delta$ in the stream-wise, span-wise and the wall-normal directions, respectively. For all the grid types, the mesh is uniform in the stream-wise and the span-wise directions. However, in the wall normal direction, the mesh has been stretched geometrically for cases 'B' and 'D'. In case of 'C' and 'E' type grids, uniform mesh is assumed in the wall normal directions as well. For each type of grid, two different polynomial orders $p = 1, 2$ are used to construct the projection coarse-space. The stretch rates (SR) and the polynomial orders for different cases have been summarised in table 5.1.

5.3.3 Results.

The DNS solutions are projected onto the coarse-spaces formed by the different grids and the L_2-optimal coarse-scale solutions are obtained using the procedure described in section 5.2. Figure 5.4 shows the filtered solutions (from A2) obtained after projection on meshes B2, C2 and D2, and plotted on a plane formed by the stream-wise and the wall-normal directions. Similarly, the projected solutions are also plotted on a plane formed by the span-wise and the wall-normal directions in figure 5.5. On closer observation, it can be seen in both figure 5.4 and figure 5.5 that the projection operation leads to a loss of fine-scale structure. The effect of filtering is more prominent in figure 5.5 compared to figure 5.4, suggesting a higher span-wise resolution requirement compared to the stream-wise direction. Figure 5.5 suggests that the fine scales near the wall for case B2 (wall-resolved LES) are well-represented in comparison to case C2 (WMLES) and case D2 (hybrid RANS-LES). However, at the center of the channel, C2 is slightly better in resolving the flow-features compared to B2 and D2 due to a finer mesh. Figure 5.6 shows the plot of the projected solution at the center of the channel i.e. $\frac{y}{\delta} \approx 1$ on a plane formed by the stream-wise and the span-wise directions. The main features of the flow at the center of the channel appear to be well-resolved in all the cases. These observations will be explored later through the energy spectra.

A plot of the sub-scales i.e. $u' = u - u_h$ for the cases B1, C1 and D1 is shown in figure 5.7 for the near-wall region. The sub-scale can be considered to be a measure of the error in the coarse-scale solution and hence, a indicator of under-resolution. The sub-scales for cases C1 (WMLES) and D1 (hybrid RANS-LES) was found to be be identical for $\frac{y}{\delta} > 0.1$. On close inspection, it can be observed that the sub-scales for cases B1 and D1, are close to zero at the wall i.e. $u' \approx 0$. However, for the case C, the sub-scales deviate significantly from zero at the wall suggesting the existence of a slip-velocity since the true solution precisely satisfies the no-slip boundary condition.

Figure 5.8 shows the mean and second-order statistics computed using the projected

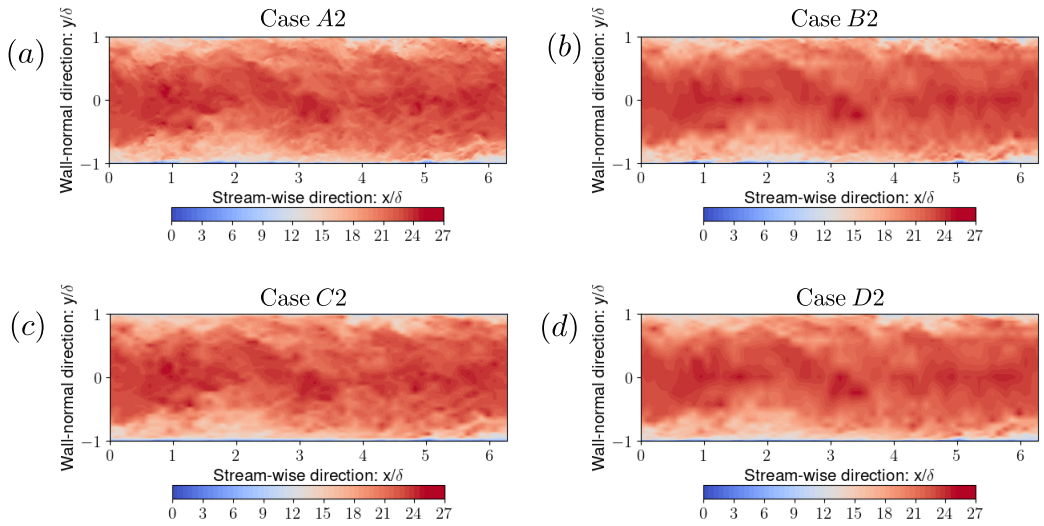


Figure 5.4: Contours of the stream-wise velocity plotted on a x-y plane for cases A2, B2,C2 and D2.

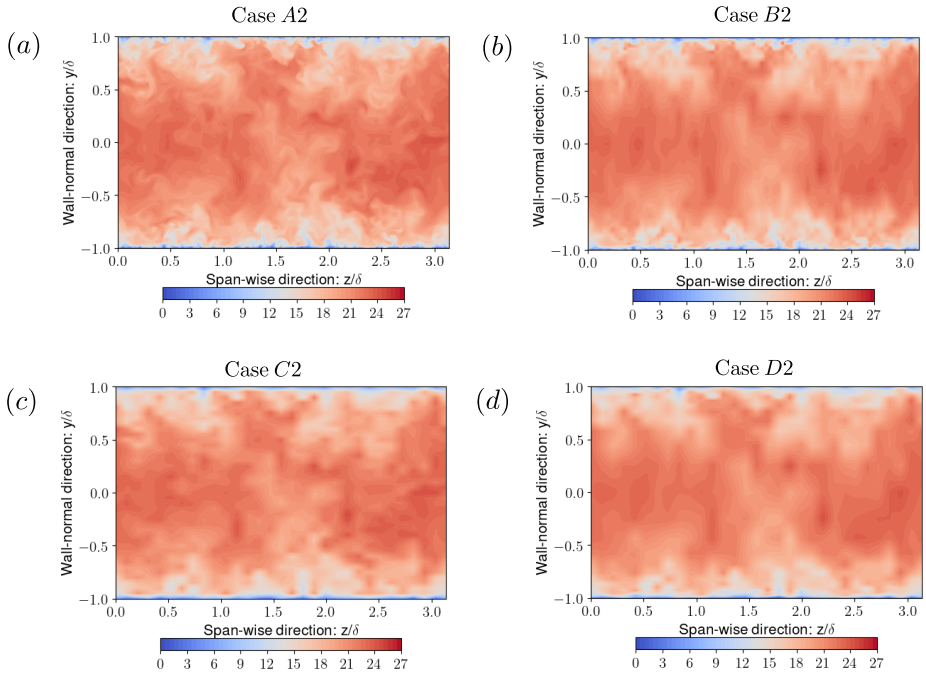


Figure 5.5: Contours of the stream-wise velocity plotted on a y-z plane for cases A2, B2,C2 and D2.

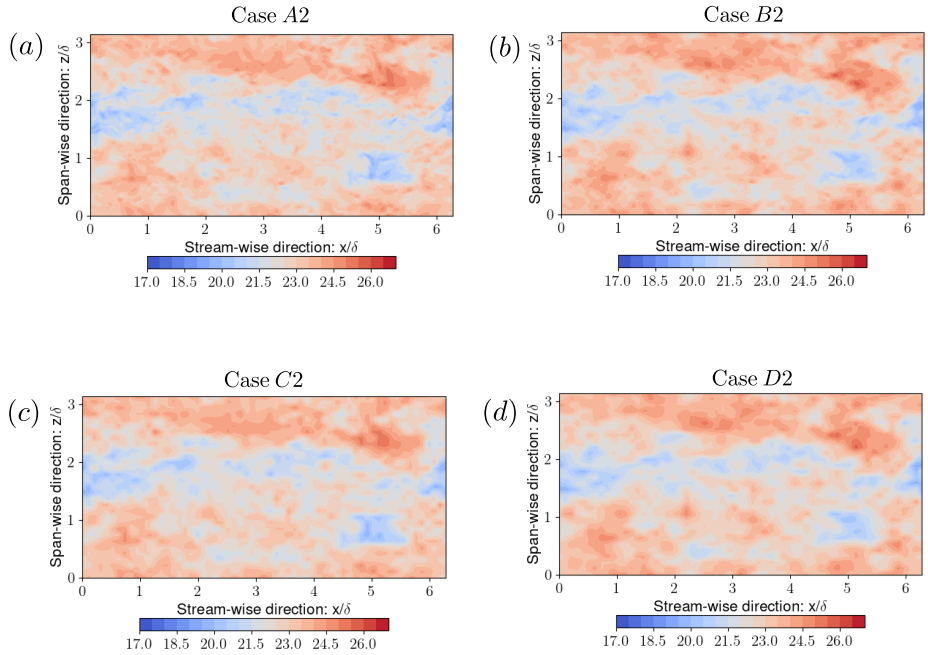


Figure 5.6: Contours of the stream-wise velocity plotted on a x-z plane for cases A2, B2,C2 and D2 at $y/\delta \approx 1$.

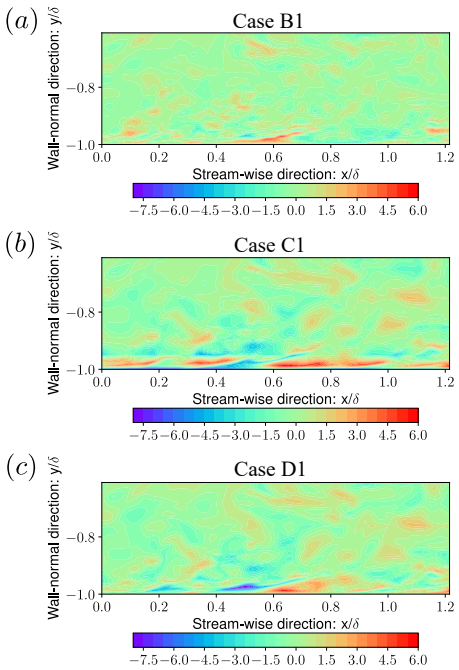


Figure 5.7: Contours of the stream-wise velocity sub-scales $u' = u - u_h$ for cases B1,C1 and D1.

solution for the cases B1,C1,D1 and E1. The goal is to compare the optimal solutions for different grids that correspond to different coarse-grained approaches, except for E1, which is an extremely coarse mesh and is not suitable for any existing method. From figures 5.8(a) and 5.8(b), it can be observed that the mean velocity is well-resolved for cases B1 and D1. For case C1, which represents an optimal WMLES solution, the mean velocity is well-resolved only after the first grid point i.e. $y/\delta > 0.05$. As can be observed in 5.8(b), Case E1 is extremely coarse and fails to resolve the mean velocity until the outer limit of the log-layer is reached. For these cases with wall-normal under-resolution, the effect of under-resolution results in slip velocity u_s at the wall. The magnitude of the slip velocity was found to increase with the under-resolution i.e. $< u_s >^+ \approx 6$ for cases C1 to $< u_s >^+ \approx 12$ for cases E1. It can also be observed in figures 5.8(c), 5.8(d) and 5.8(e) that all the methods except E1 resolve the second-order statistics outside the inner layer. Inside the near-wall region, only B1 is capable of accurately resolving the turbulence stresses. Among the second-order statistics, the effect of filtering is most strongly felt on the wall-normal fluctuations. It can be observed for both cases C1 and D1 that the wall-normal fluctuations far away from the inner layer are under-represented even when stream-wise and the span-wise fluctuations as close to the DNS solution.

Figure 5.9 shows the stream-wise velocity energy spectra of the projected solution in the span-wise and stream-wise directions for cases A2, B4, C2 and D4 at two different wall-normal locations. The choice of cases plotted here is based on the most commonly used mesh sizes for various methods. As can be seen from figures 5.9(a) and 5.9(b), the large scales are well-represented at the center of the channel ($\frac{y}{\delta} \approx 1.0$) both in the stream-wise and the span-wise directions, for all the methods. This was also inferred from figure 5.6. However, as can be seen in figure 5.9(c), the large scales are not represented accurately by the C2 (WMLES) and D4 (hybrid RANS-LES) cases in the near-wall region ($y^+ \approx 15$). However, in the span-wise direction (figure 5.9(d)), the large scales are relatively well represented in D4 in comparison to C2. In the next part, the effect of the projection order

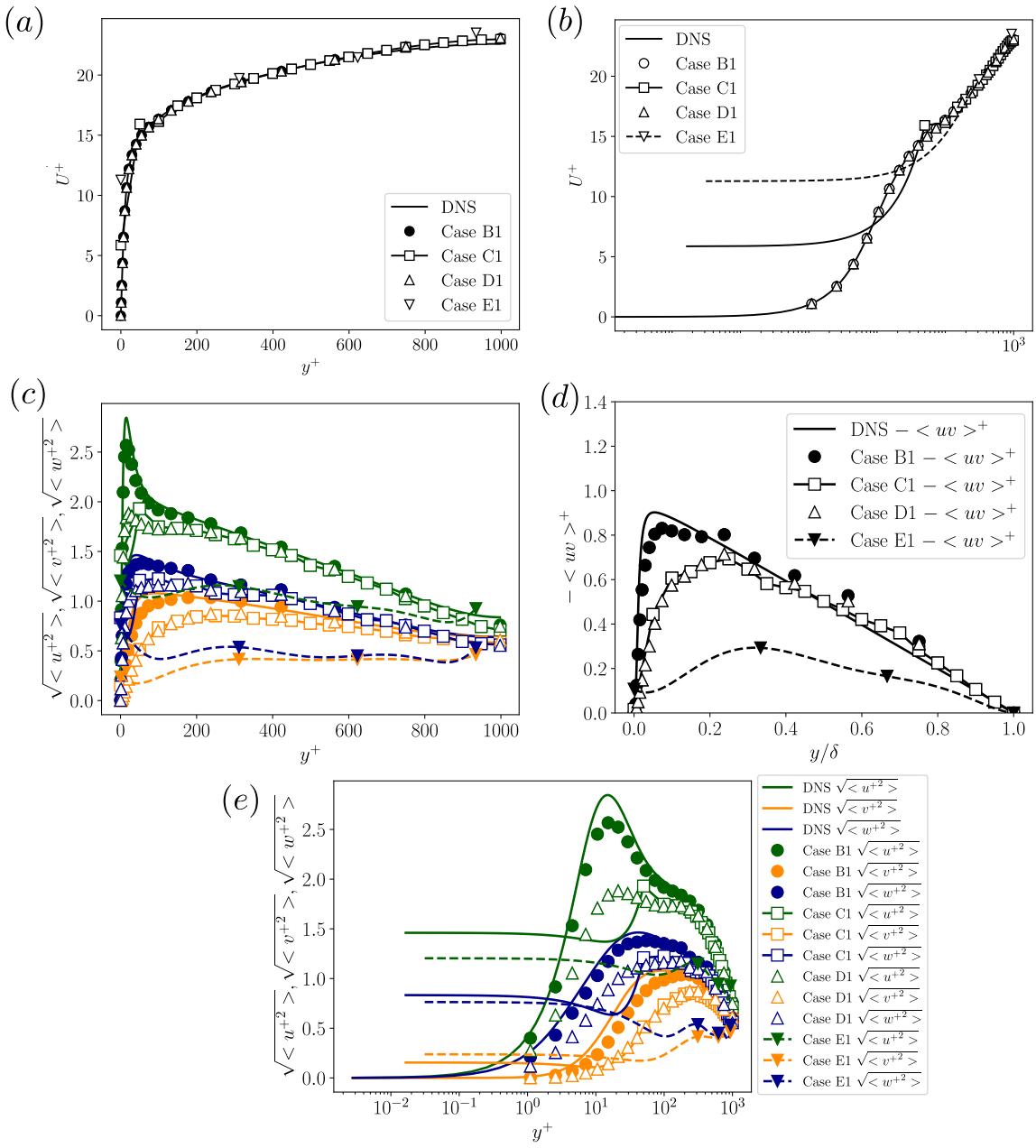


Figure 5.8: Comparison of mean and second-order statistics for cases B1,C1,D1 and E1. The symbols in all the plots correspond to the value at the nodal point. In sub-plot (b), the solutions for cases C1 (WMLES) and E1 (extremely coarse), are interpolated to the DNS mesh using the coarse finite element basis functions to show the slip velocity. Identical symbols are used for the sub-plots (c) and (e).

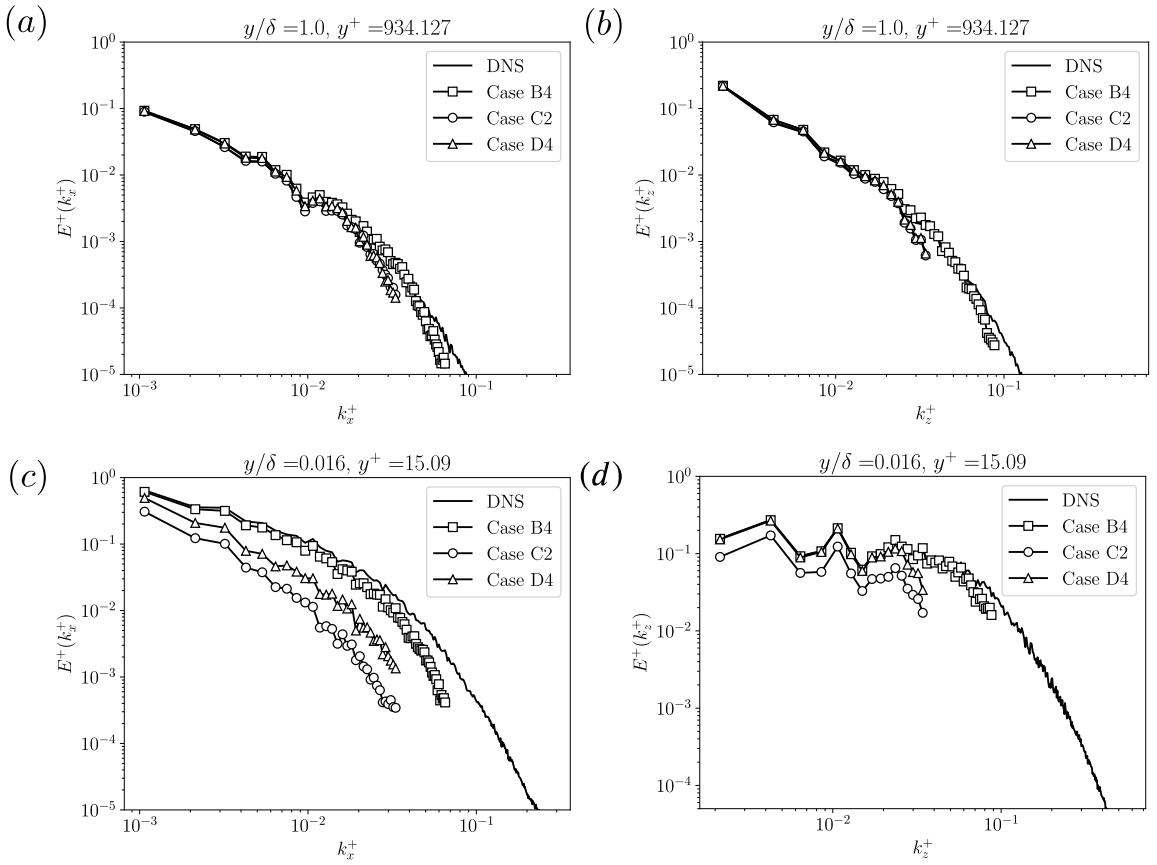


Figure 5.9: Comparison of the stream-wise velocity energy spectra at the near-wall region and the center of channel for cases A2,B4,C2 and D4.

and the stretching ratio will be discussed individually for each type of mesh.

The wall-resolved LES limit In this subsection, we will compare the cases B2, B3 and B4, which is representative of the typical grids for performing wall-resolved LES. The effective stream-wise and the span-wise resolution for all the three cases are approximately equal to $\Delta x_e^+ \approx 36.71$ and $\Delta z_e^+ \approx 36.71$, sufficient enough to resolve the near-wall energy containing scales. A very high stretch ratio of 1.33 is used to discretize the elements in the wall-normal direction for grids B2 and B3. However, for the case B3 we use a $p = 2$ basis function with a uniform node distribution inside each element, resulting in an effective lower stretch rate. The grid B4 is stretched using a stretch ratio of 1.1, which is a more acceptable value. Since, the span-wise and stream-wise are almost identical, the main distinguishing factor between the three grids is the order of polynomial used and the nodal point distribution in

Case	Re_τ	$N_x \times N_y \times N_z$	p	SR	Δ_x^+	Δ_x/δ	Δ_y^+	Δ_y/δ	Δ_z^+	Δ_z/δ
A1	1000	$512 \times 512 \times 512$	Sp.	-	12.24	0.0122	0.016 – 6.14	1.65E-5 – 0.006	6.12	0.006
A2	950	$512 \times 384 \times 512$	Sp.	-	11.47	0.0122	0.031 – 7.64	3.35E-5 – 0.008	5.73	0.006
B1	1000	$121 \times 41 \times 81$	1	1.33	39.11	0.0391	1.10 – 248.45	0.001 – 0.248	39.11	0.039
B2	950	$121 \times 41 \times 81$	1	1.33	36.71	0.0393	1.03 – 232.55	0.001 – 0.248	36.71	0.039
B3	950	$121 \times 81 \times 81$	2	1.33	36.71	0.0393	0.51 – 116.27	5.54E-4 – 0.124	36.71	0.039
B4	950	$121 \times 97 \times 81$	1	1.10	36.71	0.0393	0.97 – 85.80	0.001 – 0.091	36.71	0.039
C1	1000	$61 \times 41 \times 31$	1	1.00	104.3	0.1045	49.90	0.05	104.3	0.1045
C2	950	$61 \times 41 \times 31$	1	1.00	97.91	0.1048	46.7	0.05	97.91	0.1048
C3	950	$61 \times 81 \times 31$	2	1.00	97.91	0.1048	23.3	0.025	97.91	0.1048
C4	950	$61 \times 97 \times 31$	1	1.00	97.91	0.1048	19.46	0.021	97.91	0.1048
D1	1000	$61 \times 41 \times 31$	1	1.33	39.11	0.1045	1.10 – 248.45	0.001 – 0.248	104.3	0.1045
D2	950	$61 \times 41 \times 31$	1	1.33	97.91	0.1048	1.03 – 232.55	0.001 – 0.248	97.91	0.1048
D3	950	$61 \times 81 \times 31$	2	1.33	97.91	0.1048	0.51 – 116.27	5.54E-4 – 0.124	97.91	0.1048
D4	950	$61 \times 97 \times 31$	1	1.10	97.91	0.1048	0.97 – 85.80	0.001 – 0.091	97.91	0.1048
E1	1000	$19 \times 7 \times 10$	3	1.00	326.3	0.3500	311.27	0.334	326.3	0.35

Table 5.1: Summary of mesh parameters. Here, Δ_x^+, Δ_y^+ and Δ_z^+ are the effective grid sizes in different directions Δ_x, Δ_y and Δ_z normalised with wall units, δ is the half channel height, N_x, N_x and N_z represents the number of degrees of freedom in the stream-wise, wall-normal and span-wise directions respectively, p is order of polynomial used, SR is the stretching ratio used to generate the grid. The effective grid sizes Δ_x, Δ_y and Δ_z for the finite element grid are defined as $\Delta_x = \Delta_x^e/p$, Δ_y^e/p and Δ_z^e/p respectively. The quantities Δ_x^e, Δ_y^e and Δ_z^e represent the actual element sizes in the finite element mesh.

the wall-normal direction. The number of degrees of freedom for B3 and B4 in the wall-normal direction is almost twice the number of grid points in the wall-normal direction. Coarse-scale statistics for B2, B3 and B4 are provided in figure 5.10. It can be observed in figure 5.10(a) that all the cases perform well in resolving the mean profile. Similar trends are observed in figures 5.10(b) and 5.10(c), where the cases B3 and B4 only slightly outperform the case B2 in resolving the second-order statistics. This suggests that the sensitivity of the coarse-scale statistics to the wall-normal stretch rate is not as high as the sensitivity to the mesh resolution in the stream-wise and the span-wise directions.

The WMLES limit In this subsection, we will compare the results for cases C2, C3 and C4, which represent the typical grids used for performing wall-stress/slip-wall based WMLES. The effective stream-wise and the span-wise resolution are approximately equal to $\Delta x_e/\delta \approx 0.1$ and $\Delta z_e/\delta \approx 0.1$ for the all the cases i.e. C2, C3 and C4. These resolutions are sufficient to resolve the large energy containing scales outside the inner layer. Again, for the case C3, $p = 2$ basis functions are used in place of $p = 1$ for C2 and C4. Uniform

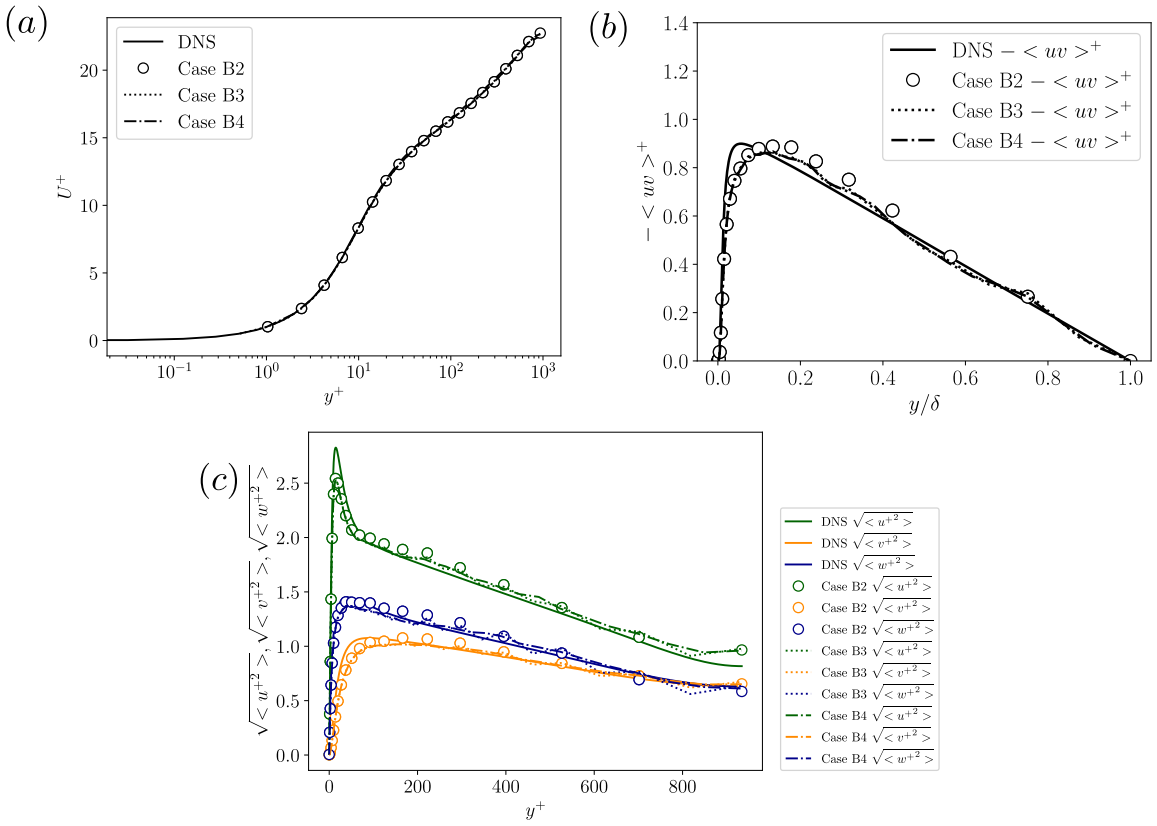


Figure 5.10: Comparison of mean and second-order statistics for wall-resolved cases B2,B3 and B4

mesh elements have been used in the wall normal direction for all the cases. This leads to a wall-normal resolutions of $\Delta y_e/\delta \approx 0.05$, $\Delta y_e/\delta \approx 0.025$ and $\Delta y_e/\delta \approx 0.021$ for cases C2, C3 and C4, respectively. Clearly, these mesh resolutions in wall units are much larger than unity and cannot resolve the mean velocity profile accurately near the wall. As a consequence, a slip velocity u_s is observed for all the cases after the projection step as shown in figure 5.12. It can also be observed from the figure 5.12 that the magnitude of slip velocity is highest for the case with the maximum wall-normal under-resolution i.e. C2. This magnitude goes down as the wall-normal mesh is refined from C2 to C3 or C2 to C4. Similar observations can also be made from the mean velocity profiles plotted in figures 5.11(a) and 5.11(b). Apart from mean velocity profiles, second-order statistics are plotted in figures 5.11(c), 5.11(d) and 5.11(e). As expected, the second-order coarse-scale statistics are only accurate outside the inner layer towards the center of the channel. It can also be observed that the stream-wise and span-wise velocity fluctuations computed using the coarse solution do not go to zero near the wall. The wall-normal velocity fluctuations, however, go to zero at the wall. This near-wall behavior of the coarse-scales is consistent with existing WMLES simulations in the literature (*Wang et al.*, 2020a; *Carton de Wiart and Murman*, 2017; *Kawai and Larsson*, 2012).

The Hybrid RANS-LES limit In this part, we will present results for cases D2, D3 and D4, which represent the grids for hybrid RANS-LES (HRLES) methods such as IDDES (*Shur et al.*, 2008). Since our discussion here is only restricted to the channel flow problem, we are only assessing the 'WMLES' branch of the hybrid RANS-LES methods. The grids used for cases D2, D3 and D4 in the wall-normal direction are identical to the grids used for performing the wall resolved cases i.e. B2, B3 and B4, respectively. The difference between the two types of grids is the mesh resolution in the stream-wise and the span-wise directions. In the stream-wise and the span-wise directions, the 'D'-type grids have a grid-spacing that is identical to the 'C'-type grids. As a result. the energy containing near-wall structures can no longer be resolved and the turbulent stresses will be under-represented . However, the

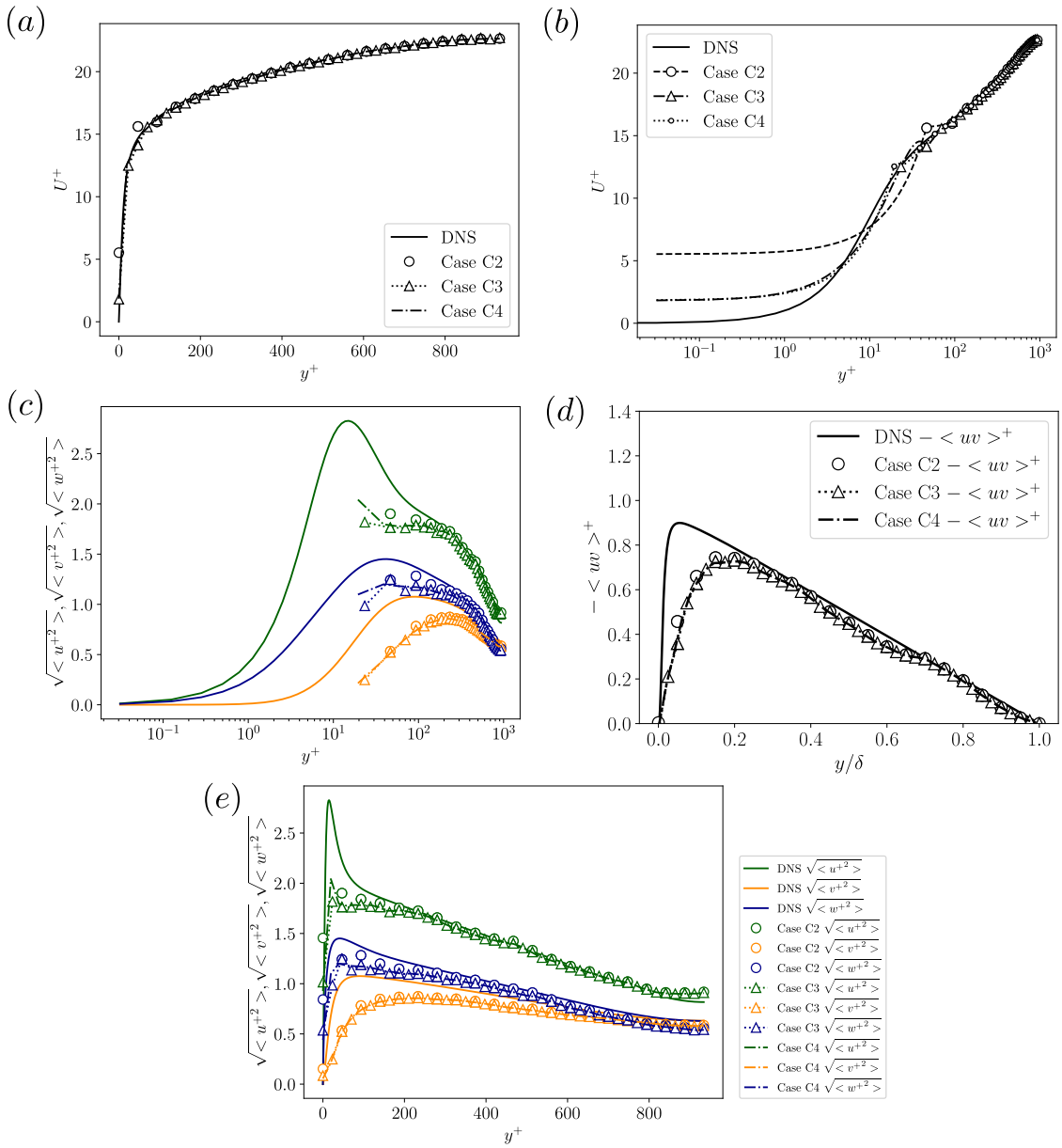


Figure 5.11: Comparison of mean and second-order statistics for WMLES cases C2,C3 and C4. In sub-plot (b), the solutions for all the cases are interpolated to the DNS mesh by using the coarse finite element basis functions to show the slip velocity.

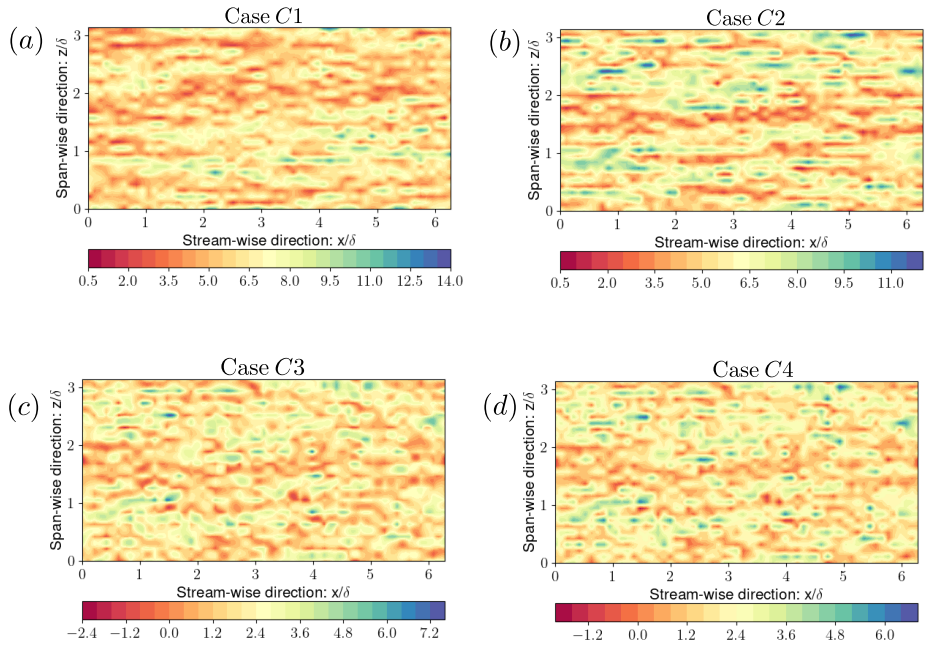


Figure 5.12: Contours of the stream-wise slip velocity at the bottom wall for the cases C1, C2,C3 and C4 which have insufficient resolution in the the wall-normal direction.

grid is still capable of resolving the mean velocity profile. Coarse-scale statistics for D2, D3 and D4 are provided in figure 5.13. It can be observed in figures 5.13(a) and 5.13(b) that for all the cases the mean profile was resolved accurately. In addition to the mean velocity profile, figures 5.13(c),5.13(d) and 5.13(e) also show the second-order statistics. If we compare the second-order statistic obtained for the present case to that obtained for the 'C'-type grids in figure 5.11 in the region that is outside of $10-20\%$ of the boundary layer, they are almost identical. However, near the wall, unlike the 'C'-type WMLES cases, all the velocity fluctuations go to zero due to the no-slip condition being satisfied by the hybrid RANS-LES cases. These observations are consistent with previous results from the literature (*Friess and Davidson*, 2020).

In this section, we have used the labels LES, WMLES and HRLES above to distinguish between the various methods. However, the difference between the later two is subtle. The traditional WMLES method essentially uses RANS knowledge to compute the wall-stress at the wall and can also be called a hybrid RANS-LES approach. Similarly, the HRLES

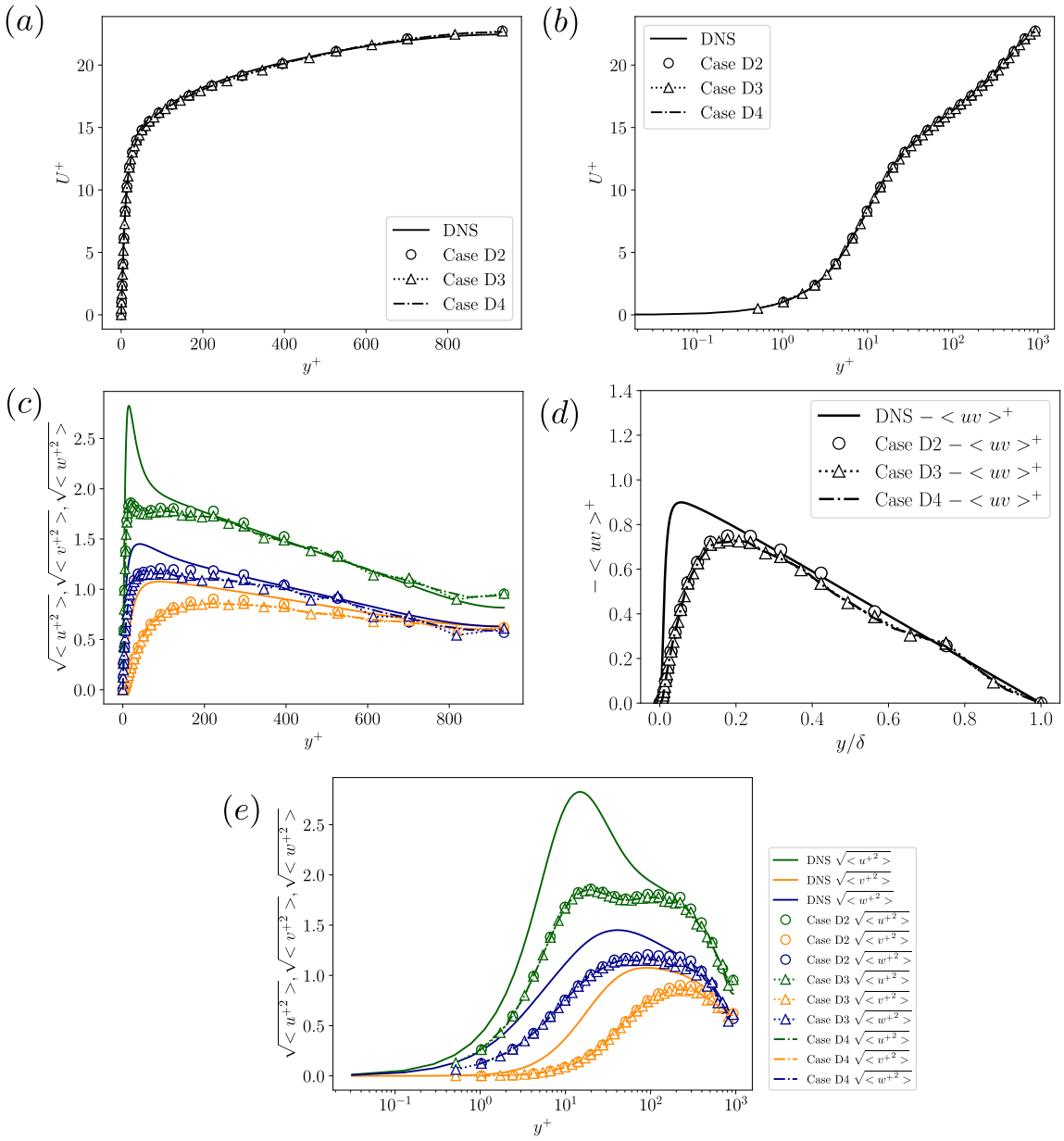


Figure 5.13: Comparison of mean and second-order statistics for HRLES cases D2,D3 and D4

approaches (such as IDDES) by virtue of solving the RANS equations near the wall reduce the computational cost associated with resolving the wall and can also be considered a WMLES. However, the context in which the labels WMLES and HRLES have been used in this chapter is based on whether these models integrate to the wall or not in a single domain, in other words if the size of the filter in the wall-normal direction is large or not in wall-units.

It is worthwhile to mention that the results presented in this section appear to be more accurate in comparison to those in the literature. For instance, a large stretching ratio of 1.33 has been used for some of the meshes which do not induce a significant error in the filtered solution, however, this stretch rate is more than the suggested limit for many methods. Additionally, no log-layer mismatch (LLM) was obtained in any of these cases. The resolution considered here for WMLES and hybrid RANS/LES is of the order of 10 points per semi-channel height δ in the streamwise and the spanwise directions, which is coarse compared with the guidelines for these approaches. A comparison between the optimal solution for the C3 case and a WMLES solution using the traditional wall-stress-based approach computed on the same grid is presented in figure 5.14. The mean solution for both cases begins to deviate from the DNS at similar locations. The resolved turbulent shear stress is under-represented near the wall, starting almost identically, however, differing in their peaks. Similarly, the wall-normal velocity fluctuations are almost identical. On the other hand, the velocity inside the first element is slightly under-predicted in the traditional WMLES approach in comparison to projected DNS. Additionally, both the stream-wise and span-wise velocity fluctuations reveal an overshoot near the second off-wall grid point in the traditional WMLES method.

One possible reason for the discrepancies between the results from the true simulation and the filtered DNS is the lack of accurate closures. With a poor model, the LLM may persist until a DNS-like resolution is reached. This is - in principle - similar to attributing inaccuracies in a wall resolved LES with a standard Smagorinsky model when a dynamic

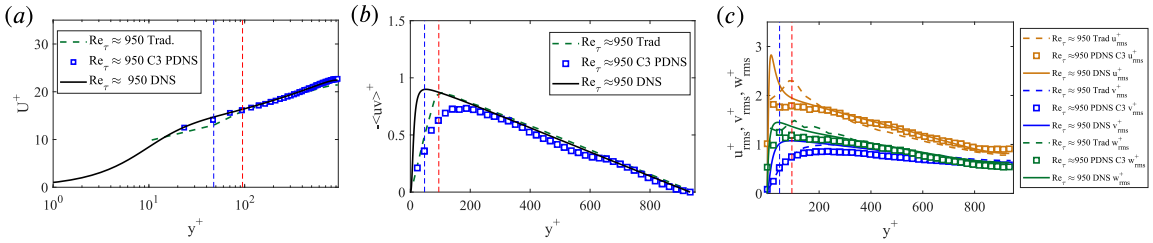


Figure 5.14: Comparison of projected DNS and solution from the traditional WMLES method at similar resolutions.

Smagorinsky model might yield near-optimal performance. However, using the optimal projection framework presented here, it is now possible to perform an analysis of the closure terms and evaluate modeling errors. By reducing the modeling errors, the goal is to force the solution to reach a near-optimal state. Ideally, we would have wanted to improve all three approaches using our optimal projection framework. However, to have a compact presentation, we only consider evaluating the modeling errors in the slip-wall-based WMLES models and improve its a posteriori performance.

5.4 Analysis of slip-based wall models.

The slip-velocity at the wall in WMLES is related to under-resolution in the wall-normal direction. In this section, we seek to quantify this slip-velocity to ensure that the resulting model generalises well to different Reynolds numbers. To understand the Reynolds number dependence, DNS from two different friction Reynolds numbers of $Re_\tau \approx 1000$ and $Re_\tau \approx 5200$ are used. As mentioned earlier, the computation of the 3-D projection of the $Re_\tau \approx 5200$ case by sequential 1-D projections in the wall-normal, stream-wise and the span-wise directions is computationally expensive. To ensure computational efficiency and utilizing the fact that this is a near-wall phenomenon, we project the DNS solution on uniform elements of size Δ with polynomial basis functions as shown in figure 5.15. This is equivalent to performing a full 3-D projection on a DG finite element solution space. The element shares its bottom face with the wall of the channel to mimic a near-wall grid.

It can be further observed that by moving the position of this element on the wall surface, different realizations of the slip velocity and coarse solution gradients in the wall normal direction can be obtained. This is possible due to the statistical homogeneity present in the stream-wise and span-wise directions.

For each realization, the projection of the DNS solution on the finite dimensional DG space leads to filtering of the DNS solution as shown in figure 5.16. Figure 5.16(a) shows the contour of the DNS solution of the stream-wise velocity component for a sample 3-D element. The projected DNS solution for the same element is shown in figure 5.16(b) shows t. The projected DNS solution does not satisfy the no-slip boundary condition at the wall and does not contain the fine-scale information present in the original DNS solution. The goal is to access the slip-wall based wall model proposed by *Bose and Moin* (2014):

$$u_s = C_w \Delta \frac{\partial u_h}{\partial n}, \tag{5.6}$$

and obtain an estimate of the model coefficient C_w. However, this requires the computation of u_s and the pre-multiplied gradient $\Delta \frac{\partial u_h}{\partial n}$. The slip-velocity u_s is obtained by evaluating the coarse-scale solution at the wall as shown in the figure 5.16(c). The pre-multiplied gradient $\Delta \frac{\partial u_h}{\partial n}$ is obtained by computing the derivatives of the coarse-scale in the wall-normal direction and multiplying with the normalized resolution $\Delta = \frac{\Delta_e}{p}$ as shown in figure 5.16(d). However, this results in not only an over-determined system for C_w but the value of C_w is linked to the filter size Δ used for the projection operation. The problem of this system being over-determined is solved by performing a least-squares minimization over many such realizations till convergence in the estimates of C_w was obtained. To solve the problem of the model coefficient C_w being dependent on the filter size Δ, we perform dimensional analysis. Other parameters that could affect C_w are a) the order of polynomial used for projection p, the viscosity ν; and b) the wall stress τ_w. After non-dimensionalization, the

following model form for C_w can be obtained:

$$C_w = g_p(\Delta^+), \tag{5.7}$$

where g_p is a function of the grid resolution normalized with wall units Δ^+ and the subscript p denotes the coarse space polynomial order used for projection. The parameter Δ^+ can be considered to be an indicator of the near-wall grid resolution. Similarly, the order of the numerical method can be encoded in p. Higher p implies that a more accurate numerical method has been used to compute the LES solution. However, this implies that for every polynomial order p we have to learn a new function. In addition to this, the numerical methods used to perform LES might work sub-optimally and the exact order might not be preserved. Hence, it is necessary that the effect of the numerical method be parameterized through a model constant similar to the Smagorinsky model coefficient C_s.

As discussed in the previous section, the inability to resolve the mean profile near the wall causes a slip velocity. Before investigating the slip velocity due to the full 3-D projection of the DNS solution, it is important to consider the contribution from the mean profile itself. As a first step, we will apply 1-D projection to the Reichardt profile which describes the mean profile in the inner layer. Since the mean solution is invariant in the stream-wise and span-wise direction for a channel, the 3-D projection is reduced to a 1-D projection in the wall-normal direction only. Figure 5.17(a) shows the estimates for C_w obtained from the Reichardt profile for different orders of projection. It can be observed that the C_w profiles for different orders are distinct even after normalisation of the element size Δ_e by p to obtain $\Delta = \frac{\Delta_e}{p}$. Inspired by the Smagorinsky model, which consists of a model constant C_s that pre-multiplies the grid-size in the final model form, we introduce a new model constant λ which in its inverted form i.e. $\frac{1}{\lambda}$ pre-multiplies the grid-size in our proposed model. Figure 5.17(b) shows the estimated valued for $C_{w,\lambda}$ for different polynomial orders along with the λ values for which all the curves collapse to the $p = 1$ curve with $\lambda = 1$. As a result, it is

possible to learn just one curve and parameterize it with an additional factor λ to obtain the C_w curves for different cases i.e.

$$C_w = C_{w,\lambda}/\lambda = g_1(\frac{\Delta^+}{\lambda})/\lambda, \tag{5.8}$$

where $C_{w,\lambda} = \frac{\lambda p u_s}{\Delta_e \frac{\partial u_h}{\partial n}}$. It is also important to check if similar relations also hold true for the 3-D projected solution.

Figure 5.18 compares the C_w curves obtained through the 3-D projection of DNS solutions to that obtained using the Riechardt mean profile for two different projection orders $p = 1, 3$. For each projection order, results for two different friction Reynolds numbers of $Re_\tau \approx 1000$ and $Re_\tau \approx 5200$ are plotted. To obtain these curves, large variations in the element sizes have been considered. We use element sizes with $\Delta_e \approx 0.011\delta - 0.28\delta$ for projecting the $Re_\tau \approx 1000$ data and element sizes with $\Delta_e \approx 0.033\delta - 0.30\delta$ for projecting the $Re_\tau \approx 5200$ data. The effective filter sizes corresponding to these grids can be approximated by normalizing the element size with p to obtain $\Delta = \frac{\Delta_e}{p}$ The results indicate that for the individual polynomial orders, the C_w estimates for different Re_τ at a particular Δ^+ are same, suggesting that C_w is a universal function of Δ^+. Further, the C_w values obtained through 1-D projections of the mean profile are already good approximations to that obtained though the 3-D projections of the DNS solution at moderate resolutions. However, at higher Δ^+, there appears to be a minor discrepancy between the two profiles in the form of a constant shift. The C_w for the span-wise velocity component is found to be negative and has a slight Δ^+ dependence. Similar to the stream-wise velocity, the C_w curves for different Re_τ suggest a Δ^+ dependence in the span-wise direction as well. The C_w in the wall normal direction is approximately zero and does not depend on the mesh resolution. This suggests that the wall normal slip can be set to zero without the loss of any generalisability. Finally, to obtain a single model form that works for different projection orders, we re-introduce the λ factor. As shown in figure 5.19, by re-using the λ values from figure 5.17(b), similar collapse in

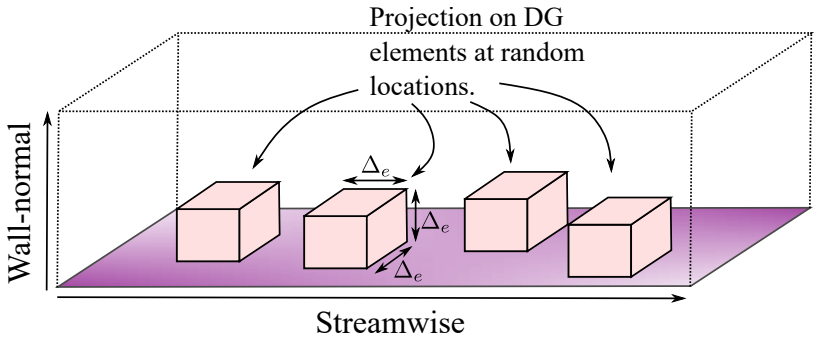


Figure 5.15: Near-wall domains located at random locations are used for projection of DNS solution on the discontinuous finite element basis functions.

the $C_{w,\lambda}$ curves was also obtained for the 3-D projection cases for the stream-wise and the span-wise velocity components. The C_w values in the wall-normal direction do not need similar scaling because they are approximately zero irrespective of the mesh resolution.

5.5 Towards accurate slip-wall models.

While the state-of-the-art dynamic slip-wall model by *Bae et al.* (2019) is found to be better in comparison to the case with no wall-model, it is found to be lacking in accuracy when compared to the traditional WMLES approach. In addition, the slip-wall model has been reported to suffer from instability issues when used with certain high-order methods *Carton de Wiart and Murman* (2017). Thus, there is a need to improve the stability and performance of slip-wall models on canonical turbulent flow problems before it can be confidently used in more complex flows. Indeed, it is recognized that one disadvantage of the traditional approach is that unlike the dynamic slip-wall model, it requires a priori specification of tunable coefficients. The authors are of the opinion, however, that tunable coefficients should not be used as a reason to replace the traditional WMLES approach which has been shown to perform well across a wider range of problems. To this end, we try to use data from existing WMLES simulations and our optimal projection techniques to improve the performance of existing slip-wall models to the level of traditional WMLES for the channel flow problem.

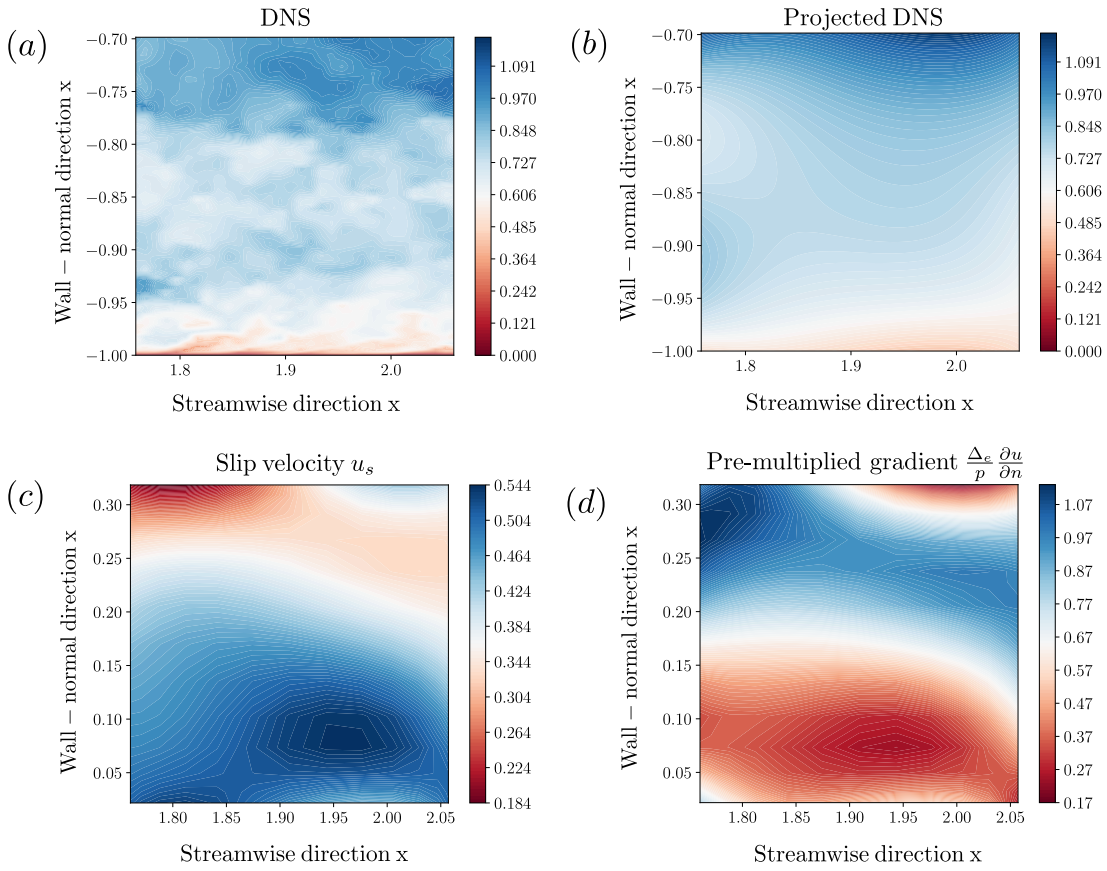


Figure 5.16: 3-D Projection of near-wall $Re_\tau \approx 5200$ channel data on finite element basis functions.

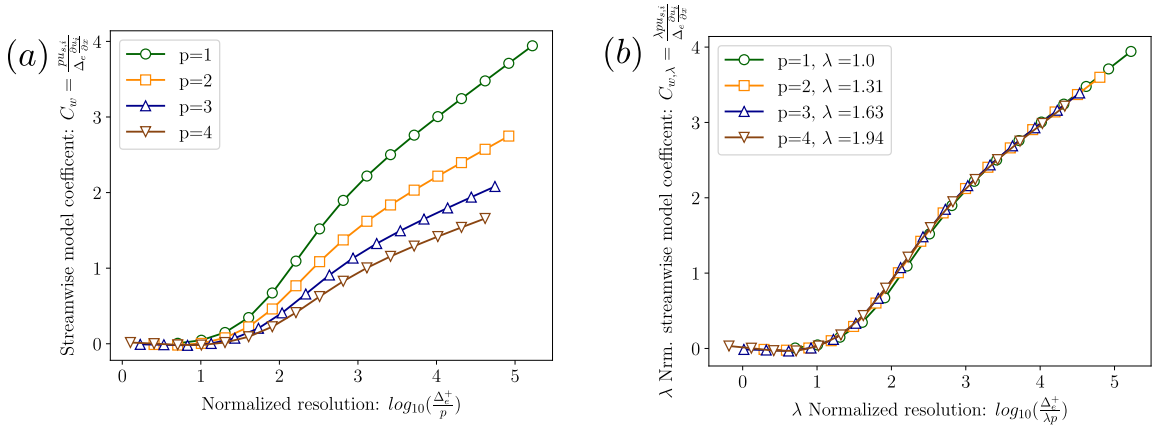


Figure 5.17: C_w computed using Reichardt profile by projecting on different polynomial basis.

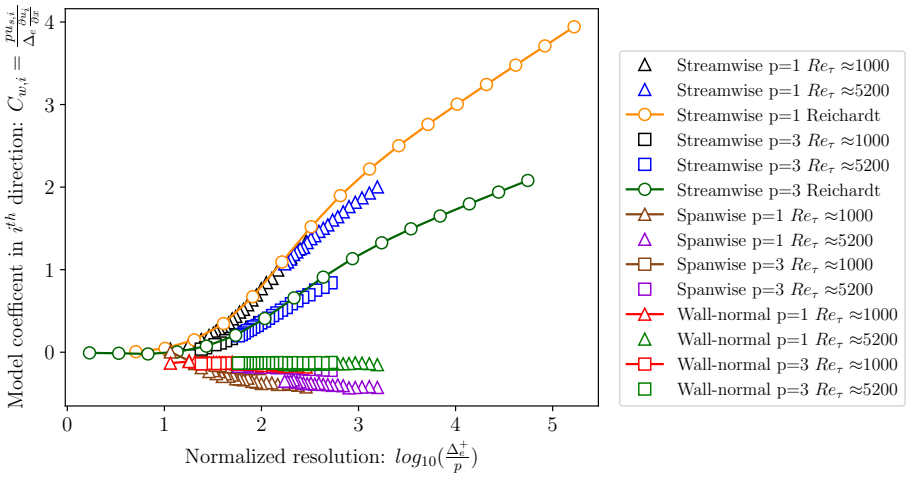


Figure 5.18: C_w computed by 3D projection of DNS on different polynomial spaces compared to 1-D projection of Reichardt profile.

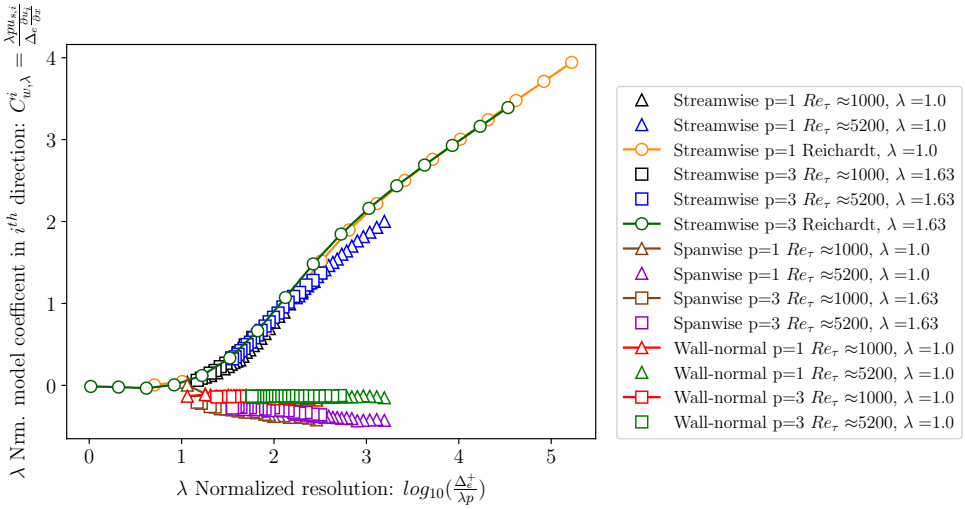


Figure 5.19: λ-normalized C_w computed by 3D projection of DNS on different polynomial spaces compared to 1-D projection of Reichardt profile.

As we observed in section 5.3, if the grid resolution is sufficiently coarse, a slip velocity is present at the wall. Hence, it is expected that even the solution from the traditional wall-stress based WMLES will have a slip-velocity at the wall. Given the excellent performance of the traditional WMLES approach for the channel flow problem, it is also expected that the universal relationship given in figure 5.17 should also hold true for the traditional WMLES approach.

Traditional WMLES solutions were computed using a DG solver with $p = 3$ discretization on different meshes using two different sub-grid models: (i.) a constant coefficient Smagorinsky model with $C_s = 0.12$; (ii.) *Vreman* (2004) model. Figure 5.20 shows the comparison of $C_{w,\lambda}$ computed using 1-D projection of the Reichardt profile to that computed using the solutions obtained using the traditional WMLES approach. To compute C_w for a traditional WMLES solution, the solution and its wall normal-gradients are evaluated at the wall to obtain the slip velocity and the pre-multiplied wall-normal gradient. Finally, a least-square fit is performed to obtain a single value of C_w. While computing C_w, the size of the element Δ_e is required. However, for all the traditional wall-model cases, the size of the element varies in each direction unlike the grids used for projection of DNS. As a first attempt, Δ_e is taken to be the size of the element in the wall-normal direction. Finally, an optimal value of λ is found such that the curves collapse asymptotically. By changing the value of λ only the slope of the asymptotic part of the $C_{w,\lambda}$ curve can be changed. However, when the slope of the $C_{w,\lambda}$ curve in the asymptotic part was made parallel to the $C_{w,\lambda}$ curve obtained for the Reichardt profile by projecting on the $p = 1$ basis functions, the intercepts were also found to match. This can be seen in figure 5.20 where the profiles appear identical at large resolutions. However, small discrepancies exist near the lower resolution limit (i.e. the wall-resolved LES limit), suggesting that either the traditional WMLES approach is inaccurate or the sub-grid model is not accurate. Figure 5.20 also suggests that a universal slip-wall model form exists irrespective of the sub-grid model or the numerical scheme as long as λ is known .

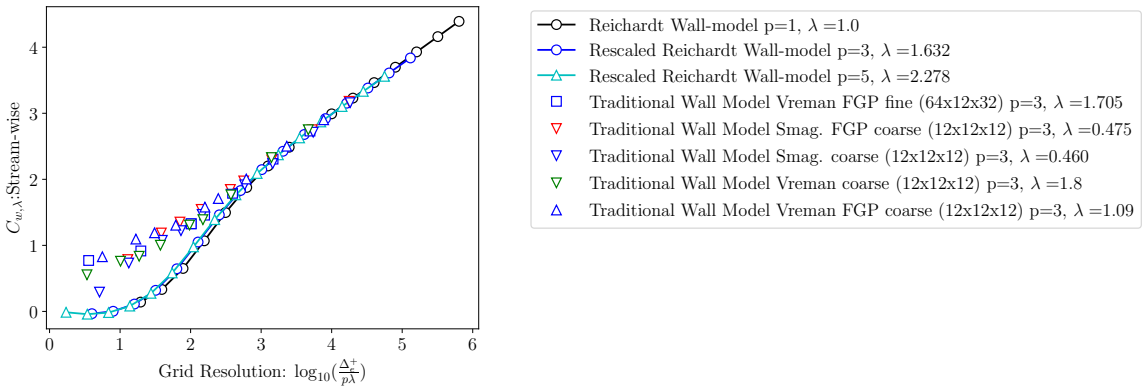


Figure 5.20: λ-normalized C_w vs. normalized grid-size $\frac{\Delta^+_e}{p\lambda}$. The λ-normalized C_w are computed by 1-D projection of Reichardt profile and compared to the same obtained using the traditional WMLES solution. The plots marked by "FGP" use only the explicit sub-grid models inside the first element and gradually change to implicit LES outside the first element.

Even if λ is known prior to the simulation or is dynamically determined, a model for $C_{w,\lambda}$ which takes as input the normalized grid-size $\frac{\Delta^+_e}{p\lambda}$ is not useful. This is because the value of Δ^+ is not known unless the wall stress is also known. One option is to use the traditional wall model to obtain the friction velocity u_τ to compute Δ^+ *Whitmore et al.* (2021). A better choice would be to represent the slip-wall model coefficient $C_{w,\lambda}$ as a function of the mean slip-velocity $< u_s >$ based Reynolds number i.e. $Re_{slip} = \frac{<u_s>\Delta}{p\lambda\nu}$, as a consequence of which the wall-stress will no longer be required to predict $C_{w,\lambda}$. Figure 5.21 shows the λ-normalized slip-wall model coefficient $C_{w,\lambda}$ as a function of the slip-velocity (mean stream-wise) based Reynolds number. As can be observed in figure 5.21, a universality in the model form similar to the curves in figure 5.20 also exists in the case when the slip-Reynolds number is used as a feature in place of the normalized grid-size. In addition, the curves were found to collapse to the $p = 1$ Reichardt curve for exactly the same value of λ used in the case of $C_{w,\lambda}$ vs. Δ^+. In addition to the plots for $C_{w,\lambda}$ for the various traditional approach obtained using various sub-grid models, a model fit is also provided in the figure 5.20. This fit can be used as a model to specify C_w at the wall as a function of the slip-wall Reynolds number once λ is known.

As a first step, we will apply the C_w computed using the traditional approach and apply

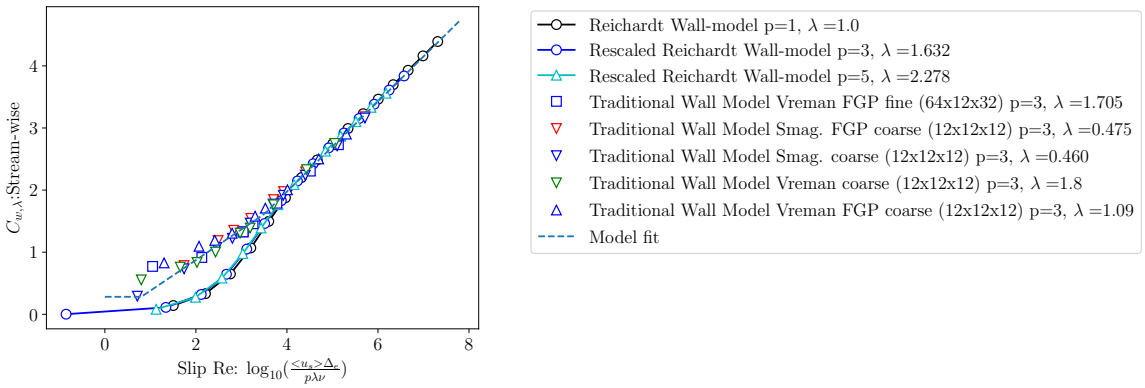


Figure 5.21: λ-normalized C_w vs. the slip-velocity u_s based Reynolds number $\frac{<u_s>\Delta}{p\lambda\nu}$. The λ-normalized C_w are computed by 1-D projection of Reichardt profile and compared to the same obtained using the traditional WMLES solution. The plots marked by "FGP" use the explicit sub-grid models inside the first element only and gradually change to implicit LES outside the first element.

it as a slip-boundary condition to check if the traditional WMLES results can be recreated with the slip boundary condition. At this stage, we are applying the same value of C_w for the stream-wise and the span-wise components. In this implementation, it is assumed that the there is no transpiration i.e. no flow through the wall. To apply the slip-wall boundary condition we first use the slip velocity components $u_{s,i}$ at the wall to compute the wall normal derivatives of the velocity components $u_{h,i}$ as follows:

$$\frac{\partial u_{h,i}}{\partial n} = \frac{u_{s,i}}{\Delta C_w}, \tag{5.9}$$

and finally compute the wall stress at any location using the following formula:

$$\tau_{w,i} = \nu \left.\frac{\partial u_{h,i}}{\partial n}\right|_w - \tau_{i,n}^{SGS}\big|_w . \tag{5.10}$$

Hence, contribution of the mean wall stress is only present from the viscous and the sub-grid stresses. In addition to the C_w obtained by post-processing the traditional approach solutions, the C_w computed using the slip-Reynolds number based model are also used. The value of λ, required for implementing the slip-Reynolds number based approach is obtained

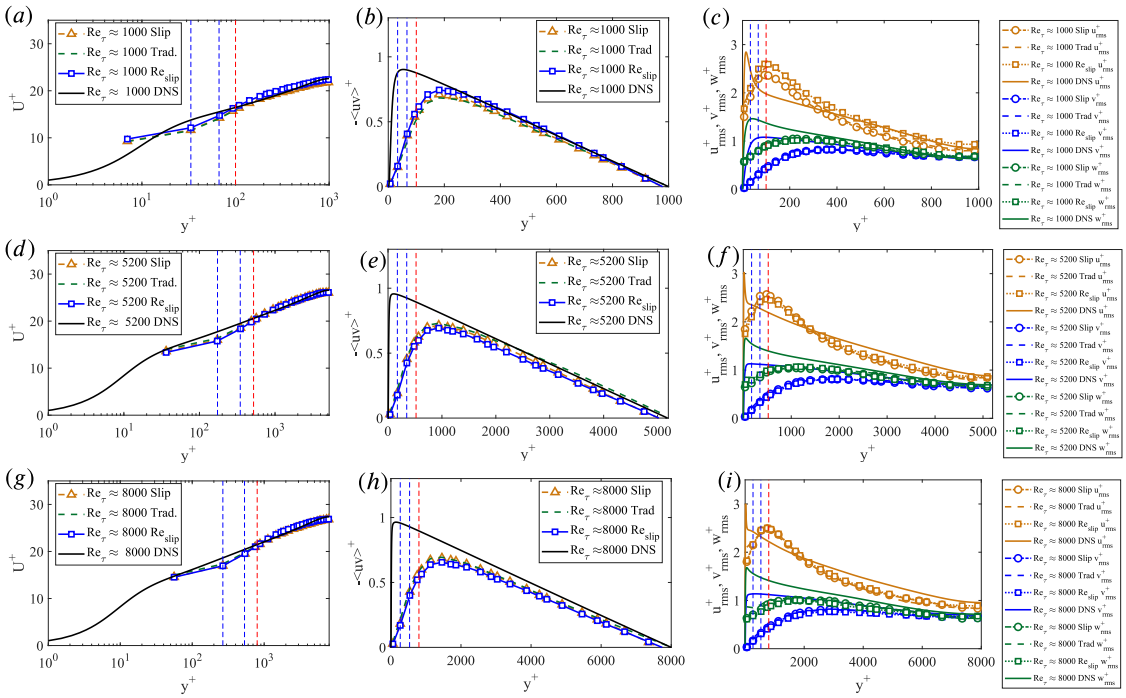


Figure 5.22: Comparison of the first-order and second-order statistics using the traditional method (Trad.), by re-using the slip-wall model with C_w computed from the traditional WMLES solution (Slip), and with the C_w computed using the slip Reynolds number formulation (Re_{slip}) at different friction Reynolds number. The vertical dashed lines show the locations of the first, second and third grid points. For the traditional wall-model, velocity is sampled at the third off-wall grid point.

from the traditional method. The slip-Reynolds number based model does not require the specification of different C_w's for each Re_τ case, however, requires one λ which remains constant across all the cases with different Re_τ. Figure 5.22 shows the stream-wise mean velocity profiles, the R.M.S of different velocity components and the Reynolds shear stress profiles at different friction Reynolds numbers. The vertical dashed lines show the location of the 1st, 2nd and 3rd off-wall grid points. For the traditional wall model, the wall-stress is computed using the velocity components at the 3rd off-wall grid point. The slip wall model does not require any such exchange location. It is clear from figure 5.22 that when the correct λ is used, the Reynolds number dependence is captured accurately and the statistics obtained using the slip-wall model are identical to the traditional wall model.

The previous tests presented in figure 5.22 showed that the model is able to capture the

Reynolds number dependence on a single grid. The next step is to change the grid-resolution and check if similar results also hold true for the new grid. Before performing numerical experiments with our proposed slip-wall model, an a priori study could be performed by using the results from the traditional WMLES solutions. Two different meshes are now used with $N_x \times N_y \times N_z$: $12 \times 12 \times 12$ and $18 \times 12 \times 18$ elements, respectively. The resolution in the wall normal direction is kept the same, whereas, the resolution in the span-wise and stream-wise case are smaller in case of the $18 \times 12 \times 18$ mesh because the size of the channel is kept constant.

Figure 5.23 shows the λ-normalized C_w obtained for different resolutions for two different types of normalizations. Different normalizations are used because the effective Δ is not known in the case where the element is not cubic. The plots marked by "WN" and "VOL" use the wall-normal grid-spacing and the cube root of the cell volume as Δ_e, respectively. It can be observed that when the wall-normal grid-spacing is used as Δ_e, the λ values required for the two different resolutions are different. This suggests that if the wall-normal grid resolution is used for Δ_e, our proposed slip-wall model will not generalise to a different grid for the same λ value. On the other hand, when the cell volume was used for Δ_e, the λ values required to ensure that both the curves collapse was found to be same. This suggests that, for the resolutions considered here, the cube root of the cell volume is an ideal candidate for Δ_e to ensure that the slip-wall model generalises to a new grid for the same value of λ. Hence, the proposed model will require the specification of the model constant λ and it is expected to work on different grids and Reynolds numbers. Figure 5.23 also shows plot of $C_{w,\lambda}$ at two other resolutions of $12 \times 16 \times 12$ and $32 \times 12 \times 16$. While constructing the model-fit these resolutions have not been considered. These plots for these specific resolutions will be later used to explain the success of the slip-wall model on these unseen resolutions.

Figures 5.24 shows the stream-wise mean velocity profiles, the R.M.S of different velocity components and the Reynolds shear stress profiles at different friction Reynolds numbers on two different meshes with $12 \times 16 \times 12$ and $32 \times 12 \times 16$ elements, respectively,

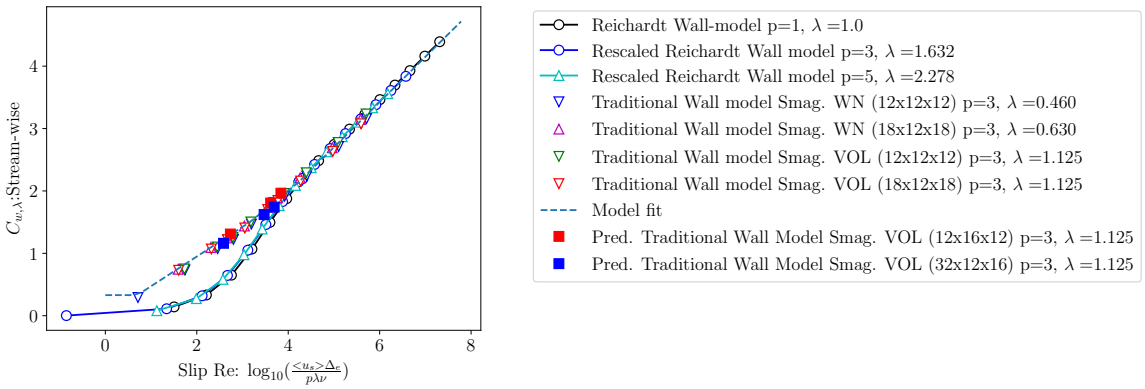


Figure 5.23: λ-normalized C_w vs. the mean slip-velocity $< u_s >$ based Reynolds number $\frac{<u_s>\Delta}{p\lambda\nu}$. The λ-normalized C_w are computed by 1-D projection of Reichardt profile and compared to the same obtained using the traditional WMLES solution. The plots marked by "WN" and "VOL" use the wall-normal grid-spacing and the cube root of the cell volume for specification of Δ_e, respectively.

that are not part of the data used for fitting the model for $C_{w,\lambda}$. Clearly, the model not only captures the effect of Re_τ but also generalises to a new resolution. The performance of the proposed slip-wall model is comparable to the traditional wall model which is a major improvement over the dynamic slip-wall model proposed by *Bae et al.* (2019). The results indicate that the proposed wall-model is able to work reasonably well even at considerably different resolutions. The excellent performance of the slip-wall model can be also explained by computing $C_{w,\lambda}$ using the traditional WMLES solutions on these grids. The $C_{w,\lambda}$ values estimated using the traditional WMLES solutions from two different meshes with $12\times16\times12$ and $32 \times 12 \times 16$ elements, respectively, are plotted in figure 5.23. The accurate prediction of $C_{w,\lambda}$ by the model fit explains the excellent predictive performance of our slip-wall model

.

5.6 Conclusion.

The projection-based scale-separation approach is an essential part of the variational multiscale method and uses the grid effectively as a filter. It is applicable to cases where the filter length is anisotopic, varies in space or filtering needs to performed on an unstructured

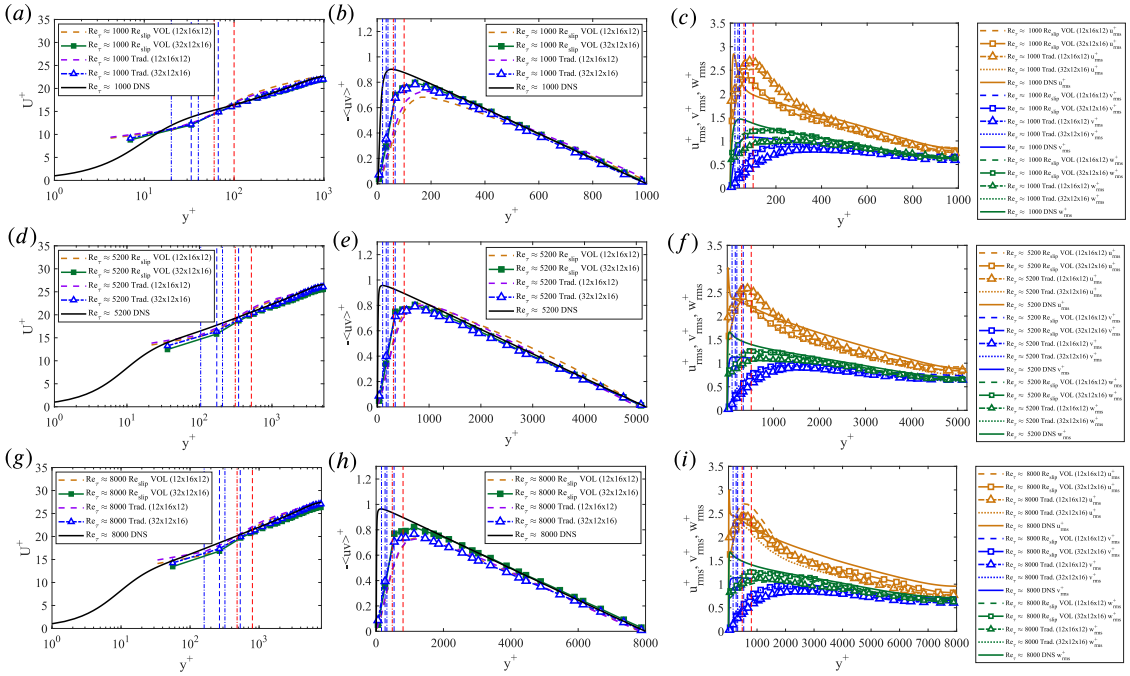


Figure 5.24: Comparison of the first-order and second-order statistics obtained using the traditional method (Trad.) and the proposed slip-wall model at different friction Reynolds numbers. The solution is computed on two different meshes with 12×16×12 and 32×12×16 elements, respectively, that is not part of the data used for fitting the model. The vertical dash-dotted and dashed lines show the locations of first, second and third grid points for the meshes with $12 \times 16 \times 12$ and $32 \times 12 \times 16$ elements, respectively. For the traditional wall-model, velocity is sampled at the third off-wall grid point.

grid. These filter properties were found to be essential for a priori assessment of existing coarse-grained methods for wall-bounded turbulent flows, where the grids can be highly anisotropic and vary in size along a particular direction.

An a priori assessment of the optimal solutions at three different limits: the wall-resolved LES, the hybrid RANS-LES and the WMLES limit, was performed by projecting DNS on different grids suitable for these scale-resolving approaches. For each of these cases, while projecting the DNS on to the coarse-space, weak imposition of the boundary condition was made by not enforcing no-slip boundary conditions at the boundary nodes. In the wall-resolved LES limit, the mean velocity was found to be well-resolved, no-slip was naturally satisfied and the turbulent stresses were well represented. In the hybrid RANS-LES limit, which was obtained by coarsening the wall-resolved LES mesh in the span-wise and stream-wise directions, the mean velocity was well-resolved and the no-slip boundary condition was naturally satisfied. However, the turbulent stresses were found to be well represented only at the center of the channel and under-represented in the near-wall region where sufficient resolution was not present. In the WMLES limit, which is obtained by further coarsening the hybrid RANS-LES grid in the wall normal direction, the mean profile is no longer represented accurately near the wall and a slip-velocity is obtained. The turbulent stresses in WMLES are relatively well-represented at the center of the channel compared to the near-wall region. In the near-wall region, the stream-wise and the span-wise velocity fluctuations were found to be non-zero at the wall, whereas, the the resolved wall-normal fluctuations and the turbulent shear-stress were found to be under-represented. All these trends were found to be consistent with existing solutions in the literature suggesting that the present framework can be utilized to assess, augment and calibrate existing methods.

The ability to obtain slip-velocity directly from 3-D projection of DNS on coarse near-wall meshes enabled further assessment of the existing slip-wall based wall-models. As a first step, estimates of the slip-wall model coefficient C_w were obtained from the mean velocity profile in the inner-layer through 1-D projections of the Reichardt profile. The

C_w estimates from the mean-profile were found to be strongly dependent on the order of projection suggesting that the numerical method has considerable impact on the optimal value of C_w. In addition to this, the resolution for a given slip velocity and projection order was found to scale with the wall units. To make modeling more tractable, we introduced an extra resolution normalizer λ to express the effect of projection order through a single coefficient, similar in scope to the Smagorinsky model coefficient C_s. When this analysis was extended to 3-D, similar dependence on the polynomial order p on C_w was found for the stream-wise and the span-wise velocity components. However, on re-introduction of resolution normalizer λ and reusing the λ values corresponding to the 1-D projections, similar collapse in the $C_{w,\lambda}$ values was also observed for the 3-D case. The value of $C_{w,\lambda}$ was also found to be different for the stream-wise, span-wise and the wall-normal velocity components.

The ultimate goal of any a-priori analysis is to improve the model performance in a posteriori calculations. As a first step towards better slip based wall models, the performance of existing slip-based wall models was compared to traditional WMLES for channel flows. To establish an equivalence between the two methods, $C_{w,\lambda}$ curves were evaluated using the solution of the traditional WMLES approach and compared with the curves obtained for the Reichardt profile. The $C_{w,\lambda}$ curves for the traditional WMLES solutions were found to be identical to those obtained using the Reichardt profile at high $\frac{\Delta_e^+}{\lambda p}$. However, at low $\frac{\Delta_e^+}{\lambda p}$, the $C_{w,\lambda}$, the curves were found to differ suggesting the presence of sub-grid modeling and wall-modelling errors in the solution. To reduce the implementation challenges associated with using $\frac{\Delta_e^+}{\lambda p}$ as a feature, a slip Reynolds number-based (Re_{slip}) feature was introduced. Finally, by choosing Δ_e to be the cube root of the cell volume and re-using λ from the traditional WMLES solution, a model form was constructed by fitting the $C_{w,\lambda}$ vs. Re_{slip} curve. The resulting model was shown to generalize to different resolutions, element aspect ratios and Reynolds numbers in a posteriori simulations.

CHAPTER VI

Perspectives.

6.1 Conclusions.

Efficient simulation of multi-scale problems requires the development of coarse-grained models which allow the computation of relevant quantities of interest at a fraction of the cost of solving the original system. This dissertation developed various coarse-grained modeling strategies, with the variational multi-scale (VMS) method as the overarching theme.

• First, the VMS method was combined with the optimal prediction framework of Chorin to develop a coarse-graining strategy for the continuous Galerkin finite element method that applies to non-linear PDEs. At a fraction of the cost of solving the full order solution, the resulting model was shown to predict the correct evolution of the large-scale dynamics for various problems such as the Burgers turbulence, the 3-D homogeneous isotropic turbulence, and the 3-D Taylor green vortex.

• Second, the VMS method was combined with a novel sub-scale neural network structure - the variational super-resolution N-N (VSRNN) to discover sub-grid closures directly from data. The structure of the VSRNN was constructed to approximate the unresolved scales as a sum of the products of individual non-dimensional functions of coarse scales and physics-informed parameters. This model was trained on a sequence of projected data and finally used to compute the continuous Galerkin subgrid terms and super-resolved state to compute the discontinuous Galerkin fluxes. The resulting formulation led

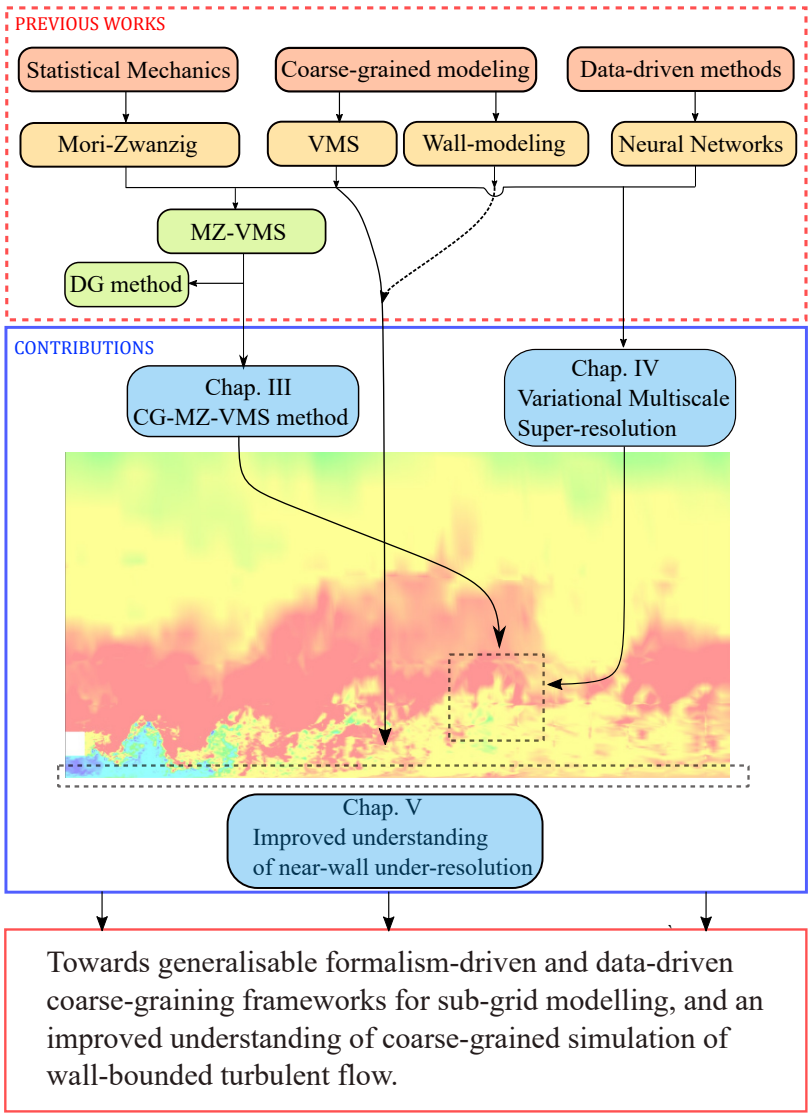


Figure 6.1: Sketch of contributions in this dissertation.

to the improvement in the optimality and the accuracy of present methods for the convection-diffusion, linear advection, and turbulent channel flow problems.

• Third, inspired by the VMS method, exact scale decomposition was performed through L_2-projection of the channel flow data on various finite element spaces leading to a new finite element-based filtering technique. The proposed filtering technique was found to explain the occurrence of slip velocity in WMLES simulations and provide further insights into the shortcomings of the slip-wall models by *Bose and Moin* (2014) and *Bae et al.* (2019). These insights were used to construct a new slip-wall wall model that uses the Reynolds number

based on the slip velocity and mesh resolution as a feature and demonstrates excellent performance by generalizing to different Reynolds numbers and mesh resolutions.

In the following few sections of this chapter, we will outline the significant contributions of this work and discuss future research opportunities.

6.2 Contributions.

The major contributions of the work are as follows:

1. **Extension of MZ-VMS to derive closures for continuous Galerkin finite element discretizations of PDEs.** In extension to work by *Parish and Duraisamy* (2017c) for spectral methods and the discontinuous Galerkin method, we apply the MZ-VMS framework to the continuous Galerkin method. This is the first continuous Galerkin closure that is built entirely using a non-linear model reduction strategy. Application of the current approach to non-linear problems does not require constructs such as transformations to linear problems such as the Oseen equations at every non-linear iteration. The current formulation does not require one to assume a particular model form for the sub-scales, as most non-linear VMS closures require.

2. **Development of a dynamic procedure for adapting the memory length present in the MZ-VMS closures for the CG method.** A robust formulation based on Germano's identity was derived for dynamically obtaining the memory length required in our finite memory-based MZ-VMS model. The test filtering methodology was adapted to the continuous Galerkin method by performing L_2-projection on a coarser grid. We further established that the predicted value of τ using this dynamic procedure was linear primarily in time, as conceptualized by the re-normalized t-model by *Stinis* (2013).

3. **Extensive evaluation of the dynamic MZ-VMS closures for the continuous Galerkin finite element method on canonical turbulent flow problems .** The dynamic MZ-

VMS framework for the continuous Galerkin approach was assessed in detail for the 1-D viscous Burgers equation and in many canonical 3-D flow problems, such as homogeneous isotropic turbulence and the Taylor-green vortex. The performance of our dynamic MZ-VMS framework for these cases was found to be better or comparable to the state-of-the-art VMS models.

4. **Discovering a link between super-resolution and the Greens functions approach used in the development of VMS closures for linear problems.** A connection between the super-resolution operator, used for up-scaling coarse-grained PDE solutions, and the fine-scale solution obtained using the Greens function approach in VMS was established. Using the learned super-resolution operator as a model for the fine-scales, data-driven sub-grid modeling was performed using the VMS formalism. In this work, we demostrated the reconstructive and predictive performance of the super-resolution models.

5. **First generalizable data-driven closure learning framework for the Variational multiscale method.** To the authors' knowledge, this is the first data-driven VMS closure learning approach that, for a limited class of problems investigated herein, can generalize to different grids and physics-informed parameters, making the current approach truly predictive. The present approach achieves generalization using a unique neural network structure called the variational super-resolution N-N (VSRNN), which uses non-dimensional input and output features and takes into account the physics-informed parameters. We demonstrated in Chapter 4 how a closure in the form of a singly trained VSRNN works for different resolutions, physics-informed parameters, and initial conditions.

6. **Improving the optimality and accuracy of existing finite element discretizations though the development of data-driven sub-grid models for various PDEs.** In our data-driven framework, the required optimality is imposed by the projection

operation used to generate the coarse-scale and the fine-scale data for training the VSRNN network. In the linear advection problem, close to L_2-optimal solution was obtained. Similarly, in the advection-diffusion problem, the closure was trained such that the coarse scale was nodally exact. In addition to optimality, the resulting numerical method was highly accurate in comparison to the traditional methods. The VSRNN also provides the ability to learn closures for high-order finite element discretizations.

7. **Development of a finite element based filtering approach for the a priori analysis of coarse-grained scale-resolving simulations of wall turbulence.** A new filtering approach for non-homogeneous flows was developed, which allows the filtering of high-resolution DNS data when non-periodic directions are present or when the mesh is stretched. The present approach enables a priori assessment of coarse-grained scale-resolving methods for wall-bounded flows, such as the wall-resolved LES, hybrid RANS-LES, and WMLES methods.

8. **Discovery of a new scaling parameter that allows the construction of a universal model form for the model coefficient present in the slip-wall based wall-models.** The effect of the sub-grid model and the numerical scheme on the slip velocity was effectively captured through the introduction of a resolution normalizer λ, which allowed the construction of a universal model form for the model coefficient in the slip-wall-based wall models.

9. **Improving the performance of the state-of-the-art slip wall model.** By capturing the effect of Reynolds number and grid resolution, the proposed wall model with a constant model coefficient outperforms the state-of-the-art dynamic slip wall models by *Bae et al.* (2019) and *Bose and Moin* (2014) on the turbulent channel flow problem.

10. **Development of caslabDG.** The high-order discontinuous Galerkin (DG) code with explicit RK3/RK4-based time-stepping is developed using MPI (Message Passing

Interface), a communication protocol for distributed parallel computing. The code shows a strong scale up to 512 processors on problems with a size comparable to that considered in this dissertation. The code is built with both traditional and slip-wall-based WMLES capabilities. Additional details are provided in appendix E.

6.3 Perspectives and future work.

6.3.1 Formalism-driven discovery of sub-grid closures

The MZ-VMS provides a general framework for obtaining the exact form of the closure for the correct evolution of the coarse scales. However, the evaluation of the memory term in the closure requires the solution to a highly high-dimensional PDE known as the orthogonal dynamics equation. As we saw in Chapter 3, the memory term needs to be modeled for all practical purposes. The accuracy of the memory model is critical to the success of the coarse-grained model. To this end, we outline the following ingredients for performing accurate memory modeling:

1. The memory model needs to be generalizable, i.e., it should adapt to the grid, time step, and problem. The constant-τ model is not adaptive to the grid, time-step, and problem. The τ in the constant-τ memory model is a time scale with the dimension of time. If the units of the time change, the value of the constant τ must be adjusted accordingly. Hence, its value cannot be universal. One solution is to scale it with the time-step Δt or a local time scale such as $\frac{\Delta}{|u_h|}$, where $|u_h|$ is a local velocity scale. Given the similarity in the structure of the VMS-ϵ model and the constant-τ model, and the similarity between the memory length and the stabilization parameter, $\frac{\Delta}{|u_h|}$ seems a natural choice. Consequently, the memory length can be assumed to have the following form:

$$\tau = C\frac{\Delta x}{|u_h|}. \tag{6.1}$$

where C is a non-dimensional constant. An alternate approach is to perform dynamic

modeling for τ as described in Chapter 3. At an additional cost, the dynamics model adapts to the grid, time-step, and problem automatically.

2. The present dynamic model requires the specification of an ad-hoc scaling law relating the memory lengths τ at two different levels of coarsening i.e.

$$\frac{\tau_1}{\tau_2} = \left[\frac{\Delta_1}{\Delta_2}\right]^{1.5}. \tag{6.2}$$

As described in Chapter 3 and *Parish and Duraisamy* (2017c), this law is obtained by computing the ratio between the integrated memory and the memory at s = 0 at different levels of coarse-graining a priori using high-resolution data from a variety of flows. Although the model form of the present finite memory-based MZ-VMS closure was formally developed, the specification of a scaling law for τ makes the final model phenomenological. Hence, it is necessary to construct models that do not require the specification of scaling laws with exponents obtained directly from existing theory or data. One approach is to assume a scaling law of the following form:

$$\frac{\tau_1}{\tau_2} = \left[\frac{\Delta_1}{\Delta_2}\right]^{m}. \tag{6.3}$$

However, the exponent m is also dynamically obtained.

3. In this dissertation, coarse-grained model development using MZ was focused on fixed memory type models leading to Markovian closures. However, other approaches leading to non-Markovian type closures can be implemented *Parish* (2018). Alternatively, different approximations to the orthogonal dynamics *Zhu and Venturi* (2018); *Gouasmi et al.* (2017) can be used to construct models.

4. The final comment is on the use of a single memory length for all the modes. For complex flows, the value of τ can vary across different regions and cannot be assumed constant. The single memory length model was successful - in part - because of the

nature of the problems investigated herein. Hence, extensions to highly anisotropic and inhomogeneous problems will require the development of local definitions for memory length. One candidate is the memory length form proposed in point #1, i.e., $\tau = C\frac{\Delta}{|u_h|}$, which is based on a local definition of a time scale. In this case, the value of the τ depends on the velocity magnitude and grid size and is not uniform across all the modes. Finally, to make this model fully predictive, the dynamic procedure can be performed on the non-dimensional parameter C in place of the memory length τ.

6.3.2 Data-driven discovery of sub-grid closures

Inspired by successes in the machine vision community, there has recently been considerable interest in the use of super-resolution in the physical sciences. Much of the existing literature has, however, focused on reconstruction performance and not on predictive modeling. Truly predictive models should not be restricted to a single mesh or flow configuration, and should generalize to a class of flows. Despite the success in the canonical problem in section 4.6, the results in section 4.7 suggest that there is much to be done before a truly predictive capability can be realized for a problem as challenging as turbulent flow. We view our work as a first step in moving towards a predictive LES capability. Along these lines, we outline the following ingredients for the discovery of sub-grid closures:

1. The model should be constructed using features that lend themselves to generalization
2. The structure of the learning model should allow one to embed physics-informed parameters efficiently.
3. The closure model should be intimately linked to the underlying numerical discretization.
4. The training should be performed in a manner that the super-resolution is consistent with the coarse scales during the prediction.

In our work, we addressed points #1 #2 above by choosing non-dimensional features that are inspired by VMS closures, and by choosing a compositional neural network structure. Further work is required to design features that satisfy additional physics-informed invariances.

Regarding point #3, in contrast to implicitly filtered approaches in which coarse space is defined ambiguously *Lund* (2003), the VMS approach formally segregates the coarse and fine spaces, thus setting a clean environment for super-resolution. Other candidates include explicitly filtering *Lund* (2003) with a large test filter.

Point #4 refers to establishing consistency between the learning and prediction environments *Duraisamy* (2020). In essence, the training is performed on DNS data, i.e. $u' = f(u_h^{DNS})$, whereas in the online prediction stage, it is used as $u' = f(u_h^{LES})$. As the error between the coarse scales in the LES and DNS grows, the super-resolution becomes less accurate. In other words, the parameters of the learning model have not been inferred for online performance. Model-consistent training has been successfully demonstrated in RANS closures *Parish and Duraisamy* (2016); *Singh and Duraisamy* (2016); *Holland et al.* (2019), the authors are aware of only one such attempt in the context of LES *Sirignano et al.* (2020). However, as mentioned above, and in more detail in Ref. *Sirignano et al.* (2020), implicitly filtered approaches are associated with other challenges. The VMS approach, on the other hand, allows for both numerics-consistent and model-consistent training, but the implementation of such a capability is a major undertaking is yet to be pursued by the authors in an LES context.

As a final point, while the appeal of VMS is the segregation of scales and the prospects of deriving closures with few phenomenological assumptions, structural models (e.g., *Pradhan and Duraisamy* (2020)) generally perform poorly when the simulation is severely under-resolved. Several attempts *Hughes et al.* (2000); *Wang and Oberai* (2010b) have been made to combine traditional VMS approaches with phenomenological models like Smagorinsky in the form of mixed models. The use of data-driven techniques potentially allows us to

account for these phenomenological relationships in the data directly in the VMS model, thus, bridging the gap between phenomenological and structural modeling.

6.3.3 Improved slip-wall models.

Slip-based wall models *Bose and Moin* (2014); *Bae et al.* (2019); *Whitmore et al.* (2021) allow for easy implementation of the wall boundary condition without the need to sample velocity components at a few grid points away from the wall and also allows the possibility to model flow separation. The dynamic slip-wall model proposed by *Bae et al.* (2019) shows excellent performance on the zero-pressure gradient flat plate case, albeit for a narrow range of Re_θ. However, this model requires improvements compared to traditional wall models for the equilibrium channel flow case at similar resolutions. Section 5.4 provided a priori results on the model form for C_w for equilibrium channel flows. In Section 5.5, we used some of the insights obtained from section 5.4 to improve the performance of the existing slip-wall model to at least the traditional WMLES level. Although the performance of our proposed slip wall model was acceptable, more insights from section 5.4 can be used to improve further the accuracy of both the proposed and the existing slip wall models. To this end, we outline the following ingredients for the construction of a more generalizable slip-based wall model form:

1. The slip model coefficients can be different in the stream-wise, span-wise, and wall-normal directions as observed in figure 5.19. In a more complex 3-D case, the choice of stream-wise, span-wise, and wall-normal direction is slightly ambiguous. However, the mean flow can be used to identify these directions. However, this needs to be iteratively done since the mean flow can change when changing these directions. Another approach is to use the flow direction at the first off-wall grid point, similar to how the traditional wall models are implemented. In Section 5.5, we used the same C_w for all directions. The effect of using different C_w for different velocity components on our proposed model is a topic of further research .

2. If a dynamic modeling procedure is performed to obtain C_w, the value of C_w cannot be assumed to be the same at the original grid and the test filtered grid. Figure 5.19 shows that C_w changes when the resolution is changed from Δ^+ to the test filtered grid resolution $2\Delta^+$. In addition to C_w being not constant across different grid levels, there is a dependence on the wall units. This dependence is generally not considered in the existing slip-wall model forms. However, this dependence is present in traditional wall models, which are found to perform excellently for equilibrium wall-bounded flow cases. In Section 5.5, we were able to improve the performance of the dynamic slip-wall model of *Bae et al.* (2019) by just augmenting the model form without performing any dynamic procedure.

3. The discrepancy in figures 5.20 and 5.21 between the $C_{w,\lambda}$ curves obtained by the optimal projection of the Reichardt profile and that obtained using the solutions of the traditional WMLES approach suggests that the current WMLES approaches are sub-optimal due to the presence of wall-modeling and sub-grid modeling errors. This discrepancy also suggests that there is ample scope for improvement. Our optimal projection framework can be used to assess the WMLES performance of the different combinations of sub-grid and wall models.

4. The final comment is on the choice of the parameter that should be used to perform the dynamic procedure. We saw in section 5.4 that the value of λ effectively captures the effect of the order of projection and hence the numerical method. The corresponding function g_1 is fairly universal for different orders. Hence, it is imperative that the dynamic modeling be performed on λ rather than C_w. The model form for g_1 can be empirically obtained from DNS data, Reichardt profile, or from the solution of an existing model such as the wall-stress-based WMLES models. We further observed in section 5.5 that if the cube root of the cell volume is used for Δ_e, λ remains fairly constant across different resolutions and Reynolds numbers for a given sub-grid

model. A dynamic model that determines λ without requiring the solutions from the traditional model is a topic of further research.

In this work, two different model forms for $C_{w,\lambda}$ were proposed as shown in figure 5.20 and figure 5.21. The first model uses Δ^+ as a feature, whereas the second uses the Reynolds number based on the slip velocity. The implementation of the first model is slightly more complex because the proposed expression for C_w is a function of two parameters: λ and Δ^+. Although the grid size Δ is known, to compute Δ^+ from Δ, an estimate of the average wall stress $< \tau_w >$ is required:

$$\langle \tau_w \rangle = \nu \left\langle \left. \frac{\partial u_{h,1}}{\partial y} \right|_w \right\rangle - \left\langle u_{h,1} u_{h,2} \big|_w \right\rangle - \left\langle \left. \tau_{12}^{SGS} \right|_w \right\rangle . \tag{6.4}$$

This average quantity influences the slip velocities through C_w, affecting the average itself. As discussed earlier, an alternate approach is to use the equilibrium wall profile to obtain τ_w as done in *Whitmore et al.* (2021). However, this requires sampling the velocity fields from the off-wall grid points, which makes the implementation of slip-wall models as cumbersome as the traditional wall model. An alternate approach is using the Reynolds number based on the slip velocity described in section 5.5. In this work, the optimal estimates of C_w were obtained from the DNS solution by projecting onto the spaces formed by uniform elements of different sizes. However, our projection framework, by using anisotropic elements, also allows us to study the effect of the grid aspect ratios. As discussed in section 5.5, one approach to account for mesh anisotropy is to replace Δ_e with an effective grid size, such as the cube root of the cell volume. However, it is advisable to include the aspect ratio in the model form to ensure optimal performance across different meshes. Finally, the present model form has been derived from the channel flow data, and its accuracy in the spatially developing flows, such as the flat plate or the separated flow regions, has not been assessed. Given the excellent performance of the traditional wall models on the flat-plat cases, we expect similar performance from our proposed model; however, this is a topic of further

research.

6.3.4 Definition of accuracy of coarse-grained models and potential data-driven improvements.

As is well-recognized, the projected solution u_h is no longer a solution to the Navier-Stokes equations and requires additional modeling in the form of closure. In context of VMS (*Hughes et al.*, 1998b), the coarse-scale equation is given by

$$\left(\frac{\partial u_h}{\partial t}, w_h\right) + (R(u_h), w_h) + (R(u) - R(u_h), w_h) = 0 \quad \forall w_h \in \mathcal{V}_h, \tag{6.5}$$

where the fine scales are considered orthogonal to the coarse-scales (*Codina*, 2002). The orthogonality condition is required to ensure that $(\frac{\partial u_h}{\partial t}, w_h) = (\frac{\partial u}{\partial t}, w_h)$. Equation 6.5 after sub-grid modelling represents the evolution equation for u_h and requires boundary conditions on u_h. As observed in previous sections, boundary conditions applicable for u do not always translate to u_h. This is especially true in the WMLES limit, where slip is observed for u_h. The focus of the present work is not on closure modeling. However, as the first step towards closure modeling, one can use the present optimal projection to estimate the closure term when the filter length is anisotropic and changes in space. We also assess the impact of coarse-graining on the boundary conditions required for u_h.

Irrespective of how closely these guidelines are followed, the results from different coarse-grained simulations (i.e., WRLES, WMLES, and hybrid RANS-LES) are extremely sensitive to the grid and the numerical method. It is challenging to define grid convergence for these methods. In this scenario, there exists no formal way to define an accurate WMLES, LES, or hybrid-RANS LES. By projecting the DNS on the same coarse mesh that is used for the coarse-grained simulation, it is possible to obtain an optimal solution for that mesh. The accuracy of the underlying coarse-graining method can be assessed by computing the difference between the coarse-scale statistics obtained from the optimally projected solution

and that obtained using the coarse-grained model. The sub-grid model can be trained in a way to minimize the difference between the optimal and the modeled coarse-scale *outputs* This approach is called the *model consistent* approach (*Duraisamy et al.*, 2019; *Duraisamy*, 2021) which requires the solver to be part of the model training process.

APPENDICES

APPENDIX A

Derivation of the memory kernel for the finite memory model.

Different ways to model the memory term have been explored in the literature *Chorin and Hald* (2009); *Chorin et al.* (2000); *Stinis* (2015, 2012); *Parish and Duraisamy* (2017b); *Zhu and Venturi* (2018). In the present formulation, we will derive the memory kernel for the finite memory model i.e. $\int_0^t K(\tilde{\mathbf{a}}(t-s), s)ds \approx \tau K(\tilde{\mathbf{a}}(t), 0)$ where the memory kernel $K(\tilde{\mathbf{a}}(t-s), s)$ at $s = 0$ is given by

$$K(\tilde{\mathbf{a}}(t), 0) = e^{\mathcal{L}t}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_{\mathbf{0}}. \tag{A.1}$$

First, we apply $\mathcal{L}$ on $\tilde{\mathbf{a}}_{\mathbf{0}}$, resulting in the RHS of Equation (3.8) given by

$$e^{\mathcal{L}t}\mathcal{L}\tilde{\mathbf{a}}_{\mathbf{0}} = \tilde{\mathbf{M}}^{-\mathbf{1}}(-(R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} - (R(u) - R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} - (b(\tilde{u}), \tilde{\mathbf{w}})_{\Gamma'} - (b(u) - b(\tilde{u}), \tilde{\mathbf{w}})_{\Gamma'} + (f, \tilde{\mathbf{w}})_{\Omega'}), \tag{A.2}$$

Second, we apply the projection $\mathcal{Q} = \mathcal{I} - \mathcal{P}$ to Equation (A.2) which results in the following expression,

$$e^{\mathcal{L}t}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_{\mathbf{0}} = \tilde{\mathbf{M}}^{-\mathbf{1}}(-(R(u) - R(\tilde{u}), \tilde{\mathbf{w}})_{\Omega'} - (b(u) - b(\tilde{u}), \tilde{\mathbf{w}})_{\Gamma'}). \tag{A.3}$$

Third, we apply the Liouville operator $\mathcal{L}$ to obtain $e^{\mathcal{L}t}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_\mathbf{0}$. The effect of application of the Liouville operator $\mathcal{L}$ to any scalar function results in the Frechet derivative evaluated in the direction of the RHS i.e. $\mathcal{L}\mathbf{a}_0$. For example, for a scalar function g we have the following:

$$\mathcal{L}g(u(\mathbf{a}_0)) = \frac{\partial g}{\partial \mathbf{a}_0}\mathcal{L}\mathbf{a}_0 \tag{A.4}$$

Recognising that $u = \mathbf{w}^T\mathbf{a}$ and applying chain rules we get

$$\mathcal{L}g(u(\mathbf{a}_0)) = \frac{\partial g}{\partial u_0}\mathbf{w}^T\mathcal{L}\mathbf{a}_0 \tag{A.5}$$

Finally $e^{\mathcal{L}t}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_\mathbf{0}$ is obtained by linearising w.r.t to u and evaluating the the RHS of Equation (3.8) and (3.9) as follows:

$$\begin{aligned}
e^{\mathcal{L}t}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_\mathbf{0} = -\tilde{\mathbf{M}}^{-1}(R'(\mathbf{w}^\mathbf{T}[\mathbf{M}^{-1}(-(R(\tilde{u})-f,\mathbf{w})_{\Omega'} - (R(u)-R(\tilde{u}),\mathbf{w})_{\Omega'} - (b(\tilde{u}),\mathbf{w})_{\Gamma'} \\
-(b(u)-b(\tilde{u}),\mathbf{w})_{\Gamma'})]) - R'(\tilde{\mathbf{w}}^\mathbf{T}[\tilde{\mathbf{M}}^{-1}(-(R(\tilde{u})-f,\tilde{\mathbf{w}})_{\Omega'} - (R(u)-R(\tilde{u}),\tilde{\mathbf{w}})_{\Omega'} - (b(\tilde{u}),\tilde{\mathbf{w}})_{\Gamma'} \\
-(b(u)-b(\tilde{u}),\tilde{\mathbf{w}})_{\Gamma'})]),\tilde{\mathbf{w}})_{\Omega'} - \tilde{\mathbf{M}}^{-1}(b'(\mathbf{w}^\mathbf{T}[\tilde{\mathbf{M}}^{-1}(-(R(\tilde{u})-f,\mathbf{w})_{\Omega'} - (R(u)-R(\tilde{u}),\mathbf{w})_{\Omega'} \\
-(b(\tilde{u}),\mathbf{w})_{\Gamma'} - (b(u)-b(\tilde{u}),\mathbf{w})_{\Gamma'}]) - b'(\tilde{\mathbf{w}}^\mathbf{T}[\tilde{\mathbf{M}}^{-1}(-(R(\tilde{u})-f,\tilde{\mathbf{w}})_{\Omega'} - (R(u)-R(\tilde{u}),\tilde{\mathbf{w}})_{\Omega'} - \\
(b(\tilde{u}),\tilde{\mathbf{w}})_{\Gamma'} - (b(u)-b(\tilde{u}),\tilde{\mathbf{w}})_{\Gamma'})]),\tilde{\mathbf{w}})_{\Gamma'}.
\end{aligned} \tag{A.6}$$

Finally, we apply the projector $\mathcal{P}$ which removes the dependence on un-resolved variables $\mathbf{a}'$ and results in

$$\begin{aligned}
\tilde{\mathbf{M}}e^{\mathcal{L}t}\mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_\mathbf{0} = -(R'(\mathbf{w}^\mathbf{T}[\mathbf{M}^{-1}(-(R(\tilde{u})-f,\mathbf{w})_{\Omega'} - (b(\tilde{u}),\mathbf{w})_{\Gamma'})]) \\
-R'(\tilde{\mathbf{w}}^\mathbf{T}[\tilde{\mathbf{M}}^{-1}(-(R(\tilde{u})-f,\tilde{\mathbf{w}})_{\Omega'} - (b(\tilde{u}),\tilde{\mathbf{w}})_{\Gamma'})]),\tilde{\mathbf{w}})_{\Omega'} - \\
(b'(\mathbf{w}^\mathbf{T}[\mathbf{M}^{-1}(-(R(\tilde{u})-f,\mathbf{w})_{\Omega'} - (b(\tilde{u}),\mathbf{w})_{\Gamma'}]) - \\
b'(\tilde{\mathbf{w}}^\mathbf{T}[\tilde{\mathbf{M}}^{-1}(-(R(\tilde{u})-f,\tilde{\mathbf{w}})_{\Omega'} - (b(\tilde{u}),\tilde{\mathbf{w}})_{\Gamma'})]),\tilde{\mathbf{w}})_{\Gamma'},
\end{aligned} \tag{A.7}$$

Equation (A.7) can be compactly written as,

$$\begin{aligned}\tilde{\mathbf{M}} e^{\mathcal{L}t} \mathcal{P}\mathcal{L}\mathcal{Q}\mathcal{L}\tilde{\mathbf{a}}_{\mathbf{0}} = &\int_{\Omega'}\int_{\Omega'} \tilde{\mathbf{w}} R'(\Pi'(x,y)(R(\tilde{u}) - f)) d\Omega'_y d\Omega'_x \\ &+ \int_{\Omega'}\int_{\Gamma'} \tilde{\mathbf{w}} R'(\Pi'(x,y)(b(\tilde{u}))) d\Gamma'_y d\Omega'_x \\ &+ \int_{\Gamma'}\int_{\Omega'} \tilde{\mathbf{w}} b'(\Pi'(x,y)(R(\tilde{u}) - f)) d\Omega'_y d\Gamma'_x \\ &+ \int_{\Gamma'}\int_{\Gamma'} \tilde{\mathbf{w}} b'(\Pi'(x,y)(b(\tilde{u}))) d\Gamma'_y d\Gamma'_x,\end{aligned} \tag{A.8}$$

where Π' is the orthogonal projector onto the space of the the fine scales i.e,

$$\Pi'(x,y) = \mathbf{w'}^T(x)\mathbf{M'^{-1}}\mathbf{w'}(y). \tag{A.9}$$

For a simpler derivation for a smooth orthogonal basis, readers are encouraged to explore Appendix A of *Parish and Duraisamy* (2017c).

APPENDIX B

The VMS-OSS coarse grain model.

In this part, we review the Othogonal Sub-Scale (OSS) *Codina* (2002) method which has been used as a model for comparison throughout Chapter 3. We start with the linearized form of the N-S equation obtained from the Picard algorithm leading to an Oseen problem at every non-linear iteration as follows

$$\partial_t \mathbf{u} - \nu \Delta \mathbf{u} + \mathbf{a} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} \quad in \quad \Omega, t \in]0, T[, \tag{B.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{B.2}$$

where $\mathbf{a}$ is the convective velocity which is defined later. The above set of equations can be written in the following form:

$$\mathbf{M}\partial_t \mathbf{U} + \mathscr{L}(\mathbf{U}) = \mathbf{F}, \tag{B.3}$$

where $\mathbf{U}$, $\mathbf{M}$, $\mathscr{L}(\mathbf{U})$, F are given by

$$\mathbf{U} = \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix}, \quad \mathbf{M} = diag(\mathbf{I}, 0), \quad \mathscr{L}(\mathbf{U}) = \begin{bmatrix} -\nu\Delta\mathbf{u} + \mathbf{a}\cdot\nabla\mathbf{u} + \nabla p \\ \nabla\cdot\mathbf{u} \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}. \tag{B.4}$$

The θ family of methods is used for the temporal discretization of Equation B.3 as follows

$$\mathbf{M}\delta_t\mathbf{U}^{\mathbf{n}} + \mathscr{L}(\mathbf{U}^{\mathbf{n}+\theta}) = \mathbf{F}^{\mathbf{n}+\theta}, \tag{B.5}$$

where $\delta_t\mathbf{U}^{\mathbf{n}}$ and $\mathbf{U}^{\mathbf{n}+\theta}$ are defined as

$$\delta_t\mathbf{U}^{\mathbf{n}} = \frac{\mathbf{U}^{n+1} - \mathbf{U}^{n}}{\delta_t}, \tag{B.6}$$

$$\mathbf{U}^{\mathbf{n}+\theta} = \theta\mathbf{U}^{n+1} + (1-\theta)\mathbf{U}^{n}. \tag{B.7}$$

The variational form of Equation (B.5) is given by

$$(\mathbf{M}\delta_t\mathbf{U}^{\mathbf{n}}, \mathbf{V}) + (\mathscr{L}(\mathbf{U}^{\mathbf{n}+\theta}), \mathbf{V}) = (\mathbf{F}^{\mathbf{n}+\theta}, \mathbf{V}), \tag{B.8}$$

where the weighting functions $\mathbf{V}$ is defined as

$$\mathbf{V} = \begin{bmatrix} \mathbf{w} \\ k \end{bmatrix}. \tag{B.9}$$

The next step is to decompose $\mathbf{U}$ and $\mathbf{V}$ into resolved and sub-grid parts i.e. $\mathbf{U} = \mathbf{U_h} + \mathbf{U}'$ and $\mathbf{V} = \mathbf{V_h} + \mathbf{V}'$ where $\mathbf{U_h}$ and $\mathbf{V_h}$ both belong to the subspaces spanned by the piecewise polynomial basis functions in a typical finite element calculation. Whereas, $\mathbf{U}'$ and $\mathbf{V}'$ can assume functions which belong to a space orthogonal to $\mathcal{V}_h$ i.e. $\mathcal{V}'$. The above decomposition leads to the following equation:

$$(\mathbf{M}\delta_t(\mathbf{U_h}^{\mathbf{n}} + \mathbf{U'}^{\mathbf{n}}), \mathbf{V_h} + \mathbf{V'}) + (\mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}} + \mathbf{U'}^{\mathbf{n+\theta}}), \mathbf{V_h} + \mathbf{V'}) = (\mathbf{F}^{\mathbf{n+\theta}}, \mathbf{V_h} + \mathbf{V'}). \quad \text{(B.10)}$$

By applying the standard VMS procedure we get

$$(\mathbf{M}\delta_t(\mathbf{U_h}^{\mathbf{n}} + \mathbf{U'}^{\mathbf{n}}), \mathbf{V_h}) + (\mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}} + \mathbf{U'}^{\mathbf{n+\theta}}), \mathbf{V_h}) = (\mathbf{F}^{\mathbf{n+\theta}}, \mathbf{V_h}), \quad \text{(B.11)}$$

$$(\mathbf{M}\delta_t(\mathbf{U_h}^{\mathbf{n}} + \mathbf{U'}^{\mathbf{n}}), \mathbf{V'}) + (\mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}} + \mathbf{U'}^{\mathbf{n+\theta}}), \mathbf{V'}) = (\mathbf{F}^{\mathbf{n+\theta}}, \mathbf{V'}). \quad \text{(B.12)}$$

On further simplification due to orthogonality of the two spaces and using integration by parts we obtain

$$(\mathbf{M}\delta_t\mathbf{U_h}^{\mathbf{n}}, \mathbf{V_h}) + (\mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}}), \mathbf{V_h}) + (\mathbf{U'}^{\mathbf{n+\theta}}, \mathscr{L}^*(\mathbf{V_h})) = (\mathbf{F}^{\mathbf{n+\theta}}, \mathbf{V_h}), \quad \text{(B.13)}$$

$$(\mathbf{M}\delta_t(\mathbf{U_h}^{\mathbf{n}} + \mathbf{U'}^{\mathbf{n}}), \mathbf{V'}) + (\mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}} + \mathbf{U'}^{\mathbf{n+\theta}}), \mathbf{V'}) = (\mathbf{F}^{\mathbf{n+\theta}}, \mathbf{V'}). \quad \text{(B.14)}$$

Using Equations (B.6) and (B.7), the equation for sub-scales can be equivalently written as

$$((\frac{\mathbf{M}}{\theta\delta_t} + \mathscr{L})\mathbf{U'}^{\mathbf{n+\theta}}, \mathbf{V'}) = (\frac{\mathbf{M}}{\theta\delta_t}\mathbf{U'}^{\mathbf{n}}, \mathbf{V'}) + (\mathbf{F}^{\mathbf{n+\theta}} - [\mathbf{M}\delta_\mathbf{t}\mathbf{U_h}^{\mathbf{n}} + \mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}})], \mathbf{V'}). \quad \text{(B.15)}$$

The above equation is true for any function $\mathbf{V'} \in \mathscr{W}'$. Consequently the above equation can be re-written as

$$(\frac{\mathbf{M}}{\theta\delta_t} + \mathscr{L})\mathbf{U'}^{\mathbf{n+\theta}} = \frac{\mathbf{M}}{\theta\delta_t}\mathbf{U'}^{\mathbf{n}} + \mathbf{F}^{\mathbf{n+\theta}} - [\mathbf{M}\delta_\mathbf{t}\mathbf{U_h}^{\mathbf{n}} + \mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}})] + \mathbf{V}_{h,ort}, \quad \text{(B.16)}$$

The solution to Equation (B.16) can approximated by

$$\mathbf{U'^{n+\theta}} = \boldsymbol{\tau}_t(\frac{\mathbf{M}}{\theta\delta_t}\mathbf{U'^{n}} + \mathbf{F^{n+\theta}} - [\mathbf{M}\delta_\mathbf{t}\mathbf{U_h}^{\mathbf{n}} + \mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}})] + \mathbf{V}_{h,ort}). \tag{B.17}$$

To find an expression for $\mathbf{V}_{h,ort}$, the sub-scales are made orthogonal to the finite element space and the τ-Projection is approximated as L_2-projection resulting in the following expression:

$$\mathbf{V}_{h,ort} = -\Pi_h[\mathbf{F^{n+\theta}} - (\mathbf{M}\delta_t\mathbf{U_h}^{\mathbf{n}} + \mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}}))]. \tag{B.18}$$

Substituting back $\mathbf{V}_{h,ort}$ in Equation (B.17) we get

$$\mathbf{U'^{n+\theta}} = \boldsymbol{\tau}_t(\frac{\mathbf{M}}{\theta\delta_t}\mathbf{U'^{n}} + \Pi^{'}(\mathbf{F^{n+\theta}} - [\mathbf{M}\delta_\mathbf{t}\mathbf{U_h}^{\mathbf{n}} + \mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}})])), \tag{B.19}$$

where $\Pi^{'}$ =I$-\Pi_h$ projects it into the subspace orthogonal to the finite element subspace. Moreover, $\Pi^{'}(\mathbf{M}\delta_t\mathbf{U_h^n}) = 0$ because the two subspaces are orthogonal. This results in the following final expression for the sub-scales:

$$\mathbf{U'^{n+\theta}} = \boldsymbol{\tau}_t(\frac{\mathbf{M}}{\theta\delta_t}\mathbf{U^n} - \Pi'(\mathscr{L}(\mathbf{U_h}^{\mathbf{n+\theta}}))). \tag{B.20}$$

Substitution of equation (B.20) in the coarse equation (B.13) results in the following formulation

$$\begin{aligned}(\delta_t\mathbf{U_h}^n, \mathbf{V_h}) + (\mathbf{a}\cdot\nabla\mathbf{U_h}^{n+\theta}, \mathbf{V_h}) + \nu(\nabla\mathbf{U_h}^{n+\theta}, \nabla\mathbf{V_h}) - ({p_h}^{n+\theta}, \nabla\cdot\mathbf{V_h}) + (q_h, \nabla\cdot\mathbf{U_h}^{n+\theta}) + \\ (\mathbf{a}\cdot\nabla\mathbf{U_h}^{n+\theta} + \nabla{p_h}^{n+\theta}, \mathbf{a}\cdot\nabla\mathbf{V_h} + \nabla q_h)_{\tau_{1,t}} + (\nabla\cdot\mathbf{U_h^{n+\theta}}, \nabla\cdot\mathbf{V_h})_{\tau_2} = (\mathbf{f}, \mathbf{V_h}) \\ + \frac{1}{\theta\delta t}(\mathbf{u'}^n, \mathbf{a}\cdot\nabla\mathbf{V_h} + \nabla q_h)_{\tau_{1,t}} + (\xi_h, \mathbf{a}\cdot\nabla\mathbf{V_h} + \nabla q_h)_{\tau_{1,t}} + (\delta_h, \nabla\cdot\mathbf{V_h})_{\tau_2},\end{aligned} \tag{B.21}$$

where ξ_h and δ_h are given by

$$\xi_h = \Pi_h(\mathbf{a}\cdot\nabla\mathbf{U_h}^{n+\theta} + \nabla{p_h}^{n+\theta}), \tag{B.22}$$

$$\delta_h = \Pi_h(\nabla \cdot \mathbf{U_h}^{n+\theta}), \tag{B.23}$$

and the convective velocity **a** is defined as

$$\mathbf{a} = \mathbf{U_h}^{n+\theta} + \mathbf{u'}^{n+\theta}. \tag{B.24}$$

As suggested by Codina *Codina* (2002), equation (B.24) adds non-linearity to the formulation.

APPENDIX C

Numerical computation of L_2-projection.

The problem of finding an L_2 projection is equivalent to the problem of finding a $u_h \in \mathcal{V}_h$ such that

$$(u_h, w_h) = (u, w_h) \quad \forall w_h \in \mathcal{V}_h. \tag{C.1}$$

The first step is to determine the coarse space $\mathcal{V}_h$. The coarse space should be low-dimensional in comparison to the original solution to ensure that the projection operation acts as a filter. The low dimensionality of the coarse-space can be ensured by using lesser number of grid points or modes. There are many choices for the coarse space (for e.g, the Fourier basis functions, the global Chebyshev polynomial basis functions and the piece-wise polynomial basis functions). Once the coarse space is fixed, the coarse solution can be written as a linear combination of the basis functions as follows:

$$u_h = \mathbf{w}_h^T \mathbf{a}_h \tag{C.2}$$

where $\mathbf{w}_h^T$ is a vector of coarse-scale basis functions spanning the coarse-space and $\mathbf{a}_h$ is vector containing the corresponding basis coefficients. Substituting equation (C.2) in equation C.1 we obtain:

$$\mathbf{M}\mathbf{a}_h = \mathbf{r}, \tag{C.3}$$

where the mass matrix $\mathbf{M}$ and the right hand side vector $\mathbf{r}$ is given by,

$$\mathbf{M} = \int \mathbf{w}_h \mathbf{w}_h^T d\Omega, \text{ and } \mathbf{r} = (u, \mathbf{w}_h). \tag{C.4}$$

The coarse-scale basis coefficients are obtained as $\mathbf{a}_h = \mathbf{M}^{-1}\mathbf{r}$. The mass matrix $\mathbf{M}$ is local (block diagonal) when DG basis functions are used. In the case of CG basis functions, the mass matrix has to be assembled by adding contributions from individual element mass matrices. The computation of $\mathbf{M}$ is not as expensive as compared to the right hand side vector $\mathbf{r}$, especially when u is high dimensional. The elements of the matrix $\mathbf{M}$ can be precisely computed using a Gauss quadrature rule appropriate for the order of the polynomial used to define the coarse-space. The computation of $(u, \mathbf{w}_h)$, however, needs special care because it requires the computation of the inner-product of a high dimensional solution u with the coarse basis functions w_h as shown in figure C.1. The high dimensional solution u can come from a finite difference, finite volume, spectral or finite element simulation.

A more general approach to compute the elements of the right hand side vector $\mathbf{r}$ is by using numerical integration. As can be observed in figure C.1, the solution obtained after multiplication of the coarse basis functions $w_{h,i}$ with u still contains high-dimensional features and requires a fine-grid for numerical integration. The grid on which u exists is assumed to be sufficiently fine for performing the numerical integration. In case the projected solution depends on the order of numerical integration used or the grid size, the solution can be injected on a more finer grid to perform the numerical integration. Once the integration grid is set, the Trapezoidal rule or the Simpson's formula can be applied to compute the integral over the $uw_{h,i}$ fields to obtain the right hand side vector $\mathbf{r}$. In case u lies on an unstructured mesh, the integration of the $uw_{h,i}$ fields has to be performed over these fine unstructured elements. This can be done by first evaluating the coarse space on the unstructured mesh element and finally computing the integrals in the reference space of the unstructured mesh elements.

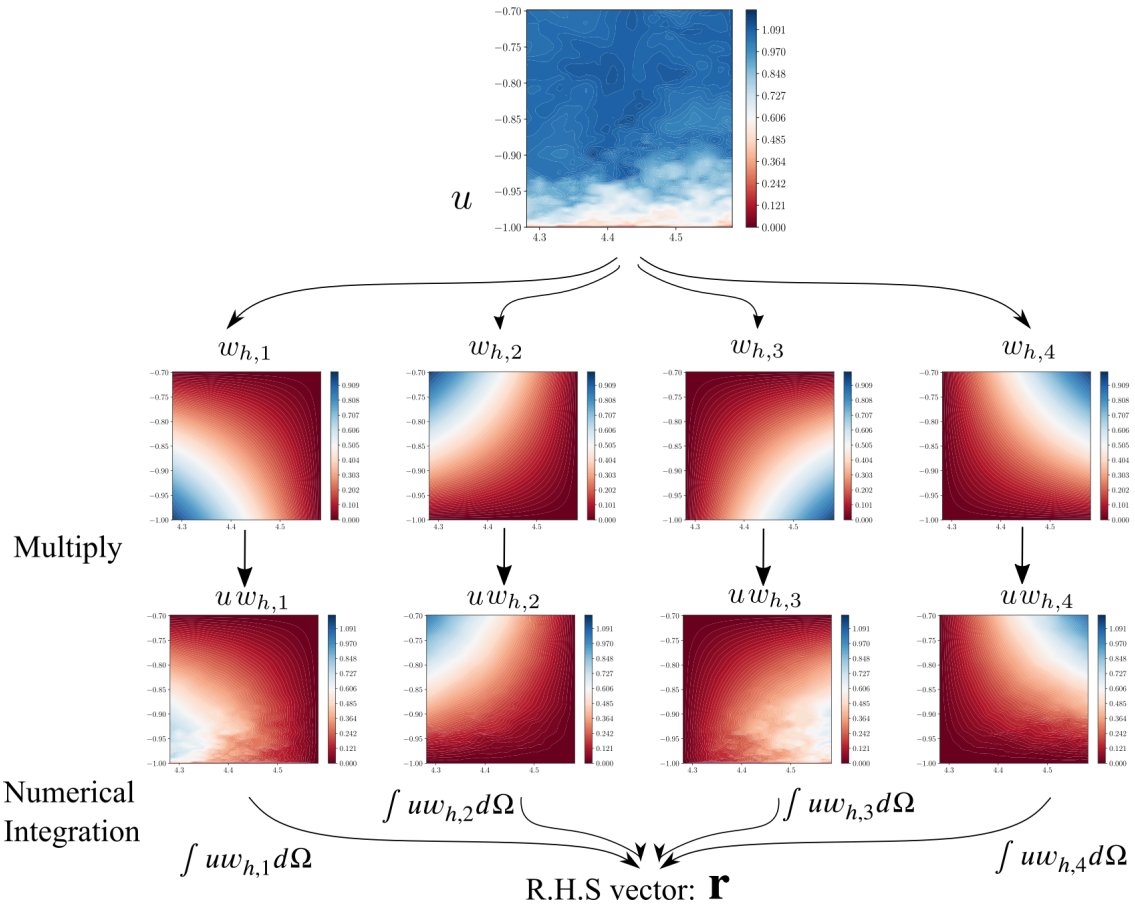


Figure C.1: The high dimensional solution u is multiplied with coarse-scale basis $w_{h,i}$ to obtain $uw_{h,i}$. The right hand side $\mathbf{r}$ is finally computed by evaluating $\int uw_{h,i}d\Omega$ for all basis function $w_{h,i}$ spanning the coarse space.

APPENDIX D

Filtering on unstructured meshes.

A majority of the work on explicitly-filtered LES is limited to structured grids where 1-D filters are applied repeatedly in each direction. Exceptions include *Bose et al.* (2011) and *Najafi-Yazdi et al.* (2015). The luxury of using dimensional splitting to perform 3-D filtering is not present in case of unstructured grids. An alternate approach is to create filters for unstructured grid is by using the neighbouring points as suggested by *Marsden et al.* (2002) and *Haselbacher and Vasilyev* (2003). In all these approaches, the basic idea is that a low dimensional interpolant is sought which is made to satisfy the high-dimensional data present at the grid points of the high dimensional data. This generally leads to an over-determined system and is solved using the least-squared approach. Another cost efficient filtering approach for unstructured grids is through the aggregation of neighboring cells to implement a top-hat filter (*Kim*, 2004).

The present projection-based approach is similar to existing approaches, except, the low dimensional space is a finite element solution having C^0 continuity at the element boundaries and it L_2-optimally satisfies the high-dimensional solution. As a result, the error is evaluated not just at the nodal points but across the domain. This high-dimensional solution can either come from a solution obtained on a finer grid or the DNS solution. The coarse finite element grid spanning the same computational domain can be un-structured

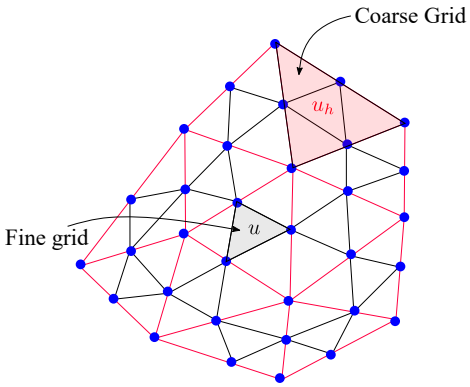


Figure D.1: The filtered solution on the coarse grid (red sides) can be obtained by L_2-projection of fine-grid solution (red and black sides).

and made of mesh elements like triangles, quadrilaterals or tetrahedrals as shown in figure D.1. The filter width $\Delta \approx \Delta_e / p$ in this case can be controlled using an approximate measure of the size of the coarse grid element Δ_e and the order of the polynomial p used for the shape function.

APPENDIX E

caslabDG: A high-order parallel DG-SEM code.

The *caslabDG* solver is an in-house parallel discontinuous Galerkin spectral element method (DG-SEM) code *Cockburn and Shu* (2001); *Reed and Hill* (1973); *Bassi and Rebay* (1997); *Hartmann and Houston* (2002) developed at CASLAB. The code is written in C and uses the MPI library for parallel computations on distributed-memory systems. The current implementation only supports a single structured block with variable order curved finite elements. Load balancing is performed by uniformly distributing Np_x,Np_y, and Np_z processors in the x,y, and z directions of the block, respectively, as shown in figure E.1. The governing equations that are solved by *caslabDG* are the compressible Navier-Stokes equations which can be written in the following form:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) - \nabla \cdot \mathbf{G}(\mathbf{u}, \nabla \mathbf{u}) = 0, \tag{E.1}$$

where $\mathbf{u} \in \mathbb{R}^5$ is a state vector of the conservative flow variables given by

$$\mathbf{u} = [\rho, \rho u, \rho v, \rho w, \rho E]^T. \tag{E.2}$$

In equation E.1, $\mathbf{F}$ and $\mathbf{G}$ denote the in-viscid and viscous fluxes corresponding to the governing equations for the state variables $\mathbf{u}$. In the present approach, the viscous flux

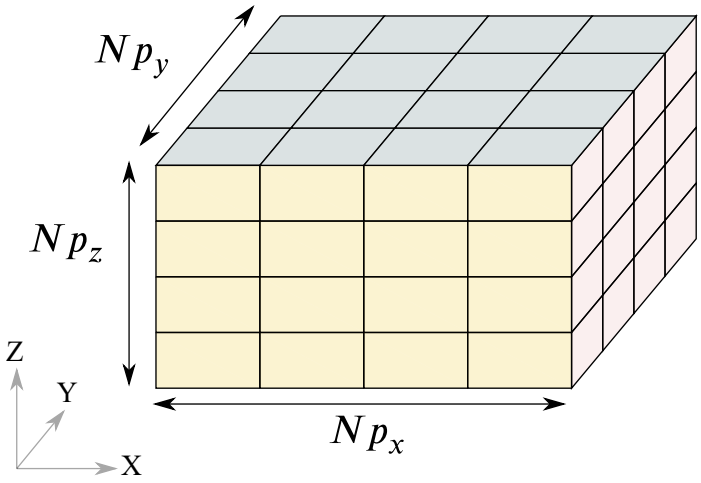


Figure E.1: Processors are distributed uniformly in each direction.

is assumed to be quasi-linear in the state gradients, i.e., $\mathbf{G}(\mathbf{u}, \nabla\mathbf{u}) = \mathbf{K}(\mathbf{u})\nabla\mathbf{u}$, where $\mathbf{K}$ denotes the diffusivity tensor. This assumption does not hold when an explicit sub-grid model is used where the eddy-viscosity is also a function of $\nabla\mathbf{u}$. In this work, to ensure that the existing discretization schemes for DG can be used with the explicit sub-grid models, we assume the quasi-linearity of $\mathbf{G}(\mathbf{u}, \nabla\mathbf{u}) = \mathbf{K}(\mathbf{u})\nabla\mathbf{u}$, even when $\mathbf{K}(\mathbf{u})$ is a function of $\nabla\mathbf{u}$. An alternative approach is to linearize the non-linear diffusive fluxes with respect to $\nabla\mathbf{u}$ as done in *Lv et al.* (2021).

Next, we perform discretization of equation (E.1) using the DG-SEM approach. This high-order numerical method allows the possibility of performing both h and p refinement of a mesh. The DG-SEM method makes it possible to obtain spectral-like accuracy on complex geometries, i.e., it keeps the flexibility of the finite volume method while allowing extremely accurate discretizations. The first step in applying the DG-SEM method is to tesselate the computational domain Ω into a set of N_e non-overlapping elements as follows:

$$\mathcal{T}_h = \{\Omega_e : \cup\Omega_e = \Omega, \cap\Omega_e = \emptyset\}. \tag{E.3}$$

The state is approximated by piece-wise polynomials lying on the approximation space $\mathcal{V}_\mathbf{h}$ having no continuity between adjacent elements. The approximation space $\mathcal{V}_\mathbf{h}$ is defined

as $\mathcal{V}_\mathbf{h} = [\mathcal{V}_h]^5$, where

$$\mathcal{V}_h = \{u \in L^2(\Omega) : u|\Omega_e \in \mathcal{P}^p, \forall \Omega_e \in \mathcal{T}_h\}, \tag{E.4}$$

and $\mathcal{P}^p$ denotes polynomials of order p on the reference space of element Ω_e. Currently, *caslabDG* only supports polynomials up to order $p = 4$. Next, the weak form of (E.1) is obtained by multiplying the equation (E.1) with the test functions present in the approximations space $\mathcal{V}_\mathbf{h}$ and performing integration by parts. Finally, the Galerkin discretization of the weak form is performed to obtain the following set of equations:

$$\begin{aligned}
&\int_{\Omega_e} \mathbf{w}_h^T \frac{\partial \mathbf{u_h}}{\partial t} d\Omega - \int_{\Omega} \nabla \mathbf{w}_h^T \cdot \left[\mathbf{F}(\mathbf{u}_h) - \mathbf{G}(\mathbf{u}_h, \nabla \mathbf{u}_h)\right] d\Omega \\
&\qquad + \int_{\partial\Omega} \mathbf{w}_h^T \left[\mathbf{F}(\mathbf{u}_h^+, \mathbf{u}_h^-) - \mathbf{G}(\mathbf{u}_h^+, \mathbf{u}_h^-, \nabla \mathbf{u}_h^+, \nabla \mathbf{u}_h^-)\right] \cdot \vec{n} dS \\
&- \int_{\partial\Omega_e} (\mathbf{u}_h^+ - \{\mathbf{u}_h\})^T \mathbf{G}(\mathbf{u}_h^+, \nabla \mathbf{w}_h^+) \cdot \vec{n} dS = \mathbf{0}, \quad \forall \mathbf{w}_h \in \mathcal{V}_\mathbf{h}.
\end{aligned} \tag{E.5}$$

On the element boundary Ω_e, the sub-scripts $(\cdot)^+$and $(\cdot)^-$ denote, respectively, the quantities taken from inside the element and its neighbor. The bracket $\{\cdot\}$ denotes the face/edge average or the boundary value, and $\hat{(\cdot)} \cdot \vec{n}$ represents the uniquely defined normal flux on element boundaries. Next, the actual fluxes are replaced with element coupling numerical fluxes. In *caslabDG*, we use the Roe approximate Riemann solver *Roe* (1981) to numerically approximate the inviscid flux $\mathbf{F}$, while for the viscous flux $\mathbf{G}$, we use the second form of Bassi and Rebay *Bassi and Rebay* (2000a), popularly known as the BR-2 scheme. The last term on the left-hand side (LHS) of (E.5) is to symmetrize the weak form to ensure the adjoint consistency of the final formulation. The stabilization parameter η in the BR-2 scheme was varied between 1.0 and 4.0 to ensure the stability of all the cases presented in this dissertation. This ends the spatial discretization of the PDE.

Finally, writing $\mathbf{u}_h$ in terms of the basis functions $\mathbf{w}_h$ and the corresponding basis

coefficients $\mathbf{U}_h$, equation (E.5) can be rewritten as a system of ordinary differential equations (ODEs) as follows:

$$\frac{d\mathbf{U}_h}{dt} + \mathbf{f}_h(\mathbf{U}_h) = \mathbf{0}, \tag{E.6}$$

where $\mathbf{f}_h = \mathbf{M}_h^{-1}\mathbf{R}_h(\mathbf{U}_h)$, $\mathbf{U}_h \in \mathbf{R}^{N_h}$ represents the discrete vector of the unknown basis function coefficients, $\mathbf{M}_h$ is the global mass matrix which is block diagonal, and $\mathbf{R}_h$ is the discrete spatial residual vector. Due to the block diagonal nature of the global mass matrix $\mathbf{M}_h$, its inverse $\mathbf{M}_h^{-1}$ can be evaluated by computing locally the inverse of the mass matrix for each element. Equation E.6 can now be discretized in time with a method of our choice. Currently, *caslabDG* only supports the explicit third-order Runge-Kutta TVD (RK3-TVD) and the fourth-order Runge-Kutta (RK4) time-stepping schemes. All the simulations presented in this dissertation have been performed using the Runge-Kutta TVD (RK3-TVD) scheme.

E.1 Parallel performance.

This part evaluates the parallel performance of *caslabDG*. The parallel performance is evaluated through a strong scaling test on a 32^3 element mesh for two different polynomial orders $p = 2, 3$. The size of this grid is comparable to most of the cases presented in this thesis. Hence, the present test indicates how effectively the computational resources have been used for this work. These tests have been performed on NASA's Pleiades supercomputer on the Intel Broadwell nodes with 28 processors. Figure E.2 shows the speed-up vs. processors curve for the two cases. It can be observed that near-optimal performance was observed for both cases for the range of processors considered here. However, the performance became slightly sub-linear as the 512 processor mark was approached.

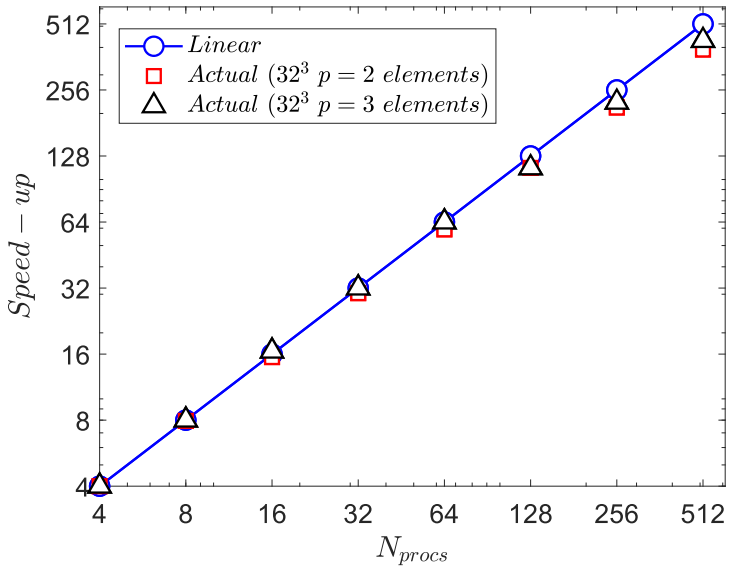


Figure E.2: Effect of p-refinement on the strong scaling performance of *caslabDG*.

E.2 A novel implict-explicit WMLES formulation.

WMLES has been previously performed within the DG-SEM framework by *Frère et al.* (2017); *Lv et al.* (2021); *Carton de Wiart and Murman* (2017). In all these works, implicit LES was performed with an implicit time-stepping scheme, or an explicit time-stepping scheme was used with an explicit sub-grid model (Smagorinsky or VREMAN). We found that the simulation was not stable when WMLES was performed using an explicit time stepping scheme without any explicit sub-grid model (implicit LES). Indeed, implicit time-stepping schemes can improve stability; however, explicit schemes benefit WMLES because the grid size can be much coarser than WRLES, making the time step less restrictive. In addition, implementing implicit methods requires developing non-linear solvers, linear solvers, and pre-conditioners. An alternate fix is to use an explicit sub-grid model *Lv et al.* (2021). However, the DG methods already have good implicit-LES properties, even outperforming explicit LES models in some cases.

As a first step to resolving this problem, the cause of the instability was identified, which was tracked down to the near-wall elements. Hence, explicit sub-grid modeling is performed

only at the first element in our formulation. The model coefficient decays linearly to zero at the start of the second element to ensure that the explicit sub-grid model turns off outside the first element. This is shown in figure E.3, where implicit LES is performed at the center of the channel, and explicit LES is performed at the first element. The resulting method has two benefits: (i.) it is stable with explicit time-stepping schemes, and (ii.) minimal dissipation at the center of the channel for high-order discretizations. The explicit sub-grid model used at the first element is the constant coefficient Smagorinsky model, which has a free parameter in the form of a model coefficient C_s. The effect of the model coefficient is tested by varying from $C_s = 0.1 - 0.2$ and found to have minimal effect on the solution. In addition to the model coefficient, explicit models also require the specification of grid size, which is slightly tricky in the case of the DG method. In this work, the adequate grid size is taken to be:

$$\Delta_p = \frac{\Delta}{p} \tag{E.7}$$

where p is the polynomial order and Δ is the cube root of the cell volume.

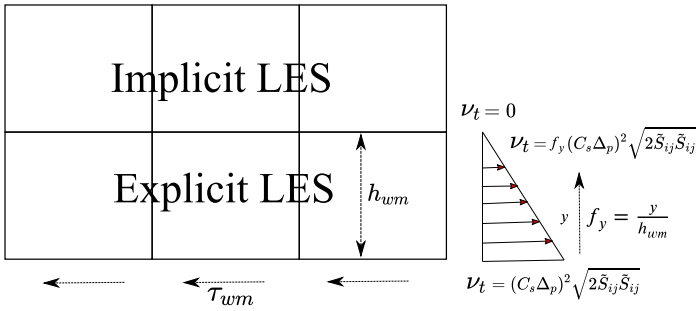


Figure E.3: An implicit-explicit approach is used to stabilize the WMLES method for DG.

To validate our model, we perform WMLES of channel flow at $Re_\tau \approx 8000$. Two different resolutions are considered: (i.) a low resolution case with $12 \times 12 \times 12$ elements; (ii) a high resolution case with $24 \times 16 \times 24$ elements. In both cases, the size of the first element in the wall-normal direction has been ensured to have a size of $\Delta y \approx 0.1\delta$. The channel size is taken to be $2\pi\delta$, $\pi\delta$, and 2δ in the stream-wise (x), span-wise (z), and wall-normal (y) directions, respectively, for all the cases. For each resolution, WMLES is

performed using two different polynomial orders of $p = 2, 3$. Figure E.4 and E.5 shows velocity statistics obtained for $p = 2$ and $p = 3$ for the low resolution mesh. Similarly, Figure E.6 and E.7 shows velocity statistics obtained for $p = 2$ and $p = 3$ for the high resolution mesh. Normalized errors are computed for different types of statistics using the following expression:

$$E_s = \frac{|\mathbf{u}_s^{WM} - \mathbf{u}_s^{DNS}|}{|\mathbf{u}_s^{DNS}|} \tag{E.8}$$

where $\mathbf{u}_s^{WM}$ and $\mathbf{u}_s^{WM}$ denote the statistics of a particular type obtained from the WMLES and DNS at the quadrature points, respectively, and $|.|$ denotes the l_2-norm. A common observation is that when resolution is increased by increasing the order ($p = 2$ to $p = 3$) or the resolution $N_x \times N_y \times N_z$ ($12 \times 12 \times 12$ to $24 \times 16 \times 24$), the normalized error in mean velocity profile U^+, the wall-normal velocity R.M.S v^+_{rms}, the span-wise velocity R.M.S w^+_{rms} and the turbulent shear stress $< uv >^+$ decreases. However, the normalized error in the stream-wise velocity R.M.S u^+_{rms} increases slightly and becomes constant. These results indicate that the traditional equilibrium wall-stress-based models is inconsistent, i.e., refinement does not always lead to better results in all the quantities, especially the stream-wise velocity fluctuations u^+_{rms}. This inconsistency is not a problem specific to the present solver and is a problem with the traditional WMLES approaches in general. Irrespective, the performance was satisfactory for all the orders and resolutions.

The validated model is finally applied to a slightly more complex case: the periodic hill. The periodic hill case is very similar to the channel flow, i.e., periodic boundary conditions in the stream-wise and span-wise directions and no slip in the top and bottom walls. However, in this case, the bottom wall is modified to mimic a periodic hill pattern. Due to this pattern, the flow separates from the back side of the hill. The flow again re-attaches downstream, thus, creating a sizeable re-circulation bubble. Once the flow re-attaches, it flows over the hill again. This pattern is repeated due to the periodic nature of the stream-wise boundary condition. To test our wall-modeling approach, we simulate the moderately high Reynolds number with $Re_b \approx 37000$. The Reynolds number is defined based on the hill height h and

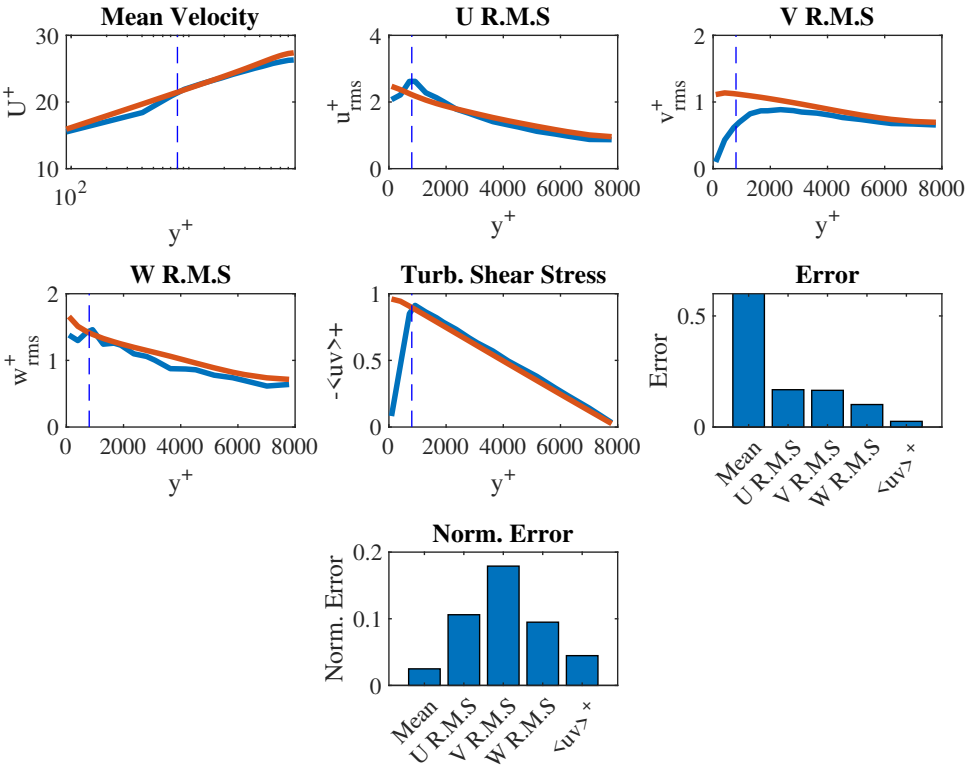


Figure E.4: $Re_\tau \approx 8000$: Low-resolution $p = 2$.

the bulk velocity at the crest. The Mach number based on this bulk velocity is set to 0.2 to keep the compressibility effects minimal while at the same time ensuring that the time-step is not too restrictive. As shown in figure E.9, the x-axis, y-axis, and z-axis are made to align with the stream-wise, span-wise, and wall-normal directions, respectively, w.r.t to the bottom wall between the hills (different from the channel flow cases).

As shown in E.8, two different grids C1 and C2, are used for all the numerical experiments presented here. Figure E.9 shows the plot of the conservative variables for the coarsest case, i.e., grid C1 discretized using polynomial order $p = 2$ basis functions. As expected, it can be observed from figure E.9(a.) that a separation region is present at the back side of the hill, which eventually re-attaches further downstream. A comparison between the mean velocity profiles obtained on the meshes C1 and C2 using different orders of discretization is shown in E.10. Additionally, the experimental measurements and the $p = 7$

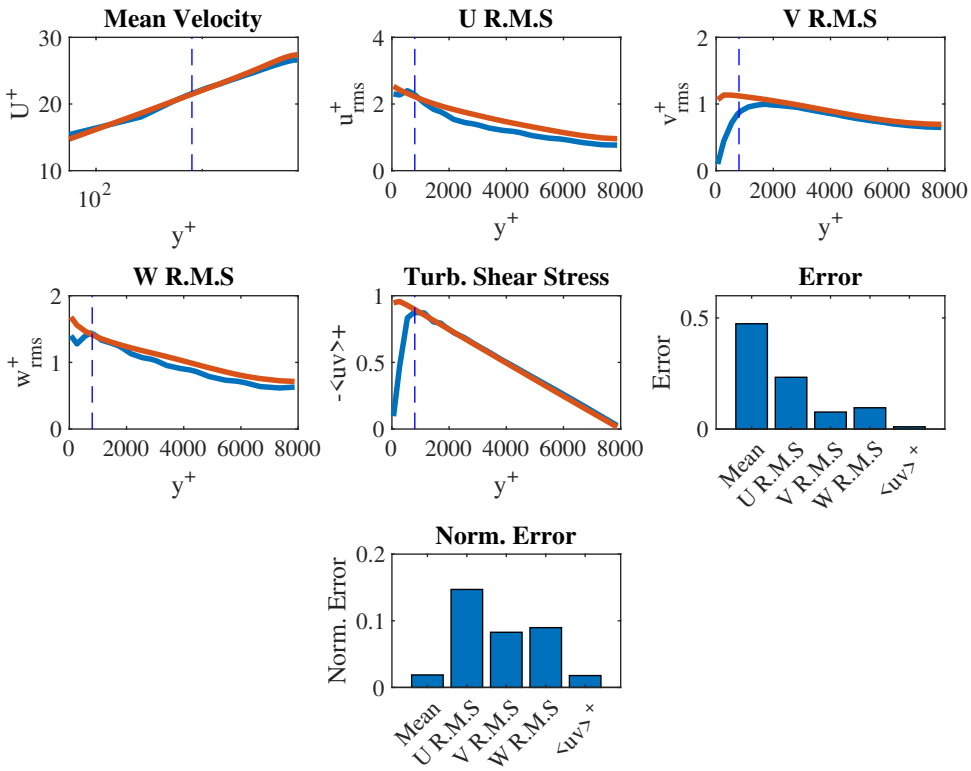


Figure E.5: $Re_\tau \approx 8000$: Low-resolution $p = 3$.

high resolution WMLES solution from *eddy* E.10 are also plotted. It is observed that the performance on the C2 grid is slightly better than on the C1 grid. It is further observed that as the DOFs are increased by increasing the discretization order p, the WMLES solutions from *caslabDG* approach the experimental observation, and the high-resolution WMLES results from *eddy*. In all the WMLES cases, the solution after the separated region is slightly incorrect, suggesting that the present wall-stress-based WMLES methods must be improved for separated flows.

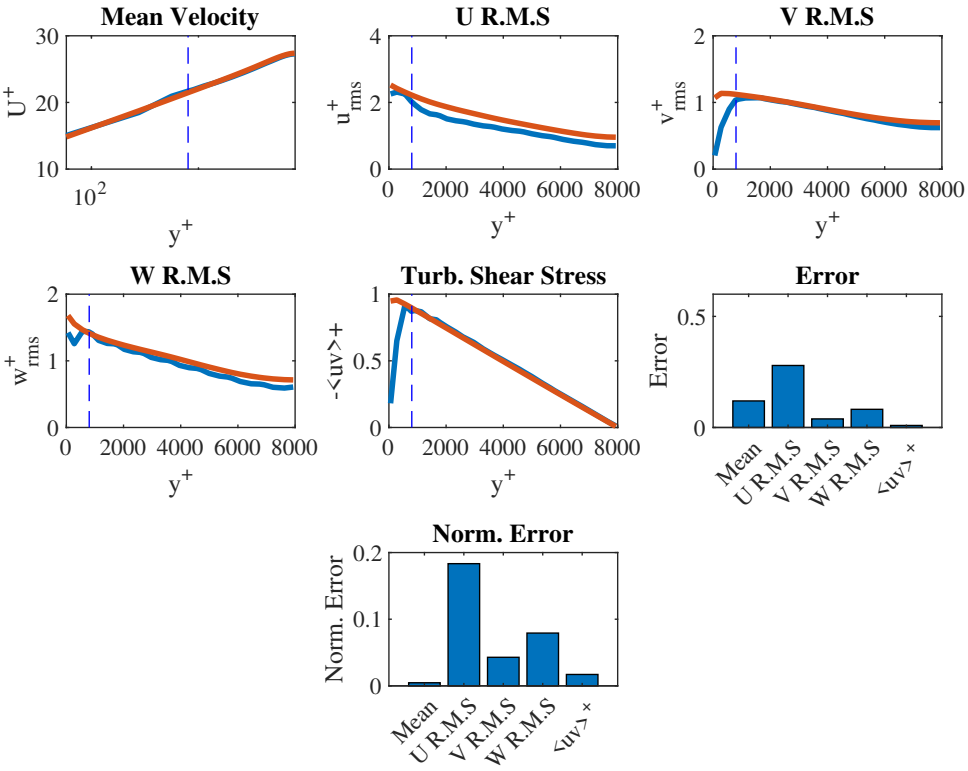


Figure E.6: $Re_\tau \approx 8000$: High-resolution $p = 2$.

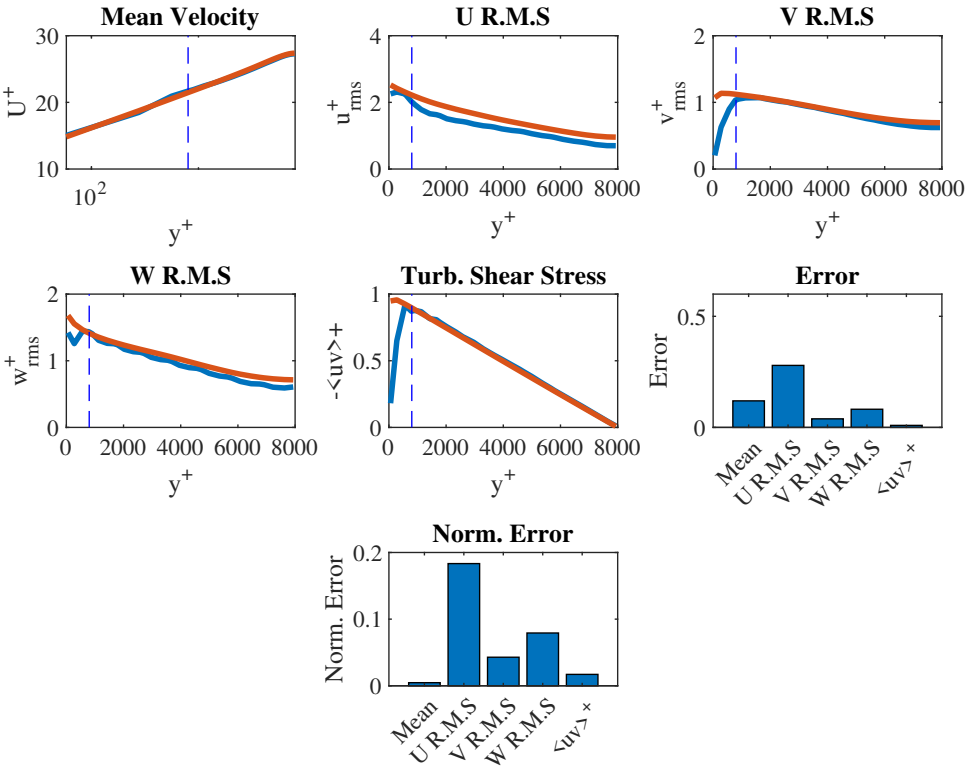


Figure E.7: $Re_\tau \approx 8000$: High-resolution $p = 3$.

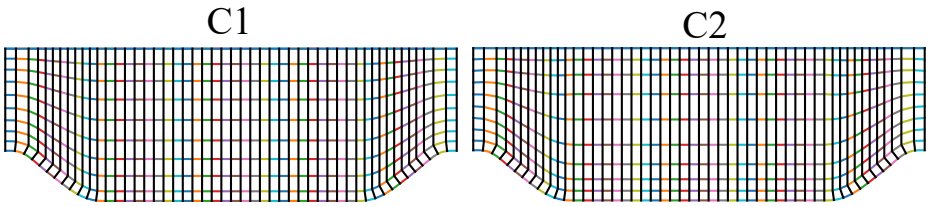


Figure E.8: Two different grids are used for WMLES of the periodic hill case: C1 and C2. The grid C2 has a lower stretching compared to C1 at the bottom wall and a smaller wall normal grid size at the top wall. The grid C1 has a smaller grid size at the center of the channel compared to C2.

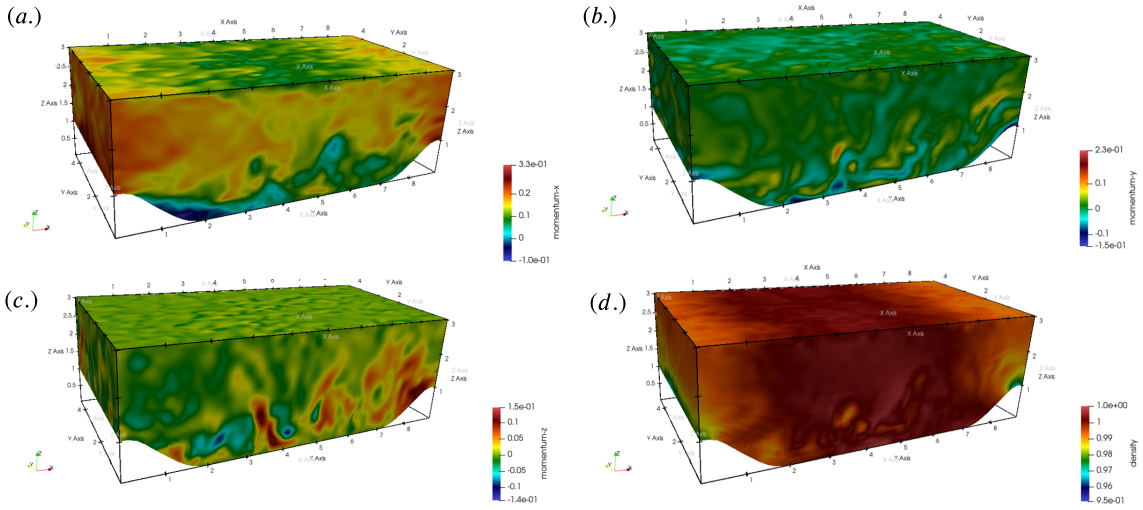


Figure E.9: Snapshots from WMLES simulation of the periodic hill case at $Re_b \approx 37000$ at $Ma \approx 0.2$ (based on the bulk velocity). Solution is computed using $N_x \times N_y \times N_z$:$50 \times 9 \times 24$ elements with polynomial order $p = 2$ (DOFs: $150 \times 27 \times 72$). The x-axis, y-axis, and z-axis are aligned with the stream-wise, span-wise, and wall-normal directions, respectively, w.r.t to the bottom wall between the hills. (a.) stream-wise momentum, (b.) span-wise momentum and (c.) wall-normal momentum, and (d.) density.

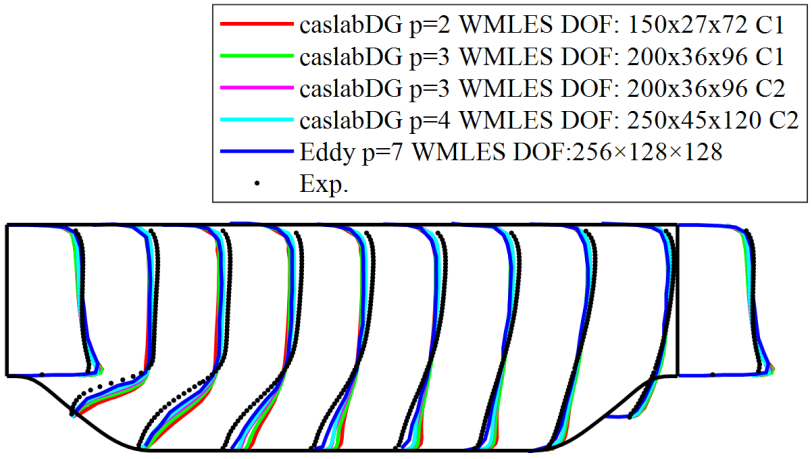


Figure E.10: Mean velocity profiles from WMLES of the periodic hill case at $Re_b \approx 37000$ for different meshes and polynomial orders. The x-axis, y-axis, and z-axis are aligned with the stream-wise, span-wise, and wall-normal directions, respectively, w.r.t to the bottom wall between the hills.

BIBLIOGRAPHY

BIBLIOGRAPHY

Akkerman, I., K. van der Zee, and S. Hulshoff (2010), A variational germano approach for stabilized finite element methods, *Computer methods in applied mechanics and engineering*, *199*(9-12), 502–513.

Bae, H. J., A. Lozano-Durán, S. T. Bose, and P. Moin (2019), Dynamic slip wall model for large-eddy simulation, *Journal of fluid mechanics*, *859*, 400–432.

Bardina, J., J. Ferziger, and W. Reynolds (1980), Improved subgrid-scale models for large-eddy simulation, in *13th fluid and plasmadynamics conference*, p. 1357.

Bassi, F., and S. Rebay (1997), A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, *Journal of Computational Physics*, *131*(2), 267–279, doi:10.1006/jcph.1996.5572.

Bassi, F., and S. Rebay (2000a), GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations, in *Lecture Notes in Computational Science and Engineering*, pp. 197–208, Springer Berlin Heidelberg, doi:10.1007/978-3-642-59721-3_14.

Bassi, F., and S. Rebay (2000b), GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations, in *Discontinuous Galerkin Methods*, pp. 197–208, Springer.

Bazilevs, Y., V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi (2007), Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, *Computer Methods in Applied Mechanics and Engineering*, *197*(1-4), 173–201.

Beck, A., D. Flad, and C.-D. Munz (2019), Deep neural networks for data-driven les closure models, *Journal of Computational Physics*, *398*, 108,910.

Bose, S., P. Moin, and F. Ham (2011), Explicitly filtered large eddy simulation on unstructured grids, *Center of Turbulence Research Annual Research Briefs*, pp. 87–96.

Bose, S. T., and P. Moin (2014), A dynamic slip boundary condition for wall-modeled large-eddy simulation, *Physics of Fluids*, *26*(1), 015,104.

Bou-Zeid, E., N. Vercauteren, M. B. Parlange, and C. Meneveau (2008), Scale dependence of subgrid-scale model coefficients: an a priori study, *Physics of Fluids*, *20*(11), 115,106.

Brezzi, F., M.-O. Bristeau, L. P. Franca, M. Mallet, and G. Rogé (1992), A relationship between stabilized finite element methods and the galerkin method with bubble functions, *Computer Methods in Applied Mechanics and Engineering*, *96*(1), 117–129.

Brooks, A. N., and T. J. Hughes (1982), Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Computer methods in applied mechanics and engineering*, *32*(1-3), 199–259.

Carlberg, K., M. Barone, and H. Antil (2017), Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction, *Journal of Computational Physics*, *330*, 693–734.

Carton de Wiart, C., and S. M. Murman (2017), Assessment of wall-modeled les strategies within a discontinuous-galerkin spectral-element framework, in *55th AIAA Aerospace Sciences Meeting*, p. 1223.

Chollet, F., et al. (2015), Keras, `https://keras.io`.

Chorin, A. J., and O. H. Hald (2009), *Stochastic tools in mathematics and science*, vol. 3, Springer.

Chorin, A. J., O. H. Hald, and R. Kupferman (2000), Optimal prediction and the mori–zwanzig representation of irreversible processes, *Proceedings of the National Academy of Sciences*, *97*(7), 2968–2973.

Chorin, A. J., O. H. Hald, and R. Kupferman (2002), Optimal prediction with memory, *Physica D: Nonlinear Phenomena*, *166*(3-4), 239–257.

Chung, S. W., and J. B. Freund (2022), An optimization method for chaotic turbulent flow, *Journal of Computational Physics*, *457*, 111,077.

Cockburn, B., and C.-W. Shu (2001), Runge–kutta discontinuous Galerkin methods for convection-dominated problems, *Journal of Scientific Computing*, *16*(3), 173–261, doi: 10.1023/a:1012873910884.

Codina, R. (2000), On stabilized finite element methods for linear systems of convection–diffusion-reaction equations, *Computer Methods in Applied Mechanics and Engineering*, *188*(1-3), 61–82.

Codina, R. (2002), Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, *Computer Methods in Applied Mechanics and Engineering*, *191*(39-40), 4295–4321.

Codina, R., J. Principe, O. Guasch, and S. Badia (2007), Time dependent subscales in the stabilized finite element approximation of incompressible flow problems, *Computer Methods in Applied Mechanics and Engineering*, *196*(21-24), 2413–2430.

Colomés, O., S. Badia, R. Codina, and J. Principe (2015), Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows, *Computer Methods in Applied Mechanics and Engineering*, *285*, 32–63.

Comte-Bellot, G., and S. Corrsin (1971), Simple eulerian time correlation of full-and narrow-band velocity signals in grid-generated,'isotropic'turbulence, *Journal of Fluid Mechanics*, *48*(2), 273–337.

Deardorff, J. W. (1970), A numerical study of three-dimensional turbulent channel flow at large reynolds numbers, *Journal of Fluid Mechanics*, *41*(2), 453–480.

Del Alamo, J. C., J. Jiménez, P. Zandonade, and R. D. Moser (2004), Scaling of the energy spectra of turbulent channels, *Journal of Fluid Mechanics*, *500*, 135–144.

Deng, Z., C. He, Y. Liu, and K. C. Kim (2019), Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework, *Physics of Fluids*, *31*(12), 125,111.

Donea, J., and A. Huerta (2003), *Finite element methods for flow problems*, John Wiley & Sons.

Duraisamy, K. (2020), Machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence, *arXiv preprint arXiv:2009.10675*.

Duraisamy, K. (2021), Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence, *Physical Review Fluids*, *6*(5), 050,504.

Duraisamy, K., G. Iaccarino, and H. Xiao (2019), Turbulence modeling in the age of data, *Annual Review of Fluid Mechanics*, *51*, 357–377.

Franca, L. P., and C. Farhat (1995), Bubble functions prompt unusual stabilized finite element methods, *Computer Methods in Applied Mechanics and Engineering*, *123*(1-4), 299–308.

Franca, L. P., S. L. Frey, and T. J. Hughes (1992), Stabilized finite element methods: I. application to the advective-diffusive model, *Computer Methods in Applied Mechanics and Engineering*, *95*(2), 253–276.

Frère, A., C. Carton de Wiart, K. Hillewaert, P. Chatelain, and G. Winckelmans (2017), Application of wall-models to discontinuous galerkin les, *Physics of Fluids*, *29*(8), 085,111.

Friess, C., and L. Davidson (2020), A formulation of pans capable of mimicking iddes, *International Journal of Heat and Fluid Flow*, *86*, 108,666.

Fukami, K., K. Fukagata, and K. Taira (2019), Super-resolution reconstruction of turbulent flows with machine learning, *Journal of Fluid Mechanics*, *870*, 106–120.

Fukami, K., K. Fukagata, and K. Taira (2021), Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows, *Journal of Fluid Mechanics*, *909*.

Gamahara, M., and Y. Hattori (2017), Searching for turbulence models by artificial neural network, *Physical Review Fluids*, *2*(5), 054,604.

Germano, M. (1986), Differential filters of elliptic type, *The Physics of fluids*, *29*(6), 1757–1758.

Germano, M. (1992), Turbulence: the filtering approach, *Journal of Fluid Mechanics*, *238*, 325–336.

Germano, M., U. Piomelli, P. Moin, and W. H. Cabot (1991), A dynamic subgrid-scale eddy viscosity model, *Physics of Fluids A: Fluid Dynamics*, *3*(7), 1760–1765.

Girimaji, S., and K. Abdol-Hamid (2005), Partially-averaged navier stokes model for turbulence: Implementation and validation, in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, p. 502.

Goc, K., S. Bose, and P. Moin (2020), Wall-modeled large eddy simulation of an aircraft in landing configuration, in *AIAA Aviation 2020 Forum*, p. 3002.

Goc, K. A., O. Lehmkuhl, G. I. Park, S. T. Bose, and P. Moin (2021), Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics, *Flow*, *1*.

Gopalan, H., S. Heinz, and M. K. Stöllinger (2013), A unified rans–les model: Computational development, accuracy and cost, *Journal of Computational Physics*, *249*, 249–274.

Gouasmi, A., E. Parish, and K. Duraisamy (2017), A priori estimation of memory effects in coarse-grained nonlinear systems using the mori-zwanzig formalism.

Gravemeier, V., M. W. Gee, M. Kronbichfler, and W. A. Wall (2010), An algebraic variational multiscale–multigrid method for large eddy simulation of turbulent flow, *Computer Methods in Applied Mechanics and Engineering*, *199*(13-16), 853–864.

Hartmann, R., and P. Houston (2002), Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, *Journal of Computational Physics*, *183*(2), 508–532, doi:10.1006/jcph.2002.7206.

Haselbacher, A., and O. V. Vasilyev (2003), Commutative discrete filtering on unstructured grids based on least-squares techniques, *Journal of Computational Physics*, *187*(1), 197–211.

Heinz, S. (2007), Unified turbulence models for les and rans, fdf and pdf simulations, *Theoretical and Computational Fluid Dynamics*, *21*(2), 99–118.

Holland, J. R., J. D. Baeder, and K. Duraisamy (2019), Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training, in *AIAA Aviation 2019 Forum*, p. 3200.

Hoyas, S., and J. Jiménez (2006), Scaling of the velocity fluctuations in turbulent channels up to re τ= 2003, *Physics of fluids*, *18*(1), 011,702.

Hughes, T. J. (1995), Multiscale phenomena: Green's functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, *Computer methods in applied mechanics and engineering*, *127*(1-4), 387–401.

Hughes, T. J., L. P. Franca, and M. Balestra (1986), A new finite element formulation for computational fluid dynamics: V. circumventing the babuška-brezzi condition: a stable petrov-galerkin formulation of the stokes problem accommodating equal-order interpolations, *Computer Methods in Applied Mechanics and Engineering*, *59*(1), 85–99.

Hughes, T. J., L. P. Franca, and G. M. Hulbert (1989), A new finite element formulation for computational fluid dynamics: Viii. the galerkin/least-squares method for advective-diffusive equations, *Computer methods in applied mechanics and engineering*, *73*(2), 173–189.

Hughes, T. J., G. R. Feijóo, L. Mazzei, and J.-B. Quincy (1998a), The variational multiscale method—a paradigm for computational mechanics, *Computer methods in applied mechanics and engineering*, *166*(1-2), 3–24.

Hughes, T. J., G. R. Feijóo, L. Mazzei, and J.-B. Quincy (1998b), The variational multiscale method—a paradigm for computational mechanics, *Computer methods in applied mechanics and engineering*, *166*(1-2), 3–24.

Hughes, T. J., L. Mazzei, and K. E. Jansen (2000), Large eddy simulation and the variational multiscale method, *Computing and visualization in science*, *3*(1-2), 47–59.

Ishida, T., P. Davidson, and Y. Kaneda (2006), On the decay of isotropic turbulence, *Journal of Fluid Mechanics*, *564*, 455–475.

Iyer, P. S., and M. R. Malik (2020), Wall-modeled les of the nasa juncture flow experiment, in *AIAA Scitech 2020 Forum*, p. 1307.

Kawai, S., and J. Larsson (2012), Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy, *Physics of Fluids*, *24*(1), 015,105.

Kim, H., J. Kim, S. Won, and C. Lee (2020), Unsupervised deep learning for super-resolution reconstruction of turbulence, *arXiv preprint arXiv:2007.15324*.

Kim, S.-E. (2004), Large eddy simulation using an unstructured mesh based finite-volume solver, in *34th AIAA fluid dynamics conference and exhibit*, p. 2548.

Kingma, D. P., and J. Ba (2014), Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.

Kiris, C. C., A. S. Ghate, J. C. Duensing, O. M. Browne, J. A. Housman, G.-D. Stich, G. Kenway, L. M. Dos Santos Fernandes, and L. M. Machado (2022), High-lift common research model: Rans, hrles, and wmles perspectives for clmax prediction using lava, in *AIAA SCITECH 2022 Forum*, p. 1554.

Krank, B., and W. A. Wall (2016), A new approach to wall modeling in les of incompressible flow via function enrichment, *Journal of Computational Physics*, *316*, 94–116.

Langford, J. A., and R. D. Moser (1999a), Optimal LES formulations for isotropic turbulence, *Journal of fluid mechanics*, *398*, 321–346.

Langford, J. A., and R. D. Moser (1999b), Optimal LES formulations for isotropic turbulence, *Journal of fluid mechanics*, *398*, 321–346.

Lee, M., and R. D. Moser (2015), Direct numerical simulation of turbulent channel flow up to, *Journal of fluid mechanics*, *774*, 395–415.

Li, Y., E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, and G. Eyink (2008), A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence, *Journal of Turbulence*, (9), N31.

Lilly, D. K. (1966), On the application of the eddy viscosity concept in the inertial sub-range of turbulence, *NCAR manuscript*, *123*.

Ling, J., A. Kurzawski, and J. Templeton (2016), Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *Journal of Fluid Mechanics*, *807*, 155–166.

Liu, B., J. Tang, H. Huang, and X.-Y. Lu (2020), Deep learning methods for super-resolution reconstruction of turbulent flows, *Physics of Fluids*, *32*(2), 025,105.

Liu, S., C. Meneveau, and J. Katz (1994), On the properties of similarity subgrid-scale models as deduced from measurements in a turbulent jet, *Journal of Fluid Mechanics*, *275*, 83–119.

Liu, S., C. Meneveau, and J. Katz (1995), Experimental study of similarity subgrid-scale models of turbulence in the far-field of a jet, *Applied scientific research*, *54*(3), 177–190.

Lozano-Durán, A., and J. Jiménez (2014), Effect of the computational domain on direct simulations of turbulent channels up to re_τ= 4200, *Physics of Fluids*, *26*(1), 011,702.

Lozano-Duran, A., S. T. Bose, and P. Moin (2020), Prediction of trailing edge separation on the nasa juncture flow using wall-modeled les, in *AIAA Scitech 2020 Forum*, p. 1776.

Lund, T. (2003), The use of explicit filters in large eddy simulation, *Computers & Mathematics with Applications*, *46*(4), 603–616.

Lv, Y., X. I. Yang, G. I. Park, and M. Ihme (2021), A discontinuous galerkin method for wall-modeled large-eddy simulations, *Computers & Fluids*, *222*, 104,933.

Mansour, N., and A. Wray (1994), Decay of isotropic turbulence at low reynolds number, *Physics of Fluids*, *6*(2), 808–814.

Marsden, A. L., O. V. Vasilyev, and P. Moin (2002), Construction of commutative filters for les on unstructured meshes, *Journal of Computational Physics*, *175*(2), 584–603.

Masud, A., and R. Calderer (2011a), A variational multiscale method for incompressible turbulent flows: Bubble functions and fine scale fields, *Computer Methods in Applied Mechanics and Engineering*, *200*(33-36), 2577–2593.

Masud, A., and R. Calderer (2011b), A variational multiscale method for incompressible turbulent flows: Bubble functions and fine scale fields, *Computer Methods in Applied Mechanics and Engineering*, *200*(33-36), 2577–2593.

Maulik, R., and O. San (2017), A neural network approach for the blind deconvolution of turbulent flows, *Journal of Fluid Mechanics*, *831*, 151–181.

Maulik, R., O. San, A. Rasheed, and P. Vedula (2018), Data-driven deconvolution for large eddy simulations of Kraichnan turbulence, *Physics of Fluids*, *30*(12), 125,109.

Maulik, R., O. San, J. D. Jacob, and C. Crick (2019), Sub-grid scale model classification and blending through deep learning, *Journal of Fluid Mechanics*, *870*, 784–812.

McMillan, O. J., and J. H. Ferziger (1979), Direct testing of subgrid-scale models, *Aiaa Journal*, *17*(12), 1340–1346.

Meneveau, C., and J. Katz (2000), Scale-invariance and turbulence models for large-eddy simulation, *Annual Review of Fluid Mechanics*, *32*(1), 1–32.

Meneveau, C., T. S. Lund, and W. H. Cabot (1996), A lagrangian dynamic subgrid-scale model of turbulence, *Journal of fluid mechanics*, *319*, 353–385.

Mohebujjaman, M., L. G. Rebholz, and T. Iliescu (2019), Physically constrained data-driven correction for reduced-order modeling of fluid flows, *International Journal for Numerical Methods in Fluids*, *89*(3), 103–122.

Mori, H. (1965), Transport, collective motion, and brownian motion, *Progress of theoretical physics*, *33*(3), 423–455.

Mou, C., B. Koc, O. San, L. G. Rebholz, and T. Iliescu (2021), Data-driven variational multiscale reduced order models, *Computer Methods in Applied Mechanics and Engineering*, *373*, 113,470.

Najafi-Yazdi, A., M. Najafi-Yazdi, and L. Mongeau (2015), A high resolution differential filter for large eddy simulation: Toward explicit filtering on unstructured grids, *Journal of Computational Physics*, *292*, 272–286.

Nicoud, F., and F. Ducros (1999), Subgrid-scale stress modelling based on the square of the velocity gradient tensor, *Flow, turbulence and Combustion*, *62*(3), 183–200.

Nicoud, F., H. B. Toda, O. Cabrit, S. Bose, and J. Lee (2011), Using singular values to build a subgrid-scale model for large eddy simulations, *Physics of Fluids*, *23*(8), 085,106.

Oberai, A., and J. Wanderer (2005), Variational formulation of the germano identity for the navier–stokes equations, *Journal of Turbulence*, (6), N7.

Orszag, S. A., and G. Patterson Jr (1972), Numerical simulation of three-dimensional homogeneous isotropic turbulence, *Physical Review Letters*, *28*(2), 76.

Parish, E. (2018), Variational multiscale modeling and memory effects in turbulent flow simulations.

Parish, E. J., and K. Duraisamy (2016), A paradigm for data-driven predictive modeling using field inversion and machine learning, *Journal of Computational Physics*, *305*, 758–774.

Parish, E. J., and K. Duraisamy (2017a), Non-markovian closure models for large eddy simulations using the mori-zwanzig formalism, *Physical Review Fluids*, *2*(1), 014,604.

Parish, E. J., and K. Duraisamy (2017b), A dynamic subgrid scale model for large eddy simulations based on the mori–zwanzig formalism, *Journal of Computational Physics*, *349*, 154–175.

Parish, E. J., and K. Duraisamy (2017c), A unified framework for multiscale modeling using the Mori-Zwanzig formalism and the variational multiscale method, *arXiv preprint arXiv:1712.09669*.

Parish, E. J., C. Wentland, and K. Duraisamy (2018), A residual-based petrov-galerkin reduced-order model with memory effects, *arXiv preprint arXiv:1810.03455*.

Park, G., and P. Moin (2016), Wall-modeled les: Recent applications to complex flows, *Annual Research Briefs*, pp. 39–50.

Perlman, E., R. Burns, Y. Li, and C. Meneveau (2007), Data exploration of turbulence simulations using a database cluster, in *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing*, pp. 1–11.

Piomelli, U., and E. Balaras (2002), Wall-layer models for large-eddy simulations, *Annual review of fluid mechanics*, *34*(1), 349–374.

Pope, S. B. (2000), *Turbulent flows*, Cambridge university press.

Pope, S. B., and S. B. Pope (2000), *Turbulent flows*, Cambridge university press.

Pradhan, A., and K. Duraisamy (2020), Variational multiscale closures for finite element discretizations using the Mori-Zwanzig approach, *Computer Methods in Applied Mechanics and Engineering*, *368*, 113,152.

Pradhan, A., and K. Duraisamy (2021), Variational multi-scale super-resolution: A data-driven approach for reconstruction and predictive modeling of unresolved physics, *arXiv preprint arXiv:2101.09839*.

Pradhan, A., and K. Duraisamy (2022), A unified understanding of scale-resolving simulations and near-wall modeling of turbulent flows using optimal finite element projections, *arXiv preprint arXiv:2207.13060*.

Reed, W., and T. Hill (1973), Triangular mesh methods for the neutron transport equation, *Tech. rep.*, Los Alamos Scientific Lab, available: https://www.osti.gov/servlets/purl/4491151.

Roe, P. (1981), Approximate Riemann solvers, parameter vectors, and difference schemes, *Journal of Computational Physics*, *43*(2), 357–372, doi:10.1016/0021-9991(81)90128-5.

Rogallo, R. S. (1981), Numerical experiments in homogeneous turbulence.

Rowley, C. W., T. Colonius, and R. M. Murray (2004), Model reduction for compressible flows using pod and galerkin projection, *Physica D: Nonlinear Phenomena*, *189*(1-2), 115–129.

Sarghini, F., G. De Felice, and S. Santini (2003), Neural networks based subgrid scale modeling in large eddy simulations, *Computers & fluids*, *32*(1), 97–108.

Shur, M., P. Spalart, M. Strelets, and A. Travin (1999), Detached-eddy simulation of an airfoil at high angle of attack, in *Engineering turbulence modelling and experiments 4*, pp. 669–678, Elsevier.

Shur, M. L., P. R. Spalart, M. K. Strelets, and A. K. Travin (2008), A hybrid rans-les approach with delayed-des and wall-modelled les capabilities, *International journal of heat and fluid flow*, *29*(6), 1638–1649.

Singh, A. P., and K. Duraisamy (2016), Using field inversion to quantify functional errors in turbulence closures, *Physics of Fluids*, *28*(4), 045,110.

Sirignano, J., J. MacArt, and J. Freund (2020), Embedded training of neural-network sub-grid-scale turbulence models, *Bulletin of the American Physical Society*.

Slotnick, J. P., A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. J. Mavriplis (2014), Cfd vision 2030 study: a path to revolutionary computational aerosciences, *Tech. rep.*

Smagorinsky, J. (1963), General circulation experiments with the primitive equations: I. the basic experiment, *Monthly weather review*, *91*(3), 99–164.

Spalart, P. R. (2009), Detached-eddy simulation, *Annual review of fluid mechanics*, *41*, 181–202.

Stinis, P. (2007), Higher order Mori–Zwanzig models for the euler equations, *Multiscale Modeling & Simulation*, *6*(3), 741–760.

Stinis, P. (2012), Mori-zwanzig reduced models for uncertainty quantification i: Parametric uncertainty, *arXiv preprint arXiv:1211.4285*.

Stinis, P. (2013), Renormalized reduced models for singular pdes, *Communications in Applied Mathematics and Computational Science*, *8*(1), 39–66.

Stinis, P. (2015), Renormalized mori–zwanzig-reduced models for systems without scale separation, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *471*(2176), 20140,446.

Stolz, S., and N. A. Adams (1999), An approximate deconvolution procedure for large-eddy simulation, *Physics of Fluids*, *11*(7), 1699–1701.

Stolz, S., N. A. Adams, and L. Kleiser (2001), An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows, *Physics of fluids*, *13*(4), 997–1015.

Tezduyar, T. E. (2001), Adaptive determination of the finite element stabilization parameters, in *Proceedings of the ECCOMAS computational fluid dynamics conference*, pp. 1–17.

Vreman, A. (2004), An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications, *Physics of fluids*, *16*(10), 3670–3681.

Vreman, B., B. Geurts, and H. Kuerten (1994), On the formulation of the dynamic mixed subgrid-scale model, *Physics of Fluids*, *6*(12), 4057–4059.

Vreman, B., B. Geurts, and H. Kuerten (1995), A priori tests of large eddy simulation of the compressible plane mixing layer, *Journal of engineering mathematics*, *29*(4), 299–327.

Wang, J.-X., J.-L. Wu, and H. Xiao (2017), Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on dns data, *Physical Review Fluids*, *2*(3), 034,603.

Wang, L., R. Hu, and X. Zheng (2020a), A comparative study on the large-scale-resolving capability of wall-modeled large-eddy simulation, *Physics of Fluids*, *32*(3), 035,102.

Wang, Q., N. Ripamonti, and J. S. Hesthaven (2020b), Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism, *Journal of Computational Physics*, *410*, 109,402.

Wang, Z., and A. Oberai (2010a), Spectral analysis of the dissipation of the residual-based variational multiscale method, *Computer Methods in Applied Mechanics and Engineering*, *199*(13-16), 810–818.

Wang, Z., and A. Oberai (2010b), A mixed large eddy simulation model based on the residual-based variational multiscale formulation, *Physics of Fluids*, *22*(7), 075,107.

Wang, Z., K. Luo, D. Li, J. Tan, and J. Fan (2018), Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation, *Physics of Fluids*, *30*(12), 125,101.

Whitmore, M., K. Griffin, S. Bose, and P. Moin (2021), Large-eddy simulation of a gaussian bump with slip-wall boundary conditions, *Center for Turbulence Research Annual Research Briefs*, pp. 45–58.

Xie, C., K. Li, C. Ma, and J. Wang (2019a), Modeling subgrid-scale force and divergence of heat flux of compressible isotropic turbulence by artificial neural network, *Physical Review Fluids*, *4*(10), 104,605.

Xie, C., J. Wang, H. Li, M. Wan, and S. Chen (2019b), Artificial neural network mixed model for large eddy simulation of compressible isotropic turbulence, *Physics of Fluids*, *31*(8), 085,112.

Xie, C., J. Wang, and E. Weinan (2020), Modeling subgrid-scale forces by spatial artificial neural networks in large eddy simulation of turbulence, *Physical Review Fluids*, *5*(5), 054,606.

Xie, X., M. Mohebujjaman, L. G. Rebholz, and T. Iliescu (2018a), Data-driven filtered reduced order modeling of fluid flows, *SIAM Journal on Scientific Computing*, *40*(3), B834–B857.

Xie, Y., E. Franz, M. Chu, and N. Thuerey (2018b), tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow, *ACM Transactions on Graphics (TOG)*, *37*(4), 1–15.

Xu, J., A. Pradhan, and K. Duraisamy (2021), Conditionally parameterized, discretization-aware neural networks for mesh-based modeling of physical systems, *Advances in Neural Information Processing Systems*, *34*, 1634–1645.

Yang, X., S. Zafar, J.-X. Wang, and H. Xiao (2019), Predictive large-eddy-simulation wall modeling via physics-informed neural networks, *Physical Review Fluids*, *4*(3), 034,602.

Yang, X. I., and K. P. Griffin (2021), Grid-point and time-step requirements for direct numerical simulation and large-eddy simulation, *Physics of Fluids*, *33*(1), 015,108.

You, D., and P. Moin (2007a), A dynamic global-coefficient subgrid-scale eddy-viscosity model for large-eddy simulation in complex geometries, *Physics of Fluids*, *19*(6), 065,110.

You, D., and P. Moin (2007b), A dynamic global-coefficient subgrid-scale eddy-viscosity model for large-eddy simulation in complex geometries, *Physics of Fluids*, *19*(6), 065,110.

Zhu, Y., and D. Venturi (2018), Faber approximation of the mori-zwanzig equation, *Journal of Computational Physics*.

Zwanzig, R. (1980), Problems in nonlinear transport theory, in *Systems far from equilibrium*, pp. 198–225, Springer.