EXTREME-SCALE COMPUTING AND STUDIES OF INTERMITTENCY, MIXING OF PASSIVE SCALARS AND STRATIFIED FLOWS IN TURBULENCE

A Dissertation
Presented to
The Academic Faculty

By

Kiran Ravikumar

In Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

December 2021

EXTREME-SCALE COMPUTING AND STUDIES OF INTERMITTENCY, MIXING OF PASSIVE SCALARS AND STRATIFIED FLOWS IN TURBULENCE

Thesis committee:

Professor P. K. Yeung, Advisor
Schools of Aerospace & Mechanical Engineering
Georgia Institute of Technology

Professor Richard Vuduc
School of Computational Science & Engineering
Georgia Institute of Technology

Professor Devesh Ranjan
Schools of Mechanical & Aerospace Engineering
Georgia Institute of Technology

Professor K. R. Sreenivasan
Tandon School of Engineering, Graduate School of Arts and Sciences, and Courant Institute of Mathematical Sciences
New York University

Professor Suresh Menon
School of Aerospace Engineering
Georgia Institute of Technology

Date approved: July 30, 2021

To explain all nature is too difficult a task for any one man or even for any one age. 'Tis much better to do a little with certainty and leave the rest for others that come after you.

Isaac Newton

To my loving parents, Usha and Ravikumar, and wife, Vinaya.

ACKNOWLEDGMENTS

This work was supported by funding from Oak Ridge National Laboratory (ORNL). The research used computational resources of the Oak Ridge Leadership Computing Facility (OLCF), which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725 and at the Texas Advanced Computing Center (TACC) at The University of Texas at Austin. I also gratefully acknowledge computational resources granted through the Summit early science award, INCITE 2019 award, Frontera early science award and Frontier Center for Accelerated Application Readiness (CAAR) program.

I am very grateful and humbled by the support and encouragement I have received from many individuals throughout my graduate study at Georgia Tech. First, I would like to thank my advisor Professor P. K. Yeung for taking me on as a student and introducing me to the field of High Performance Computing, which I have greatly fallen in love with. His constant support and encouragement has kept me motivated to tackle new and challenging problems. Most importantly, I am in awe of his research philosophy; that is to scrutinize and question the results in order to obtain a thorough understanding of the science topic being studied. My experience working with him will greatly benefit my career in the long run. I am also grateful to Professor K. R. Sreenivasan who also serves in the thesis committee. I am privileged to have worked with him on the topic of multifractals during which time contents in chapter four of this thesis progressed to completion. I would also like to thank Professors Suresh Menon, Devesh Ranjan and Richard Vuduc for taking the time to be on my thesis committee, and for providing valuable feedback and comments which have improved the quality of this work.

I am extremely grateful to Dr. David Appelhans who motivated the development of a new algorithm to target extremely large-scale problem sizes. Our collaboration started with exchanging a few ideas during a hackathon event at ORNL. Little did I know that working and learning from him will lead to a highly successful project resulting in record-

setting simulations and a paper at the Supercomputing 2019 conference. I am indebted to him for teaching me various concepts in programming Graphical Processing Units and CUDA Fortran. I also thank him for the opportunity to intern at IBM Research. Much of the work would not be possible without support from Dr. Oscar Hernandez. His constant encouragement to explore new ideas in computing have been very helpful. I have learnt a lot about compiler environments and OpenMP programming interface from him. He is always ready to answer my questions and help figure out bugs in my code. Most importantly, he has helped me with numerous opportunities to present at leading conferences and workshops. I would like to thank the many other expert HPC consultants who helped me along the way: Stephen Nichols and Reuben Budiardja of OLCF, Jeff Larkin of OLCF and NVIDIA, Lars Koesterke of TACC, and John Levesque of HPE.

I have had a pleasant time conducting research at Georgia Tech, and I thank Prof. Dimitri Mavris for granting me admission in the first place. Special thanks to lab mates, Dr. Matthew Clay, Dr. Dhawal Buaria, Dr. Xiaomeng Zhai, Bichuan Mo and Rohini Vaideswaran, for interesting discussions and help with everything. I was fortunate to find an amazing group of friends to spend time with outside of Tech. I would like to extend special thanks to Dr. Dushhyanth Rajaram and Mr. Devan Thanki for all the fun times while pursuing our respective degrees.

Most importantly, I would like to thank my family. First, my parents, Usha and Ravikumar, who have always supported me in all my academic and extracurricular endeavors. I have learnt to always strive for perfection and never give up and to be humble and grounded during success while remaining focused and persevere through tough times. Second, my wife, Vinaya, for her unwavering support throughout the years. She has been a calming influence, especially when I was stressed and anxious. I always look to her for advice and learn from her positive outlook to life. They have always stood firmly by my side and have played a major part in shaping who I am today. I would also like to thank my grandparents. Their stories of overcoming hardships have been an inspiration to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis presents results from high-resolution numerical simulations focusing on understanding the fundamental behavior of isotropic turbulence and mixing at high Reynolds numbers. Direct numerical simulations are performed using Fourier pseudo-spectral methods on leadership-class supercomputers to reach world-leading problem sizes, exceeding the prior state-of-the-art. A key emphasis is on the development of a novel computational algorithm that can simulate flows at very high Reynolds numbers and resolutions using modern heterogeneous platforms consisting of both traditional Central Processing Unit (CPU) and Graphical Processing Units (GPU). The resulting simulations provide insights into the behavior of localized high-amplitude extreme events in velocity and passive scalar fields, including the related subject of multifractal properties of turbulent flows. The simulations reported in this thesis add to a major turbulence simulation database that can be used to address many unresolved questions in turbulence theory.

Turbulent flows are well known for the intermittent occurrence of intense strain rates and local rotation acting on individual fluid elements. The intensities of these "extreme events" increase with Reynolds number. For accurate results, the small scales must be well-resolved. This places stringent resolution requirements on numerical simulations at ever-larger problem sizes. In this work, an advanced GPU algorithm employing the principle of batched asynchronism has been developed to target dense node heterogeneous architecture machines like Summit at Oak Ridge National Laboratory (ORNL), which at a theoretical peak of 200 petaflops/second is currently the second fastest supercomputer in the world. Key elements enabling high performance in this code include aggressive optimizations of data copies between CPU and GPU, network communication, and overlapping data copies and computations on the GPU. Limitations on problem size due to the smaller memory are avoided by processing data residing on the CPU in batches on the GPU while overlapping data copies and computations. Favorable performance as measured by wallclock time and

GPU speedup is obtained up to a world-leading resolution of 18432^3 (over 6 trillion grid points) on 3072 Summit nodes (two thirds of full system). The algorithm was further extended to track Lagrangian fluid particles. Significant effort has also been expended towards a more portable implementation using advanced provisions of the latest OpenMP programming interface, to be deployed on an exascale computer named Frontier to arrive at ORNL by early 2022, with 32768^3 resolution as the target. Challenges encountered in ensuring interoperability between OpenMP and GPU libraries are discussed.

Although the new algorithm developed allows high resolution simulations, high computational demands in part due to time-stepping requirements have resulted in such simulations often being short, which raises questions about the adequacy of statistical sampling and independence. The second topic in this thesis is the development of a new approach to address these simulations. In this new paradigm, referred to as "Multiple Resolution Independent Simulations" (MRIS), ensemble averaging is performed over multiple short simulation segments at high resolution, each evolving from lower-resolution snapshots possessing a strong degree of statistical independence by virtue of being well separated in time. Using this approach, various aspects of small-scale intermittency are studied through statistics of dissipation rate and enstrophy, where high resolution in both space and time helps capture extreme fluctuations in the velocity gradients accurately. Specifically, moments of 3D local averages of energy dissipation and enstrophy are computed and their behavior in the inertial range are analyzed to identify power-law scalings predicted by Kolmogorov's refined similarity theory. Conditional moments formed between these two quantities show that, while extreme energy dissipation is typically accompanied by extreme enstrophy, the reverse is not always true.

The best-resolved simulation datasets at several Reynolds numbers are used to study intermittency in the energy dissipation rate within a multifractal framework. While multifractality has been investigated experimentally at least three decades ago, results at high resolution and Reynolds numbers and using energy dissipation computed exactly using all

nine velocity gradients are not commonly found in the literature. Probability density functions of energy dissipation with tails modeled as stretched exponential are used to compute the multifractal spectrum. The results obtained are found to exhibit only a weak dependence on Reynolds number within the range considered in the present work. Results on the multifractal spectrum also indicates the existence of negative fractal dimensions corresponding to regions of very high dissipation rate. Studies of incipient singularities of different strengths are carried out to understand the energy dissipation contribution and volume occupied by such singularities in the flow at different Reynolds numbers. A related issue is the relative contributions of dissipation rate of different intensities to moments at various orders. The results show that moderate to high intensity regions ($\mathcal{O}(100)$ times the mean) contribute a significant fraction to the higher order moments while lower order moments are dominated by energy dissipation a few times ($\mathcal{O}(10)$) the mean.

The MRIS approach is also employed to study intermittency in turbulent mixing, through statistics of the dissipation rate of passively transported scalar quantities. Simulations with scalars of moderate molecular diffusivity are performed in the presence of a uniform mean gradient at moderate Reynolds numbers and high resolution. Statistics of scalar gradients show that the rate of return to local isotropy with increasing Reynolds number is slow. Local averages of scalar dissipation rate were also analyzed and power-law behaviors were observed. The results also show that stricter resolution requirements are necessary for the scalar field compared to the velocity field.

The final topic of the thesis focuses on the study of stratified flows observed in many atmospheric and oceanic settings, where density variations due to the presence of active scalars affect the behavior of the flow as well. Both stable and unstable stratification are considered. To understand the energy cascade and anisotropy development, the Reynolds-stress budget is computed in a time-resolved manner. Results show that the flow energetics are determined by energy dissipation and buoyancy flux, which can either be opposing each other or acting simultaneously to suppress the large fluctuations.

CHAPTER 1
INTRODUCTION

Turbulent flows, characterized with fluctuations in flow properties occurring over a wide range of scales (Pope, 2000), especially at high Reynolds number, are very common in science and engineering (pollutant dispersion, combustion, ocean dynamics, astrophysics, to name a few). Given that turbulence occurs in such diverse contexts, considerable effort has been expended in pursuit of universal properties of turbulence, starting with Kolmogorov's path-breaking work in 1941 (Kolmogorov, 1941). Turbulence is well known for the intermittent occurrence of intense strain rates and local rotation acting on individual fluid elements, and for its ability to provide efficient mixing, which is, for instance, essential in aircraft engines. In applications, high strain rate can lead to combustion instabilities, while high rotation rate (or vorticity) can lead to preferential concentration of inertial particles in multi-phase flows. Fluctuations representing the small scales can be extreme, with samples of order 1000 times or higher compared to averaged values. An accurate understanding of small-scale intermittency (Kolmogorov, 1962; Sreenivasan & Antonia, 1997; Frisch, 1995) is thus of both fundamental and practical interest.

A powerful strategy for fundamental understanding is to compute all the scales according to exact equations, where hypotheses of scale similarity suggest simplified boundary conditions are appropriate when the small scales are of the greatest interest. In this thesis, direct numerical simulation (DNS) using a Fourier pseudo-spectral approach (Patterson & Orszag, 1971; Canuto *et al.*, 1988) on a simple cubic domain with periodic boundary conditions is used as the primary research tool. Because of the wide range of scales that need to be resolved, DNS is computationally intensive. Indeed, turbulence simulations using DNS have been a well-recognized grand challenge in high performance computing for many years (Yokokawa *et al.*, 2002). With the evolution of computing power by over 1 million

times in the past 25 years, simulations involving of order 1 trillion grid points or more have become a reality (Lee *et al.*, 2013; Yeung *et al.*, 2015; Ishihara *et al.*, 2016). Most of these works have exploited massive parallelism, where performance ultimately tends to be limited by the cost of communication among tens or hundreds of thousands of parallel processors. However, as computing approaches the Exascale era, with heterogeneous architectures based on Graphical Processing Units (GPUs) being increasingly dominant, a major re-think is necessary.

In pseudo-spectral methods (Canuto *et al.*, 1988), nonlinear terms are evaluated in physical space and then transformed to Fourier space, avoiding extremely costly convolution integrals. These simulations are inherently communication intensive because of the need to collect complete lines of data in the machine memory before transforms can be taken. Until 2019, the largest simulations have been achieved via massive CPU-based parallelism. However, trends in the latest machine architecture towards the exascale (theoretical limit at 10^{18} floating point operations per second) appear to favor nodes with multiple cores, accelerators and large memory (dense nodes) where new programming approaches to optimize on-node and off-node data transfer become essential. For example, Summit at the Oak Ridge Leadership Computing Facility (OLCF), has fewer but much denser nodes than its predecessor machine (Titan). The upcoming exascale machines, Frontier at OLCF and Aurora at Argonne Leadership Computing Facility (ALCF) will have a similar dense node heterogeneous architecture as well. Utilizing GPUs instead of CPUs for very large problem sizes also presents new challenges, since the amount of GPU memory is substantially less than CPU memory.

The first topic of this thesis addresses the nontrivial task of implementing a new pseudo-spectral turbulence code capable of reaching unprecedented problem sizes at the high throughput needed to complete high resolution simulations on Summit. The shift to fewer but denser nodes motivates a design strategy of hierarchical parallelism, with a fine grained parallelism mapped to GPU threads coupled to a high level parallelism managed by fewer

MPI processes compared to traditional massive parallelism via CPUs.

The overall strategy, especially at scale, is based on the expected needs to (1) improve MPI performance and to (2) use GPUs efficiently for large problem sizes. To improve MPI performance, the communication overhead can be reduced by using fewer MPI processes, which is well facilitated by Summit's configuration as a modest number of dense nodes. The local problem size per MPI process can be increased by utilizing the larger CPU memory without being restricted by the smaller GPU memory. A hybrid MPI+OpenMP approach, where MPI expresses distributed memory parallelism across nodes and OpenMP expresses shared memory parallelism within a node, enables further reduction in the number of MPI ranks for the same problem size. In addition, the promise of overlapping host-based communication using nonblocking MPI collectives was studied, which provided good but not the best performance.

The considerations above tend to reduce communication overhead and hence make the code more readily amenable to the benefits of GPU acceleration. In order to utilize the GPUs for large problem sizes, the new algorithm uses a batching strategy, with the data being processed on the GPU in smaller GPU-sized pieces. The costs of data movement and computations on the GPUs are aggressively minimized. This is achieved by, firstly, enabling asynchronism to overlap computation and data movement on different batches of data. Secondly, using custom data movement kernels that are highly efficient at performing strided copies on the device. And finally, leveraging optimized NVIDIA libraries on the GPU to accelerate computations (e.g. cuFFT for 1-D FFTs). This approach is relevant to the broader question of how computations (such as 3D FFTs) that are inherently communication-intensive can be re-imagined to benefit from new architectures whose principal advantage is fast computation.

The significant role of communication in the application implies that neither sustained flop rate nor scalability are the most relevant performance metrics. Instead, the scalability of the new code is compared against a code performing only 3D transpose all-to-all

calls. The performance of the two are comparable with only a small difference to account for data copies and computations. The new algorithm makes it possible to perform simulations of three-dimensional isotropic turbulence at a problem size of 18432^3 (6 trillion) grid points, which (within some constraints discussed further in chapter 2) is the largest problem size that is feasible on Summit for production purposes. This new grid resolution is expected to be instrumental in further advances into fundamental understanding of turbulence, especially those which are highly dependent on the presence of a wide range of scales that are represented on a finite solution domain with higher accuracy than previously practiced in the literature. The CPU to GPU speedup recorded is over 4.5X for a problem size equivalent to the largest reported in the literature prior to this work. This work was presented as a Best Student Paper Finalist (Ravikumar *et al.*, 2019) at the Supercomputing 2019 conference.

Along with simulations of the velocity field alone, the newly developed batched asynchronous code can simulate both passive and active scalars in the presence of mean gradients. Passive scalars evolving according to the advection-diffusion equation can be readily handled in the pseudo-spectral framework, with the diffusion term being treated exactly through an integrating factor, similar to the viscous term for velocity. However, the two-way coupling terms between the velocity and scalar field, when tracking active scalars, introduces challenges in numerical resolution. To address these challenges, the system of governing equations in wavenumber space may be diagonalized such that the two-way coupling terms can also be treated exactly through an integrating factor.

The GPU DNS algorithm outlined above provides the values of velocity fluctuations at a set of N^3 grid points in three-dimensional (3D) space. While this so-called Eulerian framework is natural as far as partial differential equations are concerned, there is also much interest in studying turbulence using a Lagrangian description from the viewpoint of an observer moving with the instantaneous flow. The Lagrangian perspective (Yeung, 2002) is of vital importance in the study of turbulent dispersion of passive contaminants in

air quality modeling, the air- or water-borne transmission of disease agents, among many other examples. To obtain Lagrangian information requires an ability to track the motion of infinitesimal fluid particles, which is readily achieved in DNS through the calculation of particle velocity via interpolation at the grid points (Yeung & Pope, 1988). A particle tracking algorithm based on cubic spline interpolation has been added to the GPU DNS code and tested to perform with high efficiency on Summit. Cubic spline interpolation consists of two major operations, namely (1) the generation of cubic spline coefficients and (2) the use of these coefficients for the velocity of a large number of particles. The first of these is of an Eulerian, grid-based nature, and hence amenable to a GPU implementation using the principle of batched asynchronism above. The second operation carries a cost that is proportional to the particle count N_p, which is small compared to N^3. The highly scalable implementation described later in this thesis has been shown to be very effective in minimizing communication costs, thus leading to a GPU speedup even higher than that for the Eulerian velocity field alone.

The code development efforts on Summit, as described above, have relied heavily on CUDA Fortran to target the NVIDIA GPUs. However, future exascale architectures such as Frontier at OLCF and Aurora at ALCF make use of AMD and Intel GPUs respectively, while CUDA, which is specific to programming NVIDIA GPUs, will not be supported. Therefore, a portable implementation of this algorithm is essential to run on different hardware with minimal changes. The OpenMP programming model is a widely accepted standard, with more recent versions supporting advanced target offload features which are not well-supported on Summit. Recent code development efforts have focused on implementing the batched asynchronous algorithm using OpenMP to target GPUs. Advanced OpenMP 5.0 features like DETACH and OpenMP TASKs are used to ensure the desired asynchronous behavior is achieved. An important goal is to allow the future portable OpenMP-enabled GPU code to scale up to resolution as high as 32768^3 on the exascale machine Frontier when it becomes available.

In turbulence, fluctuations can arise over a wide range of scales, that increases with Reynolds number. It is important to be able to quantify, or predict, how fluctuations at different scale sizes behave and interact with one another. For example, while the velocity fluctuations (dominated by the large scales) are usually close to Gaussian distributed, the fluctuating velocity gradients, are highly non-Gaussian and prone to very large departures from the mean. These regions of intense straining and rotation, quantified by the energy dissipation rate (ϵ) and enstrophy (Ω) respectively, are short-lived in time and highly localized in space. Capturing these so-called "extreme events" accurately places stringent resolution requirements on the numerical simulations. The classical Kolmogorov 1941 scale-similarity hypothesis (Kolmogorov, 1941) has provided a good starting point in the search for a universal behavior. However, intermittency as reflected in the statistical properties of ϵ_r, the energy dissipation rate averaged over a scale size r (Frisch, 1995), must be addressed. The statistics of locally averaged dissipation rate have a key role in intermittency corrections in turbulence theory (Kolmogorov, 1962), in particular in the inertial range, which itself requires high Reynolds number. Simulations are conducted using the successfully developed numerical algorithm, capable of high Reynolds number and resolution, necessary for accurate studies of fine-scale intermittency.

The quality of results from any DNS depends on the numerical methods used, the degree to which both large scales and small scales are faithfully represented, in both time and space, as well as the adequacy of statistical sampling. For flows with a stationary state, the conventional approach to ensure good sampling is by running a long simulation for say $O(10)$ eddy-turnover times (T_E, defined to be the ratio of a longitudinal integral length scale to the r.m.s. fluctuation of a velocity component), performing post-processing on data saved at regular time intervals, and finally averaging over multiple realizations. However, since numerical stability (in Runge-Kutta methods and other schemes using explicit discretization in time) requires that the time step Δt scales with the grid spacing (Δx), computational cost for a given physical time period increases at least as fast as N^4. This

means every halving of Δx leads to a 16X increase in cost — which exceeds considerably the performance increase available in most newly installed top-ranked machines over their predecessors. It is, in fact, not surprising that most simulations considered leading edge in scale in their time (e.g., Kaneda *et al.* (2003); Yeung *et al.* (2015)) have been relatively short. It should be noted that increasing problem size is in fact being enabled by increases in memory available on leading-edge machines, but actual computational power is increasing more slowly, such that the largest possible simulations performed within finite resource constraints are at risk of being limited to short physical time periods. This leads to the ironic situation that, as computing power grows and algorithms successfully scale to larger problem sizes, the ability to conduct the next-largest long simulations actually become increasingly compromised.

The second topic in this thesis introduces a new paradigm to address the challenge posed above, for studies of small-scale processes that evolve on short time scales. Two observations are first made, which are supported by previous work (Yeung *et al.*, 2018). The first is that statistical stationarity in time, with a mild assumption of ergodicity, allows us to take samples from multiple short *simulation segments*, provided they are well separated in time, with a high degree of statistical independence. The second is that when a modestly-resolved velocity field is refined to a higher resolution, the small scales adjust quickly, potentially within a couple of Kolmogorov time scales (τ_η). These observations suggest that an alternative to a long, high-resolution (ideal but unfeasible) simulation may be to start with multiple (say M) independent lower-resolution snapshots, allow them to quickly adjust to higher resolution, and then start collecting statistics at the highest resolution after only a short period of time (say $\beta\tau_\eta$, with β not much larger than 1). The length of time spent computing on an N^3 grid would then be proportional to $\beta M\tau_\eta$ (in total), as opposed to multiple T_E's. Substantial savings are both most likely and most necessary at high Reynolds numbers, where $T_E \gg \tau_\eta$, and it is important (Yakhot & Sreenivasan, 2005; Schumacher *et al.*, 2005) to resolve down to scales smaller than the Kolmogorov scale

($\eta = (\nu^3/\langle\epsilon\rangle)^{1/4}$) based on the mean energy dissipation rate ($\langle\epsilon\rangle$). This new paradigm is referred to as Multiple Resolution Independent Simulations (MRIS for short).

Some hints to the viability of this new approach could be found in recent work (Yeung *et al.*, 2018) where events of extreme dissipation and enstrophy were seen to adjust rapidly to changes in resolution, and important conclusions could be drawn from short simulations of length less than $10\ \tau_\eta$. In Yeung *et al.* (2018), short simulations of forced isotropic turbulence at two Taylor-scale Reynolds numbers (R_λ, up to 650) were performed at three resolution levels, up to 8192^3 grid points but all starting from the same initial snapshot. This was, in effect, similar to just one MRIS realization of $10\ \tau_\eta$ long. The objective is to increase the Reynolds number by running larger simulations (Ishihara *et al.*, 2009), while also performing ensemble averaging over the initial conditions by starting from modestly-resolved snapshots originally distributed over several T_E's in time. In addition, both one-point and two-point statistics — in particular, the properties of local averages of the dissipation rate, which play a critical role in understanding intermittency (Kolmogorov, 1962; Frisch, 1995; Sreenivasan & Antonia, 1997) are studied. Availability of data on such averages over 3D volumes (instead of 1D versions) is relatively recent (Iyer *et al.*, 2015). In this thesis, new simulation results at up to 18432^3 resolution (over 6 trillion grid points) are presented, which should help us better understand some of the long-unresolved aspects of intermittency in turbulence (Frisch, 1995; Sreenivasan & Antonia, 1997). The MRIS approach and results from it were published as an invited paper in a leading journal in 2020 (Yeung & Ravikumar, 2020).

The classical cascade picture of turbulence (Frisch, 1995; Richardson, 1922) describes the mechanism with which energy transfer takes place in turbulence. The large scale fluid motions break down, due to instabilities, to smaller scales receiving a fraction of the energy in the large scales. These smaller scales break down further to even smaller scales, and the process continues until the scales of motion are small enough (of the order of Kolmogorov scale) for viscosity to dissipate the energy in the form of heat. The intermediate scales,

also referred to as inertial scales, act mainly to transfer energy from the large scales to the small scales. This self-similar "multiplicative" process of the transfer of energy in turbulent flows lends well to the theory of fractals (Mandelbrot, 1983) and its applicability to studies of turbulence (Meneveau & Sreenivasan, 1991).

Fractals, in the most basic sense, are objects that display self-similarity properties over a wider range of scales. For example, branches of trees, where the large branches originating from the trunk have smaller branches that continue to grow and have branches of their own. Each branch of the tree resembles a small scale version of the whole shape and are referred to as "fractal objects". These objects are associated with a characteristic dimension called the "fractal dimension", which unlike Euclidean dimensions need not be an integer. However, for complex dynamical systems, such as turbulence, a single dimension cannot fully describe the dynamics of the processes involved; instead a continuous spectrum of dimensions, called the multifractal spectrum, is needed. In turbulence, the most commonly studied "fractal object" is the turbulent energy dissipation rate (Sreenivasan & Meneveau, 1986; Bershadskii & Tsinober, 1992). A detailed account of the analysis of turbulence from a multifractal viewpoint can be found in Sreenivasan (1991a).

In this thesis, the turbulent energy dissipation rate will be studied under the multifractal framework using high Reynolds number and resolution data sets generated by applying the "MRIS" approach described previously (Yeung & Ravikumar, 2020). Similar to the study of intermittency, multifractal analysis makes use of 3D local averages of energy dissipation which is difficult to compute, as discussed previously, and not widely reported. This work aims to address this gap by computing and presenting the multifractal spectrum for flows at Reynolds numbers 390, 650, 1000 and 1300 generated using numerical simulations at a resolution of $k_{max}\eta$ close to 4.5. The scaling of the moments of these 3D local averages over a volume of linear size r are studied. Extrapolation of PDF tails are performed to ensure the higher order moments at different scale sizes converge statistically. The scaling exponents, for different orders of moments, are computed and used to estimate the multi-

fractal spectrum, $f(\alpha)$, which corresponds to the fractal dimension and α characterizes the strength of the singularity. Smaller values of α characterize regions of intense dissipation, while larger values of α characterize low dissipation regions.

The study of multifractal spectrum is appealing primarily because of its relative independence to Reynolds numbers while the PDFs of energy dissipation rate which can also be used to study intermittency is dependent on Reynolds number and does not show signs of an asymptotic behavior at high Reynolds numbers. It is important to note that in this study the local averages of energy dissipation rate is computed using all the nine velocity gradients and by performing 3D averages as compared to the use of 1D surrogates and 1D averages along with Taylor's frozen-flow hypothesis (Taylor, 1938) in past laboratory studies (Meneveau & Sreenivasan, 1991; Sreenivasan & Meneveau, 1986). The spectrum obtained from this work is compared with those from Meneveau & Sreenivasan (1991) to find that the use of 1D averages and approximations like 1D surrogacy result in a spectrum similar to those from 3D averages and the full definition of energy dissipation rate. This shows the robust nature of the mutifractal spectrum. Higher Reynolds number flows are ideal candidates for multifractal analysis due to the wide range of scales and highly intermittent nature of the small scales.

A key attribute of turbulent flows is their ability to provide efficient mixing. When the substance or property being mixed is of very low concentration, it does not affect the motion of the fluid. Such substances or properties are called passive scalars, examples of which include dye in water, small temperature fluctuations in air and concentration of reactants in a chemical reaction (Warhaft, 2000; Sreenivasan, 1991*b*). Along with the Reynolds number, an important non-dimensional parameter in passive scalar mixing is the Schmidt number, which is the ratio of kinematic viscosity (ν) of the fluid to the molecular diffusivity (D) of the scalar. The Schmidt number can vary widely, from $\mathcal{O}(0.01)$ for mixing in liquid metals, to $\mathcal{O}(1)$ typical of gas-phase mixing, and beyond to $\mathcal{O}(1000)$ for dyes mixing in liquids (Gotoh & Yeung, 2013). Passive scalar fields with Schmidt numbers

of $\mathcal{O}(1)$ in high Reynolds number turbulence are often observed to be highly intermittent in both the inertial-convective (via anomalous values of structure function exponents) and viscous-diffusive ranges (via intense fluctuations of the scalar dissipation rate, χ). The intermittency of scalar dissipation rate is of particular interest, which is important in combustion modeling (Bilger, 2004).

High-resolution direct numerical simulations are crucial, and reliable conclusions on the Sc-effects require that resolution be adequate for all scalars involved. Such calculations are computationally very expensive. However, the new MRIS approach (Yeung & Ravikumar, 2020) helps reduce the computational requirements while generating statistically reliable data at high resolution. It is also useful to perform ensemble averaging over the statistics of scalars with the same Sc but subjected to uniform mean gradients in different directions. Fluctuations of χ for a scalar with $Sc = 1$ are typically seen as more intermittent than those of the energy dissipation rate (ϵ). One long-standing question is whether the scalar field satisfies local isotropy, as the Reynolds number increases (Sreenivasan, 1991*b*; Warhaft, 2000). An important diagnostic is in the skewness of the scalar gradient (in the direction of the mean gradient), which does not seem to decrease with increasing Reynolds number, but may be affected by resolution for the scalar field (Donzis & Yeung, 2010). A second, more open-ended question is in the nature of fluctuations of the scalar dissipation rate, χ, which is a quadratic measure of the scalar gradients. The same questions posed for the energy dissipation rate, including the form of the probability density function (PDF), the likelihood and intensity of extreme events and the properties of local averages, are also relevant for χ (as a function of both Reynolds and Schmidt numbers). Numerical results including moments of local averages of χ and conditional moments of χ given ϵ at resolution up to 6144^3 will be presented.

The final topic of this thesis will focus on anisotropic flows in the presence of density stratification due to two active scalars (Schmitt, 1994) of different Schmidt number. Key non-dimensional parameters include the Reynolds number, Schmidt number, and Froude

number(s) (F_r, which is the ratio of buoyancy to turbulence time scales). The scalars affect the flow, through buoyancy effects that can be stabilizing or de-stabilizing (Riley & deBruynKops, 2003). The most interesting scenarios are those in which the two scalars have opposing effects, with the difference between their molecular diffusivities playing a pivotal role (Gargett *et al.*, 2003). Strong unstable stratification can also lead to new spatial and temporal resolution constraints that can be difficult to satisfy.

To improve numerical fidelity in the two-way coupling terms between the velocity and scalar field, a new approach that diagonalizes the system of conservation equations transformed to wavenumber space has been developed. These developments will facilitate work towards a detailed understanding of the effects of an active scalar on the flow energetics and anisotropy (at various scale sizes), depending on the diffusivity of the scalar. Key aspects include how the classical property of an energy cascade would be modified, and the nature of the Reynolds-stress budget in active-scalar turbulence. There is a special interest in the subject of double-diffusive convection, where one scalar is stabilizing while another is de-stabilizing, including both oscillatory and so-called fingering regimes where peculiar structural patterns emerge (Stellmach *et al.*, 2011). The goal of this topic is to perform short exploratory simulations to better understand the nature of such flows and lay a path forward for more detailed analysis.

1.1 Objectives

In summary, the major objectives of this thesis are as follows:

1. **To develop an algorithm capable of extremely large scale simulations using GPUs on pre-exascale and exascale architectures.**

   The goal is to develop a new algorithm for pseudo-spectral direct numerical simulations of turbulence capable of achieving extreme problem sizes, exceeding the prior state-of-the-art worldwide, on a heterogeneous GPU platform such as Summit at the Oak Ridge Leadership Computing Facility (OLCF) at Oak Ridge National

Laboratory. Major considerations involve optimizing network communication and GPU-CPU data movements, as well as achieving large problem sizes despite memory limitations on the GPU. Since Summit is one of the world's fastest supercomputer, it is reasonable to design a new code that is dependent on specific characteristics of this machine, which is built of IBM CPUs and NVIDIA GPUs. From a long-term perspective, it is also important to be able to port the code to more general heterogeneous platforms beyond Summit.

2. **To understand intermittency and the multifractal nature of energy dissipation rate using high resolution DNS.**

   A major motivation for ever-larger simulations is to advance understanding of intermittency and extreme events in turbulence, where both high Reynolds number and good small-scale resolution are clearly important. However, because resources available are finite, the largest simulations are often short. In the case of statistically stationary turbulent flows, this raises concerns on limitations in sampling, which is an issue made more acute by success achieved under Objective 1. A protocol consisting of multiple short simulation segments with successive grid refinement is proposed to address this. Targets of investigation include extreme fluctuations of the energy dissipation rate and enstrophy, statistics of their local averages over three-dimensional sub-domains of various sizes and the multifractal spectrum of energy dissipation rate. Using the new algorithm and approach, simulations to study intermittency of scalar dissipation rate will be performed. Statistics of local averages of scalar dissipation rate are computed and analyzed.

3. **To study turbulence in stratified flows due to more than one active scalar.**

   In addition to studies of passive scalars which do not affect the flow, simulations of active scalars such as temperature and salinity in the ocean, or concentration differences in liquid mixtures will be performed. The objective here is to understand

the complexities that arise when two scalars of different molecular diffusivities are present and contributing to opposing effects of stable versus unstable stratification. Targets of investigation include anisotropic development and differential diffusion acting at different scales depending on the Schmidt numbers involved.

CHAPTER 2

GPU ACCELERATION OF EXTREME SCALE PSEUDO-SPECTRAL SIMULATIONS OF TURBULENCE USING ASYNCHRONISM

Turbulent fluid flows governed by the Navier-Stokes equations with disorderly fluctuations over a wide range of scales in time and space (Pope, 2000) represent a major challenge in both science and computing (Mininni *et al.*, 2011; Lee *et al.*, 2013; Yeung *et al.*, 2015; Ishihara *et al.*, 2016; Watanabe *et al.*, 2016). In work focused on fundamental understanding, it is often useful to employ periodic boundary conditions on a cubic domain, with solution variables expressed in a discrete Fourier series as in pseudo-spectral simulations (Rogallo, 1981; Canuto *et al.*, 1988).

The conduct of large scale simulations of turbulence at high Reynolds number and high resolution has been a major undertaking in the computational turbulence community (Ishihara *et al.*, 2016; Yokokawa *et al.*, 2002). In most cases, scalability and/or time to solution have been limited by communication costs, which become more dominant at larger problem sizes. As the community looks towards exascale (projected to arrive in 2022), a key question is how codes that are communication intensive can benefit from heterogeneous platforms whose principal advantage is fast computation on hardware accelerators such as GPUs. In this chapter, the nontrivial task of implementing a new pseudo-spectral turbulence code (Ravikumar *et al.*, 2019) capable of reaching unprecedented problem sizes at the high throughput needed to complete production simulations on Summit is addressed.

The algorithm is further extended to support tracking of Lagrangian fluid particles, where computing the spline coefficients for cubic spline interpolation is a key aspect. With the workflow involved being similar to 3D FFTs, the batched asynchronous approach can be adapted to form the spline coefficients as well. Furthermore, communication costs are minimized by adopting a "local particle decomposition" (Buaria & Yeung, 2017) along

with "ghost layers", that hold the spline coefficients from a couple of planes in the neighboring tasks, formed using one-sided MPI communication to ensure the interpolation can be done entirely locally.

It is essential that this successful programming strategy be implemented in a portable manner. This is especially important since the CUDA programming model that we use on Summit will not be universally supported on all the future exascale machines. The batched asynchronous algorithm will be ported to use OpenMP, especially version 5.0 and beyond, to target GPUs. Efficient strided copies and interoperability between GPU libraries and OpenMP tasks are some of the important challenges that are addressed.

The following sections begin with some background on equations, numerical methods and current programming approaches, while noting key differences between Summit and several existing petascale platforms. The new algorithm is described in some detail in section 2.2 followed by MPI and data copy optimizations in section 2.3 before reporting and analyzing the performance results in section 2.4. The code development efforts and performance numbers for the particle tracking algorithm are presented in section 2.5. Finally, section 2.6 presents a brief discussion of some key ideas and challenges encountered in the OpenMP porting efforts.

2.1 Equations and Numerical Methods

The application code in this work is written to compute fluctuating velocity fields $\mathbf{u}(\mathbf{x}, t)$ in time and three-dimensional space, according to the Navier-Stokes equations expressing the principles of mass and momentum conservation, in the form

$$\partial \mathbf{u}/\partial t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla(p/\rho) + \nu \nabla^2 \mathbf{u} + \mathbf{f} \ , \tag{2.1}$$

where $\mathbf{u}$ is a solenoidal vector, p is the pressure, ρ and ν are the fluid density and viscosity (taken as constants), and $\mathbf{f}$ is a forcing term. This is a partial differential equation of the

advective-diffusive type, which occurs in many studies of transport phenomena in science and engineering.

Equation 2.1 is solved in a simplified solution domain which is periodic in all three directions. The velocity field is expressed in a finite-terms Fourier series, as $\mathbf{u}(\mathbf{x},t) = \sum_{\mathbf{k}} \hat{\mathbf{u}}(\mathbf{k},t)\exp(\iota\mathbf{k}\cdot\mathbf{x})$ where $\iota = \sqrt{-1}$, overhats denote (complex-valued) Fourier coefficients and $\mathbf{k}$ is a wavenumber vector whose coordinate components on an N^3 grid take the values $1-N/2, 2-N/2, ...0...1....N/2$. In Fourier (i.e., wavenumber) space Equation 2.1 becomes, for a given $\mathbf{k}$, an ordinary differential equation which can be written as

$$\partial\hat{\mathbf{u}}/\partial t = -\widehat{\nabla\cdot(\mathbf{uu})}_{\perp\mathbf{k}} - \nu k^2\hat{\mathbf{u}} + \hat{\mathbf{f}}\ . \tag{2.2}$$

To enforce mass conservation under the assumption of constant density, the nonlinear term is projected into a plane perpendicular to the vector $\mathbf{k}$. Aliasing errors arising from the treatment of nonlinear terms are generally controlled by a combination of phase-shifting and truncation in wavenumber space (Rogallo, 1981).

In production simulations, Equation 2.2 is typically integrated over many thousands of time steps, using explicit second- or fourth-order Runge Kutta (RK2, RK4) schemes for the nonlinear terms, while viscous terms are treated exactly via an integrating factor. In general, RK4 offers improved accuracy and numerical stability. However, experience also shows (e.g. Rogallo (1981) and numerous publications in the turbulence community adopting the algorithm described therein) RK2 results are often adequate when the time step is made sufficiently small. Timings corresponding to RK2 alone are presented in this chapter. The cost of RK4 per time step is approximately doubled, with a small increase in the memory necessary to hold partially updated values of the solution variables at different substages within a single time step.

In this code, since time advance is performed in Fourier space (per Equation 2.2) each RK substep starts and ends in Fourier space, as well. The essential mathematical operations

involved include transforming three velocity components to physical space, forming the nonlinear products there, and transforming those products back to Fourier space. The actual transforms are taken one direction at a time. The need for data movement arises from the need to collect complete lines of data (successively, along each coordinate axis) in the core memory to be operated on.

While the number of variables being transformed varies during each time step, the structure and performance of the turbulence simulation code share many similarities with 3D FFTs, which are relevant to many science disciplines and have been the subject of much software development (e.g. Pekurovsky (2012); Gholami *et al.* (2016); Dalcin *et al.* (2018)). Therefore, some of the novel programming techniques developed in the present work are potentially relevant to the wider computational science community, as well.

2.2 Algorithm description

This section presents a detailed description of the code development, starting with the choice of domain decomposition, and leading to the asynchronous algorithm for computing on GPUs while using the CPU memory capacity. The communication costs of all-to-all communication, inherent in 3D FFT algorithms, dominates overall runtime, especially as the FFT kernels themselves become very fast on the FLOP heavy GPUs. Because of this known communication bottleneck, which will always limit speedup no matter how many FLOPs the GPUs become capable of, algorithm design is driven by choices that allow as much communication overlap with computations and data copies, as well as a small number of large messages. It is important to note that the highest priority is not high scalability *per se*, but instead making feasible simulations of a size exceeding the current state-of-the-art. This includes a goal of 20 seconds or less per RK2 timestep in order to enable long-running simulations in a reasonable number of wallclock hours. However, the problem size is more important for science than the wall time. Path-breaking simulations on the Earth Simulator (Yokokawa *et al.*, 2002), that reached 4096^3 resolution, are used as a reference point even

today not for the time-per-step it took to simulate the problem but for the fact that it was a significant step forward in the state-of-the-art problem size at that time.

2.2.1 Domain decomposition

The first decision in devising a parallel algorithm for very large problem sizes is how best to distribute memory requirements among, say, P MPI processes. As seen in Figure 2.1 a N^3 solution domain can be decomposed into *slabs* or *pencils*, in one and two directions, respectively. For a slab, or 1D decomposition, each MPI process works on an integer number of planes (e.g., $x-z$) and can take FFTs in two directions forming that plane. A global collective communication call (of the "all-to-all" type) is used to re-divide the data along the third direction (e.g. y). While the concept of a 1D decomposition is straightforward, clearly, it is limited to $P \leq N$.

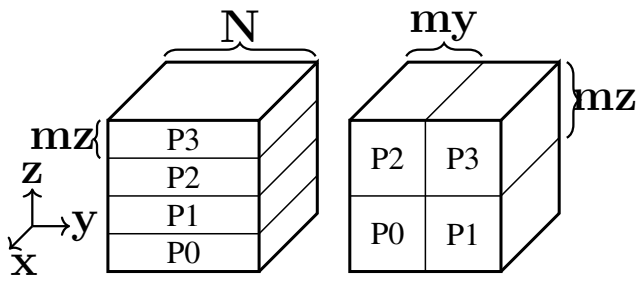

Figure 2.1: 1D and 2D domain decomposition's, illustrated for case of MPI process count $P = 4$. On the left, each slab is of size $N \times N \times mz$, where $mz = N/P$. On the right, each pencil is of size $N \times my \times mz$, where $my = N/P_r$ and $mz = N/P_c$.

A 2D domain decomposition divides the data in two directions, allowing a finer-grained decomposition (i.e. larger P). A 2D Cartesian process grid is used, with $P = P_r \times P_c$ where P_r and P_c are the sizes of the "row" and "column" communicators, respectively. Two collective communication calls are performed using the (smaller) row and column communicators instead of once globally over P processes. The best performance is usually obtained if P_r equals the number of MPI ranks per node, since some of the communication will then occur solely on the node.

State-of-the-art turbulence simulations performed on massively parallel platforms (Lee *et al.*, 2013; Yeung *et al.*, 2015; Ishihara *et al.*, 2016) have generally used a 2D domain decomposition, as do the FFT portions of several of Exascale Computing Project applications (lammps, hacc) (Plimpton, 2017). However, the latest trends in HPC landscape appear to point towards machines with fewer nodes which are more powerful in both memory and speed, with Summit being a primary example. Accordingly, the 1D (slabs) decomposition is adopted for this work, with a hybrid approach used to further parallelize within a slab.

2.2.2 Target System and Software Stack

The target architecture around which the code was designed is the IBM Power System AC922 (Vetter *et al.*, 2018) which is used in the Summit and Sierra supercomputers. A node on Summit consists of a dual socket POWER9 processor, with each socket connected via NVLink to 3 NVIDIA V-100 GPUs (2 links/GPU) and 22 cores. Each core is capable of supporting up to 4 hardware threads. Summit nodes have 512 GB of random access memory for use by the application, and each GPU has 16 GB of High Bandwidth Memory (HBM). Each V-100 GPU has 80 Streaming Multiprocessors (SMs), which are the processors on the GPU and are capable of fast computations. The overall picture is then of a very dense (or fat) node with a large amount of memory and compute resources per node. This allows for simulating massively large problem sizes on a single node.

The interaction of the CPU memory bandwidth, NVLINK bandwidth, and network card bandwidth will be important in understanding some of the limitations of data movement overlap. The Power 9 CPU memory bandwidth per socket is 135 GB/s peak unidirectional, while the CPU-GPU NVLINK connection is capable of 150 GB/s (peak, per socket) and the network card on Summit is capable of 12.5 GB/s (per socket, bi-directional) (Vetter *et al.*, 2018; IBM POWER9 NPU, 2018). This means that NVLINK data transfers alone are capable of fully saturating the Power 9 memory bandwidth, and any simultaneous use of the network card must compete with NVLINK bandwidth demands.

The code is written in Fortran and the GPUs are programmed using the CUDA Fortran programming model (Ruetsch & Fatica, 2014), implemented in the IBM XL compiler, and calls to NVIDIA's **cuFFT** library. CUDA Fortran is favored for its ease of use and similarity to Fortran-like syntax, while supporting advanced data movement patterns and asynchronous features on the GPUs. CUDA streams and CUDA events are used to control the asynchronous tasks of batching data on/off the GPU, computing FFTs, and determining when the data is available in host memory to be sent through asynchronous **MPI_IALLTOALL** calls.

2.2.3 A basic (synchronous) GPU algorithm

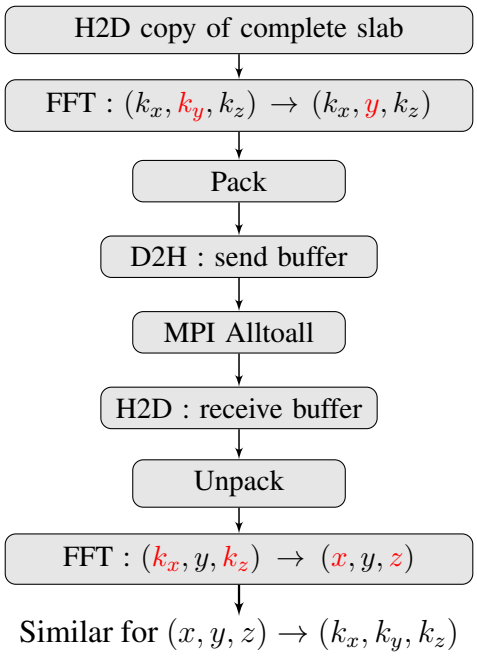

Figure 2.2: Schematic showing the different operations involved in computing 3D FFTs using GPUs in a synchronous manner. Similar operations in the reverse order are required to transform back to Fourier space.

Prior to a detailed description of the best-performing asynchronous algorithm, it is useful to review the basic requirements for a GPU implementation, in terms of the work needed to perform a complete 3D FFT. Figure 2.2 shows the basic sequence of operations, focused on how a 3D FFT is taken of solution variables, initially in Fourier space. This sequence is

also a close match with the work performed in the first half of each Runge-Kutta substage in our turbulence code, before the nonlinear terms are formed in physical space and transformed back to Fourier space. The sequence of operations shown in Figure 2.2 also reverts to that of a CPU code if the host-to-device (H2D) and device-to-host (D2H) data copies are eliminated and all operations are carried out on the CPUs.

At the beginning of the execution sequence in Figure 2.2, each MPI process holds a slab of data that consists of $x-y$ planes stacked up in N/P units in the z direction. This slab of data is copied from host to device. Next, the 1D FFTs in the y direction are computed on the GPU using **cuFFT**. An all-to-all communication is then required to transpose these partially-transformed quantities into slabs of x–z planes. Since the data to be exchanged is not contiguous in the local memory, it has to either be packed into contiguous messages locally, or else MPI derived datatypes need to be used. The performance of packing on the CPU was compared against packing on the GPU and then copying to the CPU with the latter resulting in the fastest performance. Once the packed data is copied to the host, an **MPI_ALLTOALL** is performed from the host to transpose the data. Subsequently, the transposed data is copied from host to device, unpacked into the correct memory locations, and transformed in z and then x. Recent advancements in software and hardware, such as CUDA-aware MPI and GPU-Direct (Shainer *et al.*, 2011), allow MPI to directly transfer data that is resident on the GPU. In theory, such a strategy should help avoid the additional data transfers before and after the global transpose. However, since without GPU-direct close to peak system memory bandwidth is already achieved for data transfers and the major bottleneck remains the bandwidth of the network interface card, the actual benefits of GPU-direct on Summit may be significant. In fact, after implementing CUDA-aware MPI and GPU-direct, no noticeable benefit to the runtime was observed.

Since the Fourier coefficients (being of real-valued variables) satisfy the property of conjugate symmetry, i.e., $\hat{\mathbf{u}}(-\mathbf{k}) = \hat{\mathbf{u}}^*(\mathbf{k})$, the transforms are complex-to-complex in two directions (y and z) but complex-to-real in the third (x). This ordering of the transform di-

rections (y,z,x; reversed when transforming from physical to wavenumber space) is chosen so that formation of nonlinear terms in physical space can be performed more efficiently on arrays of stride unity. In this data structure, FFTs in x are more efficient because of a unit vector stride. For FFTs in y and z the options of performing these transforms in a strided or unstrided data structure is available, but on Summit either option takes about the same time when the cost of additional local data reordering is also factored in.

The algorithm illustrated here is readily extended to the pseudo-spectral turbulence code, in which a reverse sequence of operations takes place after the nonlinear terms are formed. However, there are two hurdles to direct application of the above algorithm. The first is that the algorithm is synchronous, such that each set of operations is carried out sequentially without taking advantage of the fact that operations on different portions of data can be made to occur asynchronously (e.g. Clay *et al.* (2018)). The second is that, as written, each GPU is to process a single slab of data all at once, which creates difficulties in processing larger problem sizes where a single slab of data will not fit into the GPU memory. To compute at larger problem sizes (as is the goal here) it is beneficial to break a slab into smaller portions that will fit into the GPU memory. This data division inside a slab opens up the possibility of a significant degree of task asynchronism, where, for instance, different planes (or even parts of planes) within a slab may be copied, computed, and communicated simultaneously. Indeed, this is the motivation in the development of an asynchronous algorithm capable of larger problem sizes, as described in next.

2.2.4 Batched asynchronous algorithm

The objective with the new algorithm is to be able to run large problem sizes efficiently without being limited by the GPU memory capacity. The GPU memory issue can be addressed by dividing a slab into several (np) pencils and processing each pencil separately, as illustrated in Figure 2.3. This also provides an opportunity for overlapping operations on different pencils within the same slab.

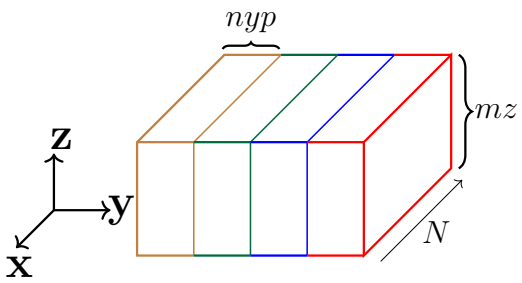

Figure 2.3: Decomposition of a slab of data into multiple (np) pencils, each of size $N \times nyp \times mz$, where $nyp = N/np$, to enable larger problem sizes where a single slab does not fit into the GPU memory.

The desired asynchronism is enabled using the programming model of CUDA execution streams and events. Two separate *CUDA streams* are specified, one for computations and one for data movement. Besides allowing overlap with data movement and compute, a distinct data transfer stream ensures that bandwidth is devoted to one direction of traffic at a time. This is beneficial on Summit, because while NVLink supports both maximal read and maximal write bandwidth simultaneously, the host memory bandwidth supports a combined read *or* write bandwidth and the maximum aggregate bandwidth is achieved when performing unidirectional movement (IBM POWER9 NPU, 2018). The choice of a single transfer stream allows the full bandwidth to be devoted to whichever transfer operation is first put into the stream, ensuring that the correct piece of data is copied into the GPU as quickly as possible. *CUDA Events* are used to enforce synchronization between operations in different streams (Ruetsch & Fatica, 2014).

The sequence of operations in the new asynchronous algorithm is shown in Figure 2.4. For brevity, only the operations needed to transform from Fourier space to physical space (with those from physical to Fourier space being very similar but reversed in order) are shown. Color coding is used to help identify operations in the transfer stream (H2D and D2H copies), compute stream (such as FFTs in separate directions), and communication. Colons within a box refer to operations carried out on a specific pencil whose ID, denoted by ip, ranges from 1 to np. The three large regions bounded by dashed lines correspond to computations carried out in the y, z and x directions, respectively. Data transfer operations

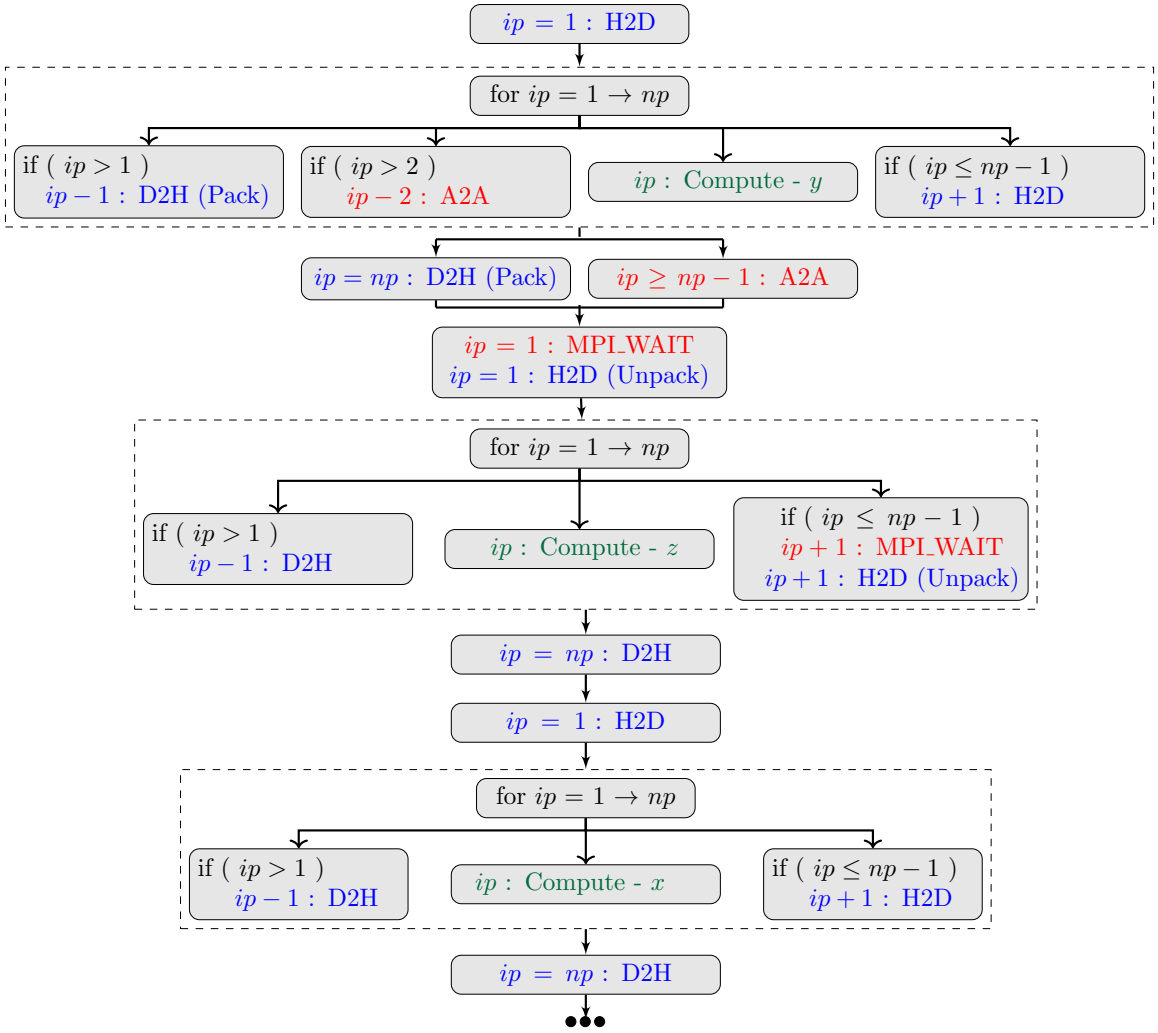

Figure 2.4: Schematic showing the asynchronous GPU algorithm where operations on the same row are performed asynchronously. The operations in blue are performed in the transfer stream, while the operations in green are in the compute stream and the operations in red are using the network. The compute operations correspond to Fourier transforms and other computations, such as forming nonlinear products in the DNS code.

on the first and last two pencils are handled separately, as shown by the blocks outside the large dashed regions.

Before entering the first dashed region, the first pencil is copied to the GPU. Then, if $ip = 1$ the code performs computations. When $ip > 1$ the $(ip - 1)^{th}$ pencil is copied back from the GPU and packed into a contiguous array on the CPU, provided the computations on it have been completed. The pack operation is performed as a strided data copy to avoid packing the data before the global transpose. In this way, both the packing and the D2H

are performed in a single operation. Although operations on the same row are executed asynchronously, our operations are launched from left to right, which is to prioritize data copy out of the GPU so that the global transpose can be initiated as soon as possible. A non-blocking MPI all-to-all is launched on the $(ip-2)^{th}$ pencil only when the D2H copy of the $(ip-2)^{th}$ pencil is completed. Computations are performed as soon as the H2D copy on the ip^{th} pencil is completed. A H2D copy for the next pencil is also posted at this time. If ip equals one of its last two values, the code takes special action to copy out the last pencils back to the CPU, and to post the global transposes.

It should be noted that copy operations in the transfer stream are performed asynchronously, i.e., the CPU can move forward to other tasks, but it does not imply the copy has completed or even started. An *event* is recorded to track the progress of the copy. This ensures the D2H copy will begin only once the computations on the $(ip-1)^{th}$ pencil are completed. Similarly, the H2D copy waits for the data in the GPU buffer into which the host data needs to be placed is copied out.

Operations in the second and third dashed regions, performed on data in $x-z$ slabs, are also scheduled in a manner that allows overlapping between the transfer and compute streams. The only MPI function call from here until completion of the entire 3D FFT sequence is an **MPI_WAIT** in the second dashed region. This is to ensure the transpose completes before the H2D copy is posted.

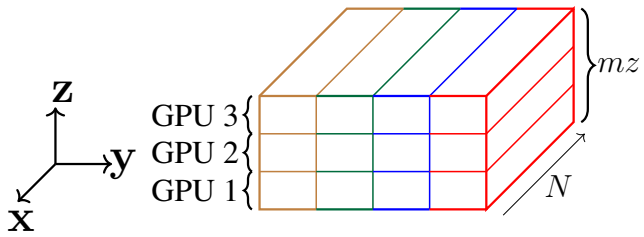

Figure 2.5: When running with multiple GPUs per MPI rank, each pencil is further divided up vertically to allow running with multiple GPUs per MPI task. The pencil fractions are processed by the different GPUs available to the MPI rank.

As to be noted in section 2.3 and section 2.4, there is some advantage in using OpenMP

threads instead of pure MPI on the CPU. On Summit, when the number of MPI ranks per node drops below 6, multiple GPUs per MPI rank will become available. Each pencil is further divided vertically such that a fraction of the pencil is run on each GPU, as shown in Figure 2.5. One OpenMP thread per GPU is used to launch the same operations as described in Figure 2.4 to the different GPUs available to each MPI rank. The device each thread works with is set using the **cudaSetDevice** API call. The global MPI transpose is posted only after the entire pencil has been processed by each of the GPUs available to the MPI rank. The code is also capable of waiting for all the pencils in a slab to be processed so that one large all-to-all can be posted instead of multiple smaller ones. The logic of the algorithm in Figure 2.4 is still applicable.

2.2.5 Problem sizes and node counts

It is useful to develop estimates of the node count necessary to meet the memory requirements of a chosen problem size. The focus is on solving the largest problem possible on Summit, subject to some constraints. The first constraint is to keep wall time per RK2 time step at all problem sizes to below roughly 20 seconds because this allows a reasonable simulation turnaround time in human hours. The second constraint is that N be powers of 2 or at least an integer rich in factors of 2 because this usually leads to the best discrete FFT library performance. Furthermore, to make full use of all 6 GPUs per node (3 per socket) on Summit, a value of N that contains a factor of 6 is advantageous. $N = 18432$ is chosen as the target because it is rich in factors of 2, divisible by 3, and (as will be shown below) it fits in Summit's memory.

For an N^3 problem involving D storage buffers at single precision on M nodes, the memory required per node is $4DN^3/M$ bytes. A detailed counting of the number of velocity components, nonlinear terms, and send/receive buffers which are used to transfer data between CPUs and GPUs, yields a total of 25 buffers ($D \approx 25$). These buffers are page-locked, by allocating as *pinned* memory, which allows the GPUs direct access to the

physical address of this buffer without any CPU intervention, allowing for much faster data copies that otherwise. It was estimated from tests that the operating system occupies approximately 64 GB on each Summit node, leaving 448 GB for user codes. Equating $4DN^3/M$ (where $D = 25$ and $N = 18432$) to 448 GB gives $M = 1302$, which is the minimum number of nodes needed. However, for load balancing on a per-node basis, the number of nodes should be a factor of N. With $N = 18432$ and noting the total system on Summit has approximately 4608 nodes, the only 2 possible values of M are thus 1536 and 3072. In the interest of a shorter time-to-solution 3072 nodes will be used which is 67% of the full system.

For a given CPU node count, it is important to also consider how the memory might fit into the GPUs, generally by processing pencil-sized portions of each slab at a time. For compute purposes, 9 pencil-sized buffers are required. This number needs to be tripled, to 27, to allow asynchronous execution in the manner described in subsection 2.2.4, while pack and unpack operations can be performed without additional buffers. If there are np pencils per slab, then each pencil contains $N^3/(M \times np)$ words (per variable). As for the GPU memory, it is reasonable to assume 96 GB of GPU memory on each node is user-accessible, with no operating systems-related tasks running on the GPU. Thus, equating $4 \times 27 \times N^3/(M \times np)$ bytes (with $N = 18432$, $M = 3072$) to 96 GB gives, nominally, $np = 2.13$. In practice, because further needs for memory arise from other smaller arrays, for $N = 18432$ it was found that np needs to exceed 3.

Table 2.1: Node counts, problem sizes, minimum number of pencils per slab and the size of each pencil (for 1 variable) that can fit into the GPU memory. Problem sizes from 3072^3 to 12288^3 are exact weak scaling cases, while 18432^3 is larger than what weak scaling suggests.

# Nodes	Problem size	Mem. occ. per node (GB)	No. of pencils	Size of pencil (GB)
16	3072^3	202.5	3	2.25
128	6144^3	202.5	3	2.25
1024	12288^3	202.5	3	2.25
3072	18432^3	227.8	4	1.90

Since np must be an integer, each slab in the 18432^3 problem is divided up into at least 4 pencils to fit in the GPU memory. The node count and the number of pencils required for a range of problem sizes are given in Table 2.1.

2.3 Code Optimizations

During code development, special emphasis is placed on optimizing cross-node data communication and also on-node data movement between the CPU (host) and GPU (device). In the two subsections below, several approaches and the performance data obtained from them on Summit are discussed.

2.3.1 MPI Configurations

Given Summit's node architecture, two natural choices for the number of MPI tasks per node are 6 (one per GPU) or 2 (one per socket). In the latter case, OpenMP threads can be used to launch operations to the 3 GPUs per socket, while the message size per MPI rank is increased by 3X. In addition, in both scenarios above, since the slab of data assigned to each MPI rank is further broken down into pencils of data which are batched on and off of the GPU for processing, each MPI rank can be made to perform communication in three possible ways: namely the entire slab all at once, one pencil at a time, or a selected number (say, Q) of pencils per call. The choice $Q = 1$, where an MPI all-to-all is called to transpose a pencil of data as soon as the pencil has been processed and copied back from device to host, is conducive to overlapping MPI with data movement and computation. However, it also leads to messages that are potentially so small that they become dominated by latency. Fewer MPI calls with larger messages can be realized by choosing $Q > 1$, up to the number of pencils present in each slab.

The network interconnect on Summit is a dual-rail EDR InfiniBand network and provides a node injection bandwidth of 23 GB/s (OLCF, 2019) and a bisection bandwidth of 46GB/s. Although these bandwidths are in principle achievable for point-to-point commu-

nications of sufficient size, in all-to-all communication the bandwidth achieved at scale can be considerably lower, because the individual peer to peer (P2P) messages involved can become small. In the batched algorithm, if each slab is divided into np pencils, then the message chunk that must be delivered to each process (P2P message size) for nv variables at single precision is $4 \times nv \times (N/np) \times (N/P)^2$ bytes.

To understand the MPI performance, tests were conducted using a standalone MPI all-to-all kernel which carries out communication operations mimicking those in the DNS code, but neither computes nor moves data between CPU and GPU. One key difference is that whereas the DNS code is able to use non-blocking **MPI_IALLTOALL** to allow for overlapping between MPI and local operations, the standalone kernel instead calls the blocking **MPI_ALLTOALL**, in order to enable direct measurement of MPI performance. A collection of MPI performance data is shown in Table 2.2, where problem sizes and node counts correspond to the information in subsection 2.2.5. The effective bandwidth is calculated by the formula

$$BW = (2 \times P2P \times P \times tpn) \; / \; time \tag{2.3}$$

where tpn is the number of MPI ranks per node, $time$ refers to the wallclock time taken to perform the all-to-all by each process, and a factor of 2 is included since all-to-all's are comprised of both sends and receives. This formula also considers on-node messages, which are exchanged through shared memory and are generally faster, when computing the effective bandwidth, but accounting for this separately becomes insignificant at larger problem sizes.

The three cases in Table 2.2 are referred to as as A, B and C, respectively. Between A and B, the P2P message size increases by 9X because data per process is tripled while, there are also 3 times fewer processes to perform the data exchange. The increase in P2P message size from B to C is a factor of np since there are np pencils per slab. Case B gives

Table 2.2: Effective bandwidth per node of MPI all-to-all on Summit at different node counts. The message size communicated between each MPI process (P2P) is reported (for 3 variables).

Nodes	A: 6 tasks/node		B: 2 tasks/node		C: 2 tasks/node	
	1 pencil/A2A		1 pencil/A2A		1 slab/A2A	
	P2P (MB)	BW (GB/s)	P2P (MB)	BW (GB/s)	P2P (MB)	BW (GB/s)
16	12	36.5	108	43.1	324	43.6
128	1.5	24.0	13.5	39.0	40.5	39.0
1024	0.19	11.1	1.69	23.5	5.06	25.0
3072	0.053	13.2	0.47	12.4	1.90	17.6

a higher bandwidth achieved than case A, up to a node count of 1024. At 3072 nodes it is surprising that case A performs slightly better than case B, because this departs from (1) the trend of larger P2P message size leading to better bandwidth and (2) our experience with the full DNS code where case B has a faster solution time than case A. However, as noted above, the DNS code uses non-blocking MPI all-to-all calls, whereas the standalone MPI code uses the blocking version. One possibility is that at sufficiently small message sizes and without overlap, case A may be able to take advantage of hardware acceleration in the network and eager limits, which allows small messages to be exchanged faster by reducing synchronization among processes.

The trend in increased P2P message size leading to increased bandwidth continues when comparing B to C, especially at scale. This is expected because, in general, a shift from a larger number of smaller messages to a smaller number of larger messages reduces the effect of network latency. All three cases were implemented in the turbulence code, and as was observed in the standalone MPI tests, the configuration of case C led to the best-performing implementation of the turbulence code as well, which is expected because of the communication-intensive nature of these codes.

2.3.2 Strided copy optimization

The batched asynchronous algorithm described in subsection 2.2.4 requires frequent strided copies of relatively small units of data between host and device. This pattern can be expected for any algorithm performing operations on complete lines of data on a distributed 3D domain. Such strided copies occur when data stored on the CPU as either $x-y$ or $x-z$ planes must be moved onto the GPU in smaller pieces, or when being packed in preparation for MPI all-to-all communication.

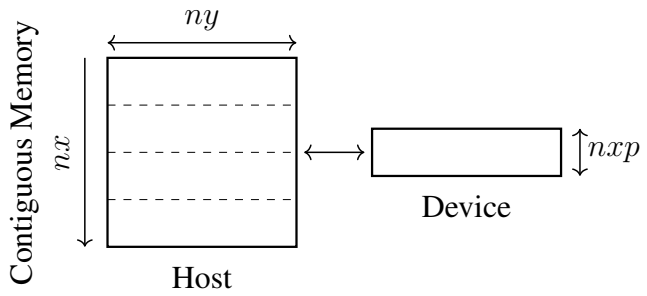

Figure 2.6: Top-down view of a slab of data on the CPU and a pencil of data on the GPU where $nxp = nx/np$. Strided (in y) copies of contiguous data (in x) is required in order to transform in y direction, since memory is linear (stride 1) in the x direction on both the CPU and GPU. Strided FFTs are performed in the y direction to avoid reordering on the GPU.

For example, for a $x - y$ slab divided into 4 pencils as shown in Figure 2.6, copying a pencil of data to the GPU requires copying a series of contiguous chunks of data to the GPU. For the 18432^3 problem the innermost dimension to be copied will have $18432/4 = 4608$ elements or 18 KB of contiguous memory. An entire pencil of such lines of data must be copied, such that, with $mz = 3$ and $nv = 3$ the pencil shape is $4608 \times 18432 \times 3 \times 3$. This gives 165888 chunks of 18 KB each, which must be copied. When the chunk size is small (and the number of chunks therefore large), the many **cudaMemCpyAsync** calls required can be very slow, presumably because the API call overhead begins to become significant. A high performing solution to this is potentially using a CUDA zero-copy kernel to have the GPU instead of the CPU initiate the many small transfers on host page-locked memory (Appelhans, 2018*b*). Therefore, for this work, two alternative approaches were tested,

namely custom written ”zero-copy” CUDA kernels and an asynchronous version of the CUDA library call **cudaMemCpy2D**, as described below.

The custom zero-copy CUDA kernel uses threads to move data between arrays allocated in the GPU memory and arrays which reside in host memory. The zero-copy kernel makes use of the fact that CUDA threads can access host resident memory directly from the GPU without having to explicitly copy the data, i.e., there are zero copies residing on the device. This is enabled by using the CUDA library call **cudaHostGetDevicePointer** to acquire a valid device pointer to pinned host memory. This host memory must be page-locked memory (pinned) which is also needed for maximum transfer speeds for host arrays that are frequently copied in and out of the GPU.

The zero-copy kernel method, however, is limited by the fact that it uses some of the GPU streaming multiprocessors to copy data, which can slow down the other computational kernels. On the other hand, the CUDA library call, **cudaMemCpy2DAsync**, uses the GPU copy engines and accepts arguments that allow for simple strided copies to be performed easily. This comes with the advantage of not having to occupy GPU SMs to achieve the movement.

Timings for strided memory copies obtained from the three different approaches considered here are compared in Figure 2.7. The copies were performed on a fixed total message size of 216 MB but the size of the contiguous memory in the strided copy is varied. It can be seen that both the zero-copy and **cudaMemCpy2DAsync** approaches perform much better than (many) **cudaMemCpyAsync** when the contiguous message sizes are below 100’s of KB. For the 18432^3 DNS problem the contiguous extent of the pencils is 18 KB, which is close to the 8.8 KB tested in the figure. From this data, two conclusions can be drawn. The first is that the many **cudaMemCpyAsync** approach is much slower than the zero-copy or **cudaMemCpy2DAsync** approaches, while the latter two give similar timings. The second is that when moving a fixed amount of data, the overhead involved in moving a finer granularity of chunks can increase the data movement time.

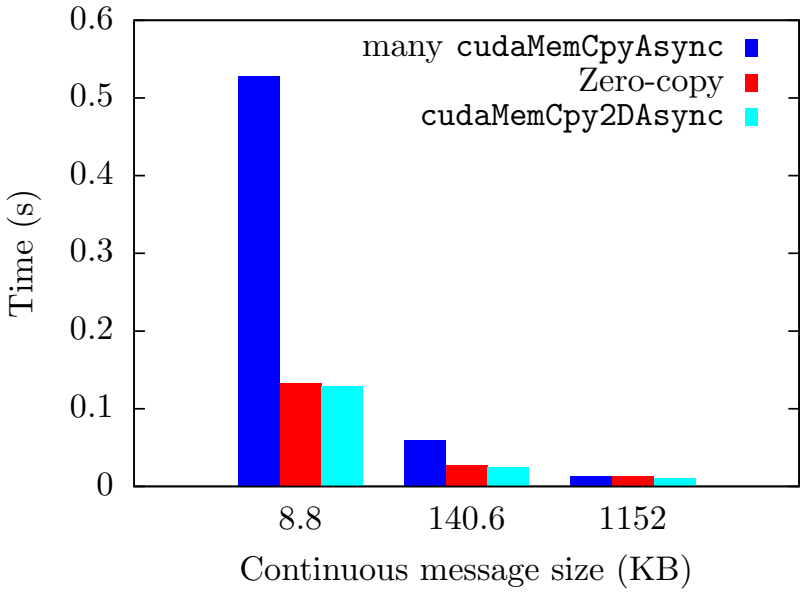

Figure 2.7: Time to transfer a total of 216MB of data with strided memory access using three different approaches. Since the total pencil size is fixed, smaller contiguous messages in this plot also correspond to more required looping over contiguous message chunks.

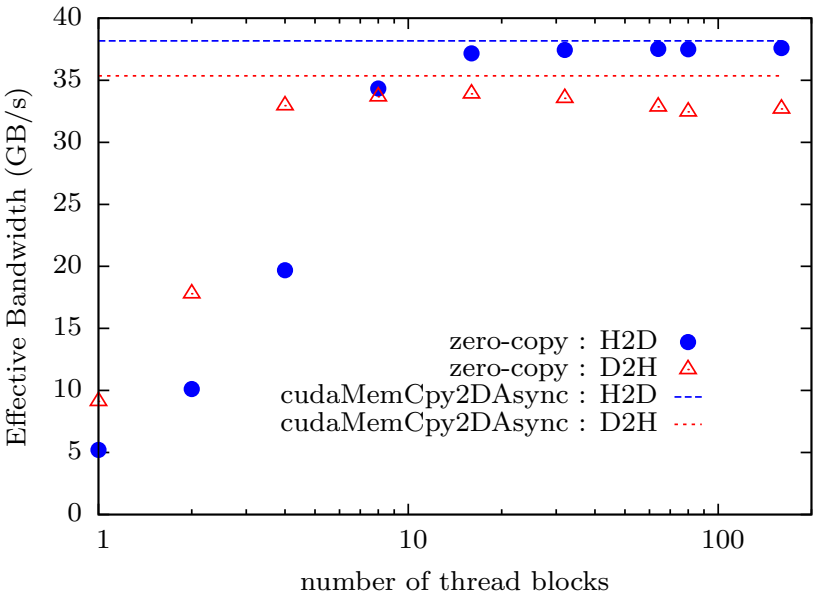

Figure 2.8: Effective bandwidth of zero-copy kernel (circles or triangles) compared to using **cudaMemCpy2DAsync** (dashed horizontal lines) for different numbers of thread blocks. The size of the thread block was 1024 threads.

To understand the GPU resources required by the zero-copy kernel to provide sufficient throughput, the behavior of the zero-copy kernel was studied using different numbers of

CUDA blocks, as shown in Figure 2.8. The kernel register usage of 32 registers per thread and the chosen thread block layout of 128×8 threads per block, allows two blocks to occupy a single SM. The **cudaMemCpy2DAsync** is a CUDA API call, not a CUDA kernel, and therefore does not use any blocks (or SMs). If the zero-copy kernel is provided with sufficient GPU resources, the bandwidth achieved is similar to **cudaMemCpy2DAsync**. It is observed that close to maximum throughput is attained even if using only a small fraction (about 16 blocks) of the GPU resources. This allows a small fraction of the GPU resources to be devoted to a zero copy kernel while simultaneously running other compute kernels on SMs of the GPU.

For best overall performance, **cudaMemCpy2DAsync** is still preferable to zero-copy kernels since the former allows all the GPU resources (streaming multiprocessors) to be available to the compute kernels. However, **cudaMemCpy2DAsync** can only handle simple strides (e.g., when copying a batch of data as shown in Figure 2.6), while the zero-copy kernel can handle data with complex stride patterns, (e.g., in unpacking data from contiguous to non-contiguous arrays after communication), while using up only a small amount of GPU resources. Thus, in the production code, most of the copying between host and device is implemented using **cudaMemCpy2DAsync**, while data unpacking is performed using the zero-copy kernel.

2.4 Performance Analysis

In this section, the overall performance of the newly developed DNS code at different problem sizes and node counts as described in subsection 2.2.5 is presented and analyzed, with all data obtained on Summit. Table 2.3 shows elapsed wall time per time step and speedup relative to performance data collected using the synchronous pencil decomposition CPU code that was used by Yeung *et al.* (2015) and is based on the principles described in subsection 2.2.1. Timings per step were obtained by taking the maximum over all MPI ranks, and taking the best over multiple time steps. Variability between MPI ranks and

Table 2.3: Performance of the slab decomposed DNS code run under different configurations and speedups are calculated with respect to the performance of the pencil decomposed synchronous CPU code.

Nodes	Problem Size	Sync CPU	Async GPU					
			6 tasks/node		2 tasks/node			
					1 pencil/A2A		1 slab/A2A	
		Time	Time	Speedup	Time	Speedup	Time	Speedup
16	3072^3	34.38	8.09	4.2	6.70	5.1	7.50	4.6
128	6144^3	40.18	12.17	3.3	8.66	4.6	8.07	5.0
1024	12288^3	47.57	13.63	3.5	12.62	3.8	10.14	4.7
3072	18432^3	41.96	25.44	1.6	22.30	1.9	14.24	2.9

between time steps was found to be minimal. It should be noted that for both CPU and GPU codes, regardless of the domain decomposition chosen, load balancing requires that the number of cores used per node to be an integer factor of the linear problem size (N). This implies, even though there are 42 cores per Summit node, only 32 cores can be used for most problem sizes except for the 18432^3 problem which uses 36 cores per node when run with 3072 nodes.

The best MPI configuration in the new asynchronous GPU algorithm gives the time to solution at a resolution of 18432^3 grid points, using 3072 nodes, at under 15 seconds per time step. Considering the problem sizes involved, this compares favorably with the largest simulations performed in the recent past (Yeung *et al.*, 2015; Ishihara *et al.*, 2016) using CPU-based massive-parallelism. A GPU-to-CPU speedup close to 3X was observed for the 18432^3 problem. Given the large problem size addressed and the communication-intensive nature of the application, this speedup is substantial.

2.4.1 2 versus 6 tasks per Node

The DNS code was approximately weak scaled (using more processes to solve a larger problem size with each process doing the same amount of work) on Summit, starting with a problem size of 3072^3 on 16 nodes up to 18432^3 on 3072 nodes. The scaling is approximate because the number of MPI ranks must be an integer factor of the number of grid

points on each side of the domain, while focusing on simulating the largest problem size that can fit into the memory available on the machine, at scale. Timings are reported for the three MPI configurations defined earlier in Table 2.2. In particular: (A) using 6 tasks per node, and communicating one pencil at a time; (B) using 2 tasks per node and communicating 1 pencil at a time (while overlapping MPI with GPU movement and compute); or (C) using 2 tasks per node but waiting to communicate until all pencils in the slab have been computed and can be sent in a larger message (no MPI overlap with GPU operations). In all three cases, data movement to/from the GPU and computation on the GPU are overlapped with each other, in the manner described in subsection 2.2.4. As might be expected for a communication dominated code, performance comparisons between these three MPI configurations follow the trends observed in the achieved MPI bandwidth studies of Table 2.2. Namely, 2 tasks per node perform better than 6 tasks per node and that at 3072 nodes sending an entire slab per all-to-all performs better than asynchronously sending each pencil as they become ready. Figure 2.9 shows the GPU timing data as well as a standalone MPI code which only performs the transposes without computation or data movement between the host and device. This helps us estimate standalone MPI costs; for example, the difference between the red line and dotted green line of Figure 2.9 is time spent in non-MPI operations, such as GPU kernels and GPU data transfer. This green line provides an upper bound on the best possible performance given the network characteristics of the machine used. Faster GPUs or optimization to the GPU kernels alone can at best approach the performance of the dotted green line. In other words, the dotted green line represents the best performance achievable in the limit of infinitely fast GPU operations.

2.4.2 Timeline and Asynchronous MPI Analysis

On Summit, the NVIDIA visual profiler (Nvidia, 2019) coupled to a Fortran interface to Nvidia's nvtx markers (Appelhans, 2018*a*) allows the visualization of a timeline of asynchronous CPU and GPU operations. To better understand the performance differences of

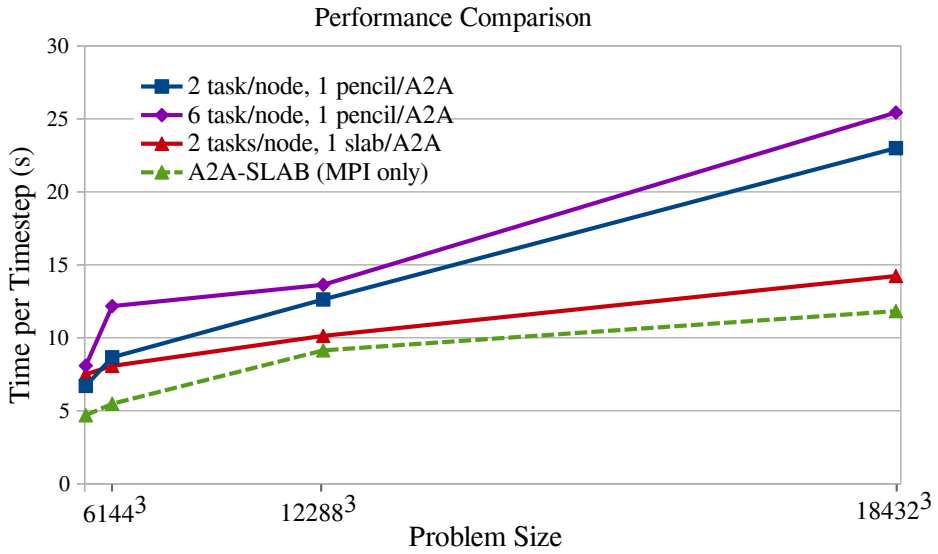

Figure 2.9: Time-per-step of the DNS slab code. The dotted green line is a benchmark of performing only the required MPI all-to-all calls (no computations). The solid lines are runs of the DNS code in various configurations benchmarked up to 3072 Summit nodes.

the code under the various configurations, Figure 2.10 shows plots of normalized (by the time per step) and aligned (to ensure the first operation shown starts at the same time on different frames) timelines for the various configurations running on 1024 nodes. These timeline plots are particularly illuminating because they directly show which parts of the code are contributing to the performance differences.

For example, the MPI time (shown in red) is immediately seen to account for the largest contribution to the overall run time. Comparison between the MPI times of the two uppermost timelines shows that MPI in the actual DNS code takes somewhat longer than in the standalone MPI code. This difference is not fully understood, but the results were very repeatable. It should be noted that both CPU-GPU data movement and MPI data movement simultaneously share the total bandwidth limit of the Power 9 memory. Our standalone tests revealed that if GPUs and the network card were requesting data movement, the MPI bandwidth suffered significantly until the GPU transfer was complete. However, even if the GPU data transfer times are subtracted from the MPI time in the second timeline, the MPI time still does not equal the MPI time without GPU operations. The discrepancy in timing is not fully understood, but another important consideration is that these timelines

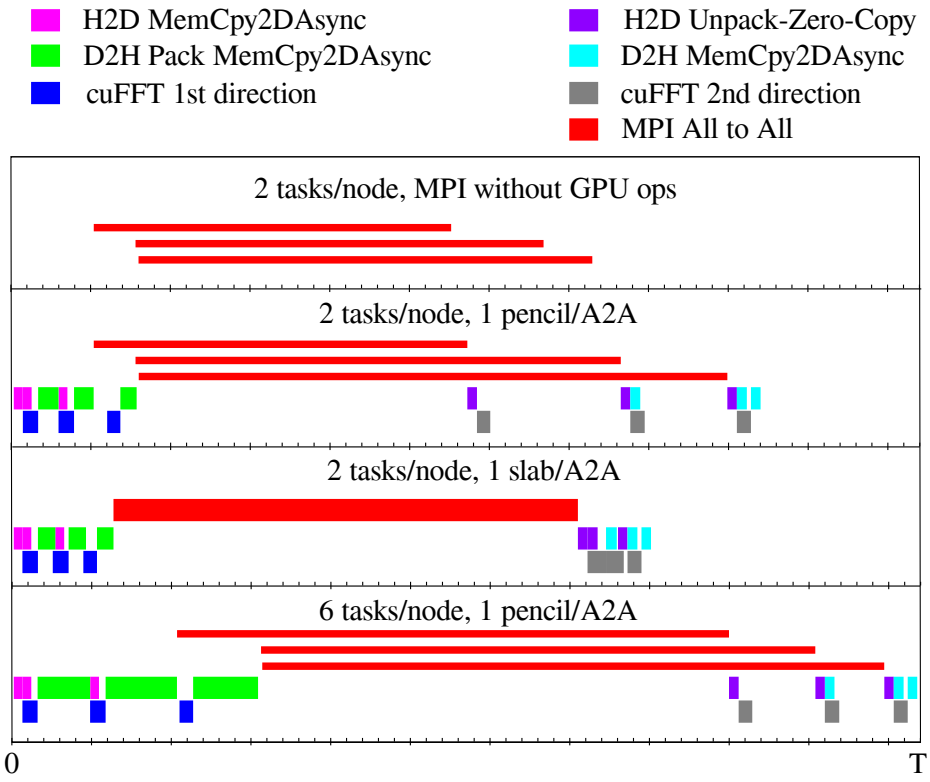

Figure 2.10: A normalized timeline comparison of various code configurations running the 12288^3 problem size on 1024 nodes. Each slab is divided into 3 pencils. The top timeline shows the behavior of an MPI only code that communicates the 3 pencils at the same points in time as the DNS code in the second timeline. The second timeline shows the actual behavior of the DNS code, where GPU operations are running asynchronously with the all-to-all communication. The third timeline shows the effect of waiting to send the entire slab in one MPI all-to-all call, instead of each pencil as it becomes ready. The final timeline shows the behavior of 6 tasks-per-node when each pencil is asynchronously sent as they become ready.

were generated using NVIDIA's profiler, which has non-trivial overhead and file system resource demands.

Comparison between the second and third timelines shows that the same amount of data can be transposed faster when processed as one, larger, message. This highlights a tradeoff: GPU operations can be overlapped with MPI communication of individual pencils (second timeline), but the resulting MPI calls will have smaller message sizes and must compete with GPU data movement bandwidth demands. Both factors work to slow down the MPI operations. The best approach is dependent on the problem size. At large problem sizes, the

individual messages in the all-to-all become smaller and the effective bandwidth drops (see Table 2.2). Beyond 16 nodes, waiting to send the entire slab at once (1 slab/A2A) is faster than overlapping computation with communications of a pencil at a time (1 pencil/A2A).

In the (bottom) timeline for the case of 6 tasks per node, each MPI call takes longer than in the case of 2 tasks per node (top). This is because the P2P message size in this all-to-all is small and there are more MPI tasks for each to communicate with, which increases latency costs. An additional drawback for this case of 6 tasks per node is that the D2H packing *MemCpy2DAsync* section of code takes much longer. This is because for the packing operation, the number of times **cudaMemCpy2DAsync** must be called is directly proportional to the number of tasks. For each GPU, the cases of 6 tasks per node and 2 tasks per node pack the same total size buffer, but with 6 tasks per node the packing must be done at a finer granularity. The number of copies required is now 3X that for the case of 2 tasks per node. This results in increased overhead, as seen in subsection 2.3.2. A zero-copy kernel can be used here, but it degrades the performance of the case of 2 tasks per node by utilizing GPU resources that are critical to the compute kernels.

The last takeaway from these timelines is that for the 2 task per node cases, the MPI cost is dominating the runtime of the code. The overhead incurred in choosing to batch data between CPU and GPU is not significant compared to the total runtime. Yet, by batching, a larger problem can be solved using the much larger CPU memory. Further gains in performance will be contingent upon code redesigns and hardware innovations can improve the performance of the all-to-all communication.

2.4.3 Scalability and efficiency of asynchronism

As noted throughout this chapter, the core objective is to solve large problems in a reasonable amount of time. In most cases, the focus has been on the largest problem size that can fit inside the memory available for a given node count. As a result, at each problem size timings over only a narrow range of node counts can be collected, making inferences

on strong scaling (to solve the same problem faster if given more resources) of limited relevance in this work. The discussion of scalability, as presented below, is this focused on weak scaling (to solve a bigger problem in proportionate time if given more resources).

The communication-intensive nature of codes dominated by 3D FFTs imply perfect weak scaling is not achievable (Pekurovsky, 2012; Czechowski *et al.*, 2012; Dalcin *et al.*, 2018; Chatterjee *et al.*, 2018). Incidentally, when computation are greatly sped up by the GPU accelerator at only modest cost for copies between host and device, the code becomes even more communication-intensive (dominated by all-to-all communication), potentially at the expense of scalability. MPI benchmarks given in Table 2.2 earlier indicate that algorithmic choices that leads to a small number of large messages are usually beneficial. However, eventually at the large scales the latency of small message sizes gives rise to longer MPI communication times (and reduced net bandwidth).

Table 2.4: Weak scaling relative to 3072^3 problem size.

Nodes	Ntasks	Problem Size	# pencils per A2A	Time (s)	Weak Scaling (%)
16	32	3072^3	1	6.70	-
128	256	6144^3	3	8.07	83.0
1024	2048	12288^3	3	10.14	66.1
3072	6144	18432^3	4	14.24	52.9

The weak scaling percentage (WS) is calculated between two problem sizes N_1^3, N_2^3 of node counts M_1, M_2 with execution times t_1 and t_2 respectively using the formula

$$WS = \frac{N_2^3}{N_1^3} \times \frac{t_1}{t_2} \times \frac{M_1}{M_2} \, . \tag{2.4}$$

Table 2.4 shows weak scaling computed with respect to the 3072^3 (16 node) problem size, using the best performance timings for each problem size as recorded earlier in Table 2.3. Considering that between 3072^3 and 18432^3 the number of grid points has increased by a factor of $6^3 = 216$, a weak scaling of 53% is actually very reasonable, when considering the communication-intensive nature of the code.

Some information on strong scaling was also obtained using the expression below,

$$SS = \frac{t_1}{t_2} \times \frac{M_1}{M_2}, \tag{2.5}$$

where t_1 and t_2 are the execution times for a fixed problem size on two different node counts M_1 and M_2 respectively (such that $M_2 > M_1$). Specifically, for the 18432^3 problem size, a doubling of node count from 1536 nodes to 3072 nodes leads to a reduction of wall time per step from 48.7 to 25.4 seconds. This gives a strong scaling of 95.7%, which is excellent.

Table 2.5: Performance data of batched synchronous and asynchronous DNS code on Summit. The percent reduction in time for the compute and data copy region of the batched asynchronous code is computed with respect to a batched synchronous version.

Nodes	Problem Size	Sync GPU		Async GPU			
		Comp & Copy (s)	Total (s)	Comp & Copy (s)	MPI (s)	Total (s)	% saved
16	3072^3	3.52	8.05	2.55	4.56	7.11	28
128	6144^3	3.81	8.76	2.68	4.94	7.62	30
1024	12288^3	4.31	10.52	3.29	6.30	9.59	24
3072	18432^3	6.90	14.92	5.93	7.98	13.91	14

Figure 2.10 presented earlier serves as a good reference to visualize the asynchronous nature of the algorithm. However, it is also useful to quantify the performance savings obtained from overlapping different operations. Table 2.5 reports the efficiency of overlap calculated with respect to the batched synchronous code for just compute and data copy regions in the code. The percentage of time saved in overlapping the data copies and computations are computed using $(t_{sync} - t_{async})/t_{sync}$, where t_{sync} and t_{async} correspond to the time to copy and compute in synchronous and asynchronous versions of the code, respectively. The performance numbers for the batched synchronous version were obtained using the same batched asynchronous CUDA Fortran code with the environment variable **CUDA_LAUNCH_BLOCKING** turned on to disable any GPU asynchronism (between computations and data copies). There is no overlap with respect to MPI as discussed in

subsection 2.4.2. Therefore, in the batched asynchronous version only data copies and computations are overlapped. A performance saving of around 15% is observed from enabling asynchronism in the code for 18432^3 with larger performance savings at smaller problem sizes.

Among the two overlapped operations (data copies and computations), the copies are dominant, especially for large problem sizes as the strides in memory increase while the computations, especially the **cuFFTs** are almost trivial. Therefore, the best achievable performance is obtained when the data copies completely overlap all computations. Figure 2.10 shows that the newly developed code is indeed successful in hiding most of the computations under the data copies, with some inefficiencies that are inevitable.

2.4.4 A performance model towards the exascale

The timings reported in Table 2.5 can be used to establish a performance model that can help predict the performance of the algorithm on future architectures based on expected system characteristics. The performance model is developed to predict the effect of future hardware improvements like faster network interconnect or CPU-GPU bandwidths on the overall performance of the algorithm before having any opportunity to actually run tests on these architectures, especially at large node counts. Noting that the data copies overlap computations almost entirely, the compute+copy timings reported under the asynchronous column of Table 2.5 can be safely assumed to represent the data copy timings alone. The difference in compute+copy timings reported under the synchronous and asynchronous columns gives the time spent in computations. The table also reports the time spent in performing the MPI all-to-all.

The data movements in the batched asynchronous algorithm are dominated by strided memory copies, as described in subsection 2.3.2. Figure 2.7 presented earlier shows that the time to copy a fixed volume of data varies with the size of the contiguous messages involved in the copies. The sizes of the contiguous messages illustrated in Figure 2.6

earlier are dependent on the number of grid points in one direction (N) and the number of batches (or pencils) per slab (np) in the form N/np (with the strides being $(np-1)N/np$). A careful counting of the number of data copies between the host and device in a single time-step reveals that a total of 86 slabs of data are copied, each of size N^3/P (as described in subsection 2.2.5).

The average time to copy a volume of data, $Q_{copy} = 86N^3/P$, depends on the peak GPU-CPU interconnect bandwidth (which is $BW_{GPU} = 150GB/s$ per task) and can be estimated as $t = Q_{copy}/BW_{GPU}$. However, it is not likely that the copies are all performed at peak bandwidth. A factor α is introduced to account for this, such that the effective bandwidth achieved is given by $BW_{GPU,eff} = \alpha BW_{GPU}$. The effective bandwidth further reduces when strided copies are performed. The time taken to perform these strided copies may be modeled as,

$$t_{copy} = Q_{copy}\left(\frac{1}{\alpha BW_{GPU}} + \frac{1}{\beta}\right), \tag{2.6}$$

where, β accounts for the effects of strided copies. It can be observed from Figure 2.7 that β will depend on the memory stride. After some curve fitting experiments, it was found that the β is roughly inversely proportional to the square of the memory strides, i.e., $\beta \sim 1/S^2$, where S is the memory stride which depends on problem size and number of pencils per batch (n_p) as $S = (n_p - 1)N/n_p$. The resulting model parameters from linear curve fits are, $\alpha \sim 0.82$ (which suggests 82% of peak bandwidth is achieved for non-strided copies, i.e., $BW_{GPU,eff} = 123\ GB/s$) and $\beta = 235000/S^2$ where S is the memory stride in units of KB. Figure 2.11a shows the performance data and the corresponding model fit described here. Because 18432^3 problem is not an exact weak scaling of the smaller problem sizes, the corresponding copy and MPI timings are scaled to account for this. Reasonably good agreement between the model and the actual data can be observed from this figure.

Many models exist in the literature to capture the behavior of collective communication calls, like all-to-all (Hoefler *et al.*, 2010; Pjesivac-Grbovic *et al.*, 2005; Hongbing *et al.*, 2018). However, these models are complex and require estimates for some key network

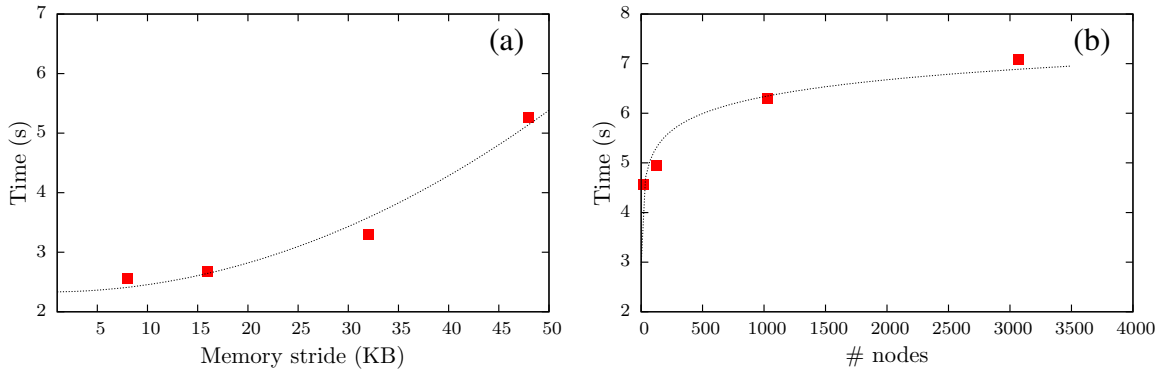

Figure 2.11: (a) Time spent in copying data between the GPU and CPU for different problem sizes, each corresponding to a different amount of stride in memory. (b) Time spent in MPI all-to-all for different problem sizes. The dashed curve shows the model fit.

parameters like latency, congestion and overhead. From an application developer perspective, such a model is not very useful. It is more desirable to develop a model that predicts the performance of an all-to-all based on input parameters such as the number of nodes (M) or the message size exchanged between each node. Curve fitting experiments, using data in Table 2.5, suggest the MPI all-to-all performance has a logarithmic dependence on the number of nodes (M) when the problem size is exactly weak scaled.

$$t_{mpi} = Q_{MPI}\left(\frac{1}{\gamma BW_{MPI}} + \frac{ln(M)}{\delta}\right), \tag{2.7}$$

The DNS code requires 14 slabs of data to be transposed per time-step, which leads to a total volume of data of $Q_{MPI} = 14N^3/P$. Equation 2.7 shows the all-to-all performance model considered here, where the effective MPI bandwidth $BW_{MPI,eff} = \gamma BW_{MPI}$ has a similar meaning to $BW_{GPU,eff}$. The interconnect on Summit is rated at a peak bandwidth of 23GB/s and the model fit parameters γ and δ are estimated to be around 0.7 and 96 respectively. Figure 2.11b shows the MPI performance data and good agreement can be seen with the corresponding model.

Using the two performance models developed above, the total performance of the code

can be estimated using the following equation,

$$t = Q_{copy}\left(\frac{1}{\alpha BW_{GPU}} + \frac{1}{\beta}\right) + Q_{MPI}\left(\frac{1}{\gamma BW_{MPI}} + \frac{ln(M)}{\delta}\right), \tag{2.8}$$

where α, β, γ and δ are the four model parameters, M, Q_{copy} and Q_{MPI} are user inputs and BW_{GPU}, BW_{MPI} are system parameters. The batched asynchronous code was also tested using 4096 nodes for a 24576^3 problem size and the performance measured was around 29.5s which compares favorably to the model prediction of 28.7s.

It is useful to re-iterate here that the model developed is not always applicable and has limitations. The curve fit parameters estimated here assume a particular system architecture with a fat tree topology interconnect. The MPI performance model was estimated using data corresponding to a weak scaled problem, which has a fixed total message size (although the peer-to-peer message size varies). The model parameters are expected to change when the total message size varied. Also, the model parameters might not be accurate for other machines with different hardware and needs to be re-tuned. Moreover, the performance model will work best with weak scaled cases and problem sizes that deviate from this condition might not be captured accurately. The model also assumes that the computations were completely hidden by the data copies, which might not always be the case.

With these limitations in mind, the model can be used to estimate the performance of the batched asynchronous algorithm for a few problem sizes on Frontier (OLCF, 2021), which is a future exascale machine with similar architecture to Summit. The machine is expected to have faster network interconnect with a peak bandwidth of around 100GB/s ($BW_{MPI} = 100$) (OLCF, 2021). The peak bandwidth for copies between the CPU and GPU is assumed to be around $BW_{GPU} = 200$GB/s. The target problem size will be 32768^3 on 8192 Frontier nodes with 4 tasks per node. This leads to Q_{copy} and Q_{mpi} being 344GB and 56GB respectively. With all the model parameters known, the performance can be

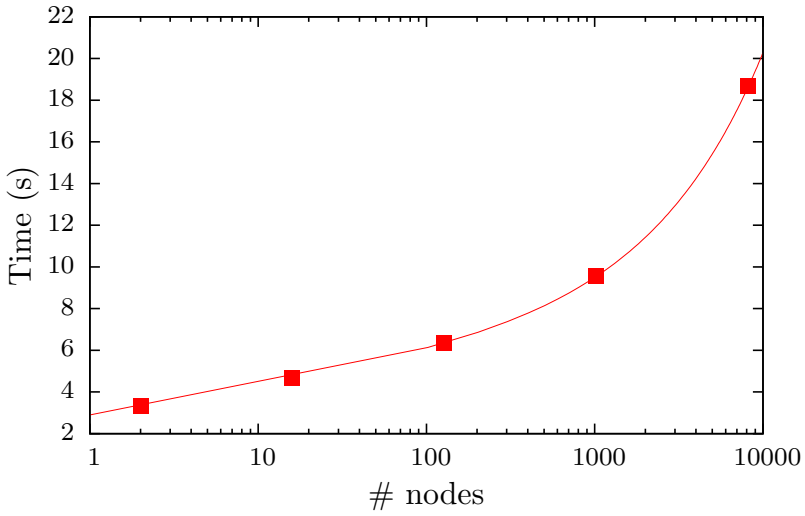

Figure 2.12: Performance estimates for the batched asynchronous algorithm on Frontier using the performance model developed. The markers indicate the estimated time-per-step for problem sizes 2048^3, 4096^3, 8192^3, 16384^3 and 32768^3 on 2, 16, 128, 1024 and 8192 nodes respectively.

estimated and is shown in Figure 2.12. The model suggests that using 8192 nodes of Frontier, the 32768^3 problem size should take around 19s per time-step. As mentioned previously, the performance model does not account for modern hardware architecture and improvements in the algorithm, but the estimate here is obtained based on basic system parameters and the characteristics of the batched asynchronous algorithm.

2.5 Particle tracking algorithm using GPUs

The Lagrangian viewpoint of fluid dynamics (Yeung & Pope, 1988; Yeung, 2002; Sawford & Pinton, 2013) is based on studying infinitesimal mass-less fluid elements, referred to as fluid particles, traveling with the instantaneous flow. A large number of these fluid particles are tracked forward in time according to a simple ordinary differential equation,

$$\frac{d\mathbf{x}^+(t)}{dt} = \mathbf{u}^+(t) \tag{2.9}$$

where, $\mathbf{x}^+$ and $\mathbf{u}^+$ denote the particle position and velocity respectively, and $\mathbf{u}^+$ is the Eulerian velocity taken at the instantaneous particle position

$$\mathbf{u}^+(t) = \mathbf{u}(\mathbf{x}^+(t), t) \tag{2.10}$$

Direct numerical simulation is a very powerful approach for Lagrangian investigations, since the Eulerian velocity field available at (N^3) grid points allows the fluid particle velocity to be obtained by interpolation (per Equation 2.9) at its instantaneous position. A preferred interpolation scheme is by cubic splines, which are fourth-order accurate and twice continuously differentiable. Each interpolated velocity can be written in the form

$$u^+ = \sum_{k=1}^{4} \sum_{j=1}^{4} \sum_{i=1}^{4} b_i(x^+) c_j(y^+) d_k(z^+) e_{ijk}(\mathbf{x}) \tag{2.11}$$

where b_i, c_j, d_k is the 1-D basis functions that are determined by the particle position coordinates ($\mathbf{x}^+ = (x^+, y^+, z^+)$) and e_{ijk} are the 3-D spline coefficients that are computed from the Eulerian velocity field $\mathbf{u}(\mathbf{x})$. The workflow to compute the 3-D spline coefficients are similar to that for 3-D FFTs. That is, 1-D spline coefficients in each of the three directions are formed with an all-to-all transpose in between from $x - z$ slabs to $x - y$ slabs. A system of tridiagonal systems modified to account for periodic boundary conditions is solved in each coordinate direction (Ahlberg *et al.*, 1967).

The batched asynchronous approach implemented for the velocity field (Ravikumar *et al.*, 2019) has been adapted to compute the spline coefficients. At each time step, once the velocity components are formed in physical space, the particle tracking proceeds, starting with velocities in $x - z$ slabs stored in a temporary array on the host. A batch (pencil oriented along the x direction) of the velocity field is copied from the host to the device, where CUDA Fortran kernels are used to compute the spline coefficients in the x direction. The partially formed spline coefficients are then copied back to the host. This process is then repeated with batches of data oriented along the z direction. The copies of batches

of data between the host and device are performed using **cudaMemCpy2DAsync**. After operations in the z direction, the memory copies, using **cudaMemCpy2DAsync**, from the device to the host are performed such that the data on the host arrives in a contiguous form ready for an all-to-all transpose. After this communication, the data are unpacked in batches, oriented along the y direction, using **zero-copy** kernels before splines in y are completed as well. When multiple GPUs per task are available, the batch is further divided up vertically, as shown in Figure 2.5, and each GPU operates on a smaller sub-division of a batch of data.

Once the spline coefficients are formed completely, particle velocities can be obtained following Equation 2.11. For each particle, $4^3 = 64$ spline coefficients and basis functions are required based on the particle positions. In the original implementation (Yeung & Pope, 1988), each MPI task has access to all the particles and could form the interpolated velocities. The basis functions are computed by each task for a subset of particles, and **MPI_ALLGATHER** is used to share this with all the tasks. The interpolation is then performed by each task for particles whose spline coefficients are local to it. **REDUCE** and **SCATTER** operations are required to complete the interpolation and share the final results globally.

As explained in Buaria & Yeung (2017) the original implementation incurs high communication costs at the large problem sizes and particle counts essential in the future. Instead, a "local particle decomposition" gives much better performance. In this approach, each MPI task is responsible for the particles that are within its physical sub-domain. If a particle crosses into a different sub-domain kept by a neighboring MPI task, then all of its information is transferred to this new MPI task. The main idea is to keep the particles 'local' to the MPI tasks, such that all information necessary for interpolation are available locally on an MPI task. To handle particles very close to the boundary of a sub-domain, spline coefficients from neighboring MPI tasks are obtained by forming the so-called "ghost layers", which hold two extra $x - y$ planes of spline coefficients on each side of the slab (It should

be noted that, instead of forming ghost layers, Buaria & Yeung (2017) employed one-sided communication using Fortran co-arrays, which however is not currently widely supported). These ghost layers are populated using one-sided MPI communication (**MPI_GET**). This allows the summation in Equation 2.11 to be formed entirely locally, and has led to very good performance, with virtually no dependence on the number of particles.

Table 2.6: Performance breakdown of slab decomposition DNS code with particle tracking for different problem sizes (N^3, where k stands for 1024) and particle counts (N_p, where M and B stand for millions (1024^2) and billions (1024^3) of particles respectively). The spline coefficients were formed on the GPU using the batched asynchronous approach. The time spent in computing the 1-D spline coefficients and data copies are shown along with the time spent in the all-to-all. The time spent in forming the ghost layers is given under the Ghost column, and the time to compute the triple summation locally is reported under spcal. All cases were run with 2 tasks-per-node and the weak scaling percentage (WS) is calculated with respect to performance numbers at 16 (or 32) nodes using Equation 2.4. The strong scaling percentage (SS) is calculated for a fixed problem size using numbers at two different node counts according to Equation 2.5.

No.	N^3	N_p	# node	Time per step						WS	SS
				Eul.	Spline coeff.		Ghost	spcal	Total		
					comp	comm					
1	$3k^3$	6M	16	7.27	0.86	0.99	0.24	0.040	9.42	-	-
2	$6k^3$	48M	128	7.85	0.69	1.13	0.35	0.043	10.08	93	-
3	$12k^3$	384M	1024	10.29	1.12	1.67	1.01	0.065	14.19	66	-
4	$18k^3$	1.5B	3072	15.02	1.88	2.45	2.06	0.083	21.50	49	-
5	$3k^3$	48M	16	7.28	0.86	0.98	0.24	0.29	9.87	-	-
6	$6k^3$	384M	128	7.83	0.68	1.13	0.35	0.33	10.63	93	-
7	$12k^3$	3B	1024	10.27	1.11	1.71	0.99	0.49	14.62	68	-
8	$3k^3$	6M	32	3.69	0.43	0.50	0.14	0.020	4.78	-	99
9	$6k^3$	48M	256	4.97	0.45	0.84	0.25	0.023	6.55	73	77
10	$12k^3$	384M	2048	6.08	0.92	1.08	0.92	0.038	9.05	53	78
11	$3k^3$	48M	32	3.68	0.43	0.50	0.14	0.15	5.00	-	99
12	$6k^3$	384M	256	4.94	0.46	0.74	0.25	0.18	6.72	74	79
13	$12k^3$	3B	2048	6.16	0.92	1.07	0.91	0.21	9.29	54	79

The performance of the code with the spline coefficients computed using the batched asynchronous approach on GPUs is shown in Table 2.6. The performance of the Eulerian only part of the code reported in this table is consistent with that seen in Table 2.3 earlier, except for the larger problem sizes (12288^3 and 18432^3) where the number of batches per

slab are higher (4 instead of 3 as in the Eulerian only runs) to accommodate additional memory requirements for spline coefficients. It may be noted that the scalability of the routines used to generate cubic spline coefficients is at present lower than desired. For example, weak scaling between Cases 6 and 7 for the computational and communication contributions for spline coefficients are only 61% and 66% respectively, while the corresponding strong scaling between Cases 7 and 13 are 60% and 80%. These numbers point to a need for further improvement in the future.

The all-to-all scalability issues affecting the Eulerian only part of the code also affects the performance of the particle tracking algorithm. The formation of the ghost layers have also been observed to scale not as well as desired. This is because the size of the ghost layers being exchanged scale as N^2. Additionally, using **MPI_GET** to populate the ghost layers without any prior data packing, requires strided memory access to select the last two planes from the neighboring MPI task in the bottom. That is, if there are N/P planes per slab each of size N^2, the first $N/P - 2$ planes need to be strided over to reach the $(N/P - 1)$th plane which will be the starting memory address for the communication to begin. This stride in memory ($N^2 \times (N/P - 2)$) decrease as the problem size N increases. A combination of these two effects, namely increase in message size and decrease in memory stride, leads to an overall drop in weak scaling.

The number of fluid particles being tracked has no effect on the spline coefficients and the ghost layer formation performance. It only affects the performance of the interpolation (triple summation) which scales near ideally with the particle count. The GPUs are not used for these parts of the particle tracking algorithm because the ghost layer formation is purely communication and the triple summation would have to be written using a batched approach because the entire slab of spline coefficients may not fit in the GPU memory. This also comes with the additional memory cost (buffers to hold particle positions, velocities, etc.) of tracking the particle properties on the GPUs. Since the time spent in this part of the code is not significant at the particle counts tested, it has not been ported to GPUs yet.

Table 2.7: Performance of spline coefficients formation routine using CPUs only in the slab decomposition DNS code. All cases were run with 2 tasks-per-node, and the weak scaling is calculated with respect to performance numbers at 16 (or 32) nodes. The number of particles tracked for problem sizes 3072^3, 6144^3 and 12288^3 was 6M, 48M and 384M respectively.

Problem Size	# nodes	Time per step						Weak Scaling (%)	Strong Scaling (%)
		Eul.	Spline oeffs		Ghost	spcal	Total		
			comp	comm					
3072^3	16	7.27	3.69	1.02	0.24	0.040	12.30	-	-
6144^3	128	7.81	4.19	1.58	0.35	0.043	14.01	88	-
12288^3	1024	10.21	6.40	2.51	1.05	0.063	20.25	61	-
3072^3	32	3.70	1.85	0.52	0.14	0.020	6.26	-	98
6144^3	256	4.96	2.13	1.19	0.25	0.023	8.56	73	82
12288^3	2048	6.12	3.57	1.53	0.94	0.042	12.21	51	83

For the largest problem size tested, the weak scaling recorded was 49% comparing cases 1 and 4 in Table 2.6 while the problem sizes between the two cases increased by a factor of 216. This is lower than that recorded for the Eulerian only case reported in Table 2.4 previously because of the additional communication requirements from the spline coefficients and ghost layers in the particle tracking algorithm. A strong scaling of approximately 80% is also observed comparing cases 7 and 13.

In order to understand the speedup obtained by computing the spline coefficients on the GPUs, the performance of the code is also measured with the splines computed using the CPUs only, as shown in Table 2.7. For the problem size of 12288^3 tested using 1024 nodes, the computations of the spline coefficients using the GPUs take 1.1s (Table 2.6) while using the CPUs it takes 6.4s (Table 2.7), giving a speedup close to 6X. A further performance improvement in communication is obtained by changing the all-to-allv used in the CPU version to the simpler all-to-all with slightly larger message sizes.

The performance of the CPU-only pencil decomposition DNS code with particle tracking using the same number of nodes with 32 tasks per node is reported in Table 2.8. Not all the 42 cores per node can be used because in the 2D (pencils) decomposition the row communicator dimension P_r must be an integer factor of the number of grid points in each

Table 2.8: Performance of pencil decomposition DNS code with particle tracking. All cases were run with 32 tasks-per-node with a domain decomposition of the form $32 \times nnodes$, where $nnodes$ is the number of nodes. The speedup of the slab decomposition batched asynchronous DNS code using GPUs with respect to the pencil decomposition DNS code is also reported.

Problem Size	# particles	# nodes	Time per step	Weak Scaling (%)	Speedup
3072^3	6M	16	51.09	-	5.4
6144^3	48M	128	62.33	82	6.2

direction. The time spent in the Lagrangian part of the code can be obtained by subtracting out the time spent in the Eulerian part alone, reported in Table 2.3. The pencil decomposition code appears to weak scale, worse than the slab decomposition code up to the largest problem size tested of 6144^3. The speedup of the GPU based batch asynchronous code is around 6X, corresponding to a problem size of 6144^3, which also happens to be the largest problem size tested using the pencil decomposition. The speedup is around twice of that observed for the Eulerian only runs, which suggests a similar speedup holds for the Lagrangian parts of the code as well.

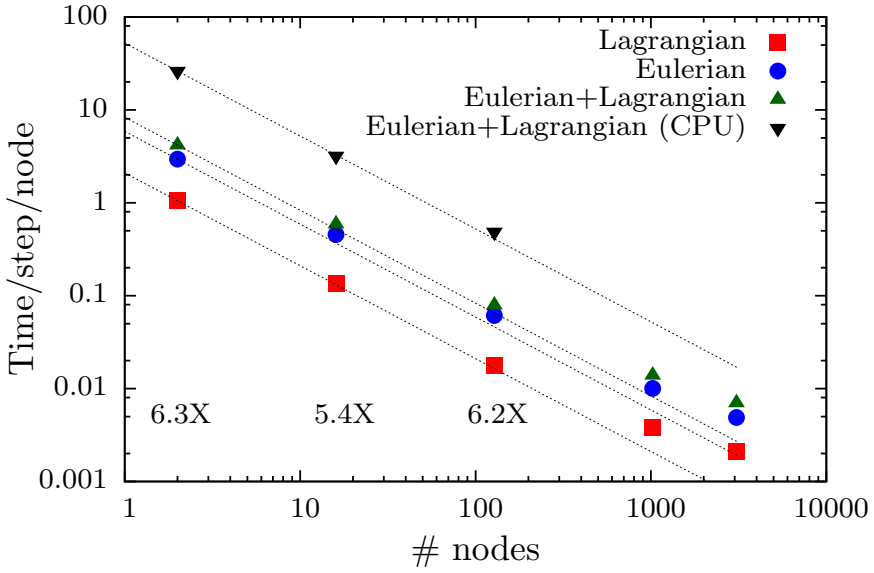

Figure 2.13: A scaling plot showing performance data on Summit for simulations without and with Lagrangian particle tracking.

Performance data for the Lagrangian simulations for different problem sizes are pre-

sented in Figure 2.13. It can be seen that scalability is well maintained for both Eulerian and Lagrangian operations, except for the largest two problem sizes. Ideally, weak scaling of 12288^3 on 1024 nodes suggests that the 18432^3 problem size should be run on 3456 nodes. However, since in a 1D decomposition the number of MPI processes must be an integer factor of the number of grid points in each direction, this problem size was run using 3072 nodes instead. GPU acceleration measured relative to a CPU only code (albeit one that uses a less-efficient nonlocal decomposition of particles) is in the range of 5-6.

2.6 OpenMP implementation strategy

Recent code development efforts, as a part of the Center for Accelerated Application Readiness for Frontier program led by the Oak Ridge Leadership Computing Facility (OLCF), have focused on adapting the batched asynchronous algorithm, described in subsection 2.2.4, to the new hardware architecture and software environment of one of very first exascale computers in the world (to be online by early 2022). With portability as the primary goal, OpenMP is used to target the GPUs. The CUDA API calls used on Summit to allocate memory and copy data between the host and device are likewise to be replaced by OpenMP **MAP**s and **TARGET UPDATE** constructs. OpenMP **TASK**s with **DEPEND** and **NOWAIT** clauses are used instead of CUDA **streams** and **events** to enforce the desired asynchronism. The one-dimensional FFTs are also changed to use the ROCm library instead of CUDA. Proper interoperability between non-blocking library calls, like **rocFFT**, will require use of the OpenMP 5.0 **DETACH** construct. Finally, the computations performed using the CUDA Fortran kernels will be ported to target the GPUs using OpenMP **TARGET TEAMS DISTRIBUTE PARALLEL DO** regions.

Memory allocators introduced in the OpenMP 5.0 standard are expected to ensure the host arrays involved in data copies are **pinned** in memory to achieve maximum bandwidth. The device memory is managed through the **TARGET DATA** regions, with **MAP** clauses used to allocate memory and copy data to/from the GPUs. Additional data copies of con-

tinuous memory sections between an array already allocated on the device and the host are performed using **TARGET UPDATE** constructs.

In the "batched" scheme described earlier in this chapter, when FFTs in y need to be computed, a sub-volume of data consisting of $N \times N/P$ lines of size N/np need to be copied to the GPU. That is, for each value of x (between 1 and N) and z (between 1 and N/P), N/np elements from the innermost dimension of size N elements, which are contiguous in memory, needs to be copied. A schematic of 2D strided copy of the data is shown in Figure 2.6 earlier. Efficient strided data transfers between the CPU and GPU are thus important. Simple approaches such as packing on the host prior to transfer, or performing multiple copies one line at a time are inefficient, because of an extra data-reordering operation on the CPU and the overhead of numerous smaller copies respectively (Ravikumar *et al.*, 2019). Instead, two different approaches are used (as described in subsection 2.3.2), depending on the complexity of the strided memory accesses.

For some strided copies, the device memory routine, **omp_target_memcpy_rect** can be used to copy a specified sub-volume inside a larger array on the host to a smaller buffer on the device, or vice versa. Since this routine is directly callable only from C/C++, when making copies we need to account for the difference in memory order between C and Fortran. A C-to-Fortran interface has been developed to enable calling this routine from a Fortran program. This OpenMP routine is conceptually similar to **cudaMemcpy2d**. The OpenMP 5.1 standard supports an asynchronous version, **omp_target_memcpy_rect_async**, which will allow for some overlap between strided data copies and computations using OpenMP. However, the rectangular memory copy calls were found to be very slow on both Summit and Spock, which is the early access system for Frontier. While the performance issues are being addressed by the compiler developers, the **hipMemCpy2DAsync** API call can be used to to copy data between the host and device, until the performance of **omp_target_memcpy_rect** is improved and support for the asynchronous version is available.

```
1  TARGET ENTER DATA MAP(to:d_buf) DEPEND(IN:indep) DEPEND(OUT:tdep) NOWAIT
2
3  TARGET TEAMS DISTRIBUTE PARALLEL DO COLLAPSE(4) IS_DEVICE_PTR(h_buf)
    DEPEND(INOUT:tdep) NOWAIT
4  do yg=1,nt
5      do z=1,mz
6          do y1=1,my
7              do x=1,N
8                  y = my*(yg-1)+y1
9                  d_buf(x,y,z) = h_buf(x,z,y1,yg)
10             end do
11         end do
12     end do
13 end do
14 END TARGET TEAMS DISTRIBUTE PARALLEL DO
15
16 TARGET EXIT DATA MAP(from:d_buf) DEPEND(IN:tdep) DEPEND(OUT:outdep)
    NOWAIT
```

Figure 2.14: Asynchronous OpenMP implementation of the **zero-copy** kernel for unpacking data from the pinned host array (**h_buf**) to the device array (**d_buf**). In this example, nt is the number of MPI ranks and $my = mz = N/nt$ where N is the number of grid points in each direction.

For more complex stride patterns, like those in unpacking operations, a **zero-copy** kernel (Appelhans, 2018*b*) is used. In this approach, GPU threads are used to initiate many small transfers between pinned memory on the host and the device memory. Figure 2.14 shows the asynchronous OpenMP implementation of the zero-copy kernel. The array residing on the host is made device accessible using the **IS_DEVICE_PTR** clause. GPU threads can then directly access data on the host in a strided manner and copy them to the device. The kernel is currently housed in a separate subroutine and the host buffer (**h_buf**) is passed in to it, since the arguments to the **IS_DEVICE_PTR** clause have to be dummy arguments. Asynchronous execution is enabled using the DEPEND and NOWAIT clauses, with the arguments to the depend clause (**indep** and **outdep**) being inputs to the subroutine.

In order to perform one-dimensional Fourier transforms, the code will call the **rocFFT** library functions. These functions are written in C/C++ but they can be called by Fortran routines via vendor-provided interfaces such as hipfort. The FFT plans are created such

that they execute on a non-blocking stream, obtained using **hipStreamCreate**. The library function is called from the host, but the device arrays need to be passed in to it. This is achieved using the **TARGET DATA USE_DEVICE_PTR** clause, which tells the OpenMP runtime to pass the device pointer of the array, already allocated through a **MAP** clause, to the library call.

As noted in earlier sections, the DNS code is communication intensive, with all-to-all library calls being dominant. On Frontier, the new slingshot interconnect (De Sensi *et al.*, 2020), is expected to help reduce this performance bottleneck. The code is also being developed with the capability to perform GPU-aware MPI, which is the recommended method of communication on Frontier for optimal performance, with the additional advantage of removing the need for some host-device copies. Similar to **rocFFT**s, the **USE_DEVICE_PTR** clause will be required to pass device pointers to the MPI library function.

In OpenMP, asynchronous execution can be achieved using the **TASK** clause for work on the host, **NOWAIT** for device kernels and data copies, as well as **DEPEND** to enforce the necessary synchronization between different *tasks*. However, when non-blocking libraries such as **cuFFT** or **rocFFT** are called from inside an OpenMP task, the desired asynchronism breaks down. Figure 2.15 shows a 1D FFT example to highlight the proposed approach to mix non-blocking FFT calls and other tasks such as data copies or computes on the GPU. This example consists of three main tasks in the example. Task A calls the non-blocking libraries to transform the data in the forward and inverse directions. Task B performs a host-to-device data copy, and Task C multiplies the data by a scalar after transformation. First, let us consider what happens without the highlighted gray lines. The host thread that is executing task A will launch the FFT kernels to the GPU, but since these library calls are non-blocking the control will return immediately back to the host thread which will proceed to end the task. Therefore, the OpenMP runtime environment does not ensure the launched device kernels have completed or even started running when task A is

```
1  TARGET DATA MAP(tofrom: a)
2
3  TASK DEPEND(out:var) DETACH(event)                                  } Ⓐ
4
5  TARGET DATA USE_DEVICE_PTR(a)
6  FFTExecute (a, forward, stream)
7  FFTExecute (a, inverse , stream)
8  END TARGET DATA
9
10 cudaStreamAddCallback (stream, ptr_callback, C_LOC(event), 0)
11 END TASK
12
13 TARGET UPDATE TO(b) DEPEND(inout:b) NOWAIT                          } Ⓑ
14
15 TARGET TEAMS DISTRIBUTE DEPEND(IN:var) NOWAIT                      } Ⓒ
16 a (:, :, :) = a (:, :, :) /nx
17 END TARGET TEAMS DISTRIBUTE
18
19 END TARGET DATA
```

```
1  subroutine callback (stream, status, event)
2     type( c_ptr ) :: event
3     integer(kind=omp_event_handle_kind) :: f_event
4      call C_F_POINTER (event, f_event)
5      call omp_fulfill_event( f_event )
6  end subroutine callback
```

Figure 2.15: Interoperability between non-blocking FFT libraries and OpenMP *tasks* using **DETACH** while ensuring correct asynchronous execution.

considered "completed". This failure to check for true completion of a pre-requisite task leads to an incorrect early release of the dependency, that then allows task C to start running prematurely, leading to incorrect results.

One way of preventing the erroneous scenario above is to use the OpenMP **DETACH** clause. Together with the call to **cudaStreamAddCallback** (lines highlighted in gray), this will ensure that the host thread launches the FFTs and introduces a callback function into the stream in which the FFTs are executing. The host thread then detaches itself from task A to proceed further with other operations, but the OpenMP runtime does not consider the task

"completed". Once the FFT kernels finish executing on the device, the callback function is invoked which "fulfills" the **event** and completes task A. This releases the dependency correctly and task C can begin executing, ensuring the desired asynchronous behavior is achieved. It is also important to note here that, while the FFT kernels in task A were executing, task B can perform data copies (or other tasks) asynchronously.

Figure 2.16 compares pseudo-code segments in CUDA Fortran with OpenMP. Buffers labeled as NEXT, CURR and PREV for different sub-volumes in a slab are allocated on the GPU, and different operations are performed on them asynchronously. In a single loop iteration, a strided host-to-device copy of the $(ip + 1)$-th sub-volume to the NEXT buffer, computations on the CURR buffer (which holds the ip-th sub-volume on the device), a strided device-to-host copy of the PREV buffer to $(ip - 1)$-th sub-volume, and all-to-all from the host on the $(ip - 2)$-th sub-volume are performed asynchronously. In CUDA Fortran, *events* are used to record and synchronize operations on different *streams* to ensure correct results. Computations on the CURR buffer (line 8) do not start before copy of the CURR buffer completes, as enforced by a **cudaStreamWaitEvent** call on line 7. In OpenMP, the **DEPEND** clause with dependency type **IN** ensures the task does not start before prior tasks using the same dependency variable with type **OUT** are completed.

```
do ip=1,np
   NEXT = mod(ip+1,3); CURR = mod(ip,3);
   PREV = mod(ip−1,3); COMM = mod(ip−2,3);
   cudaStreamWaitEvent (trans_stream, DtoH(NEXT), 0)
   cudaMemCpy2DAsync (abuf(NEXT),a(ip+1),trans_stream)
   cudaEventRecord (HtoD(NEXT),trans_stream)
   cudaStreamWaitEvent (comp_stream, HtoD(CURR), 0)
   FFTExecute  (abuf(CURR), comp_stream)
   cudaEventRecord (comp(CURR), comp_stream)

   cudaStreamWaitEvent (trans_stream, comp(PREV), 0)
   cudaMemCpy2DAsync (snd(ip-1), abuf(PREV), trans_stream)
   cudaEventRecord (DtoH(PREV), trans_stream)
   cudaEventSynchronize (DtoH(COMM))
   MPI_IALLTOALL (snd(ip-2))
end do
```

```
do ip=1,np
   NEXT = mod(ip+1,3); CURR = mod(ip,3);
   PREV = mod(ip−1,3); COMM = mod(ip−2,3);
   TASK DEPEND (IN:DtoH(NEXT), OUT:HtoD(NEXT))
   omp_target_memcpy_rect (abuf(NEXT), a(ip+1))

   TASK DEPEND (IN:HtoD(CURR), OUT:comp(CURR))
    DETACH(event)
   FFTExecute  (abuf(CURR), comp_stream)
   hipStreamAddCallback (comp_stream, callback, event, 0)
   TASK DEPEND (IN:comp(PREV), OUT:DtoH(PREV))
   omp_target_memcpy_rect (snd(ip−1),  abuf(PREV))

   TASK DEPEND (IN:DtoH(COMM))
   MPI_IALLTOALL (snd(ip-2))
end do
```

Figure 2.16: Pseudo-code showing batched asynchronous transforms in one direction using CUDA Fortran (on the left) and OpenMP (on the right). The colors red, blue and green highlight events or dependencies corresponding to the NEXT, CURR and PREV buffers or sub-volumes as shown in Figure 2.3 discussed previously.

In Figure 2.16, the OpenMP version differs from CUDA Fortran in three key aspects. The first is the use of **omp_target_memcpy_rect** in place of **cudaMemcpy2D**. The second is a **DETACH** clause with an *event* handle is attached to the **TASK** construct launching the FFTs. This *task* also calls **hipStreamAddCallback**, on line 9, to insert a *callback* function into the *stream* in which the FFTs will execute. The *callback* function is executed once the FFTs complete on the GPU, which calls **omp_fulfill_event** to indicate the completion of the event passed to the **DETACH** clause. This satisfies the **OUT** dependency and allows further tasks to execute. Finally, the use of **DEPEND** clause to enforce the necessary synchronization in place of CUDA *events*.

2.6.1 Plans for OpenMP 5.0

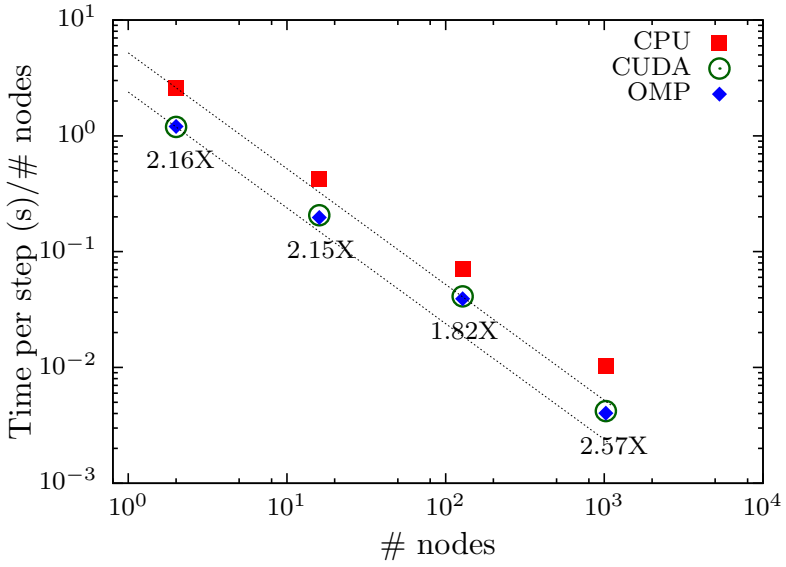

Figure 2.17: Weak scaling performance of non-batched synchronous 3D FFT kernel on Summit. Performance of CUDA Fortran (green) and OpenMP (blue) versions are comparable and overlap each other in the plot. Speedup of OpenMP offload version with respect to the CPU version is given as labels in the plot. Dashed lines indicate perfect weak scaling.

As of June 2021, basic non-batched synchronous versions of the 3D FFT kernels on Summit have been tested, with comparable performance between CUDA Fortran and OpenMP, up to $12{,}228^3$ resolution, as shown in Figure 2.17. The performance observed at large prob-

lem sizes shows a weak scaling of 60%, which is less than desired and is a consequence of dominance by MPI communication. A GPU speedup of 2.57X is observed for the $12{,}228^3$ problem size. Larger speedups are expected in the full DNS code, which has more computation that can benefit from GPU acceleration. However, for purposes of the DNS code, the full promise of OpenMP offloading is still contingent upon solutions to the challenges discussed above, associated with strided copies (even using **omp_target_memcpy_rect**) and asynchronism (with the **DETACH** clause not yet fully supported). Progress in the near future will involve working closely with vendor experts, ideally with improved Fortran support for OpenMP 5.0 and higher.

2.7 Conclusions

In this chapter, a detailed report on design and performance aspects of a new GPU algorithm for direct numerical simulations (DNS) of turbulent flow has been given. This work has in part led to a conference paper at Supercomputing 2019 which was nominated as the best student paper finalist (Ravikumar *et al.*, 2019). The chapter also highlights some key challenges in porting the code to OpenMP to target GPUs and has led to two publications (Bak *et al.*, 2021; Chapman *et al.*, 2021) which are both in press.

The algorithm was optimized for the dense node architecture of Summit, a 200 Petaflops pre-exascale computer which is currently the world's second fastest. The best implementation of this algorithm gives a favorable time to solution for a problem size of 18432^3 grid points, of under 15 seconds of wall clock for each second-order Runge-Kutta time step. This resolution is higher than that reported in previous state-of-the-art simulations, which mostly employed CPU-based massive parallelism. Speedup measured relative to the best-performing CPU code is of order 3 or higher for all problem sizes tested. The significant role of communication implies the FLOP rate, which is expected to be in the range of 1-5% of the peak for 3D FFT based applications (Czechowski *et al.*, 2012), is not a useful metric to characterize the performance of the code. Instead, the bandwidth of the MPI all-to-all

transpose was measured and found to be close to 40% of the peak node injection bandwidth for the largest problem size of 18432^3 on 3072 nodes. A larger fraction of the peak bandwidth may be possible in future system designs with improved hardware and MPI library implementations.

Using the latest GPU data movement techniques allows efficient use of the full node memory, which in turn allowed for solving larger problems and larger MPI message sizes. A close examination of code region runtimes (Figure 2.10) shows that, as a result of powerful GPUs and fast NVLink connections, the cost of FFT computation and data movement between CPU and GPU is reduced to less than one seventh of the code runtime. The bulk of the remaining runtime is spent on network all-to-all communications, which was also studied independently using a standalone code (subsection 2.3.1).

A one-dimensional decomposition combined with a hierarchy of MPI+OpenMP parallelism allows communication in the form of a smaller number of larger messages, which is crucial for achieving acceptable scaling performance, especially at larger problem sizes. The new code features capability to asynchronously overlap compute, GPU-CPU data movement, and MPI communications (Figure 2.4). However, at node counts of 128 and greater, performing MPI asynchronously become more expensive than simply waiting for the entire slab of data to be processed before initiating the MPI all-to-all.

The basic principles of batched asynchronism were applied successfully to the computations of spline coefficients required when tracking fluid particles. Communication costs in computing the cubic spline interpolation was also minimized using the local particle decomposition and "ghost layer" approach (using one-sided MPI). A speedup of around 6X was observed for the problem size of 6144^3 and the weak scaling performance was measured to be close to 50% for the largest problem size of 18432^3. The number of fluid particles being tracked has only minimal effect on the performance of the code. The batched asynchronous algorithm, in general, can be adapted to other numerical computations where multiple operations like data copies, computations and network communication are in-

volved, provided the problem size being solved is large such that it would not fit entirely in the GPU memory. While the algorithm discussed in this chapter focuses mainly on pseudo-spectral simulations of isotropic turbulence, it is also relevant to some simulations of inhomogeneous turbulence – e.g. in DNS of channel flows (Lee *et al.*, 2013) where the spanwise and perhaps streamwise directions are often taken to be periodic.

The research reported in this chapter also presented a brief discussion on some key challenges encountered in developing a portable implementation of extreme scale 3D FFTs using OpenMP to target GPUs. Efficient strided data copies are performed using **Zero-copy** kernels and **omp_target_memcpy_rect**. Although full compiler support for it is not yet available, the OpenMP 5.0 feature **DETACH** is expected to resolve an issue of inter-operability between non-blocking GPU library calls and OpenMP tasks. Comparisons of the batched synchronous code (without **DETACH**) developed using CUDA Fortran and OpenMP revealed no significant performance benefits from using one over the other. From a code development perspective, OpenMP is preferred as it is widely supported while CUDA Fortran and OpenACC are proprietary tools specific to NVIDIA GPUs. While programming GPUs using OpenMP is simpler than using CUDA Fortran, the use of features like **DETACH** increases the complexity of the resulting code. Future work will include testing the **DETACH** approach and using it to develop a batched asynchronous 3D FFT code (and eventually pseudo-spectral simulation of turbulence) capable of problem sizes beyond that recently achieved on Summit.

The lessons learned as well as successes achieved in this work are directly relevant to large computations in many science domains where 3D Fast Fourier Transforms are useful, and in fact generalizable to a variety of large production use codes characterized by substantial needs in communication. This chapter shows that it is possible to efficiently utilize the very large CPU memory while still extracting substantial benefits from the GPU as an accelerator. Continuing research in achieving higher communication performance on leadership computing platforms is still vital in the exascale era and beyond.

CHAPTER 3

ADVANCING UNDERSTANDING OF TURBULENCE THROUGH EXTREME-SCALE COMPUTATION: INTERMITTENCY AND SIMULATIONS AT LARGE PROBLEM SIZES

In turbulence, it is well known that direct numerical simulations (DNS) at massive scales are very useful for advancing physical understanding, but also very demanding in computational resources (Moin & Mahesh, 1998; Ishihara *et al.*, 2009). For a given flow geometry, strong motivation for ever-larger simulations may include reasons of a physical nature, such as a higher Reynolds number (Ishihara *et al.*, 2016), lower diffusivity in turbulent mixing (Clay *et al.*, 2017), increasing chemical complexity in reacting flows (Gruber *et al.*, 2018), higher turbulent Mach numbers in compressible turbulence (Jagannathan & Donzis, 2016); as well as numerical reasons associated with resolution or sampling in space or time (Schumacher *et al.*, 2014; Yeung *et al.*, 2018). Rapid advances in computing power (exponential growth of roughly 1 million-fold increase over the last 25 years) have enabled simulations of order 1 trillion grid points in at least isotropic turbulence (Ishihara *et al.*, 2016), channel flow (Lee & Moser, 2015) and stratified turbulence (Watanabe *et al.*, 2016). With exascale computers expected to arrive by 2022, future prospects for turbulence simulations appear to be bright, at least as far as the number of grid points is concerned.

Recent progress in developing a new algorithm to target GPU based heterogeneous architectures (Ravikumar *et al.*, 2019) have indeed enabled production simulations at unprecedented problem sizes. However, it must be noted that turbulence computations are often so demanding that extreme-scale simulations of the largest size that can fit into the computer memory are likely to be restricted to ever-shorter physical time spans. This is owing to the fact that because of time stepping, the computational resources required scale as N^4 while the memory is proportional to N^3. That is, every halving of Δx (grid spac-

ing) is accompanied by at least a 16 times increase in cost–which exceeds the performance increase enabled by most newly installed top-ranked machines over their predecessors. On the other hand, recent results (Yeung *et al.*, 2018) showed that if such highest resolution simulations are performed as a means to improve resolution in the small scale motions, then such simulations need not be long. However, it is important to ensure good sampling, which may be possible through ensemble averaging over a number of short simulation "segments" with good statistical independence, each of these being obtained by grid refinement from lower resolutions. This approach is referred to as Multiple Resolution Independent Simulations (MRIS).

The intent in this chapter is to communicate recent innovations in turbulence simulations that are important to the enduring goal of advancing understanding of turbulence through taking proper advantage of future exascale computing or beyond. The proposed MRIS technique is not all powerful: for example, in this work, a longer adjustment time for the numerical solution will be required when attempting to increase the Reynolds number at a given resolution, compared to that needed to increase the resolution at a given Reynolds number. A longer adjustment time is likely required also for simulations of wall-bounded turbulence, such as fully-developed channel flow spanning several flow-through times (Lee *et al.*, 2014). This is in addition to the obvious inapplicability of this approach for simulations with no stationary state. However, this approach is well suited to the task of obtaining well-sampled results of the small scale physics, at higher resolution in stationary isotropic turbulence. More details on the MRIS technique and a validation study are discussed in section 3.1. Results on 3D local averages from simulations at different R_λ ranging from 390 to 1300 are presented in section 3.2. Conclusions are summarized in section 3.3.

3.1 MRIS: Methodology and Validation

The subsections below begin with a more detailed discussion of the MRIS methodology and how this approach can be tested, for forced stationary isotropic turbulence. A val-

idation study is presented which addresses single-point statistics, issues of statistical independence, and two-point statistics crucial to the material of section 3.2. The forcing scheme used is designed to reduce the statistical variability of spatially-averaged statistics in time, by freezing the energy spectrum in the lowest few wavenumber shells (Donzis & Yeung, 2010), at values derived from long-time averages of results from stochastic forcing (Eswaran & Pope, 1988). However, the MRIS methodology should be compatible with other forcing schemes that share the common principle of maintaining energy by forcing at the large scales, as well.

3.1.1 The MRIS approach and a validation procedure

As noted previously, with expectations for DNS rising due to a combination of scientific need and advances in computing power, a pressing challenge is that full-length simulations spanning several large-eddy time scales at extreme-scale resolution pushing the envelope of the latest leadership-class platforms will likely be inaccessible. However, if a turbulent flow is statistically stationary and the focus is on small-scale phenomena with short time scales, a much more viable alternative is available.

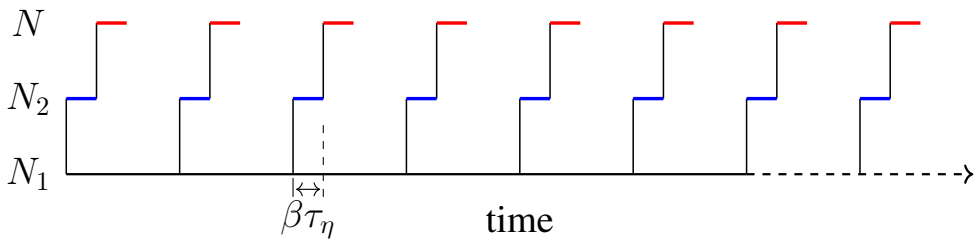

Figure 3.1: Schematic of the Multiple Resolution Independent Simulation approach, where simulations at higher resolutions are performed in steps starting from snapshots in a long-running simulation at lower resolution spaced out in time (as shown by the solid horizontal line). The dashed horizontal line represents an extension of the lower resolution simulation to generate more snapshots as required. Blue and Red lines represent short simulations, of length $\beta\tau_\eta$ in time at intermediate (N_2^3) and final (N^3) spatial resolution, respectively.

In the proposed (MRIS) approach, sampling over a long, continuous simulation at high resolution, say N^3 is replaced by sampling over a number of short simulation segments

that possess a demonstrable degree of statistical independence. Although a full-length simulation at N^3 resolution may be excessively costly, a long simulation at some lower resolution (say N_1^3) can be assumed to be available, either from prior work or through new calculations. The strategy here is to use multiple (say M) snapshots, with approximate independence through a separation in time from an N_1^3 simulation, as initial conditions for the short N^3 segments, which are then disjoint from each other. Since the small scales adjust to grid refinement ($N_1^3 \rightarrow N^3$) rapidly, these segments need not be long, only a few (say β) Kolmogorov time scales in length. Figure 3.1 illustrates the approach described here. The total cost of simulations at N^3 will then be measured in Kolmogorov time scales (τ_η) instead of large-eddy time scales (T_E). This results in major cost savings, especially when T_E/τ_η is high, which is the case at high Reynolds number. The savings in turn make well-sampled results at high resolution much more readily feasible than otherwise.

It may be noted that while ensemble averaging over multiple independent simulations in turbulence is not common, it has been used before, in situations where statistical variability per simulation can be very significant (Overholt & Pope, 1996). A critical test for MRIS is whether the results are close, within some margins of uncertainty, to those from an actual, full-length N^3 simulation. For validation, a full-length DNS that is available at some affordable value of N is considered, at a Reynolds number sufficiently high to show clear intermittency, and is very well resolved in space and time — essentially, usable as a high-accuracy benchmark that MRIS results can be compared to. Resolution in space can be expressed by the non-dimensional parameter $k_{max}\eta$, where $k_{max} = \sqrt{2}N/3$ is the highest wavenumber resolvable on an N^3 grid of length 2π units on each side, and $\Delta x/\eta \approx 2.96/k_{max}\eta$. Accuracy in time may be controlled through the Courant number, which in the present flow without a mean velocity is defined as

$$C = \Delta t \left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} \right]_{max} . \tag{3.1}$$

where u, v, w are velocity fluctuations, and the maximum is taken over all (N^3) grid points. In the second-order Runge Kutta scheme used here a combination of $k_{max}\eta \approx 1.4$ and $C = 0.6$ is usually adequate for low-order statistics, but better resolution in both time and space are important for higher-order quantities strongly impacted by intermittency.

A well-resolved instantaneous snapshot at N^3 resolution can be truncated down to N_1^3 by removing content at all Fourier modes with wavenumber higher than the value of k_{max} that corresponds to N_1^3 resolution. This removal of high-wavenumber modes leads to an immediate decrease in various quantities, including $\langle\epsilon\rangle$, that contain substantial high-wavenumber content. Next, an N^3 simulation segment is run with this truncated field as initial conditions, by filling in the "extra" Fourier coefficients beyond the value of k_{max} of an N_1^3 grid with zeroes. The desired outcome is for $\langle\epsilon\rangle$ to recover quickly to its original value in the (reference) N^3 simulation. For a given N, this "recovery time" is expected to increase with N/N_1, being longer (thus less economical) if N_1 is a very low resolution. In cases where prior data at resolutions $N/3$ or $N/4$ are conveniently available, it would be useful to reach the desired resolution via an intermediate stage such as $N_1 \rightarrow N_2$ followed by $N_2 \rightarrow N$. Incidentally, the "recovery" process examined here has some parallels with the process by which the large scales can regenerate the small scales if the latter are artificially removed (Yoshida *et al.*, 2005), provided the large scales are themselves maintained.

Since resolution effects are being investigated, comparisons should be based mainly on quantities that are sensitive to the small scales. The list we consider includes the mean dissipation rate ($\langle\epsilon\rangle$), the dissipation skewness (Kerr, 1985) (S_ϵ), the energy spectrum at high wavenumber; as well as direct indicators of intermittency such as the statistics of dissipation rate and enstrophy fluctuations evaluated at a point or averaged locally in space.

Table 3.1 shows important parameters for tests conducted in the MRIS validation study, with reference to a simulation at $R_\lambda \approx 390$ (one of the values tested in (Yeung *et al.*, 2018)), with $N = 3072$ at $k_{max}\eta \approx 4.2$ which provides good resolution for the small scales. This full-length "reference" simulation was run for 5.5 T_E, with 22 snapshots written at intervals

Table 3.1: Selected parameters in simulation segments used for MRIS validation: from N_1^3 to N^3 (via N_2^3 if applicable), $k_{max}\eta$ on N^3 grid, number of segments (M), time span in units of τ_η, and ensemble-averaged $\langle\epsilon\rangle$ and S_ϵ, at the beginning and end of each segment (subscripts b and e respectively). Initial conditions for Cases 3 and 6 were taken from the end of Cases 2 and 5. In the reference 3072^3 simulation, the time-averaged values of $\langle\epsilon\rangle$ and S_ϵ were 1.409 and 0.588 respectively. All simulations in this table were performed using a Courant number of $C = 0.25$, with the same forcing parameters and viscosity.

Case	N_1	N_2	N	$k_{max}\eta$	M	β	$\langle\epsilon\rangle_b$	$S_{\epsilon b}$	$\langle\epsilon\rangle_e$	$S_{\epsilon e}$
1	768	-	3072	4.2	11	4	1.374	0.471	1.410	0.588
2	768	-	1536	2.1	22	2	1.375	0.471	1.409	0.585
3	-	1536	3072	4.2	22	2	1.409	0.585	1.410	0.587
4	384	-	1536	2.1	22	4	1.135	0.264	1.397	0.577
5	384	-	768	1.05	22	4	1.135	0.264	1.403	0.529
6	-	768	1536	2.1	22	2	1.403	0.529	1.404	0.586

of $0.25\ T_E$ apart. Each snapshot is truncated down to 768^3, and studied to understand how the numerical solutions recover when a 4X increase in resolution back to 3072^3 is directly applied (Case 1), or through two successive 2X increases in resolution (Cases 2-3, combined). It is important to check whether a new stationary state forms in a short period of time, with statistics closely resembling those extracted from the 3072^3 reference simulation. Similar tests (Cases 4 and 5-6) are conducted to see if acceptable results can be obtained from poorly-resolved velocity fields (in this case, 384^3 with $k_{max}\eta$ as low as 0.5) similarly. Although, through the definition of η, changes in $\langle\epsilon\rangle$ lead to changes in η and hence $k_{max}\eta$ since $\eta \propto \langle\epsilon\rangle^{-1/4}$, this effect is weak even if long-time variations in the order of 10% (Donzis & Yeung, 2010) are considered. All the simulations listed in this table were performed using CPU-based codes on the machine Frontera at the Texas Advanced Computing Center, which as of 2021 is the most powerful academic-based supercomputer in the world.

3.1.2 Single-point statistics and spectra

Both Table 3.1 and Figure 3.2 provide information on the dissipation rate and dissipation skewness (Kerr, 1985), which can both be written explicitly in terms of the dissipation spec-

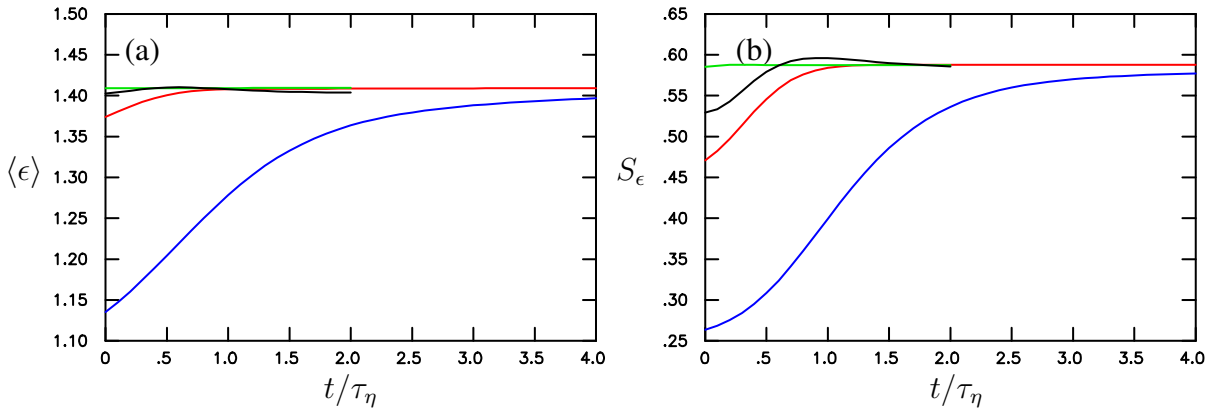

Figure 3.2: Evolution of (a) $\langle\epsilon\rangle$ and (b) S_ϵ ensemble-averaged over multiple simulation segments, for Cases 1 (red), 3 (green), 4 (blue), and 6 (black), of different lengths as noted in Table 3.1.

trum. When a substantial collection of high wavenumber modes is abruptly removed, the dissipation rate drops, while subsequent transfer of energy from the large scales (which are forced) will allow a recovery. Since forcing is applied at the large scales, its details are not expected to affect the small-scale dynamics significantly (Sreenivasan, 1998). The contrast between Cases 1 and 4 shows, as expected, that truncation down to a lower wavenumber cutoff leads to a stronger reduction of dissipation rate and a slower subsequent recovery. The route of two successive refinements (2X each) requires fewer time steps to be run on the targeted finer (N^3) grid than a direct 4X refinement — which translates to lower resource requirements overall. Similar but stronger trends are observed for the dissipation skewness, which contains more high wavenumber content than the mean dissipation rate.

In wavenumber space, an immediate consequence of grid refinement is that energy can now be transferred to higher wavenumbers that were not represented before. Figure 3.3 confirms that the small scales do adjust rapidly, with the energy spectrum ($E(k)$) at the end of the short simulation segments being nearly indistinguishable from results in the reference simulation. For Case 3, although the spectrum initially has a mild pileup at the k_{max} of the intermediate-sized (1536^3) grid (resulting from Case 2), a well-behaved functional form soon emerges.

Two important measures of intermittency and small scales are the energy dissipation

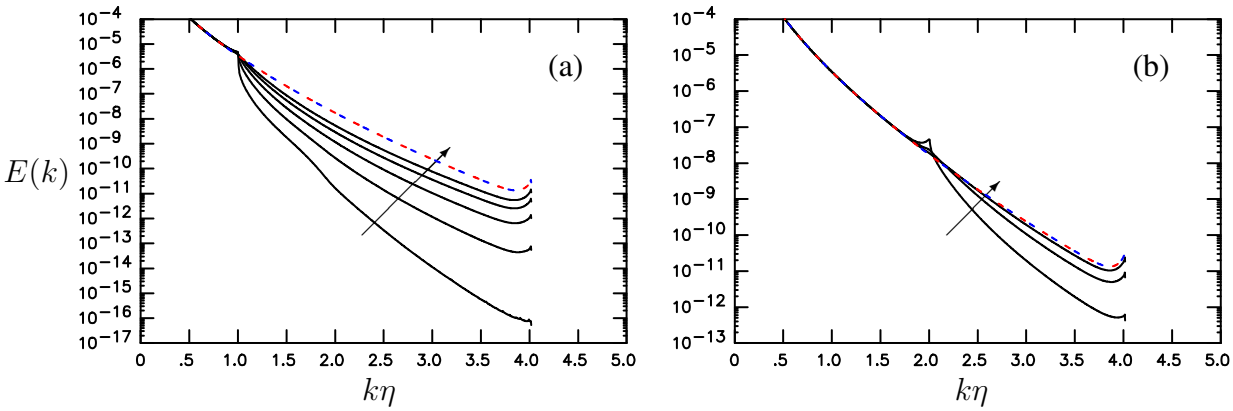

Figure 3.3: Development of the energy spectrum as a result of grid refinement for (a) Case 1 and (b) Case 3 (per Table 3.1). For clarity, only early-time data in the short segments (at increments of 0.1 τ_η, following the arrows) are shown. A blended red and blue dashed line gives spectra at the end of the short segments and time-averaged within the 3072^3 reference simulation.

rate (ϵ) and enstrophy (Ω), which are defined by,

$$\epsilon = 2\nu s_{ij} s_{ij} \qquad ; \qquad \Omega = \omega_i \omega_i \tag{3.2}$$

where ν is the kinematic viscosity and s_{ij} and ω_i are components of strain and vorticity rate respectively. Interest in the behavior of fluctuations of these quantities is a primary motivator for resolving the small scales as well as possible. Figure 3.4 shows information on the time history of (a) peak values (over all grid points) and (b) the probability density function (PDF) of normalized dissipation and enstrophy, obtained from the simulation segments of Case 3. In frame (a), despite substantial variability, the peak values can be seen to adjust to a new, stable stationary state, after only about 0.5 τ_η. The observed peak values in this new stationary state agree well with time-averaged values in the reference 3072^3 simulation (black dashed lines, partly hidden). Higher values of peak $\Omega/\langle\Omega\rangle$ also indicates enstrophy is more intermittent (Yeung *et al.*, 2018). The dissipation PDF data at different times in frame (b) are also in support of a rapid approach to a new stationary state, consistent with the reference simulation.

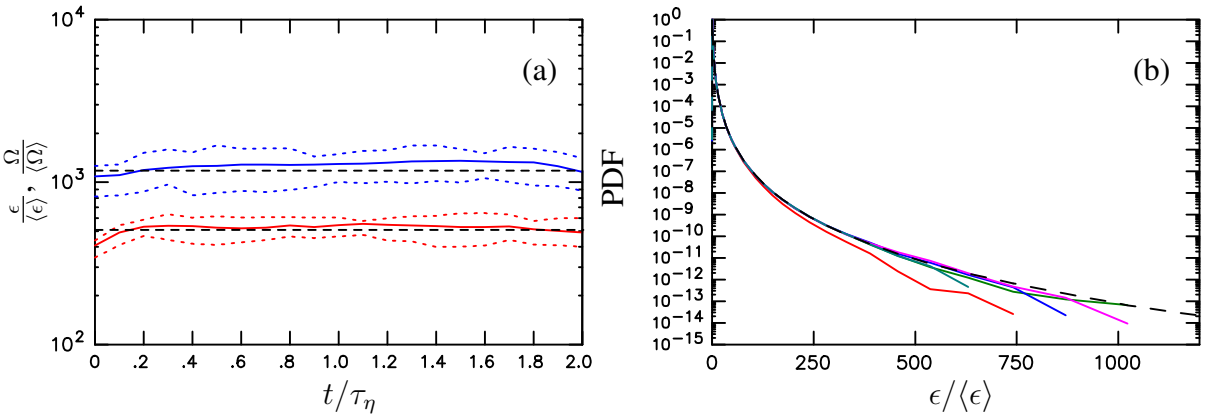

Figure 3.4: Statistics of normalized dissipation and enstrophy obtained from multiple simulation segments for case 3 in Table 3.1. (a) Peak values: ensemble-averaged (solid lines) and 25th and 75th percentiles (dashed lines), red for dissipation, blue for enstrophy. (b) PDFs: red for data at $t = 0$, green, blue, magenta, cyan for $t/\tau_\eta = 0.5$, 1.0, 1.5, 2.0 respectively. In both frames, black dashed lines (partly hidden) give results from the reference 3072^3 simulation for comparison.

3.1.3 Tests of statistical independence

The statistical quality of results from MRIS depends on the number of segments (M) available for ensemble averaging, and their degree of statistical independence. The latter is expected to be a function of scale size, and closely related to the time separation (τ_0) between lower-resolution snapshots used as initial conditions for the MRIS segments, with the overall sampling period being effectively $\mathcal{T} = M\tau_0$. Statistical errors in DNS results can often be quantified via confidence intervals computed after the fact. However, it would be useful to develop some *a priori* estimates for the minimum τ_0 desired, depending on the nature of the quantity being sampled, and in relation to the time scales τ_η and T_E. This issue is explored below using both one- and two-time statistics.

With stationary turbulence in mind, a basic question for one-time statistics, such as the volume-averaged energy dissipation rate ($\langle\epsilon\rangle$), is whether significant and random departures from the mean of either sign are consistently observed within a time period $\mathcal{T}$. If a signal shows persistent behaviors (such as monotonic variations) then the sampling period is too short. Conversely, a predominance of rapid oscillations would suggest a small τ_0 is sufficient, although strict independence over an interval of τ_0 is not necessary.

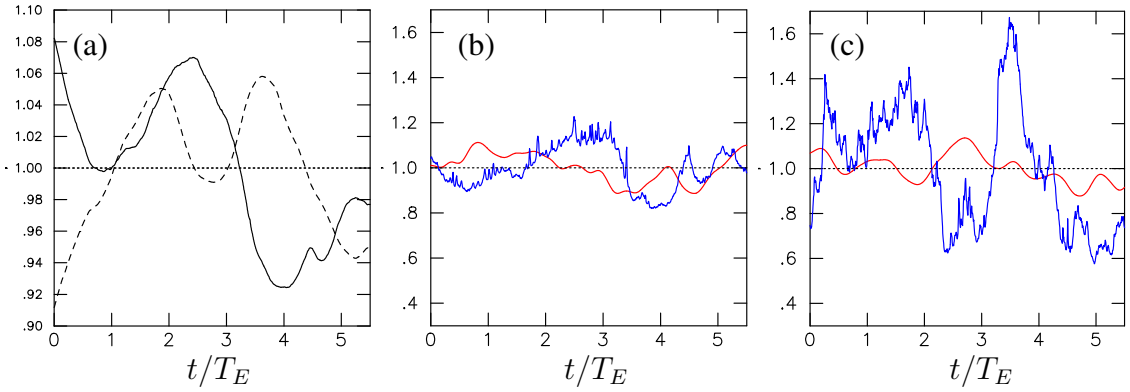

Figure 3.5: (a) $\langle\epsilon\rangle/\langle\epsilon\rangle_T$ over a period of 5.5 T_E in R_λ 390 simulations at $k_{max}\eta \approx 1.4$ (solid curve) and 4.2 (dashed curve) respectively, where the notation $\langle..\rangle_T$ denotes a time average of volume-averaged quantities. (b) Energy spectrum $E(k)$ normalized by a time average, for $k = 6$ (red) and $k = 0.95\ k_{max}$ (blue), from the $k_{max}\eta \approx 1.4$ simulation. (c) shows similar data, with $k_{max}\eta$ 4.2.

Data from two long simulations at $R_\lambda \sim 390$, of different resolutions, are shown in Figure 3.5. The first is the one used to initiate high-resolution MRIS segments, whereas the second is the high-resolution reference simulation in the validation study. In frame (a) it can be seen that, in both simulations, dissipation varies to about the same degree (of order 10% or less), and with similar time scales. This behavior is not a surprise, since the mean dissipation rate is determined by the large scales, and the same forcing is used in both datasets. However, quantities at disparate scale sizes should behave differently. Frames (b) and (c) show, that the energy spectrum $E(k)$ is indeed very dependent on wavenumber. At low wavenumber, the red lines in both (b) and (c) show slow and modest variations. In contrast, at high k (near k_{max} at each resolution) the lines in blue resemble rapid oscillations superimposed on a smooth background signal, which itself varies more strongly at high resolution. Incidentally, since $E(k)$ is (at high k) the sum of energies held in a large number of Fourier modes in a spectral shell, it can be inferred that individual Fourier modes vary in time even more rapidly than for the $E(k)$ values shown.

While one-time statistics show directly how different quantities evolve in time, it is tempting to ask if independence between two single-time snapshots can be assessed, at times t_1 and $t_2 = t_1 + \tau$, by computing some statistical correlations. For example, one may

consider the two-time correlator $\sigma(\tau) = \langle \epsilon(\mathbf{x}, t_1)\epsilon(\mathbf{x}, t_2)\rangle / \langle \epsilon^2 \rangle$ which is analogous to the two-point correlator in space related to intermittency exponents (Sreenivasan, 1993). Another possible scale-dependent measure of statistical coupling may be the coherency spectrum, defined by $\rho(k, \tau) = E_c(k, \tau)/\sqrt{E(k, t_1)E(k, t_2)}$ where $E_c(k, \tau)$ is the co-spectrum between $\hat{\mathbf{u}}(\mathbf{k}, t_1)$ and $\hat{\mathbf{u}}(\mathbf{k}, t_2)$ in wavenumber space. However, both of these quantities are subject to contamination by the "random-sweeping" effect (Tennekes, 1975), in which small-scale structures may be simply moved to another location as a result of advective transport by the large scales. Such an effect will cause an artificial drop of $\sigma(\tau)$ even if the turbulence were frozen in time. Likewise, since a coherency spectrum basically measures the phase coupling between Fourier-transformed quantities in wavenumber space (Yeung, 1996), random sweeping can also cause an artificial decrease of the coherency spectrum, especially at high Reynolds numbers.

Since random sweeping is an artifact of a fixed observer seeing differences in time while small-scale structures are swept along by the fluid, an alternative approach free of this effect is thus to consider the flow conditions experienced by an observer moving with the flow, i.e. to use a Lagrangian framework (Yeung, 2002). For a general flow variable q, we can define the Lagrangian two-time correlator as,

$$\sigma_L(q; \tau) = \langle q^+(t) q^+(t + \tau)\rangle / \langle q^2 \rangle \tag{3.3}$$

where superscripts $+$ denote Lagrangian quantities evaluated along the trajectories of fluid particles moving with the local fluid velocity. Clearly, $\sigma_L(q; \tau)$ is unity at $\tau = 0$ but approaches the ratio $\langle q\rangle^2/\langle q^2\rangle < 1$ when τ is large enough for $q^+(t)$ and $q^+(t + \tau)$ to be statistically independent. Subtracting $\langle q\rangle^2$ from both the numerator and denominator of $\sigma_L(q; \tau)$ gives the correlation coefficient $\rho_L(q; \tau)$, which approaches 0 at large τ for any q.

Figure 3.6 shows sample results in the two-time correlators (in (a)) and correlations (in (b)), for three choices of the quantity q being (i) u^2 (square of one fluctuation), (ii) ϵ or (iii)

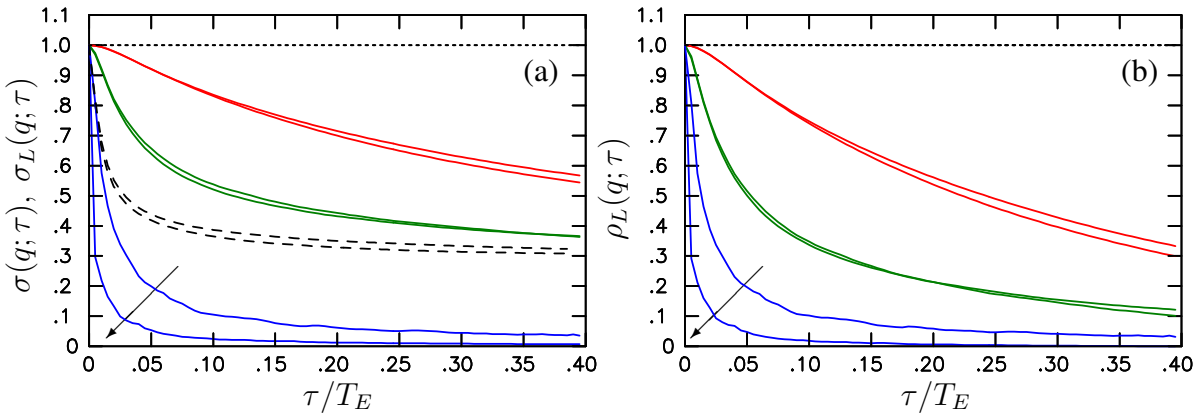

Figure 3.6: (a) Eulerian (dashed lines) and Lagrangian (solid lines) two-time correlators versus time lag in units of T_E, for $q = u^2$ (red), ϵ (green) and ϵ^2 (blue). (b) Lagrangian two-time correlations, with same color coding for each variable. Both (a) and (b) show data obtained at two resolutions, $k_{max}\eta \approx 1.4$ and 4.2, at R_λ 390. The only sensitivity evident is for ϵ^2 (resolution increasing in the direction of the arrows).

its square, whose behavior mimics extreme events of very high amplitude. In this ordering, as the dominant scales are shifted to quantities associated with increasingly smaller scales, it is not surprising that both measures of dependence or correlation decrease with time lag more rapidly. The discrepancy between the green lines for $\sigma_L(\epsilon;\tau)$ and its Eulerian counterpart $\sigma(\epsilon;\tau)$ confirms the importance of random sweeping, whose effect is strongest at small τ. The contrast between Eulerian and Lagrangian data here is also consistent with past comparisons between the statistics of Eulerian and Lagrangian time derivatives (Yeung & Pope, 1989; Tsinober *et al.*, 2001). However, at $\tau = 0.4\ T_E$ this discrepancy is mild, which is also expected, since the large-scale motions responsible for the sweeping are well-sustained only for a finite time interval. For the velocity, at $\tau/T_E = 0.4$, $\sigma(u^2;\tau)$ is not close to the asymptotic value of 1/3 (for velocity fluctuations, which are Gaussian distributed). A clearer view of the degree of independence that remains at this time lag is given by the Lagrangian correlation functions in frame (b), where a value of 0.1 for $\rho_L(\epsilon;\tau)$ suggests a high degree of independence in practice. Both frames show that ϵ^2 has short time scales, which become shorter yet as resolution is increased, consistent with the emergence of stronger extreme events of short lifetimes. Finally, although data at only one Reynolds number is given in this figure, since the Lagrangian integral timescale of the dissipation rate

decreases with respect to large-eddy time scales as the Reynolds number increases (Yeung *et al.*, 2006b), it seems likely that $\tau_0/T_E \sim 0.4$ as a criterion for statistical independence of small-scale quantities will hold better yet at higher Reynolds numbers.

The assessment of resolution effects in Figure 3.6 as discussed above suggests two high-resolution snapshots obtained by grid refinement from two modestly-resolved ones will retain the degree of independence that originally existed in the former. Although this statement is less valid for high amplitude events which are likely to be under-represented if the resolution is low, results are in support of the hypothesis that good sampling at high resolution can be derived from good sampling at modest resolution, in the MRIS approach that is proposed.

3.1.4 Moments of 3D local averages

Next, an examination of MRIS results for multi-point statistics in physical space is presented. Specifically, the scaling of moments of the 3D local averages of dissipation rate and enstrophy, over scale sizes r, ranging from the smallest (one grid spacing, Δx) to the largest (half of the length, L_0, of the periodic domain).

$$\epsilon_r(\mathbf{x}) = \frac{1}{Vol}\int_r \epsilon(\mathbf{x}+\mathbf{r})d\mathbf{r} \qquad ; \qquad \Omega_r(\mathbf{x}) = \frac{1}{Vol}\int_r \Omega(\mathbf{x}+\mathbf{r})d\mathbf{r} \tag{3.4}$$

Because the DNS is performed using Cartesian coordinates, 3D averaging over sub-cubes (instead of spheres) is convenient. In the limit of $r \to 0$ the p^{th} order moment of $\epsilon_r/\langle\epsilon\rangle$ approaches $\langle\epsilon^p\rangle/\langle\epsilon\rangle^p$, which implies (for $p > 1$) small-scale resolution is crucial. In the other limit of $r \to \infty$ all moments approach unity, regardless of order, with homogeneity in space being the only requirement. However, the most important range of r is in the inertial range $\eta \ll r \ll L_1$, where the longitudinal integral length scale L_1 is about 0.2 L_0 in simulations considered here. In this range, classical refined similarity theory suggests

$$\langle\epsilon_r^p\rangle/\langle\epsilon\rangle^p \propto (r/\eta)^{-\zeta_p} \tag{3.5}$$

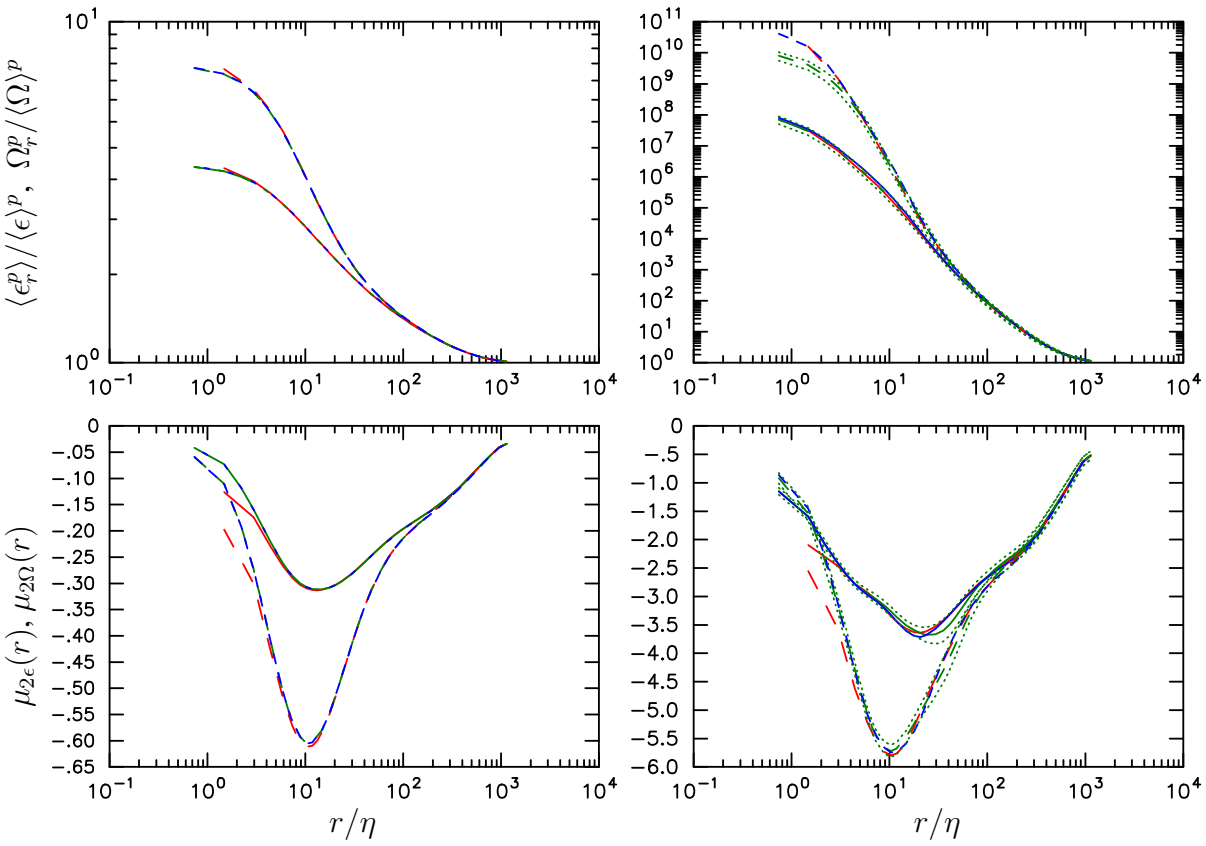

Figure 3.7: Moments (top row) and logarithmic local slopes (bottom row) for 3D local averages of $\epsilon_r/\langle\epsilon\rangle$ (solid lines) and $\Omega_r/\langle\Omega\rangle$ (dashed lines), ensemble-averaged over multiple simulation segments from Cases 2 (red) and 3 (green). Lines in blue are from the reference 3072^3 simulation. Second moments on the left, sixth moments on the right. Dotted lines in green (very close to solid and dashed lines of the same color) show $\pm 95\%$ confidence interval for the sixth order moments and local slopes.

where the dependence of the scaling exponents ζ_p (all positive) on the order p is of fundamental interest. Unfortunately since 3D averaging is challenging in both experiments and computation, many studies in the literature have, until recently (Iyer *et al.*, 2015), used instead 1D averages along a line, and/or a one-dimensional surrogate ($(\partial u/\partial x)^2$, motivated by Taylor's frozen turbulence hypothesis) of ϵ which is more intermittent than ϵ itself. Furthermore, accurate inferences of ζ_p require having a well-defined scaling range (hence a high Reynolds number), and attention to possible contamination from limitations in both resolution and sampling.

In the MRIS validation effort here, the focus is on resolution and sampling. Figure 3.7

shows results averaged over multiple MRIS segments, for orders 2 to 6 (the latter being more demanding). Scaling exponents are estimated through logarithmic local slopes: i.e. $d\ln\langle\epsilon_r^p\rangle/d\ln r$, which would be equal to $-\zeta_p$ if a well-defined plateau exists. The notations $\mu_{p\epsilon}(r)$ and $\mu_{p\Omega}(r)$ denote the local slopes for (the moments of) ϵ_r and Ω_r respectively. Since the Reynolds number in the MRIS validation study is not high, it is not surprising that local slopes in this figure do not show a clear scaling range. Instead, there is a hint of an inflection point developing in the neighborhood of $r/\eta \sim O(100)$. Stronger intermittency in Ω_r versus ϵ_r is manifested clearly in higher values of the moments at small r, an effect that is noticeable up to $r/\eta \approx 200$. Values of the moments at small r increase very strongly with p, indicating that the resolution needed to observe flat plateaus as $r \to 0$ becomes harder to achieve. For the data shown, comparison between red and green lines suggests the effects of resolution are largely confined to $r/\eta \leq O(5)$, with very little apparent effect at intermediate scales close to the inflexion point noted above. This suggests $k_{max}\eta \approx 2$ (as for the red lines) may be sufficient for investigating some aspects of inertial-range intermittency, although sufficient sampling is still necessary. With good sampling, very good agreement is seen between lines in green and blue: i.e. results on the local averages from the full-length 3072^3 reference simulation can be well recovered from much less-expensive data derived from MRIS (Case 3). It can also be seen that the data from the reference simulation, in blue, falls within the $\pm 95\%$ confidence intervals for the sixth order moments (and local slopes) of both dissipation and enstrophy. This shows good sampling from the MRIS approach is achieved, and very good agreement with the reference simulation is observed within sampling uncertainties. The difference in sixth order moments of locally averaged enstrophy at small r/η is likely due to the removal of Fourier modes contaminated by aliasing errors, as the velocity field from the reference simulation was initially truncated.

3.2 MRIS: Study of intermittency at high resolution

A major motivation in this chapter has been a desire to contribute towards a high-fidelity characterization of both dissipation range and inertial range intermittency in high Reynolds number turbulence. This pursuit is very resource intensive, and large simulations that resolve the small scales well are necessary. The works described in subsection 2.2.4 and section 3.1 were in fact undertaken in order to identify a viable path towards meeting these challenges.

Table 3.2: Parameters for production simulations at different Reynolds numbers, using the MRIS approach. All simulations in this table were performed using a Courant number of $C = 0.3$.

R_λ	N	$k_{max}\eta$	β	M	$\langle\epsilon^2\rangle/\langle\epsilon\rangle^2$	$\langle\Omega^2\rangle/\langle\Omega\rangle^2$
390	1024	1.4	2	22	3.869	7.665
390	1536	2.1	2	22	4.034	7.938
390	3072	4.2	2	22	4.074	7.969
650	2048	1.4	2	15	4.357	8.718
650	3072	2.1	2	15	4.575	9.133
650	6144	4.2	2	15	4.664	9.214
1000	4096	1.4	2	10	4.949	9.901
1000	6144	2.1	2	10	5.250	10.556
1000	12288	4.2	2	10	5.381	10.745
1300	12288	3.0	1	10	6.103	12.238
1300	18432	4.5	1	10	6.142	12.288

The resolution levels and selected parameters of production simulations that have been performed using GPUs on Summit, combined with the MRIS approach starting from modest resolutions at four targeted Reynolds numbers, are shown in Table 3.2. Results at R_λ 390 here are equivalent to those reported in the MRIS validation study of section 3.1. As resource requirements increase, the number of short simulation segments employed is fewer. Following estimates obtained in section 3.1, each segment is $2\ \tau_\eta$ long, except for those at highest Reynolds number on a 18432^3 grid. In the latter case, shorter segments of duration $1\ \tau_\eta$ are acceptable, partly because approach to a new stationary state in the manner of Figure 3.4a took only about $1\ \tau_\eta$, and partly because better overall sampling is likely from

taking averages over more segments of shorter duration than over fewer segments of longer duration. Normalized second-order single-point moments in Table 3.2 are seen to increase systematically with both Reynolds number and resolution, while being higher for enstrophy than the dissipation. Sensitivity to resolution from $k_{max}\eta \approx 2$ on-wards appears to be relatively weak, thus suggesting, at least at $k_{max}\eta \approx 4$, a certain degree of convergence has been reached.

3.2.1 Single point statistics

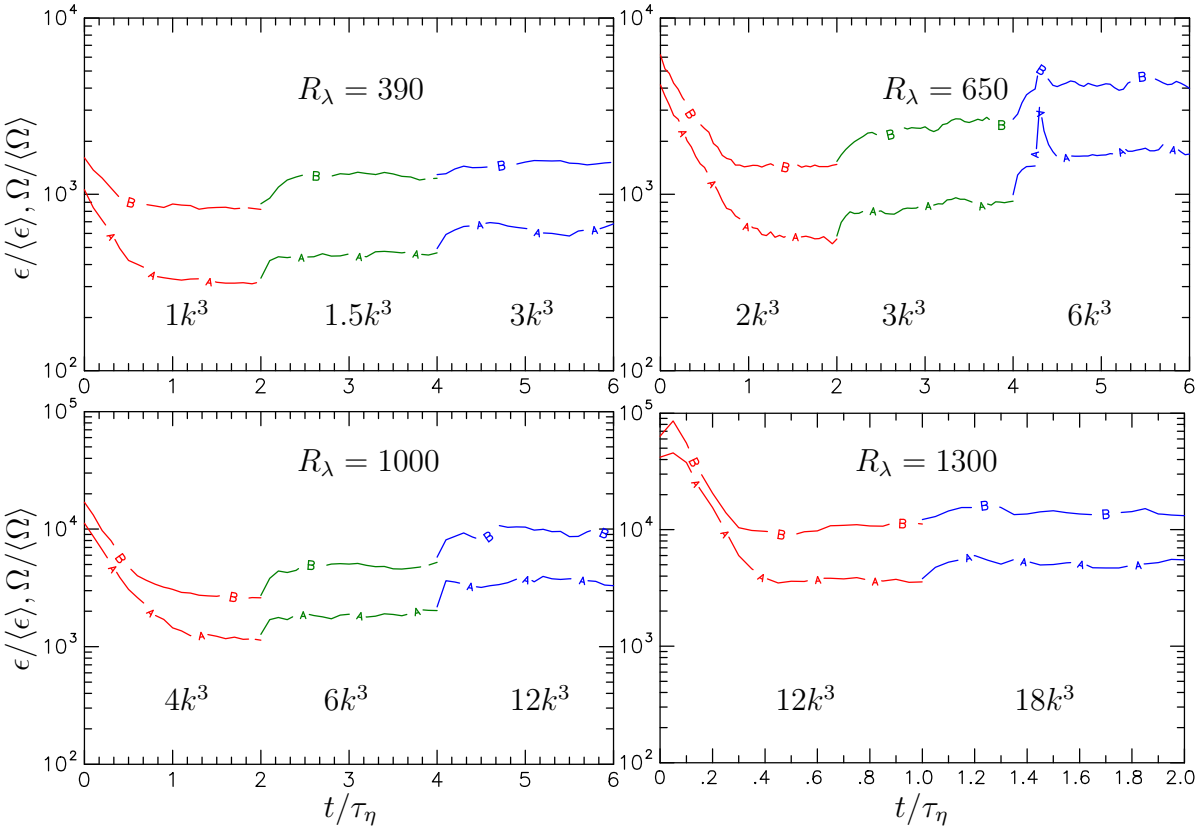

Figure 3.8: Ensemble averaged peak energy dissipation (A) and enstrophy (B) at different R_λ. The different colors in each frame correspond to different spatial resolutions, as indicated by the labels in each frame.

Two very useful diagnostics of extreme events in high Reynolds number turbulence are the peak values of dissipation and enstrophy over all grid points at a given time. Figure 3.8 shows the evolution of normalized peak dissipation ($\epsilon/\langle\epsilon\rangle$) and enstrophy ($\Omega/\langle\Omega\rangle$) with

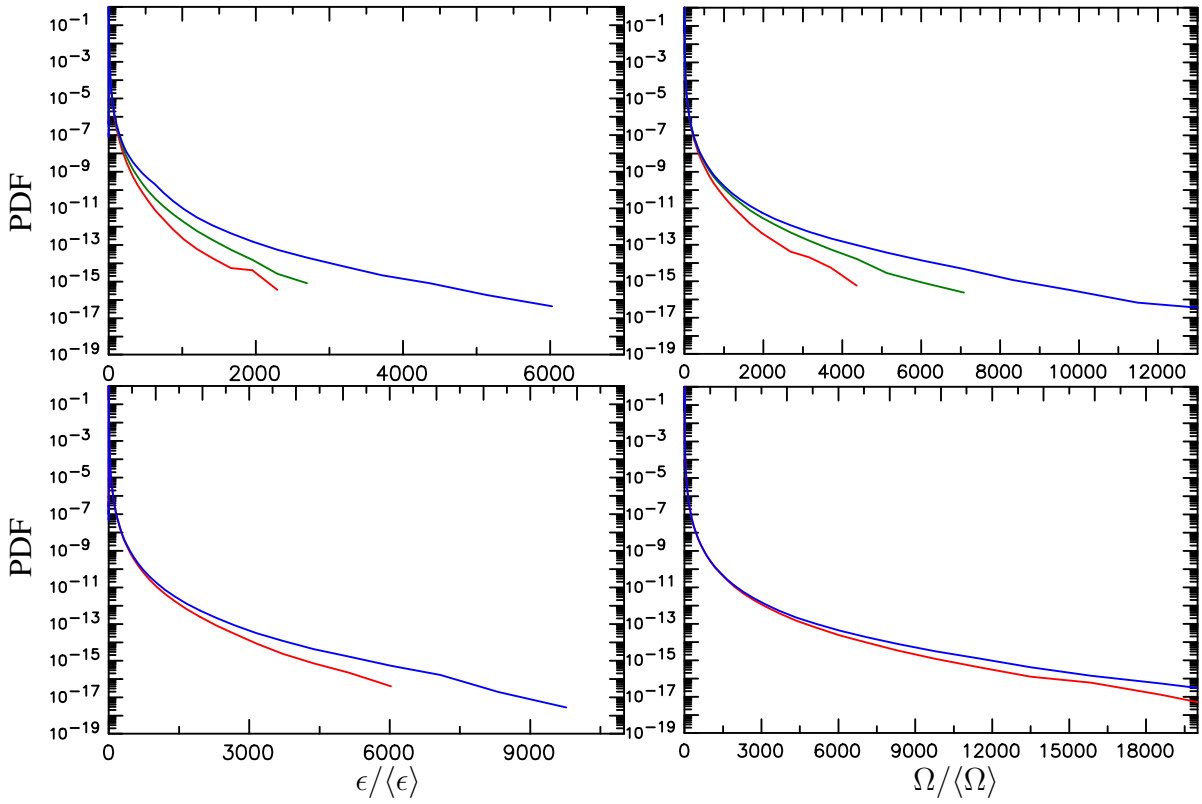

Figure 3.9: Ensemble averaged PDFs of dissipation (left) and enstrophy (right) over multiple realizations with average over time during the second half of simulations in each segment at R_λ 1000 (top) and 1300 (bottom). The different colors correspond to $k_{max}\eta$ of 1.4 (red), 2.1 (green) and 4.2 (blue) in the top row and $k_{max}\eta$ of 3.0 (red) and 4.5 (blue) in the bottom row.

time, averaged over M realizations, for each segment in the MRIS run. A substantial drop in the peak values during the first τ_η for R_λ 390, 650 and 1000 is observed. The initial drop is steeper yet for $R_\lambda \sim 1300$, because a simultaneous improvement of both spatial and temporal resolution in this case leads to stronger suppression of aliasing errors than at the other three lower Reynolds numbers. A jump in the peak values as each segment with finer grid spacing begins has also been observed. This is expected, since after reducing the alias errors, subsequent improvements in spatial resolution will enable the simulations to capture the larger magnitude velocity gradients more accurately. Spurious spikes (such as that seen at $t/\tau_\eta = 4.3$ in data at $R_\lambda \sim 650$) indicate numerical issues arising from alias errors (Yeung *et al.*, 2018) are still not completely absent. The spike in peak dissipation rate is larger than that in peak enstrophy, because (due to a constraint from incompressibility)

dissipation rates are more sensitive to numerical inaccuracies (Yeung *et al.*, 2018). The source of the spike was identified to one particular simulation among the 15 at $R_\lambda \sim 650$. The spike vanishes if the Courant number is further reduced to 0.15. Generally, differences between results at $C = 0.3$ and 0.15 are mild, except for such rare cases. However, relatively stable statistics of the peak values can be obtained using data from the second half of the simulation segments, as in this study.

Figure 3.8 also shows that the peak dissipation and enstrophy are 3 to 4 orders of magnitude larger than its mean value, with the peak enstrophy being 2 to 3 times larger than the peak dissipation rate. As R_λ increases, peak values of both dissipation and enstrophy increase, as both variables become more intermittent. Therefore, more stringent resolution requirements are needed to accurately capture the extreme events at higher R_λ.

The probability density functions (PDFs) of normalized dissipation ($\epsilon/\langle\epsilon\rangle$) and enstrophy ($\Omega/\langle\Omega\rangle$) are also of great interest, especially concerning the likelihood of extreme events. Figure 3.9 shows the PDFs ensemble averaged over multiple realizations and time averaged over the second half of the simulations in each segment at different resolutions corresponding to R_λ 1000 and 1300. Higher spatial resolution results in wider PDF tails when larger magnitude dissipation and enstrophy are observed. This is consistent with the increase in peak values discussed above in Figure 3.8 as the spatial resolution is improved. The curves also appear to be smooth, suggesting good sampling, especially at the tails.

A feel for the intricacies in the flow structure can be obtained through a visualization of the entire domain. Figure 3.10 shows the enstrophy field from a 12288^3 snapshot at R_λ 1000. The overall picture shows clusters of high activity regions, visible in a color resembling gold, which is due to the combination of multiple colors depicting enstrophy of different intensities. A closer look at the region where the peak enstrophy (approximately 5500 times the mean) occurs shows the structure of these regions in greater detail. The high intensity enstrophy structures (colored in yellow indicating regions of enstrophy more than 400 times the mean) are worm-like, as often reported in the literature (Jiménez *et al.*,

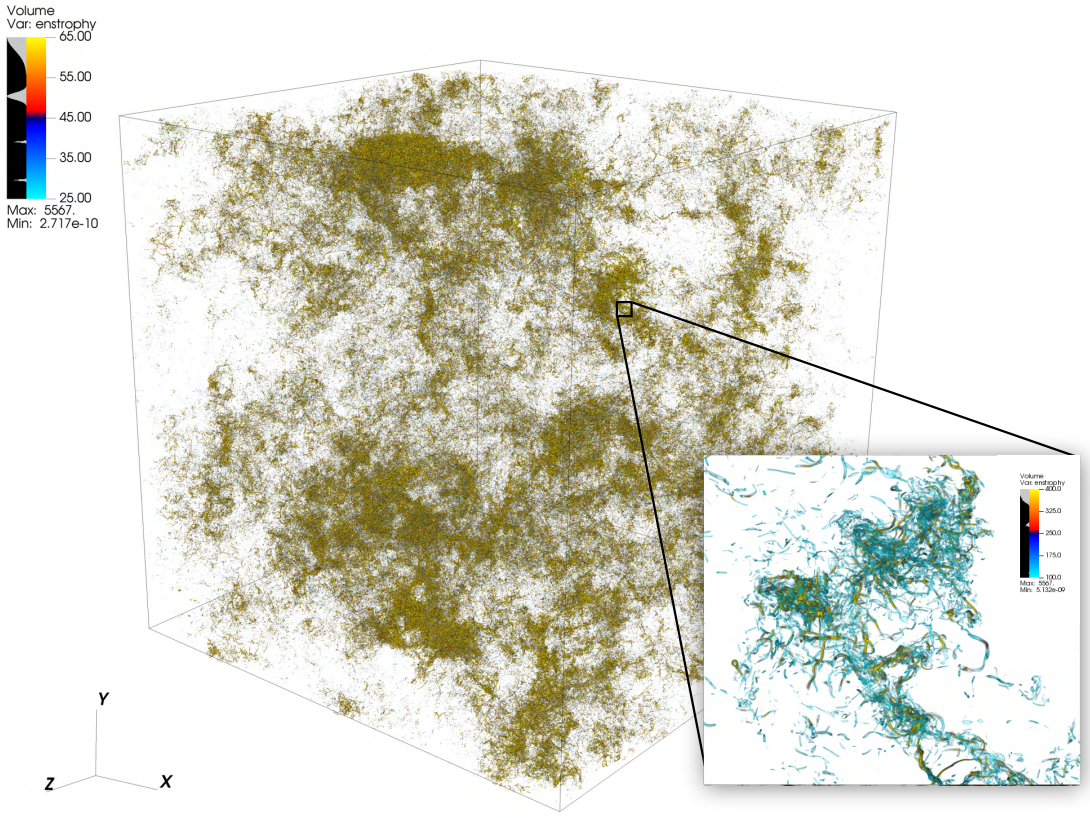

Figure 3.10: Visualization of the Enstrophy field from one simulation in the MRIS segment at $R_\lambda \approx 1000$ and $k_{max}\eta \approx 4.2$ (using 12288^3 grid points). A close-up of the high activity region (peak value of approximately 5500 times the mean) can be seen in the inset. Flow regions of intensity 400 times the mean and higher are colored in yellow in the inset. Image credit: M. A. Matheson and D. R. Pugmire, ORNL

1993; Yeung *et al.*, 2015).

3.2.2 Statistics of local averages of energy dissipation and enstrophy

The study of intermittency is a very broad subject, including the statistics of velocity gradients (Buaria *et al.*, 2019; Das & Girimaji, 2019), velocity increments (Iyer *et al.*, 2015, 2016), use of multifractal theory (Meneveau *et al.*, 1990; Meneveau & Sreenivasan, 1991), and various other aspects. A specific focus here is the behavior of the moments of the locally averaged dissipation rate (ϵ_r) and enstrophy (Ω_r), and their statistical relationships to each other while results from a multifractal viewpoint are presented later in chapter 4. The

moments of ϵ_r directly enter into a number of intermittency corrections based on the Refined Similarity Hypothesis (Kolmogorov, 1962; Obukhov, 1962; Stolovitzky *et al.*, 1992). For instance, the Refined Similarity prediction for the nth order velocity structure function is of the form

$$D_n(r) = C_n \langle \epsilon_r^{n/3} \rangle r^{n/3} \,, \tag{3.6}$$

where C_n are to be universal constants. Statistics of 3D local averages have been available only recently (Iyer *et al.*, 2015). The moments of Ω_r provide a useful contrast, as well as information on the structural differences between strain-dominated and rotation-dominated regions in the instantaneous turbulent flow.

Data on second and fourth moments from the highest resolution simulations (all with $k_{max}\eta \geq 4$) available at all four Reynolds numbers are shown in Figure 3.11, in a manner similar to that of Figure 3.7 earlier. In principle, local slopes should smoothly approach zero at both the small r and large r limits, scaling with η for the former but L_1 for the latter. For the second moment, this scaling at small r explains why the local scopes are nearly independent of Reynolds number up to r/η at least about 10, while the scaling at large r explains why, with $L_1/\eta \propto {R_\lambda}^{3/2}$ according to classical scaling, the local slope curves eventually diverge at intermediate scale ranges in the manner shown.

In Figure 3.11, two vertical dotted lines are included at $r/\eta \sim 60$ and 600, which have been proposed (Iyer *et al.*, 2015) as approximate bounds for inertial range scaling where applicable. It can be seen that as Reynolds number increases, an inflexion point gradually develops into a plateau, which is somewhat flatter for dissipation than enstrophy. The values of the exponents $\mu_{2\epsilon}$ and $\mu_{2\Omega}$ appear to differ only very slightly, with both being close to 0.23. This difference appears to be less than what past experimental data based on 1D surrogates averaged along a line suggested (Meneveau & Sreenivasan, 1991; Anselmet *et al.*, 2001; Chen *et al.*, 1997). On the other hand, greater intermittency in the dissipation range for enstrophy compared to dissipation implies local slopes at smaller r/η are of larger magnitude than those for dissipation (most significantly at $r/\eta \approx 10$ in the figure), while

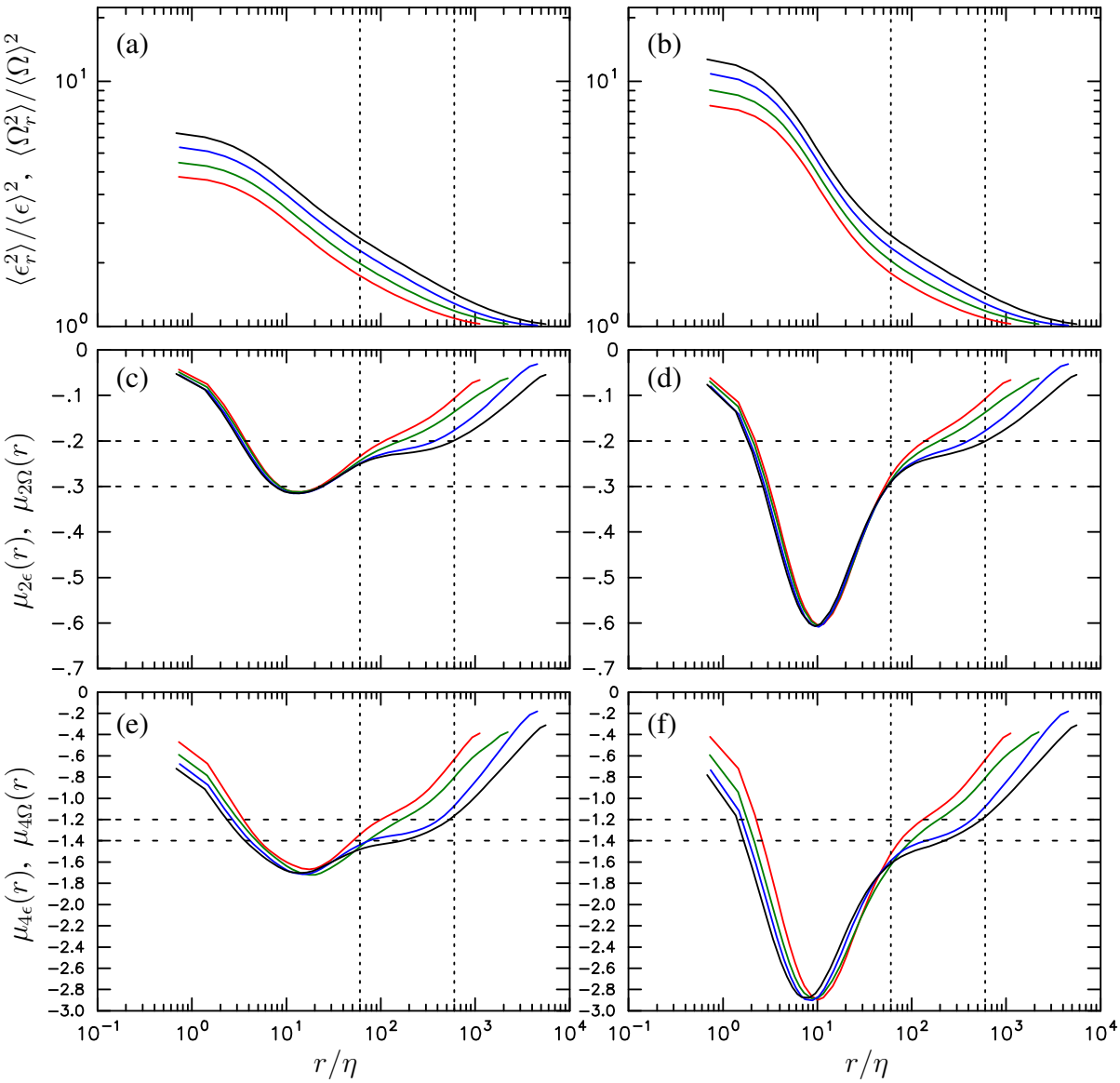

Figure 3.11: Ensemble averaged normalized second-order moments ((a) and (b)) of 3D local averages of dissipation rate, ϵ_r (left), and enstrophy, Ω_r (right), from simulations at highest resolution available at each R_λ. Ensemble average of the logarithmic local slopes of the second-order ((c) and (d)) and fourth-order ((e) and (f)) moments of local averages. The different colors correspond to different R_λ: 390 (red), 650 (green), 1000 (blue) and 1300 (black). Horizontal dashed lines in frames c-f are included to assist in inference of scaling exponents from the graphs, at the highest R_λ.

homogeneity ultimately force both sets of curves to agree with each other at sufficiently large r. Further investigations are appropriate in the future, especially when data at yet higher Reynolds numbers with a comparable degree of resolution become available.

Curves for local slopes for the fourth order moments shown in the bottom row of the

figures are of generally similar shape when compared with those for the second order moments. However, as can be expected, differences at small r indicate the small-scale resolution at higher orders is less satisfactory, especially at higher Reynolds numbers. Careful observation in the nominal inertial range of r/η also indicates inertial range behavior is less clearly developed at fourth order, while the difference between dissipation and enstrophy in the same range is more significant than that seen for the second moment.

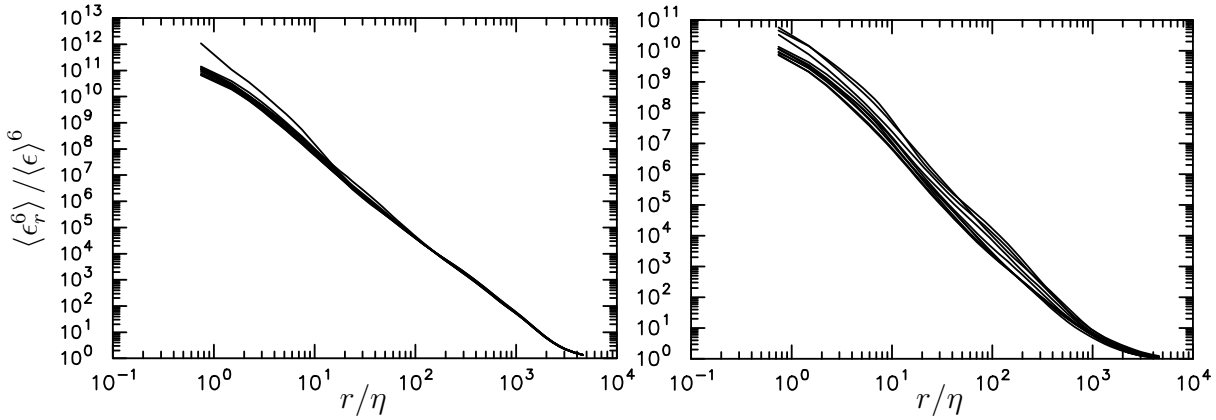

Figure 3.12: Normalized 6^{th} order moments of 3D local averages of dissipation rate ($\langle \epsilon_r^6 \rangle / \langle \epsilon \rangle^6$) from R_λ 1000 simulations. Multiple lines in each plot correspond to data from different samples. In the left, the samples are obtained from one long 12288^3 $C = 0.15$ simulation and in the right the samples are obtained from 10 ensembles of the MRIS approach using the same grid size but with $C = 0.3$.

An issue of sampling specific to local averages is that, at intermediate r, many samples of sub-cubes being counted are overlapping, and therefore not independent of each other. Figure 3.12 shows results on sixth order moments (which are more sensitive), comparing the spread among 10 snapshots for ϵ_r taken from a simulation of length $\sim 4\tau_\eta$ (on left) with those from simulation segments obtained using the MRIS approach. It can be seen that all snapshots from the case of a single simulation behave similarly, especially at intermediate and large r, whereas snapshots from independent simulation segments show significant variability for most values of r except the largest which are subjected to the constraints of a finite-sized domain. The latter variability is an indication that the MRIS approach provides more independent ensembles for averaging.

A recurrent question in the study of intermittency is whether the dissipation rate and

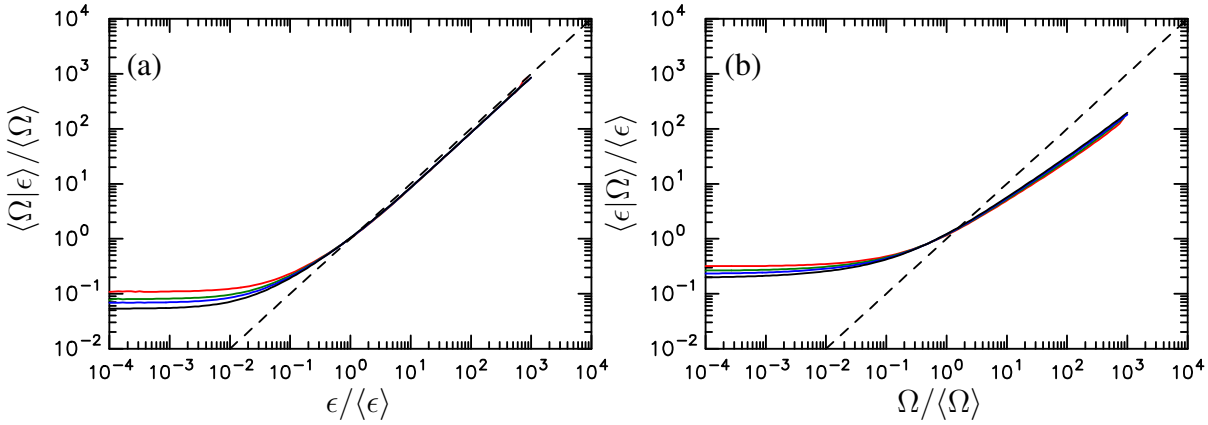

Figure 3.13: First-order conditional moments of (a) enstrophy given dissipation rate and (b) dissipation given enstrophy at R_λ 390 (red), 650 (green), 1000 (blue) and 1300 (black). Dashed line of slope 1 corresponds to enstrophy and dissipation scaling similarly.

enstrophy, as quadratic invariants of the symmetric and anti-symmetric parts respectively of the velocity gradient tensor, possess the same scaling properties (Nelkin, 1999), or even scale together (Yeung *et al.*, 2012). For an update on this question, some conditional moments derived from the present database are presented below. In the literature, conditional statistics given dissipation and or enstrophy have been used recently to study vortex stretching (Buaria *et al.*, 2020). Figure 3.13 shows the (single-point) conditional mean of (a) enstrophy given the dissipation, and (b) dissipation given the enstrophy, at four Reynolds numbers. While samples where the conditioning variable up to nearly 10^4 in magnitude do exist, results only up to 10^3 on the x-axes are shown, since data beyond that are noisy. The present results are similar to those in a previous investigation (Donzis *et al.*, 2008) at low to moderate values of the conditioning variable, but more accurate at high conditioning values of the enstrophy. The data indicate that a high ϵ is likely to be accompanied by a high Ω; but in contrast a high Ω is likely to be accompanied by a ϵ which, although still large, may be nearly an order of magnitude smaller. At the other extreme of very low dissipation or enstrophy both of the conditional means are relatively flat, with a weak trend of decrease with increasing Reynolds number. This suggests, in the limit of vanishingly small dissipation or enstrophy, both variables become independent of each other while being mostly substantially below their average intensities. This observation is also consistent with results

on joint probability density functions presented in Yeung *et al.* (2012).

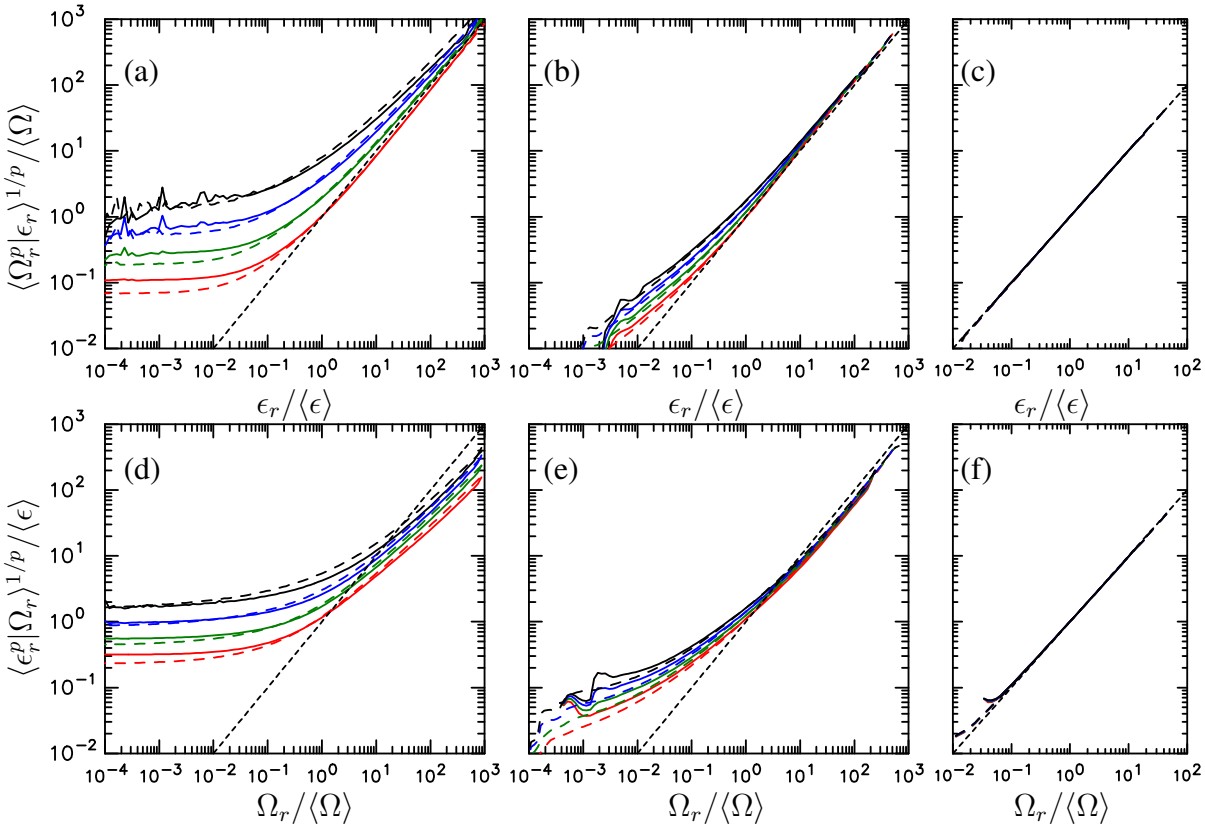

Figure 3.14: Conditional moments of order p of local averages of enstrophy given local averages of dissipation rate (top) and vice versa (bottom) for (a,d) $r/\eta \approx 0.7$, (b,e) $r/\eta \approx 11$ and (c,f) $r/\eta \approx 90$. First, second, third and fourth order moments are shown by curves in red, green, blue and black colors. Solid lines from simulations at $R_\lambda \approx 390$ and dashed lines from $R_\lambda \approx 1000$. Dashed line of slope 1 corresponds to enstrophy and dissipation scaling similarly.

Results on conditional means are extended to moments of different orders, and to local averages over volumes of linear size associated with the dissipation and inertial ranges in Figure 3.14. To facilitate the comparisons, for each $p > 1$ the pth root of the moments are taken. For a given ϵ_r, and as order p increases, moments of the conditional samples of Ω_r become increasingly dominated by samples that turn out to be very large. This explains, for instance, in the top half of the figure, why the black lines lie uniformly higher than the blue, as can be seen in frames (a) and ((b). Effects of Reynolds number on these joint statistics appear to be weak. On the other hand, in frame (c), data for moments of all orders all collapse upon the line that indicates $\langle \Omega_r | \epsilon_r \rangle = \epsilon_r$. The bottom half of this figure

shows conditional moments of dissipation given the enstrophy. It can be seen that, as r/η approaches the inertial range, both sets of conditional moments, for all Reynolds numbers, and at all orders, largely collapse together on the diagonal line that would be satisfied also if the two locally averaged variances were to take the same values. This behavior suggests, ϵ_r and Ω_r do, to a good approximation, scale together in the inertial range.

It should be apparent that the results reported in this section, involving simulations at 12288^3 and 18432^3 resolution, have required use of substantial computational power, which is itself in high demand. Recalling considerations in section 3.1, in the case of R_λ 1300, with the ratio $T_E/\tau_\eta \approx 136$ (based on Yeung *et al.* (2015)) a simulation of 5.5 T_E in length similar to the MRIS validation study earlier in this paper will be 748 τ_η in length. In contrast, to obtain 22 simulation segments (same number as in Table 3.2) only 1 τ_η each in length, the cost would be roughly equivalent to 22 τ_η. This is a factor of 34 reduction in resource requirements — changing hypothetical periods of nonstop computing from months (which is incidentally not allowed) to days, thus making a great impact on the feasibility of the computations.

3.3 Conclusions and Discussion

This chapter reports on advances in developing, and actually applying, a new capability of performing direct numerical simulations (DNS) of turbulence at extreme-scale problem sizes, that would otherwise be impossible or impractical in their resource requirements. The challenges faced here have arisen due to the fact that, despite dramatic advances in world-class computational resources, insatiable demands for high Reynolds number, improved small-scale resolution, and other needs, are pointing to increasingly unfavorable (although somewhat ironical) prospects for researchers' abilities to conduct long simulations at leadership-class problem sizes.

Despite the algorithmic advancement noted in chapter 2, it is important to point out that since resource requirements for simulating a N^3 problem for a prescribed period of

time increases at least as fast as N^4, full-length simulations spanning multiple large-eddy time scales at "leadership-class" problem sizes are essentially impossible. However, if the prime interest is in small-scale motions of short time scales, it can be shown that a much more viable alternative exists, in an approach here termed Multiple Resolution Independent Simulations (MRIS). The essence of MRIS is to first perform a (much less costly) simulation at low or modest resolution, take multiple snapshots well separated in time, and refine the grid to obtain multiple short simulation segments at the highest resolution. With appropriate attention given to statistical independence, it was shown that ensemble averaging over a number of such short segments produces results essentially equivalent to sampling from a long simulation with samples separated from each other by fractions of a large-eddy timescale. In the new paradigm, the total cost of a simulation of stationary isotropic turbulence at very high resolution can be measured in (multiples of) Kolmogorov time scales rather than eddy-turnover times, resulting in tremendous savings at high Reynolds numbers. A validation study involving several single- and multi-point diagnostics as presented in this paper has apparently been successful. In particular, results in Figure 3.3, Figure 3.4 and Figure 3.7 provided several examples of small-scale statistics obtained from the MRIS procedure being a close match with those taken directly for a full-length reference simulation at high resolution.

The new MRIS approach was successfully applied to obtain high-fidelity results concerning intermittency in both dissipative and inertial scale ranges in isotropic turbulence at four Reynolds numbers ranging from 390 to 1300 based on the Taylor scale. This work has provided an opportunity to overcome some of the limitations due to resolution and sampling in previous efforts. In particular, reliable statistics on higher order moments had been difficult to achieve, but results presented in section 3.2 are very robust, showing the benefits of leadership-class computing power applied productively. Calculations based on the statistics of 3D local averages of the dissipation rate and enstrophy show that although these quantities scale differently throughout the dissipation range, their inertial range properties

are much more (although not exactly) similar. Conditional statistics also suggest strongly that these two variables do, to a good approximation, scale together when in the inertial range. The high fidelity of results obtained in this paper gives rise readily to the search for further physical insights, such as how differences and similarities in locally averaged dissipation and enstrophy may be connected to the incidence of canonical flow structures such as local shear layers (Ishihara *et al.*, 2013) and vortex filaments of finite size.

In summary, the body of work described in this chapter provides, hopefully, a useful perspective concerning how turbulence researchers may be able to truly use emerging exascale platforms to the fullest, and the challenges that the community can expect to face, as well. The MRIS approach in this work has been developed to address the issue of how large simulations of limited time span imposed by practical constraints on resource availability can be designed to meet specific scientific needs. Other than physical problems where early-time phenomena are of the greatest interest, the MRIS approach is likely to be applicable to studies of dissipation rate, as discussed in chapter 4, and fine-scale structure in passive scalar fields (Donzis & Yeung, 2010; Iyer *et al.*, 2018), as well as the fluid particle acceleration (Yeung *et al.*, 2007), which are both dominated by intermittency and characterized by short time scales.

CHAPTER 4

EXTREME DISSIPATION AND ITS MULTIFRACTAL NATURE AT HIGH REYNOLDS NUMBERS

It is well known that a key feature in classical turbulence theory (Richardson, 1922; Kolmogorov, 1941) is that of a self-similar energy cascade, where large scales break down into smaller scales which further break down to even smaller scales each receiving a fraction of the total energy, in the manner of a so-called multiplicative process. However, it has also been well accepted for some time (Kolmogorov (1962); Frisch (1995)) that this self-similarity is only approximate, and in particular it fails to account for fluctuations of the energy dissipation rate. One of the major achievements in this thesis has been (in chapter 3) the study of the statistics of 3D local averages of the dissipation rate, with high numerical fidelity and using extreme-scale computational tools to provide such results at Reynolds numbers rivaling those in the most advanced laboratory facilities currently available.

In this chapter, further insights into the highly intermittent nature of the energy dissipation rate are obtained from a geometric, multifractal viewpoint. The intermittent, highly non-Gaussian nature of dissipation fluctuations imply that a purely statistical viewpoint (as in chapter 3) is necessarily incomplete. Low-order moments, for instance, provide little information, while high-order moments connected to fluctuations as large as 1000 times the mean are difficult to sample accurately. A multifractal framework is well-suited to describe the behavior of such highly intermittent quantities (Sreenivasan, 1991*a*). Multifractals, simply put, are built on the idea that a "measure", in this case energy dissipation, in a process is distributed unequally among the smaller scales (offspring) from the large scales (parent). As the process repeats, where the offsprings further generate smaller scales until reaching scale sizes comparable to the Kolmogorov length scale, the distribution of energy dissipation among the progressively smaller scales becomes increasingly uneven. This

chapter will focus on establishing the characteristics of multifractal scaling and computing the multifractal spectrum, which is a function that characterizes the fractal dimension of energy dissipation rate of different strengths.

As to be discussed in section 4.1, the multifractal spectrum has a direct connection to moments of the 3D local averages. In particular, moments of high positive order represent extreme events of very high amplitude, while moments of high negative order represent events of very low dissipation rate. However, because of their sensitivity to extreme events of low probability, accurate sampling is difficult. One helpful strategy for high moments is to first obtain an accurate model of the probability density function of the locally averaged dissipation, then extrapolate it beyond the tails of the PDF, and finally used the modeled PDFs to evaluate the moments. The tails of the PDF of ϵ_r have been observed to closely approximate a stretched exponential, corresponding to scale sizes ranging from the dissipation to the inertial range (Kailasnath *et al.*, 1992). The moments are computed using these extrapolated PDFs and the power-law scaling exponents, also known as generalized dimensions, D_q (Meneveau & Sreenivasan, 1991), are estimated using least square fits for different order of moments ranging from -15 to 15. These D_q's are then used to compute the multifractal spectrum.

The generalized dimensions and multifractal spectrum can be used to compute the intermittency exponents of energy dissipation and estimate the energy and volume occupied by incipient singularities (or near-singular regions) of different strengths. Past experiments in the literature have suggested that the multifractal spectrum has little dependence on the Reynolds number once a sufficiently high value has been reached. Data from DNS free from limitations including 1D averaging, 1D surrogacy and Taylor's frozen turbulence hypothesis appear to confirm these expectations. In addition, results on the multifractal spectrum support the existence of "negative dimensions" (Chhabra & Sreenivasan, 1991) corresponding to highly singular regions of the flow.

Strict resolution requirements due to the highly intermittent nature of energy dissipa-

tion rate are very challenging (Yakhot & Sreenivasan, 2005; Yeung *et al.*, 2018; Buaria *et al.*, 2019). Simulations of inadequate resolution are limited by the magnitude of velocity gradients they can accurately capture, which in turn limits the peak energy dissipation rate that can be resolved accurately. In practice, it becomes important to estimate the contribution of such extreme values to the moments of different orders, which may be quantified by computing the contribution to moments of specific orders from samples below a certain threshold.

The following sections begin with a brief summary (section 4.1) of the procedure by which the multifractal spectrum is obtained from the DNS datasets (as listed in section 4.2). Practical issues concerning the convergence of moments and extrapolation of PDF tails are addressed in section 4.3. Results on high order moments, generalized dimensions and the multifractal spectrum are presented from the largest, 18432^3, simulation in the database (Yeung & Ravikumar, 2020) at a Reynolds number of 1300 and resolution of $k_{max}\eta \sim 4.5$. In section 4.5, Reynolds number dependence is examined by comparison with data at lower Reynolds numbers. Moment contributions from regions of energy dissipation of different magnitudes will be presented in section 4.6. Conclusions are summarized in section 4.7.

4.1 Computing the multifractal spectrum using DNS data

Prior to discussions of the multifractal spectrum, a brief review of multiplicative processes is essential. In the turbulence energy cascade process, as the large scale eddies break down into smaller and smaller eddies, multiple generations can be defined. Each generation consists of a number of eddies of a characteristic linear size r. The ratio between the energy dissipation measured at two successive generations is called a "measure multiplier", defined by,

$$M_j = E_{r(j)}/E_{r(j-1)} \tag{4.1}$$

where E_r is the total energy dissipation in a certain region of space (ψ) of size r and can be obtained from the instantaneous energy dissipation rate, ϵ, as follows,

$$E_r = \int_{\mathbf{x}\in\psi} \epsilon(\mathbf{x}) d^3x \tag{4.2}$$

where ϵ is computed using the full definition involving all the velocity gradients as shown in Equation 3.2. E_r can further be related to the 3D local averages of energy dissipation rate, $\epsilon_r(\mathbf{x})$, as,

$$E_r = \epsilon_r(\mathbf{x}) r^3 \tag{4.3}$$

where ϵ_r is defined, in accordance with Equation 3.4. Similarly, $E_0 = \langle\epsilon\rangle L_0^3$ is the total energy dissipation in the cubic fluid domain of linear size $L_0 = 2\pi$ in each direction.

An abbreviated derivation following the outline of Meneveau & Sreenivasan (1991) follows. For a multiplicative process, where the energy dissipation rate is conserved at all scale sizes such that fluid motions of size r receive a fraction of the total energy dissipation rate contained in the entire fluid volume of size $(L_0)^3$, it is reasonable to assume a power-law behavior, of the form,

$$\sum E_r^q \sim E_0^q (r/L_0)^{\tau(q)} \tag{4.4}$$

where q is the order of moments, and the summation on the right is to account for the total energy dissipation over all fluid elements of size r. A scaling exponent D_q, also called generalized dimension (Sreenivasan, 1991a), is defined (Meneveau & Sreenivasan, 1991) by,

$$D_q = \tau(q)/(q-1) \tag{4.5}$$

The D_q's are estimated as the slope of the linear portion of a graph of the quantity,

$$F(r,q) = [\sum (E_r/E_0)^q]^{(1/(q-1))} \tag{4.6}$$

$$= (r/L_0)^3 [\langle\epsilon_r^q\rangle/\langle\epsilon\rangle^q]^{(1/(q-1))} \tag{4.7}$$

with respect to r/L_0 on log-log scales.

These generalized dimensions, D_q, are then used to compute the multifractal spectrum, $f(\alpha)$. Here, α is defined as the power-law scaling exponent in (Meneveau & Sreenivasan, 1991),

$$E_r/E_0 \sim (r/L_0)^{\alpha} \tag{4.8}$$

Since E_r is a random variable, with its PDF parameterized by r denoted here by $\Pi_r(\alpha)$. The summation on the right-hand side of Equation 4.4 can be represented by an integral of the form,

$$\sum E_r^q \sim \int E_r^q N_r(\alpha) d\alpha \tag{4.9}$$

where N_r is the number of "fluid pieces" of linear size r that assume a value of α within a band of width $d\alpha$. Accounting for the energy dissipation contribution of all "fluid pieces" of different strengths (α), as shown by the integral, results in the total energy dissipation contained in fluid motions of size r^3. N_r can be obtained by multiplying the total number of fluid pieces of linear size r, which is $(r/L_0)^{-3}$, by the probability $\Pi_r(\alpha)d\alpha$,

$$N_r(\alpha)d\alpha = (r/L_0)^{-3}\Pi_r(\alpha)d\alpha \tag{4.10}$$

Next, the definition of $f(\alpha)$ is introduced as (Meneveau & Sreenivasan, 1991),

$$f(\alpha) = \frac{log(N_r(\alpha))}{log(L_0/r)} \tag{4.11}$$

Using Equation 4.10, it can be shown that,

$$f(\alpha) = \frac{log(\Pi_r(\alpha))}{log(L_0/r)} + 3 \tag{4.12}$$

Equation 4.11 can be re-written as,

$$N_r(\alpha) = \rho(\alpha)(r/L_0)^{-f(\alpha)} \tag{4.13}$$

where $\rho(\alpha)$ is introduced as some α-dependent prefactor similar to that in Meneveau & Sreenivasan (1991). Substitution of Equation 4.8 and Equation 4.13 into Equation 4.9 produces,

$$\sum E_r^q \sim E_0^q \int \rho(\alpha)(r/L_0)^{q\alpha - f(\alpha)} d\alpha \tag{4.14}$$

In the limit of small r/L_0, the integrand in the above equation can be approximated by a Gaussian centered around an α value that minimizes $q\alpha - f(\alpha)$ (Meneveau & Sreenivasan, 1991) such that,

$$\int \rho(\alpha)(r/L_0)^{q\alpha - f(\alpha)} d\alpha \sim \int exp\left(-\frac{(\alpha - \alpha_*)^2}{2c^2}\right)\left(\frac{r}{L_0}\right)^{q\alpha_* - f(\alpha_*)} d\alpha \tag{4.15}$$

where α_* corresponds to the value of α which minimizes $q\alpha - f(\alpha)$ and c is the standard deviation of the Gaussian. The integral on the right-hand side in the above equation can be evaluated by taking the r/L_0 term out, which leaves only the integral of the Gaussian PDF and is unity. The integral can ultimately be evaluated as,

$$\int \rho(\alpha)(r/L_0)^{q\alpha - f(\alpha)} d\alpha \sim r^{q\alpha - f(\alpha)} \tag{4.16}$$

under the condition that,

$$df(\alpha)/d\alpha = q \tag{4.17}$$

According to Equation 4.4 and Equation 4.5,

$$\sum E_r^q \sim E_0^q (r/L_0)^{(q-1)D_q} \tag{4.18}$$

equating the powers of r in Equation 4.16 and Equation 4.18 results in,

$$f(\alpha(q)) = q\alpha(q) - (q-1)D_q \tag{4.19}$$

Further using Equation 4.17 yields,

$$\alpha(q) = \frac{d}{dq}[(q-1)D_q] \tag{4.20}$$

More details can be found in Meneveau & Sreenivasan (1991).

4.2 Description of DNS database

Table 4.1: A brief summary of DNS datasets (obtained using the MRIS method) analyzed in this chapter: showing the Reynolds number (R_λ), problem size (N^3 where $k = 1024$), number of realizations (N_r), spatial resolution (in terms of $k_{max}\eta$), corresponding simulation time span (T) in eddy turnover times (T_E), turbulence kinetic energy (K), mean energy dissipation rate ($\langle\epsilon\rangle$) and its normalized root-mean-square, Kolmogorov length scale (η) and kinematic viscosity (ν). In all cases the Courant number was 0.3. Values of T_E and integral length scale (L) are close to 0.82 and 1.1 (on a $(2\pi)^3$ domain) respectively.

R_λ	N^3	N_r	$k_{max}\eta$	t/T_E	K	$\langle\epsilon\rangle$	$\langle\epsilon^2\rangle^{1/2}/\langle\epsilon\rangle$	η $(\times 10^{-3})$	ν $(\times 10^{-4})$
390	$3k^3$	22	4.2	12.88	3.6	1.34	2.03	2.8	4.4
650	$6k^3$	15	4.2	3.17	3.5	1.31	2.16	1.4	1.7
1000	$12k^3$	10	4.2	3.78	3.9	1.43	2.32	0.69	0.69
1300	$18k^3$	10	4.5	2.53	3.8	1.35	2.48	0.5	0.44

Multifractal analyses are generally performed through post-processing of instantaneous snapshots of velocity fields archived in the direct numerical simulation database. With Reynolds number dependence as the focus, results presented in this chapter are only for the best-resolved data at each of four different Reynolds numbers listed earlier in Table 3.2. For convenience, a summary of the those "best-resolved" datasets with additional information is given in Table 4.1. It should be noted that the simulations were not run continuously for the number of eddy-turnover times given in the table. Instead, a number of short simulation

segments at high resolution were performed using the MRIS approach (as described in section 3.1) but sampled over initial conditions at lower resolution spanning the period of time shown in the table. As discussed earlier, for studies of physical phenomena with short time scales occurring in stationary turbulence (in our case, via numerical forcing), this approach provides quality results at much lower cost than "full-length" simulations at the highest grid resolutions that can be supported by leadership-class computational facilities at this time. The most important case is that at the highest Reynolds number, $R_\lambda \sim 1300$, averaged over 10 snapshots, at $k_{max}\eta \approx 4.5$ (corresponding to the grid spacing Δx being 0.68 Kolmogorov time scales).

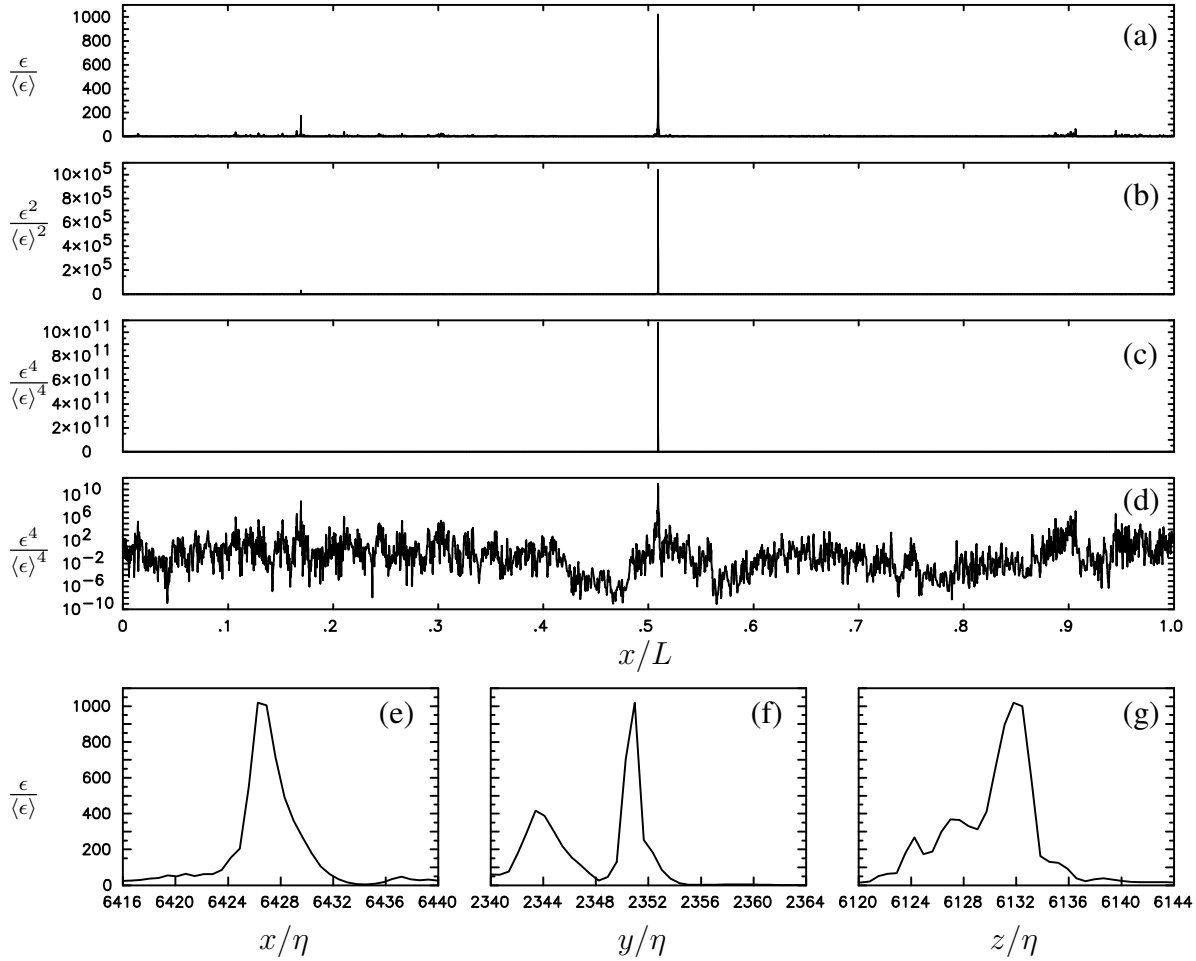

Figure 4.1: (a) Energy dissipation rate, (b) its second and (c,d) fourth powers along a line through a region of extreme dissipation that is 1/7 in intensity compared to the global peak. (e,f,g) Orthogonal line cuts around the peak.

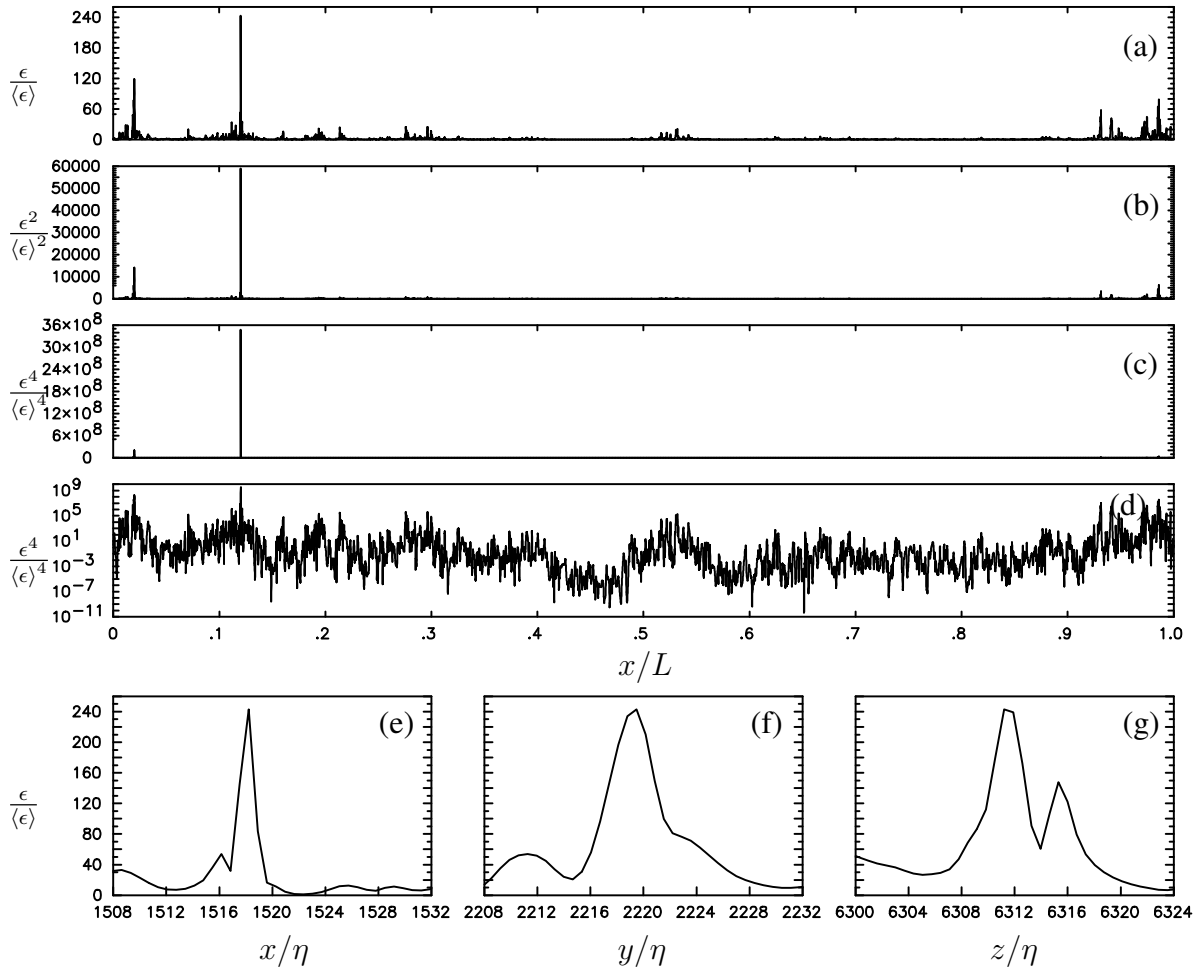

Figure 4.2: (a) Energy dissipation rate, (b) its second and (c,d) fourth powers along a line capturing more peak values as compared to Figure 4.1. (e,f,g) Orthogonal line cuts around the peak.

As noted at the beginning of this chapter, the multifractal framework involves characterizing intermittency in terms of geometric measures. Figure 4.1 shows the instantaneous energy dissipation rate, and its second and fourth powers along a line cut through a region of intense energy dissipation rate of intensity around $1/7$ of the peak value observed from one snapshot at $R_\lambda \sim 1300$. In frames (a) to (d) the x axis extends over the complete length (L_0) of one side of the solution domain. Frame (a) shows a single spike at $x/L \approx 0.5$ corresponding to the peak value recorded along this line. There are smaller variations of ϵ of the order 10 or 100 times the mean at around $x/L \approx 0.17$ and 0.9 while too low to be readily visible elsewhere. The higher powers of ϵ help emphasize the extreme events. Frames (b)

and (c) show only one discernible spike, while all the smaller variations of order 100 or lower are negligible in comparison.

The conspicuous degrees of "emptiness" in frames (a) to (c) are direct consequences of the "spottiness" nature in space of the energy dissipation rate behaving as a highly intermittent signal. Frame (d) shows the fourth power of ϵ on log scale, which confirms the presence of the energy dissipation rate at all points along the x axis. Frames (e), (f) and (g) show line cuts in orthogonal directions in the neighborhood of the peak value. It can be seen that the extreme region extends only a few Kolmogorov length scales in all three directions. Figure 4.2 shows an alternate line cut capturing a lower peak value compared to Figure 4.1, but there are a few more extreme events (of order 100 times the mean) along this line. Although the line cut captures more peaks, most of the space along the line is "empty" with events of low magnitude. Frames e, f and g in Figure 4.2 also show that the extreme region extends only for Kolmogorov length scales consistent with the higher magnitude extreme region observed previously. It should be noted that 1D cuts placed at random are likely to miss these highly localized extreme events, in a manner consistent with the observation of negative fractal dimensions (Chhabra & Sreenivasan, 1991).

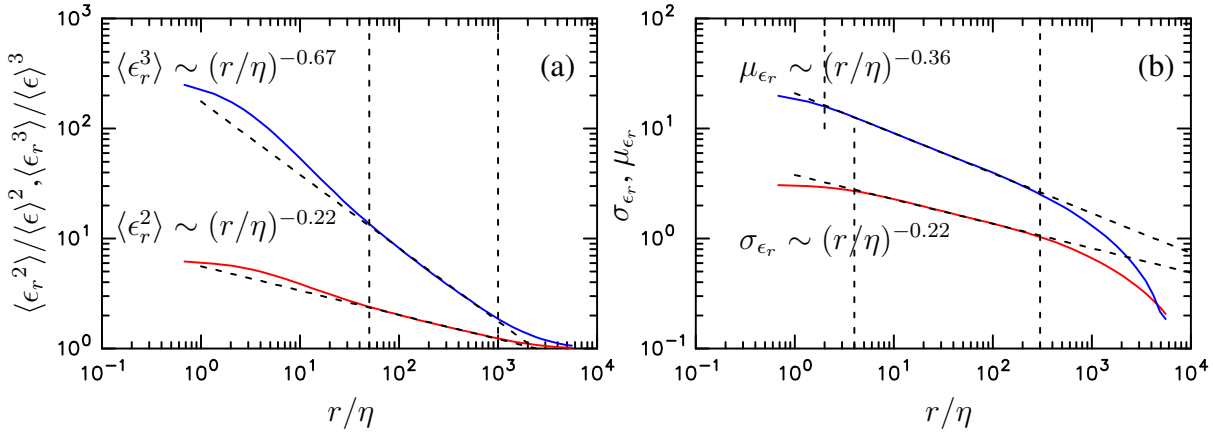

Figure 4.3: Moments of volume averaged energy dissipation rate over a subcube of size r. (a) Absolute second (red) and third order (blue) moments normalized by the global mean energy dissipation rate. (b) Standard deviation (red) and skewness (blue). Dashed slopes show the power-law scaling obtained from the least square fit performed in the scaling range marked by the vertical dashed lines. For the standardized moments, the lower limit of the scaling range decreases as the order of moment is increased.

The 3D local averages of energy dissipation rate (ϵ_r) are critical to the computation of multifractal spectrum, as discussed in section 4.1 and in Appendix A. For a basic illustration of the statistical properties of ϵ_r Figure 4.3 shows second and third order absolute and centralized moments, computed from the high-resolution data at $R_\lambda \sim 1300$. Clear scaling ranges can be observed in both the frames, where the moments show a power-law behavior with respect to the scale size r. Least-square fits are used to estimate the slopes, which are written in the body of the figures. For the absolute moments shown in frame (a), the scaling range used to estimate the scaling exponent corresponds to $50 < r/\eta < 1000$, which is wider than that used in Iyer *et al.* (2015). Frame (b) shows the second and third centralized moments, i.e. standard deviation and skewness factor. It can be seen that the moments in this form show a different scaling range, $2 < r/\eta < 300$ for the skewness, varying with the order of the moment. The second moment has the same exponent whether one uses central moments or otherwise, as was also observed by Sreenivasan & Kailasnath (1993). The numerical value of 0.22 shown in the figure is in excellent agreement with the bound of 0.25 ± 0.05 provided by those authors. For the third moment, on the other hand, it is clear that the scaling exponent depends strongly on whether one considers central moments or not. These preliminary calculations of moments of the local averages of energy dissipation show clearly the existence of a wide scaling range.

4.3 Convergence of moments and extrapolation of PDFs

Computation of the multifractal spectrum depends on the generalized dimensions, D_q, obtained from the q-th moments of E_r/E_0, as discussed in section 4.1. Therefore, it is important to ensure that data on these normalized moments, especially for higher positive and negative orders, possess a degree of statistical convergence necessary for confidence in the results presented in later sections. The PDFs of local averages of energy dissipation rate (p_r) are used to compute these moments through their connection to E_r (as given in Equation 4.3). If X denotes the random variable $\epsilon_r/\langle\epsilon\rangle$, and $p_r(x)$ denotes its PDF at the

sample value x, a straightforward assessment of convergence is to compute the integrand, $x^q p_r(x)$, that arises in computing the moments of the random variable X. Convergence can be confirmed if this term approaches zero closely at both large and small values of x corresponding to positive and negative moments q respectively.

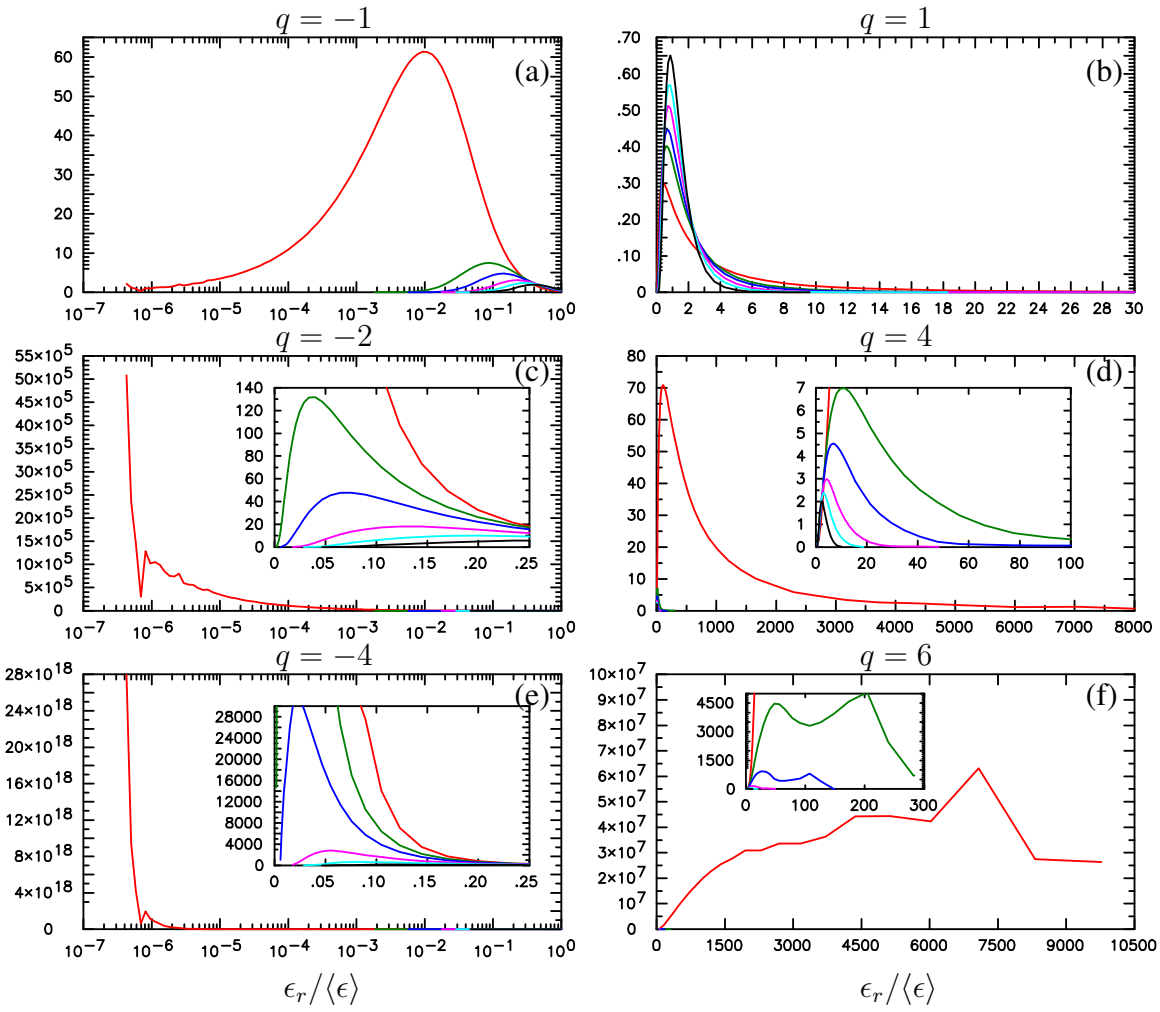

Figure 4.4: Check for statistical convergence of moments computed from the PDFs of local averages of energy dissipation rate. The integrand when computing the moments from the PDFs, $x^q p_r(x)$, are shown for different orders of magnitude, q, where X is the random sample, $\epsilon_r/\langle\epsilon\rangle$, and $p_r(X)$ is the PDF of $\epsilon_r/\langle\epsilon\rangle$. Curves of color red, green, blue, magenta, cyan and black correspond to $r/\eta \sim 0.7$ (single point statistics), 38, 76, 174, 305 and 524 respectively. A closer look at the regions of interest can be seen in the insets.

The integrand, $x^q p_r(x)$, as a function of the sample value x is shown in Figure 4.4 for different values of scale size r and moment orders q. In order to focus on the small values of $\epsilon_r/\langle\epsilon\rangle$, frames (a), (c) and (e) are shown using logarithmic scale on the x-axis. Frames (a)

and (b) show good convergence for all values of r/η in the dissipation and inertial ranges, for moments of orders 1 and -1. On the right hand, frames (c) and (e) suggest possible convergence issues for second and fourth order negative moments of the single-point ϵ (approximated by data at $r/\eta \approx 0.7$, which corresponds to a cubic grid cell of one grid spacing in size). The moments at order $q = -2$ converge well for larger values of r/η as seen in the inset in frame (c) there is potentially a mild issue for $r/\eta \approx 38$ at $q = -4$ seen in the inset in frame (e).

Frame (d) and its inset show good convergence for the positive fourth order moment for all values of r/η considered here. However, frame (f) shows convergence issues corresponding to $q = 6$ for $r/\eta < 38$. The inset in frame (f) also illustrates some problems with the values of integrands corresponding to the last few bins of the PDFs, where sampling uncertainties are inevitably present. Figure 4.4 shows that lack of convergence, if any, can be observed mostly for lower values of r/η in the dissipation range scales, while those in the inertial range remain fairly unaffected at least up to $q = 6$. It also shows that convergence for negative moments is more difficult to achieve than for positive moments. However, since the PDF values in both the lowest and highest bins are prone to uncertainty due to sampling limitations, it is useful to see whether better-behaved results can be obtained by extrapolating the tails of the PDF by some suitable means

Four different functional forms for extrapolating the PDFs are considered. The first is power-law (also called hyperbolic in Meneveau & Sreenivasan (1991)) where the tails will decay linearly on log-log plots according to

$$p(\epsilon_r/\langle\epsilon\rangle) \sim [\epsilon_r/\langle\epsilon\rangle]^{-w(r)} \tag{4.21}$$

where $w(r)$ is the scaling exponent. The second is exponential, where the tails will decay linearly on log-linear plots according to

$$p(\epsilon_r/\langle\epsilon\rangle) \sim \exp[-w(r)\epsilon_r/\langle\epsilon\rangle] \tag{4.22}$$

where $w(r)$ is a scaling exponent. The third is to assume $\epsilon_r/\langle\epsilon\rangle$ has a log-normal distribution (such that its logarithm is Gaussian distributed). Consider, $z = [\ln(\epsilon_r/\langle\epsilon\rangle) - \langle\ln(\epsilon_r/\langle\epsilon\rangle)\rangle]/\sigma_{\ln\epsilon_r}$, where $\sigma_{\ln\epsilon_r}$ is the standard deviation of $\ln(\epsilon_r/\langle\epsilon\rangle)$. The PDFs of Z and $X = \epsilon_r/\langle\epsilon\rangle$ are related by,

$$p_Z(z) = \sigma_{\ln\epsilon_r}\frac{\epsilon_r}{\langle\epsilon\rangle}p_X(x) \tag{4.23}$$

In order to test for agreement with log-normal behavior the PDF of Z, can be computed and compared with a standard normal distribution. If the lognormal behavior holds, the standard normal can be used to extrapolate the tails of the PDF of Z and then the PDF of X can be recovered by using Equation 4.23 in reverse. Finally, extrapolation of the PDF tails using a stretched exponential (SE) (Elsinga *et al.*, 2020; Buaria *et al.*, 2019) of the form shown below are considered,

$$p(\epsilon_r/\langle\epsilon\rangle) \sim \exp[-a(r)(\epsilon_r/\langle\epsilon\rangle)^{\gamma(r)} + b(r)] \tag{4.24}$$

where $\gamma(r)$ is the stretching parameter and $a(r)$ and $b(r)$ are additional parameters in this model. The PDF tails that conform to this behavior will exhibit a linear decay on log-linear plots with $[\epsilon_r/\langle\epsilon\rangle]^{\gamma(r)}$ on the x-axis, where lower values of $\gamma(r)$ represent wider tails. The case $\gamma = 0.5$ is sometimes called a "square-root" exponential.

Extrapolation of the PDF tails is relevant mostly at large ϵ_r for high-order positive moments, and at small ϵ_r for high order negative moments. The latter case corresponds to large $\langle\epsilon\rangle/\epsilon_r$, i.e. tails on the left of the PDF of $\langle\epsilon\rangle/\epsilon_r$, which is related to the PDF of $\epsilon_r/\langle\epsilon\rangle$ by the formula,

$$p\left(\frac{\langle\epsilon\rangle}{\epsilon_r}\right) \sim \left(\frac{\epsilon_r}{\langle\epsilon\rangle}\right)^2 p\left(\frac{\epsilon_r}{\langle\epsilon\rangle}\right) \tag{4.25}$$

A few tests have been performed to assess the merits of the several possible extrapolation formulas listed above. Figure 4.5 shows the PDFs of energy dissipation rate in two different forms to focus on the large values of ϵ_r in (a) with $[\epsilon_r/\langle\epsilon\rangle]^{\gamma(r)}$ on the x-axis, and small values of ϵ_r in (b) with $[\langle\epsilon\rangle/\epsilon_r]^{\gamma(r)}$ on the x-axis. The behavior of the PDF tails and

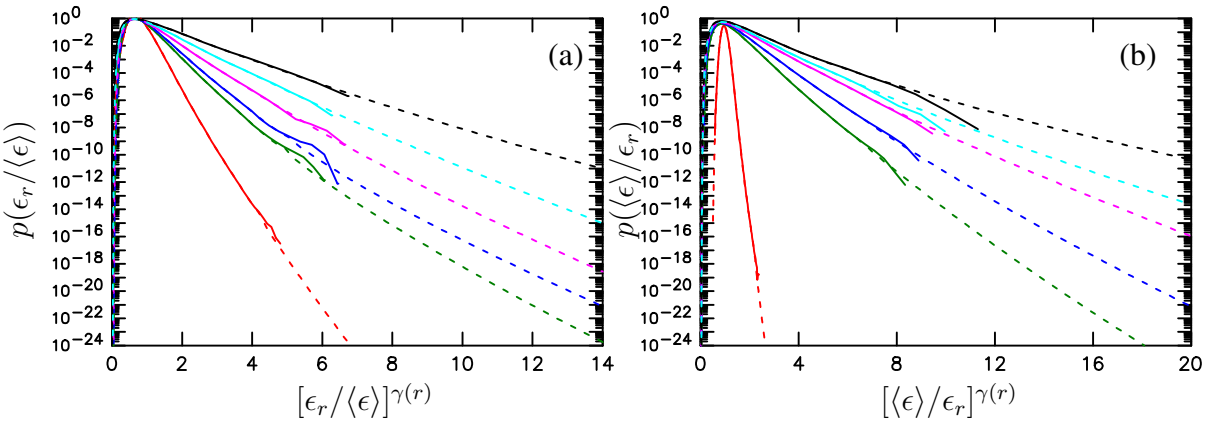

Figure 4.5: PDFs of (a) $\epsilon_r/\langle\epsilon\rangle$ and (b) $\langle\epsilon\rangle/\epsilon_r$ with the solid lines showing the actual PDFs and dashed lines showing the stretched exponential (SE) extrapolation with a variable stretching factor ($\gamma(r)$) obtained from nonlinear curve fits. Curves of color red, green, blue, magenta, cyan and black correspond to $r/\eta \sim 0.7$ (single point statistics), 38, 76, 174, 305 and 524 respectively.

the applicability of stretched exponential extrapolation can be tested from this. The values of $\gamma(r)$ for different values of r/η are obtained by performing nonlinear fits of the form shown in Equation 4.24 to the PDF tails of each individual realization, prior to ensemble averaging. To minimize contamination from sampling uncertainties in the outermost bins, the curve fits were performed using PDF data starting three points to the right of the maxima of the PDF and ending three to four points before the tail terminates. The extrapolated PDFs were then ensemble averaged, and the final results are shown. The tails of the PDFs are showing a clear linear decay, in the scales chosen, which suggests the stretched exponential fit shown by the dashed curves is in good agreement with the PDF data. The extrapolated PDFs are seen to continue to decay in a matter consistent with the stretched exponential form. Results in Elsinga *et al.* (2020); Buaria *et al.* (2019) also support the stretched exponential behavior of the PDF tails.

While tests for stretched-exponential fits are encouraging, for completeness similar tests have been performed for the other functional forms as well. Figure 4.6 is used to check the quality of a square-root exponential extrapolation, i.e., a stretched exponential with $\gamma(r) = 0.5$ being a constant. A simple least squares fit can be used to estimate the parameters $a(r)$

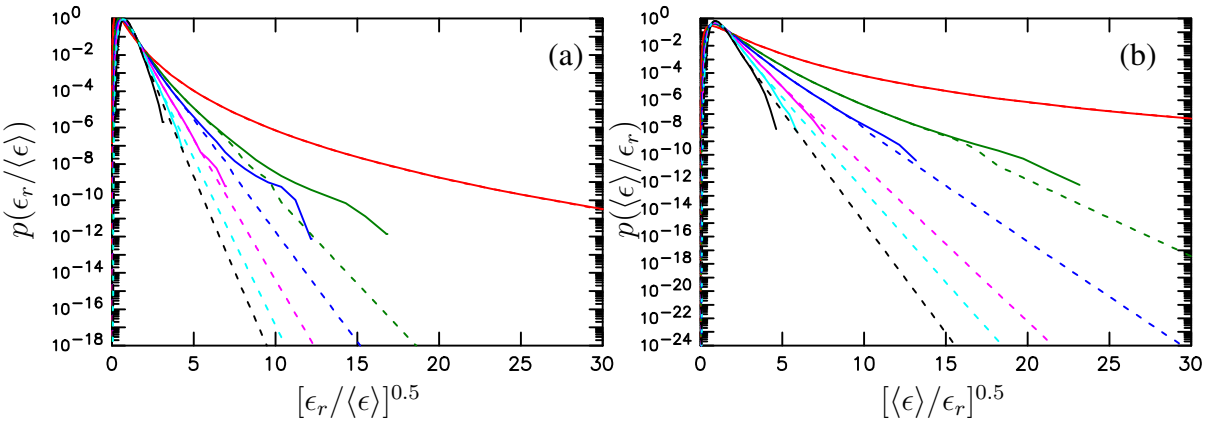

Figure 4.6: PDFs of (a) $\epsilon_r/\langle\epsilon\rangle$ and (b) $\langle\epsilon\rangle/\epsilon_r$ with the solid lines showing the actual PDFs and dashed lines showing the square-root exponential extrapolation. Curves of color red, green, blue, magenta, cyan and black correspond to $r/\eta \sim 0.7$ (single point statistics), 38, 76, 174, 305 and 524 respectively.

and $b(r)$ instead of a nonlinear fit that would be required if $\gamma(r)$ were to vary. It is clear that the PDF tails (solid lines) deviate substantially from the square-root exponential (which appears as straight lines in the linear-log scales used).

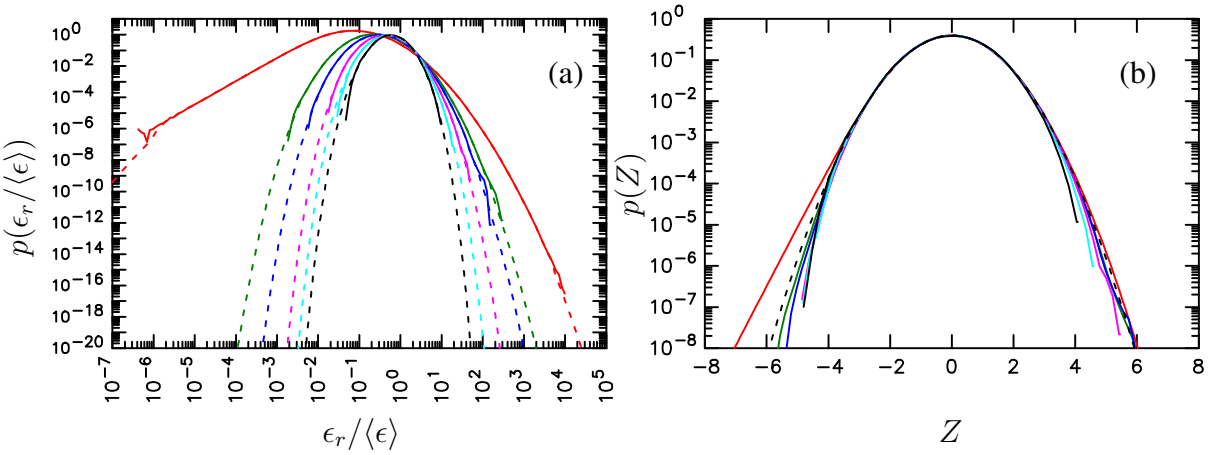

Figure 4.7: PDFs of local averages of energy dissipation rate over a subcube of size r on (a) log-log scales with the dashed curves showing stretched exponential extrapolation for comparison and (b) log-normally transformed PDFs with the dashed curve showing the log-normal extrapolation. Curves of color red, green, blue, magenta, cyan and black correspond to $r/\eta \sim 0.7$ (single point statistics), 38, 76, 174, 305 and 524 respectively.

Tests for other functional forms, namely power-law tails and log-normality assumptions as listed earlier, are shown in Figure 4.7. In frame (a) it is clear that a linear behavior on the logarithmic scales employed is not observed in the data, except perhaps in the limit of very

small ϵ_r, for values of r/η which are definitely below the inertial range. It is also worth noting that the stretched exponential describes accurately the behavior of the single point PDFs corresponding to small ϵ_r as well (Figure 4.5b). In frame (b) reasonable agreement with the Gaussian (dashed curve) can be seen for the PDF of the standardized logarithmic variable $Z = (\ln \epsilon_r - \langle \ln \epsilon_r \rangle)/\sigma_{\ln \epsilon_r}$ in the range between -3 and 3 approximately. The PDF of Z for small r/η is negatively skewed with a wide tail on the left, which is consistent with previous data on single-point moments of the dissipation rate (Yeung *et al.*, 2006). However, for larger values of r/η the PDF tails are sub-Gaussian. These observations suggest that log-normal fits may produce good results for lower-order moments, but are less satisfactory for higher-order moments where the tails of the PDF make stronger contributions.

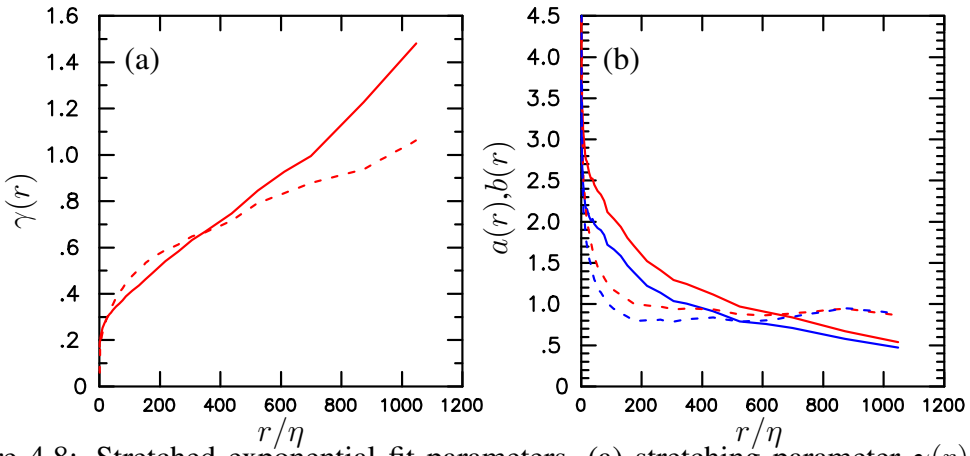

Figure 4.8: Stretched exponential fit parameters, (a) stretching parameter $\gamma(r)$ (b) $a(r)$ (red) and $b(r)$ (blue). Solid and dashed curves correspond to extrapolation of PDF tails for large and small values of ϵ_r respectively.

The discussion above highlights the fact that overall the behavior of the PDF tails corresponding to both small and large ϵ_r is best modeled by a stretched exponential (SE) with a variable stretching parameter, $\gamma(r)$ for different values of r/η ranging from the dissipation range to the inertial range. Therefore, in the following section only results from PDFs from actual data or from use of SE extrapolation will be presented. The parameters $\gamma(r)$, $a(r)$ and $b(r)$ estimated using nonlinear curve fits for PDFs at different values of r and are shown in Figure 4.8. The stretching parameter γ increases with r for tails on both sides of

the PDF, while the parameters a and b decrease rapidly at small r before starting to plateau. The general trends observed in this figure are also consistent with those in Kailasnath *et al.* (1992). However, quantitative differences arise due to the inevitable use of 1-D surrogates and Taylor's frozen turbulence hypothesis in laboratory experiments, as in Kailasnath *et al.* (1992). More recently, the tails of the single point energy dissipation PDFs have been studied and found to behave as a stretched exponential (Buaria *et al.*, 2019; Elsinga *et al.*, 2020) as well, but those papers focused only on the single point statistics. PDFs of local averages of energy dissipation were previously reported by Yeung *et al.* (2015), but the present results are more accurate and includes information on the modeling of PDF tails over a wide range of scale sizes in greater detail. The value of the stretching parameter corresponding to the single point PDFs for large ϵ is around 0.17, which is consistent with that used in Buaria *et al.* (2019). As $\gamma(r)$ increases towards a value of 2, a behavior similar to Gaussian is expected, but whether that will be observed for larger values of r is a question that is yet to be studied.

4.4 Multifractal moments and spectrum

As discussed in section 4.1, the moments of $F(r, q)$ defined in Equation 4.7 are important when computing the multifractal spectrum. In particular, the scaling exponents D_q (or generalized dimension) are estimated by identifying and estimating the slope of the moments $F(r, q)$ as a function of r/L_0 (or r/η) on log-log scales.

The moments, $F(r, q)$, and their logarithmic local slopes ($d(log(F(r, q)))/d(log(r/L_0))$) as a function of r/η are shown in Figure 4.9 on log-log plot for different orders of moments q. Here $F(r, q)$ is computed from the PDFs of $\epsilon_r/\langle\epsilon\rangle$ both without any extrapolation (solid lines) and with SE extrapolation (dotted lines). Frames (a) and (b), for negative and positive moments respectively, show that the moments scale linearly with r/η on log-log scales, especially in the scaling range ($60 < r/\eta < 600$) marked by the vertical dashed lines, which suggests a power-law behavior of the moments with scale size r does exist. As the order of

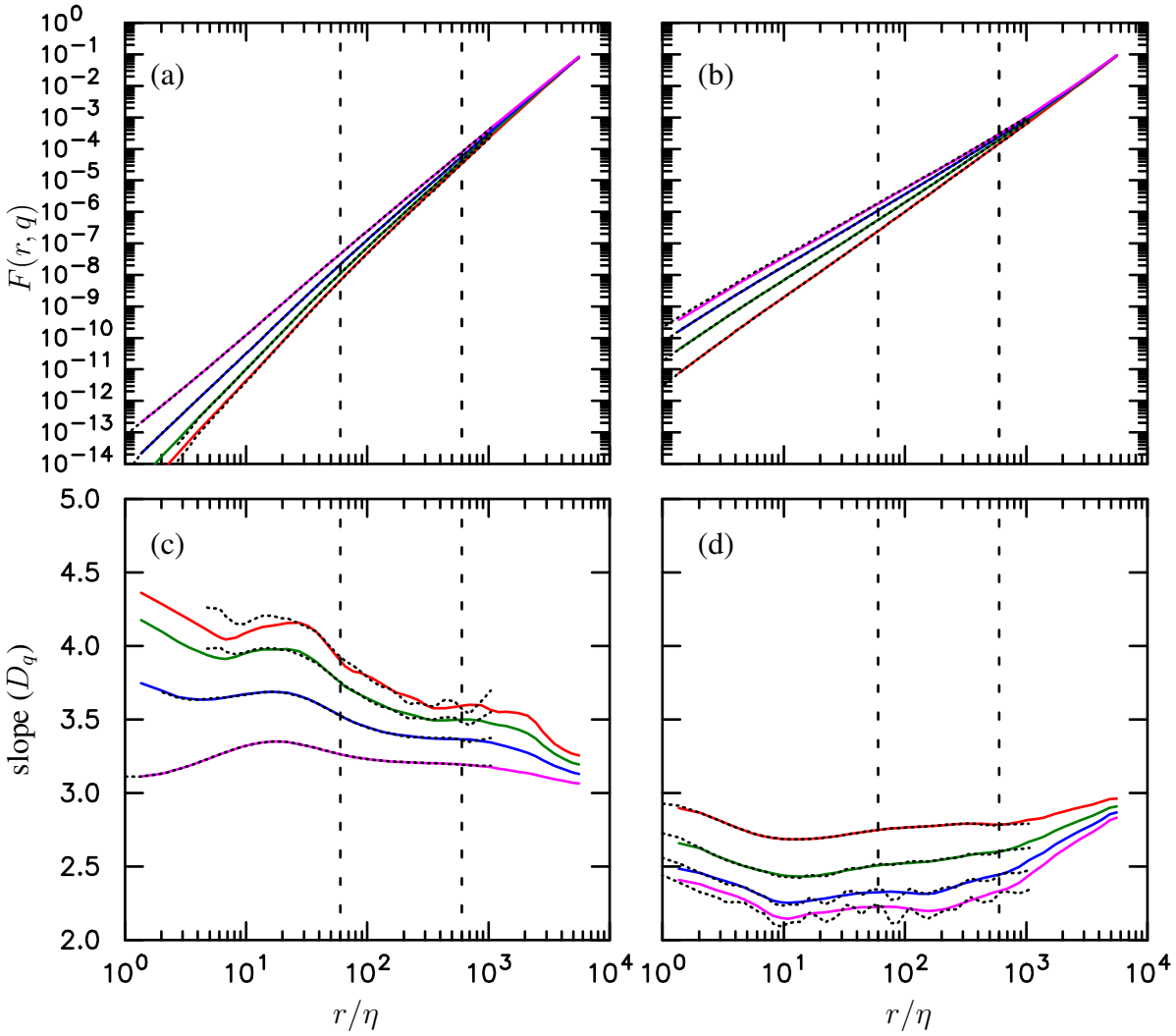

Figure 4.9: Multifractal moments (a,b) of $\epsilon_r/\langle\epsilon\rangle$ and their logarithmic local slopes (c,d) for different orders. The colors red, green, blue and magenta in each frame corresponds to (a,c) $q = -8, -6, -4, -2$, (b,d) $q = 2, 4, 6, 8$ respectively. The dotted lines correspond to moments obtained from SE-extrapolated PDFs. Vertical dashed lines corresponding to a fixed scaling range of $60 \leq r/\eta \leq 600$ are shown.

moment q increases, the slope of the moments in the scaling range is observed to decrease. The logarithmic local slopes are computed using a quadratic interpolating polynomial that makes use of three points to estimate the slope when the data involved has variable grid spacing. An exact power-law behavior would be represented by a plateau (or trough) in the plot of local slopes in frames (c) and (d). For the positive moments, especially for higher q, a flat scaling range starts to develop for $20 \leq r/\eta \leq 200$. The range of scales corresponding to which a clear plateau is also emerging for $q = -2$ but this not very clear

for high-order negative moments. Although, an exactly flat scaling range is not observed corresponding to the high positive and negative order moments, the variability observed in the scaling range ($60 < r/\eta < 600$) is approximately 10% or less.

The moments computed from the SE-extrapolated PDFs have been observed to agree well with those obtained from the actual PDFs without any extrapolation. This holds true even for the local slopes, especially for $-4 \leq q \leq 4$. However, for larger values of $|q|$ some noise in the local slopes can be observed that arise potentially due to the extrapolation parameters in Figure 4.8 not being entirely smooth. However, the local slopes, in the scaling range, are varying around some nominal value which remains constant for a wide range of scales similar to those shown by the local slopes from PDFs without extrapolation. Also, since a least-squares method is used to estimate the power-law scaling exponents from the moments in frames (a) and (b), the slight variability in the local slopes is not expected to affect the exponents estimated. The SE extrapolations were only performed for $r/\eta < 1100$ since the tails of the PDFs are not wide for larger r/η and the number of bins that can be used for the curve fit are few.

Before the generalized dimensions can be computed, it is important to identify the scaling range where the least-square fit should be performed. Two approaches are examined here. The first is a fixed scaling range corresponding to $60 \leq r/\eta \leq 600$ for all order of moments, q. This corresponds to the inertial range used in Iyer *et al.* (2015) and in section 3.2. Such a fixed scaling range is reasonable, since the local slopes shown in Figure 4.9 vary by less than 10% over this range of scales for the values of q shown there. The second option is to use a scaling range that is a function of q. In the first approach, the scaling range considered corresponds to the range of r/η, mentioned above, where the exact Kolmogorov 1941 result for the third-order velocity structure function is observed to hold well. Here, the Kolmogorov length scale (η) is defined based on the mean energy dissipation rate, $\langle\epsilon\rangle$. However, for moments of higher orders the scaling range tends to shift to lower values of r/η as evidenced by the local slopes. In order to account for this variability, a generalized

version specific for each order q is proposed by replacing $\langle\epsilon\rangle$ by $\langle\epsilon^q\rangle^{1/q}$ in the definition of Kolmogorov length scale and is referred to as the order-dependent generalized Kolmogorov length scale η_q.

$$\eta_q = \left(\frac{\nu^3}{\langle\epsilon^q\rangle^{1/q}}\right)^{1/4} \quad ; \quad \frac{\eta_q}{\eta} = \left(\frac{\langle\epsilon\rangle}{\langle\epsilon^q\rangle^{1/q}}\right)^{1/4} \tag{4.26}$$

The scaling range is defined in terms of r/η_q instead of r/η as before. That is, the order-dependent variable scaling range considered is $60 \leq r/\eta_q \leq 600$. Using Equation 4.26 the scaling range limits can be expressed in terms of r/η as,

$$\frac{r}{\eta} = 60\frac{\eta_q}{\eta} = 60\left(\frac{\langle\epsilon\rangle}{\langle\epsilon^q\rangle^{1/q}}\right)^{1/4} \tag{4.27}$$

Similarly the upper limit of the scaling range can be computed using,

$$\frac{r}{\eta} = 600\left(\frac{\langle\epsilon\rangle}{\langle\epsilon^q\rangle^{1/q}}\right)^{1/4} \tag{4.28}$$

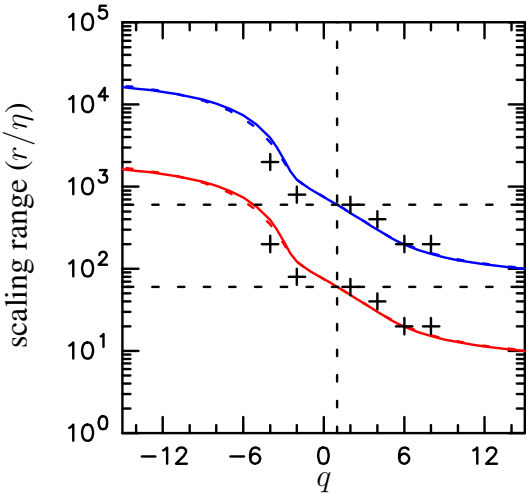

Figure 4.10: Scaling range as a function of q obtained from actual PDFs (solid lines) and PDFs with SE extrapolation (dashed lines). Red and blue are for lower and upper limits of the scaling range computed using Equation 4.27 and Equation 4.28. Empirically determined scaling ranges are marked by + signs. Dashed horizontal lines corresponding to r/η of 60 and 600 are drawn for reference to mark the chosen scaling range corresponding to $q = 1$.

The upper and lower limits of the scaling range computed from moments of energy dissipation rate according to Equation 4.27 and Equation 4.28 are shown in Figure 4.10. This suggests that for moments of order 6 the scaling range will be $20 \leq r/\eta \leq 200$ and for moments of order -6 the scaling range will be $740 \leq r/\eta \leq 7400$. The range of scales corresponding to which the local slopes remain fairly constant, for some values of q shown in Figure 4.9 previously, is determined empirically and marked by + symbols. The empirically determined scaling ranges seem to agree relatively well with those determined using the approach described above, based on the order-dependent generalized Kolmogorov length scale. The scaling range limits for the negative order moments correspond to very large values of r/η where reliable curve fits for extrapolation of PDF tails may not be available. The figure also shows good agreement between the scaling ranges obtained from SE-extrapolated PDFs and the actual PDFs.

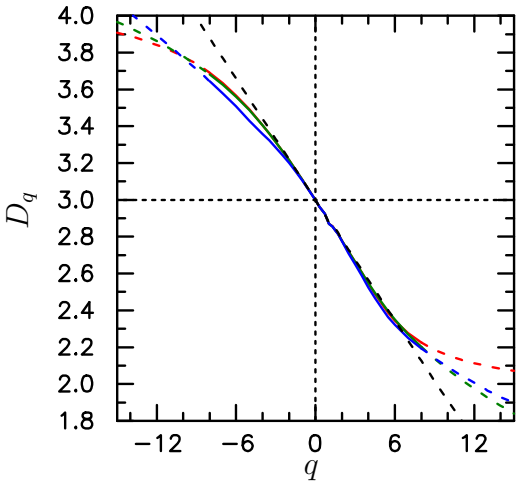

Figure 4.11: Generalized dimensions (D_q) for different orders of magnitude estimated in the scaling range using a least-square fit. Data from PDFs with no extrapolation using a fixed scaling range (red), SE-extrapolated PDFs with a fixed scaling range (green) and SE-extrapolated PDFs with variable scaling range (blue) are shown. Results from log-normal extrapolation of PDFs are shown using a black dashed curve for reference.

The generalized dimensions, D_q, can be estimated using the least square method to fit a straight line to the curves of $log(F(r,q))$ vs $log(r/L_0)$ shown in Figure 4.9 previously. This fit was performed for moments of order q ranging from -15 to 15 using both the fixed scaling range based on the classical Kolmogorov length scale and the variable scaling range

based on the order-dependent generalized Kolmogorov length scale. The parameters of the fit are estimated for each realization individually and then ensemble averaged. The slope of the line fit gives the generalized dimension, D_q and is shown in Figure 4.11 for different values of q. The values of D_q estimated from the actual PDFs and SE-extrapolated PDFs are in good agreement with each other for q in the range -8 to 8. For $|q| > 8$ results shown as dashed lines do differ significantly from one another, reflecting the degree of uncertainty for higher-order moments which does persist, regardless of whether the tails of the PDFs are extrapolated (as stretched exponential). The power-law scaling exponents from the actual PDFs show a clear tendency to level off for both, higher positive and negative order moments. The values of D_q estimated using a variable scaling range using SE extrapolated PDFs also show good agreement with results obtained from a fixed scaling range. The difference is more prominent for higher orders of $q \geq 10$ and $q \leq -3$. Also, at $q = 0$, the value of D_q from all the curves in the figure is 3.0 which is expected as the quantity is space filling for $q = 1$.

Before proceeding further to the actual multifractal spectrum, it is important to understand the significance of the quantity α which characterizes the strength of "near-singularities" or flow regions of varying energy dissipation rate. Frame (a) in Figure 4.12 shows α as a function of the order q. Moments at large values of q, which are dominated by regions of very high energy dissipation rate lead to lower values of α while for small values of q, which are dominated by regions of very low energy dissipation rate lead to higher values of α.

As noted earlier, accurate knowledge of the functional form of D_q versus q allows the quantity α and ultimately the multifractal spectrum $f(\alpha)$ to be determined, according to Equation 4.20 and Equation 4.19 respectively. Frame (b) of Figure 4.12 shows the multifractal spectra for the same three cases as in Figure 4.11, with a number of corresponding features. Specifically, the spectrum computed from the actual PDFs and extrapolated PDFs are in good agreement at least for values of α corresponding to q in the range of -8 to 8

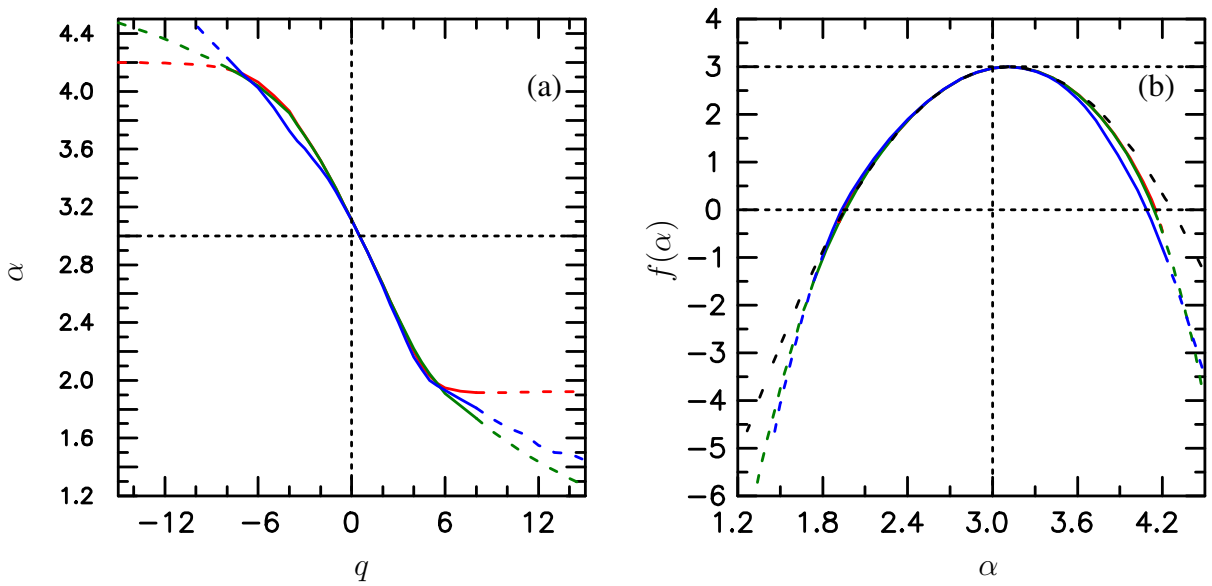

Figure 4.12: (a) Strength of the near-singularities characterized by α as a function of the order q and (b) multifractal spectrum $f(\alpha)$ vs α. Data from PDFs with no extrapolation using a fixed scaling range (red), SE-extrapolated PDFs with a fixed scaling range (green) and SE-extrapolated PDFs with variable scaling range (blue) are shown. The multifractal spectrum in the form of a parabola corresponding to log-normal extrapolation of PDFs are shown using black dashed curve in frame (b) for reference.

shown by the solid lines. The spectrum computed from the SE extrapolated PDFs extend further out to lower (and higher) values of α as compared to the spectrum from actual PDFs. It is also important to note here that the $f(\alpha)$ values are negative, corresponding to small and large values of α. The spectrum from the SE-extrapolated PDFs with variable scaling range also is in good agreement with the other curves, except for large and small values of α similar to the D_q curves.

If the PDFs of energy dissipation rate were log-normal, the corresponding generalized dimensions, D_q, would be scaling linearly with the order q as shown in frame (a) of Figure 4.11 and in Meneveau & Sreenivasan (1991). An equation of a straight line of the form $D_q \sim mq$ can be used to describe this, where m is the slope of the line. Simplifying

Equation 4.20 and Equation 4.19 using the linear form of D_q the following can be obtained,

$$\alpha = 2mq + 3 - a \tag{4.29}$$

$$f(\alpha) = mq^2 + 3 \tag{4.30}$$

This shows that the corresponding spectrum expected will be a parabola if the log-normal hypothesis holds and is shown by the dashed black curve in frame (b) of Figure 4.12 above. The figure shows the spectrum from actual and SE-extrapolated PDFs are narrower than those expected from log-normal PDFs for large and small α, i.e., log-normal theory predicts a larger fractal dimension corresponding to singularities of fixed strength (α).

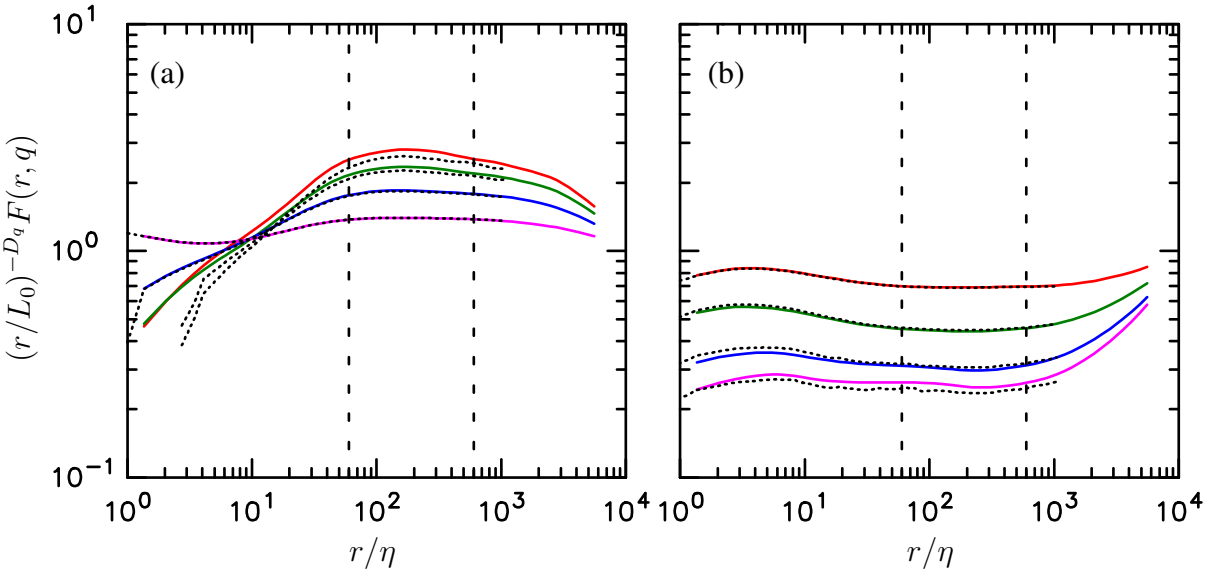

Figure 4.13: Compensated multifractal moments weighted by $(r/L_0)^{-D_q}$ for different orders of moments obtained from actual PDFs (solid lines) and PDFs with SE extrapolation (dashed lines) similar to Figure 4.9. Vertical dashed lines corresponding to a fixed scaling range of $60 \leq r/\eta \leq 600$ are shown. Colors red, green, blue and magenta correspond to moments (a) q =-8,-6,-4,-2 and (b) q =2,4,6,8 respectively.

The results discussed here show that the data from SE-extrapolated PDFs with fixed scaling range can be used for further analysis, as very little difference is observed when a variable scaling range is used, as shown in Figure 4.12. Additionally, the large scaling range required to capture the higher-order negative moments cannot be met because SE-

extrapolated PDFs are not available for $r/\eta > 1100$.

In order to test the accuracy of the generalized dimensions estimated previously, the moments, $F(r,q)$ defined in Equation 4.7, are weighted by $(r/L_0)^{-D_q}$ using D_q values from Figure 4.11. Figure 4.13 shows these compensated moments for different orders of moments, q. A plateau in the plot of these compensated moments corresponding to the scaling range $60 \leq r/\eta \leq 600$ suggests the values of D_q estimated are reasonably accurate. The figure indeed shows that a relatively flat region is observed for the different orders of moments considered here and for data from both actual PDFs and SE extrapolated PDFs.

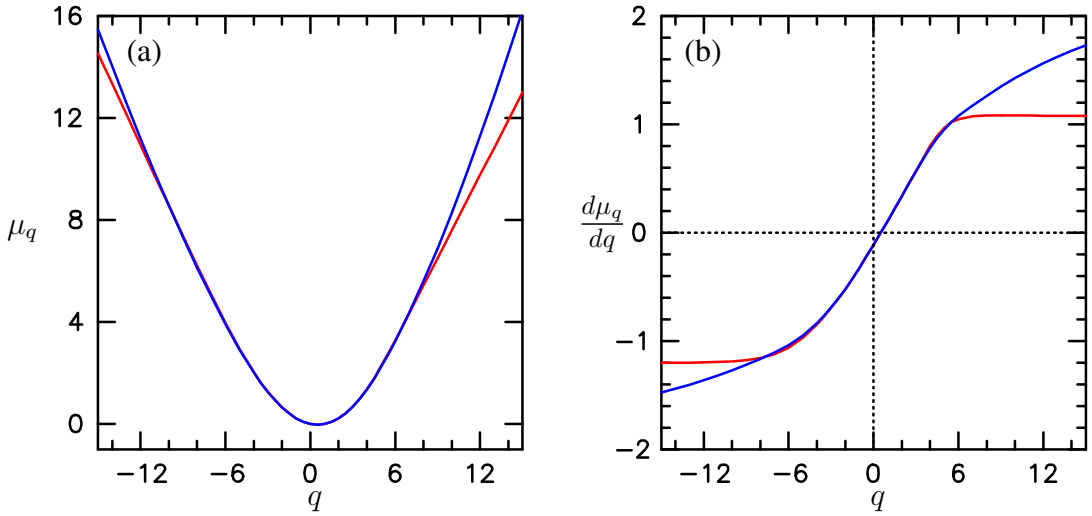

Figure 4.14: (a) Intermittency exponent computed using D_q. (b) Slope of the intermittency exponent with respect to the moment order q. Data from PDFs with no extrapolation is shown in red and from SE extrapolation in blue. Asymptotic value of the local slopes for large q gives D_∞.

One of the important uses of the generalized dimensions is to compute the intermittency exponent, μ_q, defined by the scaling behavior $\langle {\epsilon_r}^q \rangle \sim \langle \epsilon \rangle^q (r/L_0)^{-\mu_q}$ for high orders q. These exponents are important to many intermittency theories (Sreenivasan, 1993; Watanabe & Fujisaka, 2000) and are related to D_q as

$$\mu_q = (d - D_q)(q - 1) \tag{4.31}$$

where d is the topological dimension of the space (which is 3 in 3D turbulence). As noted

in Bershadskii & Mikishev (1994), differentiation of both sides of Equation 4.31 with respect to q, in the limit where $dD/dq \rightarrow 0$ as $q \rightarrow \infty$ gives the result $d\mu_q/dq \rightarrow D_\infty$. Figure 4.14 shows the intermittency exponent computed according to Equation 4.31 and its slope with respect to the moment order q. The exponents estimated from both actual PDFs and SE-extrapolated PDFs agree well up to $q = 8$, beyond which the two curves diverge. The intermittency exponent corresponding to the second order moments is the most fundamental (Sreenivasan, 1993) and the corresponding value estimated here is around 0.22, which is consistent with the results presented in section 3.2 and section 4.2.

The slope of the intermittency exponent in frame (b) shows that, for $q \geq \pm 8$, the local slopes are nearly constant corresponding to results from PDFs with no extrapolation while those from SE-extrapolated PDFs are not showing a clear tendency to flatten out in the range of q considered here. The value of the slope corresponding to large q is around 1.1 for data from no extrapolation, while data from SE-extrapolated PDFs suggest a plateau is expected at a much larger value. The asymptotic value of the slope corresponds to D_∞, as discussed previously. As q increases, the D_q curves in Figure 4.11 discussed previously will continue to decrease and eventually flatten out close to the value of D_∞ estimated from Figure 4.14.

4.5 Multifractal spectrum at different Reynolds numbers

While section 4.3 and section 4.4 contained only results at the highest Reynolds number available (R_λ 1300), useful information on Reynolds number dependence can be obtained by comparing results at different (lower) Reynolds numbers. Figure 4.15 shows the generalized dimensions, D_q, and multifractal spectrum for $R_\lambda \sim$ 390, 650, 1000 and 1300, all at comparably high resolution of the small scales. Frames (a) and (b) show results obtained from PDFs without any extrapolation, while frames (c) and (d) show results from PDFs with extrapolated stretched exponential tails. Generalized dimensions for negative moment orders show less dependence to Reynolds number than those for positive orders.

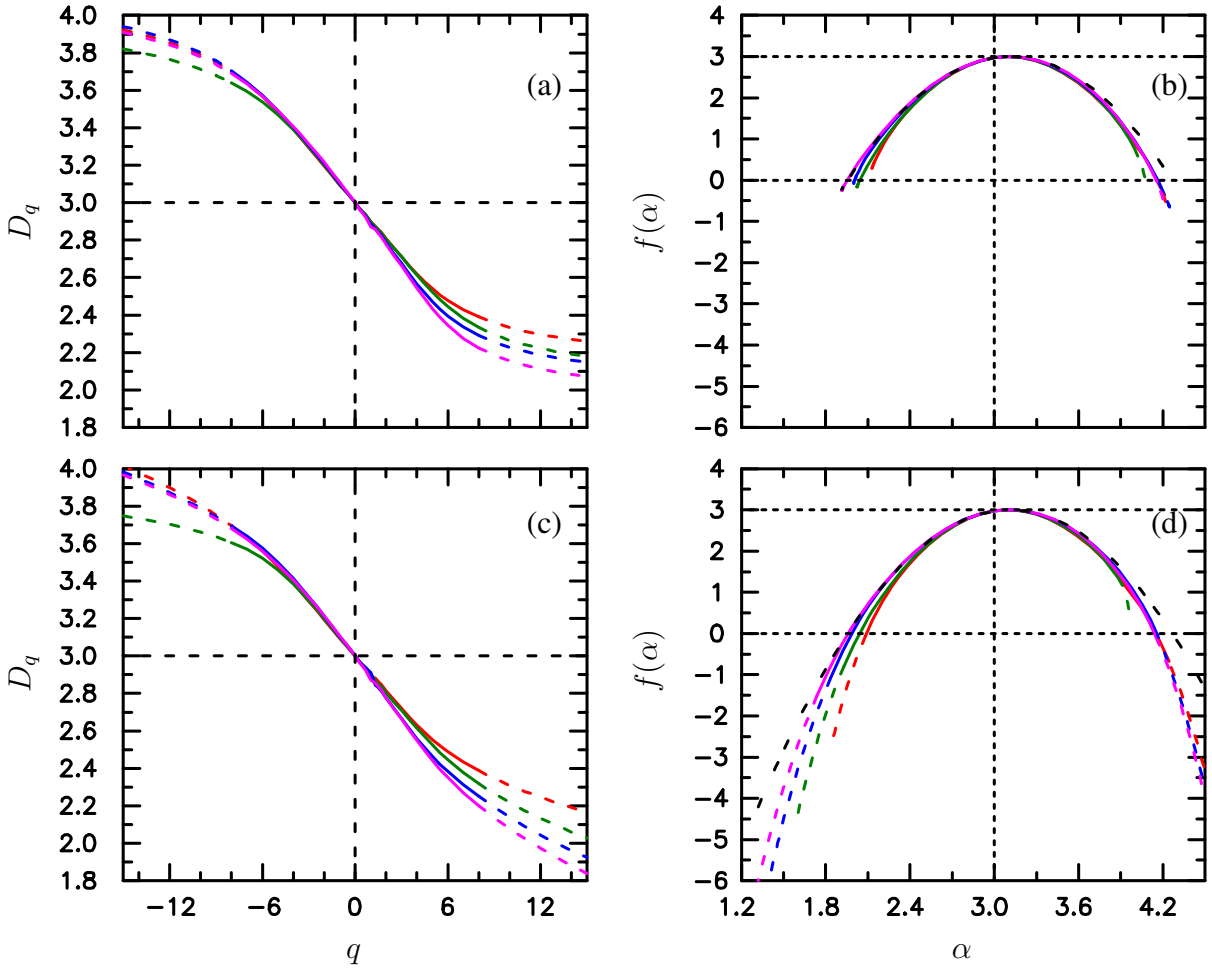

Figure 4.15: Generalized dimensions (a,c) and the multifractal spectrum (b,d) for $R_\lambda \approx$ 390 (red), 650 (green), 1000 (blue) and 1300 (magenta). The top row corresponds to data obtained from actual PDFs without extrapolation, and the bottom row corresponds to data from SE extrapolated PDFs. A parabola corresponding to the expected spectrum from log-normal behavior of PDFs is shown by the black dashed curves in frames (b) and (d). Dashed curves correspond to data for $q > \pm 8$.

At high negative orders, the case of $R_\lambda \sim 650$ appear to be an outlier, which may be due to undersampling of samples of dissipation rate several orders of magnitude smaller than the mean. The positive moments, however, show a clear trend of decreasing generalized dimensions as the R_λ increases for a fixed q. This observation is more prominent in the D_q values computed using SE-extrapolated PDFs in frame (c).

The multifractal spectra in frames (b) and (d) both show little dependence on Reynolds number at high α. The figures show that the spectrum is narrower than the parabola (black

dashed line) which corresponds to a log-normal behavior in the energy dissipation rate, similar to observations corresponding to Figure 4.12. However, at lower values of α the multifractal spectrum clearly becomes wider at higher Reynolds number. An interesting observation is that the solid lines representing the spectrum terminate at lower values of $f(\alpha)$ and α when the R_λ is higher. Therefore, near-singularities of greater strength (lower α) and lower fractal dimension are observed as R_λ increases. It is also important to observe that the fractal dimensions $f(\alpha)$ are negative (Chhabra & Sreenivasan, 1991), even more so, when SE-extrapolated PDFs are used. Large α corresponds to the higher order negative moments which are dominated by the regions of very low energy dissipation, whereas small α correspond to the higher order positive moments which are dominated by the regions of very high energy dissipation, as shown in Figure 4.12a previously.

Negative fractal dimensions are mostly observed for small or large values of α corresponding to very high or very low energy dissipation. These regimes, as indicated by the PDFs in Figure 4.7a, have very low probability of occurrence. Consider that the events are being sampled using box cuts in the entire domain. Such events would be missed by any single cubic box in space, but averaging over many such boxes would amount to collecting information about such low probability events. A negative exponent in Equation 4.13 corresponds to a decrease in the number of boxes as r decreases, which means that smaller boxes are not likely to capture these rare events, with many of them remaining empty. This provides a unique geometrical viewpoint to intermittency as compared to the statistical viewpoint taken in chapter 3. It is important to note here that such large negative dimensions are observed only corresponding to very high and low values of energy dissipation, which are obtained from extrapolating the PDF tails. Therefore, the results need to be interpreted carefully, as these extrapolations do not confirm the occurrences of such rare events in physically realizable flows. Another important factor is the small number of samples that capture these rare events, which may contribute to undersampling. This consideration raises the possibility that if more samples show the occurrence of these rare events, then

the resulting dimensionality may not be negative. Chhabra & Sreenivasan (1991) provided a more detailed account of "negative dimensions" as observed in the multifractal spectrum. On the other hand, moderately high or low values of energy dissipation correspond to positive values of fractal dimensions. Such events are more likely, and hence many if not all the box cuts are likely to capture occurrences of these events. As the box size r decreases, there are more such boxes capturing these events. Therefore, a positive exponent corresponds to an increase in N_r when r decreases corresponding to the argument above.

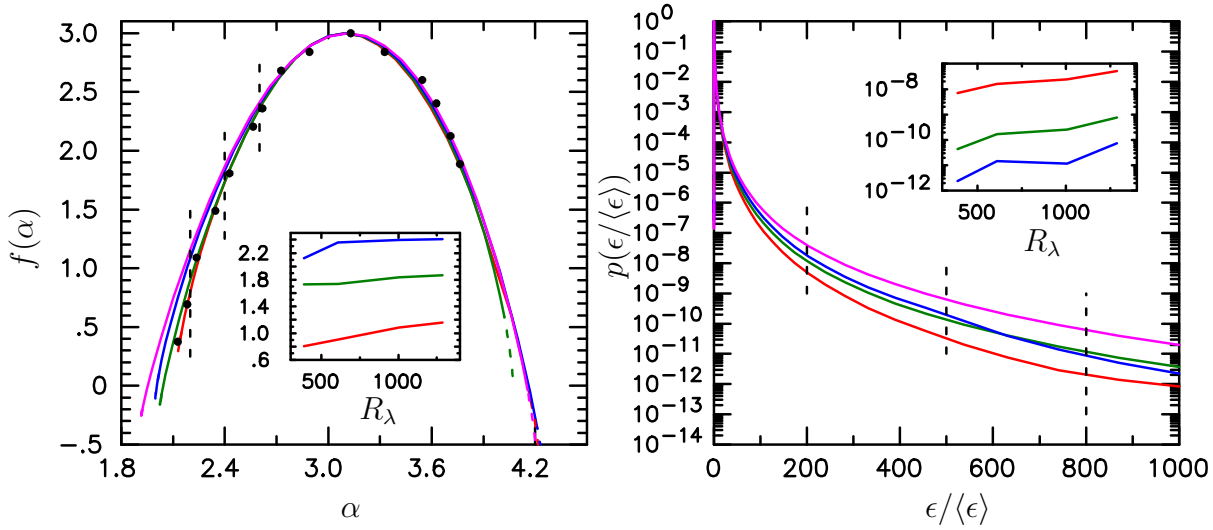

Figure 4.16: (a) Multifractal spectrum at R_λ 390 (red), 650 (green), 1000 (blue) and 1300 (magenta). Solid circles (•) show data obtained from figure 38a in Meneveau & Sreenivasan (1991). Inset shows the values of the multifractal spectrum at fixed values of α corresponding to 2.2 (red), 2.4 (green) and 2.6 (blue) as a function of R_λ on the x-axis. Vertical dashed lines in the main frame are used to mark the intersection with the $f(\alpha)$ curves corresponding to the fixed values of α. (b) PDFs of energy dissipation rate at R_λ 390 (red), 650 (green), 1000 (blue) and 1300 (magenta). The inset shows the values of the PDFs at fixed values of $\epsilon/\langle\epsilon\rangle$ corresponding to 200 (red), 500(green) and 800 (blue) as a function of R_λ on the x-axis. Vertical dashed lines in the main frame are used to mark the intersection with $p(\epsilon/\langle\epsilon\rangle)$ curves corresponding to the fixed values of $\epsilon/\langle\epsilon\rangle$.

An attractive characteristic of the multifractal spectrum is that it is independent of Reynolds number, especially at higher R_λ. While the PDFs of energy dissipation rate can also be used to study intermittency, as in chapter 3 previously, they are highly dependent on R_λ. Figure 4.16 shows the multifractal spectrum (same as in Figure 4.15b) alongside the PDFs of normalized energy dissipation rate at different R_λ. In order to compare the

variability of the two quantities with R_λ, fixed values of α and $\epsilon/\langle\epsilon\rangle$ are chosen, as both these variables characterize the intensity of the extreme events, and the corresponding values of $f(\alpha)$ and $p(\epsilon/\langle\epsilon\rangle)$ are shown as a function of R_λ in the insets in each frame. The largest variation in the multifractal spectrum among the tested values of α is observed, corresponding to $\alpha = 2.2$. The relative variation between results at $R_\lambda \sim 390$ and 1300 with respect to those at $R_\lambda \sim 1300$, defined as $(f_{1300}(\alpha) - f_{390}(\alpha))/f_{1300}(\alpha)$, is found to be close to 30% while the same corresponding to $\alpha = 2.6$ is approximately 2%. It is also clear from frame (a) that the variation with R_λ is low corresponding to $\alpha > 3$ which corresponds to regions of weak energy dissipation rate. Similarly, the relative variation between the PDFs at $\epsilon/\langle\epsilon\rangle = 800$ for $R_\lambda \sim 390$ and 1300, using the definition described above, is approximately 97% while the same for $\epsilon/\langle\epsilon\rangle = 200$ is approximately 86%. This shows that the tails of the PDFs vary more significantly with R_λ in comparison to the multifractal spectrum. Although variations as high as 30% are recorded for small values of α, at lower R_λ very few samples with such strong energy dissipation rate are observed, which could lead to sampling uncertainties contributing to less than accurate values for the multifractal spectrum at low R_λ. For example, the relative variability in the multifractal spectrum corresponding to $\alpha = 2.2$ between $R_\lambda \sim 1000$ and 1300 is approximately 6% only, while the variation in the PDFs corresponding to $\epsilon/\langle\epsilon\rangle = 800$ for the same two Reynolds numbers is close to 84%. This shows that the R_λ-dependence of the spectrum indeed reduces at higher R_λ while the R_λ-dependence is still significant in the PDFs.

Figure 4.16a also shows some data points corresponding to the multifractal spectrum, taken from figure 38a of Meneveau & Sreenivasan (1991). The spectrum was obtained from measurements in laboratory flows like wakes and boundary layers with R_λ around 100. A 1D surrogate for the energy dissipation rate was used, with Taylor's frozen turbulence hypothesis and 1D local averaging. In contrast, in DNS the results from the 3D local averages of full energy dissipation rate computed using all nine velocity gradient components and at R_λ of 390, the lowest in this work, are found to be relatively in good agreement with

each other. This suggests the multifractal spectrum is very robust to both R_λ and use of approximations like 1D surrogates and Taylor's frozen turbulence hypothesis.

The multifractal spectrum is especially useful in the study of near-singularities in the equations of fluid motion. In practical turbulent flows, strictly singular regions (where velocity gradients are infinite) are not observed, since they are smoothed out by viscosity. However, regions of very high velocity gradients or energy dissipation may be considered as incipient singularities or near-singular regions. An important question in the study of these incipient singularities is regarding the amount of energy dissipation contributed, and the volume occupied by these regions. Let S denote the set of near-singularities characterized by their strength α. (Sreenivasan & Meneveau, 1988) showed that the volume occupied ($V(S)$) and fraction of total energy dissipation contained ($E(S)$) in the near-singular set (where $\alpha < 3$ indicates high dissipation) are given by the formulas

$$V(S) = \int_{\alpha \in S} (r/L_0)^{d-f(\alpha)} d\alpha \tag{4.32}$$

$$E(S) = \int_{\alpha \in S} (r/L_0)^{\alpha-f(\alpha)} d\alpha \tag{4.33}$$

where $d = 3$ in three-dimensional flow.

Results from a theoretical calculation of the quantities $V(S)$ and $E(S)$ are shown as a function of Reynolds number in Figure 4.17. It should be noted that, of course, the bulk of the R_λ range is this figure is beyond the reach of DNS at present and even potentially far into the future. Data points in this figure are obtained by (i) taking $f(\alpha)$ to be the same as that from DNS data at R_λ 1300 using SE-extrapolated PDFs, and (ii) that $r/L_0 = (r/\eta)R_\lambda^{-3/2}$. Assumption (i) is supported by the weak sensitivity of the multifractal spectrum with Reynolds number (as seen in Figure 4.16 earlier), while assumption (ii) is just based on classical scaling for the ratio L/η (with L/L_0 also nearly constant in forced turbulence). Frame (a) shows that, for near-singular regions defined by $\alpha < 3$, dominated by high dissipation, the energy dissipation contained increases with R_λ while the

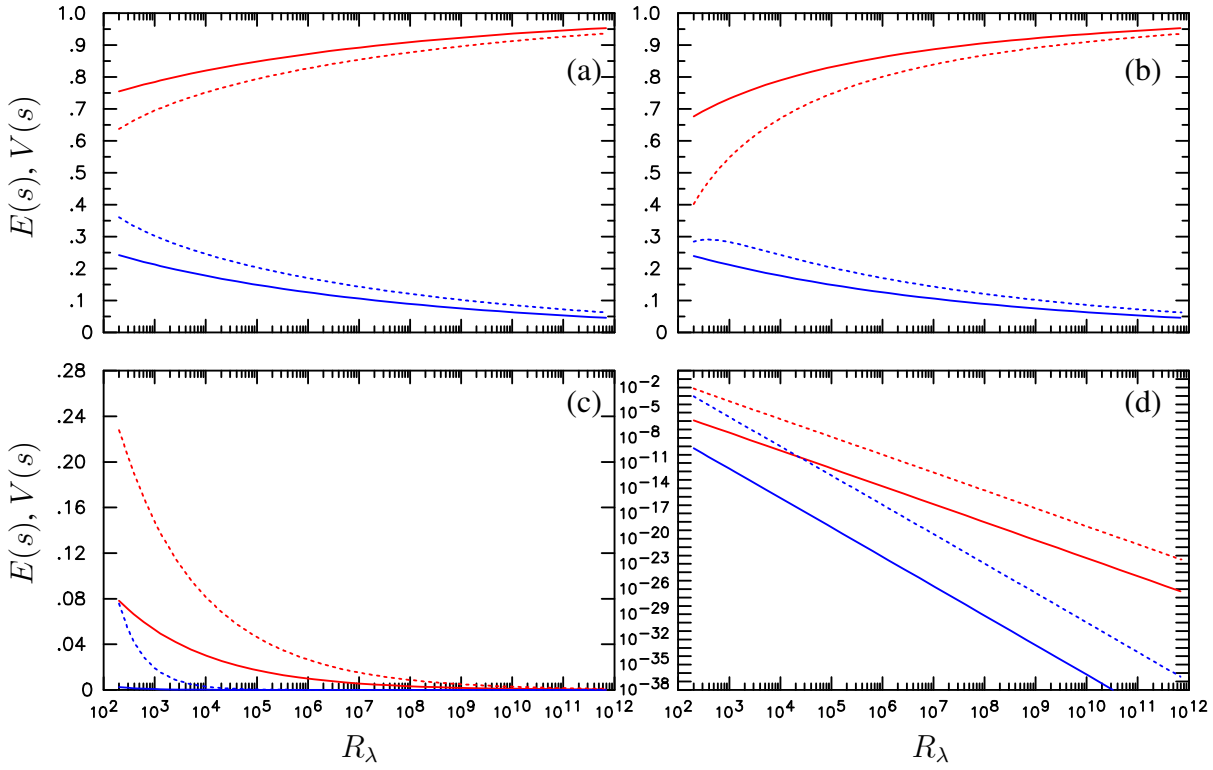

Figure 4.17: Fraction of energy dissipation contributed (red) and volume occupied (blue) by near-singular regions, forming the set S, of strength (a) $-\infty < \alpha < 3$, (b) $2.7 < \alpha < 3$, (c) $2.1 < \alpha < 2.7$ and (d) $-\infty < \alpha < 2.1$. The multifractal spectrum at $R_\lambda \sim 1300$ was used to compute these quantities at all R_λ. Near-singularities correspond to $r/\eta \sim 0.7$ (solid curves) and volume averaged energy dissipation (dotted curves, $r/\eta \sim 305$).

corresponding volume occupied decreases. That is, regions of small α (large dissipation), occupy small volumes because the fractal dimension ($f(\alpha)$) is largely negative while contributing most of the energy dissipation. Frames (b), (c) and (d) focus on different bands of α such that the sum of contributions from each of these three frames will add up to that shown in frame (a). Frame (b) shows that the majority of the energy dissipation is contained in the near-singular regions of strength $2.7 < \alpha < 3$ and occupies most of the volume, especially at higher R_λ. Singular regions of strength of $2.1 < \alpha < 2.7$ contribute some energy dissipation corresponding to lower values of R_λ while occupying very little volume in space (close to negligible when viewed on linear scales). In contrast to the other two regions, the region with α at 2.1 or below occupies little space and contributes almost negligibly to the total dissipation. In frame (d) shows both the energy dissipation contribu-

tion and volume occupied decrease with R_λ. This suggests these extremely high regions of energy dissipation do not contribute much to the total energy dissipation and occupy very small volumes; hence they are unimportant, especially for $\alpha < 2.1$. It may be noted that, as seen in frame (a), the rate at which $V(S)$ and $E(S)$ approach their high Reynolds number limits is very slow: it appears that asymptotic values are not truly reached at Reynolds numbers encountered in any terrestrial applications of interest.

Results for 3D averages at scale size $r/\eta \approx 305$ (representative of the inertial range) are shown as dashed lines in the figure. The figure shows that for the case when near-singular regions defined by $2.7 < \alpha < 3$ (frame (b)) are considered, the volume occupied is larger while the energy dissipation contained is lower than those for $r/\eta \approx 0.7$, i.e., the energy dissipation averaged over scale size representative of the inertial range is less dense compared to those representative of the dissipation range. But considering near-singularities corresponding to bands of α in frames (c) and (d), it can be seen that both the energy contributed and the volume occupied are greater than those for $r/\eta \approx 0.7$.

4.6 Contribution of samples below a threshold to moments of energy dissipation rate

Given the nature of dissipation as a highly intermittent quantity, it is useful to estimate how much of the total dissipation is derived from peaks of different sizes in space. The contribution to the overall moments of energy dissipation rate by values below a threshold, ϵ_t, are computed at different orders q and scale sizes r is given by

$$G(\epsilon_t/\langle\epsilon\rangle) = \frac{1}{\langle\epsilon_r^q\rangle} \int_0^{\epsilon_t/\langle\epsilon\rangle} \epsilon_r^q \, p(\epsilon_r/\langle\epsilon\rangle) \, d(\epsilon_r/\langle\epsilon\rangle) \tag{4.34}$$

where $p(\epsilon_r/\langle\epsilon\rangle)$ is the PDF of the normalized local average. It is clear that $G(0) = 0$ and $G(\infty) = 1$. Figure 4.18 shows these contributions for different orders q and for volume averages of different scale sizes. The SE-extrapolated PDFs were used in Equation 4.34 to compute this quantity. Data from a Gaussian distribution with the same mean and stan-

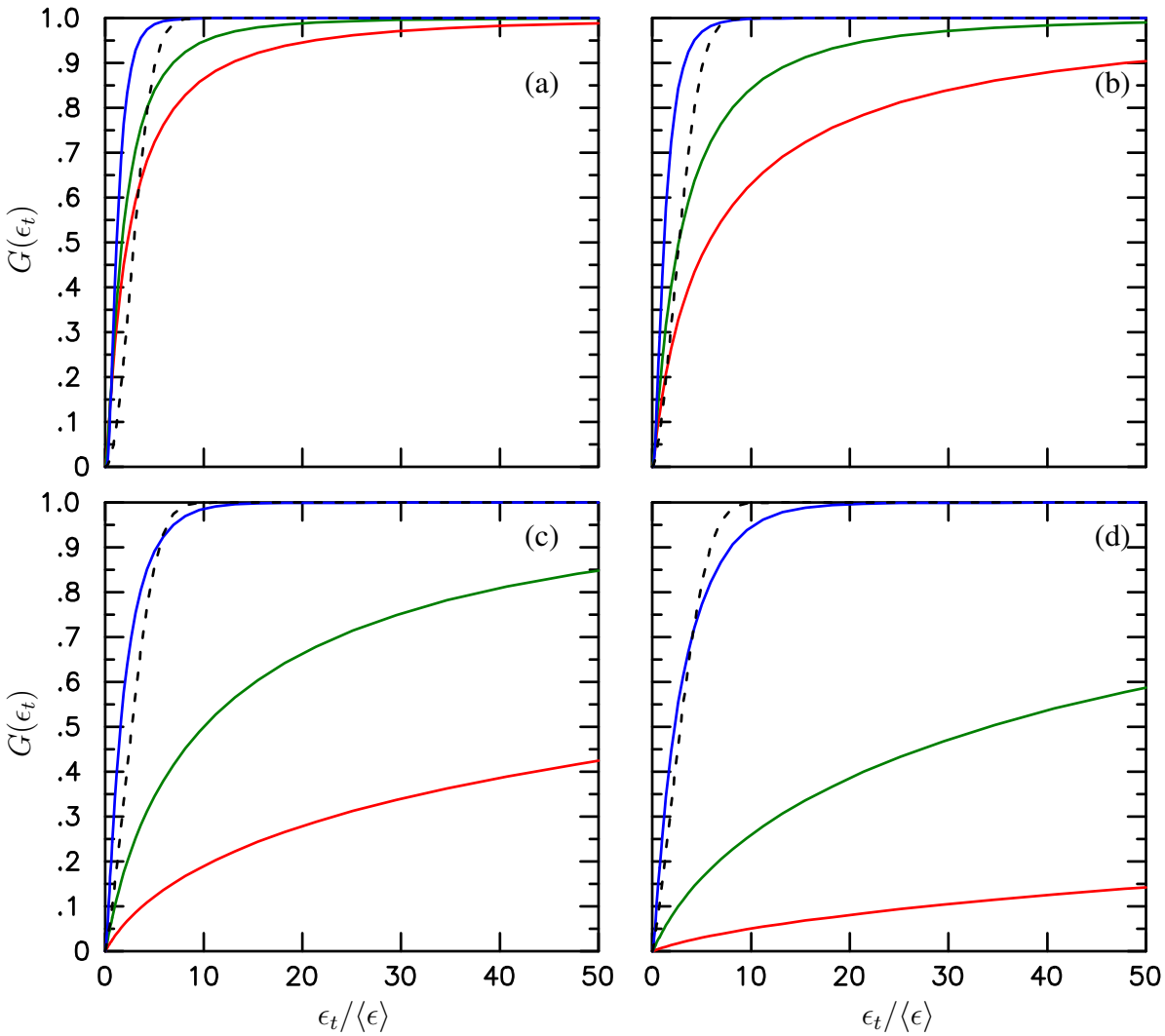

Figure 4.18: Contribution to moments of order (a) $q = 1$, (b) $q = 2$, (c) $q = 4$ and (d) $q = 6$ by samples below different values of threshold ϵ_t. Different curves correspond to local averages over volumes of size $r/\eta \approx 0.7$ (red), $r/\eta \approx 16$ (green) and $r/\eta \approx 305$ (blue). Data corresponding to a Gaussian distributed random variable with mean ($\mu_\epsilon = \langle\epsilon\rangle = 1.35$) and standard deviation ($\sigma_\epsilon = 2.3$) similar to that of the PDFs at $r/\eta \approx 0.7$ is shown by black dashed curves.

dard deviation as that of the PDFs at $r/\eta \approx 0.7$ (black dashed curves) are included for comparison.

Contributions to the mean value of energy dissipation rate are shown in frame (a). These first-order moments are dominated by samples of energy dissipation rates only as large as a few mean values. Specifically, for $r/\eta \approx 0.7$ (red curve), samples up to 50 times the mean value account for most of the dissipation. Data corresponding to r/η of 16 (green curve)

and 305 (blue curve) show that the contributions to the mean value are mostly dominated by samples less than 25 and 5 times the mean, respectively. The Gaussian is observed to return to the actual value by considering samples less than a threshold of 5 times the mean. Upon closer observation of frame (a), a crossover between the curve corresponding to the Gaussian and others can be observed. Since the Gaussian is based on the mean and standard deviation of $r/\eta \approx 0.7$ it is reasonable to only compare against the curves corresponding to this value of r/η, i.e., the red curve. The crossover between these two curves (red and black) can be interpreted as the actual PDFs showing probabilities higher than those shown by the Gaussian corresponding to sample values around the mean. However, since the tails are far from Gaussian more samples far away from the mean value need to be considered before the actual mean can be captured while the Gaussian quickly returns to its mean as its tail rolls off exponentially.

At higher orders, the figure shows that the thresholds need to be larger, especially for $r/\eta \approx 0.7$ (red curve), while the Gaussian consistently captures the actual moments considering samples less than 5 times the mean. For $q = 2$ shown in frame (b), approximately 90% of the contribution for $r/\eta \approx 0.7$ is dominated by flow regions with energy dissipation rate less than 50 times the mean. Contributions corresponding to higher r/η (green and blue curves) are dominated by samples of lower magnitudes, similar to that observed in frame (a). However, contributions from flow regions with higher energy dissipation rate dominate the higher order moments, as can be expected. Frame (c) and (d) show that only around 40% and 15% of the contributions come from samples less than a threshold of 50 times the mean for $r/\eta \approx 0.7$ (red curve) respectively. For volume averages over scale sizes typical of the inertial range (blue curve), contributions to moments as high as the sixth-order can be accurately captured by samples less than a threshold of 20 times the mean. It is also interesting to note that the crossover observed in frame (a) between the Gaussian and the contribution to moments for $r/\eta \approx 0.7$ is not observed at higher orders (frames (b), (c) and (d)) since these higher orders are mostly dominated by the tails of the PDFs and the

samples close to the mean do not contribute much.

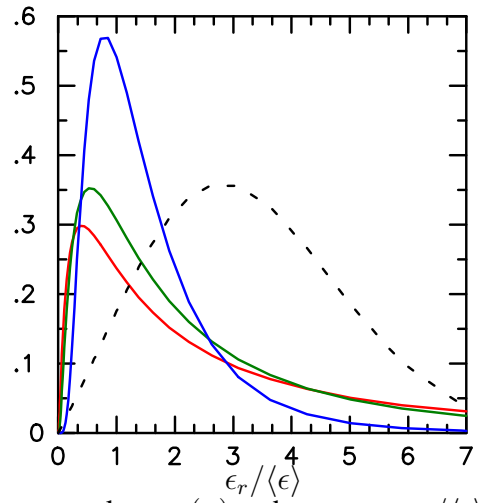

Figure 4.19: Plot of the integrand, $x\ p_r(x)$, where $x = \epsilon_r/\langle\epsilon\rangle$, observed in computation of the first moment from the PDFs of local averages of energy dissipation rate for $r/\eta =$ 0.7 (red), 16 (blue) and 306 (green) on linear scales. The black dashed curve shows the corresponding integrand from Gaussian PDF with mean ($\mu = \langle\epsilon\rangle = 1.35$) and standard deviation ($\sigma_\epsilon = 3.06$) same as that of the PDF at $r/\eta \approx 0.7$.

To understand the crossover observed in frame (a) of Figure 4.18 more clearly, the integrand in Equation 4.34 as a function of the sample value x is shown in Figure 4.19. As noted earlier, results from Gaussian and PDF at $r/\eta \approx 0.7$ are compared as they correspond to the same mean and standard deviation. The figure shows that for small $\epsilon_r/\langle\epsilon\rangle$ the integrand increases quickly for the PDFs compared to the Gaussian, leading to a greater contribution to the moments. However, as sample values increase, the integrand for the Gaussian continues to grow while those for $r/\eta \approx 0.7$ (red curve) drops. This leads to the observed crossover in Figure 4.18.

Table 4.2 shows the same data for chosen values of the thresholds normalized by standard deviations, instead of mean values as before, at the three different scale sizes. The table shows that close to 90% of the mean energy dissipation rate can be recovered, considering samples less than a threshold of 5 standard deviations. For larger scale sizes, more than 90% of the mean can be recovered. When considering higher order moments, it is clear that as the scale size increases, more of the actual value can be recovered for a fixed value of the threshold. For third and fourth order moments, thresholds as high as 10

Table 4.2: Percentage of actual value recovered for different values of thresholds (in units of standard deviations) for first, second, fourth and sixth order moments computed using PDFs with stretched exponential extrapolation. The standard deviations (σ_{ϵ_r}) corresponding to $r/\eta = 0.7$, 16 and 305 are 3.06, 2.05 and 1.03 respectively.

r/η	$q=1$				$q=2$				$q=4$				$q=6$			
	$\epsilon_t/\sigma_{\epsilon_r}$				$\epsilon_t/\sigma_{\epsilon_r}$				$\epsilon_t/\sigma_{\epsilon_r}$				$\epsilon_t/\sigma_{\epsilon_r}$			
	1	3	5	10	1	3	5	10	1	3	5	10	1	3	5	10
0.7	50	76	89	95	30	51	66	79	7	14	20	29	2	3	5	8
16	42	80	90	97	32	64	76	89	13	31	42	57	6	15	20	30
305	24	83	96	100	29	79	92	99	24	64	81	95	19	50	67	87

standard deviations can only capture 30% and 10% of the total contribution respectively, which implies larger values of energy dissipation rate dominate these moments as can be expected.

4.7 Conclusions

An analysis undertaken from a multifractal perspective has been presented in this chapter, using primarily three-dimensional local averages of the energy dissipation rate obtained from direct numerical simulations (DNS) at high Reynolds number with good small-scale resolution. In contrast to past data, especially from experiments, the full energy dissipation rate (including all nine velocity gradients) computed from DNS data is used to compute the multifractal spectrum. The effect of Reynolds number, extrapolation of PDFs of local averages of energy dissipation, as well as the contributions of different ranges of the dissipation rate magnitude to moments of the 3D local averages have been examined.

Higher-order moments of the dissipation rate averaged over scales sizes in the dissipation range are sensitive to issues of statistical convergence associated with finite sampling. A similar, and somewhat more prominent, issue is observed for high-order negative moments, as well. Extrapolation of the PDF tails has been considered as a possible means of alleviating the impact of statistical noise at extremely small or extremely large sample values. A stretched exponential with variable stretching parameter appears to be the best functional form compared to several others. Nonlinear curve fits are performed for scale

sizes ranging from dissipation to inertial range scales to estimate the parameters in the stretched exponential model.

Moments of energy dissipation rate in the form necessary to compute the power-law scaling exponents D_q, also known as generalized dimensions, are analyzed. Clear power-law scaling behavior is observed for a wide range of moments. The generalized dimensions were computed by performing least square fits using a scaling range defined based on either the classical Kolmogorov length scale or an order-dependent generalized Kolmogorov length scale. Results showed very little difference between the two, at least for moments at order $q \leq \pm 8$. Accordingly, a fixed scaling range $60 \leq r/\eta \leq 600$ proposed in recent work is used to obtain generalized dimensions and other multifractal quantities in the inertial range. The multifractal spectrum ($f(\alpha)$), where α characterizes the strength of the near-singularities, is shown to give negative values corresponding to regions of high dissipation, especially if calculated from extrapolated dissipation PDFs. Comparing the spectrum from different Reynolds numbers showed that higher R_λ leads to lower values of α and $f(\alpha)$. This result can be interpreted as a greater incidence of near-singularities with lower or negative fractal dimensions in high Reynolds number turbulence with increased intermittency. The multifractal spectrum computed from high resolution DNS data in this thesis are found to be in relatively good agreement with Meneveau & Sreenivasan (1991), especially for the lower values of Reynolds number. However, it is important to isolate the effects of the differences in methodology between data obtained from laboratory experiments, where 1D surrogacy, Taylor's frozen turbulence hypothesis and 1D averages were used, in contrast to DNS, where 3D averages of the full energy dissipation rate are now available.

The multifractal spectrum was used to compute the intermittency exponent (μ_q) which suggested a value of 0.22 for the second order moments ($q = 2$) of local averages of energy dissipation, similar to that observed in section 3.2. The multifractal spectrum was also used to estimate the energy dissipation contributed and volume occupied by the near-singularities of different strengths. A major fraction of the total energy dissipation comes

from near-singular regions of strength, $2.7 \leq \alpha \leq 3$ while the volume occupied by these regions is small. Stronger near-singular regions contribute very little to the energy dissipation while occupying very small volumes in space. It was also found that as R_λ increases, singularities of the same strength contribute more to the total energy dissipation while occupying even smaller volumes as the R_λ is increased. However, the curves did not converge to values corresponding to a full singularity, i.e., all the energy dissipation being contributed by singular regions of strength $\alpha < 3$ while occupying zero volume, even for R_λ as high as 10^{12}, which exceeds the Reynolds number expected in all terrestrial applications.

Finally, the relative contributions of samples of different values to moments of different orders are estimated for local averages of energy dissipation rate. The results show that the mean value receives most of the contributions from samples less than a few times the mean while samples of larger values contribute a more significant fraction to the higher order moments. As the local averaging scale size increases, samples of lower values have a more significant contribution to the moments at a fixed order of the moment.

In summary, the work presented in this chapter provides results from a multifractal viewpoint using state-of-the-art numerical simulation data. Specifically, results on the behavior of PDF tails, the multifractal spectrum, intermittency exponents, singularities and contribution of different values of energy dissipation rate to its overall moments are presented, from well resolved DNS data at Reynolds numbers rivaling those in the best laboratory facilities at this time. The present results provide a significant extension to past work on multifractality in turbulence, in part through avoidance of one-dimensional surrogacy (versus the full energy dissipation) 1D versus 3D local averaging in space, and the Taylor's frozen turbulence hypothesis necessary in many laboratory settings.

CHAPTER 5
HIGH RESOLUTION STUDIES OF INTERMITTENCY IN SCALAR DISSIPATION RATES

Turbulence is an efficient agent of mixing of substances or properties known as scalars (Warhaft, 2000; Sreenivasan, 1991*b*). When the concentration of these substances being mixed are very low such that it does not affect the flow itself, they are called passive scalars. It is generally known (Gotoh & Yeung, 2013) that passive scalar fields of Schmidt number of order unity or higher exhibit stronger intermittency in the small scales than does the velocity field. Extreme values of the scalar dissipation rate, are directly related to large scalar gradients, which can be difficult to resolve and sample. Incorporation of passive scalar fields also adds to the cost of the numerical simulations, making long and well-resolved simulations even more difficult. As a result, the MRIS approach discussed for the velocity field in chapter 3 is appealing for the investigation of passive scalar intermittency.

As for the velocity field, simulations where the degree of resolution of small scales, reflected by the parameter $k_{max}\eta$, is higher than 4 are performed. These simulations will provide detailed information on fluctuations of scalar dissipation rate, which could not be resolved adequately in previous work. The new data will provide a new and much-needed test concerning whether prior observations of departures of the scalar field from the classical theory of local isotropy may have been contaminated by insufficient resolution.

The highly intermittent nature of scalar dissipation rates raises key questions regarding the refined similarity hypothesis of passive scalars. The moments of 3D local averages of scalar dissipation rate are important to understanding this. However, the difficulty in computing such quantities has rendered such results hard to come by. This chapter will present some results comparing the moments of 3D local averages of energy and scalar dissipation rate at different resolutions and Reynolds numbers studied. The relationship

between energy and scalar dissipation rates at different scale sizes will be studied through conditional moments as well.

A brief overview of the numerical approach to simulate passive scalars is made in section 5.1. Basic simulation parameters for the MRIS technique and single point statistics of scalar dissipation rates and scalar gradients are given in section 5.2. Results on the volume averages of scalar dissipation rate of passive scalars are presented in section 5.4. Conclusions are summarized in section 5.5.

5.1 Numerical approach

Turbulent mixing being a small-scale process, it is appropriate to study its fundamentals in incompressible isotropic turbulence as well. Numerically, scalar fluctuations can be sustained by an isotropic forcing mechanism at the large scales (analogous to $\mathbf{f}$ in Equation 2.1), or by a uniform mean gradient. The latter is chosen here, since it allows to study the question whether the scalar field satisfies local isotropy when the mean gradient imposes a preferential direction. In addition to solving for the velocity fluctuations (Equation 2.1), the scalar fluctuations (ϕ) are also computed from its advection-diffusion transport equation of the form,

$$\partial\phi/\partial t + \mathbf{u}\cdot\nabla\phi = -\mathbf{u}\cdot\nabla\Phi + D\nabla^2\phi \tag{5.1}$$

where D is the molecular diffusivity of the passive scalar and $\nabla\Phi$ is a uniform mean scalar gradient, which (along with the velocity fluctuations) is responsible for generating the scalar fluctuations and preserves homogeneity.

The Fourier transform of Equation 5.1 are taken, where the scalar Fourier coefficient $\hat{\phi}(\mathbf{k},t)$ satisfies,

$$\partial\hat{\phi}/\partial t + \widehat{\nabla\cdot\mathbf{u}\phi} = -\hat{\mathbf{u}}\nabla\Phi - Dk^2\hat{\phi} \tag{5.2}$$

Integration in time is performed using the RK2 scheme similar to that used for the velocity

fluctuations. Multiple scalars that are both dynamically passive and chemically inert may be incorporated by simply writing separate equations in the same form as Equation 5.2, say with different molecular diffusivities.

Computing the scalar fluctuations using the Fourier pseudo-spectral framework (Canuto *et al.*, 1988) is fairly straightforward. The beginning of each time step in the DNS code starts with velocity and scalar fluctuations in the wavenumber space. There will be a total of $3 + N_c$ flow variables (3 for the velocity, N_c being the number of scalars). The velocity and scalar Fourier coefficients along with the y-derivatives of scalar fluctuations are transformed from wavenumber to physical space in y direction alone ($3 + 2N_c$ transforms). Next, the z and x derivatives for the scalars are computed. Finally, the velocity fluctuations and the three derivatives ($3 + 3N_c$ transforms) of the scalars are transformed to physical space, where the nonlinear products are formed. This approach allows for statistics of the scalar gradients to be computed readily in physical space. In addition to the nonlinear products necessary for the momentum equation, the products of velocity and scalar derivatives, $u_i d\phi/dx_i$'s, are formed. The mean scalar gradient term is also accounted for in physical space. These nonlinear products along with the mean gradient term, only 1 for each scalar, are then transformed from physical space back to wavenumber space before the integration in time can be performed. The diffusion term is treated exactly through an integrating factor similar to the viscous term in the momentum equation.

Pseudo-spectral direct numerical simulations using the approach described here have been used for many decades (Gotoh & Yeung, 2013). Majority of these simulations used a massive CPU-based parallel implementation, as in Donzis & Yeung (2010). While more recent work (Clay *et al.*, 2017) focusing on high Schmidt number scalars have used a multi-grid approach to account for the difference in size of small-scales in the velocity and scalar fields and have been extended to use OpenMP to target GPUs in Clay *et al.* (2018). In this work, the batched asynchronous code, described in chapter 2, has been extended to include passive scalars. This enables simulations are very large problem sizes,

unattainable previously. However, due to limited computational resources on Summit, results from simulations of passive scalars, using the CPU-based pencil decomposition code, up to Reynolds numbers 650 using a grid of 6144^3 performed on Frontera supercomputer at Texas Advanced Computing Center (TACC) are presented in this thesis. CUDA kernels to compute the derivatives of scalars in each of the three directions are written, and the kernel to compute the nonlinear products in physical space is extended to include the formation of velocity and scalar derivative products relevant to the advection-diffusion equation. Additional memory requirements due to the presence of scalars are accounted for by increasing the number of batches per slab (4 simulations with up to 2 scalars versus 3 for velocity field only). Additional cost for including passive scalars is about 40% per scalar at each time step.

5.2 Description of DNS database

Table 5.1: Basic parameters for simulations conducted at different R_λ using the MRIS approach along with the second order moments of energy and scalar dissipation rate and scalar gradient skewness. For each Schmidt number mean gradients in all three directions are considered, i.e., the $R_\lambda = 390$ and 650 simulations have 6 and 3 scalars in total. All simulations were performed using a Courant number of 0.3.

R_λ	N	$k_{max}\eta$	Sc		β	M	$\langle\epsilon^2\rangle/\langle\epsilon\rangle^2$	$\langle\chi^2\rangle/\langle\chi\rangle^2$		$\mu_3(\nabla_{\|\|}\phi)$	
			Sc_1	Sc_2				Sc_1	Sc_2	Sc_1	Sc_2
390	1024	1.4	1	0.125	2	11	4.19	13.42	11.99	1.24	1.85
390	1536	2.1	1	0.125	2	11	4.35	15.32	11.95	1.36	1.85
390	3072	4.2	1	0.125	2	11	4.39	15.77	11.90	1.38	1.84
650	2048	1.4	1	-	2	21	4.62	16.90	-	1.19	-
650	3072	2.1	1	-	2	21	4.88	19.70	-	1.32	-
650	6144	4.2	1	-	1	21	4.98	20.52	-	1.35	-

The MRIS approach has been used to conduct simulations of passive scalar mixing at a combination of Reynolds numbers and Schmidt numbers, as shown in Table 5.1. As in section 3.2, in each case, a short segment at the lowest resolution is run after reducing the Courant number to 0.3 (from 0.6 which was used to generate the initial snapshots). Subsequently, the grid spacing is refined twice, until the resolution parameter $k_{max}\eta$ reaches

4.2 (such that $\Delta x/\eta \approx 0.7$). Because of present limitations on resource allocations on Summit, the simulations reported in Table 5.1 have been limited to 6144^3 in size, and were conducted instead using a CPU code on the Frontera supercomputer at the Texas Advanced Computing Center (TACC). As was the case for simulations of the velocity field (chapter 3), statistics of the small scales are found to adjust to increases in resolution quickly, well within 1 Kolmogorov time scale for every halving of the grid spacing.

In principle, since the turbulence is isotropic the mean scalar gradient can be in any direction, say one of the coordinate axes, such that the mean gradient term in Equation 5.1 is coupled to any one of the three velocity components (u, v, w). However, in practice, especially if sampled over only a short period of time, the r.m.s values of u, v, w can differ significantly. This introduces statistical variability into results on the scalar fluctuations. This issue can be addressed by (i) a larger domain with more grid points to allow better sampling of the large scales in the velocity field while keeping the grid spacing fixed, or (ii) simulating multiple scalars with $\nabla\Phi$ in different directions and taking ensemble averages over them (Buaria *et al.*, 2016). The second approach is preferable because of lower computational cost compared to the first, which would have required 8 times as many grid points.

Similar to the studies of intermittency in chapter 3, an important measure for the passive scalar intermittency is the fluctuating scalar dissipation rate χ defined as,

$$\chi = 2D\frac{\partial\phi}{\partial x_i}\frac{\partial\phi}{\partial x_i} \tag{5.3}$$

which is a quadratic measure of the scalar gradient fluctuations. Second-moment data given in Table 5.1 indicates the second moment of the normalized scalar dissipation is some 3–4 times larger than for the energy dissipation. It is apparent that the these moments for χ are 3-4 times higher in magnitude compared to ϵ. This is consistent, at least at the second moment levels, with claims (Overholt & Pope, 1996; Yeung *et al.*, 2005) that the scalar

dissipation is more intermittent than the energy dissipation. It can also be seen that a scalar of $Sc = 1$ has larger high-order moments than that of $Sc = 0.125$, which is not a surprise, because higher molecular diffusivity in the scalar of low Sc tends to smooth out the scalar gradients before they can grow to larger magnitudes.

It is also important to note the variability of these second-order moments with respect to the spatial resolution. For R_λ 650, when the resolution is improved from $k_{max}\eta$ 2.1 to 4.2, the moments of ϵ increase by 2% while the second moments of χ increase by 4% suggesting further improvements in resolution will have a greater impact on the accuracy of higher-order statistics of the scalar dissipation rate. The increase in these moments with Reynolds number (if computed at same $k_{max}\eta$) is also much stronger for χ than ϵ. This further supports the argument that the scalar field is more intermittent, especially at higher R_λ.

Given the presence of the mean scalar gradient in the equations above (and their counterpart in laboratory experiments (Mydlarski & Warhaft, 1998; Warhaft, 2000)) it is important to explain observed departures from local isotropy, as seen in the skewness and other higher odd-order moments of the gradient $\nabla_{||}\phi$ parallel to $\nabla\Phi$ (Donzis & Yeung, 2010; Sreenivasan, 1991*b*). Classical Kolmogorov phenomenology suggests the small scales of turbulence become increasingly isotropic at high R_λ. However, the small scales of the passive scalar have been widely observed (Warhaft, 2000; Gotoh & Yeung, 2013) to deviate strongly from local isotropy in the presence of a mean scalar gradient. A convenient diagnostic is the skewness of the mean scalar gradient parallel to the mean gradient, $\mu_3(\nabla_{||}\phi)$ which is computed for the data sets studied here and reported in Table 5.1. For the R_λ 390 simulations, as the Sc number increases, the scalar gradient skewness is observed to increase as well. Increasing spatial resolution helps capture the higher magnitude scalar gradients more accurately can lead to an increase in the gradient skewness. Because of some partial cancellation between positive and negative values of the scalar gradient, this increase may, however, be weaker than that in even order moments. In Table 5.1 the skew-

ness also appears to decrease slightly with Reynolds number, but the increase is less significant when resolution is high These trends are consistent with results in Donzis & Yeung (2010) and highlight the need for high resolution simulations for an accurate study of this topic.

5.3 Single point statistics

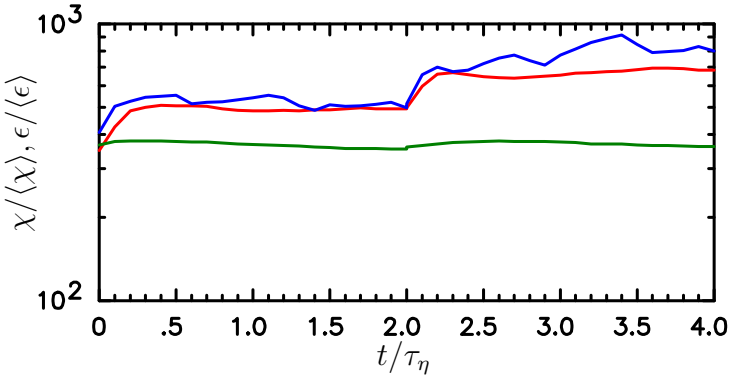

Figure 5.1: Ensemble averaged peak energy dissipation rate (blue) and scalar dissipation rate for $Sc = 1$ (red) and $Sc = 0.125$ (green) from the $R_\lambda \sim 390$ simulation.

The evolution of the peak normalized energy and scalar dissipation rates at $R_\lambda \sim 390$, over a period of 4 τ_η with resolution increases from $k_{max}\eta$ 1.4 to 2.1 and 2.1 to 4.2 at $t/\tau_\eta = 0$ and 2 respectively are shown in Figure 5.1. Ensemble averaging over 11 realizations (see Table 5.1) has been applied. The scalar at $Sc = 0.125$, being of higher molecular diffusivity, is, of course better resolved; it is not surprising that results for this scalar show little sensitivity to further improvements in grid resolution. Results for the scalar at $Sc = 1$ display sensitivities to resolution that are qualitatively similar to those seen for the energy dissipation rate. The figure further shows that the peak values adjust rapidly, within $0.5\tau_\eta$, to the higher resolution grid. This re-affirms the applicability of the MRIS approach to studies of fine-scale intermittency in passive scalar fields. It is interesting to note that the peak values in both χ (for $Sc = 1$) and ϵ are comparable in amplitude, with the energy dissipation being slightly higher.

The probability distribution of the scalar dissipation rate allows to quantify the likeli-

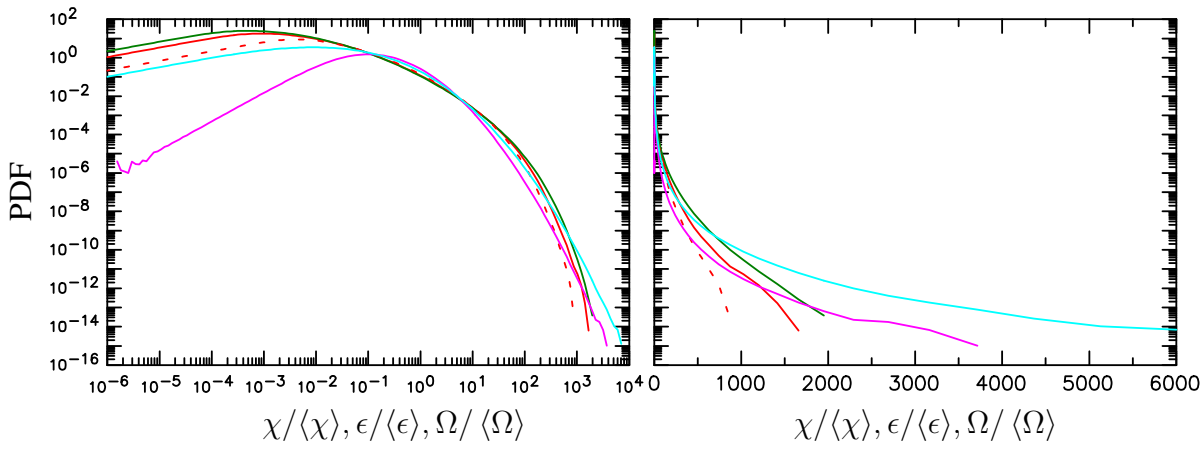

Figure 5.2: PDFs of normalized scalar dissipation rate on log-log (left) and log-linear (right) scales for $Sc = 1$ (solid lines) and $Sc = 0.125$ (dashed line) at R_λ 390 (red) and R_λ 650 (green) at $k_{max}\eta \approx 4.2$. PDFs of normalized energy dissipation and enstrophy are shown in magenta and cyan, respectively.

hood of events of different amplitudes, ranging from the very small to very high. Figure 5.2 shows the PDFs of scalar dissipation rate for $Sc = 1$ at different R_λ for $k_{max}\eta \approx 4.2$. It also shows the PDF of χ for $Sc = 0.125$ scalar at R_λ 390, ϵ and Ω at R_λ 650. The frame on the left is helpful for focusing on the small magnitude events. Yeung *et al.* (2012) argued that the behavior of velocity gradients at small sample values is similar to that of a Gaussian Random Field (GRF). This implies, enstrophy being the sum of squares of three velocity gradients, will have a chi-square distribution of order three at these small sample values. This corresponds to a power-law as observed in the PDFs. Similarly, the scalar gradient PDFs also show a Gaussian behavior corresponding to small sample values. Therefore, scalar dissipation rate, which is the sum of squares of the three Gaussian scalar gradients will lead to a chi-square distribution of order three. This explains the observed that the slope of the tails corresponding to small values for enstrophy and scalar dissipation are similar. The slope is different for energy dissipation as it behaves as a chi-square of order five due to incompressibility as described in Yeung *et al.* (2012). It can also be seen that as the R_λ increases, the probability of these low magnitude events also increase.

The right frame in Figure 5.2 provides greater clarity in the regime of extreme events of high amplitude. It is clear that the tails of enstrophy are widest, followed by the tails of

energy dissipation rate. Consistent with the lower peak values observed in Figure 5.1, the scalar dissipation tails are narrower than those for ϵ. The PDFs of χ become wider with increasing R_λ which is consistent with increased intermittency.

Another interesting observation in Figure 5.2 is that even though the extreme events in χ are not as strong and frequent as those in ϵ, at more moderate values, say between 10 and 1000, the probability of such events occurring is much higher for χ than ϵ. This suggests that the higher values for the second-order moments observed in Table 5.1 are mostly due to contributions from the moderately large values of χ. For higher order moments, it is expected that the trend will reverse. The cross-over observed between the PDFs of χ and ϵ warrants a more detailed analysis.

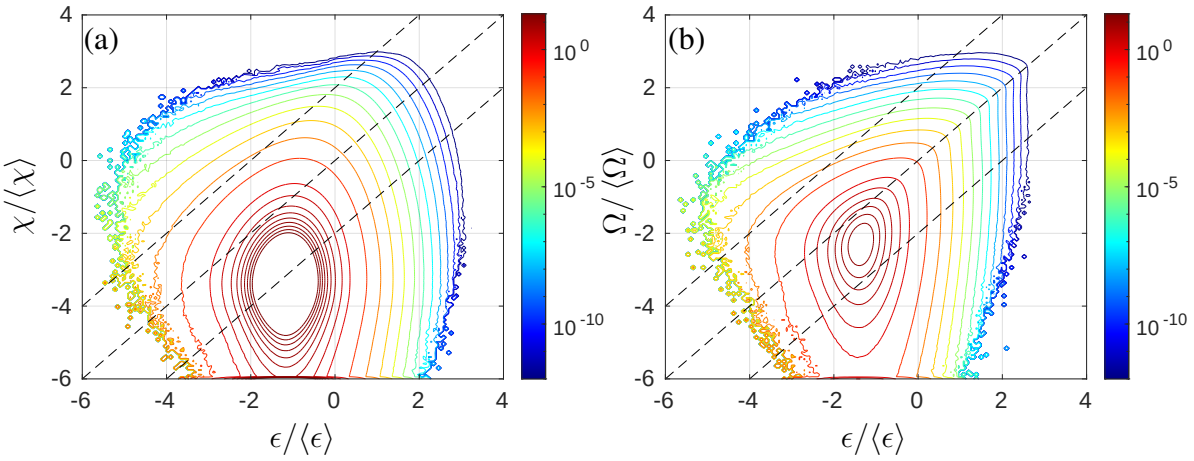

Figure 5.3: Ensemble averaged joint PDFs of (a) scalar and energy dissipation rate and (b) enstrophy and energy dissipation rate on log-log scales from simulations at $R_\lambda \sim 650$ and $k_{max}\eta = 4.2$. The contour levels are logarithmically spaced, as shown by the color bar.

Statistical interrelationships between energy and scalar dissipation rates can be explored using their joint PDF, which is compared with the joint PDF of dissipation and enstrophy (in log-log scales) in Figure 5.3. Outermost contour lines in the first quadrant corresponding to large values of the variables concerned have a more rounded appearance in frame (a) in contrast to that in frame (b). This suggests the extreme events of scalar dissipation and energy dissipation become less tightly coupled to each other while enstrophy and energy dissipation evolve with each other, indicated by the fact that the contours are stretched

along the diagonal.

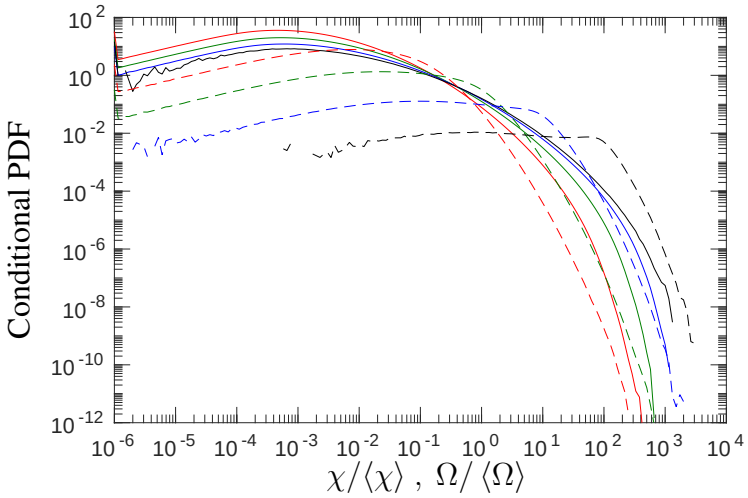

Figure 5.4: Conditional PDFs of scalar dissipation given energy dissipation (solid lines) and enstrophy given energy dissipation (dashed lines) from simulations at $R_\lambda \sim 650$ and $k_{max}\eta = 4.2$. Different colours correspond to different values of the conditioning variable: $\epsilon/\langle\epsilon\rangle = 0.1$ (red), 1 (green), 10 (blue) and 95 (black).

A more precise assessment of the degree of independence between two random variables can be made by inquiring whether the conditional PDF of one variable given the other is the same for different choices of the conditioning variable. Conditional PDFs of scalar dissipation given energy dissipation, as well as enstrophy given energy dissipation extracted from joint PDFs are compared in Figure 5.4. The behaviors seen in this figure differ significantly between low and high values of each variable. For example, for $\chi/\langle\chi\rangle$ up to 1, the spacings between solid lines of different colors are quite small, suggesting that low χ can occur with comparable probability in regions of low ϵ as well as high ϵ. In contrast, the relative positions of dashed lines of different colors indicate that low Ω becomes increasingly unlikely as the conditioning ϵ becomes large. At the other end, curves corresponding to high-intensity extreme events in χ or Ω behave quite differently. If the conditioning ϵ is low (lines in red) χ has wider tails than Ω. But if the conditioning ϵ is high (lines in black) then Ω has considerably wider tails. This means high ϵ has a strong likelihood to lead to high Ω but its effect on the incidence of high χ is much weaker.

As noted earlier, departure of scalars from local isotropy is an important issue which

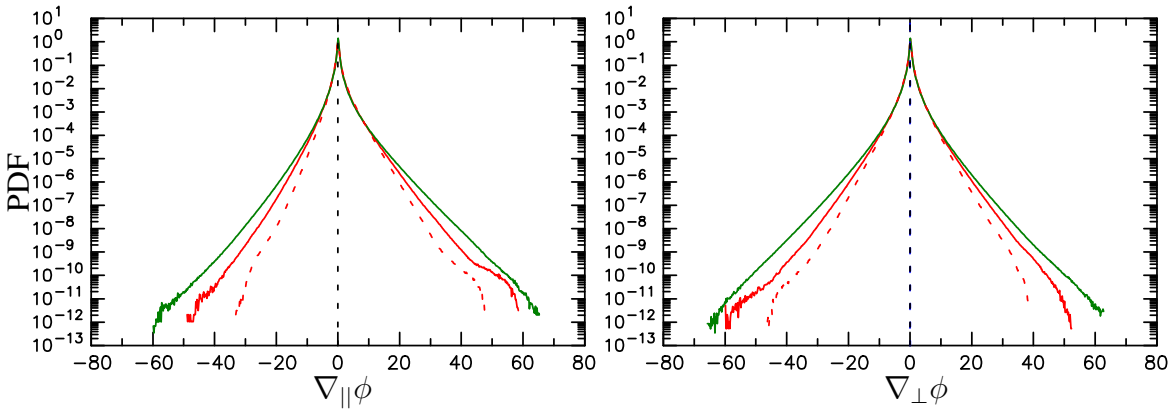

Figure 5.5: PDFs of scalar gradients parallel (left) and perpendicular (right) to the mean gradient at $R_\lambda \approx 390$ (red) and 650 (green). The solid curve corresponds to $Sc = 1$ and dashed curve to $Sc = 0.125$.

can be investigated through the gradient skewness, which in turn implies some degree of asymmetry in the PDF of the scalar gradient. Figure 5.5 shows PDFs of scalar gradients both parallel and perpendicular to the imposed mean gradient, at two different Reynolds numbers. The latter is close to symmetric, as expected. However, the PDF of the parallel component shows a mild degree of skewness to the right, which is consistent with a positive skewness. As Reynolds number increases, the the tails on both sides become wider, possibly widening slightly more on the left versus the right. This observation is consistent with trends seen for the scalar gradient skewness in Table 5.1. However, since the effect appears to be weak, a higher Reynolds number as well as better sampling are necessary to confirm this result.

5.4 Study of intermittency using 3D local averages of scalar dissipation rate

This section presents a brief study of the refined similarity hypothesis as applied to passive scalars. Moments of locally averaged scalar dissipation rate are computed and analysed for a power-law behavior in the inertial range. Conditional moments are used to investigate the statistical connections between locally averaged energy and scalar dissipation rates.

Similar to the case for energy dissipation rate discussed at length in chapter 3, the

locally averaged scalar dissipation,

$$\chi_r(\mathbf{x}) = \frac{1}{Vol}\int_r \chi(\mathbf{x}+\mathbf{r})d\mathbf{r} \tag{5.4}$$

is an important quantity in the study of passive scalar intermittency. It has been computed in the same manner as described earlier for ϵ_r. Normalized moments of 3D local averages and corresponding characteristic local slopes ($d\ln\langle\chi_r^q\rangle/dr$) of scalar dissipation rate for $Sc = 1$ are shown in Figure 5.6. Frames on the left and right show the sensitivity of the data to spatial resolution and Reynolds number respectively, including comparisons with corresponding results for energy dissipation and enstrophy.

In frame (a) of Figure 5.6 lines in red, green, blue are almost coincident, showing that the second moment of χ_r is sufficiently well-resolved at least with $k_{max}\eta \sim 4.2$, for most values of r/η. Single-point moments approximated by χ_r data at the smallest possible r/η (with $r = \Delta x$) are highest for χ, followed by Ω and then by ϵ. Although the curves in (a) appear to be reaching a plateau as $r \to 0$ the logarithmic local slopes, which would be scaling exponents if well-defined power-laws were to apply, do show some significant departures from zero at both second and fourth moments (frames (c) and (e), respectively). However, results for scale sizes in the inertial range are well-converged as far as resolution is concerned. While attainment of flat scaling ranges is not perfect, a comparison between frames (c) and (e) show that the scalar dissipation has a larger scaling exponent than both energy dissipation and enstrophy at second order but the difference is significantly smaller at fourth order.

In frames (b), (d) and (f), although the Reynolds number range reported (R_λ 390 and 650) is somewhat limited, it can been seen clearly that a high Reynolds number leads to an increase in the moments of χ_r, as well as (more importantly) a significantly better-defined inertial-convective scaling range. The latter observation is also encouraging for future simulations of yet higher Reynolds number. It should be noted that results at low

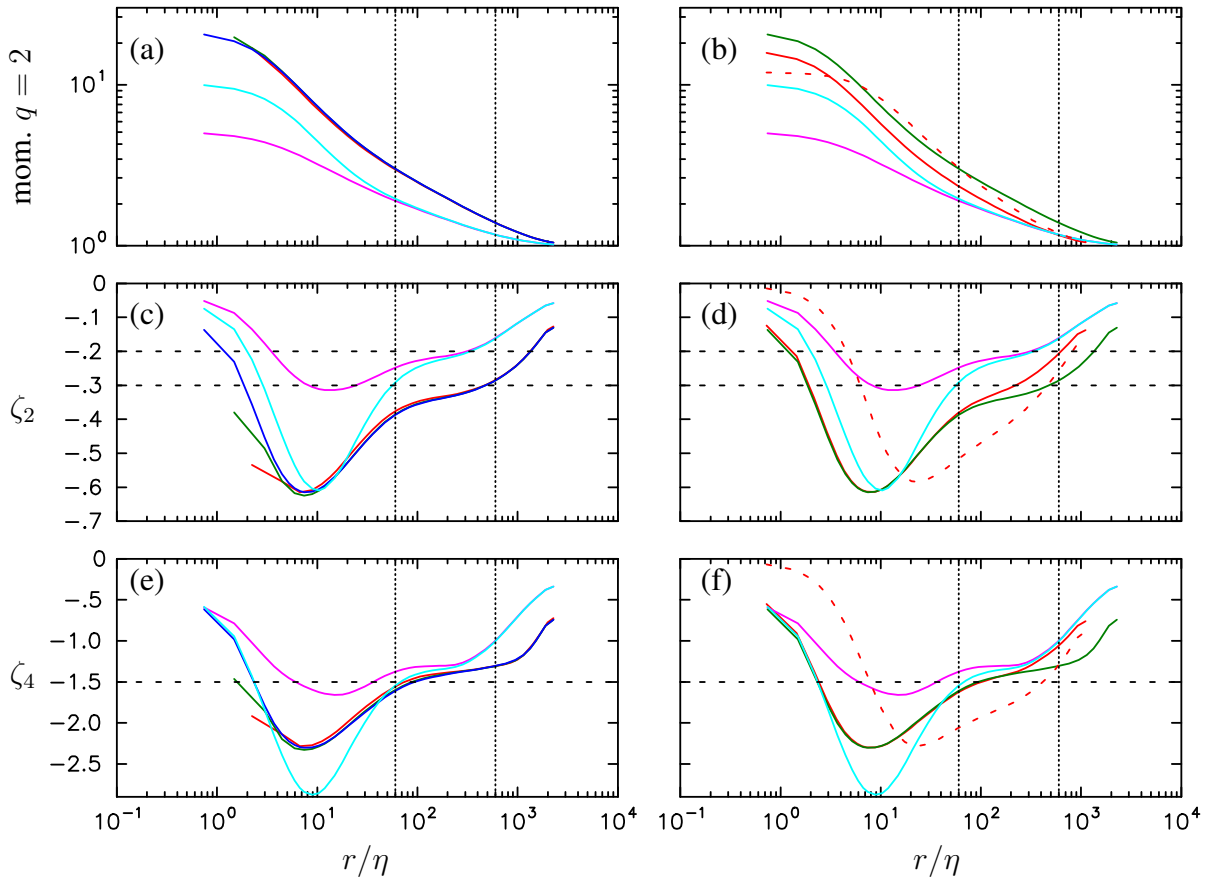

Figure 5.6: Ensemble averaged normalized second-order (a,b) moments of 3D local averages of scalar dissipation rate (of $Sc = 1$), their logarithmic local slopes (c,d) and the logarithmic local slopes of fourth-order moments (e,f). Results from (left) simulations at $R_\lambda \sim 650$ for $k_{max}\eta$ 1.4 (red), 2.1 (green) and 4.2 (blue) and (right) from simulations at $k_{max}\eta \sim 4.2$ for R_λ 390 (red) and 650 (green). The red dashed curve corresponds to moments of scalar with $Sc = 0.125$. The magenta and cyan curves correspond to moments of energy dissipation and enstrophy respectively from $R_\lambda \sim 650$ and $k_{max}\eta \sim 4.2$ for comparison. Vertical dashed lines mark the inertial range, $60 < r/\eta < 600$.

Schmidt number $Sc = 0.125$ also differ significantly. In the dissipation range the moments of χ_r are smaller than those for $Sc = 1$ and very well resolved, as can be expected from higher molecular diffusivity. However, at intermediate scale sizes this scalar gives larger local slopes while development of a proper inertial-convective scaling range (at this low Sc) may require a considerably higher Reynolds number than currently available.

Another important question is how, and to what degree, samples of locally averaged energy dissipation, enstrophy and scalar dissipation are related to each other. As in Yeung

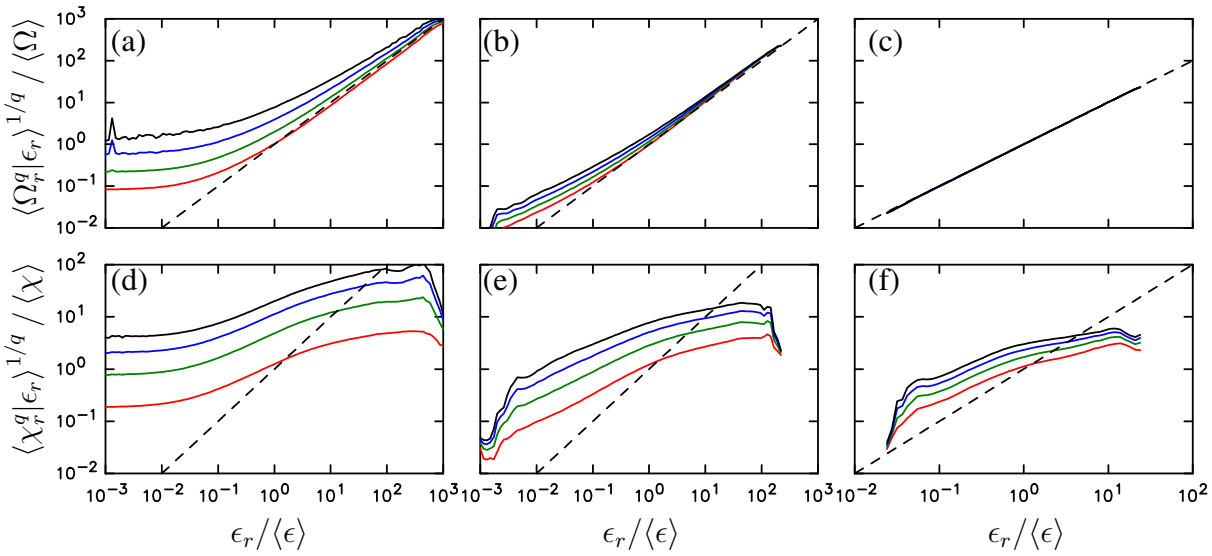

Figure 5.7: Ensemble averaged conditional moments of scalar dissipation of $Sc = 1$ given energy dissipation rate (d,e,f) at R_λ 650. For comparison the conditional moments of enstrophy given energy dissipation rate are shown in frames a,b and c. The three columns correspond to a scale size (r/η) of 0.7 (a and d, single point statistics), 11 (b and e) and 90 (c and f). The curves of different colours correspond to moments of order 1 (red), 2 (green), 3 (blue) and 4 (black). Dashed line indicates the two variables scaling perfectly with each other.

& Ravikumar (2020) and Vedula *et al.* (2001) it is useful to study conditional expectations at various orders, such as (for passive scalars) the quantity $\langle \chi_r^q|\epsilon_r\rangle$. Figure 5.7 shows data for these quantities, at R_λ 650 and $Sc = 1$. The q-th root of these moments are taken to allow for easier comparison in a single frame. For comparison the conditional moments of enstrophy given energy dissipation are also shown in the top row.

As reported in Yeung & Ravikumar (2020) (and chapter 3 of this thesis), conditional statistics in the top row of Figure 5.7 indicate that dissipation and enstrophy scale differently in the dissipation range but together in the inertial range; while homogeneity in space ensures that all data points converge on the diagonal (dashed) line in the limit of large r. However, the scalar dissipation rate paints a different picture. Curves for conditional moments of χ_r are generally flatter than those for Ω_r, indicating the connection between the variables ϵ_r and χ_r is not a strong one. The position of the red curve in frame (d) suggests, in the dissipation range, a low ϵ_r may co-exist with a larger but still below-average χ_r,

whereas a large ϵ_r may be associated with χ_r that is only slightly above average. As the scale sizes increase, the curves are seen to move towards the diagonal line, which represent the scenario of both variables scaling together. However, even at considerably larger scale sizes (frames (c) and (f)) differences between conditional moments of Ω_r and χ_r are still strong. One reason for the persistent differences is the fact that in the limit of the largest r possible, ϵ_r, Ω_r and χ_r all approach their mean values. In such a scenario homogeneity ensures that $\langle\epsilon\rangle = \nu\langle\Omega\rangle$ (with ν being constant). As r increases thus, as ϵ_r and Ω_r become less prone to extreme values, they also become closer to each other. In contrast, no such constraint exists between ϵ_r and Ω_r.

5.5 Conclusions

This chapter is concerned with an application of the MRIS approach (Yeung & Ravikumar, 2020) described in chapter 3 to the study of passive scalar intermittency of Schmidt numbers 1 and 0.125 at higher Reynolds number and resolution. The MRIS technique has proven useful in generating valuable results without being limited by the computational resource requirements, especially since the addition of scalars leads to an increase in computational costs. Sampling of the large scales in the scalar field is also improved by simulating scalars of the same Schmidt number with mean gradients in all three directions. The results presented are averaged over mean gradients in these three directions.

Numerical results on the moments of energy and scalar dissipation rates along with their PDFs suggest that scalar dissipation rate is more intermittent than energy dissipation although the peak values of scalar dissipation, are lower than those of energy dissipation. The PDFs of scalar and energy dissipation were observed to cross over with the probability of moderately large scalar dissipation rates being higher than those for energy dissipation. Therefore, lower order moments of scalar dissipation will dominate those of energy dissipation but at higher orders the opposite is expected. Further, the PDFs and skewness of scalar gradients parallel to the mean gradient were computed. This shows the strong devia-

tion from local isotropy as has been widely observed in flows where the passive scalar fields are sustained by a mean gradient. A slow return to isotropy was observed with increasing R_λ but yet higher Reynolds numbers are required to make a definite argument. The highest spatial resolution of $k_{max}\eta \sim 4.2$ considered is found to be sufficient for studies of local isotropy, with the gradient skewness changing very little upon a doubling of small-scale resolution from $k_{max}\eta \sim$ 2.1 to 4.2.

Studies of refined similarity hypothesis were also performed, and a clear power-law behavior in the inertial range was observed for scalars of $Sc = 1$. It was also found that the scalar dissipation rate had stricter resolution requirements compared to energy dissipation and enstrophy. Further, the conditional moments showed that while the peak values of energy dissipation and enstrophy scale well with each other, scalar dissipation and energy dissipation do not. Results such as those presented in this chapter are not widely available, especially since 1D surrogates which are commonly used along with Taylor's frozen-flow hypothesis in laboratory measurements were not used to compute energy and scalar dissipation.

CHAPTER 6
AN EXPLORATORY STUDY OF ACTIVE SCALAR TURBULENCE AND DOUBLE DIFFUSIVE PHENOMENA

In the final thesis topic, the behavior of stratified flows in the presence of two active scalars is explored. Active scalar turbulence refers to problems characterized by two-way couplings between the velocity field and one or more diffusing scalars, typically via small changes in the fluid density leading to buoyancy forces which may either suppress or amplify motions in the vertical direction. A most important application is in the ocean, with temperature and salinity at high Schmidt number being the active scalars concerned. There is also fundamental interest for this also in other contexts where Schmidt numbers may be low (Garaud, 2018). The physics is most intriguing when two scalars with different molecular diffusivities having opposing stabilizing versus de-stabilizing influences are present, where differential diffusion clearly plays a pivotal role (Turner, 1974; Schmitt, 1994). Stable stratification usually leads to oscillatory behavior and suppressed turbulence in the vertical direction, while unstable stratification leads to a strong growth of turbulence that also poses more stringent resolution requirements. Both types of stratification lead to non-stationarity and anisotropy, such that time-dependent Reynolds stress budgets can provide much useful information.

Simulations of active scalars can be performed using both the pencil decomposition code on CPUs or the GPU-based batched asynchronous slab decomposition code, where anisotropic domains with unequal number of grid points in each direction may be required. A rigorous check of the numerics under unstable stratification is also provided by comparisons with a numerical scheme (detailed derivations given in Appendix B) that diagonalizes the system of velocity-scalar equations that is advanced in time at every time step in Fourier space. Unfortunately, resource limitations have made it difficult to perform simulations at

sizes approaching those of the prior chapters. Thus, work presented in this chapter is somewhat exploratory. Nevertheless, it provides an appropriate foundation for further work in the future.

6.1 Numerical approach

The incompressible Navier-Stokes equations are used with Boussinesq approximation and advection-diffusion equation for each scalar, consistent with Kimura & Herring (1996), to describe the behavior of flows under stable and unstable density stratification.

$$\nabla \cdot \mathbf{u} = 0 \tag{6.1}$$

$$\partial \mathbf{u}/\partial t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla(p/\rho_0) - \rho' g/\rho_0 \mathbf{e_z} + \nu \nabla^2 \mathbf{u} \tag{6.2}$$

$$\partial \phi_1/\partial t + \mathbf{u} \cdot \nabla \phi_1 = -\mathbf{u} \cdot \nabla \Phi_1 + D_1 \nabla^2 \phi_1 \tag{6.3}$$

$$\partial \phi_2/\partial t + \mathbf{u} \cdot \nabla \phi_2 = -\mathbf{u} \cdot \nabla \Phi_2 + D_2 \nabla^2 \phi_2 \tag{6.4}$$

where ϕ_1 and ϕ_2 are active scalars with molecular diffusivities D_1 and D_2, g is the acceleration due to gravity, $\mathbf{e_z}$ in the unit vector in the vertical, ρ_0 is a fixed reference density and ρ' is the density fluctuation defined as

$$\rho'/\rho_0 = -c_1\phi_1 - c_2\phi_2 \tag{6.5}$$

In this equation c_1 and c_2 are fixed constants of nature analogous to the coefficient of thermal expansion in the case of temperature. If they are positive, then an increase in scalar fluctuation (of either scalar) leads to a decrease in density fluctuation. In the DNS code, if scalar fluctuations are initially absent, the values of both scalars are proportional to their mean gradients at all times. Without loss of generality, the values of g, ρ_0, c_1 and c_2 are set to unity and use the magnitude and signs of $d\Phi_1/dz$ and $d\Phi_2/dz$ to control the type

and strength of the respective contributions to stratification. The strength of stratification can be characterized by comparing the turbulence time scales (L/u') to the buoyancy time scales, $T_i = 2\pi/(gc_i|d\Phi_i/dz|)^{1/2}$, where L is the integral length scale and u' is the r.m.s velocity. The ratios between these time scales define the Froude number, $F_i = T_i/(L/u')$. Lower values of F_i correspond to stronger stratification. Positive values of the mean scalar gradient correspond to stable stratification, and negative for unstable stratification. The most interesting cases are when there are two scalars of different Sc opposing each other with equal magnitude scalar gradients. In such cases, differential diffusion between the two scalars plays a vital role.

Strong anisotropy inherent in stratified flows suggest the use of domains that are flattened and shorter in the case of stable stratification, and domains that are vertically elongated and of refined resolution in the case of unstable stratification. However in the preliminary results below focus is on understanding the effects of differential diffusion at low Schmidt numbers, in the context of application towards turbulence subjected to a strong magnetic field. In the so-called low-magnetic Reynolds number regime of magnetohydrodynamic turbulence, the turbulence develops very long length scales in the direction of the magnetic field, making the use of (horizontally) elongated domains important (Zhai & Yeung, 2018). An increase in domain size in the direction of mean gradient is also useful because of very low Sc which leads to longer length scales for the scalar field itself. In an effort to minimize any numerical uncertainties associated with rapid growth of turbulence under unstable stratification, a numerical scheme was developed that also integrates the velocity-scalar coupling terms exactly in wavenumber space, using a diagonalized linear system, which is described in Appendix B.

6.2 Preliminary results

A number of simulations initially motivated by interest in mixing in MHD turbulence (section 6.1) have been performed (without a magnetic field), on the 39-Petaflop supercomputer

Table 6.1: Basic parameters of simulations using two active scalars of $Sc_1 = 0.1$ and $Sc_2 = 0.01$ for different problem sizes. Subscript 0 and n indicates data at the beginning and end of the simulations, respectively. Simulation S1 was run with both diagonalized and non-diagonalized scheme, while S2 to S4 were run using the non-diagonalized scheme.

No.	Problem size	Froude No.		Mean Gradient		R_{λ_0}	R_{λ_n}	$k_{max}\eta_0$	$k_{max}\eta_n$
		Sc_1	Sc_2	Sc_1	Sc_2				
S1	4096×512^2	1	1.1	93	-77	22	89	1.64	0.88
S2	16384×2048^2	2	2	82	-82	64	129	1.29	1.41
S3	16384×2048^2	1	2	328	-82	64	166	1.29	3.83
S4	16384×2048^2	1	1	328	-328	64	143	1.29	0.70

Frontera at the Texas Advanced Computing Center. While many distinct parameter regimes arise, in this work focus has been on the case of two scalars at $Sc = 0.1$ and 0.01 providing opposing effects of stable and unstable stratification respectively. Table 6.1 lists three simulations. The first (S1) was a (successful) test of the numerics performed using a simulation where the respective Froude numbers were 1.0 and 1.1, with excellent agreement observed between numerical schemes without and with the diagonalization procedure described in Appendix B. Results from simulations S2 to S4 are discussed further below.

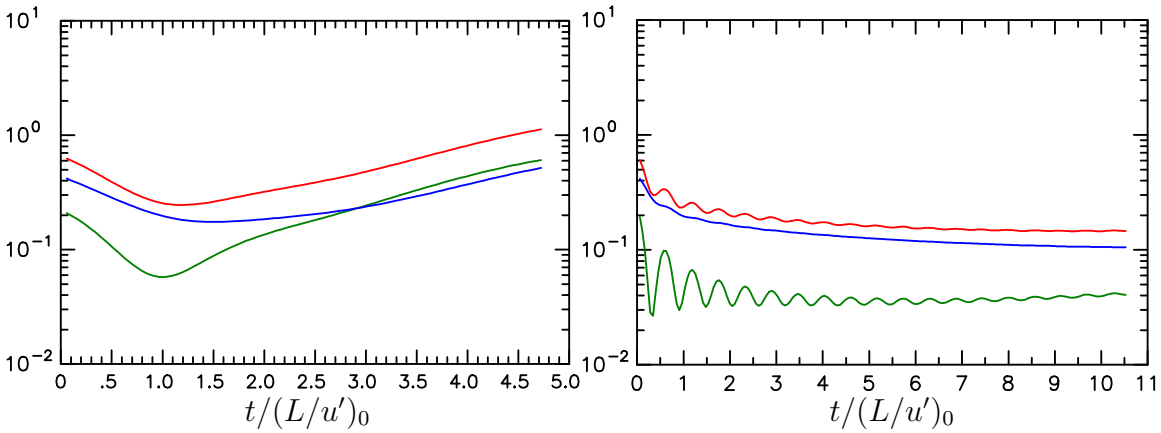

Figure 6.1: Turbulence kinetic energy, $\langle K \rangle$ (red), vertical contributions to K, $0.5\langle w^2 \rangle$ (green) and horizontal contributions to K, $0.5(\langle u^2 \rangle + \langle v^2 \rangle)$ (blue), from simulations S2 (left) and S3 (right). These results are obtained from the non-diagonalized scheme.

Figure 6.1 shows the evolution of turbulence kinetic energy (TKE) as the sum of horizontal and vertical contributions. The active scalar simulations here have been performed

with naturally decaying isotropic turbulence as initial conditions, with integral length scales small enough that numerical distortion due to a non-cubic domain is minimal. Scalars of low Sc tend to have smaller variances, since strong molecular diffusivity tends to dissipate the scalar fluctuations quickly, preventing large buildup of fluctuations. As a result, although simulation S2 was run with equal Froude numbers, the unstable stratification from the scalar of $Sc = 0.1$ (higher of the two) may be expected to dominate. After a period of about one initial eddy-turnover time, the turbulence kinetic energy grows. If the stratification were made stronger, as in simulation S4, the small scales can become poorly resolved, as indicated by the value of $k_{max}\eta$ falling below unity at the final time. This indicates a need for progressive grid refinement at later times for simulations with strongly unstable stratification. In addition there are two other questions about long-time behaviors. The first is, since the two scalars have different time scales, effects of the low Sc scalar may potentially become stronger at later times. The second is whether the buoyancy contributions to the equations of motion may ultimately raise concerns about the validity of the Boussinesq approximation.

In the right half of Figure 6.1 it can be seen that the vertical TKE is quickly subjected to oscillatory damping due to an overall stable stratification. The horizontal TKE also decays, but slowly. The ratio between horizontal and vertical TKEs indicate a strong degree of anisotropy, at least in the large scales that dominate the Reynolds stress tensor.

To further study the development of anisotropy, consider the behaviors of different terms in the budget equation for the Reynolds stress tensor, in the form

$$d\langle u_i u_j\rangle/dt = \langle 2p^{(s)} s_{ij}\rangle + \langle 2p^{(b)} s_{ij}\rangle - \langle \rho' u_i \delta_{j3}\rangle - \langle \rho' u_j \delta_{i3}\rangle - \langle \epsilon_{ij}\rangle \tag{6.6}$$

where terms on the r.h.s represent, respectively, contributions from correlation between slow pressure and strain rate, buoyancy pressure and strain rate, buoyancy flux (including effects of both scalars), and viscous dissipation. The first two of these are (because

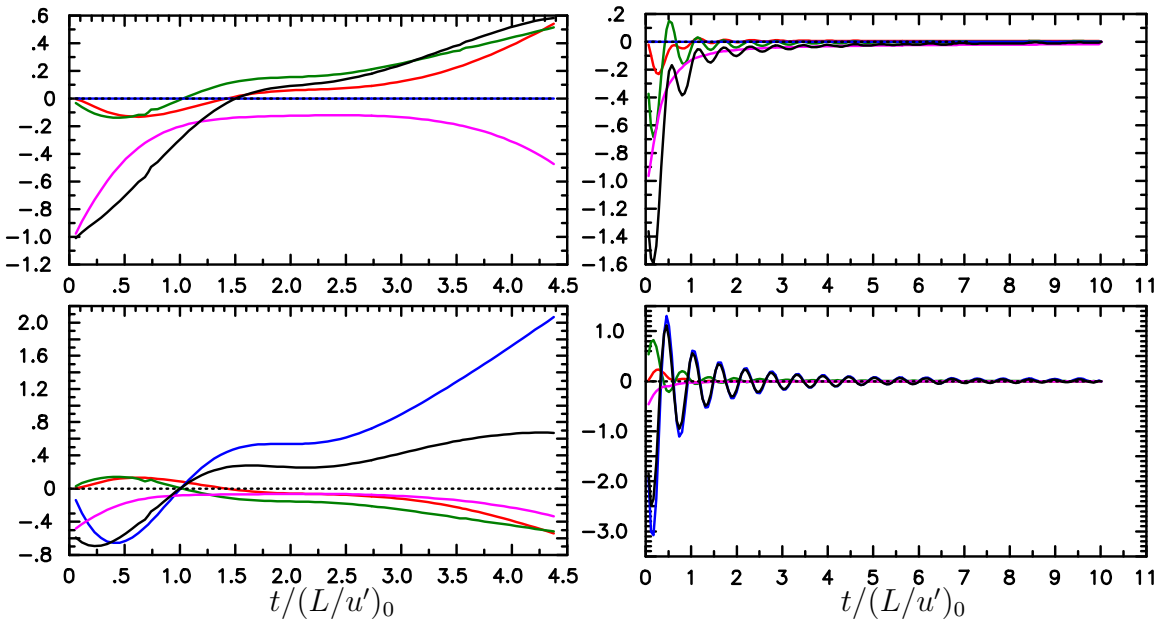

Figure 6.2: Reynolds-stress tensor budget of $0.5(\langle u^2\rangle+\langle v^2\rangle)$ (top) and $0.5\langle w^2\rangle$ (bottom) for S2 (left) and S3 (right). The different colors represent different terms in the Reynolds stress evolution equation: slow pressure strain (red), buoyancy pressure strain (green), buoyancy flux (blue), dissipation rate ($-\epsilon$, magenta), and rate of change of Reynolds-stress component in black.

of incompressibility) traceless and hence re-distributive in nature, with a direct effect on anisotropy. The buoyancy flux is nonzero only for the vertical velocity component, i.e. $\langle u_3 u_3\rangle$.

Figure 6.2 shows data on the Reynolds stress budget corresponding to Figure 6.1. On the left, where overall conditions are unstable, at early times the buoyancy flux is negative, whereas the slow pressure acts to transfer energy from the horizontal components to the vertical. However at later times the buoyancy flux becomes a strong positive production effect, and the direction of inter-component energy transfer reverses. The buoyancy contribution to the pressure strain appears to follow a similar trend. These behaviors are clearly drastically different in the case of stable stratification (on the right). Oscillatory behavior in the buoyancy flux is clearly seen, being the result of a net effect between two scalars leading to opposing stratification effects. A corresponding oscillatory behavior in the pressure-strain terms is also observed. The frequent changes in sign of the buoyancy

flux term imply a large production effect cannot be sustained. In the absence of sustained production, viscous dissipation leads to an almost monotonic decay of the TKE, as seen earlier in the right half of Figure 6.1.

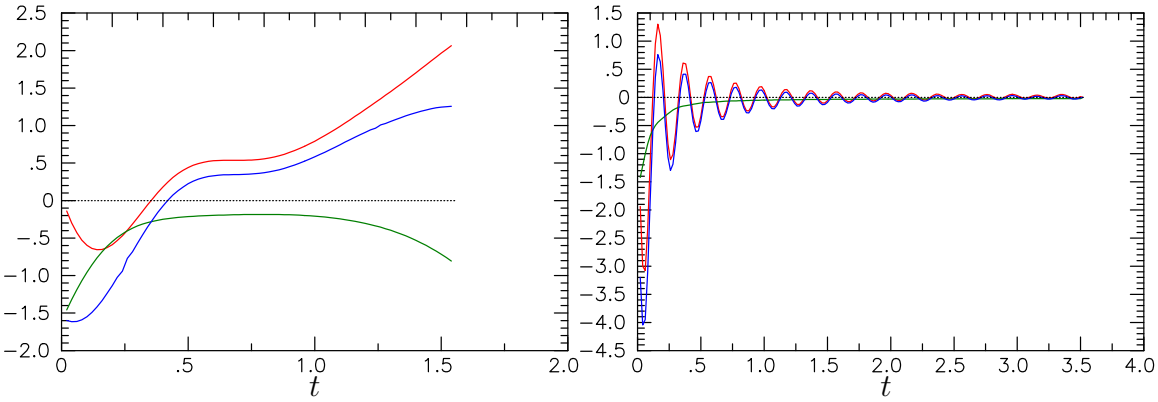

Figure 6.3: Turbulence kinetic energy budget for S2 (left) and S3 (right). The different colors represent different terms in the turbulence kinetic energy equation: buoyancy flux (red), dissipation rate ($-\epsilon$, green), and rate of change of kinetic energy (blue).

It can be seen in the frame on the left of Figure 6.3, which represents unstable stratification, that the buoyancy flux is generating turbulence while viscosity acts to dissipate the energy. At early times the buoyancy flux is negative, which suggests the high Sc stable scalar determines the behavior, before the unstable scalar begins to dominate. The rate of change term shows this behavior clearly, with negative values at early times indicating decaying turbulence followed by positive values corresponding to a growth in the energy. The frame on the right shows the terms from a stable case, where an oscillatory behavior in the buoyancy flux can be observed. The amplitude of oscillations decrease with time as the turbulence decays.

6.3 Conclusions

A numerical scheme based on linear, diagonalized equations in Fourier space was developed to minimize the numerical discretization errors associated with the two-way coupling term in the conservation of momentum equation. However, preliminary simulations suggest the solution from the new diagonalized scheme behaves exactly the same as the non-

diagonalized scheme.

Basic trends for active scalar turbulence covering cases of (overall) unstably and stably stratified turbulence extracted from some fairly large exploratory simulations have been demonstrated in Figure 6.1 and Figure 6.2. Preliminary results have raised important questions that need to be addressed.

Simulations using vertically elongated domains for unstably stratified flows with finer spatial resolution along the mean gradient direction and using flattened domains for stably stratified flows may help alleviate concerns of growing integral length scales. The validity of the Boussinesq approximation needs to be assessed, using properties of the active scalar fields involved. With buoyancy being just one type of additional forces in typical fluid flow problems, comparisons can also be made with simulations in which the velocity field is also subjected to an electromagnetic (Lorentz) force due to an imposed magnetic field.

CHAPTER 7
CONCLUSIONS AND FUTURE DIRECTIONS

The complex, yet common nature of turbulence has garnered the interests of many researchers over the past century (Lumley & Yaglom, 2001; Sreenivasan, 1999). The search for universal theories to understand and predict the behavior of turbulent flows, especially in the small-scales (Schumacher *et al.*, 2014), is of particular interest to many. Tests for such theories are crucial and dependent on attainment of high Reynolds number, which is a challenge in both simulations and laboratory experiments (Bodenschatz *et al.*, 2014). The intermittent nature of turbulence where large fluctuations over a wide range of scales in both space and time make the problem harder yet to study. Direct Numerical Simulations (DNS) of turbulence are used as the primary research tool in this thesis. Insatiable needs for high resolution and Reynolds numbers have placed DNS of turbulence as a grand challenge topic in High Performance Computing (HPC) (Yokokawa *et al.*, 2002). Newer exascale architectures provide a unique opportunity to push the boundaries of large scale scientific computing to reach for simulations at world-leading problem sizes. In particular, this has motivated the GPU-based algorithmic advances reported in this thesis and have formed the basis of an exascale coding project selected in 2019 to be part of the Center for Accelerated Application Readiness (CAAR) for Frontier, which is expected to offer 1.5 exaflops (1.5×10^{18} operations per second) in theoretical peak performance in early 2022. The overarching scheme of this thesis is concerned with the development of new state-of-the-art computational tools for turbulence, and the application of such tools to the fundamental physics of intermittency, mixing and stratification in turbulence. This chapter presents a summary of major outcomes resulting from work described in this thesis, with a view towards challenges and potential solutions which can further improve our ability to simulate and understand turbulence.

7.1 Summary of results and contributions

Heterogeneous platforms where computational work is offloaded to fast processors of smaller memory are currently primary drivers towards the next generation of leadership computers. However the adaptation of codes to fully exploit these rapidly evolving new architectures is not trivial. The research reported in this thesis has demonstrated how HPC and modern exascale computing platforms can be used efficiently in the pursuit of fundamental understanding of turbulence. The following subsections provide a summary of the major results and contributions from each chapter of this thesis.

7.1.1 GPU acceleration of extreme scale pseudo-spectral simulations of turbulence using asynchronism

A batched asynchronous algorithm allowing a currently world-leading grid resolution of 18432^3 (more than 6 trillion grid points) has been developed for dense node heterogeneous architectures, and presented at a prominent conference (SC'19) after a rigorous review process (Ravikumar *et al.*, 2019). The algorithm was recently further extended to simulate passive and active scalars, as well as track fluid particles to obtain Lagrangian statistics. This code was designed for a specific machine (Summit, an IBM Power 9 with NVIDIA GPUs) but significant work has also been performed to develop a more portable version. For example, data management and computations on the GPU are ported to OpenMP instead of CUDA Fortran. Interoperability of asynchronous library calls (such as cuFFT or rocFFT) and OpenMP task management is addressed using new features in the OpenMP 5.0 parallel programming standard. Porting the new CUDA Fortran code to OpenMP is essential for execution on a future machine Frontier that is expected to be the first GPU machine to break the Exascale barrier, in early 2022. Basic elements of the OpenMP porting strategy and challenges faced are reported in a group paper accepted for publication in the Parallel Computing Journal (Bak *et al.*, 2021) and in the proceedings of the IWOMP

2021 conference (Chapman *et al.*, 2021). This work has led to many presentations at leading conferences and workshops in the high-end HPC community.

7.1.2 Advancing understanding of turbulence through extreme-scale computation: intermittency and simulations at large problem sizes

Although the batched asynchronism algorithm above provides good performance per time step at extreme problem sizes, the overall cost of any production simulation also depends on the number of time steps needed to span a prescribed physical time period, which increases with the number of grid points and the Reynolds number of the turbulent flow simulated. A new technique of "Multiple Resolution Independent Simulations" (MRIS) using short segments at high resolution evolved from longer lower-resolution runs was developed and validated to meet this challenge. The approach was used in the studies of intermittency in stationary isotropic turbulence for flows of R_λ as high as 1300 and a spatial resolution of $k_{max}\eta = 4.5$ using 18432^3 grid points, which is, to our knowledge, the highest resolution reached in published work at this time. Scaling exponents of the moments of local averages of energy dissipation and enstrophy were computed, and a wide inertial scaling range was observed. Conditional moments of locally-averaged energy dissipation given enstrophy and vice versa show that while localized regions of high energy dissipation scales with enstrophy, the opposite is not always true. Advances through both computation and the MRIS paradigm have been instrumental in securing well-sampled results on fine-scale intermittency at much lower cost than full-length simulations at extreme-scale resolutions, which remain infeasible otherwise. This work was published as an invited paper in the Physical Review Fluids journal (Yeung & Ravikumar, 2020) and has also been presented at a number of recent conferences and invited talks.

7.1.3 Extreme dissipation and its multifractal nature at high Reynolds numbers

Turbulence being a self-similar multiplicative process can be studied under a multifractal framework. The study of energy dissipation rate from a multifractal viewpoint helps understand the intermittent nature of the small-scales from a geometric perspective as compared to a statistical perspective employed in chapter 3. High resolution simulations at Reynolds numbers as high as 1300 have been used to compute the multifractal spectrum of energy dissipation rate. The behaviors of the PDF tails of locally-averaged dissipation rate were studied and modeled using stretched exponential for local averages over different scale sizes. The multifractal spectrum and the generalized dimensions were computed at different R_λ and compared with results from (Meneveau & Sreenivasan, 1991). The results show that the spectra are in good agreement with each other and is relatively robust to changes in R_λ and the use of approximations like 1D surrogates and Taylor's frozen turbulence hypothesis in laboratory experiments. However, further numerical experiments using 1D surrogates might help understand the effect of such approximations more clearly. The spectrum from extrapolated PDFs were found to strongly support the existence of "negative dimensions" (Chhabra & Sreenivasan, 1991). It was also used to compute the energy dissipation contributions and the volume occupied by incipient singularities of different strengths. Finally, the contribution of energy dissipation rates, of different magnitudes to the overall moments, was computed. It was found that the mean value can be captured accurately by considering samples that are only as large as a few mean values, while higher order moments need samples of much larger magnitude ($\mathcal{O}(100)$ times the mean). A journal manuscript on this work is under preparation.

7.1.4 High resolution studies of intermittency in scalar dissipation rates

Direct numerical simulations have been performed of intermittency in the turbulent mixing of a passive scalar field subjected to a uniform mean gradient, at high small-scale resolution and moderately high Reynolds numbers. The MRIS technique is again used to avoid long-

running simulations that come with high computational resource demands, especially with addition of passive scalars. The PDFs of scalar dissipation rate and the moments of their 3D local averages show that stricter resolution requirements are necessary to accurately capture the small-scales of the scalar field. The PDFs also show that moderately large scalar dissipation is more probable than energy dissipation rate, which supports the fact that lower-order moments are higher for scalar dissipation compared to energy dissipation. The moments of scalar gradients, joint PDFs and conditional PDFs of scalar and energy dissipation rates were computed. The conditional moments of locally averaged scalar dissipation given energy dissipation was analyzed and found that their peak values do not scale with each other, unlike the case between enstrophy and energy dissipation.

7.1.5 Active scalar turbulence and double diffusive phenomena

A preliminary study of stratified turbulence with two active scalars of different molecular diffusivities has been conducted. Special attention was given to to anisotropy development in the Reynolds stress tensor for stably and unstably stratified flows. The numerical fidelity in the treatment of two-way coupling between the velocity and active scalar fields has been checked with reference to a diagonalized scheme that allows the body force term to be computed exactly via an integrating factor in wavenumber space. Important questions regarding the need for elongated domains with finer spatial resolution along the direction of unstable stratification are explored.

7.2 Future considerations

The research presented in this thesis has included computational algorithm development in anticipation of heterogeneous exascale computing, turbulence simulations at world-leading resolution, intermittency in velocity and passive scalar fields, multifractal analysis of turbulence and anisotropic turbulence generated by density stratification. Advances in both GPU implementation (chapter 2) and the MRIS paradigm (chapter 3) are expected to be

prime drivers for ambitious simulations at yet-higher resolution on the 1.5 exaflops computer called Frontier that will be available in early 2022. In this section, some of the many ways in which the current research can be extended to further improve our capabilities in large scale numerical simulations and understanding of turbulence are briefly described.

7.2.1 Using GPU-Direct technology for global transposes

In the current implementation of the batched asynchronous algorithm, data is copied from the GPU to the CPU in order to perform a global transpose through the network and returned to the GPU afterwards. However, future architectures are expected to have hardware support capable of performing network communication directly from the GPU. The batched asynchronous approach can be adapted to send the batch of data directly out of the GPU to the network, after computation, while receiving the transposed data on the CPU. This ensures that additional buffers are not needed on the GPU to hold the transposed while some other batches are being processed. Using this feature will also eliminate the need for additional, which should lead to further potential performance improvements.

7.2.2 Use of 3D domain decomposition and other alternatives

Message-passing communication is well known to be a dominant cost in HPC applications based on pseudo-spectral methods in multi-dimensional space. A central theme of optimization is thus to communicate either less, faster, or overlap with other operations. It may be possible to develop new 3D FFT algorithms that can compute the transforms partially on each parallel process using only distributed data. For example, use of a 3D domain decomposition can also potentially help reduce the communication costs by localizing the communication, within sub communicators, rather than performing large global transposes. However, in order to perform FFTs entire lines of data are required, which is not possible with a 3D decomposition. The use of FFT libraries that can handle distributed memory will be required, but these are not expected to give the best performance. An in-house

FFT implementation can be developed, which might open up some unique opportunities for overlap. Alternative to handle the global transpose such as the use of one-sided MPI communication or Partitioned Global Address Space (PGAS) libraries like OpenSHMEM can be explored.

7.2.3 Alternatives to FFT based DNS

A primary reason for use of pseudo-spectral approaches versus other discretization schemes is superior accuracy. However one interesting alternative, compatible with periodic boundary conditions, is the Combined Compact Finite Difference scheme developed by Gotoh *et al.* (2012) and demonstrated to provide spectral-like accuracy. This code is especially appealing for the study of high Schmidt number mixing with dual grid resolution (Clay *et al.*, 2017, 2018). Unlike FFTs, this does not require a global transpose, instead it relies on nearest neighbor communication. A DNS code using this scheme might have better communication characteristics than the pseudo-spectral code and can potentially show much better scalability. It will be interesting to see if such a scheme can be used for the velocity field as well. However, the pressure term in the Navier-Stokes equation will require careful consideration.

7.2.4 Numerical simulations of Stoke's particles

While fluid particles strictly follow the flow, many transported entities in practice do not. Examples include tracer particles used in laboratories studies, suspended particulate material such as soot in a flame, and even disease agents carried in the ambient environment. In order to study such flows, effects of particle inertia as well as finite particle sizes need to be considered. As noted in chapter 2 the GPU turbulence code has been extended to enable particle tracking for very large particle counts (over a billion). Further extension to inertial point particles is very straightforward. The study of particle size effects leading to two-way coupling involving use of an immersed boundary method is a potential task for the future.

7.2.5 Passive scalar simulations at high Schmidt number using GPUs

Turbulent mixing at high Schmidt number (i.e. low molecular diffusivity), such as salinity in the ocean, imposes severe resource requirements because of the need to resolve the Batchelor scale which is smaller than the Kolmogorov scale of the velocity field by a factor of $\sqrt{Sc}$. The complexities arising due to the difference in the size of the small-scales of the velocity and scalar field are challenging to address using computations. Clay *et al.* (2017) developed a dual-grid approach where the high Schmidt number passive scalar was handled on a finer grid using compact finite differences while the velocity field was on a coarser grid handled using a Fourier pseudo-spectral approach. This work was extended to GPUs in Clay *et al.* (2018). The batched asynchronous algorithm discussed in this thesis can be further extended, following the work referenced here, to include a dual-grid approach with the pseudo-spectral method being used for both the velocity and scalar fields.

7.2.6 Studies of intermittency at higher Reynolds numbers via extreme scale computing

With the advent of exascale architectures like Frontier, the future for extreme-scale numerical simulations of turbulence is bright. Recent progress in developing a portable implementation of the batched asynchronous algorithm provides the perfect platform for ever-larger simulations. On Frontier, simulations using 32768^3 (close to 34 trillion) grid points, which is 5.5 times larger than the current largest simulation using 18432^3 grid points, will be the target. Studies of intermittency using the MRIS approach, discussed in this thesis, for flows of Reynolds number as large as 1950 can be attempted at a spatial resolution $k_{max}\eta \approx 4.5$. These simulations will go a long way in advancing our understanding of the small-scales in turbulent flows.

7.2.7 Circulation statistics to study intermittency

Local averages of energy dissipation rate are an integral part of the refined similarity hypothesis (Kolmogorov, 1962). However, recent work (Iyer *et al.*, 2019) shows an alternative

approach where line integrals of the velocity field (called circulation) can be used to study intermittency. The new high resolution simulation data generated in this thesis can be used to further investigate the use of circulation instead of local averages.

7.2.8 Passive scalar intermittency at high Reynolds number

Current simulations (as presented in chapter 5) were limited by the available computational resources. The batched asynchronous code can be used for simulations of turbulence mixing using 18432^3 grid points with 1 passive scalar of moderate Schmidt number. Such simulations enable reaching Reynolds numbers as high 1300 and a spatial resolution of $k_{max}\eta \approx 4.5$. The resulting data will prove to be very valuable in studies of passive scalar intermittency and multifractal analysis. Also, recent code development efforts to target exascale architectures, discussed in this thesis, can be leveraged to enable even larger simulations of turbulence mixing.

7.2.9 Multifractal analysis of passive scalars

The multifractal analysis of energy dissipation rate was presented in some detail in this thesis. The scalar dissipation rate is also known to be highly intermittent and evolve in a self-similar manner. The multifractal formalism can be applied to the study of scalar dissipation rate. Comparisons of the multifractal spectra of energy and scalar dissipation rates, the effect of Schmidt number on the multifractal spectrum and the study of incipient singularities in the passive scalar field are some interesting topics to address.

7.2.10 Refined similarity theory of passive scalars at high resolution and Reynolds number

The results discussed in this thesis corresponding to the refined similarity hypothesis of passive scalars using 3D local averages provide a strong motivation to explore such tests for datasets available at higher Reynolds number and resolution. It will also be interesting to study the refined similarity hypothesis using spatial increments of velocity and scalar as

in Stolovitzky *et al.* (1995).

7.2.11 Resolution effects and Boussinesq approximation in active scalar simulations

More work (beyond chapter 6) can be done to address issues concerning insufficient spatial resolution along the mean gradient in strongly unstable stratified flows by performing simulations with vertically elongated domains with finer resolution. Similarly, flattened domains for stably stratified flows might help alleviate concerns of growing integral length scales along the plane perpendicular to the mean gradient. Statistics of density fluctuations should be computed to understand the validity of the Boussinesq approximation as the turbulence grows under unstable stratification.

7.2.12 Differential diffusion effects in stratified flows

Some preliminary results for density stratified flows in the presence of two active scalars were discussed in this thesis. A key problem of focus is on the behavior of turbulence due to differential diffusion between two scalars with different molecular diffusivities but same intensity of stratification. Interesting questions regarding the change in behavior of the flow under conditions where the unstable scalar, in one case, is of lower Schmidt number and in the other case is of higher Schmidt number can be studied.

Appendices

APPENDIX A

COMPUTING MOMENTS OF MEASURE MULTIPLIERS USING PDFS OF LOCAL AVERAGES OF ENERGY DISSIPATION RATE

As the large eddies break down into progressively smaller and smaller eddies, characteristic of multiplicative processes, such that each smaller piece receives a fraction of the total dissipation from the larger piece. This defines a cascade with multiple steps, where each step has N_r pieces of size r. The total dissipation over a fluid volume, $\psi(r)$, characterized by a linear size r is given by,

$$E_r \quad = \int_{\mathbf{x}\in\psi} \epsilon(\mathbf{x}) d^3x \tag{A.1}$$

where $\epsilon(\mathbf{x})$ is the energy dissipation per unit mass given by,

$$\epsilon(\mathbf{x}) = \sum_{i,j} \frac{\nu}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)^2 \tag{A.2}$$

The quantity E_r can be computed from the 3D local averages of energy dissipation rate, $\epsilon_r(\mathbf{x})$, using,

$$E_r \quad = \epsilon_r(\mathbf{x}) r^3 \tag{A.3}$$

where

$$\epsilon_r(\mathbf{x}) \quad = \tfrac{1}{r^3} \int_{\mathbf{x}\in\Omega} \epsilon(x) d^3x \tag{A.4}$$

Consider the q-th order moment of the total energy dissipation rate of all fluid volumes of linear size r. By conservation laws, this is related to the total energy dissipation (E_0)

over the largest fluid volume in the flow of size L_0^3 such that

$$\sum E_r^q \sim E_0^q (r/L_0)^{\tau(q)} \tag{A.5}$$

where $L_0 = 2\pi$ is the size of the cubic domain in one direction and the summation on the right-hand side is over all fluid volumes of size r. The exponents D_q can now be defined as Hentschel & Procaccia (1983),

$$D_q = \tau(q)/(q-1) \tag{A.6}$$

Equation A.5 can re-written as,

$$\sum_{n=1}^{N_r} (E_r/E_0)^q \sim (r/L_0)^{D_q(q-1)} \tag{A.7}$$

$$\left[\sum_{n=1}^{N_r} (E_r/E_0)^q\right]^{1/(q-1)} \sim (r/L_0)^{D_q} \tag{A.8}$$

where N_r is the number of samples of size r. The value of the exponents D_q can then be estimated by computing the slope of
$[\sum_{n=1}^{N_r} (E_r/E_0)^q]^{1/(q-1)}$ vs r/L_0. Using Equation A.3 the right-hand side of Equation A.7 can be re-written by

$$\sum_{n=1}^{N_r} (E_r/E_0)^q \sim \sum_{n=1}^{N_r} (\epsilon_r/\langle\epsilon\rangle)^q (r/L_0)^{3q} \tag{A.9}$$

Therefore, the moments of E_r/E_0 can be obtained from the moments of $\epsilon_r/\langle\epsilon\rangle$, which are computed from the DNS data using post-processing codes without having to compute moments of E_r/E_0 explicitly. The post-processing code generates moments up to order 8. However, some of these higher order moments have not converged statistically, making the data unreliable (see section 4.3). These moments can be computed from the PDFs of 3D local averages of energy dissipation rate ($p_r(X)$), with appropriate extrapolation of the

PDF tails. Using the PDFs, the number of samples of size r whose total dissipation falls in the range $X \pm \frac{1}{2}\Delta X$ can be estimated by using $n_r(X) = p_r(X)(L_0/r)^3$. Summing over all such values of X allows Equation A.9 to be re-written as,

$$\sum_{n=1}^{N_r}(E_r/E_0)^q = (r/L_0)^{3(q-1)}\sum_{M}(\epsilon_r^q/\langle\epsilon\rangle^q)p_r(M)\Delta M \tag{A.10}$$

where the summation above is nothing but the q-th moments of local averages of energy dissipation rate, $\langle\epsilon_r^q\rangle/\langle\epsilon\rangle^q$. Therefore,

$$\sum_{n=1}^{N_r}(E_r/E_0)^q \;\; = \;\; (r/L_0)^{3(q-1)}(\langle\epsilon_r^q\rangle/\langle\epsilon\rangle^q) \tag{A.11}$$

Equation A.10 shows that the moments of E_r/E_0 can be readily computed from the PDFs of $\epsilon_r/\langle\epsilon\rangle$.

Once the moments on the left-hand side of Equation A.8 are computed using Equation A.10 the value of exponents D_q can be estimated by computing the slope of the linear portion of the curve of $[\sum_{n=1}^{N_r}(E_r/E_0)^q]^{1/(q-1)}$ vs r/L_0 on log-log scales. For simplicity, the following has been defined,

$$F(r,q) = \left[\sum_{n=1}^{N_r}(E_r/E_0)^q\right]^{(1/(q-1))} = (r/L_0)^3[(\langle\epsilon_r^q\rangle/\langle\epsilon\rangle^q)]^{(1/(q-1))} \tag{A.12}$$

Using these values of D_q for different values of q the multifractal spectrum, $f(\alpha)$ can be computed. Using central difference, the quantity $\alpha(q))$ is computed by,

$$\alpha(q) = \frac{d}{dq}[(q-1)D_q] \tag{A.13}$$

Now using the values of $\alpha(q)$ computed at different values of q, $f(\alpha)$ can be computed

using,

$$f(\alpha(q)) = q\alpha(q) - (q-1)D_q \tag{A.14}$$

Special case: q=1

Consider the following equation from Hentschel & Procaccia (1983),

$$\sigma = -\lim_{r\to 0}\lim_{N\to\infty} S(r)/log(r) \tag{A.15}$$

where σ is the information dimension and is related to D_1.

The limit on N says that the number of samples is very large and has been left out in the equations below. Applying the L'Hospital's rule for the limit on r yields,

$$\sigma = -\lim_{r\to 0} r\frac{dS(r)}{dr} \tag{A.16}$$

$$= -\lim_{r\to 0} \frac{dS(r)}{dlog(r)} \tag{A.17}$$

where, $dlog(r) = dr/r$ is used in the last step to simplify. This is the same as D_1 (Hentschel & Procaccia, 1983), that is, $\sigma = -D_1$. Therefore to compute D_q for $q = 1$ the slope of $S(r)$ vs $log(r)$ needs to be computed. Here, $S(r)$ is defined as,

$$S(r) \;=\; \sum_{n=1}^{N_r} e_n log(e_n) \tag{A.18}$$

where $e_n = E_r/E_l$. Therefore, Equation A.18 can re-written in terms of local averages of

dissipation rates using Equation A.3 as,

$$\begin{aligned} S(r) &= \sum_{n=1}^{N_r}(E_r/E_0)log(E_r/E_0) \\ &= \sum_{n=1}^{N_r}(\epsilon_r/\langle\epsilon\rangle)(r/L_0)^3 log((\epsilon_r/\langle\epsilon\rangle)(r/L_0)^3) \\ &= \sum_{X}(\epsilon_r/\langle\epsilon\rangle)(r/L_0)^3 log((\epsilon_r/\langle\epsilon\rangle)(r/L_0)^3) n_r(X)\Delta X \end{aligned}$$

where, as seen before, $n_r(X)$ is the number of boxes of size r whose energy dissipation rate ($\epsilon_r/\langle\epsilon\rangle$) take on values in a range $X \pm \frac{1}{2}\Delta X$. A summation over all such values of X is then performed to obtain the required sum over all N_r boxes of size r. Re-writting the above equation in terms of the PDF of $\epsilon_r/\langle\epsilon\rangle$, i.e., using $n_r(M) = p_r(M)(L_0/r)^3$ yields,

$$S(r) = \sum_{M}(\epsilon_r/\langle\epsilon\rangle)log((\epsilon_r/\langle\epsilon\rangle)(r/L_0)^3)p_r(M)\Delta M \tag{A.19}$$

Equation A.19 can be used to plot $S(r)$ vs $log(r/L_0)$ and the slope of the resulting line will give the value of D_1.

APPENDIX B

LINEAR DIAGONALIZED SCHEME TO TREAT THE TWO-WAY COUPLING TERMS EXACTLY

The first step, on account of incompressibility, involves projecting the velocity vector in wavenumber space onto two coordinate axes, $\mathbf{e}_1$ and $\mathbf{e}_2$, orthogonal to each other, $\mathbf{e}_z$ and the wavenumber vector $\mathbf{k}$, such that $\mathbf{k} \cdot \hat{\mathbf{u}} = 0$. The two resulting velocity Fourier coefficients are denoted by v_1 and v_2,

$$v_1 = \hat{\mathbf{u}} \cdot \mathbf{e}_1 = (k_y \hat{u} - k_x \hat{v}) / \sqrt{k_x^2 + k_y^2} \tag{B.1}$$

$$v_2 = \hat{\mathbf{u}} \cdot \mathbf{e}_2 = -\hat{w} / \sin\beta \tag{B.2}$$

where β is the angle between k and $\mathbf{e}_z$. We now rewrite the governing equations Equation 6.2-Equation 6.4, in Fourier space and project it onto the new coordinate axes.

$$\partial v_1/\partial t = \partial \hat{\mathbf{u}}/\partial t \cdot \mathbf{e}_1 = -k^2 \nu v_1 + \mathcal{N}_1 \tag{B.3}$$

$$\partial v_2/\partial t = \partial \hat{\mathbf{u}}/\partial t \cdot \mathbf{e}_2 = -c_1 \sin\beta\ \hat{\phi_1} - c_2 \sin\beta\ \hat{\phi_2} - k^2 \nu v_2 + \mathcal{N}_2 \tag{B.4}$$

$$\partial \hat{\phi_1}/\partial t = d\Phi_1/dz\ \sin\beta\ v_2 - k^2 D\phi_1 + \mathcal{N}_3 \tag{B.5}$$

$$\partial \hat{\phi_2}/\partial t = d\Phi_2/dz\ \sin\beta\ v_2 - k^2 D\phi_2 + \mathcal{N}_4 \tag{B.6}$$

The pressure gradient vector is balanced out by the non-linear terms vector parallel to the wavenumber vector. $\mathcal{N}_1$ and $\mathcal{N}_2$ are the non-linear products projected onto $\mathbf{e}_1$ and $\mathbf{e}_2$. $\mathcal{N}_3$ and $\mathcal{N}_4$ are the non-linear products from the equation for the scalar fluctuations. The buoyancy force acts only along the z direction, and hence the buoyancy force does not

appear in Equation B.3. The resulting system can be expressed compactly as,

$$d\mathbf{Y}/dt = \sin\beta\ M\mathbf{Y} - k^2\lambda\mathbf{Y} + \mathcal{N} \tag{B.7}$$

where $\mathbf{Y}$ is a vector of the four unknowns, v_1, v_2, ϕ_1 and ϕ_2, $\mathcal{N}$ is the non-linear terms vector, λ is a 4×4 diagonal matrix with elements ($\nu,\ \nu,\ D_1,\ D_2$) and M is given by,

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -c_1 & -c_2 \\ 0 & B_1 & 0 & 0 \\ 0 & B_2 & 0 & 0 \end{bmatrix} \tag{B.8}$$

where B_1 and B_2 are used as alternate notations for $d\Phi_1/dz$ and $d\Phi_2/dz$. The matrix M can be diagonalized using its Eigen values which are 0, 0, $\pm\iota\sqrt{c_1B_1 + c_2B_2}$. We form a matrix S consisting of the Eigen vectors of the matrix M such that $\Lambda = S^{-1}MS$ is a diagonal matrix of the Eigen values.

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i\frac{\sqrt{C}}{\sqrt{B_1^2+B_2^2+C}} & -i\frac{\sqrt{C}}{\sqrt{B_1^2+B_2^2+C}} \\ 0 & -\frac{c_2}{\sqrt{c_1^2+c_2^2}} & \frac{B_1}{\sqrt{B_1^2+B_2^2+C}} & \frac{B_1}{\sqrt{B_1^2+B_2^2+C}} \\ 0 & \frac{c_1}{\sqrt{c_1^2+c_2^2}} & \frac{B_2}{\sqrt{B_1^2+B_2^2+C}} & \frac{B_2}{\sqrt{B_1^2+B_2^2+C}} \end{pmatrix} \tag{B.9}$$

where $C = c_1B_1 + c_2B_2$. We now pre-multiply equation Equation B.7 by S^{-1} and use $\mathbf{Y} = S\mathbf{X}$ such that,

$$d\mathbf{X}/dt = \sin\beta\ \Lambda\mathbf{X} - k^2S^{-1}\lambda S\mathbf{X} + \mathbf{f} \tag{B.10}$$

where $\mathbf{f} = S^{-1}\mathcal{N}$. When both scalars are of unity Sc ($D = \nu$), $S^{-1}\lambda S$ is diagonal and we can solve for $\mathbf{X}$ directly using integrating factors of the form e^{at},

$$\partial e^{at}\mathbf{X}/\partial t = e^{at} f \tag{B.11}$$

where $a = -\nu k^2$ for equations corresponding to v_1 and v_2 and $a = \pm\iota\sqrt{C}\sin\beta\ - \nu k^2$ for equations corresponding to ϕ_1 and ϕ_2. The above equation can be numerically integrated in time using Runge-Kutta second or fourth order schemes.

However, when we have non-unity Sc the matrix $S^{-1}\lambda S$ is no longer diagonal and hence the diffusion term can not be treated exactly using integrating factors. This can be avoided by using an operator splitting approach as described below,

1. Consider the following equation, without the viscous/diffusion term

$$\partial\mathbf{X}/\partial t = \Lambda\mathbf{X} + \mathbf{f}$$

   We first obtain the solution for the above equation using integrating factors. Next, we recover $\mathbf{Y}$ using $\mathbf{Y} = S\mathbf{X}$ and finally project v_1 and v_2 (in $\mathbf{Y}$) onto e_x, e_y and e_z to recover $\hat{u}$, $\hat{v}$ and $\hat{w}$. The third and fourth components of $\mathbf{Y}$ gives the scalars $\hat{\phi_1}$ and $\hat{\phi_2}$.

2. We can use the solution from the step above as the initial value for the current step and solve for $\hat{\mathbf{u}}$, $\hat{\phi_1}$ and $\hat{\phi_2}$ according to the following equation

$$\begin{aligned}
\partial\hat{\mathbf{u}}/\partial t &= -\nu k^2\hat{\mathbf{u}} \\
\partial\hat{\phi_1}/\partial t &= -D_1 k^2\hat{\mathbf{u}} \\
\partial\hat{\phi_2}/\partial t &= -D_2 k^2\hat{\mathbf{u}}
\end{aligned}$$

   The solution above can be obtained using the viscous and diffusive integrating fac-

tors.

Below, we describe the steps involved to make use of the above diagonalization scheme in the DNS code. We start with knowledge of $\hat{\mathbf{u}}$, $\hat{\phi}_1$ and $\hat{\phi}_2$ in Fourier space, at time t_n. Then we do the following:

Step 1. Go to physical space, compute nonlinear terms, transform back, project onto the coordinate axes $\mathbf{e}_1$ and $\mathbf{e}_2$. This gives us $\mathcal{N}$.

Step 2. Calculate v_1 and v_2 (at t_n) by projecting $\hat{\mathbf{u}}$ onto the coordinate axes $\mathbf{e}_1$ and $\mathbf{e}_2$ (equations Equation B.1-Equation B.2).

Step 3. Form $\mathbf{X} = S^{-1}\mathbf{Y}$ and $\mathbf{f} = S^{-1}\mathcal{N}$.

Step 4. Obtain predictor estimate of $\mathbf{X}$ at the next time step using the equation where the viscous/diffusive terms are neglected as described above.

Step 5. Recover $\mathbf{Y}$ from $\mathbf{X}$ using $\mathbf{Y} = S\mathbf{X}$. We can then recover the intermediate velocity using $\hat{\mathbf{u}} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2$. $\hat{\phi}_1$ and $\hat{\phi}_2$ are obtained from the third and fourth components of $\mathbf{Y}$.

Step 6. Obtain final predictor estimates of velocity and scalar using the viscous and diffusive integrating factors with $a = \nu k^2$ for the three velocity components and $a = D_i k^2$ for each of the two scalars. This gives the new values of velocity and scalar fluctuations at the predictor step.

Step 7. In the corrector step, the regular approach described in Rogallo (1981) can be used. However, when forming the non-linear terms for the scalar, the mean gradient term should not be included.

APPENDIX C

GPU ACCELERATION OF EXTREME SCALE PSEUDO-SPECTRAL SIMULATIONS OF TURBULENCE USING ASYNCHRONISM

K. Ravikumar, D. Appelhans and P. K. Yeung. 2019 *Proceedings of The International Conference for High Performance Computing, Networking and Storage Analysis (SC'19)*, Denver, CO.

Abstract

This paper presents new advances in GPU-driven Fourier pseudo-spectral numerical algorithms, which allow the simulation of turbulent fluid flow at problem sizes beyond the current state of the art. In contrast to several massively parallel petascale systems, the dense nodes of Summit, Sierra, and expected exascale machines can be exploited with coarser MPI decomposition's which result in improved MPI all-to-all scaling. An asynchronous batching strategy, combined with the fast hardware connection between the large CPU memory and the fast GPUs allows effective use of the GPUs on problem sizes which are too large to reside in GPU memory. Communication performance is further improved by a hybrid MPI+OpenMP approach. Favorable performance is obtained up to a 18432^3 problem size on 3072 nodes of Summit, with a GPU to CPU speedup of 4.7 for a 12288^3 problem size (the largest problem size previously published in turbulence literature).

APPENDIX D

ADVANCING UNDERSTANDING OF TURBULENCE THROUGH EXTREME-SCALE COMPUTATION: INTERMITTENCY AND SIMULATIONS AT LARGE PROBLEM SIZES

P. K. Yeung and K. Ravikumar. 2020 *Physical Review Fluids* **5**, 110517.

Abstract

Sustained and rapid advances in computing have enabled the conduct of direct numerical simulations (DNS) at increasing problem sizes and higher levels of physical realism, which have in turn contributed to many advances in understanding turbulence. However, continuing and future success at the "extreme-scale" level will likely require new algorithms adapted to emerging heterogeneous architectures, and even then, long simulations at extreme problem sizes are probably still too costly. In this paper we first describe the essential elements of an asynchronous parallel algorithm for DNS of incompressible isotropic turbulence, which scales effectively up to 18432^3 resolution (more than 6 trillion grid points) on a world-class IBM-NVIDIA CPU-GPU machine. We then propose a simulation paradigm, built on the idea that, for physical quantities of short timescales, sampling over well- separated snapshots in a long simulation at high resolution can be replaced by sampling over short simulation segments with a high degree of statistical independence evolved from snapshots at modest or even low resolution. The total computational cost is now counted in Kolmogorov timescales instead of large-eddy timescales, leading to tremendous savings at high Reynolds number. This "Multiple Resolution Independent Simulations" (MRIS) approach is validated through a series of comparisons, and subsequently applied to obtain results on fine-scale intermittency, at Taylor-scale Reynolds numbers 390 to 1300, with grid spacing smaller than the Kolmogorov length scale. The results show conclusively that extreme fluctuations of the dissipation rate are usually accompanied by extreme enstrophy,

while extreme enstrophy is usually accompanied by less-intense dissipation. Statistics of the locally averaged dissipation and enstrophy suggest these two variables scale together at inertial-range scale sizes (but not in the dissipation range). Finally, brief remarks are made concerning perspectives on likely major challenges in an exascale future, and several other topics of study where the MRIS approach may be useful.

APPENDIX E

EXTREME-SCALE COMPUTING FOR PSEUDO-SPECTRAL CODES USING GPUS & FINE-GRAINED ASYNCHRONISM, WITH APPLICATION TO TURBULENCE

K. Ravikumar, D. Appelhans and P. K. Yeung. Nov. 2018 71^{st} *Annual Meeting of the Division of Fluid Dynamics of The American Physical Society*, Atlanta, GA.

Abstract

As computing advances to the pre-Exascale era dominated by accelerators such as Graphical Processing Units, a substantial re-thinking is necessary for many communication-intensive applications, including turbulence simulations based on pseudo-spectral methods. We have developed an asynchronous algorithm with one-dimensional domain decomposition optimized for machines with large CPU memory and fast GPUs, in particular SUMMIT at the Oak Ridge National Laboratory, which consists of IBM Power-9 CPU's and NVIDIA V100 GPU's. Data located in the CPU memory are processed in a fine-grained (batch) manner by overlapping high BW NVLINK transfers, with fast GPU computations and high BW system interconnect allowing a much larger problem to be run than the much smaller GPU memory might suggest. Pinned memory and zerocopy approaches are used to transfer strided data between the GPU and CPU obtaining high NVLINK throughput. Several advanced communication protocols are explored in order to obtain maximum network throughput for collective communication. Benchmarks at the scale of 12288^3 grid points on 1024 SUMMIT nodes show good weak scaling, with a speedup of over 3X compared to the multi-threaded CPU-only algorithm.

APPENDIX F

REACHING HIGH RESOLUTION FOR STUDIES OF INTERMITTENCY IN ENERGY AND SCALAR DISSIPATION RATES

K. Ravikumar, P. K. Yeung and K. R. Sreenivasan. Nov. 2020 73^{rd} *Annual Meeting of the Division of Fluid Dynamics of The American Physical Society*, Virtual (Chicago time), IL.

Abstract

Passive scalar fields in high Reynolds number turbulence are often observed to be highly intermittent in both the inertial-convective (via the extreme anomaly of structure function exponents) and viscous-diffusive ranges (via intense fluctuations of the scalar dissipation rate, χ), with the Schmidt number, Sc, acting as an additional parameter. High-resolution direct numerical simulations are clearly crucial, and reliable conclusions on the Sc-effects require that resolution be adequate for all scalars involved. Such calculations are computationally very expensive. However, it is possible (Yeung & Ravikumar, to appear in Phys. Rev. Fluids, 2020) to replace long simulations of stationary isotropic turbulence at high resolution by multiple short segments evolved from lower-resolution data, at much lower cost. It is also useful to perform ensemble averaging over the statistics of scalars with the same Sc but subjected to uniform mean gradients in different directions. Fluctuations of χ for a scalar with Sc = 1 are more intermittent than those of the energy dissipation rate (ϵ). Numerical results including moments of local averages of χ and conditional moments of χ given ϵ at resolution up to 6144^3 will be presented.

APPENDIX G

ACHIEVING PORTABILITY FOR A HIGHLY OPTIMIZED GPU CODE FOR 3D FOURIER TRANSFORMS AT EXTREME PROBLEM SIZES

K. Ravikumar, O. Hernandez, J. Levesque, S. Nichols and P. K. Yeung. Sep. 2020 *Performance, Portability, and Productivity in HPC Forum*, Virtual.

Abstract

The current drive towards exascale performance in the supercomputing community has seen a sustained rise of heterogeneous architectures where GPUs provide massive computational power. Communication intensive codes require a high degree of adaptability in both network and GPU usage to achieve optimal performance on such architectures. For instance, differences in the types and numbers of GPU devices attached to each CPU host, compilers and programming libraries influence how best to construct a code that can scale up efficiently to largest node counts or largest problem sizes. This is the case for a recently developed pseudo-spectral code for the three-dimensional Navier-Stokes equations, where an asynchronous batched approach with efficient memory usage is used to solve large problem sizes without being limited by the available GPU memory (Ravikumar *et al.*, 2019). On Summit at the Oak Ridge Leadership Computing Facility, CUDA Fortran was used to target NVIDIA Volta 100 GPUs and achieve good speedups and scalability to very large problem sizes.

A natural follow-up task is to extend this programming strategy to other machines which are populated with other types of GPUs, or for which CUDA Fortran support is not available. We consider targeting GPUs using OpenMP, especially version 5.0 and beyond, which is expected to provide enhanced support for management of asynchronous library calls. Ideally, we hope to achieve substantial interoperability between different GPU platforms, to a degree approaching that for CPU codes, which can often be ported from one

CPU machine to another, with very few changes. In this talk we will discuss some possible ideas and recent progress made towards achieving objectives along these lines. We also address the potential use of alternative approaches for overlapping computation and all-to-all communication.

BIBLIOGRAPHY

AHLBERG, J. H., WILSON, E. N. & WALSH, J. J. 1967 *The Theory of Splines and Their Applications*. New York: Academic Press.

ANSELMET, F., ANTONIA, R. A. & DANAILA, L. 2001 Turbulent flows and intermittency in laboratory experiments. *Planet. Space Sci.* **49**, 1177 – 1191.

APPELHANS, D. 2018*a* Github repository: gpu-tips/nvtx/nvtx_mod.f90.

APPELHANS, D. 2018*b* Tricks, Tips, and Timings: The Data Movement Strategies You Need to Know. In *GPU Technology Conference*.

BAK, S., BERTONI, C., BOEHM, S., BUDIARDJA, R., CHAPMAN, B., DOERFERT, J., EARL, C., EISENBACH, M., FINKEL, H., HERNANDEZ, O., HUBER, J., IWASAKI, S., KALE, V., KENT, P. R. C., KWACK, J. H., LIN, M., LUSZCZEK, P., LUO, Y., POPHALE, S., RAVIKUMAR, K., SARKAR, V., SCOGLAND, S., TIAN, S. & YEUNG, P. K. 2021 OpenMP application experiences: porting to accelerated nodes. *Paral. Comput.* (accepted) .

BERSHADSKII, A. & MIKISHEV, A. 1994 Multifractality of large turbulent fluctuations and the topology of strange attractors. *Phys. Rev. E* **50**, 3723–3729.

BERSHADSKII, A. & TSINOBER, A. 1992 Asymptotic fractal and multifractal properties of turbulent dissipative fields. *Phys. Lett. A* **165**, 37–40.

BILGER, R. W. 2004 Some aspects of scalar dissipation. *Flow, Turb. & Combust.* **72**, 93–114.

BODENSCHATZ, E., BEWLEY, G. P., NOBACH, J., SINHUBER, M. & XU, H. 2014 Variable density turbulence tunnel facility. *J. Scient. Instr.* **85**, 093908.

BUARIA, D., BODENSCHATZ, E. & PUMIR, A. 2020 Vortex stretching and enstrophy production in high Reynolds number turbulence. *Phys. Rev. Fluids* **5**, 104602.

BUARIA, D., PUMIR, A., BODENSCHATZ, E. & YEUNG, P. K. 2019 Extreme velocity gradients in turbulent flows. *New J. Phys.* **21**, 043004.

BUARIA, D., SAWFORD, B. L. & YEUNG, P. K. 2016 A Lagrangian study of turbulent mixing: forward and backward dispersion of molecular trajectories in isotropic turbulence. *J. Fluid Mech.* **799**, 352–382.

BUARIA, D. & YEUNG, P.K. 2017 A highly scalable particle tracking algorithm using partitioned global address space (PGAS) programming for extreme-scale turbulence simulations. *Comput. Phys. Commun.* **221**, 246–258.

CANUTO, C, HUSSAINI, M.Y., QUARTERONI, A. & T.A., ZANG 1988 *Spectral Methods in Fluid Dynamics*. Berlin: Springer-Verlag.

CHAPMAN, B., PHAM, B., YANG, C., DALEY, C., BERTONI, C., KULKARNI, D., ORYSPAYEV, D., D'AZEVEDO, E., HE, H., DOERFERT, J., ZHOU, K., RAVIKUMAR, K., GORDON, M., DEL BEN, M., LIN, M., ALKAN, M., KRUSE, M., HERNANDEZ, O., YEUNG, P. K., XU, P., POPHALE, S., SATTASATHUCHANA, T., KALE, V. & HUHN, W. 2021 Outcomes of OpenMP hackathon: OpenMP application experiences with the offloading model. In *Proc. 17th International Workshop on OpenMP (IWOMP)*.

CHATTERJEE, A. G., VERMA, M. K., KUMAR, A., SAMTANEY, R., HADRI, B.. & KHURRAM, R. 2018 Scaling of a Fast Fourier Transform and a pseudo-spectral fluid solver up to 196608 cores. *J. Parallel & Distrib. Comput.* **113**, 77–91.

CHEN, S., SREENIVASAN, K. R. & NELKIN, M. 1997 Inertial range scalings of dissipation and enstrophy in isotropic turbulence. *Phys. Rev. Lett.* **79**, 1253–1256.

CHHABRA, ASHVIN B. & SREENIVASAN, K. R. 1991 Negative dimensions: Theory, computation, and experiment. *Phys. Rev. A.* **43**, 1114–1117.

CLAY, M. P., BUARIA, D., GOTOH, T. & YEUNG, P. K. 2017 A dual communicator dual grid-resolution algorithm for Petascale simulations of turbulent mixing at high Schmidt number. *Comp. Phys. Comm.* **219**, 313–328.

CLAY, M. P., BUARIA, D., YEUNG, P. K. & GOTOH, T. 2018 GPU acceleration of a petascale application for turbulent mixing at high Schmidt number using OpenMP 4.5. *Comp. Phys. Comm.* **228**, 100–114.

CZECHOWSKI, H., BATTAGLINO, C., MCCLANAHAN, C., IYER, K., YEUNG, P. K. & VUDUC, R. 2012 On the communication complexity of 3D FFTs and its implications for exascale. In *Proceedings of the International Supercomputing Conference (ISC'12)*, pp. 205–214. San Servolo Island, Venice, Italy: ACM.

DALCIN, L., MORTENSEN, M. & KEYES, D. E. 2018 Fast parallel multidimensional FFT using advanced MPI. *arXiv:1804.09536 [cs]* ArXiv: 1804.09536.

DAS, R. & GIRIMAJI, S. S. 2019 On the Reynolds number dependence of velocity-gradient structure and dynamics. *J. Fluid Mech.* **861**, 163–179.

DE SENSI, DANIELE, DI GIROLAMO, SALVATORE, MCMAHON, KIM H., ROWETH, DUNCAN & HOEFLER, TORSTEN 2020 An In-Depth Analysis of the Slingshot Interconnect. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, *SC '20* . IEEE Press.

DONZIS, D. A. & YEUNG, P. K. 2010 Resolution effects and scaling in numerical simulations of passive scalar mixing in turbulence. *Physica D* **239**, 1278–1287.

DONZIS, D. A., YEUNG, P. K. & SREENIVASAN, K. R. 2008 Energy dissipation rate and enstrophy in isotropic turbulence: resolution effects and scaling in direct numerical simulations. *Phys. Fluids* **20**, 045108.

ELSINGA, GERRIT E., ISHIHARA, TAKASHI & HUNT, JULIAN C. R. 2020 Extreme dissipation and intermittency in turbulence at very high Reynolds numbers. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **476**, 20200591.

ESWARAN, V. & POPE, S. B. 1988 An examination of forcing in direct numerical simulations of turbulence. *Comput. Fluids* **16**, 257–278.

FRISCH, U. 1995 *Turbulence*. Cambridge University Press.

GARAUD, P. 2018 Double-Diffusive Convection at Low Prandtl Number. *Annu. Rev. Fluid Mech.* **50**, 275–298.

GARGETT, A. E., MERRYFIELD, W. J. & HOLLOWAY, G. 2003 Direct numerical simulation of differential scalar diffusion in three-dimensional stratified turbulence. *J. Phys. Oceanogr.* **33**, 1758–1782.

GHOLAMI, A., HILL, J., MALHOTRA, D. & BIROS, G. 2016 Accfft: A library for distributed-memory FFT on CPU and GPU architectures. *CoRR* **abs/1506.07933**, arXiv: 1506.07933.

GOTOH, T., HATANAKA, S. & MIURA, H. 2012 Spectral compact difference hybrid computation of passive scalar in isotropic turbulence. *J. Comput. Phys.* **231**, 7398–7414.

GOTOH, T. & YEUNG, P. K. 2013 Passive scalar transport in turbulence: a computational perspective. In *Ten Chapters in Turbulence* (ed. P. A. Davidson, Y. Kaneda & K. R. Sreenivasan). Cambridge University Press.

GRUBER, A., RICHARDSON, E. S., ADITYA, K. & CHEN, J. H. 2018 Direct numerical simulations of premixed and stratified flame propagation in turbulent channel flow. *Phys. Rev. Fluids* **3**, 110507.

HENTSCHEL, H. G. E. & PROCACCIA, I. 1983 The infinite number of generalized dimensions of fractals and strange attractors. *Physica D* **8**, 435–444.

HOEFLER, T., GROPP, W., THAKUR, R. & TRÄFF, J. L. 2010 Toward performance models of mpi implementations for understanding application scaling issues. In *Recent Advances in the Message Passing Interface* (ed. Rainer Keller, Edgar Gabriel, Michael Resch & Jack Dongarra), pp. 21–30. Berlin, Heidelberg: Springer Berlin Heidelberg.

HONGBING, H., XIAOXIA, Z. & YONG, W. 2018 Research on performance model for alltoall collective communication. *J. Frontiers Comput. Sci. & Tech.* **12**, 559.

IBM POWER9 NPU, TEAM 2018 Functionality and performance of nvlink with ibm power9 processors. *IBM Journal of Research and Development* **62** (4/5), 9:1–9:10.

ISHIHARA, T., GOTOH, T. & KANEDA, Y. 2009 Study of high-Reynolds number isotropic turbulence by direct numerical simulation. *Annu. Rev. Fluid Mech.* **41**, 165–180.

ISHIHARA, T., KANEDA, Y. & HUNT, J. C. R. 2013 Thin shear layers in high Reynolds number turbulence – DNS results. *Flow, Turb. & Combust.* **91**, 895–929.

ISHIHARA, T., MORISHITA, K., YOKOKAWA, M., UNO, A. & KANEDA, Y. 2016 Energy spectrum in high-resolution direct numerical simulation of turbulence. *Phys. Rev. Fluids* **1**, 082403.

IYER, K. P., SCHUMACHER, J., SREENIVASAN, K. R. & YEUNG, P. K. 2018 Steep cliffs and saturated exponents in three-dimensional scalar turbulence. *Phys. Rev. Lett.* **121**, 264501.

IYER, K. P., SREENIVASAN, K. R. & YEUNG, P. K. 2015 Refined similarity hypotheses using 3D local averages. *Phys. Rev. E.* **92**, 063024.

IYER, K. P., SREENIVASAN, K. R. & YEUNG, P. K. 2016 Reynolds number scaling of velocity increments in isotropic turbulence. *Phys. Rev. E.* **95**, 021101.

IYER, K. P., SREENIVASAN, K. R. & YEUNG, P. K. 2019 Circulation in High Reynolds Number Isotropic Turbulence is a Bifractal. *Phys. Rev. X* **9**, 041006.

JAGANNATHAN, S. & DONZIS, D. A. 2016 Reynolds and Mach number scaling in solenoidally-forced compressible turbulence using high-resolution direct numerical simulations. *J. Fluid Mech.* **789**, 669–707.

JIMÉNEZ, J., WRAY, A. A., SAFFMAN, P. G. & ROGALLO, R. S. 1993 The structure of intense vorticity in isotropic turbulence. *J. Fluid Mech.* **255**, 65–90.

KAILASNATH, P., SREENIVASAN, K. R. & STOLOVITZKY, G. 1992 Probability density of velocity increments in turbulent flows. *Phys. Rev. Lett.* **68**, 2766–2769.

KANEDA, Y., ISHIHARA, T., YOKOKAWA, M., ITAKURA, T. & UNO, A. 2003 Energy dissipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box. *Phys. Fluids* **15**, L21–L24.

KERR, R. M. 1985 Higher-order derivative correlations and the alignment of small-scale structures in isotropic numerical turbulence. *J. Fluid Mech.* **153**, 31–58.

KIMURA, Y. & HERRING, J. R. 1996 Diffusion in stably stratified turbulence. *J. Fluid Mech.* **328**, 253–269.

KOLMOGOROV, A. N. 1941 The local structure of turbulence in an incompressible fluid with very large Reynolds numbers. Dokl. Akad. Nauk SSSR **30**, 301–305 See also *Turbulence and Stochastic Processes: Kolmogorov's ideas 50 years on*, eds. J.C.R. Hunt, O.M. Phillips & D. Williams, Royal Society, London, 1991.

KOLMOGOROV, A. N. 1962 A refinement of previous hypotheses concerning the local structure of a viscous incompressible fluid. *J. Fluid Mech.* **13**, 82–85.

LEE, M., MALAYA, N. & MOSER, R. D. 2013 Petascale direct numerical simulation of turbulent channel flow up to 786K cores. In *International Conference for High Performance Computing, Networking and Storage Analysis (SC), Denver, CO*, pp. 61:1–61:11. New York, NY, USA: ACM.

LEE, M. & MOSER, R. D. 2015 Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. *J. Fluid Mech.* **774**, 395–415.

LEE, M., ULERICH, R., MALAYA, N. & MOSER, R. D. 2014 Experiences from leadership computing in simulations of turbulent fluid flows. *Comput. Sci & Engrg.* **16**, 24–31.

LUMLEY, J. L. & YAGLOM, A. M. 2001 A century of turbulence. *Flow, Turb. & Combust.* **66**, 241–286.

MANDELBROT, B. B. 1983 *The fractal geometry of nature*, 3rd edn. W. H. Freeman and Comp.

MENEVEAU, C. & SREENIVASAN, K. R. 1991 The multifractal nature of turbulent energy-dissipation. *J. Fluid Mech.* **224**, 429–484.

MENEVEAU, C., SREENIVASAN, K. R., KAILASNATH, P. & FAN, M. S. 1990 Joint multifractal measures: Theory and applications to turbulence. *Phys. Rev. A.* **41**, 894–913.

MININNI, P. D., ROSENBERG, D., REDDY, R. & POUQUET, A. 2011 A hybrid MPI-OpenMP scheme for scalable parallel pseudospectral computations for fluid turbulence. *Parallel Computing* **37**, 316 – 326.

MOIN, P. & MAHESH, K. 1998 Direct numerical simulation: A tool in turbulence research. *Annu. Rev. Fluid Mech.* **30**, 539–578.

MYDLARSKI, L. & WARHAFT, Z. 1998 Passive scalar statistics in high-péclet-number grid turbulence. *J. Fluid Mech.* **358**, 135–175.

NELKIN, M. 1999 Enstrophy and dissipation must have the same scaling exponents in the high Reynolds number limit of fluid turbulence. *Phys. Fluids* **11**, 2202–2204.

NVIDIA 2019 The nvidia visual profiler.

OBUKHOV, A. M. 1962 Some specific features of atmospheric turbulence. *J. Fluid Mech.* **13**, 77–81.

OLCF, OAK RIDGE LEADERSHIP COMPUTING FACILITY 2019 Summit system user guide.

OLCF, OAK RIDGE LEADERSHIP COMPUTING FACILITY 2021 Frontier webpage https://www.olcf.ornl.gov/frontier/.

OVERHOLT, M. R. & POPE, S. B. 1996 Direct numerical simulation of a passive scalar with imposed mean gradient in isotropic turbulence. *Phys. Fluids* **8**, 3128–3148.

PATTERSON, G. S. & ORSZAG, S. A. 1971 Spectral calculations of isotropic turbulence: efficient removal of aliasing interactions. *Phys. Fluids* **14**, 2538–2541.

PEKUROVSKY, D. 2012 P3DFFT: A framework for parallel computations of Fourier transforms in three dimensions. *Siam. J. Sci. Comput.* **34**, C192–C209.

PJESIVAC-GRBOVIC, J., ANGSKUN, T., BOSILCA, G., FAGG, G.E., GABRIEL, E. & DONGARRA, J.J. 2005 Performance analysis of mpi collective operations. In *19th IEEE International Parallel and Distributed Processing Symposium*.

PLIMPTON, S. J 2017 FFTs for (mostly) Particle Codes within the DOE Exascale Computing Project. *Tech. Rep.*. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

POPE, S. B. 2000 *Turbulent Flows*. Cambridge, UK: Cambridge University Press.

RAVIKUMAR, K., APPELHANS, D. & YEUNG, P. K. 2019 GPU acceleration of extreme scale pseudo-spectral simulations of turbulence using asynchronism. In *Proceedings of The International Conference for High Performance Computing, Networking and Storage Analysis (SC'19), Denver, CO, USA. ACM, New York, NY, USA, https://doi.org/10.1145/3295500.3356209*.

RICHARDSON, LEWIS F. 1922 *Weather prediction by numerical process*. Cambridge University Press.

RILEY, J. J. & DEBRUYNKOPS, S. M. 2003 Dynamics of turbulence strongly influenced by buoyancy. *Phys. Fluids* **15**, 2047–2059.

ROGALLO, R. S. 1981 Numerical experiments in homogeneous turbulence. *NASA Tech. Memo. 81315*, NASA Ames Research Center .

RUETSCH, G. & FATICA, M. 2014 *CUDA Fortran for Scientists and Engineers*. Morgan Kaufmann Publishers.

SAWFORD, B. L. & PINTON, J-F. 2013 A Lagrangian view of turbulent dispersion and mixing. In *Ten Chapters in Turbulence* (ed. P. A. Davidson, Y. Kaneda & K. R. Sreenivasan). Cambridge University Press.

SCHMITT, R W 1994 Double diffusion in oceanography. *Annu. Rev. Fluid Mech.* **26**, 255–285.

SCHUMACHER, J., SCHEEL, J. D., KRASNOV, D., DONZIS, D. A., YAKHOT, V. & SREENIVASAN, K. R. 2014 Small-scale universality in fluid turbulence. *Proc. Nat. Acad. Sci.* **111**, 10961–10965.

SCHUMACHER, J., SREENIVASAN, K. R. & YEUNG, P. K. 2005 Very fine structures in scalar mixing. *J. Fluid Mech.* **531**, 113–122.

SHAINER, G., AYOUB, A., LUI, P., LIU, T., KAGAN, M., TROTT, C. R., SCANTLEN, G. & CROZIER, P. S. 2011 The Development of Mellanox/NVIDIA GPUDirect over InfiniBand–a New Model for GPU to GPU Communications. *Comput. Sci.* **26**, 267–273.

SREENIVASAN, K R 1991*a* Fractals and multifractals in fluid turbulence. *Annu. Rev. Fluid Mech.* **23**, 539–604.

SREENIVASAN, K. R. 1991*b* On local isotropy of passive scalars in turbulent shear flows. *Proc. Roy. Soc. Lond. A* **434**, 165–182.

SREENIVASAN, K. R. 1993 An update on the intermittency exponent in turbulence. *Phys. Fluids A* **5**, 512–514.

SREENIVASAN, K. R. 1998 An update on the energy dissipation rate in isotropic turbulence. *Phys. Fluids* **10**, 528–529.

SREENIVASAN, K. R. 1999 Fluid turbulence. *Rev. Mod. Phys.* **71**, S383–S395.

SREENIVASAN, K. R. & ANTONIA, R. A. 1997 The phenomenonology of small-scale turbulence. *Annu. Rev. Fluid Mech.* **29**, 435–472.

SREENIVASAN, K. R. & KAILASNATH, P. 1993 An update on the intermittency exponent in turbulence. *Physics of Fluids A: Fluid Dynamics* **5** (2), 512–514, arXiv: https://doi.org/10.1063/1.858877.

SREENIVASAN, K. R. & MENEVEAU, C. 1986 The fractal facets of turbulence. *J. Fluid Mech.* **173**, 357–386.

SREENIVASAN, K. R. & MENEVEAU, C. 1988 Singularities of the equations of fluid motion. *Phys. Rev. A* **38**, 6287–6295.

STELLMACH, S., TRAXLER, A., GARAUD, P., BRUMMELL, N. & RADKO, T. 2011 Dynamics of fingering convection. part 2 the formation of thermohaline staircases. *J. Fluid Mech.* **677**, 554–571.

STOLOVITZKY, G., KAILASNATH, P. & SREENIVASAN, K. R. 1992 Kolmogorov's refined similarity hypotheses. *Phys. Rev. Lett.* **69**, 1178–1181.

STOLOVITZKY, G., KAILASNATH, P. & SREENIVASAN, K. R. 1995 Refined similarity hypotheses for passive scalars mixed by turbulence. *J. Fluid Mech.* **297**, 275–291.

TAYLOR, G. I. 1938 The spectrum of turbulence. *Proc. Roy. Soc. A* **164**, 476–490.

TENNEKES, H. 1975 Eulerian and Lagrangian time microscales in isotropic turbulence. *J. Fluid Mech.* **67**, 561–567.

TSINOBER, A., VEDULA, P. & YEUNG, P. K. 2001 Random taylor hypothesis and the behavior of local and convective accelerations in isotropic turbulence. *Phys. Fluids* **13**, 1974–1984.

TURNER, J S 1974 Double-diffusive phenomena. *Annu. Rev. Fluid Mech.* **6**, 37–54.

VEDULA, P., YEUNG, P. K. & FOX, R. O. 2001 Dynamics of scalar dissipation in isotropic turbulence: a numerical and modelling study. *J. Fluid Mech.* **433**, 29–60.

VETTER, S., CALDEIRA, A.B. & REDBOOKS, IBM 2018 *IBM Power System AC922 Introduction and Technical Overview*. IBM Redbooks.

WARHAFT, Z. 2000 Passive scalars in turbulent flows. *Annu. Rev. Fluid Mech.* **32**, 203–240.

WATANABE, T. & FUJISAKA, H. 2000 Asymptotic behavior of the q-th order intermittency exponent in fully developed turbulence. *J. Phys. Soc. Japan* **69**, 1672–1675.

WATANABE, T., RILEY, J. J., DE BRUYN KOPS, S. M., DIAMESSIS, P. J. & ZHOU, Q. 2016 Turbulent/non-turbulent interfaces in wakes in stably stratified fluids. *J. Fluid Mech.* **797**, R1.

YAKHOT, V. & SREENIVASAN, K. R. 2005 Anomalous scaling of structure functions and dynamic constraints on turbulence simulations. *J. Stat. Phys.* **121**, 823–841.

YEUNG, P. K. 1996 Multi-scalar triadic interactions in differential diffusion with and without mean scalar gradients. *J. Fluid Mech.* **321**, 235–278.

YEUNG, P. K. 2002 Lagrangian investigations of turbulence. *Annu. Rev. Fluid Mech.* **34**, 115–142.

YEUNG, P. K., DONZIS, D. A. & SREENIVASAN, K. R. 2005 High-Reynolds-number simulation of turbulent mixing. *Phys. Fluids* **17**, 081703.

YEUNG, P. K., DONZIS, D. A. & SREENIVASAN, K. R. 2012 Dissipation, enstrophy and pressure statistics in turbulence simulations at high Reynolds numbers. *J. Fluid Mech.* **700**, 5–15.

YEUNG, P. K. & POPE, S. B. 1988 An algorithm for tracking fluid particles in numerical simulations of homogeneous turbulence. *J. Comput. Phys.* **79**, 373–416.

YEUNG, P. K. & POPE, S. B. 1989 Lagrangian statistics from direct numerical simulations of isotropic turbulence. *J. Fluid Mech.* **207**, 531–586.

YEUNG, P. K., POPE, S. B., KURTH, E. A. & LAMORGESE, A. G. 2007 Lagrangian conditional statistics, acceleration and local relative motion in numerically simulated isotropic turbulence. *J. Fluid Mech.* **582**, 399–422.

YEUNG, P. K., POPE, S. B., LAMORGESE, A. G. & DONZIS, D. A. 2006 Acceleration and dissipation statistics of numerically simulated isotropic turbulence. *Phys. Fluids* **18**, 065103.

YEUNG, P. K., POPE, S. B. & SAWFORD, B. L. 2006b Reynolds number dependence of Lagrangian statistics in large numerical simulations of isotropic turbulence. *J. Turb.* **7(58)**, 1–12.

YEUNG, P. K. & RAVIKUMAR, K. 2020 Advancing understanding of turbulence through extreme-scale computation: Intermittency and simulations at large problem sizes. *Phys. Rev. Fluids* **5**, 110517.

YEUNG, P. K., SREENIVASAN, K. R. & POPE, S. B. 2018 Effects of finite spatial and temporal resolution in direct numerical simulations of incompressible isotropic turbulence. *Phys. Rev. Fluids* **3**, 064603.

YEUNG, P. K., ZHAI, X. M. & SREENIVASAN, K. R. 2015 Extreme events in computational turbulence. *Proc. Nat. Acad. Sci.* **112**, 12633–12638.

YOKOKAWA, M., ITAKURA, T., UNO, A., ISHIHARA, T. & KANEDA, Y. 2002 16.4-Tflops direct numerical simulation of turbulence by a Fourier spectral method on the Earth Simulator. In *Proceedings of the Supercomputing Conference*. Baltimore.

YOSHIDA, K., YAMAGUCHI, J. & KANEDA, Y. 2005 Regeneration of small scales by data assimilation in turbulence. *Phys. Rev. Lett.* **94**, 014501.

ZHAI, X. M. & YEUNG, P. K. 2018 The evolution of anisotropy in direct numerical simulations of mhd turbulence in a strong magnetic field on elongated solution domains. *Phys. Rev. Fluids* **3**, 084602.

VITA

Kiran Ravikumar was born in Bangalore, India. During his childhood, Kiran's interests were in sports, specifically Cricket, Volleyball and Football. By the time he was in his tenth grade, he developed an interest in Physics and Aircraft's and was fascinated by Engineering. In 2011, Kiran attended Manipal Institute of Technology (MIT) in the Department of Aeronautical Engineering to develop his interests in the field further. He was a member of the student satellite team at MIT where he worked on designing and analyzing the structural aspects of the satellite. He later switched his focus to Computational Fluid Dynamics, where he worked on optimizing airfoil shapes to improve their aerodynamic efficiency. After a semester-long internship at Aeronautical Development Establishment in India working on preliminary design of a solar-powered aircraft, he obtained the degree of Bachelor of Technology in Aeronautical Engineering in 2015. He then joined the School of Aerospace Engineering at Georgia Institute of Technology to focus on optimization and fluid dynamics. During his first year at graduate school, he was exposed to the topic of turbulence in greater detail and decided to pursue a doctoral degree in this topic under the supervision of Prof. P. K. Yeung. He enjoys designing and developing large-scale numerical simulations to run on supercomputers. Outside academic pursuits, Kiran is interested in playing Chess, watching sports games and movies and travelling.