University of Alberta

MULTISPECTRAL REDUCTION OF TWO-DIMENSIONAL TURBULENCE

by

Malcolm Roberts

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mathematics

Department of Mathematical and Statistical Sciences

Fall 2011

Edmonton, Alberta

This dissertation is dedicated to my mother, Barbara Stewart, who was always an inspiration and a guide for me.

Abstract

Turbulence is a chaotic motion of fluid that can be described by the Navier–Stokes equations or even highly simplified shell models. Under the continuum limit, standard shell models of turbulence are shown to reduce to a common evolution equation that reproduces many predictions of the classical Kolmogorov theory. In the spectral domain, the quadratic advective nonlinearity of the Navier–Stokes equations appears as a convolution, which is often calculated using pseudospectral collocation. An implicit dealiasing method, which removes spurious contributions from wave beating in these convolutions more efficiently than conventional dealiasing techniques, is investigated. Even with efficient dealiasing, the simulation of highly turbulent flow is still a formidable task. Decimation schemes such as spectral reduction replace the many degrees of freedom in a turbulent flow by a limited set of representative quantities. A new method called multispectral reduction is proposed to overcome a significant drawback of spectral reduction: the requirement that all scales be decimated uniformly. Multispectral reduction, which exploits a hierarchy of synchronized spectrally reduced grids, is applied to both shell models and two-dimensional incompressible turbulence.

Acknowledgements

I would like to acknowledge the support of my supervisor, Dr. John Bowman, for his help during my studies at the University of Alberta. I chose an ambitious project, and it was only through his continued support that the research was able to continue. Likewise, I would like to thank the other members of my supervisory committee for participating in my candidacy examination, and my thesis committee for reading this dissertation. Special thanks go to Dr. Remkes Kooistra, for reading early thesis drafts and sharing his scotch.

Dr. Bruno Eckhardt is also deserving of thanks for hosting me in his work group at Philipps Universität Marburg. This visit was financially supported by Education Abroad at the University of Alberta. I also received support from the Kavli Institute for Theoretical Physics at the University of California at Santa Barbara. I also thank Dr. Al Weiss for his help with the number-theoretic aspects of the raw spectrum.

This research was supported in part by the computing resources provided by WestGrid and Compute/Calcul Canada.

I would like to thank the Natural Sciences and Engineering Research Council of Canada and the Department of Mathematical and Statistical Sciences at the University of Alberta for financial support and our graduate secretary Tara Schuetz for her hard work.

Finally, I thank my family and friends for their support and camaraderie during my graduate program.

Table of Contents

1 Introduction **1**

1.A Outline 2

1.B The Navier–Stokes Equations 3

1.C Shell Models of Turbulence 25

1.C.1 The DN Model 27

1.C.2 The GOY Model 28

1.C.3 The Sabra Model 33

2 Analytic Results from Shell Models of Turbulence **35**

2.A Low-resolution shell models 36

2.A.1 DN model 36

2.A.2 GOY model 37

2.B High-resolution limits 38

2.B.1 DN model 39

2.B.2 GOY model 40

2.B.3 Flux Formulation 41

2.B.4 Higher-Order Approximations 41

2.B.5 Steady-State Solutions 42

2.B.6 Recursive Solutions 46

2.C Simulations 49
2.C.1 Dissipation 49
2.C.2 Energy Spectra of Simulations with White-Noise Forcing 50
2.C.3 Structure Functions and Anomalous Scaling 52

3 Implicitly Dealiased Convolutions 56
3.A Dealiasing Binary Convolutions 59
3.A.1 Dealiasing Binary Convolutions via Phase-Shift Dealiasing 59
3.A.2 Dealiasing Binary Convolutions via Explicit Zero Padding 61
3.A.3 Dealiasing Binary Convolutions via Implicit Zero Padding 64
3.B Dealiasing n-ary Convolutions 74
3.B.1 Dealiasing n-ary Convolutions via Zero Padding 78
3.C Comparison of Dealiasing Techniques 79
3.C.1 Non-Centered Convolutions 79
3.C.2 Centered Convolutions 83
3.D Parallel Computation of Implicitly Dealiased Convolutions . . 87

4 Spectral Reduction 91
4.A Spectral Reduction of Shell Models of Turbulence 93
4.A.1 Binned Spectral Reduction 96
4.A.2 Interpolated Spectral Reduction 97
4.A.3 Simulations 100
4.A.4 Heuristic explanation of instabilities arising from interpolation 107
4.B Spectral Reduction of Navier–Stokes Turbulence 110
4.B.1 Binned Spectral Reduction 111
4.C Spectral Reduction of Complex Modes with Uncorrelated Phase 114

5 Multispectral Reduction **116**

5.A General Properties of Synchronization Operators 120

5.B Multispectral Reduction of Shell Models of Turbulence 123

5.B.1 Synchronization Schemes 124

5.B.2 Simulations 126

5.C Multispectral Reduction of 2D Incompressible Navier–Stokes Turbulence 129

5.C.1 Coincident Mode Synchronization 130

5.C.2 Binned Mode Synchronization 148

6 Conclusion **161**

6.A Summary of Results 161

6.B Future Work 165

Bibliography **168**

Index **173**

List of Tables

3.1 Comparison of terms contributing to the first component of the ternary convolution $*(f,g,h)_k = \sum_{a,b,c=-1}^{1} f_a g_b h_c \, \delta_{a+b+c,k}$ and the double binary convolution $(f * (g * h))_k = \sum_{a,\ell=-1}^{1} f_a (g * h)_\ell \delta_{a+\ell,k}$, with $(g * h)_\ell = \sum_{b,c=-1}^{1} g_b h_c \, \delta_{b+c,\ell}$. 77

3.2 Comparison of methods for dealiasing one-dimensional non-centered complex binary convolutions of length m. 80

3.3 Comparison of methods for dealiasing d-dimensional non-centered complex non-centered binary convolutions of size m^d. 81

3.4 Comparison of explicit and implicit methods for dealiasing d-dimensional n-ary non-centered complex convolutions on data of size m^d. 83

3.5 Comparison of methods for dealiasing one-dimensional centered Hermitian convolutions of length m. 84

3.6 Comparison of methods for dealiasing d-dimensional centered Hermitian convolutions of size $(2m-1)^{d-1} \times m$. 85

3.7 Comparison of methods for dealiasing d-dimensional n-ary centered Hermitian convolutions on data of size $(2m-1)^{d-1} \times m$. 86

List of Figures

1.1 Fjørtoft diagram, showing the transfer of energy and enstrophy injected at intermediate wavenumber k_2, which is then transported to lower wavenumber k_1 and higher wavenumber k_3. . . 11

1.2 Equipartition spectrum for the two-dimensional Navier–Stokes equation, with equipartition spectrum parameters $\alpha = 15$, $\beta = 1$. 18

1.3 Retained wave vectors for the vorticity-based two-dimensional Navier–Stokes. 19

1.4 Diagram of distribution of binned energy shells. 21

1.5 Binned energy spectrum for an inviscid, unforced run at late times, with equipartition spectrum parameters $\alpha = 15$, $\beta = 1$. 22

1.6 Diagram of distribution of radii for the raw spectrum. 22

1.7 Number of radii achieved as a function of grid size m. 24

1.8 Distribution of modes for shell models of turbulence. 26

1.9 Equipartition spectrum on the DN model. 27

1.10 The energy spectrum of the DN model with white-noise forcing at low wavenumbers and molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $a = -1/4$ $b = -1$, and 32 shells. 28

1.11 Structure function exponents ζ_p of the DN model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $a = -1/4$ $b = -1$, and 32 shells. Dashed lines show the predictions of Kolmogorov theory. 29

1.12 A comparison of the energy spectrum and flux spectrum for the forced-dissipative GOY model. 31

1.13 The equipartition spectrum of the inviscid, unforced GOY model. 31

1.14 The energy spectrum of the GOY model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $\alpha = 1$ $b = -1/2$, and 32 shells. 32

1.15 Structure function exponents ζ_p of the GOY model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $\alpha = 1$ $b = -1/2$, and 32 shells. Dashed lines show experimental values. 32

1.16 The equipartition spectrum of the inviscid, unforced Sabra model. 33

1.17 The energy spectrum of the Sabra model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $\alpha = 1$ $b = -1/2$, and 32 shells. 34

1.18 Structure function exponents ζ_p of the Sabra model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $\alpha = 1$ $b = -1/2$, and 32 shells. Dashed lines show the predictions of Kolmogorov theory. 34

2.1 Recursive solution of the complex-valued DN model, with constant, real boundary conditions. Imaginary contribution shown dotted. 47
2.2 Recursive solution of the real-valued DN model. 48
2.3 Recursive solution of the GOY model. 49
2.4 Rescaled DN model real velocities, $ia = -1$, $b = 0$. 50
2.5 Energy spectra DN simulations with different values of ν. The vertical dashed lines are the dissipation wavenumber as given in equation (2.29), and the vertical dotted lines are the dissipation wavenumber defined as being the median wavenumber for energy dissipation. 51
2.6 Energy spectra from rescaled DN model with $a = -1/4$, $b = -1$. 51
2.7 Rescaled energy for rescaled DN model with $a = -1$, $b = -1$. . 52
2.8 Structure function exponents for $p = 2, 3, 4, 5$ for the DN model, $ia = -1$, $b = 0$, $\nu = 10^{-8}$. Shell spacing factors are $\lambda = 2$ (blue) and $\lambda = 2^{1/2^5}$ (red). Here, ζ_p is shown for $p = 4, 3, 2$, top to bottom. 53
2.9 Structure function exponents for the DN model with complex velocity, $a = -1$, $b = -1$ $\lambda = 2$, and $\nu = 10^{-8}$. Here, ζ_p is shown for $p = 6, 5, 4, 3, 2$, top to bottom. 53
2.10 Structure function exponents for the DN model with complex velocity, $a = -1$, $b = -1$, $\nu = 10^{-8}$, $\lambda = 2^{1/2}$. Here, ζ_p is shown for $p = 6, 5, 4, 3, 2$, top to bottom. 54
2.11 Structure function exponents for the DN model with complex velocity, $(a, b) = (-1, 1)$, $bc_0 = 1$, $\lambda = 2$, and $\nu = 10^{-8}$ $\lambda = 2^{1/2^2}$. Here, ζ_p is shown for $p = 6, 5, 4, 3, 2$, top to bottom. . . . 54

2.12 Structure function exponents for the DN model with complex velocity, $(a, b) = (-1, 1)$, $bc_0 = 1$, $\lambda = 2$, and $\nu = 10^{-8}$ $\lambda = 2^{1/2^3}$. Here, ζ_p is shown for $p = 6, 5, 4, 3, 2$, top to bottom. . . . 55

3.1 Comparison of computation times for explicitly and implicitly dealiased complex in-place 1D convolutions of length m. . . . 80

3.2 Comparison of computation times for explicitly and implicitly dealiased complex in-place 2D convolutions of size m^2. 82

3.3 Comparison of computation times for explicitly and implicitly dealiased complex non-centered in-place 3D convolutions of size m^3. 82

3.4 Comparison of computation times for explicitly and implicitly dealiased centered Hermitian in-place 1D convolutions of length m. 84

3.5 Comparison of computation times for explicitly and implicitly dealiased centered Hermitian in-place 2D convolutions of size $(2m - 1) \times m$. 86

3.6 Demonstration of the x transform involved in a parallel, transpose-free FFT-based two-dimensional convolution of implicitly zero-padded data. Dark gray indicates input data, with output captured in the both light and dark gray blocks. 88

3.7 Demonstration of the y transform involved in a parallel, FFT-based two-dimensional convolution of implicitly 1/2-padded zero-padded data. Dark- and medium- gray blocks represent input data to the y transform, with output stored in all blocks in the Figure. 90

4.1 Energy spectra from the spectrally reduced DN model with constant σ_n, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$. 98
4.2 Energy spectra from the spectrally reduced GOY model with constant σ_n, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$. 98
4.3 Rescaled DN energy spectra comparing a control run with once-decimated runs using $\sigma = 1$ and with interpolation. The simulations used $a = -1/4$, $b = -1$, and $\nu = 10^{-7}$. 101
4.4 Rescaled DN energy spectra comparing a control run with decimated runs using with interpolation. The simulations used $a = -1/4$, $b = -1$, and $\nu = 10^{-7}$. 101
4.5 Rescaled GOY energy spectra comparing a control run with once-decimated runs using $\sigma = 1$ and with interpolation. The simulations used $a = -1/4$, $b = -1$, and $\nu = 10^{-7}$. 102
4.6 Reset DN energy spectra for interpolated decimation with $\sigma \in (0.52, 3.09)$ for three levels of decimation, with $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$. 102
4.7 Rescaled and reset DN energy spectra of comparing a control run with once-decimated runs using $\sigma = 1$, with interpolation, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$. 103
4.8 Rescaled and reset GOY energy spectra of comparing a control run with once-decimated runs using $\sigma = 1$, with interpolation, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$ 104
4.9 Energy spectra for the interpolated DN model with $\sigma \in (0.0026, 3.09)$, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$. 105
4.10 Reset DN energy spectra for interpolated decimation with $\sigma \in (0.52, 3.09)$ for three levels of decimation. 105

4.11 Reset GOY energy spectra for interpolated decimation with $\sigma \in (0.0026, 3.09)$ for three levels of decimation, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$. 106
4.12 Reset GOY energy spectra for interpolated decimation with $\sigma \in (0.52, 3.09)$ for three levels of decimation, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$. 106
4.13 Schematic diagram of the instability in DN→DN interpolated simulations. 108
4.14 Polar wavenumber bin geometry. 113

5.1 Stage one of serial advancement: advancing the grid from time t to time $t + \tau$. 119
5.2 Stage two of serial advancement: projecting from the grid to the subgrid. 119
5.3 Stage three of serial advancement: advancing the subgrid from time t to time $t + \tau$. 119
5.4 Stage four of serial advancement: prolong from the subgrid to the grid. 120
5.5 Stage one of parallel advancement: advancing both the grid and the subgrid from time t to time $t + \tau$. 120
5.6 Stage two of parallel advancement: synchronizing the grid and the subgrid simultaneously. 121
5.7 The DN shell model with nonlinear interactions represented by arrows. 124
5.8 Nonlinear source diagram for the multispectral DN model. . . 124
5.9 Projection diagram for the multispectral DN model. 125

5.10 Prolongation diagram for the multispectral DN model. 126
5.11 Parallel synchronization of the multispectral DN model. 126
5.12 Energy spectrum for the inviscid unforced multispectral DN model with two grids and serial synchronization, exhibiting a k^{-1} spectral slope. 127
5.13 Multispectral simulation of the forced-dissipative DN model with two grids and serial synchronization. 127
5.14 Multispectral simulation of the forced-dissipative DN model, with three grids and serial synchronization. 128
5.15 Multispectral simulation of the forced-dissipative DN model, with four grids and serial synchronization. 129
5.16 Grid and subgrid geometry for the oblique coincident-mode multispectral method. Black dots indicate grid modes and blue circles indicate subgrid modes. 131
5.17 Grid and subgrid geometry for the radix-two coincident-mode multispectral method. Black dots indicate grid modes and blue circles indicate subgrid modes. 132
5.18 Grid and subgrid geometry for the radix-three coincident-mode multispectral method. Black dots indicate grid modes and blue circles indicate subgrid modes. 132
5.19 The energy spectrum of an inviscid unforced serially synchronized coincident-mode multispectral simulation of the two-dimensional Navier–Stokes equations, showing only stochastic deviations from the theoretical equipartition spectrum, with $\alpha = 25$, $\beta = 1$. The grid and subgrid have dimension 15×8. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$. 136

5.20 Energy spectrum from a forced-dissipative serially synchronized coincident-mode two-dimensional Navier–Stokes simulation with an oblique subgrid, showing an energy pile-up at the grid/subgrid interface. The grid and subgrid have dimension 63×32. The simulation was performed with $\nu = 5 \times 10^{-4}$, and white-noise forcing with unit enstrophy injection rate on modes with $k \in [2, 4]$. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$. . 137

5.21 Energy spectrum from a forced-dissipative serially synchronized coincident-mode two-dimensional Navier–Stokes simulation with radix two, using the parameters of Figure 5.20. An increased energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$. 138

5.22 Energy spectrum from a forced-dissipative serially synchronized coincident-mode two-dimensional Navier–Stokes simulation with radix three, using the parameters of Figure 5.20. A large energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$. 138

5.23 The energy spectrum of an unforced, inviscid coincident-mode parallel-synchronized multispectral simulation of the two-dimensional Navier–Stokes equations showing only stochastic deviations from the theoretical equipartition spectrum, with $\alpha = 25$, $\beta = 1$. The grid and subgrid have dimension 15×8. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$. 141

5.24 Energy spectrum from a forced-dissipative parallel-synchronized coincident-mode two-dimensional Navier–Stokes subgrid simulation with an oblique subgrid, using a grid and subgrid of dimension 63×32, along with the parameters of Figure 5.20. An energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$. 141

5.25 Energy spectrum from a forced-dissipative parallel-synchronized coincident-mode two-dimensional Navier–Stokes simulation with radix two, using the parameters of Figure 5.24. An increased energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$. 142

5.26 Energy spectrum from a forced-dissipative parallel-synchronized coincident-mode two-dimensional Navier–Stokes simulation with radix three, using the parameters of Figure 5.24. A large energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$. 142

5.27 Fourier lattice modes in the radix-two decimated DNS. All modes are contained in one variably decimated grid. 143

5.28 Fourier lattice modes in the oblique decimated DNS. All modes are contained in one variably decimated grid. 144

5.29 Fourier lattice modes in the radix-three decimated DNS. All modes are contained in one variably decimated grid. 145

5.30 Energy spectrum from an inviscid unforced serially synchronized oblique decimated DNS simulation, with $\alpha = 25$ and $\beta = 1$, using a grid and subgrid of dimension 63×32. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$. 145

5.31 Energy spectrum from a forced-dissipative serially synchronized oblique decimated DNS simulation, using the parameters of Figure 5.20. An energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$. 146

5.32 Energy spectrum from a forced-dissipative serially synchronized radix-two decimated DNS simulation, using the parameters of Figure 5.20. An increased energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$. 146

5.33 Grid/subgrid with radix-three decimation on the subgrid. Grid modes are represented by black dots, and subgrid bins are represented by blue squares. 149

5.34 Energy spectrum from an unforced inviscid binned subgrid wavenumber simulation, with $\alpha = 121$, $\beta = 1$. The grid and subgrid have dimension 63×32. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$. 155

5.35 Energy spectrum from an unforced inviscid binned subgrid wavenumber simulation with increased subgrid nonlinearity, using the parameters of Figure 5.34. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$. 156

5.36 Energy spectrum from an unforced inviscid binned subgrid wavenumber simulation with decreased subgrid nonlinearity, using the parameters of Figure 5.34. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$. 156

5.37 Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation compared to a full-resolution DNS simulation, using the parameters of Figure 5.20. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$. 157

5.38 Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with increased subgrid source term compared to a full-resolution DNS simulation, using the parameters of Figure 5.20. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$. 158

5.39 Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with decreased subgrid source term compared to a full-resolution DNS simulation, using the parameters of Figure 5.20. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$. 158

5.40 Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with a grid of size 15×8, a subgrid of size 63×32, and a circular grid boundary compared to a full-resolution DNS simulation. Other parameters are as in Figure 5.20. The grid/subgrid boundary occurs at $k = 7$. 159

5.41 Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with a grid of size 15×8, a subgrid of size 63×32, a circular grid boundary, and an increased subgrid source compared to a full-resolution DNS simulation. Other parameters are as in Figure 5.20. The grid/subgrid boundary occurs at $k = 7$. 160

5.42 Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with a grid of size 15×8, a subgrid of size 63×32, a circular grid boundary, and a decreased subgrid source compared to a full-resolution DNS simulation. Other parameters are as in Figure 5.20. The grid/subgrid boundary occurs at $k = 7$. 160

Chapter 1

Introduction

Wherein the models under consideration are introduced, and the questions and problems to be addressed are discussed.

Most of the matter in the universe is undergoing chaotic motion in response to external forces and internal damping. This is equally true if we consider the blood running through our veins, stellar plasma, or the thin dust from which all of this was formed.

This behaviour is called turbulence, and is characterized by activity at many scales, from waves the size of the system, and through all intermediate scales down to the relatively microscopic scales where the motion is diminished by friction. Despite the relative simplicity of the equations that we use to describe this motion and the near ubiquity of the phenomenon itself, our understanding of turbulence is surprisingly incomplete and we often lack the ability to make accurate probabilistic predictions of turbulent systems analytically. Moreover, the huge range of spatial and temporal scales involved in modelling such systems vastly outstrips the ability of modern computers,

and will continue to do so for the foreseeable future. This complexity leaves us unable to predict basic properties that are fundamental to both practical applications and our understanding of the natural world.

1.A Outline

This dissertation is divided into six chapters. Chapter 1 introduces the reader to the models under consideration, namely the Navier–Stokes equations and shell models of turbulence, and various properties of these models are discussed.

In Chapter 2, we demonstrate that the standard shell-models of turbulence produce the same evolution equation under the continuum limit, and that this limit reproduces many predictions of classical Kolmogorov theory.

The Navier–Stokes equations possess a quadratic nonlinear term, the calculation of which is a difficult and slow step in performing simulations of turbulent systems. In Chapter 3, we describe a computational technique for implicitly dealiasing convolutions, which improves upon the standard zero-padding technique for calculating Fourier-transform based convolutions.

While the implicitly dealiased convolutions of Chapter 3 allow one to perform simulations with significantly less computational effort, the simulation of highly turbulent systems is still too difficult for even the most modern of computer systems, and will remain so for the foreseeable future. Chapter 4 describes spectral reduction, which is a decimation scheme for turbulent systems that allows one to perform simulations with greatly decreased effort.

Unfortunately, spectral reduction requires that one decimate at all length scales, which is a significant drawback to the method. This problem may

be overcome by using the method of multispectral reduction, which is the subject of Chapter 5. The multispectral method is a numerical technique that makes use of a hierarchy of spectrally reduced grids in Fourier space, allowing one to decimate non-uniformly. In principle, this method is applicable to all time-dependent partial differential equations that are amenable to spectral reduction.

The above-mentioned results are summarized in Chapter 6, in which open problems and future work are also discussed.

1.B The Navier–Stokes Equations

The Eulerian velocity of an infinitesimal fluid parcel with position $\boldsymbol{x} = (x, y, z)$ is denoted $\boldsymbol{u}(\boldsymbol{x}, t) = (u(\boldsymbol{x}, t), v(\boldsymbol{x}, t), w(\boldsymbol{x}, t))$. The motion of this fluid parcel is effected by *advection*, *pressure*, *viscosity*, and *external forces*.

Let ρ be the density of the fluid. The change in the momentum per unit volume is given by

$$\frac{\partial(\rho \boldsymbol{u})}{\partial t} = \sum \text{Forces per unit volume}, \tag{1.1}$$

where the right-hand side is a sum over the advective, pressure, viscous, and external forces. In the case where $\rho(\boldsymbol{x}, t) = \rho$, (i.e. constant density) one can divide the external forces by the density to isolate the time derivative of $\boldsymbol{u}$ on the left-hand side of equation (1.1), yielding

$$\frac{\partial \boldsymbol{u}}{\partial t} = \sum \text{Forces per unit mass}. \tag{1.2}$$

The advective force can be represented via a *Lagrangian derivative*,

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}, \tag{1.3}$$

which represents how the fluid parcel reacts to the motion of fluid parcels in its vicinity; i.e. the fluid parcel, in absence of other forces, will move with the fluid around it.

The fluid parcel will be accelerated by pressure forces. Considering pressure and advective forces, the equation of motion is

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho} \nabla P, \tag{1.4}$$

where $P(\boldsymbol{x}, t)$ is the pressure. Note that equation (1.4) is a three-dimensional equation (i.e. it has x, y, and z components), but contains four unknowns, namely u, v, w, and P. As such, an *equation of state*, which relates the pressure field to the velocity is required to close equation (1.4). In this monograph, we restrict ourselves to *incompressible* flows; that is, we insist that the density has zero Lagrangian derivative. In other words, the fluid is incompressible if and only if

$$\frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = 0. \tag{1.5}$$

This, in combination with the *continuity equation*

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{1.6}$$

which guarantees conservation of mass, implies that

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1.7}$$

everywhere. If the initial conditions are such that ρ is initially constant (which we will assume for the remainder of this monograph), then incompressibility guarantees that ρ is constant at all time. Thus, without loss of generality, let $\rho \equiv 1$. Equation (1.7) is particularly advantageous when performing simulations, as it allows one to reduce the dimensionality of the problem.

Most (though not all) of the models that we will consider include friction. For fluids, this is known as the kinematic viscosity, governed by the physical parameter ν. In the case of *molecular viscosity*, the velocity changes as

$$\nu \nabla^2 \boldsymbol{u}. \tag{1.8}$$

Molecular viscosity preferentially diminishes small-scale features of the flow field. Some models incorporate *hyperviscosity*, in which ∇^2 is replaced by $(-1)^{p+1}\nabla^{2p}$ for some integer $p > 1$.

In some cases (for example bounded two-dimensional simulations) there are two viscous terms that remove energy from both the large and small scales. One typically uses two coefficients ν_{L} to denote the viscosity that damps large scales (i.e. low wavenumbers) and ν_{H} to denote the viscosity that damps small scales (i.e. high wavenumbers). The effect of this on the velocity field is given by

$$-\nu_{\mathrm{L}} \boldsymbol{u} + \nu_{\mathrm{H}} \nabla^2 \boldsymbol{u}, \tag{1.9}$$

where we have chosen a constant damping at the low wavenumbers and molecular viscosity to damp high wavenumbers.

Finally, models often include an external body force, be it gravity, the influence of a rotating reference frame, or a more purely mathematical construct. We denote such force $\boldsymbol{F}(\boldsymbol{x}, t)$. For the purposes of this monograph, we generally consider $\boldsymbol{F}$ to be a white-noise banded force, i.e. one that is delta-correlated in time and active over only a certain range of scales. This allows one to control the energy injection rate (Novikov [1964]), which is useful for testing statistical theories of fluid motion.

By combining the forces of advection, pressure, viscosity, and the external force, and assuming the fluid to be incompressible, we arrive at the

incompressible Navier–Stokes equations,

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} &= -\frac{1}{\rho} \nabla P + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{F}, \\ \nabla \cdot \boldsymbol{u} &= 0. \end{aligned} \tag{1.10}$$

This can be formulated in terms for the *vorticity*,

$$\boldsymbol{\omega} \doteq \boldsymbol{\nabla} \times \boldsymbol{u} \tag{1.11}$$

(the notation $\doteq$ is used to emphasize a definition). In the case where $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$, equation (1.11) is invertible up to some function $\boldsymbol{A}$ with $\nabla \times \boldsymbol{A} = \boldsymbol{0}$ since

$$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u} = \boldsymbol{\nabla} \times (\boldsymbol{u} + \boldsymbol{A}). \tag{1.12}$$

The Navier–Stokes equation may be expressed in terms of the vorticity by taking the curl of equation (1.10), which yields

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \nu \nabla^2 \boldsymbol{\omega} + \boldsymbol{\nabla} \times \boldsymbol{F}. \tag{1.13}$$

One closes the equation by inverting Equation (1.11) to obtain $\boldsymbol{u}(\boldsymbol{\omega})$.

Properties of the Navier–Stokes Equations

The incompressible Navier–Stokes equations conserve energy when both the forcing and viscosity are zero. The energy is defined as

$$E \doteq \frac{1}{2} \int_{\Omega} |\boldsymbol{u}(\boldsymbol{x}, t)|^2 \, d\boldsymbol{x}, \tag{1.14}$$

where Ω is the physical domain of the fluid. One can prove that energy is conserved (assuming periodic boundary conditions and that the velocity is

uniformly continuous) by taking the time derivative of the energy:

$$\begin{aligned}
\frac{dE}{dt} &= \frac{1}{2}\frac{d}{dt}\int_\Omega |\boldsymbol{u}(\boldsymbol{x},t)|^2 \, d\boldsymbol{x} \\
&= \int_\Omega \boldsymbol{u}\cdot\frac{\partial \boldsymbol{u}}{\partial t}\, d\boldsymbol{x} \\
&= \int_\Omega \boldsymbol{u}\cdot(-\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla}P)\, d\boldsymbol{x} \\
&= -\int_\Omega \boldsymbol{u}\cdot\boldsymbol{\nabla}\left(\frac{|\boldsymbol{u}|^2}{2}+P\right) d\boldsymbol{x} \\
&= \int_\Omega (\boldsymbol{\nabla}\cdot\boldsymbol{u})\left(\frac{|\boldsymbol{u}|^2}{2}+P\right) d\boldsymbol{x} \\
&= 0
\end{aligned} \tag{1.15}$$

since $\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$.

In three dimensions, equation (1.10) admits another non-trivial invariant called *helicity*. The helicity is defined as

$$H \doteq \frac{1}{2}\int_\Omega \boldsymbol{u}\cdot(\boldsymbol{\nabla}\times\boldsymbol{u})\, d\boldsymbol{x}, \tag{1.16}$$

and is conserved when ν and $\boldsymbol{F}$ are zero since (e.g., Frisch [1995])

$$\frac{dH}{dt} = \nu\int_\Omega [\boldsymbol{\omega}\cdot(\boldsymbol{\nabla}\times\boldsymbol{\omega})]\, d\boldsymbol{x}. \tag{1.17}$$

Helicity is not a sign-definite quantity, so it doesn't play as obvious a rôle in determining the behaviour of a fluid system as, for example, energy.

In two dimensions, helicity is conserved trivially since $\boldsymbol{\omega}$ and $\boldsymbol{u}$ are perpendicular, and thus $H = 0$. Two-dimensional inviscid flows also conserve the *enstrophy*, defined as

$$Z = \frac{1}{2}\int_\Omega |\boldsymbol{\omega}|^2 \, d\boldsymbol{x}, \tag{1.18}$$

which is conserved since

$$\begin{aligned}\frac{dZ}{dt} &= \int_\Omega \boldsymbol{\omega} \cdot \frac{\partial \boldsymbol{\omega}}{\partial t}\, d\boldsymbol{x} \\ &= \int_\Omega \boldsymbol{\omega} \cdot (-\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{\omega}\, d\boldsymbol{x} \\ &= \int_\Omega (\boldsymbol{\nabla} \cdot \boldsymbol{u}) \frac{|\boldsymbol{\omega}|^2}{2}\, d\boldsymbol{x} \\ &= 0\end{aligned} \tag{1.19}$$

since $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$. Since enstrophy is positive-definite, it plays an important role in two-dimensional fluid mechanics. Two-dimensional turbulence also conserves an infinitude of other quantities known as *Casimir invariants*, which are arbitrary L^1 functions of vorticity. However, these invariants are lost in finite-resolution simulations, and one typically focuses primarily on the conservation of energy and enstrophy alone.

Solutions to the Navier–Stokes equations are invariant under the map

$$(t, \boldsymbol{x}, \boldsymbol{u}, \nu) \to \left(\gamma t, \lambda \boldsymbol{x}, \frac{\lambda}{\gamma}\boldsymbol{u}, \frac{\lambda^2}{\gamma}\nu\right), \tag{1.20}$$

for arbitrary non-zero λ and $\gamma \in \mathbb{R}$. If U is a characteristic velocity and L a characteristic length scale, then the *Reynolds number*

$$R = \frac{UL}{\nu} \tag{1.21}$$

is invariant under the mapping in equation (1.20). This allows one to characterize solutions to the Navier–Stokes equations by considering all solutions with equal Reynolds number. Low Reynolds-number solutions are dominated by viscosity, whereas high Reynolds-number solutions are more influenced by the nonlinear forces. At low Reynolds number, the fluid will move laminarly, with the nonlinear part of equation (1.10) not playing an important rôle. When the Reynolds number reaches a certain value, the nonlinear term will suddenly

become more active, causing the fluid to move in complicated fashions and energy to be transported to smaller scales, where it is dissipated. This behaviour is called *turbulence*. As it is ubiquitous, we have been aware of the existence of turbulence for many centuries; as it is complex and irregular, our understanding of turbulence falls far from what one would otherwise expect of such an important, ancient, and omnipresent phenomenon.

Our modern understanding of turbulence is founded on the work done by A.N. Kolmogorov [Kolmogorov 1941] in which he considered the transfer of energy between scales to be a *cascade*, wherein energy moves from larger scales to smaller scales via intermediate scales. The viscous term in the Navier–Stokes equation is predominantly active at the small scales, where it acts to remove energy, while energy injection (via forcing) is typically restricted to large scales. In statistically steady three-dimensional turbulence, the energy is transferred by the nonlinear (advective) term from the large scales to the small scales. The region of Fourier space in which this energy transfer takes place, in the absence of forcing and dissipation, is called the *inertial range*.

For scales within the inertial range, viscous forces are sub-dominant. Let ϵ be the energy injection rate per unit mass, and $u(k)$ be the typical velocity at wavenumber k. The *energy spectrum* $E(k)$ is defined by the equation $\int E(k)\,dk = E$. Kolmogorov argued that only the typical velocity magnitude u, the energy injection rate ϵ and the wavenumber k are relevant in the inertial range. Dimensional consistency gives $\epsilon \sim u^3 k$, so $u \sim (\epsilon/k)^{1/3}$ and $u^2 \sim (\epsilon/k)^{2/3}$. But $u^2 = 2\int E(k)\,dk$ has the same dimensions as $kE(k)$, so

$$E(k) = Ck^{-1}\left(\frac{\epsilon}{k}\right)^{\frac{2}{3}} = C\epsilon^{\frac{2}{3}}k^{-\frac{5}{3}} \tag{1.22}$$

for some dimensionless quantity C.

The $k^{-5/3}$ energy spectrum was predicted by Kolmogorov [1941], who conjectured that C is a universal constant for all fully developed turbulent flows. He also established formulae for structure functions of turbulent flows, namely the parallel velocity increments

$$\delta \boldsymbol{u}_{\parallel}(\boldsymbol{x}, \boldsymbol{\ell}) \doteq (\boldsymbol{u}(\boldsymbol{x} + \boldsymbol{\ell}) - \boldsymbol{u}(\boldsymbol{r})) \cdot \hat{\boldsymbol{\ell}}, \tag{1.23}$$

where $\hat{\boldsymbol{\ell}}$ is a unit vector parallel to $\boldsymbol{\ell}$. Letting $\langle \dots \rangle \doteq \int \dots d\boldsymbol{x} / \int d\boldsymbol{x}$, Kolmogorov used isotropy to determine a scaling for the p^{th} order structure function

$$\begin{aligned} S_p(\ell) &\doteq \left\langle \left| \delta \boldsymbol{u}_{\parallel}(\boldsymbol{x}, \boldsymbol{\ell}) \right|^p \right\rangle \\ &= C_p \epsilon^{\frac{p}{3}} \ell^{\frac{p}{3}} \\ &\sim \ell^{\zeta_p}, \end{aligned} \tag{1.24}$$

where ζ_p is the p^{th}-order *structure function exponent*. Moreover, Kolmogorov's prediction that $\zeta_p = p/3$ this was shown to be exact for $p = 3$. Equation (1.24) produces, with some additional arguments, the Kolmogorov energy spectrum with more rigour than mere dimensional analysis.

While Kolmogorov's argument is exact for $p = 3$, other cases are subject to corrections known as *anomalous scaling*. Both numerical and physical experiments show a deviation from $\zeta_p = p/3$ for $p \neq 3$, with ζ_1 and ζ_2 being less than predicted, and ζ_p being higher than predicted for $p \geq 4$.

The energy spectrum is divided into three parts: the energy-injection region, which encompasses the scales at which the external force is active; the inertial range, in which energy is transferred from large scales to small scales, and the dissipation range, in which kinetic energy is removed from the system. The dissipation range has a characteristic scale given by the *Kolmogorov dissipation scale*,

$$\eta_d \doteq \left(\frac{\nu^3}{\epsilon} \right)^{\frac{1}{4}}. \tag{1.25}$$

This determines the smallest scales that are active in a turbulent system. When ν is generally small, η_d is also quite small, producing a very wide range between the largest and smallest active scale, with no quiescent intermediate scale. The time-scale corresponding to velocity fluctuations that obey Kolmogorov scaling with length scale near η_d is

$$\tau_d \doteq \left(\frac{\nu}{\epsilon}\right)^{\frac{1}{2}}. \tag{1.26}$$

Turbulent systems with high Reynolds number (i.e. small ν) have a wide range of physical and temporal scales.

In two dimensions, the situation is complicated by the presence of two quadratic positive-definite invariants, namely energy and enstrophy. Let ϵ be the energy injection rate and η be the enstrophy injection rate due to the external forcing $\boldsymbol{F}$. The energy cascade of two-dimensional turbulence was first considered by Kraichnan, Leith, and Batchelor, who considered truncated two-dimensional Navier–Stokes systems [Kraichnan 1967], [Leith 1968], [Batchelor 1969], [Kraichnan 1971], [Kraichnan 1975]. The picture of the energy and enstrophy cascades, first proposed by Fjørtoft [1953], is shown in Figure 1.1. Energy (and hence enstrophy) is injected into the central set of wave

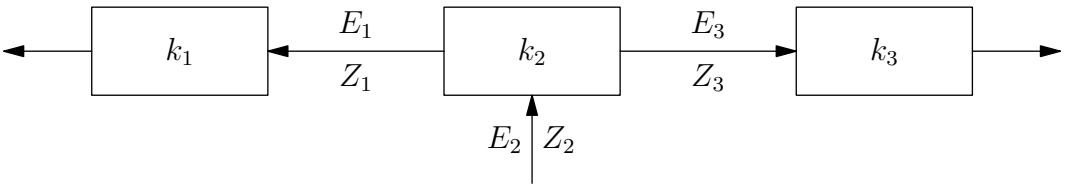

Figure 1.1: Fjørtoft diagram, showing the transfer of energy and enstrophy injected at intermediate wavenumber k_2, which is then transported to lower wavenumber k_1 and higher wavenumber k_3.

vectors with characteristic wavenumber k_2. Nonlinear interactions between the wave vector sets will move an amount of energy E_1 from the middle set with characteristic wavenumber k_2 to the left set with characteristic wavenumber k_1. The characteristic wavenumbers are defined as satisfying $k_i^2 = E_i/Z_i$. This redistribution of energy carries with it a redistribution of enstrophy, which we label Z_1. Similarly, energy E_3 and enstrophy Z_3 moves from the middle set to the right set with characteristic wavenumber k_3. If k_1, k_2, and k_3 are typical wavenumber for the boxes, then the equations describing the balance of energy and enstrophy transfers are

$$E_2 = E_1 + E_3, \tag{1.27}$$

$$Z_2 = Z_1 + Z_3, \tag{1.28}$$

that is,

$$E_2 = E_1 + E_3, \tag{1.29}$$

$$k_2^2 E_2 = k_1^2 E_1 + k_3^2 E_3. \tag{1.30}$$

This implies that

$$E_1 = E_2 \frac{k_3^2 - k_2^2}{k_3^2 - k_1^2}. \tag{1.31}$$

If, for example, $k_1 = k$, $k_2 = 2k$, and $k_3 = 4k$, then we should expect

$$\begin{aligned} E_1 &= \frac{4}{5} E_2, \\ E_3 &= \frac{1}{5} E_2. \end{aligned} \tag{1.32}$$

Thus, we expect to see a *dual cascade*, with energy moving primarily to lower wavenumbers and enstrophy moving to higher wavenumbers. In an unbounded domain, the modes approaching $\boldsymbol{k} = \boldsymbol{0}$ act as an infinite reservoir that the energy cascade cannot saturate. In this case, the infinite-reservoir allows an inverse cascade of energy, and one expects a $k^{-5/3}$ inverse cascade and a k^{-3} direct cascade.

The Spectral Navier–Stokes Equations

The *spectral Navier–Stokes* equations are the result of performing a *Fourier transform* on the incompressible Navier–Stokes equations given in equation (1.10). That is, we consider the Fourier transform $\boldsymbol{u}_{\boldsymbol{k}}$ of $\boldsymbol{u}(\boldsymbol{x})$,

$$\boldsymbol{u}_{\boldsymbol{k}} \doteq \int \boldsymbol{u}(\boldsymbol{x}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}}\, d\boldsymbol{x}, \tag{1.33}$$

where $\boldsymbol{k}$ is the wave vector for the mode with complex amplitude $\boldsymbol{u}_{\boldsymbol{k}}$. Since the $\boldsymbol{x}$-space data $\boldsymbol{u}(\boldsymbol{x})$ is real-valued, the Fourier-transformed data exhibits *Hermitian symmetry*, i.e.

$$\boldsymbol{u}_{-\boldsymbol{k}} = \boldsymbol{u}_{\boldsymbol{k}}^{*}, \tag{1.34}$$

where $(\dots)^*$ denotes complex-conjugation. Taking the divergence of the Navier–Stokes equations and applying $\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$ results in the equation

$$-\nabla^2 P = \boldsymbol{\nabla}\cdot[(\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}], \tag{1.35}$$

the solution to which yields the pressure field from the velocity. On using this constraint to eliminate the pressure from equation (1.10), the Fourier transform of the three-dimensional incompressible Navier–Stokes equation is

$$\frac{\partial \boldsymbol{u}_{\boldsymbol{k}}}{\partial t} = \left(\boldsymbol{I} - \frac{\boldsymbol{k}\boldsymbol{k}}{k^2}\right) \sum_{\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{k}} i(\boldsymbol{k}\cdot\boldsymbol{u}_{\boldsymbol{p}})\boldsymbol{u}_{\boldsymbol{q}} - \nu k^2 \boldsymbol{u}_{\boldsymbol{k}} + \boldsymbol{F}_{\boldsymbol{k}}. \tag{1.36}$$

For simulations in a periodic domain, with $\boldsymbol{x} \in (2\pi)^d \mathbb{R}^d / \mathbb{Z}^d$, the wave vectors $\boldsymbol{k}$ lie in $\mathbb{Z}^d$. Since

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \quad \Longleftrightarrow \quad \boldsymbol{k} \cdot \boldsymbol{u}_{\boldsymbol{k}} = 0, \tag{1.37}$$

one can eliminate one component of $\boldsymbol{u}_{\boldsymbol{k}}$.

For two-dimensional turbulence, equation (1.37) allows one to eliminate all but one component of the velocity field, and it is particularly efficacious to use the vorticity formulation given in equation (1.13). In two dimensions, $\boldsymbol{\omega} = \omega \hat{\boldsymbol{z}}$, and $\boldsymbol{\omega} \cdot \boldsymbol{u} = 0$, so we may rewrite the vorticity-based Navier–Stokes equation (1.13) as

$$\frac{\partial \omega}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\, \omega = \nu \nabla^2 \omega + \hat{\boldsymbol{z}} \cdot \boldsymbol{\nabla} \times \boldsymbol{F}. \tag{1.38}$$

Denoting $f \doteq \hat{\boldsymbol{z}} \cdot \boldsymbol{\nabla} \times \boldsymbol{F}$, the Fourier transform of equation (1.38) is

$$\frac{\partial \omega_{\boldsymbol{k}}}{\partial t} = \sum_{\boldsymbol{p}} \frac{p_x k_y - p_y k_x}{|\boldsymbol{k} - \boldsymbol{p}|^2} \omega_{\boldsymbol{p}} \omega_{\boldsymbol{k}-\boldsymbol{p}} - \nu k^2 w_{\boldsymbol{k}} + f_{\boldsymbol{k}}, \tag{1.39}$$

the numerical solution of which will be the main subject of Chapter 5.

Equipartition

The vorticity formulation of two-dimensional spectral Navier–Stokes equations with $\nu = 0$ and no forcing have the form

$$\frac{\partial \omega_{\boldsymbol{k}}}{\partial t} = \sum_{\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{k}} \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \omega^*_{\boldsymbol{p}} \omega^*_{\boldsymbol{q}}, \tag{1.40}$$

where

$$\epsilon_{\boldsymbol{kpq}} = (\hat{\boldsymbol{z}} \cdot \boldsymbol{p} \times \boldsymbol{q}) \delta(\boldsymbol{k} + \boldsymbol{p} + \boldsymbol{q}) \tag{1.41}$$

is the nonlinear interaction coefficient. The three-dimensional version conserves energy and helicity, while the two-dimensional version conserves energy and enstrophy. Equation (1.40) can be written as a noncanonical Hamiltonian system, i.e.

$$\frac{\partial \omega_{\boldsymbol{k}}}{\partial t} = \sum_{\boldsymbol{q}} J_{\boldsymbol{kq}} \frac{\partial H}{\partial \omega_{\boldsymbol{q}}} \tag{1.42}$$

where $H = \frac{1}{2}\sum_{\boldsymbol{k}} |\omega_{\boldsymbol{k}}|^2/k^2$ is the Hamiltonian and $J_{\boldsymbol{kq}} = \sum_{\boldsymbol{p}} \epsilon_{\boldsymbol{kpq}}\omega^*_{\boldsymbol{p}}$. Note that $J_{\boldsymbol{kq}} = -J_{\boldsymbol{qk}}$, which implies that

$$\begin{aligned}\sum_{\boldsymbol{k}} \frac{\partial \dot{\omega}_{\boldsymbol{k}}}{\partial \omega_{\boldsymbol{k}}} &= \sum_{\boldsymbol{k},\boldsymbol{q}} \frac{\partial J_{\boldsymbol{kq}}}{\partial \omega_{\boldsymbol{k}}} + J_{\boldsymbol{kq}} \frac{\partial^2 H}{\partial \omega_{\boldsymbol{k}} \partial \omega_{\boldsymbol{q}}} \\ &= \sum_{\boldsymbol{k},\boldsymbol{q}} \epsilon_{\boldsymbol{k}(-\boldsymbol{k})\boldsymbol{q}} + \sum_{\boldsymbol{k},\boldsymbol{q}} J_{\boldsymbol{kq}} \frac{\partial^2 H}{\partial \omega_{\boldsymbol{k}} \partial \omega_{\boldsymbol{q}}} \\ &= 0,\end{aligned} \tag{1.43}$$

where the first sum vanishes because $\epsilon_{\boldsymbol{k}(-\boldsymbol{k})\boldsymbol{q}} = \epsilon_{\boldsymbol{qk}(-\boldsymbol{k})} = 0$ and the second term vanishes since it is the sum of the product of an antisymmetric and a symmetric factor. Equation (1.43) is known as the *Liouville theorem* for equation (1.40), and implies that the system conserves volume in phase-space.

Assuming that equation (1.40) is ergodic, then the entropy

$$S(t) \doteq -\sum_i N_i(t) \log N_i(t), \tag{1.44}$$

where $N_i(t)$ is the number of modes in state i at time t, will increase to a maximum value. By the Gibbs H theorem [Carnevale *et al.* 1981], the system will tend towards a statistical equilibrium that maximizes S subject to the constraint that the invariants must be conserved, i.e. the conserved quantities will be in *equipartition*.

The entropy of the system is constrained by the invariants of the system, namely energy and enstrophy. Let ϵ_i and ξ_i be the energy and enstrophy of each mode in state i, respectively. If a system is ergodic, the Gibbs H theorem states that the equilibrium state of the system is the unique state that maximizes the entropy of the system. Indeed, we expect the entropy of a system to increase, not necessarily monotonically, until it reaches an equilibrium state where the entropy is constant and maximal. However, this extremization must

be subject to the constraints of energy and enstrophy conservation. In general, we may have many conserved quantities, which we label $E_j \doteq \frac{1}{2}\sum_{\boldsymbol{k}} \sigma^j_{\boldsymbol{k}} |\omega_{\boldsymbol{k}}|^2$, for certain functions $\sigma^j_{\boldsymbol{k}}$, where $j = 1, \ldots, n_c$. Each of these provides a constraint $E_j = \sum_i \epsilon_{i,j} N_i$, where $\epsilon_{i,j}$ is the amount of quantity E_j for each mode in state i. In addition, the total number of modes $\sum_i N_i$ is constrained to be N. The equilibrium state will therefore minimize

$$\sum_i N_i \log N_i - \alpha_0 \left(N - \sum_i N_i \right) - \sum_j \alpha_j \left(E_j - \sum_i \epsilon_{i,j} N_i \right), \quad (1.45)$$

where the α_j are Lagrange multipliers. On setting the derivative of (1.45) with respect to N_ℓ to zero and solving for N_ℓ, one obtains the *Gibbs distribution*

$$N_i = \exp\left(-1 - \alpha_0 - \sum_j \alpha_j \epsilon_{i,j} \right). \quad (1.46)$$

If there are N independent complex amplitudes $\omega_{\boldsymbol{k}}$, there will be $2N$ degrees of freedom, which we arbitrarily label by $\kappa = 1, 2, \ldots 2N$. Let $X = (\omega_1, \ldots, \omega_{2N})$ be a point in phase space. On denoting the state of mode κ at X by i_κ, one can compute the probability that the system is in state X:

$$P(X) \propto \prod_\kappa N_{i_\kappa} \propto \exp\left(-\sum_{\kappa,j} \alpha_j \epsilon_{i_\kappa, j} \right) = \exp\left(-\sum_j \alpha_j E_j \right), \quad (1.47)$$

where α_i is determined by the initial conditions of the system. The expected value for the energy in mode $\boldsymbol{k}$ is therefore

$$\left\langle {\omega^{\mathrm{r}}_{\boldsymbol{k}}}^2 \right\rangle = \left\langle {\omega^{\mathrm{i}}_{\boldsymbol{k}}}^2 \right\rangle = \frac{\int {\omega^{\mathrm{r}}_{\boldsymbol{k}}}^2 P(X)\, d\omega_1 \ldots d\omega_{2N}}{\int P(X)\, d\omega_1 \ldots d\omega_{2N}}, \quad (1.48)$$

where $\omega_{\boldsymbol{k}} = \omega^{\mathrm{r}}_{\boldsymbol{k}} + i\omega^{\mathrm{i}}_{\boldsymbol{k}}$. In two dimensions, the conserved quantities are energy, enstrophy, and other (nonquadratic) Casimir invariants associated with infinitesimal parcel rearrangement. Since we choose to work in a finite spectral domain we lose conservation of the Casimir invariants. Only the two quadratic invariants (energy and enstrophy) survive this spectral truncation, so $\{\sigma^i_{\boldsymbol{k}} : i = 1, \ldots, n_c\} = \{1/k^2, 1\}$. Since $\int_{-\infty}^{\infty} e^{-ax^2}\, dx = \sqrt{\pi/a}$, one finds, on accounting for the Hermiticity condition $\omega_{\boldsymbol{k}} = \omega^*_{-\boldsymbol{k}}$, that

$$\left\langle |\omega_{\boldsymbol{k}}|^2 \right\rangle = \left\langle \omega^{\mathrm{r}\,2}_{\boldsymbol{k}} \right\rangle + \left\langle \omega^{\mathrm{i}\,2}_{\boldsymbol{k}} \right\rangle = \frac{1}{\alpha/k^2 + \beta} \tag{1.49}$$

for some constants α and β that are determined by the total energy and enstrophy present in the initial conditions. On noting that $E(k) \doteq 2\pi k \langle \frac{1}{2} |\omega_{\boldsymbol{k}}|^2 \rangle / k^2$, we obtain the equipartition spectrum,

$$E(k) = \pi \frac{k}{\alpha + \beta k^2}, \tag{1.50}$$

which is shown in Figure 1.2.

Three-dimensional turbulence follows a similar argument, except that helicity is conserved, while enstrophy is not. Because helicity is not positive-semidefinite, it is not thought to play a role in the equipartition spectrum. Thus, $\{\sigma^i_{\boldsymbol{k}} : i \in I\} = \{1\}$, and the three-dimensional equipartition spectrum is

$$E(k) = \frac{2\pi k^2}{\alpha}. \tag{1.51}$$

Reproduction of the equipartition spectrum is an important test of any numerical code, which can be quite difficult. It is prudent to make use of conservative integrators (Bowman *et al.* [1997], Shadwick *et al.* [1999]) to avoid sec-

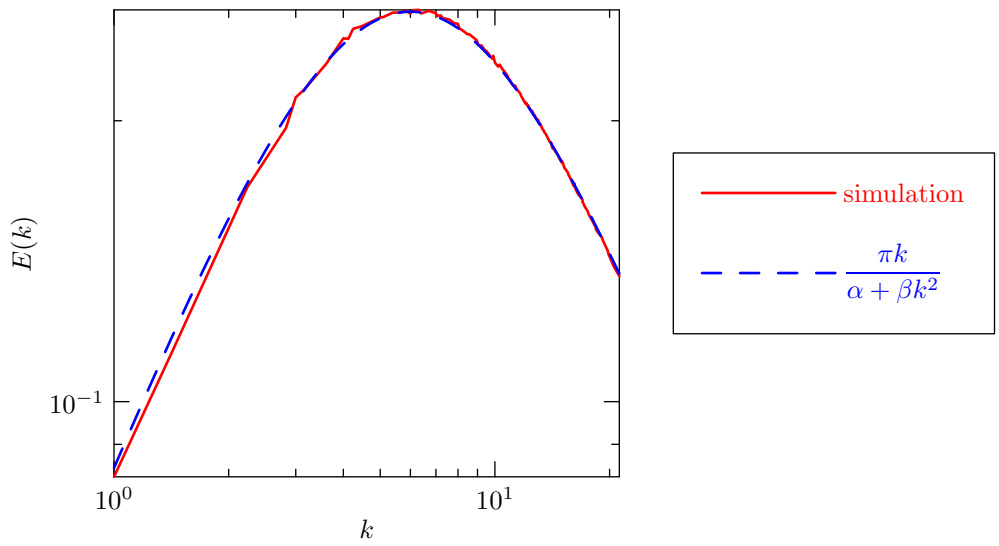

Figure 1.2: Equipartition spectrum for the two-dimensional Navier–Stokes equation, with equipartition spectrum parameters $\alpha = 15$, $\beta = 1$.

ular energy growth, as these errors can dominate the simulation quite quickly when $\nu = 0$.

Computational Domain in Fourier Space

Equation (1.39) determines the vorticity field for a two-dimensional flow using just one complex scalar quantity, $\omega(\boldsymbol{k}, t)$. For systems topologically equivalent to $S^1 \times S^1$ (S^1 is the unit circle) the wave vectors $\boldsymbol{k}$ will lie on a square lattice. For wave vectors significantly higher than $1/k_d$, ω is dominated by viscous damping and hence negligible. Thus, one can safely truncate the spectrum at high wavenumber. For numerical reasons, one typically truncates so as to leave a rectangular grid (see Chapter 3 for more details). From the Hermiticity condition in equation (1.34), one need only evaluate ω on half of this grid, as shown in Figure 1.3.

For inviscid runs, this argument is invalid since there is no reason that

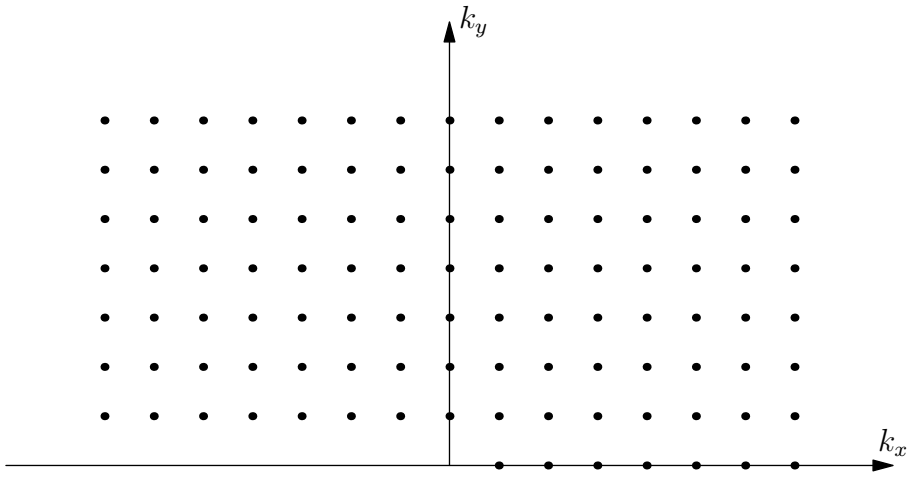

Figure 1.3: Retained wave vectors for the vorticity-based two-dimensional Navier–Stokes.

modes with high wavenumbers should be negligible. Indeed, the three-dimensional equipartition spectrum has average modal energies that are independent of wavenumber. To justify using a finite Fourier grid for inviscid simulations we provide the following theorem:

Theorem 1.1: *Consider a system consisting of the modes* $\{\omega_{\boldsymbol{k}}\}, \boldsymbol{k} \in D$ *for some set* D*, with*

$$\frac{\partial \boldsymbol{\omega_k}}{\partial t} = \sum_{\substack{\boldsymbol{p},\boldsymbol{q}\in D \\ \boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{0}}} \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \boldsymbol{\omega}_{\boldsymbol{p}}^* \boldsymbol{\omega}_{\boldsymbol{q}}^* \tag{1.52}$$

where

$$\epsilon_{\boldsymbol{kpq}} = -\epsilon_{\boldsymbol{qpk}}, \tag{1.53}$$

then the system conserves the energy

$$E = \frac{1}{2} \sum_{\boldsymbol{k}\in D} \frac{|\boldsymbol{\omega_k}|^2}{k^2}. \tag{1.54}$$

Proof. The rate of change of energy is given by

$$\begin{aligned}\frac{dE}{dt} &= \operatorname{Re}\sum_{\boldsymbol{k}\in D}\frac{\omega_{\boldsymbol{k}}^*}{k^2}\frac{\partial\omega_{\boldsymbol{k}}}{\partial t}\\ &= \operatorname{Re}\sum_{\boldsymbol{k}\in D}\frac{\omega_{\boldsymbol{k}}^*}{k^2}\sum_{\substack{\boldsymbol{p},\boldsymbol{q}\in D\\ \boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{0}}}\frac{\epsilon_{\boldsymbol{kpq}}}{q^2}\boldsymbol{\omega}_{\boldsymbol{p}}^*\boldsymbol{\omega}_{\boldsymbol{q}}^*\\ &= \operatorname{Re}\sum_{\substack{\boldsymbol{k},\boldsymbol{p},\boldsymbol{q}\in D\\ \boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{0}}}\frac{\epsilon_{\boldsymbol{kpq}}}{k^2q^2}\boldsymbol{\omega}_{\boldsymbol{k}}^*\boldsymbol{\omega}_{\boldsymbol{p}}^*\boldsymbol{\omega}_{\boldsymbol{q}}^*\\ &= 0\end{aligned} \tag{1.55}$$

since $\epsilon_{\boldsymbol{kpq}}$ is antisymmetric under exchange of $\boldsymbol{k}$ and $\boldsymbol{q}$. □

A similar calculation establishes enstrophy conservation.

Thus, the grid of wave vectors shown in Figure 1.3 will conserve the same invariants as if we had chosen our wave vectors $\boldsymbol{k}$ to come from all of $\mathbb{Z}^2$. Since the wave vectors on the truncated lattice are interacting, ergodicity still holds, and the truncated system will tend towards equipartition.

Computation of Discrete Spectra

Kolmogorov formulated his theory for the case where the physical domain is infinite, which implies that the Fourier basis is all of $\mathbb{R}^d$, where d is the dimension. The energy spectrum wavenumber k is then a continuous quantity, and the definition of $E(k)$ is straightforward. In the case where the physical domain is finite, we present two different methods for calculating the spectrum.

One choice is to use a *binned spectrum*. The spectrum is given by

$$E(k) = \sum_{\ell\in\left[k-\frac{1}{2},k+\frac{1}{2}\right)}\sum_{\boldsymbol{\ell}}\frac{1}{2}\frac{|\omega_{\boldsymbol{\ell}}|^2}{\ell^2}\Bigg/\sum_{\ell\in\left[k-\frac{1}{2},k+\frac{1}{2}\right)}\sum_{\boldsymbol{\ell}}1, \quad k=1,2,3,\ldots \tag{1.56}$$

That is, we take $E(k)$ to be the average energy for the modes with $\omega_{\boldsymbol{\ell}}$ for ℓ within $1/2$ of k, treating k as an integer. The distribution of spectral shells is shown in Figure 1.4. This produces a fairly smooth spectrum for averaging

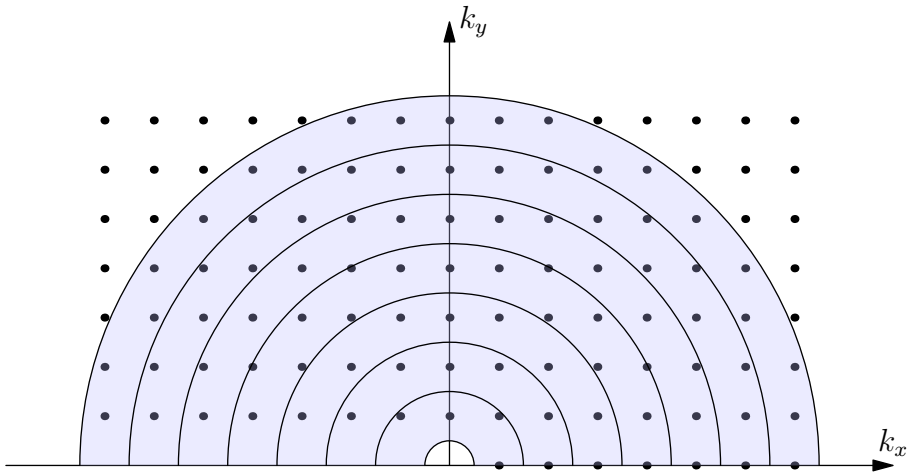

Figure 1.4: Diagram of distribution of binned energy shells.

over relatively small time windows because more modes are included per shell, particularly for large values of k. For a simulation consisting of m by $2m-1$ modes, there will be $\lceil\sqrt{2}m\rceil$ such shells. However, the modes with $|\boldsymbol{\ell}| > m$ are highly oscillatory, and are usually ignored in favour of concentrating on modes that lie in the semi-circle inscribed in the upper-half plane of the truncated grid, in which case there are m shells, as shown in Figure 1.4. Figure 1.5 gives an example of the raw spectrum.

Alternatively, one can determine $E(k)$ for only those values of k for which there exists at least one mode $\omega_{\boldsymbol{\ell}}$ with $|\boldsymbol{\ell}| = k$. We refer to this as the *raw spectrum*. The first such value is $k = 1$, which is attained when $\boldsymbol{\ell} = (\pm 1, 0)$ or $\boldsymbol{\ell} = (0, \pm 1)$. The next value is $k = \sqrt{2}$, and so on. The distribution of achieved radii is shown in Figure 1.6. The question arises as to how the number of achieved radii grows as m becomes large; if this is some sizable fraction of $2m^2$,

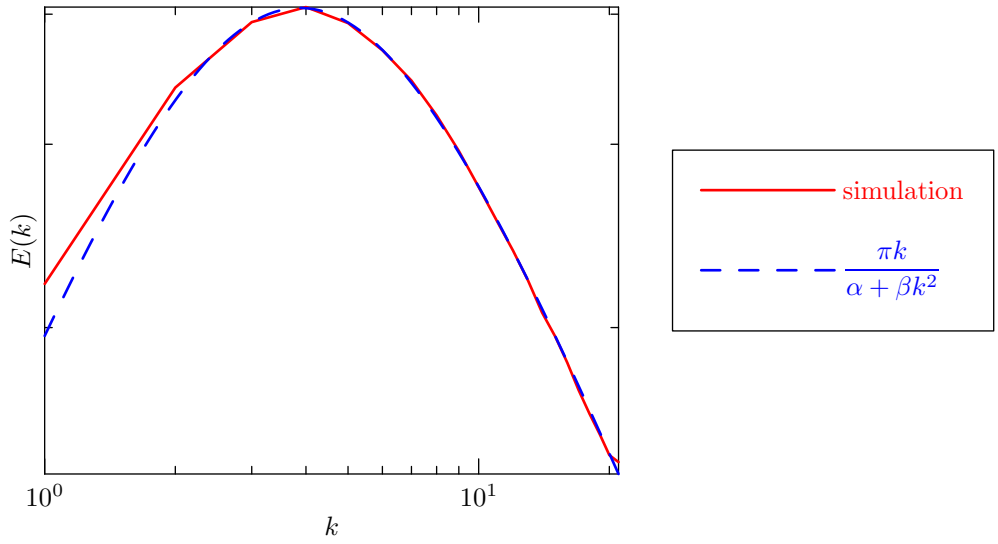

Figure 1.5: Binned energy spectrum for an inviscid, unforced run at late times, with equipartition spectrum parameters $\alpha = 15$, $\beta = 1$.

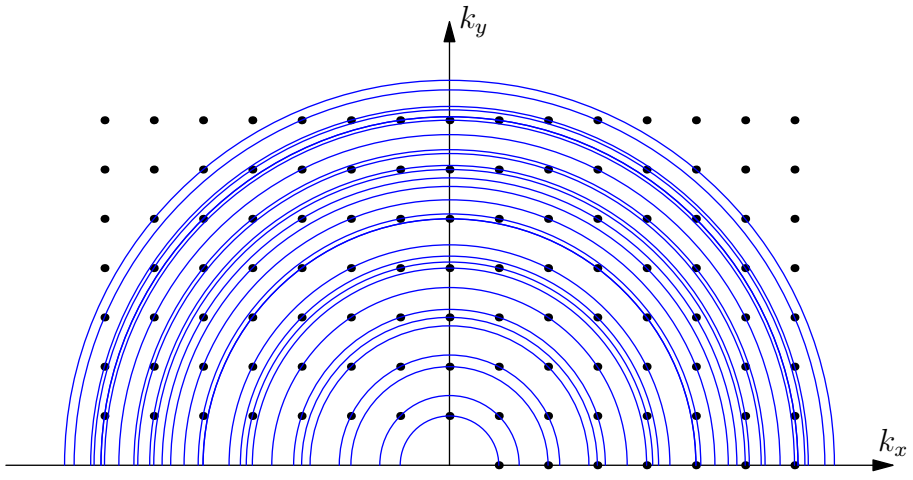

Figure 1.6: Diagram of distribution of radii for the raw spectrum.

then computing the spectrum in this fashion would be somewhat expensive. Certainly, there are at least m radii achieved (i.e. those along the axis), and no more than $m(m+1)/2$, the number of modes in an octant. Ramanujan established a more accurate estimate in the early 20^{th} century, which is detailed in [Hardy 1999]. The leading order estimate for the number of numbers less than x that are sums of squares is asymptotic to

$$\frac{Kx}{\sqrt{\log x}}, \tag{1.57}$$

with

$$K = \left[\frac{1}{2}\left(\prod_r \frac{1}{1-r^{-2}}\right)\right]^{\frac{1}{2}} \approx 0.764\ldots \tag{1.58}$$

where r runs through the primes that are equal to $-1 \bmod 4$. Thus, the number of radii achieved that are less than m is equal to the number of numbers less than m^2 that are sums of squares, i.e.

$$\frac{Km^2}{\sqrt{\log m^2}}, \tag{1.59}$$

as shown in Figure 1.7. The computational complexity for calculating averaged values of $E(k)$ for all of these radii, while higher than the binned case, is not leading order for pseudospectral simulations. The difference in speed between the two methods is less than is detectable on a computer. Figure 1.2 gives an example of the raw spectrum.

Numerical Solutions to the Navier–Stokes Equations

This monograph focuses on the *pseudospectral* method for solving the spectral Navier–Stokes equations numerically. Upon taking the Fourier transform of a given nonlinear evolution equation, the nonlinearity is transformed into a spectral convolution. While the linear terms may be efficiently computed in

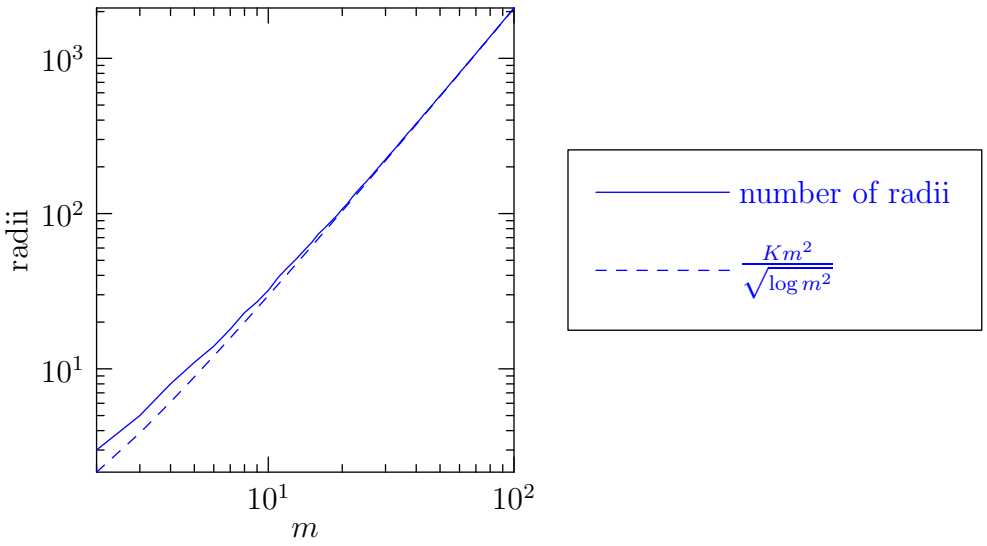

Figure 1.7: Number of radii achieved as a function of grid size m.

Fourier space, a direct calculation of the convolution takes $\mathcal{O}(n^2)$ operations (or more, for cubic and higher-order nonlinearities). Thus, it is preferred to calculate the nonlinear terms by returning from $\boldsymbol{k}$-space to $\boldsymbol{x}$-space via an inverse transformation where the nonlinear term may be calculated as a multiplication, taking $\mathcal{O}(n)$ operations. Since the *fast Fourier transform* (FFT) takes only $\mathcal{O}(n \log n)$ operations, this method is much faster than a purely spectral method. Moreover, the FFT has better numerical accuracy than a direct convolution. This method is discussed at great length in Chapter 3.

Using Kolmogorov's estimate for the dissipation scale as an estimate for the smallest scale required to capture the dissipation range, the highest wavenumber grows like

$$k_d \approx \frac{1}{\eta_d} \sim R^{3/4}, \tag{1.60}$$

where R is the Reynolds number. The total number of modes N grows like

$$N \sim R^{9/4}, \tag{1.61}$$

for three-dimensional turbulence. Physical systems of interest have Reynolds

numbers of the order 10^9, resulting in systems with more than 10^{20} active modes, far out-stripping the memory of any computer currently available. Even in two dimensions, where $\eta_d \sim R^{-1/2}$, the number of modes grows linearly with Reynolds number.

In order to bridge the gap between the simulations that one would like to perform and the simulations that one is able to perform, one turns to subgrid models, which require less computational power, and produce approximate solutions. These are discussed in Chapter 5.

1.C Shell Models of Turbulence

The incompressibility condition eliminates any possibility of the one-dimensional Navier–Stokes equation being interesting. And, while not as difficult to solve numerically as three-dimensional turbulence, two-dimensional turbulence still requires a great deal of computational power and, more importantly, patience.

Shell models of turbulence are ad-hoc models that mimic the behaviour and share characteristics with the Navier–Stokes equation, while being simpler both analytically and computationally. We present three models, namely the DN model (subsection 1.C.1), the GOY model (subsection 1.C.2), and the Sabra model (subsection 1.C.3).

Shell models are purely spectral models with a single complex quantity u_n representing a typical modal amplitude for all modes $\boldsymbol{u}_{\boldsymbol{k}}$ with $|\boldsymbol{k}| \in [k_n, k_{n+1})$, with $k_n = k_0\lambda^n$, where k_0 is the amplitude of the zeroth mode, and λ is the geometric shell spacing factor. These shells are shown in Figure 1.8. The modes u_n interact via a quadratic nonlinearity (as in the spectral Navier–

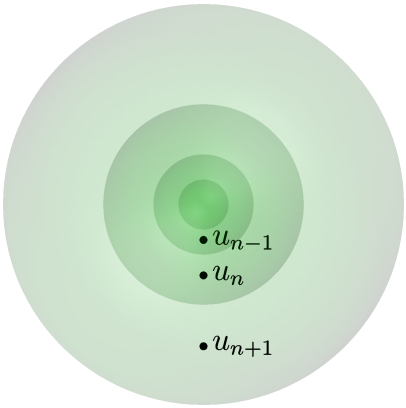

Figure 1.8: Distribution of modes for shell models of turbulence.

Stokes equations), which has the general form

$$k_n \sum_{\ell,m} A_{\ell,m} u_\ell^* u_m^*. \tag{1.62}$$

For the models under consideration herein, the sum is restricted to modes that neighbour (or are next to neighbours) of mode n. The shells are damped by a viscous term $-\nu k_n^2 u_n$, and forced with force F_n. The general evolution equation for shell models is

$$\frac{\partial u_n}{\partial t} + \nu k_n^2 u_n = k_n \sum_{\ell,m} A_{\ell,m} u_\ell^* u_m^* + F_n. \tag{1.63}$$

The energy of such models is defined as

$$E = \frac{1}{2} \sum_n |u_n|^2 , \tag{1.64}$$

and the energy spectrum is

$$E(k_n) = \frac{1}{2} \frac{|u_n|^2}{k_{n+1} - k_n}. \tag{1.65}$$

Higher-order moments of shell models of turbulence are also available, with the p^{th} moment defined as

$$S_p = \langle |u_n|^p \rangle , \tag{1.66}$$

where $\langle \dots \rangle$ indicates averaging in time. The moments scale like $S_p(k) \sim k^{\zeta_p}$, where ζ_p is known as the structure function exponent.

1.C.1 The DN Model

If one restricts the nonlinear term in equation (1.63) to nearest-neighbour interactions, energy is conserved if and only if the nonlinear term has form

$$ik_n\big(a_n u_{n-1}^2 - \lambda a_{n+1} u_n u_{n+1} + b_n u_{n-1} u_n - \lambda b_{n+1} u_{n+1}^2\big)^*, \tag{1.67}$$

where a_n and b_n are the nonlinear interaction coefficients, which are typically constant with respect to n. In this case, the evolution equation is

$$\left(\frac{\partial}{\partial t} + \nu k_n^2\right) u_n = ik_n\big[a\big(u_{n-1}^2 - \lambda u_n u_{n+1}\big) + b\big(u_{n-1}u_n - \lambda u_{n+1}^2\big)\big]^* + F_n, \tag{1.68}$$

which is known as the *DN model*, developed by Desnyansky & Novikov [1974]. Unlike the Navier–Stokes equations, the DN model has only one conserved quadratic invariant, namely energy. The equipartition is of the energy of shells, resulting in an equipartition spectrum proportional to $1/k$, as shown in Figure 1.9. The nonlinearity of equation (1.68) has a fixed point when

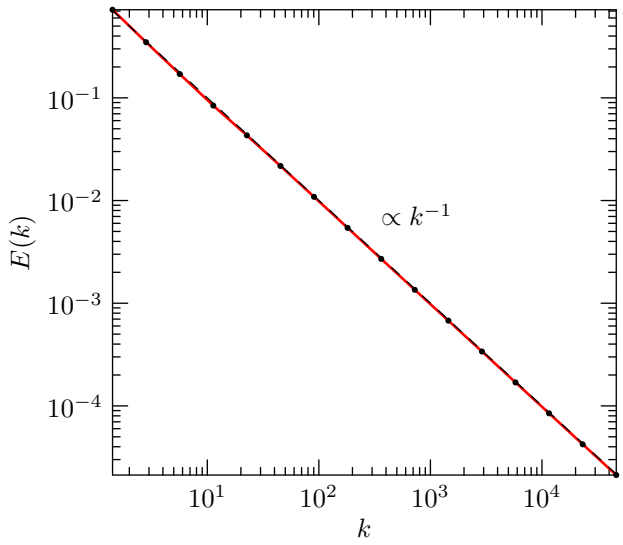

Figure 1.9: Equipartition spectrum on the DN model.

$u_n \propto k_n^{-1/3}$, which corresponds to $E(k) \propto k_n^{-5/3}$, i.e. a Kolmogorov spectrum.

Bell & Nelkin [1977] concluded that this fixed point is linearly stable if a and b have opposite sign. The DN model reproduces the Kolmogorov spectrum even with white-noise forcing and viscous damping, as shown in Figure 1.10. The higher-order moments S_p for the DN model are anomalous, but do not

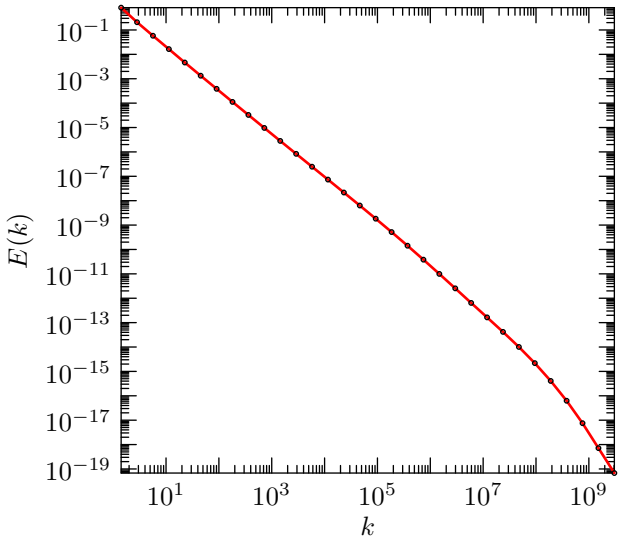

Figure 1.10: The energy spectrum of the DN model with white-noise forcing at low wavenumbers and molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $a = -1/4$ $b = -1$, and 32 shells.

display the same scaling with p as does the Navier–Stokes equation. The structure function exponents for one realization of the DN model are shown in Figure 1.11.

1.C.2 The GOY Model

The GOY model, first proposed as a model with u_n real-valued by Gledzer [1973] and generalized to complex u_n by Yamada & Ohkitani [1987], features a next-nearest neighbour quadratic nonlinearity. The evolution equation for the GOY

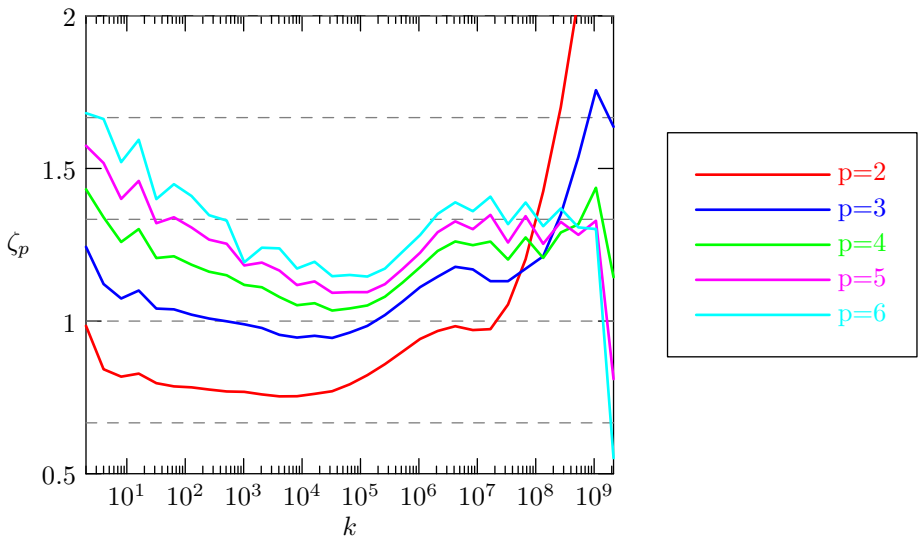

Figure 1.11: Structure function exponents ζ_p of the DN model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $a = -1/4$ $b = -1$, and 32 shells. Dashed lines show the predictions of Kolmogorov theory.

model is

$$\frac{\partial u_n}{\partial t} = ik_n\left(\alpha u_{n+1}u_{n+2} + \frac{\beta}{\lambda}u_{n-1}u_{n+1} + \frac{\gamma}{\lambda^2}u_{n-1}u_{n-2}\right)^* - \nu k_n^2 u_n + F_n. \quad (1.69)$$

The GOY model has a fixed point when $F = \nu = 0$ with $u_n \propto k_n^{-1/3}$, but, unlike the DN model, this fixed point is unstable [Biferale *et al.* 1995]. The GOY model conserves the energy $E = \frac{1}{2}\sum_n |u_n|^2$ if the nonlinear coupling coefficients α, β, and γ are such that

$$\alpha + \beta + \gamma = 0. \quad (1.70)$$

One typically rescales time so that $\alpha = 1$, leaving two free parameters.

A second quantity,

$$\frac{1}{2}\sum_n k_n^p |u_n|^2 \quad (1.71)$$

is conserved, where $p = -\log_\lambda(-\beta - 1)$. With $\beta = -1/2$, and $\lambda = 2$ this quantity is equal to

$$H = \frac{1}{2}\sum_n (-1)^n k_n |u_n|^2, \quad (1.72)$$

which we call the *shell-model helicity*. Like helicity in 3D turbulence, this quantity is not positive definite, and does not play an important rôle in determining the equipartition spectrum. The choice $\beta = -5/4$ and $\lambda = 2$ implies that the vorticity

$$Z = \frac{1}{2} \sum_n k_n^2 \left| u_n \right|^2 \tag{1.73}$$

is conserved.

The GOY model exhibits a period-three oscillation [Biferale *et al.* 1995], as is evident from the fact that

$$u_n = k_n^{-\frac{1}{3}} \begin{cases} A_0 \text{ for } n = 0 \pmod 3, \\ A_1 \text{ for } n = 1 \pmod 3, \\ A_2 \text{ for } n = 2 \pmod 3 \end{cases} \tag{1.74}$$

is a fixed point of the nonlinearity in the GOY model for arbitrary A_0, A_1, and A_2. This introduces noise into the energy spectrum. Kadanoff *et al.* [1995] suggested that one should instead compute the spectrum based on the flux

$$\Sigma_n \doteq \left\langle \left| \operatorname{Im} \left(u_n u_{n+1} u_{n+2} + \frac{1+\beta}{\lambda} u_{n-1} u_n u_{n+1} \right) \right|^{2/3} \right\rangle, \tag{1.75}$$

where $\langle \dots \rangle$ is a windowed average in time, to avoid this problem. A comparison between the energy spectrum and the flux spectrum is given in Figure 1.12. The GOY model has a k^{-1} equipartition spectrum, as shown in Figure 1.13, and shows Kolmogorov-style scaling in forced-dissipative simulations, as shown in Figure 1.14. The higher-order moments of the GOY model scale in very much the same fashion as Navier–Stokes turbulence. Structure function exponents for the GOY model are shown in Figure 1.15, which are compared with experimental values from Jensen *et al.* [1991].

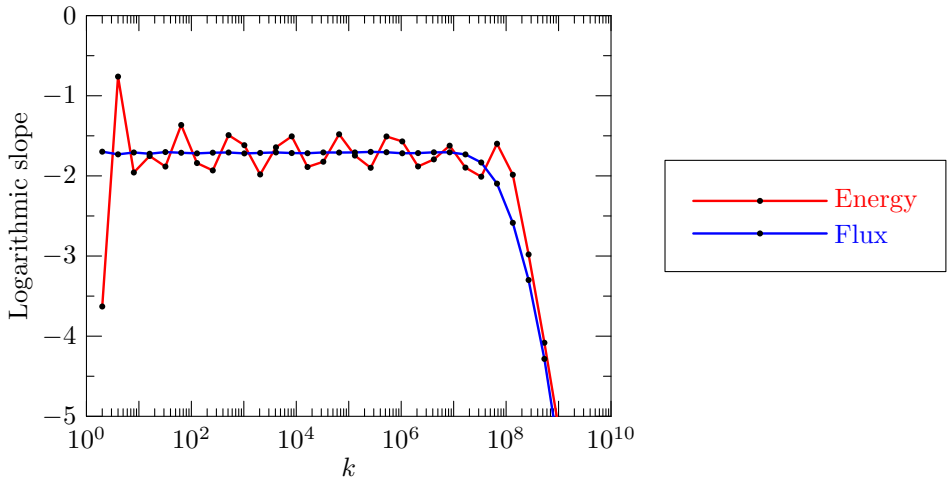

Figure 1.12: A comparison of the energy spectrum and flux spectrum for the forced-dissipative GOY model.

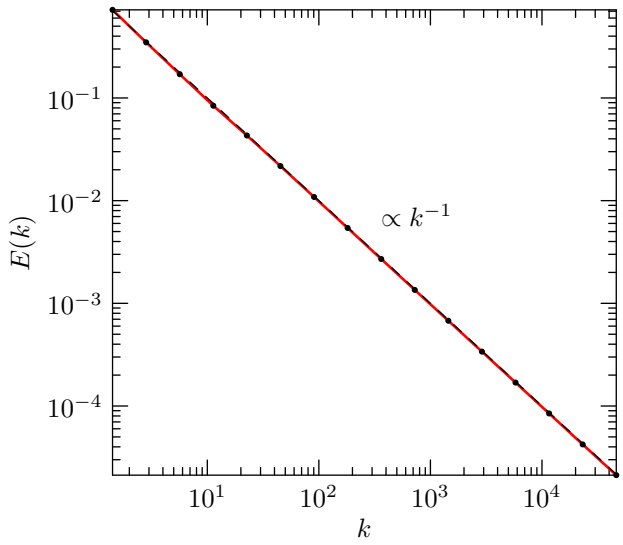

Figure 1.13: The equipartition spectrum of the inviscid, unforced GOY model.

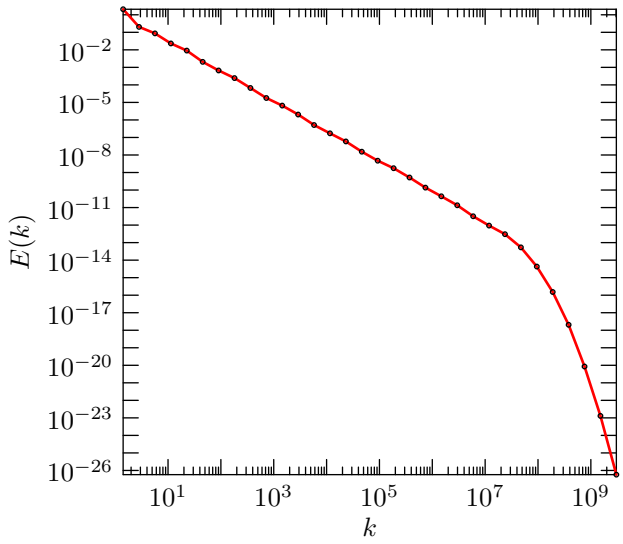

Figure 1.14: The energy spectrum of the GOY model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $\alpha = 1$ $b = -1/2$, and 32 shells.

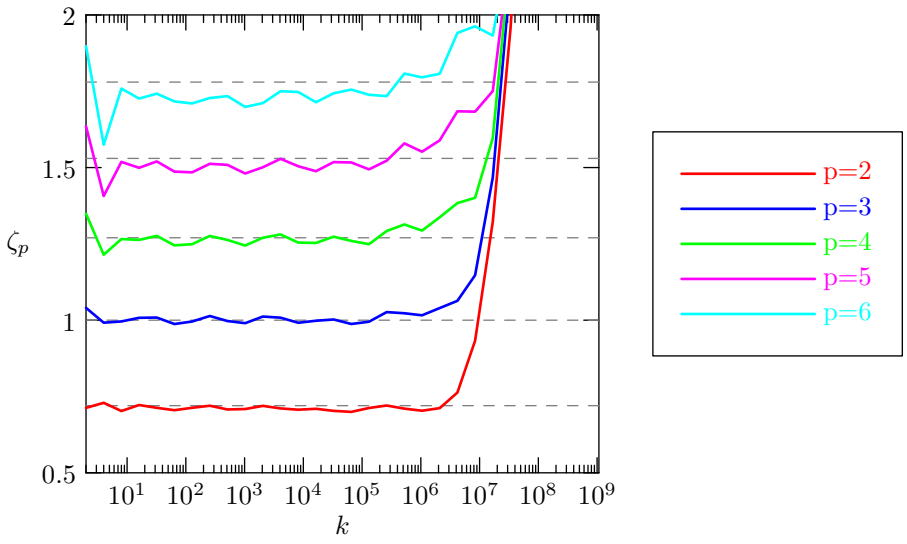

Figure 1.15: Structure function exponents ζ_p of the GOY model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $\alpha = 1$ $b = -1/2$, and 32 shells. Dashed lines show experimental values.

1.C.3 The Sabra Model

The Sabra model, developed by L'vov *et al.* [1998], is a next-nearest neighbour model that lacks the period-three oscillation found in the GOY model. This is accomplished by modifying the nonlinear term to take the complex conjugate of some terms but not others. The evolution equation of the Sabra model is

$$\frac{\partial u_n}{\partial t} = ik_n\left(\alpha u^*_{n+1}u_{n+2} + \frac{\beta}{\lambda}u^*_{n-1}u_{n+1} - \frac{\gamma}{\lambda^2}u_{n-1}u_{n-2}\right) - \nu k_n^2 u_n + F_n \quad (1.76)$$

Like the GOY model, the Sabra model has a k^{-1} equipartition spectrum (see Figure 1.16) and exhibits a Kolmogorov spectrum (see Figure 1.17) in forced-dissipative simulations. The higher-order moments of the Sabra model are

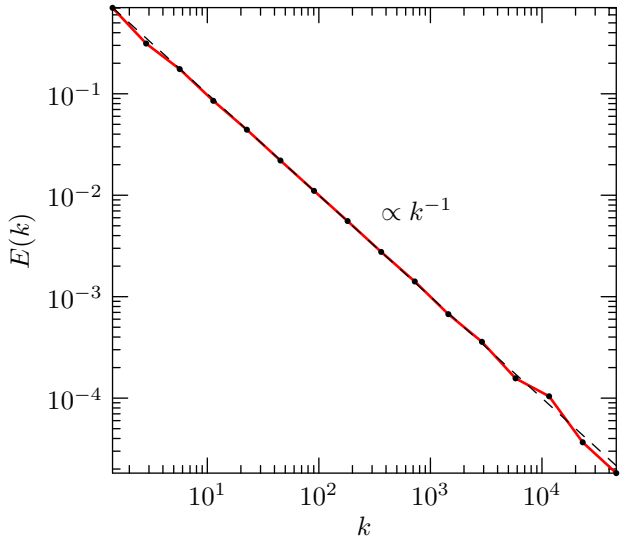

Figure 1.16: The equipartition spectrum of the inviscid, unforced Sabra model.

very similar to those of the GOY model, however, the advantage of having a smoother spectrum seems to have been lost with higher moments, which are much less smooth than those of the GOY model. Structure function exponents for the Sabra model are shown in Figure 1.18.

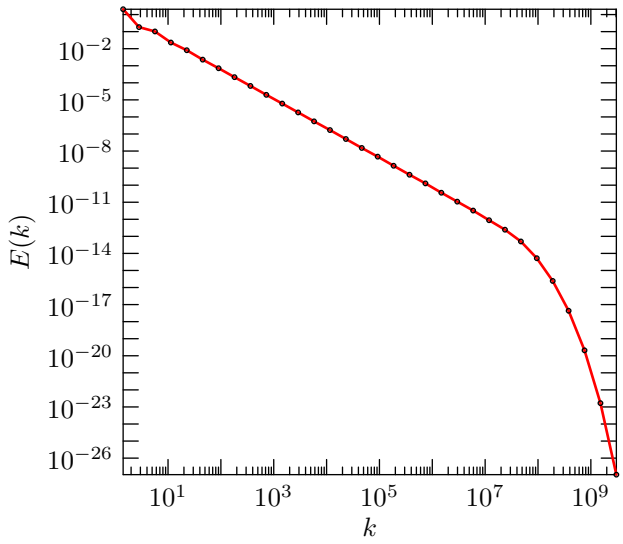

Figure 1.17: The energy spectrum of the Sabra model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $\alpha = 1$ $b = -1/2$, and 32 shells.

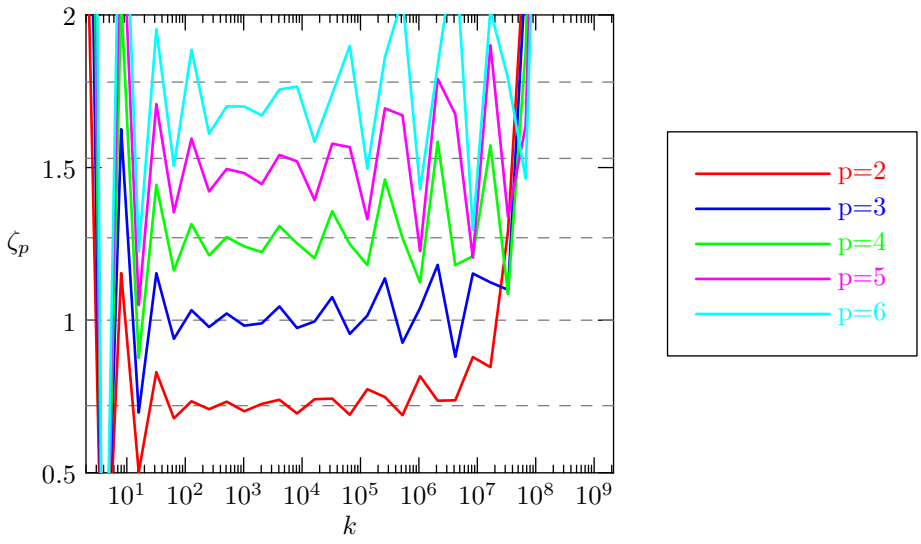

Figure 1.18: Structure function exponents ζ_p of the Sabra model with the first shell forced with a white-noise forcing with $\epsilon = 1$, molecular viscosity with $\nu = 10^{-10}$, nonlinear interaction coefficients $\alpha = 1$ $b = -1/2$, and 32 shells. Dashed lines show the predictions of Kolmogorov theory.

Chapter 2

Analytic Results from Shell Models of Turbulence

Wherein much insight is gained by considering shell models of turbulence with unusual resolutions, characteristics of classical Kolmogorov theory are reproduced, and the relationship between shell-model parameters and anomalous scaling are explored.

Shell models are toy models for turbulent systems; they mimic the behaviour of solutions to the Navier–Stokes equations but lack the computational and analytic complexity. They are simpler but non-trivial; they, like the Navier–Stokes equations, involve a quadratic nonlinear term, which makes direct analysis difficult.

The shell models considered consist of geometrically spaced shells in Fourier space, with shell n having characteristic wavenumber $k_n = k_0\lambda^n$, where k_0 is the wavenumber of mode 0, and λ is the dimensionless geometric spacing factor. For typical simulations $\lambda = 2$ and $k_0 = 1$, allowing simulations with

$Re \approx 10^{11}$ using only 32 shells. The shells interact according to an evolution equation of the form

$$\frac{\partial u_n}{\partial t} + \nu k_n^2 u_n = k_n \sum_{\ell,m} A_{\ell,m} u_\ell^* u_m^* + F_n, \tag{2.1}$$

where ν is the viscosity, $A_{\ell,m}$ is the nonlinear interaction coefficient, and F_n is an external force.

2.A Low-resolution shell models

Shell models with $\lambda > 2$ require fewer shells to model the same range of scales as shell models with $\lambda = 2$, and are considered *low-resolution shell models*. In the limit of λ being very large, the range of active scales is eventually modelled by one single shell.

2.A.1 DN model

The nonlinear source term of the DN model,

$$\frac{du_n}{dt} = ik_n \big[a\big(u_{n-1}^2 - \lambda u_n u_{n+1}\big) + b\big(u_{n-1}u_n - \lambda u_{n+1}^2\big)\big]^* - \nu k_n^2 u_n, \tag{2.2}$$

is dominated by high-wavenumber shells when $\lambda \gg 1$. Moreover, the nonlinear coupling term increases as λ increases. To maintain the magnitude of the nonlinear coupling as λ becomes very large, we modify the nonlinear coupling coefficients as

$$(a, b) \to \left(\frac{a}{\lambda}, \frac{b}{\lambda}\right). \tag{2.3}$$

The DN model is then mapped to

$$\frac{du_n}{dt} = ik_n \left[a\left(\frac{u_{n-1}^2}{\lambda} - u_n u_{n+1}\right) + b\Big(\frac{u_{n-1}u_n}{\lambda} - u_{n+1}^2\Big)\right]^* - \nu k_n^2 u_n. \tag{2.4}$$

In the limit of large λ, the first and third terms in the nonlinearity are insignificant with respect to the other terms, leaving

$$\frac{du_n}{dt} = -ik_n\left(au_nu_{n+1} + bu_{n+1}^2\right)^* - \nu k_n^2 u_n. \tag{2.5}$$

The wavenumber for shell n increases as λ increases, so, if ν is kept constant, this system is ultimately dominated by viscosity, as the term $\nu k_n^2 u_n$ will increase faster than the nonlinear term. This is expected; increasing λ is equivalent to approximating more and more scales by a single thick shell, so as one includes higher and higher wavenumbers into a single shell, one would expect that viscous dissipation from smalls scales would dominate. The main advantage of considering this limit is that equation (2.5) shows the rôle of the nonlinear coupling coefficients a and b: a is associated with the term u_nu_{n+1} and thus provides self-coupling i.e. the shell self-interacts, whereas b, associated with u_{n+1}^2, controls back-scatter, i.e. an inverse energy cascade from high wavenumbers to low wavenumbers.

2.A.2 GOY model

We perform a similar analysis with the GOY model,

$$\frac{du_n}{dt} = ik_n\left(\alpha u_{n+1}u_{n+2} + \frac{\beta}{\lambda}u_{n-1}u_{n+1} + \frac{\gamma}{\lambda^2}u_{n-1}u_{n-2}\right)^* - \nu k_n^2 u_n. \tag{2.6}$$

The GOY model conserves energy when $\alpha + \beta + \gamma = 0$. Using $\beta = -\alpha - \gamma$, equation (2.6) becomes

$$\frac{\partial u_n}{\partial t} = ik_n\left[\frac{\alpha}{\lambda}(\lambda u_{n+1}u_{n+2} - u_{n+1}u_{n-1}) + \frac{\gamma}{\lambda^2}(u_{n-1}u_{n-2} - \lambda u_{n+1}u_{n-1})\right]^* - \nu k_n^2 u_n. \tag{2.7}$$

Letting $a = \gamma/\lambda$ and $b = \alpha$, this becomes

$$\frac{\partial u_n}{\partial t} = ik_n\left[b\left(u_{n+1}u_{n+2} - \frac{1}{\lambda}u_{n+1}u_{n-1}\right) + a\left(\frac{u_{n-1}u_{n-2}}{\lambda} - u_{n+1}u_{n-1}\right)\right]^* - \nu k_n^2 u_n. \tag{2.8}$$

For large λ, one arrives at

$$\frac{\partial u_n}{\partial t} = ik_n(-au_{n+1}u_{n-1} + bu_{n+1}u_{n+2})^* - \nu k_n^2 u_n. \tag{2.9}$$

As with the large-λ DN model, the large-λ GOY model given in equation (2.9) is dominated by the dissipative term. The low-resolution limit of the GOY model has the nonlinear coupling coefficient a governing self-coupling and b governing back-scatter, as in the low-resolution limit of the DN model.

2.B High-resolution limits

Shell models with $\lambda < 2$ require more shells to capture the same range of scales as shell models with $\lambda = 2$, and are thus considered *high-resolution shell models*. High-resolution limits of shell models of turbulence provide a different perspective; if the resolution is very high, one can successfully apply a continuum approximation, making the problem amenable to analytic results [Benzi *et al.* 1996], [Jensen & Olesen 1998].

The continuum limit is achieved in the case $\lambda \to 1$. That is, the geometric spacing λ of the characteristic shell wavenumber is unity, and the shell spacing goes to zero. We introduce a continuous shell index η, given by

$$\eta = \log \lambda^n = n \log \lambda,$$

which maps the geometrically-spaced shells back to a linear spacing. Taking $k_0 = 1$ for simplicity, the wavenumber k_n for mode n is given by

$$k_n = \lambda^n = e^{\eta},$$

with $\eta \in (0, \infty)$. Let $\Delta = \log \lambda$ be the shell-spacing. Since shell models exhibit a smooth spectrum, we treat neighbouring modes via Taylor approximations:

$$u_{n\pm1} = u(\eta) \pm \Delta \frac{\partial u}{\partial \eta} + O(\Delta^2). \tag{2.10}$$

2.B.1 DN model

The DN model has nonlinearity

$$ik_n \big[a\big(u_{n-1}^2 - \lambda u_n u_{n+1}\big) + b\big(u_{n-1}u_n - \lambda u_{n+1}^2\big)\big]^*, \tag{2.11}$$

which, under the infinite-resolution limit and continuum approximation, becomes

$$\begin{aligned} ie^{\eta} \Bigg[a\Bigg(\left(u - \Delta \frac{\partial u}{\partial \eta}\right)^2 - \lambda u \left(u + \Delta \frac{\partial u}{\partial \eta}\right) \Bigg) \\ + b\Bigg(\left(u - \Delta \frac{\partial u}{\partial \eta}\right) u - \lambda \left(u + \Delta \frac{\partial u}{\partial \eta}\right)^2 \Bigg) \Bigg]^* . \end{aligned} \tag{2.12}$$

Keeping terms to first order in Δ, one arrives at

$$\begin{aligned} & ie^{\eta} \left[a\left(u^2 - 2\Delta \frac{\partial u}{\partial \eta} - \lambda u^2 + \lambda \Delta u \frac{\partial u}{\partial \eta} \right) + b\left(u^2 - \Delta \frac{\partial u}{\partial \eta} - \lambda u^2 - 2\Delta\lambda \frac{\partial u}{\partial \eta} \right) \right]^* \\ & = ie^{\eta} \left[u^2(a+b)(1-\lambda) + \Delta u \frac{\partial u}{\partial \eta}(-2a + \lambda a - b - 2b\lambda) \right]^* \\ & = ie^{\eta} \left[Au^2 + Bu \frac{\partial u}{\partial \eta} \right]^*, \end{aligned} \tag{2.13}$$

where $A = (a+b)(1-\lambda)$ and $B = \Delta(-2a + \lambda a - b - 2b\lambda)$. In order to have a non-zero nonlinear term as $\lambda \to 1$, we must choose the nonlinear coupling coefficients a and b via

$$a \to \frac{a}{\lambda - 1}, \qquad b \to \frac{b}{\lambda - 1},$$

which is equivalent to the scaling from section 2.A when $\lambda \gg 1$. This gives

$$A = -(a+b), \qquad B = \frac{\Delta}{\lambda - 1}[a(\lambda - 2) + b(-1 - 2\lambda)].$$

Taking $\lambda \to 1$ the continuum DN equation is

$$\frac{\partial u}{\partial t} = ie^{\eta}\left[(-a-b)u^2 + (-a-3b)u\frac{\partial u}{\partial \eta}\right]^* - \nu e^{2\eta}u. \tag{2.14}$$

2.B.2 GOY model

The GOY model has nonlinearity

$$ik_n\left(\alpha u_{n+1}u_{n+2} + \frac{\beta}{\lambda}u_{n-1}u_{n+1} + \frac{\gamma}{\lambda^2}u_{n-1}u_{n-2}\right)^*. \tag{2.15}$$

Applying the continuum approximation $u_{n\pm1} = u(\eta) \pm \Delta\frac{\partial u}{\partial \eta}$ with $k_n = e^{\eta}$ and $\Delta = \log\lambda$ this becomes

$$\begin{aligned} ie^{\eta}\Bigg[&\alpha\left(u + \Delta\frac{\partial u}{\partial \eta}\right)\left(u + 2\Delta\frac{\partial u}{\partial \eta}\right) \\ &+ \frac{\beta}{\lambda}\left(u - \Delta\frac{\partial u}{\partial \eta}\right)\left(u + \Delta\frac{\partial u}{\partial \eta}\right) + \frac{\gamma}{\lambda^2}\left(u - \Delta\frac{\partial u}{\partial \eta}\right)\left(u - 2\Delta\frac{\partial u}{\partial \eta}\right)\Bigg]^*, \end{aligned} \tag{2.16}$$

and keeping terms that are first order in Δ yields

$$ie^{\eta}\left[\left(\alpha + \frac{\beta}{\lambda} + \frac{\gamma}{\lambda^2}\right)u^2 + 3\left(\alpha - \frac{\gamma}{\lambda^2}\right)\Delta u\frac{\partial u}{\partial \eta}\right]^* = ie^{\eta}\left(Au^2 + Bu\frac{\partial u}{\partial \eta}\right)^*, \tag{2.17}$$

where $A = \alpha + \frac{\beta}{\lambda} + \frac{\gamma}{\lambda^2}$ and $B = \Delta 3\left(\alpha - \frac{\gamma}{\lambda^2}\right)$. Since A and B go to zero as $\lambda \to 1$, we rescale the nonlinear coupling coefficient via the map

$$(\alpha, \beta, \gamma) \to \frac{(\alpha, \beta, \gamma)}{\lambda - 1}, \tag{2.18}$$

in accordance with the rescaling performed for the DN model.

Including the energy-conservation requirement $\alpha + \beta + \gamma = 0$ and taking the limit as $\lambda \to 1$ this continuum GOY model reduces to

$$\frac{\partial u}{\partial t} = ie^{\eta}K\left(u^2 + 3u\frac{\partial u}{\partial \eta}\right)^* - \nu e^{2\eta}u, \tag{2.19}$$

where $K = \alpha$ is the nonlinear scaling factor for the continuum GOY model.

2.B.3 Flux Formulation

The nonlinear term in the continuum limit shell model is energy-conserving if and only if

$$0 = \int u e^{\eta} \left(A u^2 + B u \frac{\partial u}{\partial \eta} \right) d\eta. \tag{2.20}$$

This may be expressed as a flux, if $B = 3A$, in which case

$$\begin{aligned} \int u e^{\eta} K \left(u^2 + 3u \frac{\partial u}{\partial \eta} \right) d\eta &= \int K \partial_{\eta} \left(e^{\eta} u^3 \right) d\eta \\ &= K e^{\eta} u^3 \Big|_{\eta_{min}}^{\eta_{max}}, \end{aligned} \tag{2.21}$$

that is, the nonlinear term only changes the energy due to boundary conditions. If $B \neq 3A$, then the nonlinear part of the continuum shell model does not conserve energy, and therefore cannot be written as a flux. For the DN model, this implies that $a = 0$. The continuum GOY model is expressible in flux form automatically if $\alpha + \beta + \gamma = 0$, i.e. the continuum GOY model conserves energy whenever the discrete GOY model conserves energy.

2.B.4 Higher-Order Approximations

Taking second-order continuum approximations of the DN model gives a nonlinearity of the form

$$\begin{aligned} & u^2(a - a\lambda + b - b\lambda) + \Delta u \frac{\partial u}{\partial \eta}(2a - a\lambda - b - 2b\lambda) \\ & + \Delta^2 \left[\left(\frac{\partial u}{\partial \eta} \right)^2 (a + b\lambda^2) + u \frac{\partial^2 u}{\partial \eta^2} \left(a - \frac{a\lambda}{2} + \frac{b}{2} - b\lambda \right) \right]. \end{aligned} \tag{2.22}$$

The term multiplied by Δ^2 will change the energy of the system since Δ^2 is a sign-definite operator. This secular energy change will not be generally corrected by other terms if one were to expand to yet higher powers in Δ. We therefore only consider the continuum equation to first order in Δ.

2.B.5 Steady-State Solutions

Both the DN and GOY models reduce to the same energy-conserving continuous PDE,

$$\frac{\partial u}{\partial t} = ie^{\eta} K \left(u^2 + 3u \frac{\partial u}{\partial \eta} \right)^* - \nu e^{2\eta} u \tag{2.23}$$

for appropriate nonlinear coefficients choices. In the steady state one has $\frac{\partial u}{\partial t} = 0$. Then,

$$ie^{\eta} K \left(u^2 + 3u \frac{\partial u}{\partial \eta} \right)^* - \nu e^{2\eta} u = 0, \tag{2.24}$$

which implies that

$$\frac{\partial u}{\partial \eta} = \frac{-i\nu}{3K} e^{\eta} \frac{u^*}{u} - \frac{u}{3}. \tag{2.25}$$

Unfortunately, this isn't easily solved for $u(\eta)$.

In the case that $u \in \mathbb{R}$, and K is purely imaginary, we arrive at the continuous PDE,

$$\frac{\partial u}{\partial t} = ie^{\eta} K \left(u^2 + 3u \frac{\partial u}{\partial \eta} \right) - \nu e^{2\eta} u, \tag{2.26}$$

which has steady-state solution

$$u = \left[\frac{\nu}{4Ki} \left(e^{\frac{4}{3}\eta} - 1 \right) + u_0 \right] e^{-\eta/3}, \tag{2.27}$$

where $u_0 \doteq u(0)$. If u_0 is real-valued and K purely imaginary, then

$$\frac{i\nu}{4K} \left(e^{\frac{4}{3}\eta} - 1 \right) + u_0 \tag{2.28}$$

equals zero for some value of η, and the solution to equation (2.26) reaches zero for some η, which we denote η_d. In particular, if $\eta_d \gg 1$, then one expects a dissipation wavenumber

$$e^{\eta_d} \sim \left(\frac{4Kiu_0}{\nu} \right)^{\frac{3}{4}}, \tag{2.29}$$

analytically reproducing Kolmogorov's prediction for Navier–Stokes turbulence [Kolmogorov 1941].

On taking the continuum shell model with $u \in \mathbb{R}$, and let $\hat{K} = -iK$ be real-valued and positive, so that the continuum shell model is

$$\frac{\partial u}{\partial t} = -\hat{K}\left(u^2 + 3\frac{\partial u}{\partial \eta}\right) - \nu e^{2\eta} u. \tag{2.30}$$

If we take equation (2.30) and multiply by u^{p-2} and apply the time-average operator $\langle \dots \rangle \doteq \int_0^T \dots dt/T$:

$$\begin{aligned}\left\langle u^{p-2}\frac{\partial u}{\partial t}\right\rangle &= \left\langle -\hat{K}e^{\eta}\left(u^2 + 3u\frac{\partial u}{\partial \eta}\right)u^{p-2} - \nu e^{2\eta}u^{p-1}\right\rangle \\ &= -\hat{K}e^{\eta}\left(\langle u^p\rangle + \frac{3}{p}\frac{\partial \langle u^p\rangle}{\partial \eta}\right) - \nu e^{2\eta}\left\langle u^{p-1}\right\rangle.\end{aligned} \tag{2.31}$$

In the statistically stationary state, the left-hand side of equation (2.31) is zero, and we can solve for moments iteratively via the equation

$$\frac{\partial \langle u^p\rangle}{\partial \eta} + \frac{p}{3}\langle u^p\rangle = -\frac{\nu p}{3\hat{K}}e^{\eta}\left\langle u^{p-1}\right\rangle, \tag{2.32}$$

which is a linear first-order differential equation in $\langle u^p\rangle$ having solution

$$\langle u^p\rangle = e^{-\frac{p}{3}\eta}\left(C - \frac{\nu}{\hat{K}}\frac{p}{3}\int e^{\frac{p+3}{3}\eta}\left\langle u^{p-1}\right\rangle d\eta\right). \tag{2.33}$$

Noting that $\langle u^0\rangle = 1$ and using the notation $c_p \doteq \langle u^p\rangle|_{\eta=0}$

$$\left\langle u^1\right\rangle = e^{-\eta/3}\left[u_0 - \frac{\nu}{3\hat{K}}\left(e^{\frac{4}{3}\eta} - 1\right)\right] \tag{2.34}$$

The next few moments are:

$$\left\langle u^2\right\rangle = e^{-\frac{2}{3}\eta}\left[c_2 + \frac{c_1}{2}\frac{\nu}{\hat{K}}\left(e^{\frac{4}{3}\eta} - 1\right) + \frac{1}{6}\frac{\nu^2}{\hat{K}^2}\left(e^{\frac{4}{3}\eta} - \frac{1}{2}e^{\frac{8}{3}\eta} - \frac{1}{2}\right)\right] \tag{2.35}$$

$$\begin{aligned}\left\langle u^3\right\rangle = e^{\eta}\Bigg[&c_3 - c_2\frac{3}{4}\frac{\nu}{\hat{K}}\left(e^{\frac{4}{3}\eta} - 1\right) + c_1\frac{3}{8}\frac{\nu^2}{\hat{K}^2}\left(e^{\frac{4}{3}\eta} - \frac{e^{\frac{8}{3}\eta}}{2} - \frac{1}{2}\right) \\ &+\frac{1}{16}\frac{\nu^3}{\hat{K}^3}\left(e^{\frac{4}{3}\eta} + e^{\frac{8}{3}\eta} - \frac{e^{4\eta}}{3} - \frac{5}{3}\right)\Bigg]\end{aligned} \tag{2.36}$$

$$\begin{aligned}\left\langle u^4\right\rangle = e^{\frac{4}{3}\eta}\Bigg[c_4 + \frac{\nu}{\hat{K}}\frac{2}{5}c_3\Big(1-e^{\frac{10}{3}\eta}\Big) \\ +c_2\frac{\nu^2}{\hat{K}^2}\left(\frac{3}{10}e^{2\eta}-\frac{3}{14}e^{\frac{10}{3}\eta}-\frac{3}{35}\right) \\ +c_1\frac{\nu^3}{\hat{K}^3}\left(-\frac{3}{40}e^{2\eta}+\frac{3}{28}e^{\frac{10}{3}\eta}-\frac{1}{24}e^{\frac{14}{3}\eta}+\frac{1}{105}\right) \\ +\frac{\nu^4}{\hat{K}^4}\left(\frac{5}{84}e^{\eta}-\frac{1}{44}e^{\frac{7}{3}\eta}-\frac{1}{60}e^{\frac{11}{3}\eta}+\frac{1}{228}e^{5\eta}+\frac{393011}{403788}\right)\Bigg]\end{aligned} \tag{2.37}$$

and so on. Using these, we can derive decay exponents for the velocity moments. The structure function exponent is defined as

$$\zeta_p \doteq -\lim_{\eta\to 0}\frac{d\left\langle u^p\right\rangle/d\eta}{\left\langle u^p\right\rangle}. \tag{2.38}$$

We find

$$\zeta_1 = \frac{1}{3}+\frac{1}{c_1}\frac{4}{3}\frac{\nu}{\hat{K}} \tag{2.39}$$

$$\zeta_2 = \frac{2}{3}+\frac{c_1}{c_2}\frac{2}{3}\frac{\nu}{\hat{K}} \tag{2.40}$$

$$\zeta_3 = 1-\frac{\nu}{\hat{K}}\frac{c_2}{c_3}+\frac{1}{c_3}\frac{1}{6}\frac{\nu^2}{\hat{K}^2} \tag{2.41}$$

$$\zeta_4 = \frac{4}{3}-\frac{\nu}{\hat{K}}\frac{c_3}{c_4}\frac{4}{3}+\frac{\nu^2}{\hat{K}^2}\frac{c_2}{c_4}\frac{8}{315}-\frac{\nu^3}{\hat{K}^3}\frac{c_1}{c_4}\frac{1367}{1260}-\frac{\nu^4}{\hat{K}^4}\frac{1}{c_4}\frac{5053}{920997}, \tag{2.42}$$

giving analytic predictions for finite-viscosity deviations from the classic Kolmogorov scaling. Since we are interested in the case where $\nu \ll 1$, we can discard terms involving factors of ν^2. This yields a finite-viscosity correction

to the K41 structure function exponents:

$$\zeta_1 = \frac{1}{3} + \frac{1}{c_1}\frac{4}{3}\frac{\nu}{\hat{K}} \tag{2.43}$$

$$\zeta_2 = \frac{2}{3} + \frac{c_1}{c_2}\frac{2}{3}\frac{\nu}{\hat{K}} \tag{2.44}$$

$$\zeta_3 = 1 - \frac{\nu}{\hat{K}}\frac{c_2}{c_3} \tag{2.45}$$

$$\zeta_4 = \frac{4}{3} - \frac{\nu}{\hat{K}}\frac{c_3}{c_4}\frac{4}{3}. \tag{2.46}$$

Having an analytic description of the velocity moments for the steady state, we can calculate the energy dissipation rate for arbitrary viscosities. On taking the evolution equation and multiplying by u and integrating over η, we get the energy balance equation:

$$\frac{dE}{dt} = \int \frac{\partial u^2/2}{\partial t}\, d\eta = \int \left[-\hat{K}\partial_\eta\left(e^\eta u^3\right) - \nu e^{2\eta}u^2\right] d\eta.$$

The energy dissipation rate ϵ is defined as

$$\epsilon \doteq \left.\frac{dE}{dt}\right|_{\text{dissipative}} = -\nu \int e^{2\eta}u^2\, d\eta.$$

The time-averaged dissipation of a statistically steady-state continuous shell model is given by

$$\begin{aligned}\langle\epsilon\rangle &= -\nu \int_{\eta_0}^{\eta_d} e^{2\eta}\left\langle u^2\right\rangle d\eta \\ &= -\nu\left[c_2\frac{3}{4}e^{\frac{4}{3}\eta} + c_1\frac{\nu}{\hat{K}}\left(-\frac{3}{8}e^{\frac{4}{3}\eta} + \frac{3}{16}e^{\frac{8}{3}\eta}\right) + \frac{\nu^2}{\hat{K}^2}\left(\frac{1}{48}e^{4\eta} - \frac{1}{16}e^{\frac{4}{3}\eta} + \frac{1}{16}e^{\frac{8}{3}\eta}\right)\right]_{\eta_0}^{\eta_d}.\end{aligned} \tag{2.47}$$

Using the definition of the dissipation wavenumber $k_d = e^{\eta_d}$ with η_d defined as in equation (2.29), and taking the limit as $\nu \to 0$, the dissipation approaches

$$\langle\epsilon\rangle = -\hat{K}\left(3c_2c_1 + 80c_1^3\right). \tag{2.48}$$

That is, taking steady-state solutions and then letting the viscosity approach zero, the energy dissipation from the viscous term tends towards a finite non-zero limit, just as Kolmogorov hypothesized would occur for a similar limit of the Navier–Stokes equations.

2.B.6 Recursive Solutions

Given that we are looking for steady-state solutions, we can verify the results of section 2.B.5 by solving the discrete shell-model equations recursively.

Recursive Solutions of the DN model

The steady state of the discrete DN model is given by

$$\nu k_n^2 u_n = i k_n \left[a\left(u_{n-1}^2 - \lambda u_n u_{n+1}\right) + b\left(\lambda u_n u_{n-1} - u_{n+1}^2\right)\right]^*. \tag{2.49}$$

We introduce the variable v_n, $u_n = v_n k_n^{-1/3}$ for convenience; when u_n obeys a K41 scaling, v_n is constant. Then

$$\nu k_n^{5/3} v_n = i k_n^{1/3} \left[a\lambda^{2/3}\left(v_{n-1}^2 - v_n v_{n+1}\right) + b\lambda^{1/3}\left(v_{n-1} v_n - v_{n+1}^2\right)\right]^*, \tag{2.50}$$

which gives

$$\begin{aligned} v_{n+1} = {} & \frac{-a}{2b}\lambda^{1/3} v_n \\ & \pm \frac{\sqrt{k_n^{2/3} a^2 \lambda^{4/3} v_n^2 + 4 k_n^{1/3} b \lambda^{1/3}\left[k_n^{1/3}\left(a\lambda^{2/3} v_{n-1}^2 + b\lambda^{1/3} v_{n-1} v_n\right) - i\nu k_n^{5/3} v_n^*\right]}}{-2 k_n^{1/3} b \lambda^{1/3}}. \end{aligned} \tag{2.51}$$

Equation (2.51) can be solved quite easily on a computer. Keeping the velocity complex and treating a and b as real, the energy spectrum matches

the Kolmogorov prediction until near the dissipation scale, by which point the imaginary component of the velocity has reached the same magnitude as the real component of the velocity, and the solution diverges at wavenumbers higher than the dissipation wavenumber (see Figure 2.1). If we consider ia, ib,

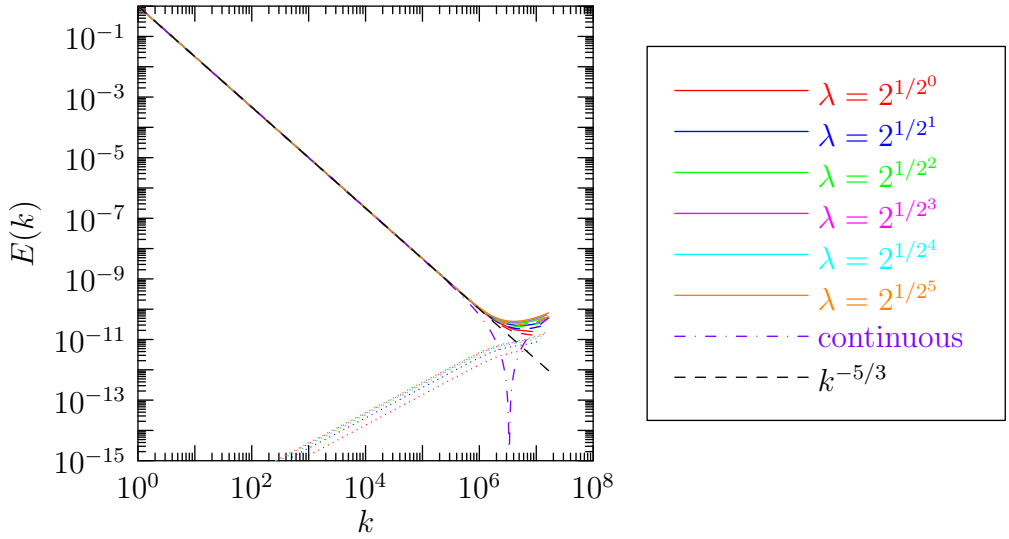

Figure 2.1: Recursive solution of the complex-valued DN model, with constant, real boundary conditions. Imaginary contribution shown dotted.

and the velocity u_n to be real, then the system goes to zero at the dissipation scale and does not diverge at higher wavenumbers. See Figure 2.2.

When $b = 0$, equation (2.51) is invalid, and we must instead solve

$$\nu k_n^2 u_n = i k_n a \left(u_{n-1}^2 - \lambda u_n u_{n+1} \right) \tag{2.52}$$

for u_{n+1}, giving

$$v_{n+1} = \frac{v_{n-1}^2}{v_n} - \frac{\nu}{-i\lambda a} \frac{k_n^{4/3}}{\lambda^{2/3}} \frac{v_n^*}{v_n}. \tag{2.53}$$

Unfortunately, this recursion formula is not numerically robust.

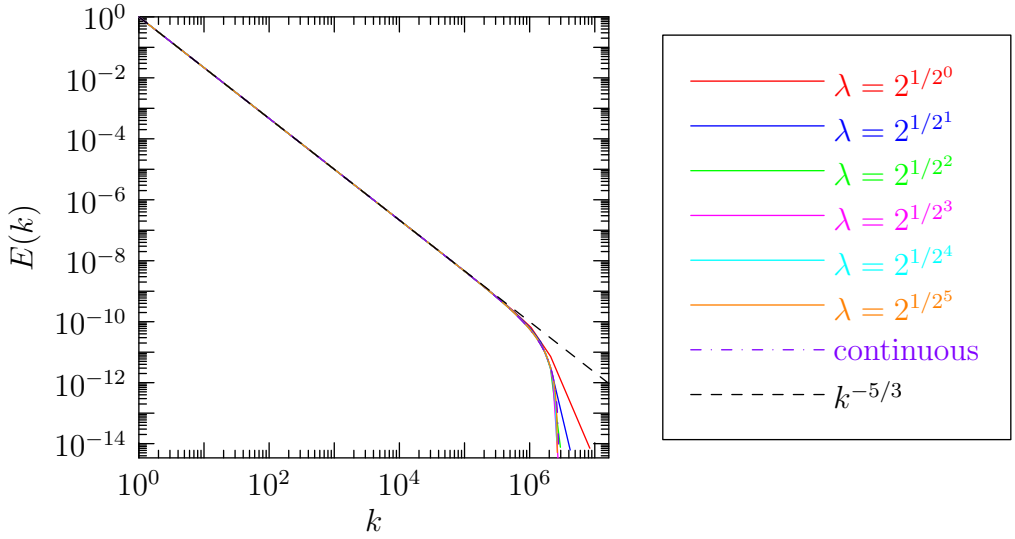

Figure 2.2: Recursive solution of the real-valued DN model.

Recursive Solutions of the GOY model

The GOY model presents a simpler problem. The stationary case is

$$\nu k_n^2 u_n = ik_n \left(\alpha u_{n+1} u_{n+2} + \frac{\beta}{\lambda} u_{n-1} u_{n+1} + \frac{\gamma}{\lambda^2} u_{n-1} u_{n-2} \right)^* + F_n, \quad (2.54)$$

which, using v_n, $u_n = v_n k_n^{-1/3}$ becomes

$$\nu k^{5/3} v_n = \frac{ik^{1/3}}{\lambda} (\alpha v_{n+2} v_{n+1} + \beta v_{n-1} v_{n+1} + \gamma v_{n-1} v_{n-2})^* + F_n. \quad (2.55)$$

We may solve for v_{n+2} uniquely:

$$v_{n+2} = \frac{i\lambda\nu k^{4/3} v_n^* - \beta v_{n-1} v_{n+1} - \gamma v_{n-1} v_{n-2} - F_n^*}{\alpha v_{n+1}}. \quad (2.56)$$

We solve this with $v_0 = v_1 = 1$ in Figure 2.3. Solutions with smaller λ demonstrate increasing amounts of noise in the inertial range. This noise is likely attributable to the well-known period-three oscillation exhibited by the GOY model.

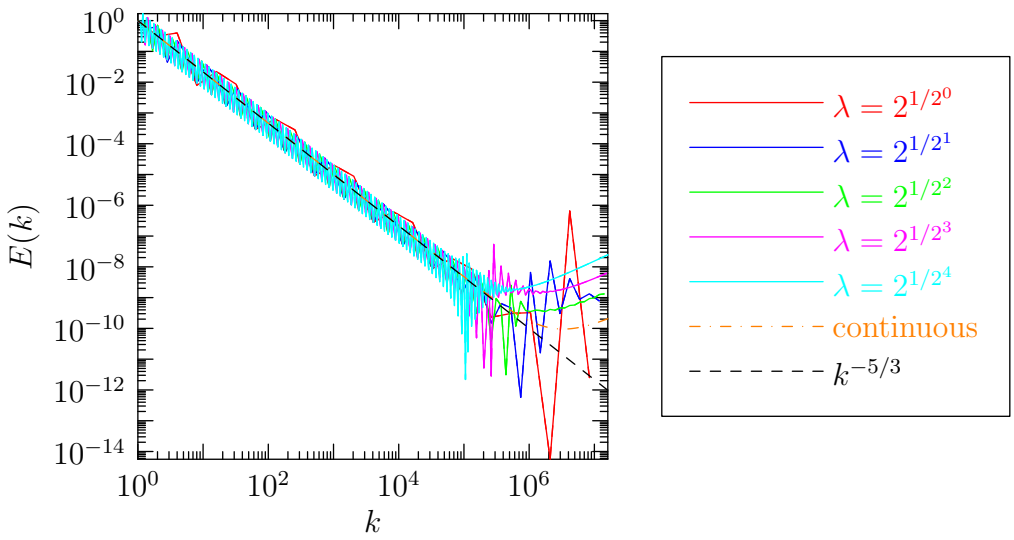

Figure 2.3: Recursive solution of the GOY model.

2.C Simulations

2.C.1 Dissipation

The steady-state continuum shell model with real-valued velocity yields analytic solutions. One prediction from this is that velocity goes to zero at wavenumber e^{η_d}, as given in equation (2.29). As can be seen in Figure 2.4, this prediction holds reasonably well for even large λ, and becomes more and more accurate as $\lambda \to 1$.

The energy spectra of finite-resolution shell models with complex-valued velocity does not go to zero as sharply as those with real-valued velocity. However, equation (2.29) provides a useful prediction even in this case, as can be seen in Figure 2.5. The simulations depicted were forced with a white-noise force instead of a constant boundary mode, and the nonlinear coupling coefficients were treated as real. The definition of the dissipation wavelength as the wavelength for which the half the energy dissipation occurs at higher wavenum-

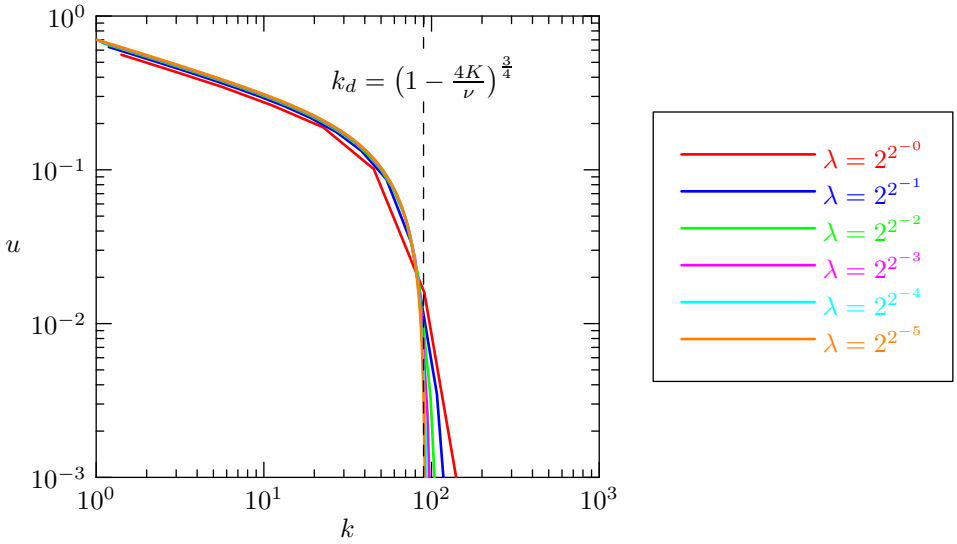

Figure 2.4: Rescaled DN model real velocities, $ia = -1$, $b = 0$.

bers and half at lower wavenumbers is taken from Bowman *et al.* [2006].

2.C.2 Energy Spectra of Simulations with White-Noise Forcing

White-noise forcing allows one to control the average energy-injection rate, which is useful when looking at turbulence in terms of energy cascades. As may be seen in Figure 2.6, simulations of the complex DN model with constant forcing-amplitude exhibit increased energy at low wavenumbers and an advanced dissipation range. In order to rectify this, one must rescale the energy-injection rate by $1/\log\lambda$. Under this modification, the spectra and dissipation wavelengths approach an asymptotic limit as $\lambda \to 1$, as shown in Figure 2.7, which closely matches the K41 prediction.

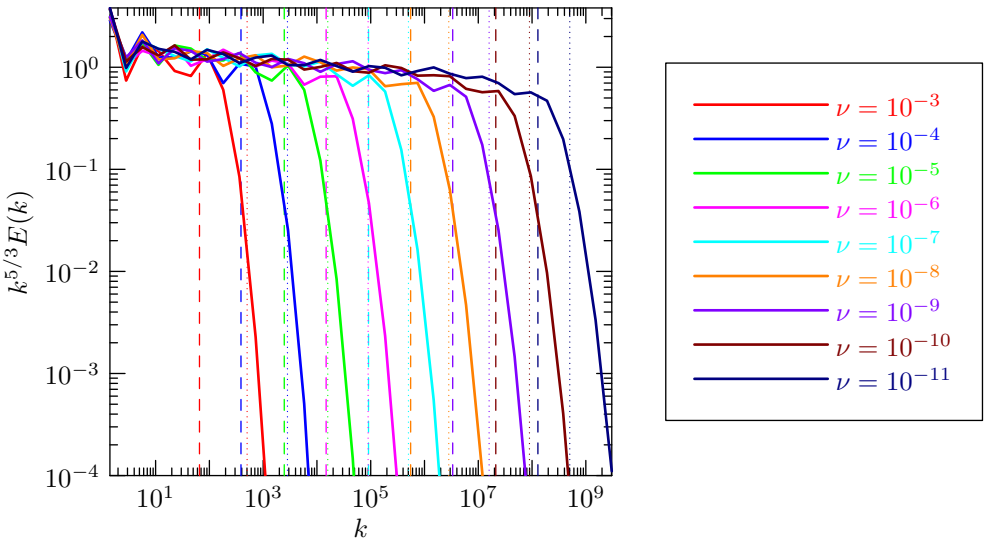

Figure 2.5: Energy spectra DN simulations with different values of ν. The vertical dashed lines are the dissipation wavenumber as given in equation (2.29), and the vertical dotted lines are the dissipation wavenumber defined as being the median wavenumber for energy dissipation.

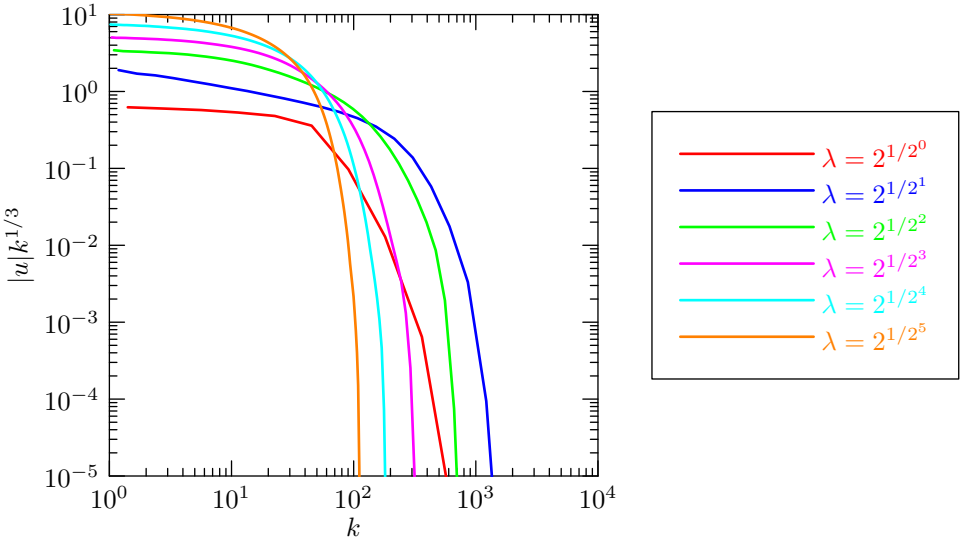

Figure 2.6: Energy spectra from rescaled DN model with $a = -1/4$, $b = -1$.

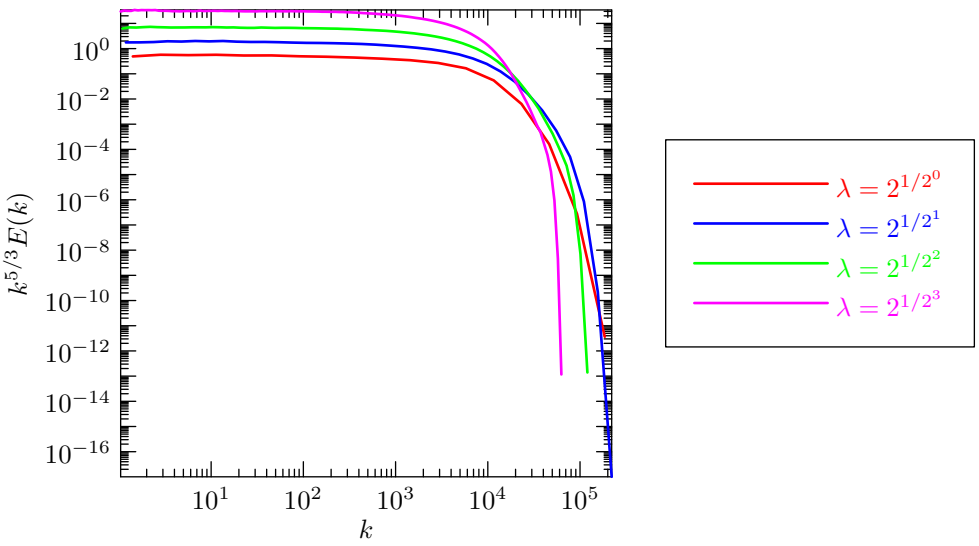

Figure 2.7: Rescaled energy for rescaled DN model with $a = -1$, $b = -1$.

2.C.3 Structure Functions and Anomalous Scaling

Real-valued shell models replicate K41 structure-function scaling as may be seen in Figure 2.8. In contrast, complex-valued shell models exhibit modifications to the anomalous scaling exponents.

For standard, complex-valued shell models, anomalous scaling seems to decrease as $\lambda \to 1$, even if the nonlinear term is rescaled. This can be seen in Figures 2.9, 2.10, 2.11, and 2.12. The effect of resolution changes on corrections in structure-function exponents suggests that anomalous scaling in shell models of turbulence may not be due to back-scatter since these terms become less significant when the nonlinear term is rescaled.

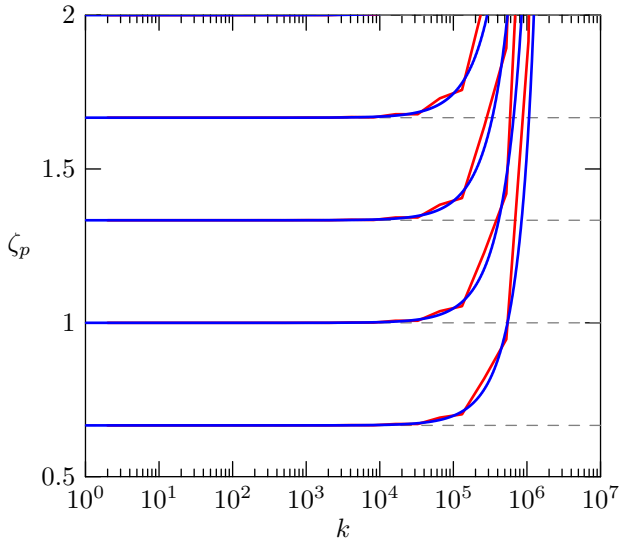

Figure 2.8: Structure function exponents for $p = 2, 3, 4, 5$ for the DN model, $ia = -1$, $b = 0$, $\nu = 10^{-8}$. Shell spacing factors are $\lambda = 2$ (blue) and $\lambda = 2^{1/2^5}$ (red). Here, ζ_p is shown for $p = 4, 3, 2$, top to bottom.

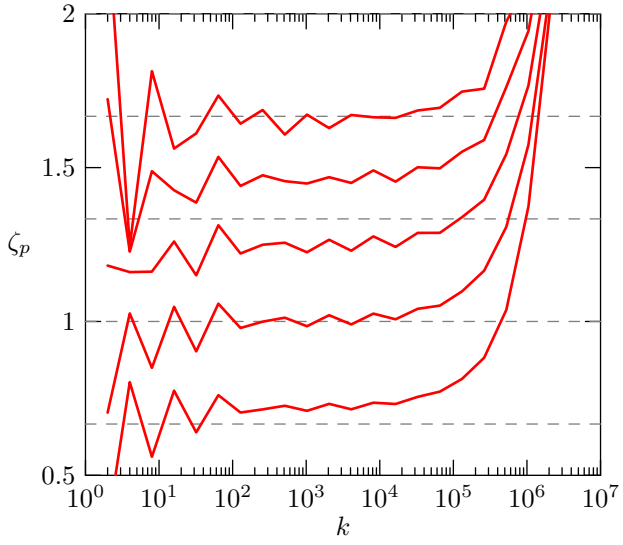

Figure 2.9: Structure function exponents for the DN model with complex velocity, $a = -1$, $b = -1$ $\lambda = 2$, and $\nu = 10^{-8}$. Here, ζ_p is shown for $p = 6, 5, 4, 3, 2$, top to bottom.

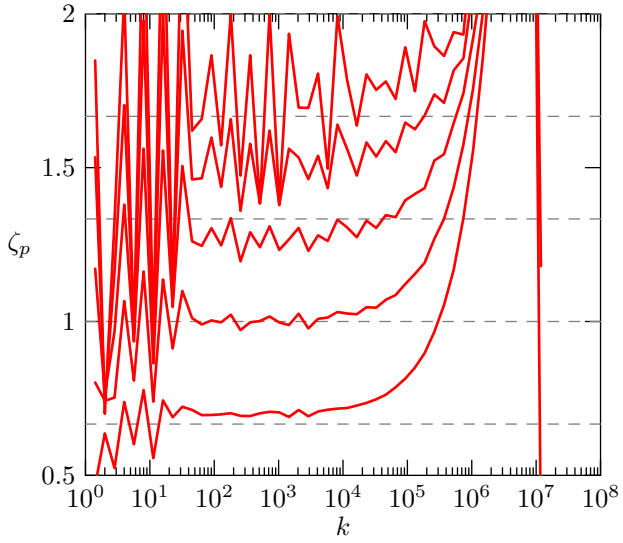

Figure 2.10: Structure function exponents for the DN model with complex velocity, $a = -1$, $b = -1$, $\nu = 10^{-8}$, $\lambda = 2^{1/2}$. Here, ζ_p is shown for $p = 6, 5, 4, 3, 2$, top to bottom.

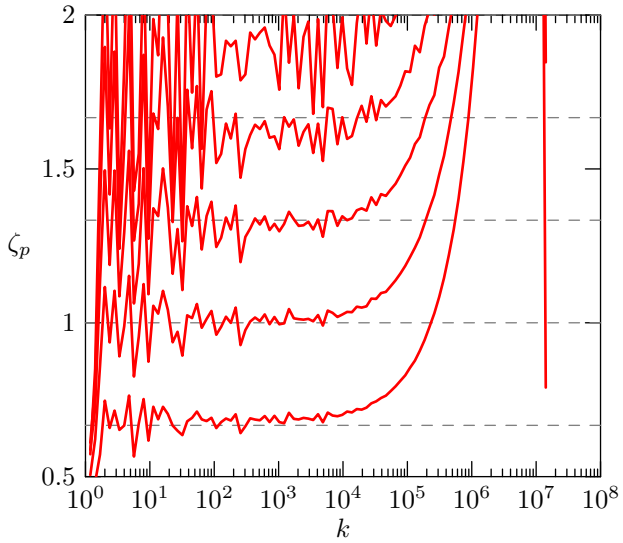

Figure 2.11: Structure function exponents for the DN model with complex velocity, $(a, b) = (-1, 1)$, $bc_0 = 1$, $\lambda = 2$, and $\nu = 10^{-8}$ $\lambda = 2^{1/2^2}$. Here, ζ_p is shown for $p = 6, 5, 4, 3, 2$, top to bottom.

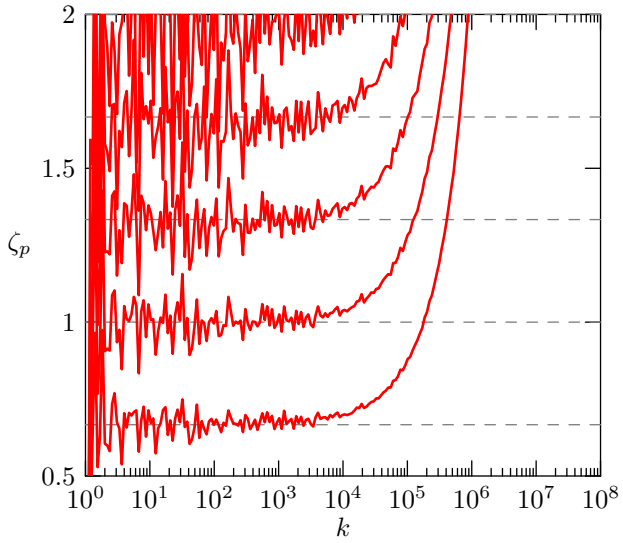

Figure 2.12: Structure function exponents for the DN model with complex velocity, $(a, b) = (-1, 1)$, $bc_0 = 1$, $\lambda = 2$, and $\nu = 10^{-8}$ $\lambda = 2^{1/2^3}$. Here, ζ_p is shown for $p = 6, 5, 4, 3, 2$, top to bottom.

Chapter 3

Implicitly Dealiased Convolutions

Wherein we discuss several methods for the efficient calculation of the convolution of functions by means of Fourier transforms. Care is given to remove aliasing errors caused by Fourier transforms treating all arrays as periodic.

The *convolution* of the functions F and G in $L^2(\mathbb{R})$ is written $F * G$ and defined as

$$(F * G)(k) \doteq \int F(\kappa)G(k-\kappa)\,d\kappa. \tag{3.1}$$

This definition can extended to other function spaces, such as $L^2(\mathbb{C})$, for example. The convolution is an important tool in such areas as image filtering, digital signal processing, correlation analysis, and when performing spectral simulations involving nonlinear differential equations. The convolution has some useful properties, such as commutativity and associativity. The convolutive identity is the Dirac delta function, with $(F * \delta(k)) = (\delta(k) * F) = F(k)$.

The definition in equation (3.1) is equivalent to

$$(F * G)(k) = \int \int F(\kappa_1) G(\kappa_2) \delta(k - \kappa_1 - \kappa_2) \, d\kappa_1 \, d\kappa_2. \tag{3.2}$$

Using this notation, one may create n-ary convolutions, defined as

$$* \left(F_1, \ldots, F_n\right)(k) \doteq \int \ldots \int F(\kappa_1) \ldots F(\kappa_n) \delta \left(k - \sum_{i=1}^{n} \kappa_i \right) d\kappa_1 \ldots d\kappa_n. \tag{3.3}$$

Moreover, one is able to define the unary convolution, $*(f)(t) = f(t)$, and the nullary convolution, $*()(t) = \delta(t)$, i.e. the empty convolution simply returns the identity for the binary convolution.

The convolution of two functions F and G in $\ell^2(\mathbb{R})$ (or $\ell^2(\mathbb{C})$, etc.) is defined similarly, with

$$\begin{aligned} (F * G)_k &\doteq \sum_{\kappa \in \mathbb{Z}} F_\kappa G_{k-\kappa} \\ &= \sum_{\kappa_1, \kappa_2 \in \mathbb{Z}} F_{\kappa_1} G_{\kappa_2} \delta_{\ell, \kappa_1 + \kappa_2}, \end{aligned} \tag{3.4}$$

with $\delta_{\ell, p_1 + p_2}$ the Kronecker delta. As per equation (3.3), the n-ary discrete convolution is

$$* \left(F_1, \ldots, F_n\right)_k \doteq \sum_{\kappa_1, \ldots, \kappa_n \in \mathbb{Z}} F_{1, \kappa_1} \ldots F_{n, \kappa_n} \delta_{k, \kappa_1 + \cdots + \kappa_n}. \tag{3.5}$$

Discrete convolutions are important in many numerical applications, where the input vectors will generally be *non-centered*, with $\{F_k\}_{k=0}^{m-1}$ and $\{G_k\}_{k=0}^{m-1}$, and

$$\{F * G\}_k = \sum_{\kappa=0}^{k} F_\kappa G_{k-\kappa}, \quad k = 0, \ldots, m-1,$$

or *centered*, with $\{F_k\}_{k=-m+1}^{m-1}$ and $\{G_k\}_{k=-m+1}^{m-1}$, and

$$\{F * G\}_k = \sum_{\kappa=-m+1}^{k-m+1} F_\kappa G_{k-\kappa}, \quad k = -m+1, \ldots, m-1.$$

Centered data is typical when the data comes from spectral simulations, in which case it is often *Hermitian-symmetric*, i.e. $F_{-k} = F_k^*$, where the asterisk denotes complex conjugation. Hermitian data in Fourier space is mapped to real-valued data in physical space, and such transforms can be done in half the time of a complex-to-complex transformation.

If the input vectors F and G are of length m, then calculation $F * G$ as in equation (3.4) requires $\mathcal{O}(m^2)$ operations and introduces significant numerical error when m is large. One can avoid calculating the sum directly by applying the convolution theorem:

$$\mathcal{F}^{-1}[(F * G)] = \mathcal{F}^{-1}[F]\,\mathcal{F}^{-1}[G], \tag{3.6}$$

where $\mathcal{F}^{-1}[F]$ is the inverse Fourier transform of F. Equation (3.6) applies equally the continuous and discrete case.

Finite-length discrete Fourier transforms may be calculated using an FFT, which takes, to leading order, only $Km \log m$ operations, with $K = 34/9$ optimal [Johnson & Frigo 2007], [Lundy & Van Buskirk 2007], for data of length m, and produces much less numerical error than using equation (3.4).

Given two input vectors $\{F\}_{k=0}^{m-1}$ and $\{G\}_{k=0}^{m-1}$, their linear convolution is given by

$$(F * G)_k = \sum_{\kappa=0}^{k} F_\kappa G_{k-\kappa}, \quad k = 0, \ldots, m-1. \tag{3.7}$$

The Fourier transform maps periodic data to periodic data. As such,

$$\begin{aligned} \{\mathcal{F}\big[\mathcal{F}^{-1}[F]\mathcal{F}^{-1}[G]\big]\}_k &= \{F *_m G\}_k \\ &\doteq \sum_{\kappa=0}^{m-1} F_{\kappa \bmod m} G_{(k-\kappa) \bmod m}, \end{aligned} \tag{3.8}$$

where $F *_m G$ is the *discrete cyclic convolution* of period m. The differences between $F *_m G$ and $F * G$ are called *aliasing errors*, and the sum in equation (3.8) is said to be *aliased*. The process of recovering the linear convolution

$F * G$ from the cyclic convolution $F *_m G$ is called *dealiasing*, and is accomplished by either *phase-shift dealiasing* (subsection 3.A.1) or *zero-padding* the input data (subsections 3.A.2 and 3.A.3).

3.A Dealiasing Binary Convolutions

3.A.1 Dealiasing Binary Convolutions via Phase-Shift Dealiasing

One-dimensional binary convolutions can be dealiased by taking the difference between the naïve FFT-based convolution and a convolution based on the shifted FFT [Patterson Jr. & Orszag 1971], [Canuto *et al.* 2006],

$$\mathcal{F}_\Delta^{-1}[F]_j \doteq \sum_{k=0}^{m-1} e^{\frac{2\pi i}{m}(j+\Delta)k} F_k. \tag{3.9}$$

Using the notation $f_\Delta \doteq \mathcal{F}_\Delta^{-1}[F]$, the phase-shifted convolution is

$$F *_\Delta G \doteq \mathcal{F}_\Delta(f_\Delta g_\Delta). \tag{3.10}$$

With the choice $\Delta = \frac{1}{2}$, the phase-shifted convolution equals

$$\{F *_\Delta G\}_k = \sum_{\kappa=0}^{k} F_\kappa G_{k-\kappa} - \sum_{\kappa=k+1}^{m-1} F_\kappa G_{k-\kappa+m}. \tag{3.11}$$

Since the (unshifted) cyclic convolution yields

$$\{F *_m G\}_\ell = \sum_{\kappa=0}^{k} F_\kappa G_{k-\kappa} + \sum_{\kappa=k+1}^{m-1} F_\kappa G_{k-\kappa+m}, \tag{3.12}$$

we may recover the linear convolution by taking the arithmetic mean of a cyclic and a shifted convolution,

$$F * G = \frac{1}{2}[(F *_m G) + (F *_\Delta G)]. \tag{3.13}$$

This requires computing two convolutions, each involving three FFTs of length m, and the computational complexity of this technique is therefore $6\,Km\log m$. Applying this technique to d dimensions requires 2^d convolutions to remove all aliases, with a computation complexity of $3\cdot 2^d d\,Km^d\log m$. Table 3.5 on page 84 shows this technique in comparison to other methods. Removing all aliases from d-dimensional via phase-shift dealiasing is very expensive.

Partial Phase-Shift Dealiasing

Because of the high cost of this method, one often dealiases only some of the modes when performing multi-dimensional convolutions. In three dimensions, all of the modes can be recovered via the 3-dimensional transform shifted once by $\Delta = \frac{1}{2}$ in each direction, i.e. we apply equation (3.9) in the x, y, and z directions.

Terms that are aliased in an odd number of directions (e.g. if the term is an aliasing error from the x convolution but not from the y or z convolutions) are multiplied by -1, whereas terms with aliasing errors from an even number of directions (e.g. if the term is an aliasing error from the x convolution and the z convolution but not the y convolution) are multiplied by 1. Thus, the single- and triple-aliased terms may be eliminated by taking a geometric mean as in equation (3.13), while the doubly aliased terms are removed from the system by setting all terms with $|k| > 2\sqrt{2}m/3$ to zero (i.e. zero-padding, see subsection 3.A.2), which also eliminates the triply aliased modes. The maximum wavenumber is then $2\sqrt{2}m/3 \approx 0.94m$ [Patterson Jr. & Orszag 1971], so one retains approximately $3.47m^3$ of the original $8m^3$ modes (see Table 3.6 on page 85). Since not all modes are retained, phase-shift dealiasing is not useful for many applications such as image filtering. In certain applications, this

loss is not so bad; for example, in pseudospectral simulations of turbulence, if one eliminates modes with wavenumber greater than m from the system, the time step is greatly reduced, while the loss of high-wavenumber modes is not so dire for energy dissipation since only a fraction of the modes with wavenumber greater than m were available in the first place.

3.A.2 Dealiasing Binary Convolutions via Explicit Zero Padding

Aliasing errors can also be removed by extending the input arrays with enough zeroes so that the extra terms are removed. This technique is known as *explicit zero-padding*, and removes all the aliasing errors at the expense of having to perform an FFT-based convolution on a larger data set.

One-Dimensional Non-Centered Binary Convolutions

The non-centered inputs vectors $\{f_\ell\}_{\ell=0}^{m-1}$ and $\{g_\ell\}_{\ell=0}^{m-1}$ are padded with zeroes to length $2m$. In this fashion, the $2m$-cyclic convolution equals the length m linear convolution:

$$\begin{aligned}(F *_{2m} G)_k &= \sum_{\kappa=0}^{2m-1} F_{\kappa(\mathrm{mod}\,2m)} G_{(k-\kappa)(\mathrm{mod}\,2m)} \\ &= \sum_{\kappa=0}^{m-1} F_\kappa G_{(k-\kappa)(\mathrm{mod}\,2m)} \\ &= \sum_{\kappa=0}^{k} F_\kappa G_{k-\kappa} \\ &= (F * G)_k, \quad k = 0, \ldots, m-1.\end{aligned} \tag{3.14}$$

This method requires 3 FFTs to be done on data of length $2m$, which may be done in $6Km \log m$ operations. The data must also be increased in length;

this requires $2m$ extra words of memory. This method is compared with other methods in Table 3.2 on page 80. The results of timing tests for this method are given in Figure 3.1 on page 80.

For centered inputs $\{F_k\}_{k=-m+1}^{m-1}$ and $\{G_k\}_{k=-m+1}^{m-1}$, the data need only be padded from length $2m-1$ to length $3m$ [Orszag 1971]:

$$\begin{aligned}(F *_{3m} G)_k &= \sum_{\kappa=0}^{2m-1} F_{\kappa(\mathrm{mod}\,3m)} G_{(k-\kappa)(\mathrm{mod}\,3m)} \\ &= \sum_{\kappa=0}^{m-1} F_\kappa G_{(k-\kappa)(\mathrm{mod}\,3m)} \\ &= \sum_{\kappa=0}^{k-m+1} F_\kappa G_{k-\kappa} \\ &= (F * G)_k, \quad k = -m+1, \ldots, m-1.\end{aligned} \tag{3.15}$$

This requires $2m$ extra words of memory and 3 FFTs of length $3m$, for a computational complexity of $\frac{9}{2} K m \log m$. See Table 3.5 on page 79 a comparison to other methods, and Figure 3.4 on page 84 for timing tests.

Higher-Dimensional Centered Binary Convolutions

The convolution of two functions F and G in $L^2(\mathbb{R}^d)$ or $L^2(\mathbb{C}^d)$ is defined as

$$F * G \doteq \int \int F(\boldsymbol{\ell}_1) G(\boldsymbol{\ell}_2)\, \delta_{\boldsymbol{k}-\boldsymbol{\ell}_1-\boldsymbol{\ell}_2}\, d\boldsymbol{\ell}_1\, d\boldsymbol{\ell}_2, \tag{3.16}$$

where bold notation denotes vector quantities. The convolution of F and G in $\ell^2(\mathbb{R}^d)$ or $\ell^2(\mathbb{C}^d)$ is defined as

$$F * G \doteq \sum_{\boldsymbol{\ell}_1, \boldsymbol{\ell}_1 \in \mathbb{Z}^d} F(\boldsymbol{\ell}_1) G(\boldsymbol{\ell}_2)\, \delta_{\boldsymbol{k}, \boldsymbol{\ell}_1+\boldsymbol{\ell}_2}. \tag{3.17}$$

Higher-dimensional convolutions are dealiased by performing a one-dimensional dealiased convolution in each direction. The computational cost of such

a method for a non-centered convolution is

$$3 \cdot 2^d d\, K m^d \log m, \tag{3.18}$$

and requires $2(2^d - 1)m^d$ extra words of memory just for zero-padding. See Table 3.3 on page 81 for comparisons to other methods and Figure 3.2 on page 82 and Figure 3.3 on page 82 for timing tests. A centered Hermitian convolution has complexity

$$\frac{3^{d+1}}{2} d\, K m^d \log m, \tag{3.19}$$

also requiring $2(2^d - 1)m^d$ words of work memory. A comparison to other methods is given in Table 3.6 on page 85 and timing tests are given in Figure 3.5 on page 86.

In theory, the computational complexity may be reduced upon noticing that some Fourier transforms are performed on data known a priori to be zero. For a *p/q-rule* dealiased convolution, one must pad the data from length m to length mq/p, i.e. one performs a Fourier transform on data that is non-zero for p/q of its length. In d dimensions, the first Fourier transform can skip all but $(p/q)^{d-1}$ of the data. The second Fourier transform is applied to only $(p/q)^{d-2}$ of the data, and so on, with the last transform being applied to $(p/q)^{d-d} = 1$, i.e. all of the data. In this fashion, the complexity of a p/q-padded convolution in d dimensions may be decreased by a factor

$$\frac{\sum_{i=1}^{d} \left(\frac{p}{q}\right)^{d-i}}{d} = \frac{1}{d}\left[\frac{1 - \left(\frac{p}{q}\right)^d}{1 - \frac{p}{q}}\right]. \tag{3.20}$$

For the non-centered case, $p/q = 1/2$, this is equal to

$$\frac{2}{d}\left[1 - \left(\frac{1}{2}\right)^d\right], \tag{3.21}$$

which reduces the computational complexity of a non-centered d-dimensional binary convolution to

$$6\left(2^d - 1\right)\, Km^d \log m, \tag{3.22}$$

which is compared with other methods in Table 3.3 on page 81. For the centered Hermitian case, the savings from *pruning transforms* is

$$\frac{3}{d}\left[1 - \left(\frac{2}{3}\right)^d\right], \tag{3.23}$$

which reduces the computational complexity to

$$\frac{9}{2}\left(3^d - 2^d\right)\, Km^d \log m, \tag{3.24}$$

which is compared in Table 3.6 on page 85. The memory requirements for both types of convolution remain the same. In practice, *pruning* such transforms (i.e. skipping transforms whose results is known a priori to be zero) is often not useful because existing FFT libraries are generally optimized to deal with data in contiguous blocks, and convolution calculations that attempt to take advantage of the fact that transforms of zero vectors are also zero can be slower than the unpruned calculation [Bowman & Roberts 2011a].

3.A.3 Dealiasing Binary Convolutions via Implicit Zero Padding

Explicitly dealiased convolutions perform calculations on data that is known a priori to be zero. Reading zero data from memory and performing calculations on it can be avoided by using implicitly padded Fourier transforms, which form the basis for the implicitly padded FFT-based convolution.

Implicitly Zero-Padded FFTs

Suppose that we would like to take the Fourier transform of an input vector F of length qm, of which only a fraction p/q is non-zero. Let pm be the length of the non-zero portion. The Fourier transform of F is

$$f_j = \mathcal{F}^{-1}\left[F\right]_j = \sum_{k=0}^{qm} \zeta_{qm}^{jk} F_k \tag{3.25}$$

where $\zeta_N = \exp(-i2\pi/N)$ is the N^{th} root of unity. We can reduce the range of summation by using the fact that $F_k = 0$ if $k \geq pm$, equation (3.25) can be reduced to

$$f_j = \sum_{k=0}^{pm} \zeta_{qm}^{jk} F_k \tag{3.26}$$

Unfortunately, equation (3.26) is not a Fourier transform, and is not amenable to calculation via an FFT. The computational cost of computing f_j for $j = 0, \ldots, qm-1$ via equation (3.26) is asymptotic to qpm^2, compared to $qm \log qm$ by using equation (3.25) and an FFT.

We rearrange the output of equation (3.26) into groups with $k = q\ell + r$, with $\ell = 0, \ldots, m-1$ and $r = 0, \ldots, q-1$. Then, one computes f via the *implicitly padded Fourier transform*,

$$f_{q\ell+r} = \sum_{k=0}^{pm-1} \zeta_{qm}^{(q\ell+r)k} F_k = \sum_{s=0}^{m-1} \zeta_m^{\ell s} \sum_{t=0}^{p-1} \zeta_{qm}^{r(tm+s)} F_{tm+s}, \tag{3.27}$$

which is the Fourier transform of length m of $\sum_{t=0}^{p-1} \zeta_{qm}^{r(tm+s)} f_{tm+s}$, and therefore amenable to calculation via an FFT. Since r ranges from 0 to $q-1$, we require q such transforms, yielding a leading-order computational complexity of $qm \log m$. The inverse of equation (3.27) is

$$F_k = \frac{1}{qm} \sum_{\ell=0}^{m-1} \zeta_{qm}^{-rk} \sum_{r=0}^{q-1} \zeta_{qm}^{-kl} f_{q\ell+r}, \qquad k = 0, \ldots, pm-1. \tag{3.28}$$

Note that while the output of equation (3.27) is scrambled, $f_{q\ell+r}$ need not be unscrambled before taking the inverse transformation as given in equation (3.28). To perform an implicitly p/q-padded convolution on input vectors F and G, one determines $f_{q\ell+r}$ and $g_{q\ell+r}$, multiplies the scrambled output term-by-term to get $f_{q\ell+r}g_{q\ell+r}$, and then uses equation (3.28) to determine

$$(F * G)_k = \frac{1}{qm} \sum_{\ell=0}^{m-1} \zeta_{qm}^{-rk} \sum_{r=0}^{q-1} \zeta_{qm}^{-kl} \, f_{q\ell+r} \, f_{q\ell+r}, \qquad k = 0, \ldots, pm-1. \quad (3.29)$$

Calculating the Fourier-transformed one-dimensional convolution as per equation (3.27) offers only a minimal decrease in computational complexity over explicit zero-padding, but removes the restriction that the data and work arrays must be contiguous. As will be seen in the following sections, this seemingly inconsequential advantage allows one to take convolutions of multi-dimensional data while using less memory than explicitly padded convolutions. Moreover, while the convolution may be done in-place, all but two of the sub-transforms may be done out-of-place, giving one the option of doing out-of-place FFTs, which are typically faster than in-place FFTs (cf. Figures 1–6 of Frigo & Johnson [2005]), while not increasing the memory footprint of the convolution calculation.

Non-Centered Binary One-Dimensional Convolutions

Non-centered binary convolutions require "1/2" padding rule, i.e. $p = 1$, and $q = 2$. The implicitly 1/2-padded Fourier transform for the input $\{F_k\}_{k=0}^{m-1}$ is

$$f_{2\ell} = \sum_{k=0}^{m-1} \zeta_m^{\ell k} \, F_k, \qquad f_{2\ell+1} = \sum_{k=0}^{m-1} \zeta_{2m}^{\ell k} \, \zeta_{2m}^{k} F_k, \quad (3.30)$$

which is described in procedure fftpadBackward for complex data. The inverse of equation (3.30) is

$$f_j = \frac{1}{2m}\left(\sum_{p=0}^{m-1} \zeta_m^{kp} F_{2k} + \zeta_{2m}^{-j} \sum_{p=0}^{m-1} \zeta_m^{kp} F_{2k+1}\right), \quad k = 0, \ldots, m-1 \quad (3.31)$$

and given algorithmically in procedure fftpadForward. The convolution of input vectors f and g attained performed by first applying equation (3.30), multiplying the scrambled output term-by-term, then inverting the transform by applying equation (3.31) to produce the dealiased convolution, as shown in function cconv. This technique is compared to other methods in Table 3.2 on page 80, and the results from timing tests are given in Figure 3.1 on page 80. These algorithms are implemented for complex data in Bowman & Roberts [2010].

Input: vector f
Output: vector f, vector u
for $k = 0$ **to** $m - 1$ **do**
  u$[k] \leftarrow \zeta_{2m}^{k}f[k]$;
end
f $\leftarrow$ fft^{-1}(f);
u $\leftarrow$ fft^{-1}(u);

Procedure `fftpadBackward(f,u)` stores the scrambled $2m$-padded backward Fourier transform values of a vector f of length m in f and an auxiliary vector u of length m.

One-Dimensional Centered Binary Convolutions

For centered data, one uses a "2/3" padding rule, i.e. $p = 2$ and $q = 3$. The implicitly 2/3-padded Fourier transform of the input $\{F_k\}_{k=-m+1}^{m-1}$ is given by

$$f_{3\ell+r} = \sum_{\kappa=1}^{m-1} \zeta_m^{\ell\kappa} \zeta_{3m}^{r(\kappa-m)} F_{\kappa-m} + \sum_{k=0}^{m-1} \zeta_m^{\ell k} \zeta_{3m}^{rk} F_k. \quad (3.32)$$

Input: vector f, vector u
Output: vector f
f $\leftarrow$ fft(f);
u $\leftarrow$ fft(u);
for $k = 0$ **to** $m - 1$ **do**
  f$[k] \leftarrow$ f$[k] + \zeta_{2m}^{-k}$u$[k]$;
end
return f/(2m);

Procedure fftpadForward(f,u) returns the inverse of fftpadBackward(f,u).

Input: vector f, vector g
Output: vector f
u $\leftarrow$ fft^{-1}(f);
v $\leftarrow$ fft^{-1}(g);
u $\leftarrow$ u $*$ v;
for $k = 0$ **to** $m - 1$ **do**
  f$[k] \leftarrow \zeta_{2m}^{k}f[k]$;
  g$[k] \leftarrow \zeta_{2m}^{k}g[k]$;
end
v $\leftarrow$ fft^{-1}(f);
f $\leftarrow$ fft^{-1}(g);
v $\leftarrow$ v $*$ f;
f $\leftarrow$ fft(u);
u $\leftarrow$ fft(v);
for $k = 0$ **to** $m - 1$ **do**
  f$[k] \leftarrow$ f$[k] + \zeta_{2m}^{-k}$u$[k]$;
end
return f/(2m);

Function cconv(f,g,u,v) computes an in-place implicitly dealiased convolution of two complex vectors f and g using two temporary vectors u and v, each of length m.

This transformation can be simplified by introducing

$$\widetilde{F}_{k,r} \doteq \begin{cases} F_0 & \text{if } k = 0, \\ \zeta_{3m}^{rk}(F_k + \zeta_3^{-r} F_{k-m}) & \text{if } 1 \le k \le m-1, \end{cases} \tag{3.33}$$

in which case

$$f_{3\ell+r} = \sum_{k=0}^{m-1} \zeta_m^{\ell k}\, \widetilde{F}_{k,r}. \tag{3.34}$$

The inverse transform is

$$F_k = \frac{1}{3m} \sum_{r=-1}^{1} \zeta_{3m}^{-rk} \sum_{\ell=0}^{m-1} \zeta_m^{-\ell k} f_{3\ell+r}, \qquad k = -m+1, \ldots, m-1. \tag{3.35}$$

This method is given algorithmically in Procedure conv and compared to other methods in Table 3.5 on page 84. The results from timing tests are given in Figure 3.4 on page 84.

Input: vector f
Output: vector f, vector u
$\mathsf{u}[0] \leftarrow \mathsf{f}[m-1]$;
for $k = 1$ **to** $m-1$ **do**
  $\mathsf{A} \leftarrow \zeta_{3m}^{k}\left[\operatorname{Re}\mathsf{f}[m-1+k] + \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)\operatorname{Re}\mathsf{f}[k]\right]$;
  $\mathsf{B} \leftarrow i\zeta_{3m}^{k}\left[\operatorname{Im}\mathsf{f}[m-1+k] + \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)\operatorname{Im}\mathsf{f}[k]\right]$;
  $\mathsf{f}[m-1+k] \leftarrow \mathsf{A} + \mathsf{B}$;
  $\mathsf{u}[k] \leftarrow \overline{\mathsf{A} - \mathsf{B}}$;
  $\mathsf{f}[0] \leftarrow \mathsf{f}[k]$;
  $\mathsf{f}[k] \leftarrow \mathsf{f}[k] + \mathsf{f}[m-1+k]$;
end
$\mathsf{f}[0, \ldots, m-1] \leftarrow \mathtt{fft}^{-1}(\mathsf{f}[0, \ldots, m-1])$;
$\mathsf{u}[m] \leftarrow \mathsf{f}[m-1]$;
$\mathsf{f}[m-1] \leftarrow \mathsf{u}[0]$;
$\mathsf{f}[m-1, \ldots, 2m-2] \leftarrow \mathtt{fft}^{-1}(\mathsf{f}[m-1, \ldots, 2m-2])$;
$\mathsf{u}[0, \ldots, m-1] \leftarrow \mathtt{fft}^{-1}(\mathsf{u}[0, \ldots, m-1])$;

Procedure `fft0padBackward(f,u)` stores the scrambled $3m$-padded centered backward Fourier transform values of a vector f of length $2m-1$ in f and an auxiliary vector u of length $m+1$.

Input: vector f, vector u
Output: vector f
$\mathsf{f}[m-1,\ldots,2m-2] \leftarrow \texttt{fft}(\mathsf{f}[m-1,\ldots,2m-2])$;
$\mathsf{u}[m] \leftrightarrow \mathsf{f}[m-1]$;
$\mathsf{f}[0,\ldots,m-1] \leftarrow \texttt{fft}(\mathsf{f}[0,\ldots,m-1])$;
$\mathsf{u}[0,\ldots,m-1] \leftarrow \texttt{fft}(\mathsf{u}[0,\ldots,m-1])$;
$\mathsf{u}[m] \leftarrow \mathsf{f}[0] + \mathsf{u}[m] + \mathsf{u}[0]$;
for $k = 1$ **to** $m-1$ **do**
  $\mathsf{f}[k-1] = \mathsf{f}[k] + \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)\zeta_{3m}^{-k}\mathsf{f}[m-1+k] + \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)\zeta_{3m}^{k}\mathsf{u}[k]$;
  $\mathsf{f}[m-1+k] = \mathsf{f}[k] + \zeta_{3m}^{-k}\mathsf{f}[m-1+k] + \zeta_{3m}^{k}\mathsf{u}[k]$;
end
$\mathsf{f}[m-1] \leftarrow \mathsf{u}[m]$;
return $\mathsf{f}/(3\mathrm{m})$;

Procedure `fft0padForward(f,u)` returns the inverse of `fft0padBackward(f,u)`.

Input: vector f
Output: vector f, vector u
$\mathsf{u}[0] \leftarrow \mathsf{f}[0]$;
$\mathsf{F} \leftarrow \overline{f[2c-1]}$;
$\mathsf{f}[2c-1] \leftarrow \mathsf{f}[0]$;
for $k = 1$ **to** $c-1$ **do**
  $\mathsf{A} \leftarrow \zeta_{6c}^{k}\left[\mathrm{Re}\,\mathsf{f}[k] + \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)\mathrm{Re}\,\mathsf{F}\right]$;
  $\mathsf{B} \leftarrow -i\zeta_{6c}^{k}\left[\mathrm{Im}\,\mathsf{f}[k] + \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)\mathrm{Im}\,\mathsf{F}\right]$;
  $\mathsf{f}[k] \leftarrow \mathsf{f}[k] + \mathsf{F}$;
  $\mathsf{u}[k] \leftarrow \mathsf{A} - \mathsf{B}$;
  $\mathsf{F} \leftarrow \overline{\mathsf{f}[2c-1-k]}$;
  $\mathsf{f}[2c-1-k] \leftarrow \mathsf{A} + \mathsf{B}$;
end

Procedure `build(f,u)` builds the FFT arrays required for Function `conv` from an unpadded vector f of length $2c$ into f and an auxiliary vector u of length $c+1$.

Input: vector f, vector g
Output: vector f
$\mathsf{F} \leftarrow \mathsf{f}[c]$;
`build` (f,u);
$\mathsf{C} \leftarrow \mathsf{f}[c]$;
$\mathsf{f}[c] \leftarrow 2\operatorname{Re}\mathsf{F}$;
$\mathsf{u}[c] \leftarrow \operatorname{Re}\mathsf{F} + \sqrt{3}\operatorname{Im}\mathsf{F}$;

$\mathsf{G} \leftarrow \mathsf{g}[c]$;
`build` (g,v);
$\mathsf{D} \leftarrow \mathsf{g}[c]$;
$\mathsf{g}[c] \leftarrow 2\operatorname{Re}\mathsf{G}$;
$\mathsf{v}[c] \leftarrow \operatorname{Re}\mathsf{G} + \sqrt{3}\operatorname{Im}\mathsf{G}$;

$\mathsf{u} \leftarrow$ `crfft`(u);
$\mathsf{v} \leftarrow$ `crfft`(v);
$\mathsf{v} \leftarrow \mathsf{v} * \mathsf{u}$;
$\mathsf{u} \leftarrow$ `rcfft`(v);

$\mathsf{v} \leftarrow$ `crfft`$(\mathsf{f}[0,\ldots,c])$;
$\mathsf{f}[0,\ldots,c] \leftarrow$ `crfft`$(\mathsf{g}[0,\ldots,c])$;
$\mathsf{v} \leftarrow \mathsf{v} * \mathsf{f}[0,\ldots,c]$;
$\mathsf{f}[0,\ldots,c] \leftarrow$ `rcfft`(v);

$\mathsf{S} \leftarrow \mathsf{f}[c-1]$;
$\mathsf{T} \leftarrow \mathsf{f}[c]$;
$\mathsf{f}[c-1] = \operatorname{Re}\mathsf{F} - \sqrt{3}\operatorname{Im}\mathsf{F}$;
$\mathsf{f}[c] = \mathsf{C}$;
$\mathsf{g}[c-1] = \operatorname{Re}\mathsf{G} - \sqrt{3}\operatorname{Im}\mathsf{G}$;
$\mathsf{g}[c] = \mathsf{D}$;

$\mathsf{v} \leftarrow$ `crfft`$(\mathsf{g}[c-1,\ldots,2c-1])$;
$\mathsf{g}[c-1,\ldots,2c-1] \leftarrow$ `crfft`$(\mathsf{f}[c-1,\ldots,2c-1])$;
$\mathsf{g}[c-1,\ldots,2c-1] \leftarrow \mathsf{g}[c-1,\ldots,2c-1] * \mathsf{v}$;
$\mathsf{v} \leftarrow$ `rcfft`$(\mathsf{g}[c-1,\ldots,2c-1])$;

for $k = 1$ **to** $c-2$ **do**
  $\mathsf{f}[k] = \mathsf{f}[k] + \zeta_{6c}^{-k}\mathsf{v}[k] + \zeta_{6c}^{k}\mathsf{u}[k]$;
  $\mathsf{f}[2c-k] = \overline{\mathsf{f}[k]} + \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)\zeta_{6c}^{k}\overline{\mathsf{v}[k]} + \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)\zeta_{6c}^{-k}\overline{\mathsf{u}[k]}$;
end
$\mathsf{f}[c-1] = S + \zeta_{6c}^{1-c}\mathsf{v}[c-1] + \zeta_{6c}^{c-1}\mathsf{u}[c-1]$;
$\mathsf{f}[c] = T - \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)\mathsf{v}[c] - \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)\mathsf{u}[c]$;
if $c > 1$ **then**
  $\mathsf{f}[c+1] = \overline{S} + \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)\zeta_{6c}^{c-1}\overline{\mathsf{v}[c-1]} + \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)\zeta_{6c}^{1-c}\overline{\mathsf{u}[c-1]}$;
end
s**return** f/(6c);

Function `conv(f,g,u,v)` uses Procedure `build` to compute an in-place implicitly dealiased convolution of centered Hermitian vectors f and g of length $2c$ using temporary vectors u and v of length $c+1$.

Implicitly Dealiased Binary d-Dimensional Convolutions

Higher-dimensional FFT-based convolutions are typically performed by taking the inverse Fourier transform of the inputs F and G in all d dimensions, multiplying the result, and transforming back to the original domain. One can also calculate the convolution in multiple stages; consider the two-dimensional convolution of $F_{(\ell_{1,1},\ell_{2,1})}$ and $G_{(\ell_{1,2},\ell_{2,2})}$, $\ell_{1,\cdot} = 0,\ldots,m_1-1$, $\ell_{2,\cdot} = 0,\ldots,m_2-1$:

$$\begin{aligned}&(F * G)_{(k_1,k_2)}\\ &\quad= \sum_{\substack{\ell_{1,1}=0,\ldots,m_1-1\\ \ell_{2,2}=0,\ldots,m_2-1}} \sum_{\ell_{1,2}=0,\ldots,m_1-1} F_{(\ell_{1,1},\ell_{2,1})}\ G_{(\ell_{1,2},\ell_{2,2})}\ \delta_{(k_1,\ell_{1,1}+\ell_{1,2})}\ \delta_{(k_2,\ell_{2,1}+\ell_{2,2})}.\end{aligned} \tag{3.36}$$

This can be rearranged to give

$$(F*G)_{(k_1,k_2)} = \sum_{\ell_{1,1},\ell_{2,1}=0}^{m_1-1} \delta_{(k_1,\ell_{1,1}+\ell_{1,2})} \left(\sum_{\ell_{1,2},\ell_{2,2}=0}^{m_2-2} \delta_{(k_2,\ell_{2,1}+\ell_{2,2})}\ F_{(\ell_{1,1},\ell_{2,1})}\ G_{(\ell_{1,2},\ell_{2,2})} \right), \tag{3.37}$$

that is, the two-dimensional convolution $F * G$ is equal to the convolution in the k_1-direction of the convolution of F and G in the k_2-direction. Applying this to FFT-based convolutions, one has

$$F * G = \mathcal{F}_{k_1}\mathcal{F}_{k_2}\left[\mathcal{F}_{k_2}^{-1}\mathcal{F}_{k_1}^{-1}[F] \cdot \mathcal{F}_{k_2}^{-1}\mathcal{F}_{k_1}^{-1}[G]\right], \tag{3.38}$$

where $\mathcal{F}_{k_i}$ is the Fourier transform in direction k_i, with inverse $\mathcal{F}_{k_i}^{-1}$. This extends from two dimensions to d dimensions in a straightforward fashion.

For explicitly zero-padded convolutions, there are two options: one can either extend the data with zero padding in all d dimensions before applying any transformation, or one can copy the data after each transformation to a new zero-padded memory buffer. In the first case, this has the computational complexity and memory requirements of simply Fourier-transforming

in all d dimensions at once. The second case has the possibility of using less memory, but the cost of copying the data is great enough that the resulting method would somewhat negate the advantage gained from a decreased memory footprint. Explicitly padded dealiased convolutions are compared to other methods in Table 3.3 on page 81 and Table 3.6 on page 85.

On the other hand, implicitly zero-padded Fourier transforms do not require that the work array be contiguous with the data arrays. With this in mind, we can use the one-dimensional implicitly padded convolution routines as building-blocks to create convolution routines in higher dimensions. For example, two-dimensional implicitly padded convolutions on non-centered data arrays F and G are achieved by first applying equation (3.30) in the k_2-direction. This requires a work array of size $m_1 m_2$ words per input array. One then applies equation (3.30) in the k_1-direction, multiplies the output, and then returns to the original domain by applying equation (3.31) in the k_1-direction and then the k_2-direction.

In d dimensions, one first performs an implicitly padded fast Fourier transform in the direction k_d, leaving a $d-1$-dimensional convolution still to be performed. If the resulting convolution is 1-dimensional, then one uses either Function cconv (for non-centered data) or Function conv (for centered Hermitian-symmetric data). Recursively defined functions are given in Function cconvd for non-centered data and Function convd for centered Hermitian-symmetric data. The d-dimensional convolution function requires two work arrays of size $m_1 \times \ldots \times m_d$. Each time the function is recursed, two additional work arrays of size $m_1 \times \ldots \times m_{d-1}$ are required. The total work memory required is then $2 \sum_{i=1}^{d} m_1 \times \ldots \times m_i$. Since all terms with $i \neq d$ are of lower dimension than the $i = d$ term, these terms are inconsequential. The leading-

order work array size requirement is then simply $2\,m_1 \times \ldots \times m_d$. Implicitly padded d-dimensional convolutions are compared favourably to other methods in Table 3.3 on page 81 and Table 3.6 on page 85. The results of timing tests are given in Figure 3.5 on page 86.

```
Input: vector f, vector g
Output: vector f
if d > 1 then
    for j = 0 to m_i − 1 do
        fftpadBackward(f, u);
        fftpadBackward(g, v);
        cconvd(f, g, d − 1);
        cconvd(u, v, d − 1);
        fftpadForward(f, u);
    end
else
    cconv(f, g);
end
return f;
```

Function `cconvd(f,g,d)` computes an in-place implicitly dealiased convolution of two complex vectors `f` and `g` using two temporary arrays `u` and `v`, each of size $m_1 \times \ldots \times m_d$. Fourier transforms are done in the direction k_d.

3.B Dealiasing n-ary Convolutions

The ℓ^{th} component of the n-ary convolution of n vectors $f^1, \ldots, f^n$ is defined as

$$*\big(f^1, \ldots, f^n\big)_\ell \doteq \sum_{\ell_1,\ldots,\ell_n \in \mathbb{Z}} f^1_{\ell_1} \ldots f^n_{\ell_n} \delta_{\ell_1+\cdots+\ell_n,\ell}.$$

```
Input: vector f, vector g
Output: vector f
if d > 1 then
    for j = 0 to m_i − 1 do
        fft0padBackward(f, u);
        fft0padBackward(g, v);
        convd(f, g, d − 1);
        convd(u, v, d − 1);
        fft0padForward(f, u);
    end
else
    conv(f, g);
end
return f;
```

Function `convd(f,g,d)` computes an in-place implicitly dealiased convolution of two Hermitian-symmetric complex vectors `f` and `g` using two temporary arrays `u` and `v`, each of size $m_1 \times \ldots \times m_d$. Fourier transforms are done in the direction k_d.

The non-centered n-ary convolution of the finite-length vectors $f_k^1, \ldots, f_k^n$, for $k = 0, \ldots, m-1$, is defined as

$$* \left(f^1, \ldots, f^n\right)_\ell \doteq \sum_{\ell_1, \ldots, \ell_n = 0}^{m-1} f_{\ell_1}^1 \ldots f_{\ell_n}^n \delta_{\ell_1 + \cdots + \ell_n, \ell}, \quad \ell = 0, \ldots, m-1,$$

and the centered n-ary convolution of the centered finite-length vectors $f_k^1, \ldots, f_k^n$, for $k = -m+1, \ldots, m-1$, is defined as

$$* \left(f^1, \ldots, f^n\right)_\ell \doteq \sum_{\ell_1, \ldots, \ell_n = -m+1}^{m-1} f_{\ell_1}^1 \ldots f_{\ell_n}^n \delta_{\ell_1 + \cdots + \ell_n, \ell}, \quad \ell = -m+1, \ldots, m-1.$$

Such higher-order convolutions arise when performing simulations of the compressible Navier–Stokes equations or when considering high-order Casimir invariants in pseudospectral simulations [Bowman 2011].

Theorem 3.1: *An n-ary convolution of the sequences $f^1, \ldots, f^n \in \ell^2$ can be*

computed via $n-1$ binary convolutions.

Proof.

$$\begin{aligned} *\left(f^1,\dots,f^n\right)_\ell &= \sum_{\ell_1,\dots,\ell_n\in\mathbb{Z}} f^1_{\ell_1}\cdots f^n_{\ell_n}\delta_{\ell_1+\cdots+\ell_n,\ell} \\ &= \sum_{\ell_1,\dots,\ell_n\in\mathbb{Z}} f^1_{\ell_1}\cdots f^n_{\ell_n}\delta_{\ell_1+\cdots+\ell_{n-1},\widehat{\ell}}\,\delta_{\ell_n+\widehat{\ell},\ell} \\ &= \sum_{\widehat{\ell},\ell_n\in\mathbb{Z}} f^n_{\ell_n}\delta_{\ell_n+\widehat{\ell},\ell} \sum_{\ell_1,\dots,\ell_{n-1}\in\mathbb{Z}} f^1_{\ell_1}\cdots f^{n-1}_{\ell_{n-1}}\delta_{\ell_1+\cdots+\ell_{n-1},\widehat{\ell}} \\ &= \left[*\left(f^1,\dots,f^{n-1}\right)*f^n\right]_\ell. \end{aligned} \tag{3.39}$$

By induction on n, we have $*(f^1,\dots,f^n) = f^1 * f^2 * \cdots * f^n$. □

By the convolution theorem and Theorem 3.1,

$$\mathcal{F}\left[*\left(f^1,\dots,f^n\right)\right] = \mathcal{F}\left[f^1\right]\dots\mathcal{F}[f^n] \tag{3.40}$$

One can also perform the binary convolutions iteratively since

$$\begin{aligned} *\left(f^1,\dots,f^n\right) &= f^1 * \left[*\left(f^2,\dots,f^n\right)\right] \\ &= \mathcal{F}^{-1}\left[\mathcal{F}\left[f^1\right]\ \mathcal{F}\left[*(f^2,\dots,f^n)\right]\right]. \end{aligned} \tag{3.41}$$

Unfortunately, Theorem 3.1 does not hold for linear convolutions of fixed finite length.

Theorem 3.2: *For general input vectors $f_1,\dots,f_n$ of fixed finite length,*

$$*\left(f_1,\dots,f_n\right) \neq f_1 * \left(f_2 * \dots\left(f_{n-1} * f_n\right)\dots\right). \tag{3.42}$$

Moreover, the right-hand side of equation (3.42) is not associative.

Proof. It suffices to consider the convolution of three non-centered vectors f, g, and h with two entries, i.e. with $m=2$ and $f=(f_{-1},f_0,f_1)$, $g=(g_{-1},g_0,g_1)$, and $h=(h_{-1},h_0,h_1)$. In this case, the ternary convolution is given by

$$*\left(f,g,h\right)_k = \sum_{a,b,c} f_a g_b h_c\,\delta_{a+b+c,k}, \quad k=-1,0,1. \tag{3.43}$$

$*(f,g,h)_1$			$(f*(g*h))_1$			
a	b	c	a	ℓ	b	c
1	0	0	1	0	0	0
0	1	0	0	1	1	0
0	0	1	0	1	0	1
1	1	-1	1	0	1	-1
1	-1	1	1	0	-1	1
-1	1	1	N/A			

Table 3.1: Comparison of terms contributing to the first component of the ternary convolution $*(f,g,h)_k = \sum_{a,b,c=-1}^{1} f_a g_b h_c \, \delta_{a+b+c,k}$ and the double binary convolution $(f*(g*h))_k = \sum_{a,\ell=-1}^{1} f_a (g*h)_\ell \delta_{a+\ell,k}$, with $(g*h)_\ell = \sum_{b,c=-1}^{1} g_b h_c \, \delta_{b+c,\ell}$.

Performing binary convolutions, we have, for example,

$$(f*(g*h))_k = \sum_{a,\ell} f_a \left(\sum_{b,c} g_b h_c \, \delta_{b+c,\ell} \right) \delta_{a+\ell,k}, \quad k = -1, 0, 1. \tag{3.44}$$

The terms contributing to the first component of equation (3.43) and equation (3.44) are shown in Table 3.1. Since $(g*h)$ has only three components, namely $(g*h)_{-1}$, $(g*h)_0$, and $(g*h)_1$, the intermediary variable ℓ can only equal 0 or 1. Thus, the last row, which would require $\ell = 2$ is excluded from the calculation of $f*(g*h)$, but is included in the calculation of $*(f,g,h)$. Therefore, $*(f,g,h) \neq (f*(g*h))$. Moreover, it is clear that $(f*(g*h))$, $((f*g)*h))$, and $(g*(f*h))$ will exclude different terms, implying that the finite-length binary convolution is not associative. □

Due to Theorem 3.1, one must be careful when performing higher-order convolutions, or taking the convolutions of vectors that are themselves the result of a convolution.

3.B.1 Dealiasing n-ary Convolutions via Zero Padding

Dealiasing n-ary Convolutions in 1 Dimension

Computing a non-centered n-ary convolution requires $n+1$ Fourier transforms. The input vectors must be zero-padded from length m to nm, i.e. "$1/n$" padding. The computational complexity is then

$$(n+1)n\,Km\log nm. \tag{3.45}$$

Dealiasing an n-ary convolution of centered Hermitian data requires the input data of length m to be padded with zeroes so that the resulting array has length $\frac{n+1}{2}m$, i.e. "$2/(n+1)$" padding. The computational complexity of this method is

$$\frac{(n+1)^2}{2}\,Km\log\frac{n+1}{2}m \tag{3.46}$$

for the centered case.

Dealiasing n-ary Convolutions in d Dimensions

Performing a convolution on n non-centered input arrays requires $(n+1)$ FFTs. The inputs must be padded from length m to length nm in each direction, so each transform has complexity $Kn^dm^d\log m$. Some transforms are performed on data that are known to be zero a priori, and can therefor be skipped. This reduces the computational complexity by a factor of

$$\frac{1}{d}\left[\frac{1-\left(\frac{1}{n}\right)^d}{1-\frac{1}{n}}\right]. \tag{3.47}$$

The total cost of performing an n-ary convolution in this fashion is then

$$n(n+1)\frac{n^d-1}{n-1}\,Km^d\log nm. \tag{3.48}$$

Implicit and explicit zero-padding techniques for dealiasing n-ary non-centered convolutions are compared in Table 3.4 on page 83.

For the centered case, the input arrays must be padded to length from length $2m-1$ to length $(n+1)m$ in each dimension, so each FFT takes

$$K((n+1)m)^d \log (n+1)m)^d \tag{3.49}$$

operations. The computational savings from pruning these transforms is

$$\frac{1}{d}\left[\frac{1-\left(\frac{2}{n+1}\right)^d}{1-\frac{2}{n+1}}\right]. \tag{3.50}$$

The total cost of the convolution is then

$$3(n+1)^d\left[\frac{1-\left(\frac{2}{n+1}\right)^d}{1-\frac{2}{n+1}}\right] Km^d \log m. \tag{3.51}$$

Implicit and explicit zero-padding techniques for dealiasing n-ary centered convolutions are compared in Table 3.7 on page 86.

3.C Comparison of Dealiasing Techniques

3.C.1 Non-Centered Convolutions

Non-Centered Binary Convolutions

Non-centered convolutions can be dealiased by either explicit or implicit zero-padding. As seen in Table 3.2, one-dimensional non-centered convolutions have identical theoretical performance characteristics, which is confirmed empirically in Figure 3.1. While implementing explicit zero-padding is more straightforward, implicit zero-padding has the important advantage of not requiring a contiguous work array. An implementation of implicitly padded convolutions is already publically available [Bowman & Roberts 2010].

Method	Complexity	Memory Footprint
Explicit Zero Padding	$6\,Km\log m$	$4m$
Implicit Zero Padding	$6\,Km\log m$	$4m$

Table 3.2: Comparison of methods for dealiasing one-dimensional non-centered complex binary convolutions of length m.

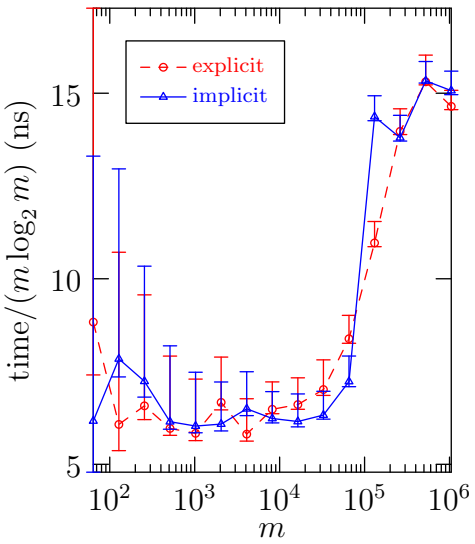

Figure 3.1: Comparison of computation times for explicitly and implicitly dealiased complex in-place 1D convolutions of length m.

Method	Complexity	Memory Footprint
Explicit Zero Padding without Pruning	$3 \cdot 2^d d\, K m^d \log m$	$2^{d+1}\, m^d$
Explicit Zero Padding with Pruning	$6\left(2^d - 1\right)\, K m^d \log m$	$2^{d+1}\, m^d$
Implicit Zero Padding	$6\left(2^d - 1\right)\, K m^d \log m$	$4\, m^d$

Table 3.3: Comparison of methods for dealiasing d-dimensional non-centered complex non-centered binary convolutions of size m^d.

Non-centered convolutions in d dimensions can be calculated in three ways; via explicit zero-padding without pruning, via explicit zero-padding with pruning, or via implicit zero-padding. As may be seen in Table 3.3, both types of explicit padding have greater memory requirements than that of implicit padding. Moreover, explicit zero-padding with pruning and implicit zero-padding have reduced computational complexity, making implicit padding clearly superior. Moreover, implicit padding is able to avoid a bit-reversal stage of the fast Fourier transform, which partly contributes to its superior empirical performance, as show in Figures 3.2 and 3.3.

Non-Centered n-ary Convolutions

Implicit padding being superior for binary convolutions, we now compare methods for implicitly padding n-ary non-centered convolutions. Implicit and explicit zero padding are equivalent in one dimension. In higher dimensions, pruning transforms provides a significant decrease in computational complexity of which both pruned explicit padding and implicit zero padding can take advantage. Moreover, implicit zero padding offers significantly decreased memory requirements, as shown in Table 3.4.

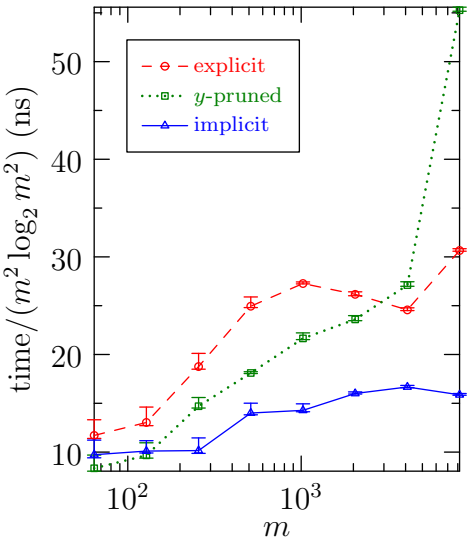

Figure 3.2: Comparison of computation times for explicitly and implicitly dealiased complex in-place 2D convolutions of size m^2.

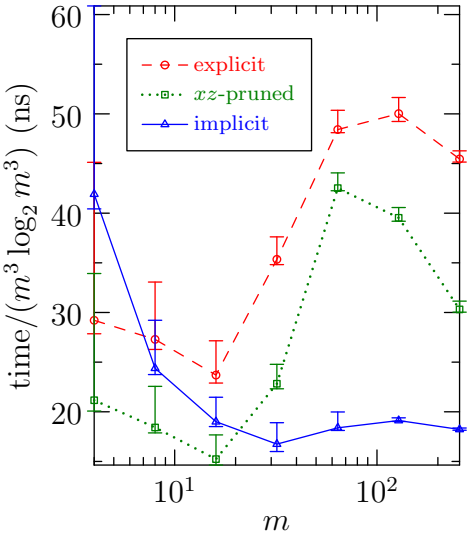

Figure 3.3: Comparison of computation times for explicitly and implicitly dealiased complex non-centered in-place 3D convolutions of size m^3.

Method	Complexity	Memory Footprint
Explicit padding	$(n+1)n^d d\, K m^d \log nm$	$n^{d+1}m^d$
Explicit padding with pruning	$n(n+1)\frac{n^d-1}{n-1}\, K m^d \log nm$	$n^{d+1}m^d$
Implicit padding	$n(n+1)\frac{n^d-1}{n-1}\, K m^d \log nm$	$n^2 m^d$

Table 3.4: Comparison of explicit and implicit methods for dealiasing d-dimensional n-ary non-centered complex convolutions on data of size m^d.

3.C.2 Centered Convolutions

Centered Binary Convolutions

There are three methods available for dealiasing centered convolutions in one dimension: phase-shift dealiasing, explicit zero-padding, and implicit zero-padding. As shown in Table 3.5, phase shift-dealiasing is both more computationally difficult and requires more memory than zero-padding. Both explicit and implicit zero-padding have identical characteristics to leading order, and, as may be seen in Figure 3.4, both implicit and explicit zero-padding perform calculations in roughly the same amount of time. Implicit zero-padding offers the one advantage that the work arrays need not be contiguous with the data arrays.

Centered binary convolutions in d dimensions may be dealiased in five different ways: via phase-shift dealiasing, via partial phase-shift dealiasing (for $d = 3$), via explicit zero-padding without pruning, via explicit zero-padding with pruning, and via implicit zero padding. As can be seen in Table 3.6, phase-shift dealiasing is very expensive in terms of both computational effort and memory use. However one may choose to dealias only some of the convo-

Method	Complexity	Memory Footprint
Phase-Shift Dealiasing	$6\,Km\log m$	$4m$
Explicit Zero Padding	$\frac{9}{2}\,Km\log m$	$3m$
Implicit Zero Padding	$\frac{9}{2}\,Km\log m$	$3m$

Table 3.5: Comparison of methods for dealiasing one-dimensional centered Hermitian convolutions of length m.

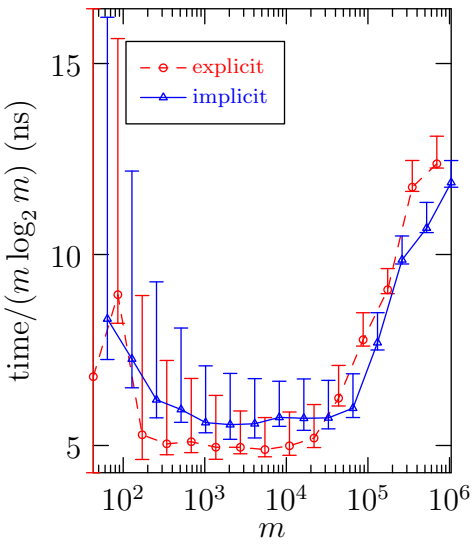

Figure 3.4: Comparison of computation times for explicitly and implicitly dealiased centered Hermitian in-place 1D convolutions of length m.

Method	Complexity	Memory Footprint
Phase-Shift Dealiasing	$3 \cdot 2^{2d-1} dK m^d \log m$	$2^{2d} m^d$
Partial Phase-Shift Dealiasing	$3 \cdot 2^d dK m^d \log m$	$2^{d+1} m^d$
Explicit Zero Padding without Pruning	$\frac{3^{d+1}}{2} d\, K m^d \log m$	$3^d m^d$
Explicit Zero Padding with Pruning	$\frac{9}{2}\left(3^d - 2^d\right)\, K m^d \log m$	$3^d m^d$
Implicit Zero Padding	$\frac{9}{2}\left(3^d - 2^d\right)\, K m^d \log m$	$3 \cdot 2^{d-1} m^d$

Table 3.6: Comparison of methods for dealiasing d-dimensional centered Hermitian convolutions of size $(2m-1)^{d-1} \times m$.

lution, with the rest taken care of by zero-padding. In three dimensions, this leaves all modes inside a sphere with radius $0.94m$, constituting approximately 44% of the original modes. This is particularly advantageous since the memory footprint is greatly reduced, though the computational expense is still significant. Explicit zero-padding offers only slightly higher memory and dealiases all of the data. However, it is quite computational expensive in the absence of pruning transforms on data that are known a priori to be zero, and pruning, while theoretically faster, is often not worthwhile in practice due to the lack of FFT routines specifically optimized for this purpose (cf: Figure 3.5). Implicit zero-padding is the clear choice for multi-dimensional centered binary convolutions, requiring less memory and less time per calculation. This result holds empirically, as shown in Figure 3.5.

Centered n-ary Convolutions

For multidimensional centered n-ary convolutions, pruning Fourier transforms provides a significant reduction in computational complexity for pruned ex-

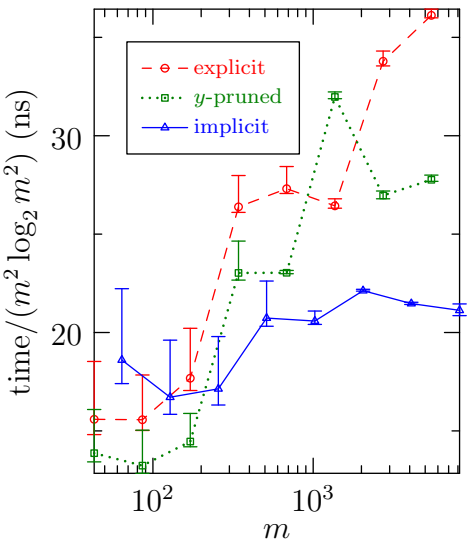

Figure 3.5: Comparison of computation times for explicitly and implicitly dealiased centered Hermitian in-place 2D convolutions of size $(2m-1) \times m$.

plicit zero-padding and implicit zero-padding as compared with un-pruned explicit zero-padding. Implicit zero-padding also greatly reduces the memory requirements for dealiasing such convolutions, as shown in Table 3.7.

Method	Complexity	Memory Footprint
Explicit padding	$\frac{1}{2}(n+1)^{d+1} d\, K m^d \log nm$	$\frac{n(n+1)^d}{2} m^d$
Explicit padding with pruning	$\frac{1}{2}(n+1)^2 \frac{(n+1)^d - 2^d}{n-1} K m^d \log nm$	$\frac{n(n+1)^d}{2} m^d$
Implicit padding	$\frac{1}{2}(n+1)^2 \frac{(n+1)^d - 2^d}{n-1} K m^d \log nm$	$n(n+1)2^{d-2} m^d$

Table 3.7: Comparison of methods for dealiasing d-dimensional n-ary centered Hermitian convolutions on data of size $(2m-1)^{d-1} \times m$.

3.D Parallel Computation of Implicitly Dealiased Convolutions

The fact that implicitly zero-padded Fourier transforms produce output that is non-contiguous can be used to develop an FFT-based convolution that can be parallelized. For example, to perform a two-dimensional convolution, one must first perform an inverse Fourier transform in the x direction, and then an inverse Fourier transform in the y direction. One then multiplies the outputs, takes a forward Fourier transform in the y direction, and finally takes a forward Fourier transform in the x direction. We demonstrate how this might be computed in parallel for a p/q-dealiased binary convolution using $N = \lambda q$ processors.

To set up the x transform, the input data must be sent to the sub-processors, for a communications cost of m^2 (or $\frac{N-1}{N}m^2$ if some data can stay on the master processor). For $i = 0, \ldots, N-1$, processor i then performs an implicitly padded $\lambda p/lq$ Fourier transform in the x direction on the contiguous memory $\texttt{block}_{i,j}$ for $j = 0, \ldots, \lambda p - 1$ for each input, as shown in Figure 3.6. Each x transform produces $\lambda q = N$ output blocks: processor i uses a work array consisting of memory $\texttt{block}_{i,j}$ for $j = \lambda p, \ldots, N-1$, with the output from the x transform stored in $\texttt{block}_{i,j}$, $j = 0, \ldots, N-1$, with $\texttt{block}_{i,j}$ containing x transformed data with x index equal to $j \bmod N$ and y wavenumber in $\left(\frac{mi}{N}, \frac{m(i+1)}{N}\right)$.

We must move the data in $\texttt{block}_{i,j}$ for $i = 0, \ldots, N-1$, $j = 0, \ldots, N-1$, $i \neq j$. To perform the y transform. Each block has size

$$\frac{m}{N} \times m\frac{q}{p}\frac{1}{N} = m^2\frac{q}{p}\frac{1}{N^2}. \tag{3.52}$$

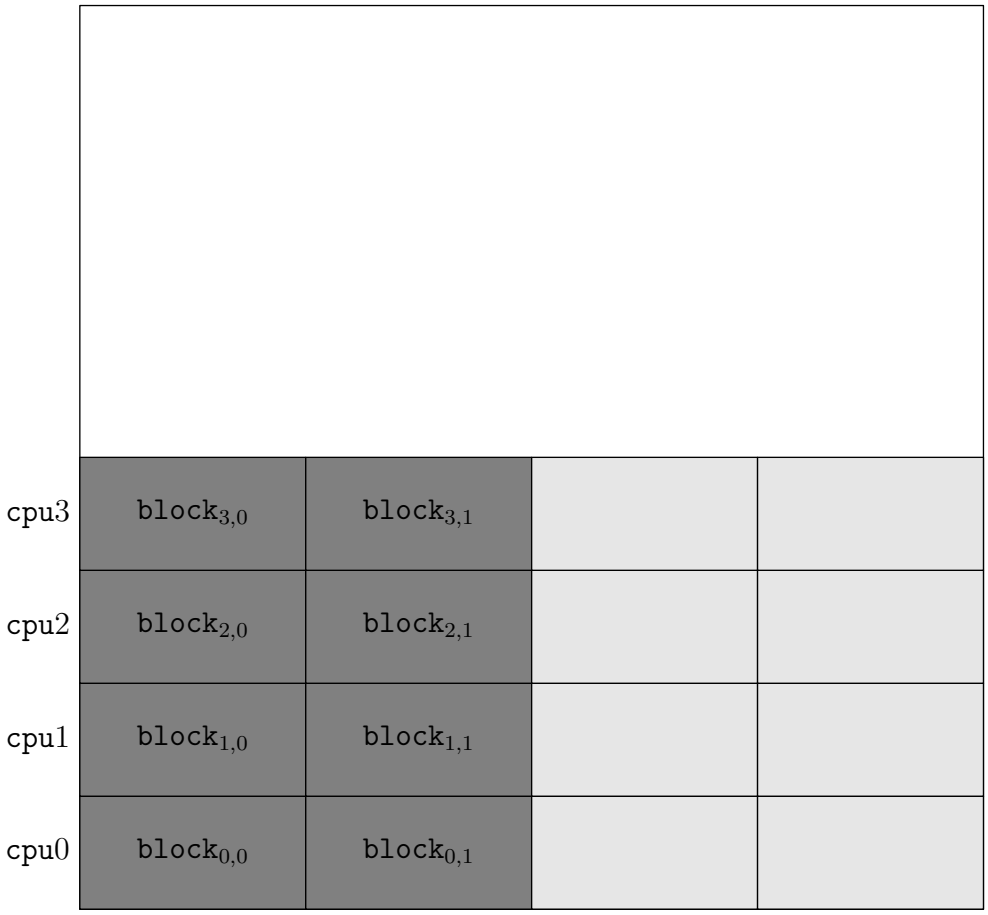

Figure 3.6: Demonstration of the x transform involved in a parallel, transpose-free FFT-based two-dimensional convolution of implicitly zero-padded data. Dark gray indicates input data, with output captured in the both light and dark gray blocks.

All but N of the N^2 block must be moved to another processor for the y transform, which is to say that, for each input array, data of size

$$m^2 \frac{q}{p} \frac{N^2 - N}{N^2} \tag{3.53}$$

must be communicated in order to perform an implicitly padded Fourier transform in the y direction by processor j on input arrays $\texttt{block}_{i,j}$ for $i = \lambda p, \ldots, N-1$, producing output in arrays $\texttt{block}_{i,j}$ for $i = \lambda p, \ldots, \frac{q}{p}N - 1$, as shown in Figure 3.7. The data in $\texttt{block}_{i,j}$ is fully in physical space, with x coordinate equal to $j \bmod N$ and y coordinate equal to $i \bmod N$.

The output from the inverse Fourier transforms of input arrays f and g performed by processor j are multiplied by processor j. One then inverts the y and x direction implicitly padded Fourier transforms, incurring an additional communication cost of

$$m^2 \frac{q}{p} \frac{N^2 - N}{N^2} + m^2, \tag{3.54}$$

for a total communication cost of

$$3\left(m^2 \frac{q}{p} \frac{N^2 - N}{N^2} + m^2 \right). \tag{3.55}$$

This method has computational complexity and memory requirements equal to serial implicitly padded FFT-based convolution, but that problem is split between N processors, allowing the calculation to be performed in less time. The situation is complicated for Hermitian-symmetric data, where the implicitly padded FFT (Function conv) stores produces overlapping output. This minor inconvenience can be dealt with by rearranging data during the communication step. An implementation of this algorithm would be of great benefit, allowing one to perform implicitly dealiased pseudospectral simulations on distributed-memory architectures.

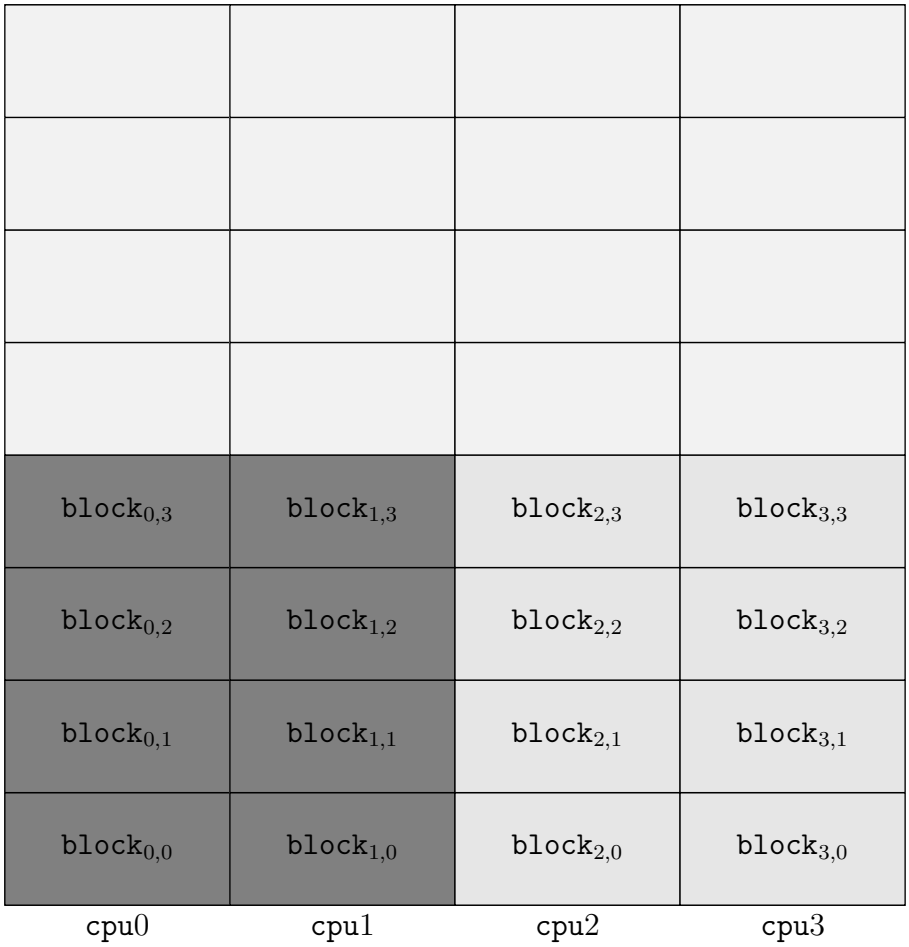

Figure 3.7: Demonstration of the y transform involved in a parallel, FFT-based two-dimensional convolution of implicitly 1/2-padded zero-padded data. Dark- and medium- gray blocks represent input data to the y transform, with output stored in all blocks in the Figure.

Chapter 4

Spectral Reduction

Wherein a method for uniformly decimating spectral simulations is described, which is applied both to shell models of turbulence, including higher-order approximations, and to the two-dimensional Navier–Stokes equations.

Turbulent systems exhibit many degrees of freedom, which is a major obstacle when performing simulations of such systems. However, it is suspected that the information content of high Reynolds-number Navier–Stokes turbulence may not be as severe as per Kolmogorov's prediction that the number of active modes N grows as $N \sim \mathrm{Re}^{9/4}$ [Landau & Lifshitz 1959], as some modes may be quiescent (for example, [Paladin & Vulpiani 1987], [Farge 1992]) and there is the possibility of performing useful simulations of such systems using a restricted basis.

Moreover, we may not always be interested in the incredible detail contained in all the degrees of freedom from turbulent systems, and could be very happy indeed with a solution providing only the average motion of the fluid

in question. This problem has been approached by taking statistical closures of moments of turbulence [Kraichnan 1958], [Kraichnan 1961], [Martin *et al.* 1973], [Orszag 1977], [Holloway & Hendershott 1977], [Bowman *et al.* 1989], or by considering reduced systems [Lorenz 1972], [Kraichnan 1985], [Williams *et al.* 1987], [Kraichnan & Chen 1989], [Lee 1989], [Vázquez-Semadeni & Scalo 1992], [She & Jackson 1993], [Grossmann *et al.* 1996], [Holmes *et al.* 1996].

The method of spectral reduction [Bowman *et al.* 1999] allows one to reduce the dimensionality of a system by evolving "bins" of Fourier modes instead of the individual modes themselves. This coarse-graining drastically reduces the computational cost of simulations without modifying the flow qualitatively. Each bin Ω_K represents the modes $\{w_k\}_{k\in S_K}$ for some set S_K of modes from the original system. The evolution equation for Ω_K is determined by averaging the source term for the modes. The method is closed by approximating the modes $\{\omega_k\}_{k\in S_K}$ by some function of Ω_K.

This technique reproduces the expected equipartition spectrum for inviscid, unforced dynamics when the sets S_K are all the same size. However, when the number of modes in S_K varies with K, the system forgets that the bins were originally composed of modes, and thus evolves to create an equipartition not of modal energies but of bin energies. While this can be corrected by rescaling time by the bin size, the resulting system is too numerically stiff to be of much use [Bowman *et al.* 2001].

4.A Spectral Reduction of Shell Models of Turbulence

Shell models are not an immediately obvious candidate for spectral reduction, as they do not exhibit a large number of degrees of freedom, so a reduced model is not greatly in demand in the first place. Our goal in applying spectral reduction to shell models of turbulence is not to make these models yet more amenable to numerical methods; rather, we use them to clarify and test ideas and techniques before applying them to more complicated systems.

Shell models of turbulence represent the fluid velocity by shells of modes characterized by the complex amplitude u_n. These shells are geometrically spaced with characteristic wavenumber $k_n = k_0\lambda^n$. It is convenient to first consider a radix-two spectral reduction of shell models, which reduces the number of evolved quantities by half. In this case spectral reduction represents modes u_{2n} and u_{2n+1} by the single quantity

$$u_n^{(1)} = \frac{u_{2n} + \sigma_n^{(0)*} u_{2n+1}}{1 + \left|\sigma_n^{(0)}\right|^2}, \tag{4.1}$$

where $\sigma_n^{(0)} \doteq u_{2n+1}/u_{2n}$ is the ratio between adjacent modes. The original modes can be recovered via

$$u_{2n} = u_n^{(1)}, \qquad u_{2n+1} = \sigma_n^{(0)} u_n^{(1)}.$$

The evolution equation for $u_n^{(1)}$ is derived from the evolution equations of u_{2n} and u_{2n+1}. If we were to start with the GOY model, where

$$\frac{du_n}{dt} = ik_n\left(\alpha u_{n+1}u_{n+2} + \frac{\beta}{\lambda}u_{n-1}u_{n+1} + \frac{\gamma}{\lambda^2}u_{n-1}u_{n-2}\right)^* - \nu k_n^2 u_n + F_n, \tag{4.2}$$

then the evolution equation for $u_n^{(1)}$, assuming that $\sigma_n^{(0)}$ is constant with respect to time, is

$$\begin{aligned}\frac{du_n^{(1)}}{dt} = &-\nu k_{2n}^2 \frac{1+\sigma_n^{(0)*}\lambda^2}{1+|\sigma_n^{(0)}|^2} u_n^{(1)} + \frac{F_{2n}+\sigma_n^{(0)*}F_{2n+1}}{1+|\sigma_n^{(0)}|^2} \\ &+\frac{1}{1+|\sigma_n^{(0)}|^2} ik_{2n}\Bigg[\sigma_{n-1}^{(0)}\frac{\gamma}{\lambda^2}u_{n-1}^{(1)2} + \sigma_n^{(0)}(\alpha+\beta)u_n^{(1)}u_{n+1}^{(1)} \\ &+\sigma_{n-1}^{(0)}\sigma_n^{(0)}\left(\frac{\beta}{\lambda}+\frac{\gamma}{\lambda}\right)u_{n-1}^{(1)}u_n^{(1)} + \lambda\alpha\sigma_n^{(0)}\sigma_{n+1}^{(0)}u_{n+1}^{(1)2}\Bigg]^*. \end{aligned} \tag{4.3}$$

The coarse-grained energy $E = \frac{1}{2}\sum_n |u_n^{(1)}|^2(1+|\sigma_n^{(0)}|^2)$ is conserved in the absence of forcing and viscosity if the original model conserved energy and $\sigma_n^{(0)}$ is constant with respect to time. Other quadratic invariants (i.e. helicity or enstrophy) are lost. Since the original modes had next-nearest-neighbour interactions, the resulting system has nearest-neighbour interactions.

If $\alpha+\beta+\gamma = 0$, then the underlying GOY model conserves energy, and equation (4.3) is the DN model with parameters given by

$$\begin{aligned} &u_n^{(1)} = \frac{u_{2n}+\sigma_n^{(0)*}u_{2n+1}}{1+|\sigma_n^{(0)}|^2},\ a_n^{(1)} = \frac{\gamma}{\lambda^2}\left(\frac{\sigma_{n-1}^{(0)}}{1+|\sigma_n^{(0)}|^2}\right),\ b_n^{(1)} = \frac{-\alpha}{\lambda}\left(\frac{\sigma_{n-1}^{(0)}\sigma_n^{(0)}}{1+|\sigma_n^{(0)}|^2}\right), \\ &\lambda^{(1)} = \lambda^2, \quad \nu_n^{(1)} = \nu\frac{1+|\sigma_n^{(0)}|^2\lambda^2}{1+|\sigma_n^{(0)}|^2}, \quad \text{and} \quad F_n^{(1)} = \frac{F_{2n}+\sigma_n^{(0)*}F_{2n+1}}{1+|\sigma_n^{(0)}|^2}. \end{aligned} \tag{4.4}$$

Further applications of spectra reduction map the DN model to the DN model, which is a fixed point of spectral reduction [Eckhardt 2004].

The spectrally reduced DN model uses the same binned mode as in equation (4.1), but the evolution equation is found by binning the source from the DN equation. Following the same procedure as with the GOY model, we can also start with the evolution equation for the DN model,

$$\frac{du_n}{dt} = ik_n\big[a\big(u_{n-1}^2 - \lambda u_n u_{n+1}\big) + b\big(u_{n-1}u_n - \lambda u_{n+1}^2\big)\big]^* - \nu k_n^2 u_n + F_n, \tag{4.5}$$

which, when binned according to equation (4.1), yields for constant $\sigma_n \doteq \sigma_n^{(0)}$ the evolution equation for $u_n^{(1)}$:

$$\begin{aligned}\frac{du_n^{(1)}}{dt} &= \frac{1}{1+|\sigma_n|^2}\left(\frac{du_{2n}}{dt} + \sigma_n^* \frac{du_{2n+1}}{dt}\right) \\ &= \frac{ik_{2n}}{1+|\sigma_n|^2}\Big(a_{2n}\sigma_{n-1}^2 {u_{n-1}^{(1)}}^2 - a_{2n+2}\lambda^2\sigma_n^2 u_n^{(1)} u_{n+1}^{(1)} \\ &\qquad + b_{2n}\sigma_{n-1} u_{n-1}^{(1)} u_n^{(1)} - b_{2n+2}\lambda^2\sigma_n {u_{n+1}^{(1)}}^2\Big)^* \\ &\qquad + \frac{F_{2n} + \sigma_n^* F_{2n+1}}{1+|\sigma_n|^2} - \nu k_{2n}^2 u_1^{(n)} \frac{1+\lambda^2|\sigma_n|^2}{1+|\sigma_n|^2}.\end{aligned} \tag{4.6}$$

Repeating spectral reduction ℓ times maps the nonlinear coefficients to

$$a_n^{(\ell)} = a_{2^\ell n}^{(0)} \prod_{i=0}^{\ell-1} \left(\sigma_{2^i n}^{(i)}\right)^2, \qquad b_n^{(\ell)} = b_{2^\ell n}^{(0)} \prod_{i=0}^{\ell-1} \sigma_{2^i n}^{(i)} \tag{4.7}$$

and we also divide the nonlinear term by $\prod_{i=0}^{\ell-1}\left(1+\left|\sigma_{2^i n}^{(i)}\right|^2\right)$ to account for binning. The dissipative coefficient changes as

$$\nu_n^{(\ell)} = \nu \prod_{i=0}^{\ell-1} \frac{1+{\lambda^{(i)}}^2\left|\sigma_{2^i n}^{(i)}\right|^2}{\left(1+\left|\sigma_{2^i n}^{(i)}\right|^2\right)}. \tag{4.8}$$

Since the nonlinear term in the DN model is nearest-neighbour, spectral reduction is also nearest-neighbour.

Both the DN and GOY shell models spectrally reduce to the DN model, with nonlinear interaction coefficients, forcing, and viscosity. The resulting model is amenable to simulation, and exhibits fewer degrees of freedom than the original system.

The energy spectrum of spectrally reduced shell models is given by

$$E_n^{(\ell)} = \delta_n^{(\ell)} \left|u_n^{(\ell)}\right|^2 = \prod_{i=0}^{\ell-1} \left(1 + \left|\sigma_{2^i n}^{(i)}\right|^2\right) \left|u_n^{(\ell)}\right|^2 \tag{4.9}$$

and the total energy is $E_\ell = \sum E_n^{(\ell)}$. We modify the white-noise banded forcing so that the energy injection rate remains invariant under binning. That is, if we force just mode one, we require that

$$\begin{aligned} \frac{d}{dt}E^{(\ell)} &= \frac{d}{dt}E_1^{(\ell)} \\ &= \frac{1}{2}\frac{d}{dt}\prod_{i=0}^{\ell-1}\left(1 + \left|\sigma_{2^i}^{(i)}\right|^2\right)\left|u_1^{(\ell)}\right|^2 \\ &= \frac{1}{2}\prod_{i=0}^{\ell-1}\left(1 + \left|\sigma_{2^i}^{(i)}\right|^2\right)\left\langle |f|^2 \right\rangle \\ &= \epsilon, \end{aligned} \tag{4.10}$$

where ϵ is the energy injection rate. Thus the mean forcing amplitude must be given by

$$f = \sqrt{\frac{2\epsilon}{\prod_{i=0}^{\ell-1}\left(1 + \left|\sigma_{2^i}^{(i)}\right|^2\right)}}. \tag{4.11}$$

As in the once-reduced case, exact knowledge of the values of σ_n and $u^{(\ell)}$ implies exact knowledge of the unreduced velocity modes. Since the goal of spectral reduction is to represent the system using a reduced basis, we use an approximation to determine σ_n to close the system.

4.A.1 Binned Spectral Reduction

The simplest approximation is to assume that the velocity modes are constant over a bin. That is, $\sigma_n = 1$ for all n. In this case, the reduced mode is just

the mean of the unreduced velocities:

$$u_n^{(1)} = \frac{u_{2n} + u_{2n+1}}{2}. \tag{4.12}$$

The mapping of nonlinear coefficients is also simplified. In the case of the GOY model being reduced to the DN model,

$$a = \frac{1}{2}\frac{\gamma}{\lambda^2}, \qquad b = \frac{1}{2}\frac{-\alpha}{\lambda},$$

and the DN model just reduces the nonlinear interaction coefficients by half:

$$a^{(1)} = \frac{1}{2}a^{(0)}, \qquad b^{(1)} = \frac{1}{2}b^{(0)}.$$

The viscosity in both cases is mapped to

$$\nu^{(1)} = \nu^{(0)}\frac{1+\lambda^2}{2}, \tag{4.13}$$

which compensates for the fact that part of the bin reaches higher wavenumbers where the Laplacian has more effect. As can be seen in Figures 4.1 and 4.2, treating σ_n as constant over a bin is quite effective, producing reasonable results even when the system is decimated repeatedly.

4.A.2 Interpolated Spectral Reduction

The shell amplitudes u_n typically obey a power-law relationship. One can use this to approximate the unresolved variables $\sigma_n = u_{2n+1}/u_{2n}$ by taking a geometric mean of the reduced modes via

$$\sigma_n \approx \sqrt{\frac{u_{2n+2}}{u_{2n}}} = \sqrt{\frac{u_{n+1}^{(1)}}{u_n^{(1)}}}.$$

This can be applied recursively for repeated reductions, giving

$$\sigma_n^{(\ell)} = \sqrt{\sigma_{2n}^{(\ell-1)}}, \tag{4.14}$$

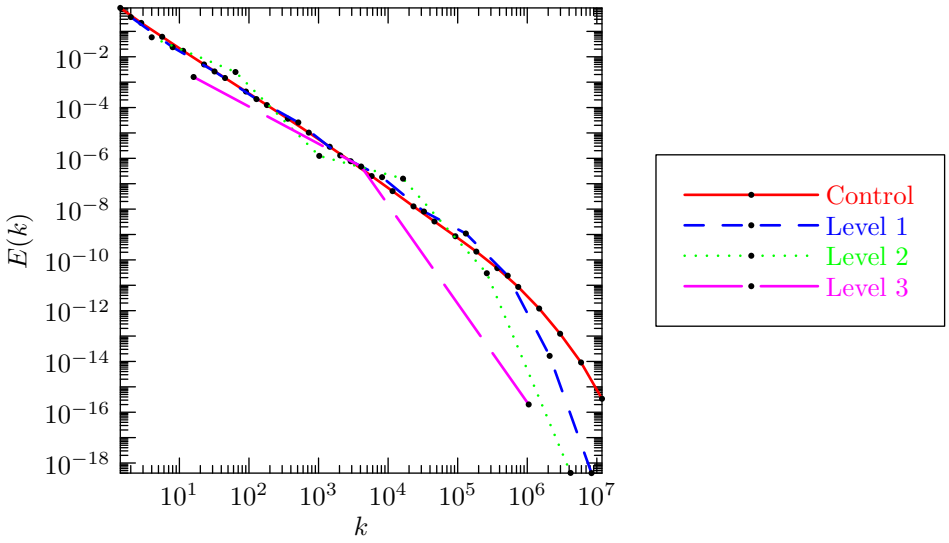

Figure 4.1: Energy spectra from the spectrally reduced DN model with constant σ_n, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$.

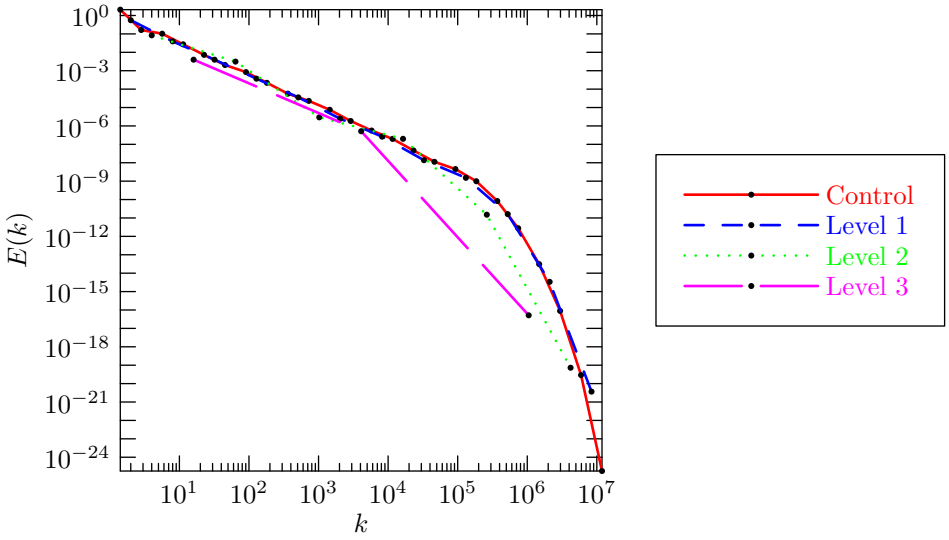

Figure 4.2: Energy spectra from the spectrally reduced GOY model with constant σ_n, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$.

under the assumption that $\sigma^{(\ell-1)}$ is constant across the ℓ-level bin $u_n^{(\ell)}$.

Since the evolved modes are complex-valued, taking square-roots is problematic due to branch-cut choices. We therefore treat σ_n as real-valued, and determine it via the formula

$$\sigma_n = \sqrt{\frac{\left|u_{n+1}^{(1)}\right|}{\left|u_n^{(1)}\right|}}. \tag{4.15}$$

We are most interested in statistically stationary states. To increase numerical stability, we replace the instantaneous values in equation (4.15) by time-averaged quantities. These can either be the time-mean of the evolved modes from initial-conditions on, or a periodically reset quantity, i.e. we take a windowed average over either $[0, t]$ or over more recent values in order to more closely represent the current state of the simulation.

Energy conservation requires that σ_n be constant over a single time step, but one is free to change σ_n between time steps to more closely reflect current statistics. Since σ_n appears explicitly in the definition of energy (equation (4.9)), one must be careful that changing σ_n does not modify the total energy of the reduced system. That is, if σ_n changes while the velocity is constant, then, energy is conserved if

$$\sum \left(1 + |\sigma_n(t_1)|^2\right) |u_n(t_1)|^2 = \sum \left(1 + |\sigma_n(t_2)|^2\right) |u_n(t_2)|^2 . \tag{4.16}$$

It is necessary to rescale the velocities from time t_1 to time t_2 in order to conserve energy if σ_n changes from time t_1 to time t_2. This can be accomplished in a straightforward fashion by requiring that equation (4.16) be satisfied for each mode.

4.A.3 Simulations

Interpolation Based on Time Average

The spectra of interpolated and uninterpolated simulations are fairly similar. This is likely due to the fact that the value of σ_n is stabilized somewhat since we calculate σ_n via

$$\sigma = \left(\int_0^t \left| u_{n+1}^{(1)} \right| \, dt \Big/ \int_0^t \left| u_n^{(1)} \right| \, dt \right)^{\frac{1}{2}}, \tag{4.17}$$

where the long-time average decreases the variation of σ_n. If the initial conditions are such that u_{n+1} is not significantly smaller than u_n at early times, then the time-averaged quantities $\langle |u_n| \rangle$ and $\langle |u_{n+1}| \rangle$ will change in ratio much slower than the ratio of the unaveraged values of $|u_n|^2$ and $|u_{n+1}|^2$. This will stabilize the scheme by maintaining σ_n near its value at early times. The energy spectra of a full-resolution simulation compared with the spectra of reduced simulations with and without interpolation is shown in Figure 4.3. Further decimation produces an instability that increases the energy at the high-wavenumber end of the inertial range while basically eliminating the energy in the dissipation range. For typical simulations (i.e. with $\lambda = 2$), this instability is present after two spectral reductions (cf: Figure 4.4).

Interpolation Based on Running Time-Average

The interpolation factor σ_n as given in equation (4.17) can be dominated by initial conditions. As such, the value of σ_n may not match that of the current properties of the flow. If we instead use the running-time average

$$\sigma = \left(\int_{t-\Delta}^t \left| u_{n+1}^{(1)} \right| \, dt \Big/ \int_{t-\Delta}^t \left| u_n^{(1)} \right| \, dt \right)^{\frac{1}{2}} \tag{4.18}$$

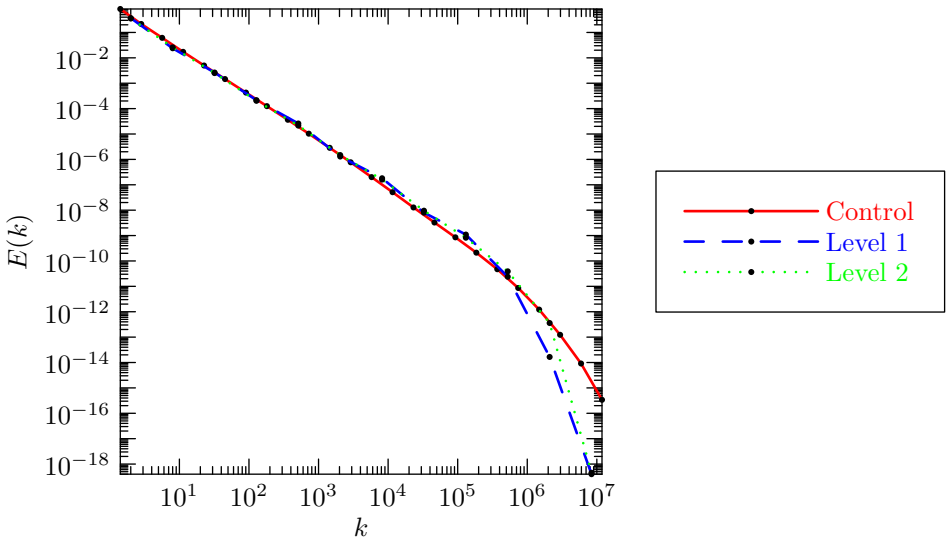

Figure 4.3: Rescaled DN energy spectra comparing a control run with once-decimated runs using $\sigma = 1$ and with interpolation. The simulations used $a = -1/4$, $b = -1$, and $\nu = 10^{-7}$.

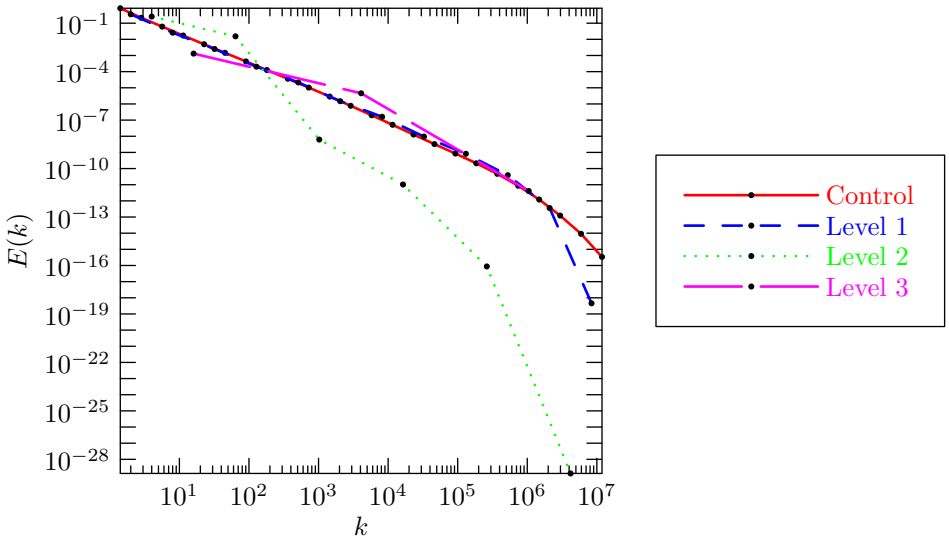

Figure 4.4: Rescaled DN energy spectra comparing a control run with decimated runs using with interpolation. The simulations used $a = -1/4$, $b = -1$, and $\nu = 10^{-7}$.

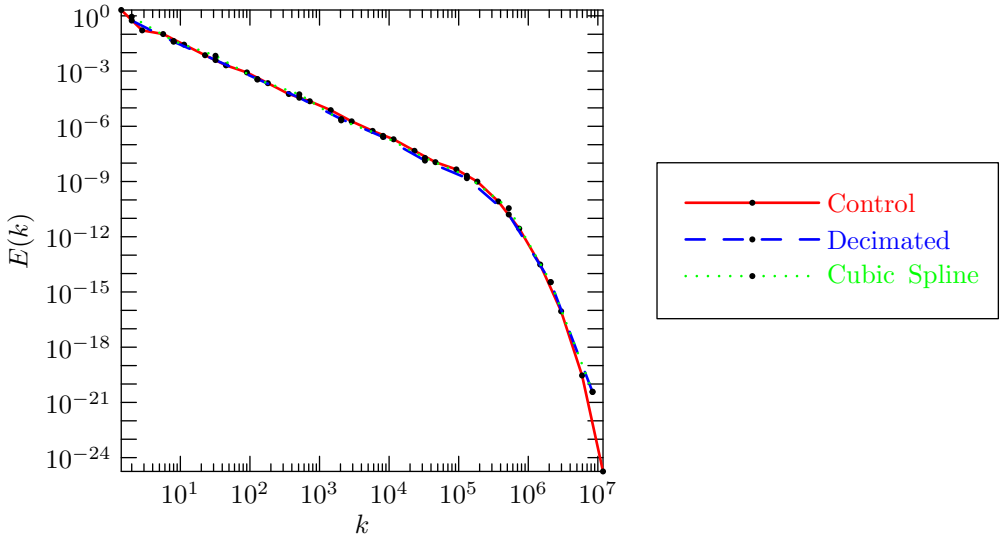

Figure 4.5: Rescaled GOY energy spectra comparing a control run with once-decimated runs using $\sigma = 1$ and with interpolation. The simulations used $a = -1/4$, $b = -1$, and $\nu = 10^{-7}$.

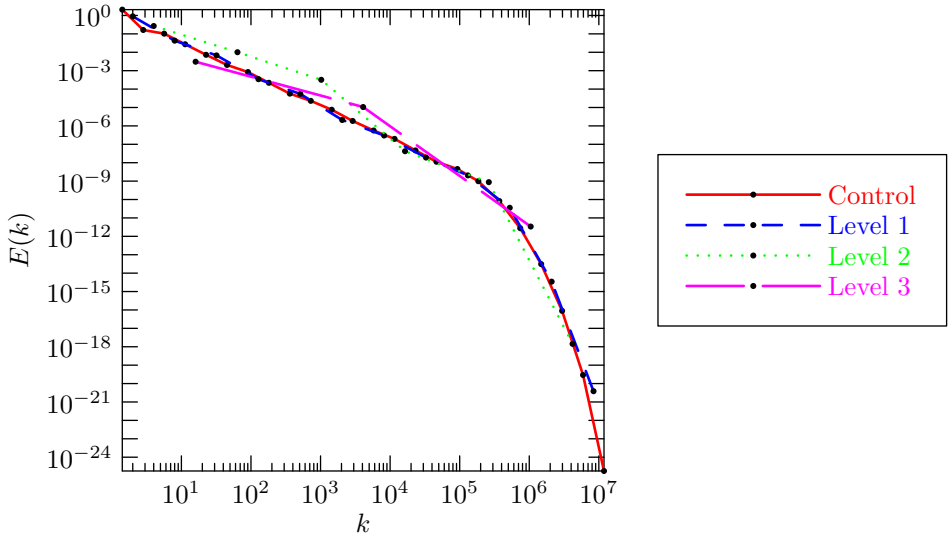

Figure 4.6: Reset DN energy spectra for interpolated decimation with $\sigma \in (0.52, 3.09)$ for three levels of decimation, with $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$.

over a moving window $[t-\Delta, t]$, then σ_n will relax to the value of the current statistically stationary state much more quickly, and we hope to achieve better accuracy using this definition. Unfortunately, these more accurate values for σ_n produce an instability for even a single spectral reduction, as shown in Figure 4.7.

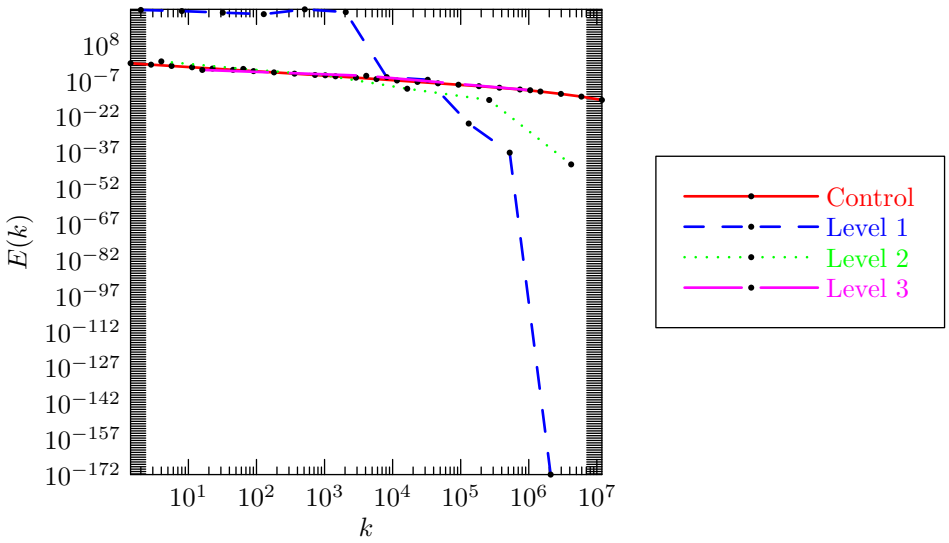

Figure 4.7: Rescaled and reset DN energy spectra of comparing a control run with once-decimated runs using $\sigma = 1$, with interpolation, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$.

Since the problem with this method is an instability in the value of σ_n, one approach to stabilize the technique would be to restrict σ_n to more moderate values. Given bounding values $\sigma_{\min}$ and $\sigma_{\max}$, we seek a monotonic map that sends $1 \to 1$, $0 \to \sigma_{\min}$, and $\infty \to \sigma_{\max}$. This is provided by the *bilinear map* (or *fractional linear map*),

$$\sigma \to \frac{(1-\sigma_{\min})\sigma_{\max}\sigma + (\sigma_{\max}-1)\sigma_{\min}}{(1-\sigma_{\min})\sigma + (\sigma_{\max}-1)}. \tag{4.19}$$

The bounds $\sigma_{\min}$ and $\sigma_{\max}$ are chosen a posteriori, based on a full-resolution simulation. We determine $(\sigma_{\min}, \sigma_{\max})$ by performing a full-resolution and

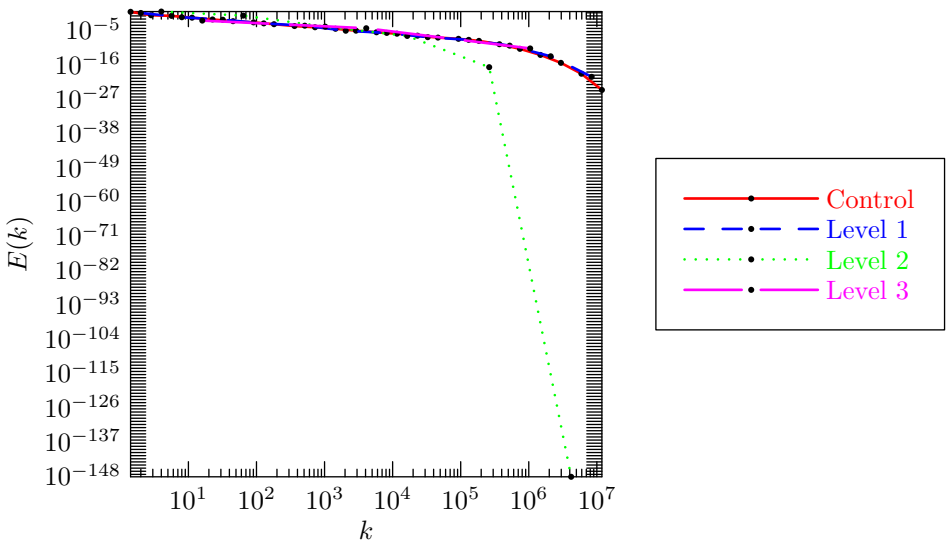

Figure 4.8: Rescaled and reset GOY energy spectra of comparing a control run with once-decimated runs using $\sigma = 1$, with interpolation, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$

looking at the maximal and minimal ratios of bin energies when the system is in a statistically steady state. Taking these values from the inertial range, we have $\sigma \in (0.52, 3.09)$. Taking the bounds from the entire spectrum yields $\sigma \in (0.0026, 3.09)$. The inertial-range bounds prevent the instability from being expressed, but do not offer a significantly better result than those produced without interpolation. The spectrum is improved with using less restrictive bounds for the first decimation. Unfortunately, the instability reappears upon successive decimations, whether one uses bounds from just the entire spectrum (Figure 4.9) or just the inertial range (Figure 4.10).

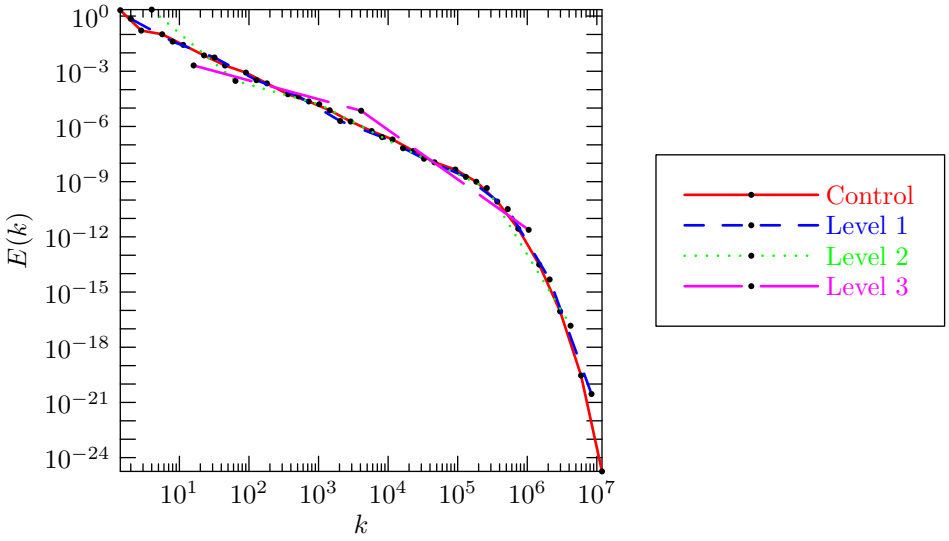

Figure 4.9: Energy spectra for the interpolated DN model with $\sigma \in (0.0026, 3.09)$, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$.

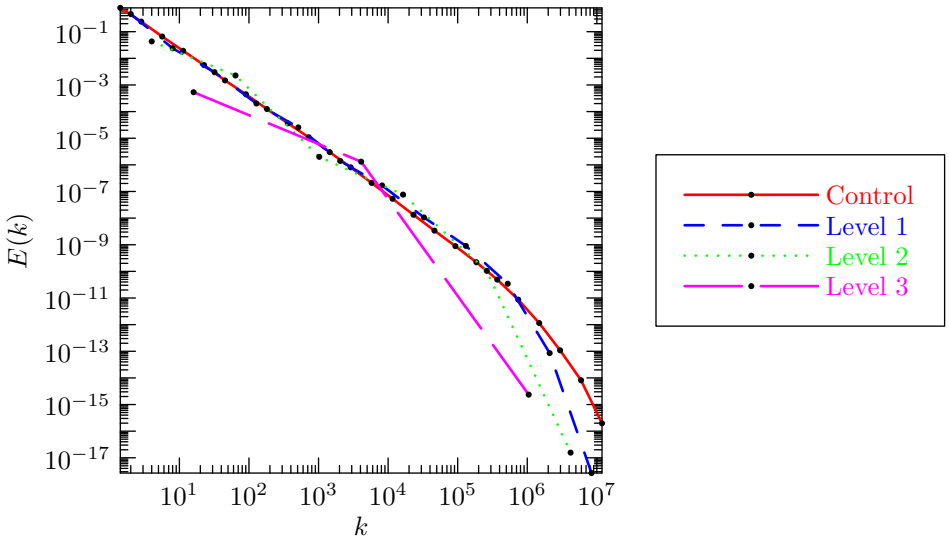

Figure 4.10: Reset DN energy spectra for interpolated decimation with $\sigma \in (0.52, 3.09)$ for three levels of decimation.

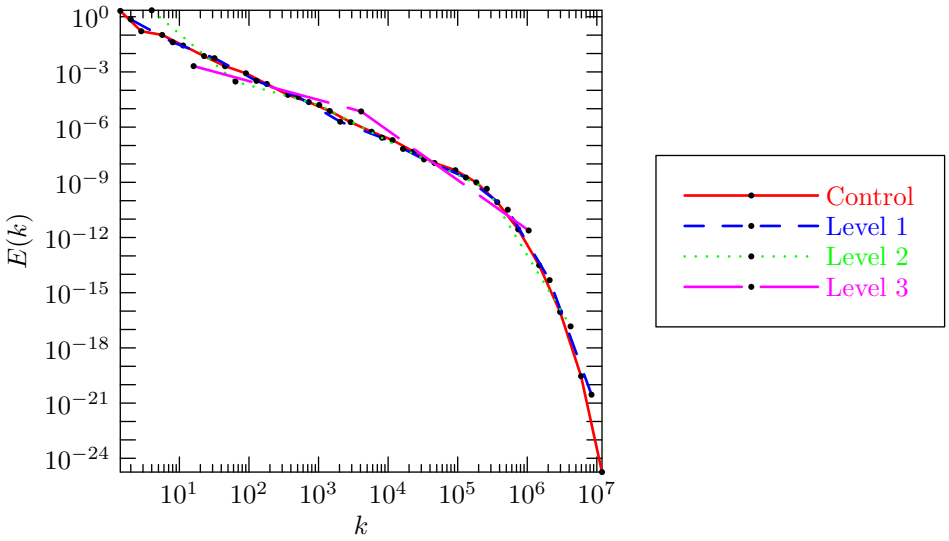

Figure 4.11: Reset GOY energy spectra for interpolated decimation with $\sigma \in (0.0026, 3.09)$ for three levels of decimation, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$.

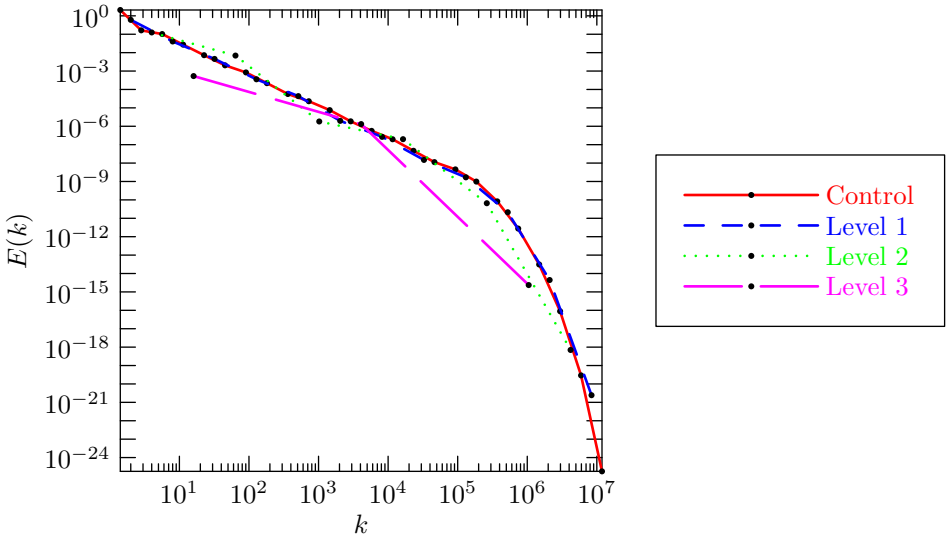

Figure 4.12: Reset GOY energy spectra for interpolated decimation with $\sigma \in (0.52, 3.09)$ for three levels of decimation, $\alpha = 1$, $\beta = -1/4$, and $\nu = 10^{-7}$.

4.A.4 Heuristic explanation of instabilities arising from interpolation

Highly decimated runs tend to evolve to have artificially high spectra at the high-wavenumber end of the inertial range, i.e. at wavenumbers just lower than the dissipation range. Such behaviour suggests an instability in the numerical scheme.

Instabilities in DN-derived interpolated simulations

Interpolated spectrally reduced simulations derived from the DN model have a nonlinear source term, denoted $\frac{du_n}{dt}\big|_{\mathrm{NL}}$, which depends on σ_{n-1} and σ_n. When σ_{n-1} becomes small, this nonlinear source term decreases. Since it is the nonlinear source term that provides mode n with energy, decreasing the nonlinear source in the inertial range will decrease $|u_n|^2$. This will in turn decrease $\sigma_{n-1} = \left\langle |u_n|^2 / |u_{n-1}|^2 \right\rangle^{1/4}$, completing the cycle. This is shown schematically in Figure 4.13. If $|u_n|^2$ decreases, then it will continue to decrease in a positive-feedback loop until it is u_n is very small.

Comparison of instabilities in DN- and GOY-derived simulations.

Spectral reduction maps both the GOY and DN models to a reduced DN model. As can be seen in the above sections, $\mathrm{DN}\underset{\mathrm{SR}}{\rightarrow}\mathrm{DN}$ simulations experience instability, whereas $\mathrm{GOY}\underset{\mathrm{SR}}{\rightarrow}\mathrm{DN}$ are much more stable. Of course, these instabilities are only present under interpolation, but it is strange that such similar models exhibit such different behaviour.

The explanation must be found in the mapping of the nonlinear interaction coefficients for the resulting model. The GOY model has nonlinear interac-

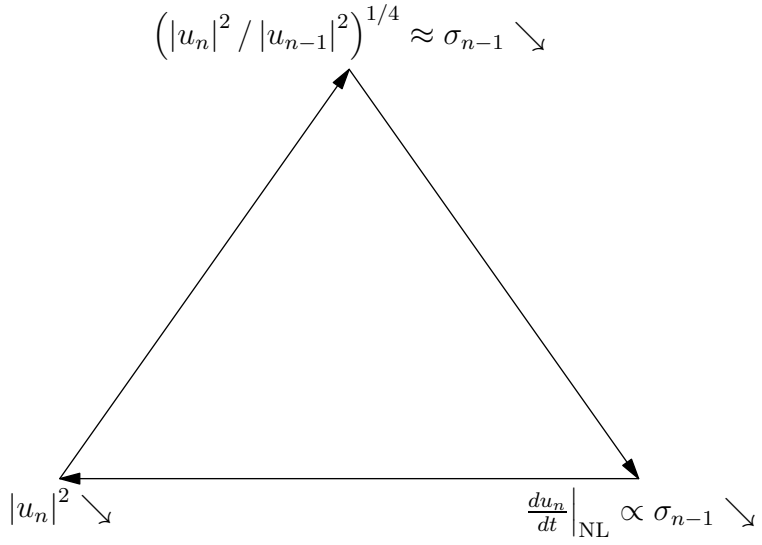

Figure 4.13: Schematic diagram of the instability in DN→DN interpolated simulations.

tion parameters (α, β, γ), with energy conserved if $\alpha + \beta + \gamma = 0$. Denoting $a_0 \doteq \gamma/\lambda^2$ and $b_0 = -\alpha/\lambda$, spectral reduction of the GOY model maps the parameters as

$$a_0 \to a_n = a_0 \sigma_{n-1} \tag{4.20}$$

$$b_0 \to b_n = b_0 \sigma_{n-1} \sigma_n. \tag{4.21}$$

Spectral reduction of the DN model maps the parameters as

$$a_0 \to a_n = a_0 \sigma_{n-1}^2 \tag{4.22}$$

$$b_0 \to b_n = b_0 \sigma_{n-1}. \tag{4.23}$$

These mappings are symmetric in $a \leftrightarrow b$ except for the change of one index between equations (4.21) and (4.22).

In order to determine why DN$\underset{\text{SR}}{\rightarrow}$DN is even less stable than GOY$\underset{\text{SR}}{\rightarrow}$DN, consider the fact that, to first order in σ_n, DN$\underset{\text{SR}}{\rightarrow}$DN has coefficients (a, b) proportional to $(0, 1)$, and GOY$\underset{\text{SR}}{\rightarrow}$DN has coefficients proportional to $(1, 0)$. Numerical simulations show that the choice of nonlinear coupling coefficients $(a, b) = (1, 0)$ has only a direct cascade, i.e. energy is only sent to higher wavenumbers, while $(a, b) = (0, 1)$ has both a forward and an inverse cascade. Thus, energy in simulations based on GOY$\underset{\text{SR}}{\rightarrow}$DN doesn't have the opportunity to reach equipartition at low wavenumbers because the nonlinear term drives it to smaller scales more exclusively. On the other hand, the presence of an inverse cascade in DN$\underset{\text{SR}}{\rightarrow}$DN-derived simulations is a mechanism for energy at lower wavenumbers to be transported away from the instability, which then further drops σ_n, and the instability is perpetuated.

Spectral reduction of shell models of turbulence reproduces the behaviour of the original system when the modes are approximated as constant over a bin. The introduction of interpolation factors, which account for modes having different magnitudes over a bin, exhibits an instability that prevents interpolated spectral reduction from being a useful technique for performing simulations of shell models of turbulence.

The experience gained from working with shell models of turbulence guides us in developing spectral reduction of two-dimensional turbulence. Since interpolated spectral reduction proved unstable, we restrict our focus to approximating modes as constant over a bin.

4.B Spectral Reduction of Navier–Stokes Turbulence

The simulations of n-dimensional systems using spectral techniques are particularly straightforward to implement when the physical domain can be reasonably considered as periodic, i.e. $\boldsymbol{x} \in 2\pi x_0 \mathbb{R}^n/\mathbb{Z}^n$ where n is the dimension and $2\pi x_0$ is the length of the box. This is the case when, for example, one is interested in turbulence far way from boundaries, and is a reasonable approximation provided that the correlation length of the fluid is smaller than the linear dimensions of the periodic domain.

The motion of the fluid may be expressed as a superposition of a countably infinite number of discrete modes, $\{\boldsymbol{u_k}, \boldsymbol{k} = k_0\boldsymbol{p}, \boldsymbol{p} \in \mathbb{Z}^n\}$ where k_0 is the lowest wavenumber. Viscosity effectively eliminates energy in all modes $\boldsymbol{u_k}$ with $k > k_d$, so we truncate the system at high wavenumbers, and consider a solution in $\{\boldsymbol{u_k}, \boldsymbol{k} = k_0\boldsymbol{p}, \boldsymbol{p} \in (-\lceil k_d/k_0 \rceil, \ldots, \lceil k_d/k_0 \rceil)^n\}$. Since k_d/k_0 is very large, we seek to modify the system so that we can capture the dynamics of the system using a reduced basis. We typically take $x_0 = k_0 = 1$ in simulations for the sake of simplicity. As a first step, we consider motion in two dimensions.

The Navier–Stokes equations for an incompressible fluid in two dimensions is

$$\frac{\partial \omega_{\boldsymbol{k}}}{\partial t} + \nu k^2 \omega_{\boldsymbol{k}} = \sum_{\boldsymbol{p},\boldsymbol{q}} \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \omega^*_{\boldsymbol{p}} \omega^*_{\boldsymbol{q}} + \boldsymbol{F_k}, \tag{4.24}$$

where $\omega \doteq \hat{\boldsymbol{z}} \cdot \boldsymbol{\nabla} \times \boldsymbol{u}$ is the vorticity and

$$\epsilon_{\boldsymbol{kpq}} \doteq \hat{z} \cdot (\boldsymbol{p} \times \boldsymbol{q})\, \delta(\boldsymbol{k} + \boldsymbol{p} + \boldsymbol{q}) \tag{4.25}$$

is the nonlinear interaction coefficient governing the effect of modes $\omega_{\boldsymbol{p}}$ and $\omega_{\boldsymbol{q}}$ on mode $\omega_{\boldsymbol{k}}$. The delta function in $\epsilon_{\boldsymbol{kpq}}$ imposes a geometrical constraint on the nonlinear interaction in equation (4.24).

One simple choice is to replace k_0 by a larger value, say $k_0 \to 2k_0$. However, this is equivalent to changing the physical domain, mapping $x_0 \to x_0/2$. Changing physical parameters in this fashion does not reproduce higher-resolution simulations. Instead of simply changing the grid spacing, we use spectral reduction to capture the dynamics of the original system while using fewer modes.

4.B.1 Binned Spectral Reduction

The goal of spectral reduction is to approximate equation (4.24) on a reduced grid composed of bins of modes. To each bin we assign a characteristic wave vector $\boldsymbol{K}$. Each wave vector $\boldsymbol{K}$ is associated with a number of fine-mesh wave vectors from the original grid. We denote this set $V_{\boldsymbol{K}}$. These modes are chosen so that the coarse-mesh wave vector $\boldsymbol{K}$ is representative of the wave vectors $V_{\boldsymbol{K}}$, i.e. the wave vectors in $V_{\boldsymbol{K}}$ are close to $\boldsymbol{K}$.

To each wave vector $\boldsymbol{K}$ we associate the variable

$$\Omega_{\boldsymbol{K}} \doteq \frac{1}{|V_{\boldsymbol{K}}|} \sum_{\boldsymbol{k} \in V_{\boldsymbol{K}}} \omega_{\boldsymbol{k}}, \tag{4.26}$$

where $|V_{\boldsymbol{K}}|$ is the size of the set $V_{\boldsymbol{K}}$. We can use this prescription to calculate the evolution of $\Omega_{\boldsymbol{K}}$, which is

$$\frac{d\Omega_{\boldsymbol{K}}}{dt} + \langle \nu_{\boldsymbol{k}} \omega_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} = \sum_{\boldsymbol{P},\boldsymbol{Q}} |V_{\boldsymbol{P}}||V_{\boldsymbol{Q}}| \left\langle \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \omega^*_{\boldsymbol{p}} \omega^*_{\boldsymbol{q}} \right\rangle_{\boldsymbol{KPQ}} + \langle F_{\boldsymbol{k}} \rangle_{\boldsymbol{K}}\,, \tag{4.27}$$

where the notation $\langle f \rangle_{\boldsymbol{KPQ}}$ denotes the mean value of f over the bins $\boldsymbol{K}$, $\boldsymbol{P}$,

and $\boldsymbol{Q}$. We close the equation by approximating $\omega_{\boldsymbol{k}}$ with $\Omega_{\boldsymbol{K}}$:

$$\frac{d\Omega_{\boldsymbol{K}}}{dt} + \langle \nu_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} \Omega_{\boldsymbol{K}} = \sum_{\boldsymbol{P},\boldsymbol{Q}} |V_{\boldsymbol{P}}||V_{\boldsymbol{Q}}| \left\langle \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \right\rangle_{\boldsymbol{KPQ}} \Omega^*_{\boldsymbol{p}} \Omega^*_{\boldsymbol{q}} + \langle F_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} \,. \tag{4.28}$$

While equation (4.27) is an exact prescription for $\partial \Omega_{\boldsymbol{K}} / \partial t$, the interaction coefficient

$$\left\langle \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \right\rangle_{\boldsymbol{KPQ}} \tag{4.29}$$

in equation (4.28) is not antisymmetric under the transformation $\boldsymbol{K} \leftrightarrow \boldsymbol{P}$, which breaks enstrophy conservation. In order to restore this symmetry, one uses the modified interaction coefficient

$$\frac{\langle \epsilon_{\boldsymbol{kpq}} \rangle_{\boldsymbol{KPQ}}}{Q^2}. \tag{4.30}$$

The resulting equation,

$$\frac{d\Omega_{\boldsymbol{K}}}{dt} + \langle \nu_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} \Omega_{\boldsymbol{K}} = \sum_{\boldsymbol{P},\boldsymbol{Q}} |V_{\boldsymbol{P}}||V_{\boldsymbol{Q}}| \frac{\langle \epsilon_{\boldsymbol{kpq}} \rangle_{\boldsymbol{KPQ}}}{Q^2} \Omega^*_{\boldsymbol{P}} \Omega^*_{\boldsymbol{Q}} + \langle F_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} \,, \tag{4.31}$$

conserves both the coarse-grained energy $\frac{1}{2} \sum_{\boldsymbol{K}} |V_{\boldsymbol{K}}| \, |\Omega_{\boldsymbol{K}}|^2 / K^2$ and enstrophy $\frac{1}{2} \sum_{\boldsymbol{K}} |V_{\boldsymbol{K}}| \, |\Omega_{\boldsymbol{K}}|^2$ in the absence of viscosity and forcing.

One particularly advantageous coarse-mesh choice is the set of radially spaced bins illustrated in Figure 4.14, which reach high wavenumbers and have high resolution near the origin while using very few modes. In one simulation the energy spectrum of forced-dissipative turbulence on a fine mesh of 683×683 dealiased modes was reproduced using only a few dozen radially spaced mesh points [Bowman *et al.* 1999].

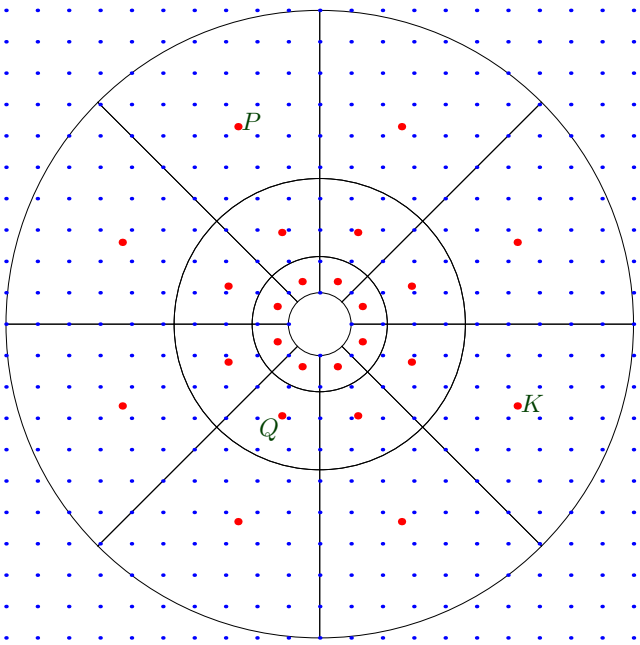

Figure 4.14: Polar wavenumber bin geometry.

However, this method of spectral reduction requires modification to reproduce equipartition spectra. If the bins are not uniformly spaced (as in radial spacing, for example), time needs to be rescaled in equation (4.31) by the bin area. That is,

$$\frac{|V_{\min}|}{|V_{\boldsymbol{K}}|}\frac{d\Omega_{\boldsymbol{K}}}{dt} + \langle \nu_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} \Omega_{\boldsymbol{k}} = \sum_{\boldsymbol{P},\boldsymbol{Q}} |V_{\boldsymbol{P}}||V_{\boldsymbol{Q}}| \frac{\langle \epsilon_{\boldsymbol{kpq}} \rangle_{\boldsymbol{KPQ}}}{Q^2} \Omega^*_{\boldsymbol{P}} \Omega^*_{\boldsymbol{Q}} + \langle F_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} , \quad (4.32)$$

where $|V_{\min}|$ is the minimum bin size. Under this transformation, equation (4.32) is able to reproduce the two-dimensional equipartition spectrum. Unfortunately, the resulting system is very stiff if the modes are radially spaced, where $|V_{\boldsymbol{K}}| \propto K^2$, which forces the time step to be very small, and the efficient numerical approximation of this equation is an open problem [Bowman *et al.* 2001]. Furthermore, energy is not conserved when time is

rescaled (although another related invariant is conserved).

In the case where the bins are of uniform size, the system is not abnormally stiff, and the correct equipartition spectrum is reproduced. However, the averaging operator smooths the delta function in $\epsilon_{\boldsymbol{kpq}}$, with the result that the sum in equation (4.31) is no longer a convolution. Because of this, fast-Fourier-transform methods are not directly applicable, leaving one with only purely spectral simulations, which is computationally expensive as compared with pseudospectral methods.

A pseudospectral implementation of spectral reduction that exploits the efficiency of the FFT has recently been developed by Bowman & Roberts [2011b]. This reduced model is shown to predict energy spectra in close agreement with pseudospectral simulations of the two-dimensional forced-dissipative energy and enstrophy cascades.

4.C Spectral Reduction of Complex Modes with Uncorrelated Phase

So far, spectral reduction has treated the modes as having the same amplitude and phase. However, it is straightforward to demonstrate that the phase of shell models are uncorrelated with each other, as is the case for the Navier–Stokes equations if the flow is spatially statistically homogeneous.

Consider the case where a bin consists of two modes with identical amplitudes and uncorrelated phase. The expectation value of the magnitude squared

of the sum of the two modes is

$$\begin{aligned}\left\langle |u_1 + u_2|^2 \right\rangle &= \left\langle |u_1|^2 \right\rangle + \cancel{\left\langle u_1 u_2^* \right\rangle} + \cancel{\left\langle u_1^* u_2 \right\rangle} + \left\langle |u_2|^2 \right\rangle \\ &= \left\langle |u_1|^2 \right\rangle + \left\langle |u_2|^2 \right\rangle .\end{aligned} \tag{4.33}$$

Spectral reduction defines a bin as

$$u_1^{(1)} \doteq \frac{u_1 + u_2}{2}, \tag{4.34}$$

as is required for energy conservation. However, the magnitude of the source term will, on average, correspond to

$$u_1^{(1)} = \frac{u_1 + u_2}{\sqrt{2}}, \tag{4.35}$$

which corresponds to an increase in the source term by a factor of $\sqrt{2}$.

This can be generalized to a radix-n decimation scheme where n modes are combined into a bin. If the phases of the binned modes are uncorrelated, then this is described by a discrete Wiener process, and the magnitude of the binned modes of uncorrelated phase will be decreased by $\sqrt{n}$ from the case where the phases are equal; the source term must therefore be increased by $\sqrt{n}$. While the importance of this factor is not immediately apparent in spectrally reduced simulations, its importance becomes more obvious when synchronizing spectrally reduced grids in the multispectral method.

Chapter 5

Multispectral Reduction

Wherein we describe the multispectral method, a decimation scheme that uses a hierarchy of spectrally reduced simulations to achieve non-uniform decimation in Fourier space. This method is applied to shell models of turbulence and the two-dimensional Navier–Stokes equations.

Spectral reduction allows one to simulate turbulent systems using a reduced basis, but does not reproduce the equipartition spectrum of the original system unless the decimation is uniform. This has serious disadvantages; often, one is interested in precise information about the large, more energetic scales, and would prefer to decimate only at small scales, which are often of less physical interest. For such systems, we would prefer to decimate variably with the high wavenumbers decimated while leaving the low wavenumbers untouched, allowing us to capture both large-scale and small-scale dynamics without resorting to a full-resolution (i.e. DNS) simulation.

Unable to decimate variably, as we would like, we instead decimate uni-

formly, as we must. Since we can only decimate uniformly, we employ a hierarchy of grids, each itself uniformly decimated, but with each grid decimated to a different degree. More-decimated grids reach the highest wavenumbers, moderately decimated grids reach moderate wavenumbers, and an undecimated grid covers low wavenumbers. This has the effect of decimating only at high wavenumbers, keeping the low wavenumbers intact.

Consider such a system consisting of two grids. Let G be the undecimated grid (the *grid*) with wave vectors $\boldsymbol{k}$ and harmonic vorticity amplitudes $\omega_{\boldsymbol{k}}$. Let SG be the decimated grid (the *subgrid*) with wave vectors $\boldsymbol{K}$ and harmonic vorticity amplitudes $\Omega_{\boldsymbol{K}}$. The enstrophy is defined as

$$Z = \frac{1}{2}\sum_{\boldsymbol{k}\in G} |\omega_{\boldsymbol{k}}|^2 + \frac{1}{2}\lambda \sum_{\substack{\boldsymbol{K}\in SG \\ \boldsymbol{K}\notin G}} |\Omega_{\boldsymbol{K}}|^2 , \tag{5.1}$$

where λ is a real constant factor that reflects the normalization of the subgrid amplitudes. Note that subgrid modes only contribute if they are not already present as part of the grid. The system conserves enstrophy if

$$\frac{dZ}{dt} = \mathrm{Re}\sum_{\boldsymbol{k}\in G} \omega_{\boldsymbol{k}} s^*_{\boldsymbol{k}} + \lambda\,\mathrm{Re}\sum_{\substack{\boldsymbol{K}\in SG \\ \boldsymbol{K}\notin G}} \Omega_{\boldsymbol{K}} S^*_{\boldsymbol{K}} = 0, \tag{5.2}$$

where $s_{\boldsymbol{k}}$ is the source term for mode $\omega_{\boldsymbol{k}}$ on the grid and $S_{\boldsymbol{K}}$ is the source term for mode $\Omega_{\boldsymbol{K}}$ on the subgrid, and * denotes complex conjugation. The energy is defined as

$$E = \frac{1}{2}\sum_{\boldsymbol{k}\in G} \frac{|\omega_{\boldsymbol{k}}|^2}{k^2} + \frac{1}{2}\lambda \sum_{\substack{\boldsymbol{K}\in SG \\ \boldsymbol{K}\notin G}} \frac{|\Omega_{\boldsymbol{K}}|^2}{K^2}, \tag{5.3}$$

with the subgrid modes only contributing if they are not already present as part of the grid. The system conserves energy if

$$\frac{dE}{dt} = \mathrm{Re}\sum_{\boldsymbol{k}\in G} \frac{\omega_{\boldsymbol{k}}}{k^2} s^*_{\boldsymbol{k}} + \lambda\,\mathrm{Re}\sum_{\substack{\boldsymbol{K}\in SG \\ \boldsymbol{K}\notin G}} \frac{\Omega_{\boldsymbol{K}}}{K^2} S^*_{\boldsymbol{K}} = 0. \tag{5.4}$$

The grid's temporal evolution is given by the undecimated evolution equation. The subgrid's temporal evolution is given by a decimated evolution equation that is filtered to remove source terms that are already present in the grid. That is, if the subgrid modes $\Omega_{\boldsymbol{P}}$ and $\Omega_{\boldsymbol{Q}}$ influence $\Omega_{\boldsymbol{K}}$, then we remove this interaction from the subgrid source if $\boldsymbol{K}$, $\boldsymbol{P}$, and $\boldsymbol{Q}$ are also present in the grid. Similarly, linear and forcing terms are removed from the subgrid if they are also already accounted for by the grid.

The grid and the subgrid are advanced in time using a conventional explicit numerical method, such as a Runge–Kutta method. The source term for the grid differs from the source term for the subgrid, so the grids will drift out of sync with one another over time. One must therefore synchronize the two grids, so that changes in the grid will be reflected in the subgrid, and vice versa. Moreover, energy conservation, given by equation (5.4), and enstrophy conservation, given by equation (5.2), will not necessarily hold if the grids are left unsynchronized.

One is tempted to perform this action by adding the grid's source term onto the subgrid and the subgrid's source term onto the grid. This ends up being similar to a variably decimated spectral reduction, which produces the incorrect equipartition spectrum for inviscid, unforced systems [Roberts 2006].

There are two basic ways to combine numerical integration of the independent grids and synchronization; either via *serial synchronization*, with separate projection and prolongation operators, or via *parallel synchronization*, where the projection and prolongation operators are performed simultaneously.

With serial synchronization, one first advances the grid from time t to $t + \tau$, as in Figure 5.1. Then, one *project*s from the grid onto the subgrid,

which copies the changes undergone by the grid as $t \to t + \tau$ onto the subgrid at time t, as in Figure 5.2. Then, the subgrid is advanced from time t to time $t + \tau$, as in Figure 5.3. Finally, the changes that the subgrid underwent as $t \to t + \tau$ are copied from the subgrid onto the grid via *prolongation*. This is shown schematically in Figure 5.4.

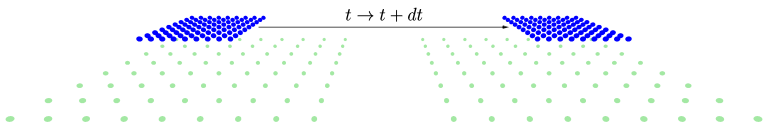

Figure 5.1: Stage one of serial advancement: advancing the grid from time t to time $t + \tau$.

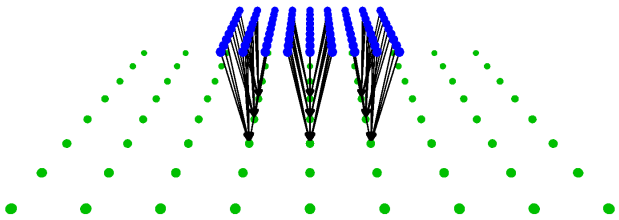

Figure 5.2: Stage two of serial advancement: projecting from the grid to the subgrid.

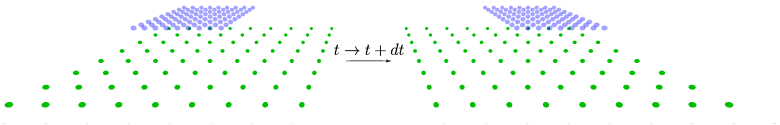

Figure 5.3: Stage three of serial advancement: advancing the subgrid from time t to time $t + \tau$.

Parallel synchronization advances the grid and the subgrid from time t to time $t + \tau$ simultaneously, as in Figure 5.5. The two grids are then synchro-

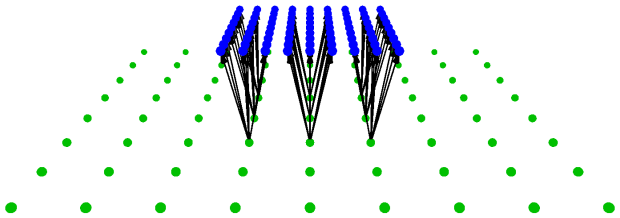

Figure 5.4: Stage four of serial advancement: prolong from the subgrid to the grid.

nized simultaneously, as in Figure 5.6. As will be seen in the remainder of this chapter, grid geometries and conserved invariants play important roles in determining the details for synchronization operators.

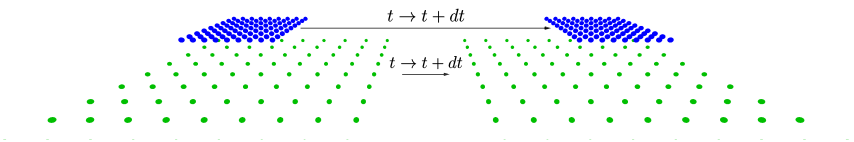

Figure 5.5: Stage one of parallel advancement: advancing both the grid and the subgrid from time t to time $t + \tau$.

5.A General Properties of Synchronization Operators

The synchronization operators play an important rôle in the multispectral method, and choices in their design can have a dramatic effect on the result of the simulation. In this section, we lay out some general principles to which synchronization operators should adhere.

First, the synchronization operators should preserve the invariants of the

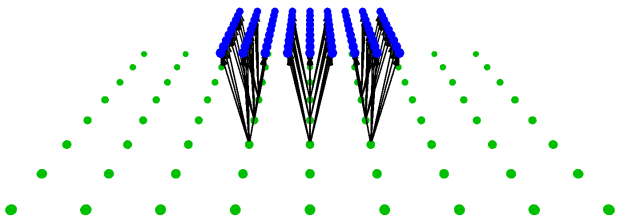

Figure 5.6: Stage two of parallel advancement: synchronizing the grid and the subgrid simultaneously.

original system. The spectrally reduced shell models discussed in this work have only energy as a conserved quantity, whereas the two-dimensional Navier–Stokes equations conserve both energy and enstrophy, and care must be taken to preserve these invariants. Since the subgrid overlaps with the grid, the subgrid evolution equation may transfer some energy (or another invariant) to this overlapping region, which must be transferred to the grid in order to be counted as part of the energy of the entire system.

The synchronization operators should preserve as much detail in the system as possible. For instance, if modes on the grid are synchronized with a bin of modes on the subgrid, then the synchronization operator should preserve the heterogeneity of the grid modes associated with the subgrid bin. Following this principle keeps as much detail as possible in the system, which is, after all, the aim of variable-decimation.

The synchronization operators should be local in Fourier space. The overlapping grid modes and subgrid bins occupy the same region of Fourier space, and thus have the same physical interpretation, i.e. they denote the amplitude of certain waves in the system. It is reasonable to require that this interpretation be preserved by the multispectral method.

If the source term is zero on the subgrid then the synchronization operator should not introduce any changes on the grid. That is, if the evolution equation on the subgrid does not indicate that anything should change, then the synchronization operator should not impose any change. Similarly, if the source term on the grid is zero, then the synchronization operator should not introduce any changes on the subgrid. Moreover, the effect of the synchronization operator should be proportional, in some norm, to the effect of the evolution equations from time t to time $t + \tau$.

While this is not technically a property of the synchronization operator, it is worth mentioning at this point that the multispectral method should avoid double-counting interactions in Fourier space. That is, if modes $\omega_{\boldsymbol{k}}$, $\omega_{\boldsymbol{p}}$, and $\omega_{\boldsymbol{q}}$ interact on the grid, they should not interact on the subgrid. This can be accomplished by simply not calculating such subgrid interactions (as in the case of shell models) or, when this is not computationally convenient, by removing such interactions from the complete evolution equation (as is done for the Navier–Stokes equations).

One important test case to consider when designing multispectral methods is when the nonlinear interaction is not present and the source term for mode $\omega_{\boldsymbol{k}}$ is proportional to $\omega_{\boldsymbol{k}}$. In this case, $\omega_{\boldsymbol{k}}$ is simply exponential in time. Designing methods with this problem in mind allows one to eliminate a number of possibilities that would otherwise seem, a priori, to be reasonable choices.

5.B Multispectral Reduction of Shell Models of Turbulence

Shell models of turbulence provide excellent test beds for the multispectral method [Roberts 2006]; they are much more computationally accessible than Navier–Stokes turbulence, have simple geometry, and yet possess many of the same challenges of Navier–Stokes turbulence, having a nonlinear source term and a wide range of active scales.

The shell models under consideration use a single complex quantity u_n to represent all modes with wave vector $\boldsymbol{k}$ in a geometrically spaced shell with $|\boldsymbol{k}| \in [k_n, k_{n+1})$, $k_n = k_0\lambda^n, n = 0, \ldots, N$. Since the DN and GOY shell models have only nearest- or next-nearest-neighbour nonlinear source terms, removing interactions from the subgrid that are present on the grid is straightforward. Moreover, the subgrid source term affects very few subgrid modes that are redundant with grid modes, so few modes actually require synchronization.

The GOY model spectrally reduces to the DN model, and the DN model spectrally reduces to a DN model with modified parameters. Since the DN model is a fixed point for spectral reduction, we develop the multispectral method for the DN model, with applications to the GOY model being relatively straightforward. In addition to spectral reduction, it is necessary to scale the nonlinearity by the square-root of the binning factor, which accounts for the fact that the shell amplitudes u_n are uncorrelated with each other.

5.B.1 Synchronization Schemes

The DN model has nearest-neighbour interactions, as shown in Figure 5.7. Some of these interactions on the subgrid are present on the grid, and are thus removed from the subgrid source term, as shown in Figure 5.8, in which it can be seen that the subgrid source function is equivalent to a shell model with its first shell coinciding with the last two shells of the grid.

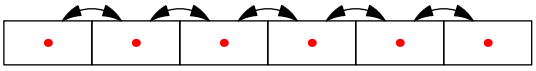

Figure 5.7: The DN shell model with nonlinear interactions represented by arrows.

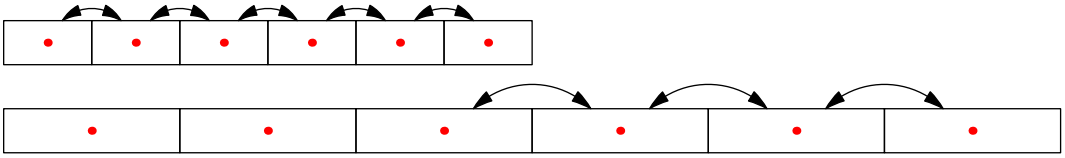

Figure 5.8: Nonlinear source diagram for the multispectral DN model.

The subgrid has shell spacing $k_{0\mathrm{SG}}$ given by $k_{0\mathrm{SG}} = k_{0\mathrm{G}}^2$, requiring half as many shells to cover the same range of scales as would the grid. Thus, the first active shell on the subgrid represents modes with $\{\boldsymbol{k} : |\boldsymbol{k}| \in (k_0^2\lambda^n, k_0^2\lambda^{n+1})\}$, which is the same range as the last two shells for the grid. Thus, we require that the energy in the last two shells of the grid be equal to that in the first shell of the subgrid.

For serial synchronization, the grid is first advanced from time t to time $t+\tau$ via a numerical integrator, which evolves the shell amplitudes on the grid from $u_n^{(0)}(t)$ to $\widetilde{u}_n^{(0)}(t+\tau)$. Here, a tilde over a quantity indicates an intermediate quantity that still needs to be synchronized, and with ${u_n}^{(0)}$ denoting a grid

mode and $u_n{}^{(1)}$ denoting a subgrid mode/bin. Then, we project the changes from the grid onto the subgrid, as shown in Figure 5.9. Since the DN model has

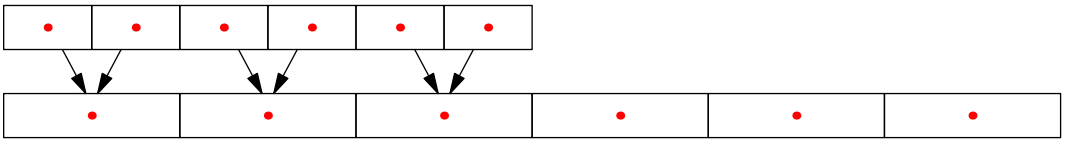

Figure 5.9: Projection diagram for the multispectral DN model.

only energy as a conserved quantity, projection should ensure that the energy of the subgrid shell has the energy of the two corresponding grid shells. That is, the subgrid shell amplitude $u_n^{(1)}(t)$ should be rescaled to have amplitude $\widetilde{u}_n^{(1)}(t)$ as given by

$$\Delta \frac{1}{2} \left| \widetilde{u}_n^{(1)}(t) \right|^2 = \frac{1}{2} \left| \widetilde{u}_{2n}^{(0)}(t+\tau) \right|^2 + \frac{1}{2} \left| \widetilde{u}_{2n+1}^{(0)}(t+\tau) \right|^2, \tag{5.5}$$

where $\Delta = 2$ is the decimation factor. While one would normally project over the entire region on the subgrid that overlaps with the grid, one need only project the last two shells of the grid onto the subgrid since, as can be seen in Figure 5.8, the other subgrid modes are not evolved on the subgrid. One then evolves the subgrid modes $\widetilde{u}_n^{(1)}(t)$ from time t to time $t+\tau$, with value $u_n^{(1)}(t+\tau)$. One then prolongs from the subgrid onto the grid, as shown diagrammatically in Figure 5.10. We again require that the energies match in the corresponding shells, which may be accomplished by prescribing

$$\frac{1}{2} \left| u_{2n}^{(0)}(t+\tau) \right|^2 + \frac{1}{2} \left| u_{2n+1}^{(0)}(t+\tau) \right|^2 = \Delta \frac{1}{2} \left| u_n^{(1)}(t+\tau) \right|^2, \tag{5.6}$$

where $u_n^{(0)}(t+\tau)$ is the new value for the shell amplitudes on the grid at time $t+\tau$.

Alternatively, one can choose to synchronize in parallel, as shown diagrammatically in Figure 5.11. In this case, both the grid and the subgrid are

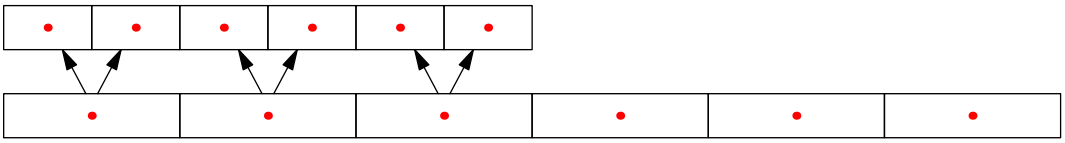

Figure 5.10: Prolongation diagram for the multispectral DN model.

advanced from time t to time $t+\tau$, and then projection and prolongation are done simultaneously. In this case, one scales $u_{2n+1}^{(0)}(t+\tau)$, $u_{2n}^{(0)}(t+\tau)$, and

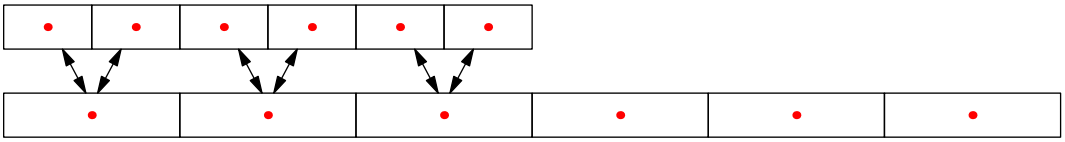

Figure 5.11: Parallel synchronization of the multispectral DN model.

$u_n^{(1)}(t+\tau)$ to account for energy changes due to source terms from both the grid and the subgrid. In the event that both the grid and the subgrid attempt to remove more energy from the overlapping shells than was present in the first place, one simply repeats the integration using a reduced time step.

5.B.2 Simulations

Spectral reduction maintains the H theorem for each grid in the multispectral simulation, so each grid will tend towards the same equipartition spectrum. Therefore, the two grids tend towards the same equipartition spectrum together, as can be seen in Figure 5.12.

The multispectral DN shell model reproduces a Kolmogorov energy spectrum for forced-dissipative simulations, as can be seen in Figure 5.13, while requiring fewer degrees of freedom. Energy spectra are compared with that

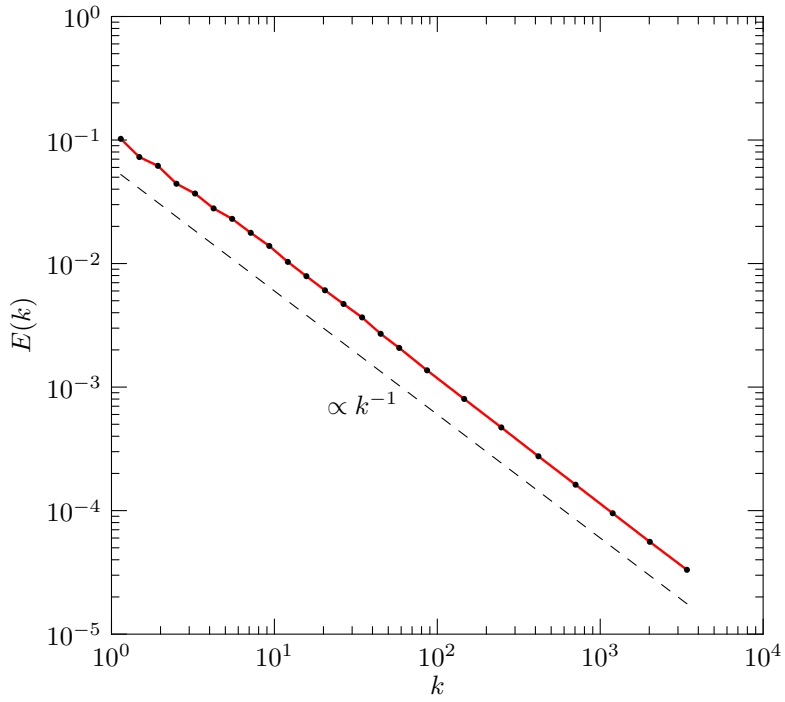

Figure 5.12: Energy spectrum for the inviscid unforced multispectral DN model with two grids and serial synchronization, exhibiting a k^{-1} spectral slope.

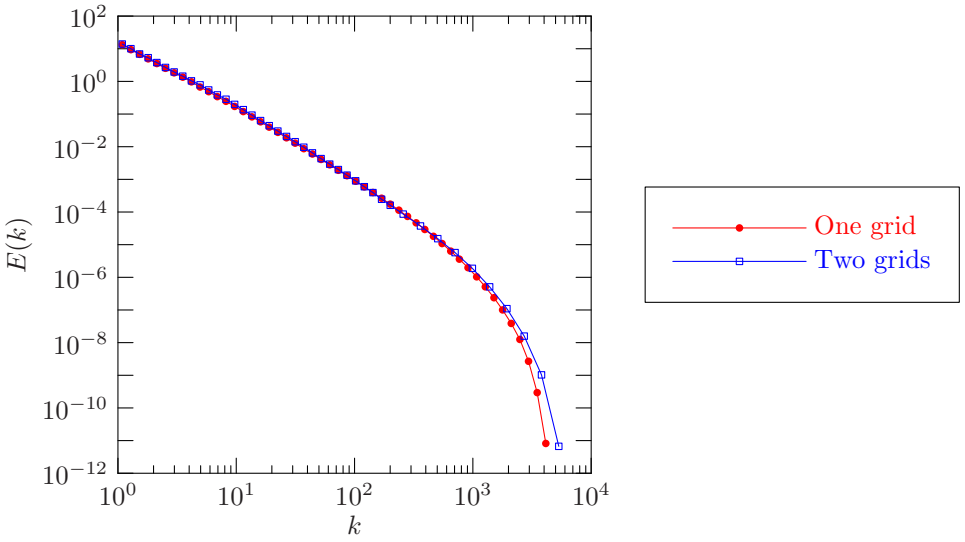

Figure 5.13: Multispectral simulation of the forced-dissipative DN model with two grids and serial synchronization.

of the undecimated simulation, which is also shown after having been binned in post-processing. Serial synchronization is straightforward to generalize to a hierarchy of grids, and we show forced-dissipative energy spectra with three grids in Figure 5.14 and four grids in Figure 5.15. Each successive grid is spectrally reduced using a radix-two scheme. In order to account for the fact that the modes have uncorrelated phase, the source term is multiplied by $\sqrt{2}$ for each successive decimation, as described in Section 4.C.

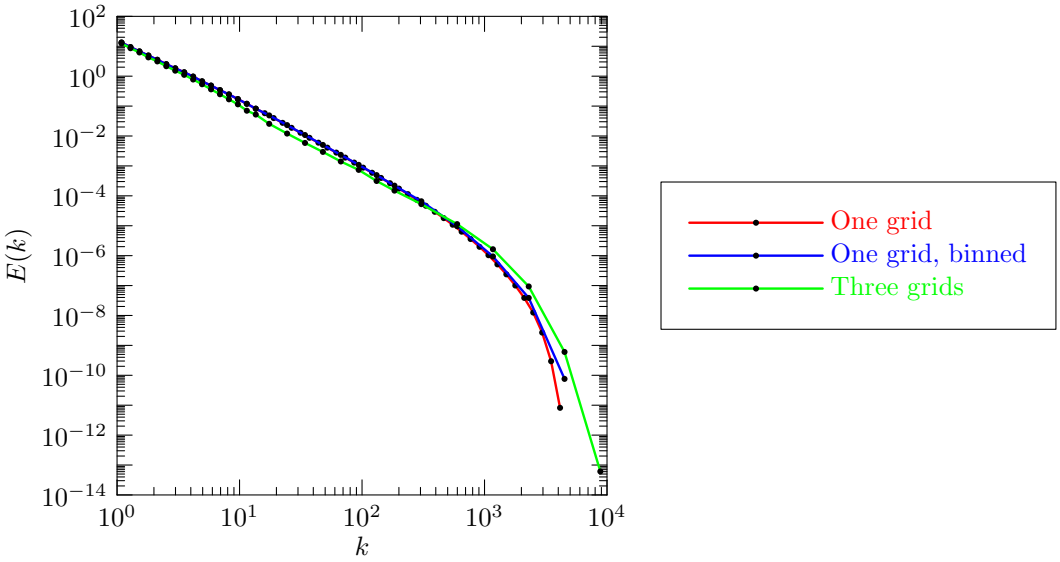

Figure 5.14: Multispectral simulation of the forced-dissipative DN model, with three grids and serial synchronization.

Shell models of turbulence work well with the multispectral method. Solutions to shell models of turbulence are very simple to perform as compared to solutions of the Navier–Stokes equations; their rôle here is as a test bed for developing the multispectral method, allowing us to develop an understanding of basic properties of the multispectral method before applying the technique to more complicated systems.

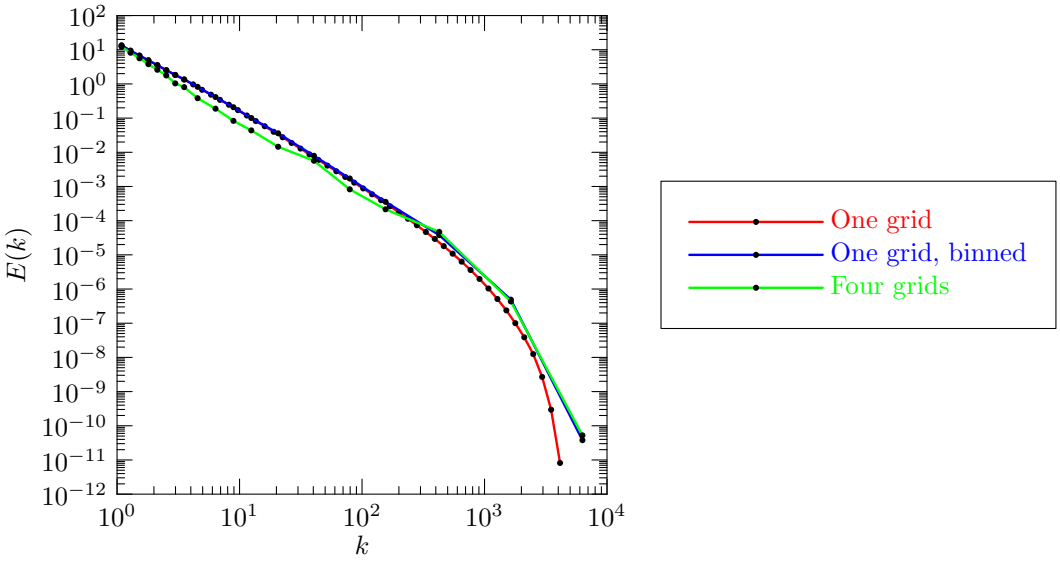

Figure 5.15: Multispectral simulation of the forced-dissipative DN model, with four grids and serial synchronization.

5.C Multispectral Reduction of 2D Incompressible Navier–Stokes Turbulence

The application of multispectral reduction to the two-dimensional Navier–Stokes equations poses new difficulties, in addition to those faced by shell models of turbulence.

First, the geometry is more complicated. The radix-two binning used on the geometrically spaced one-dimensional grid for shell models of turbulence is not available for the Cartesian grid on which the spectral Navier–Stokes equations are solved where the bin boundaries do not line up.

Second, the spectral reduction map has a fixed point when applied to shell models of turbulence, but we are not aware of any similar behaviour with the Navier–Stokes equations.

Third, the two-dimensional incompressible Navier–Stokes equations conserve two sign-definite quadratic invariants, i.e. energy and enstrophy, which must be maintained by the synchronization operator, whereas the spectrally reduced shell models considered in this work have only energy as a conserved quantity.

Finally, simulations of the Navier–Stokes equations are more difficult to perform, requiring highly optimized code and much more computing power than shell models.

5.C.1 Coincident Mode Synchronization

The multispectral two-dimensional Navier–Stokes system must conserve both energy and enstrophy. We choose to evolve the system according to the vorticity equation (1.13), with vorticity being the fundamental quantity. This implies that enstrophy conservation is relatively straightforward as compared to energy conservation, which is wave-vector dependent.

Since the grid and the subgrid overlap in Fourier space, we must synchronize the grids to ensure consistent energy and enstrophy contributions from modes in the overlapping region. Since the subgrid is decimated, the overlapping region does not have a one-to-one map between grid modes and subgrid modes. This problem can be avoided by choosing to synchronize only a subset of the grid modes with those on the subgrid. In particular, some modes on the grid and subgrid coincide exactly in Fourier space (i.e. they correspond to identical wave vectors). We call such modes *coincident*. Synchronizing the grid and the subgrid by modifying these modes is called *coincident-mode synchronization*.

Coincident-mode synchronization has been developed for systems where the subgrid has been decimated to include half of the modes, one quarter of the modes, or one ninth of the modes, which we denote oblique, radix-two, and radix-three, respectively. Note that the oblique subgrid is rotated by 45° with respect to the grid.

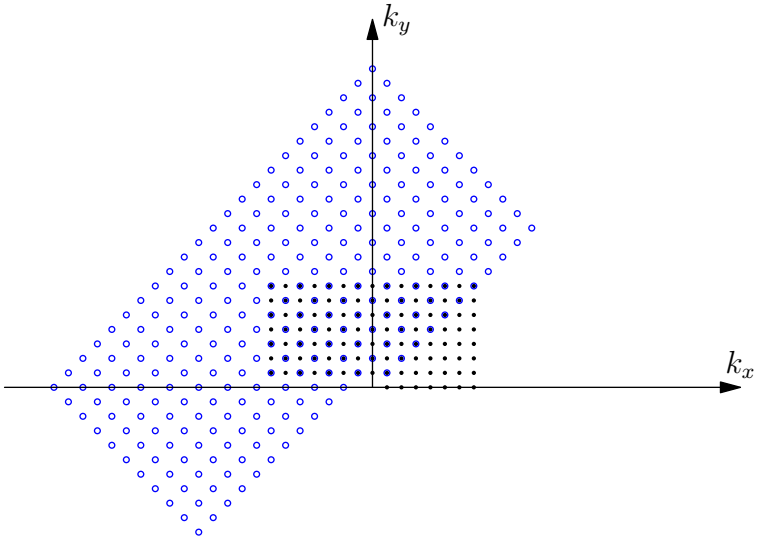

Figure 5.16: Grid and subgrid geometry for the oblique coincident-mode multispectral method. Black dots indicate grid modes and blue circles indicate subgrid modes.

Let G denote the set of modes on the grid, and SG denote the set of modes on the subgrid. The enstrophy of the coincident mode two-dimensional multispectral system is

$$Z = \frac{1}{2}\sum_{\boldsymbol{k}\in G} |\omega_{\boldsymbol{k}}|^2 + \frac{1}{2}\sum_{\substack{\boldsymbol{k}\in SG \\ \boldsymbol{k}\notin G}} |\omega_{\boldsymbol{k}}|^2 , \tag{5.7}$$

and the energy is

$$E = \frac{1}{2}\sum_{\boldsymbol{k}\in G} \frac{|\omega_{\boldsymbol{k}}|^2}{k^2} + \frac{1}{2}\sum_{\substack{\boldsymbol{k}\in SG \\ \boldsymbol{k}\notin G}} \frac{|\omega_{\boldsymbol{k}}|^2}{k^2}. \tag{5.8}$$

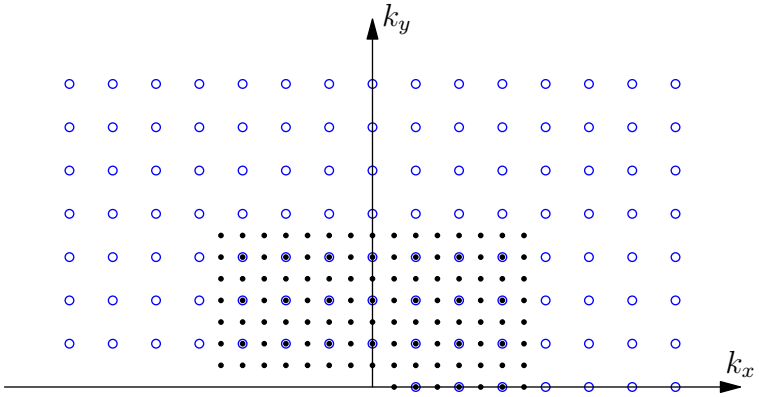

Figure 5.17: Grid and subgrid geometry for the radix-two coincident-mode multispectral method. Black dots indicate grid modes and blue circles indicate subgrid modes.

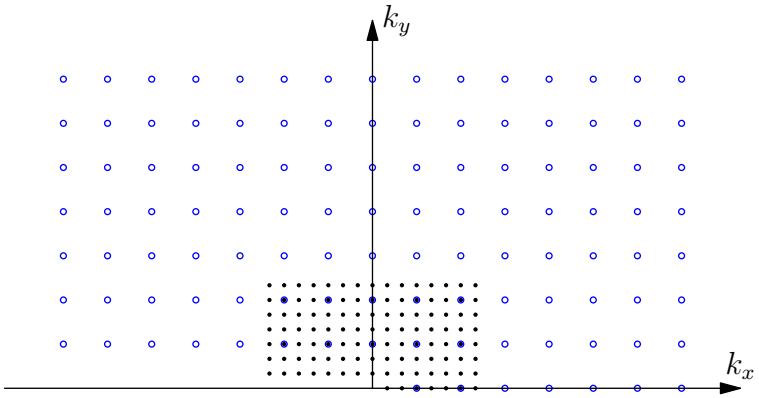

Figure 5.18: Grid and subgrid geometry for the radix-three coincident-mode multispectral method. Black dots indicate grid modes and blue circles indicate subgrid modes.

The nonlinear source term is calculated on both the grid and the subgrid. Synchronization operators copy the change from the grid onto the subgrid and vice-versa. In order to avoid double counting, interactions that are included in the grid evolution are removed from the subgrid. This is accomplished by calculating the nonlinear source term on just those subgrid modes that coincide with grid modes, and then subtracting this from the general subgrid source term. This is straightforward for the radix-two and radix-three decimation schemes, where the overlapping region is rectangular and thus amenable to FFT-based convolutions. The oblique scheme is slightly more troublesome, and requires calculating the convolution on a rectangular grid that includes the triangular overlapping region, and then using Theorem 1.1 to restrict the source term to just the overlapping region.

The coincident-mode multispectral method has been developed for both serial and parallel synchronization.

Serial Synchronization

Serially advanced coincident-mode synchronization follows the general scheme for serial synchronization discussed in the beginning of this chapter. We assume that the grids are synchronized at the beginning of the time step, i.e. the energy and enstrophy of coincident modes are equal. The first stage is to advance the grid from time t to time $t + \tau$. The second stage is to project from the grid onto the subgrid. Let ω and Ω be coincident modes from the grid and subgrid, respectively. Since they are coincident, they have identical wave vectors. Therefore, if

$$\frac{1}{2}\left|\omega\right|^2 = \frac{1}{2}\left|\Omega\right|^2, \tag{5.9}$$

then

$$\frac{1}{2}\frac{|\omega|^2}{k^2} = \frac{1}{2}\frac{|\Omega|^2}{K^2}, \tag{5.10}$$

since $|k^2| = |K^2|$. In other words, if the grid mode and the subgrid mode have the same enstrophy, then, since the grid mode and the subgrid mode have the same wavenumber, the grid mode and the subgrid mode will also have the same energy. This provides a simple way to guarantee energy and enstrophy conservation when projecting from the grid onto the subgrid. Let $\omega(t)$ be the amplitude of the grid mode at time t, and let $\Omega(t) = \omega(t)$ be the amplitude of the subgrid mode at the beginning of the time step at time t. Let $\widetilde{\omega}(t+\tau)$ be the amplitude of the grid mode after being advanced τ units in time. When projecting from the grid onto the subgrid, we simply set

$$\widetilde{\Omega}(t) \leftarrow \widetilde{\omega}(t+\tau). \tag{5.11}$$

This communicates the changes due to the evolution equation on the grid onto the coincident subgrid modes, including changes in energy, enstrophy, and phase. The next stage is to evolve the subgrid mode $\widetilde{\Omega}(t)$ from time t to time $t+\tau$, the result of which we denote $\Omega(t+\tau)$. Finally, one prolongs from the subgrid onto the grid by setting

$$\omega(t+\tau) \leftarrow \Omega(t+\tau), \tag{5.12}$$

which completes the time step.

Theorem 5.1: *If the grid evolution equation conserves the grid energy E_G and the grid enstrophy Z_G given by*

$$E_G = \frac{1}{2}\sum_{\boldsymbol{k}\in G}\frac{|\omega_{\boldsymbol{k}}|^2}{k^2}, \qquad Z_G = \frac{1}{2}\sum_{\boldsymbol{k}\in G}|\omega_{\boldsymbol{k}}|^2,$$

and the subgrid conserves

$$E_{SG} = \frac{1}{2}\sum_{\boldsymbol{k}\in SG}\frac{|\omega_{\boldsymbol{k}}|^2}{k^2}, \qquad Z_{SG} = \frac{1}{2}\sum_{\boldsymbol{k}\in SG}|\omega_{\boldsymbol{k}}|^2,$$

then the serial coincident-mode scheme conserves the energy E and enstrophy Z given by

$$E = \frac{1}{2}\sum_{\boldsymbol{k}\in G}\frac{|\omega_{\boldsymbol{k}}|^2}{k^2} + \frac{1}{2}\sum_{\substack{\boldsymbol{k}\in SG\\ \boldsymbol{k}\notin G}}\frac{|\omega_{\boldsymbol{k}}|^2}{k^2},$$
$$Z = \frac{1}{2}\sum_{\boldsymbol{k}\in G}|\omega_{\boldsymbol{k}}|^2 + \frac{1}{2}\sum_{\substack{\boldsymbol{k}\in SG\\ \boldsymbol{k}\notin G}}|\omega_{\boldsymbol{k}}|^2. \tag{5.13}$$

Proof. Let G_{c} denote the set of grid modes that coincide with subgrid modes. Let G_{nc} be the set of grid modes that do not coincide with subgrid modes. The change in invariants on G_{c} and the change in invariants on G_{nc} sum to zero since the grid's evolution equation is conservative. Let SG_c be the coincident subgrid modes and SG_{nc} be the non-coincident subgrid modes. The change in invariants on SG_{c} and the change in invariants on SG_{nc} also sum to zero since the subgrid's evolution equation is conservative.

The new value of the vorticity on G_c is projected onto SG_c, which then changes the invariants on the subgrid while not affecting invariants of the two-grid system, which does not sum over SG_c. Advancing the subgrid from time t to time $t+\tau$ conserves the invariants on the subgrid. Therefore, the change in invariants on SG_c and the change in invariants on SG_{nc} must sum to zero. The change in the invariants on SG_c is copied to G_c during prolongation.

Thus, the total change in the invariants is equal to the change in the invariants on $G_{\mathrm{c}} \cup G_{\mathrm{nc}} \cup SG_c \cup SG_{\mathrm{nc}}$. Since the total change in invariants on $G_{\mathrm{c}} \cup G_{\mathrm{nc}}$ is zero, and the total change in invariants on $SG_c \cup SG_{\mathrm{nc}}$ is zero, the total change in invariants for the multispectral system is also zero. □

The grid and the subgrid maintain the original evolution equation and maintain a large number of triadic interactions, both of which are necessary to reproduce the equipartition spectrum correctly. Coincident-mode synchronization reproduces the equipartition spectrum, as shown in Figure 5.19.

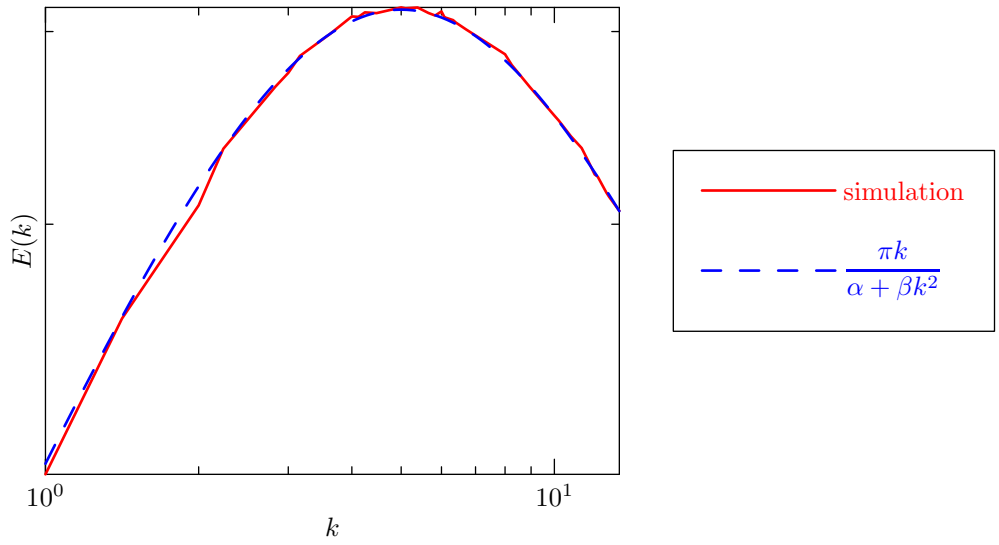

Figure 5.19: The energy spectrum of an inviscid unforced serially synchronized coincident-mode multispectral simulation of the two-dimensional Navier–Stokes equations, showing only stochastic deviations from the theoretical equipartition spectrum, with $\alpha = 25$, $\beta = 1$. The grid and subgrid have dimension 15×8. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$.

Unfortunately, serial synchronization of coincident modes does not do as good a job reproducing the energy spectrum of forced-dissipative simulations, as shown in Figure 5.20. This defect becomes more pronounced as one decimates more and more modes, as shown in Figure 5.21, and Figure 5.22. In order to further understand the source of this discrepancy, we first examine the parallel synchronization scheme in order to determine whether the choice of serial synchronization is the cause of the error.

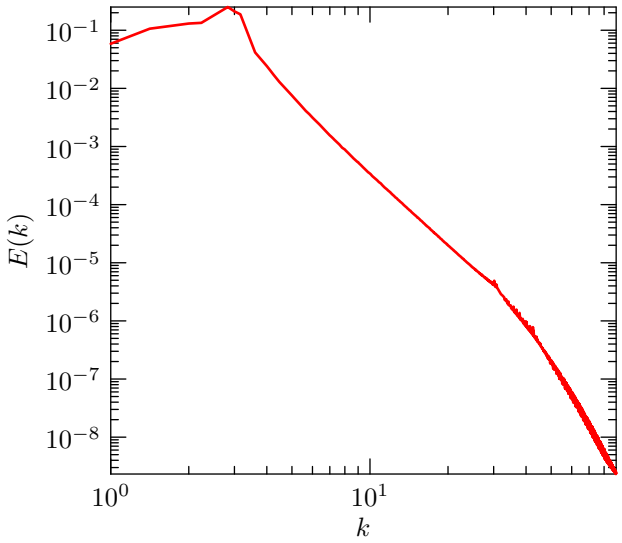

Figure 5.20: Energy spectrum from a forced-dissipative serially synchronized coincident-mode two-dimensional Navier–Stokes simulation with an oblique subgrid, showing an energy pile-up at the grid/subgrid interface. The grid and subgrid have dimension 63×32. The simulation was performed with $\nu = 5 \times 10^{-4}$, and white-noise forcing with unit enstrophy injection rate on modes with $k \in [2, 4]$. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$.

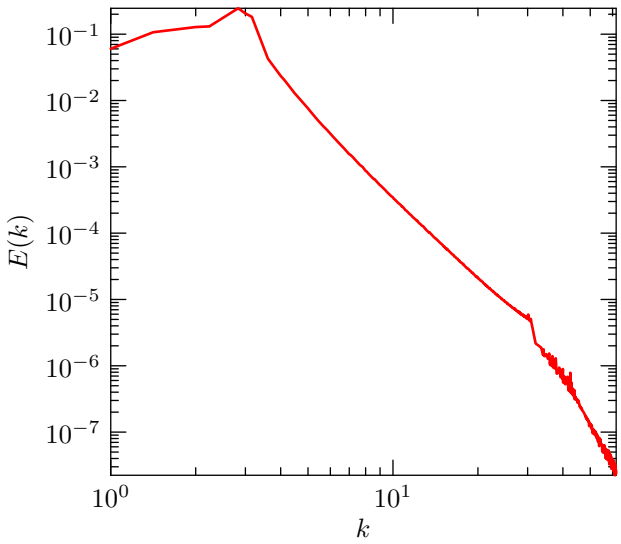

Figure 5.21: Energy spectrum from a forced-dissipative serially synchronized coincident-mode two-dimensional Navier–Stokes simulation with radix two, using the parameters of Figure 5.20. An increased energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$.

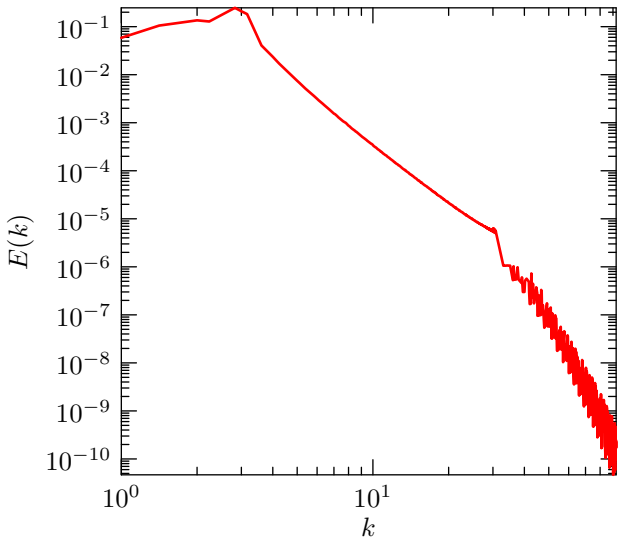

Figure 5.22: Energy spectrum from a forced-dissipative serially synchronized coincident-mode two-dimensional Navier–Stokes simulation with radix three, using the parameters of Figure 5.20. A large energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$.

Parallel Synchronization

We can also synchronize the coincident modes in parallel. With this method, the grid and subgrid are both advanced from time t to time $t+\tau$, at which point the coincident modes on both the grid and the subgrid are synchronized.

Suppose that a coincident grid mode $\omega_{\boldsymbol{k}}$ and subgrid mode $\Omega_{\boldsymbol{k}}$ have the same magnitude squared at time t. These modes are evolved from their values $\omega_{\boldsymbol{k}}(t)$ and $\Omega_{\boldsymbol{k}}(t)$ at time t to $\widetilde{\omega}_{\boldsymbol{k}}(t+\tau)$ and $\widetilde{\Omega}_{\boldsymbol{k}}(t+\tau)$ by their respective evolution equations. The change in enstrophy for the grid mode is given by

$$\Delta Z_G \doteq \frac{1}{2}\big(|\widetilde{\omega}_{\boldsymbol{k}}(t+\tau)|^2 - |\omega_{\boldsymbol{k}}(t)|^2\big), \tag{5.14}$$

and the change in enstrophy for the subgrid modes is given by

$$\Delta Z_{SG} \doteq \frac{1}{2}\left(\left|\widetilde{\Omega}_{\boldsymbol{k}}(t+\tau)\right|^2 - |\Omega_{\boldsymbol{k}}(t)|^2\right). \tag{5.15}$$

In order to conserve enstrophy, we require that the grid and subgrid modes each have new enstrophy

$$Z_{\boldsymbol{k}} = \frac{1}{2}\,|\omega_{\boldsymbol{k}}(t)|^2 + \Delta Z_G + \Delta Z_{SG}, \tag{5.16}$$

which may be accomplished by setting

$$|\omega_{\boldsymbol{k}}(t+\tau)| = |\Omega_{\boldsymbol{k}}(t+\tau)| = \sqrt{2Z_{\boldsymbol{k}}}, \tag{5.17}$$

providing us with a framework for synchronizing the grid and the subgrid. In practice, we use

$$\omega_{\boldsymbol{k}}(t+\tau) \leftarrow \widetilde{\omega}_{\boldsymbol{k}}(t+\tau)\sqrt{\frac{Z_{\boldsymbol{k}}}{\frac{1}{2}\,|\widetilde{\omega}_{\boldsymbol{k}}(t+\tau)|^2}}, \tag{5.18}$$

$$\Omega_{\boldsymbol{k}}(t+\tau) \leftarrow \widetilde{\Omega}_{\boldsymbol{k}}(t+\tau)\sqrt{\frac{Z_{\boldsymbol{k}}}{\frac{1}{2}\left|\widetilde{\Omega}_{\boldsymbol{k}}(t+\tau)\right|^2}}, \tag{5.19}$$

which maintains the phase information for the grid and subgrid. In the case where $Z_{\boldsymbol{k}}$ is negative, the time step is repeated with τ reduced. In practice, this has never been necessary.

Theorem 5.2: *The synchronization operators given in equation* (5.18) *and equation* (5.19) *conserve energy and enstrophy.*

Proof. As before, let G_{c} and SG_{c} denote coincident modes on the grid and subgrid, and G_{nc} and SG_{nc} denote non-coincident modes. The evolution equations will conserve energy and enstrophy on $G_{\mathrm{c}} \cup G_{\mathrm{nc}}$ and $SG_{\mathrm{c}} \cup SG_{\mathrm{nc}}$. The synchronization operator given in equation (5.18) copies the change in enstrophy from SG_{c} onto G_{c}. Since enstrophy is calculated based on all modes in $G_{\mathrm{c}} \cup G_{\mathrm{nc}} \cup SG_{\mathrm{nc}}$, and the synchronization operator guarantees that the change in enstrophy on $G_{\mathrm{c}} \cup SG_c$ is present in G_{c}, the system will conserve enstrophy. Moreover, since the coincident modes have identical wave vectors, enstrophy conservation implies energy conservation. □

As with the serial-synchronization method, the parallel method does a good job reproducing the equipartition spectrum, as shown in Figure 5.23. The use of serial synchronization is evidently not the reason that the forced-dissipative spectrum deviates from the full-resolution simulation, as this defect is found with the parallel method as well, as shown in Figures 5.24–5.26. The source of error is most likely owing to our choice of decimation, the effect of which is made clear by considering the decimated direct numerical simulation.

Decimated Direct Numerical Simulations

Decimated direct numerical simulation (decimated DNS) is a full-resolution simulation, including all interactions on a fully resolved grid, which is then

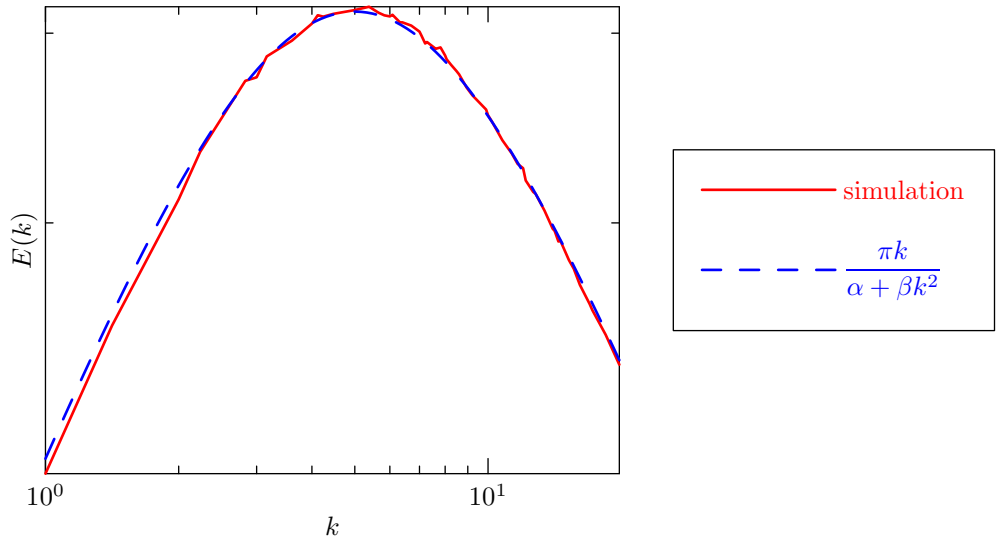

Figure 5.23: The energy spectrum of an unforced, inviscid coincident-mode parallel-synchronized multispectral simulation of the two-dimensional Navier–Stokes equations showing only stochastic deviations from the theoretical equipartition spectrum, with $\alpha = 25$, $\beta = 1$. The grid and subgrid have dimension 15×8. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$.

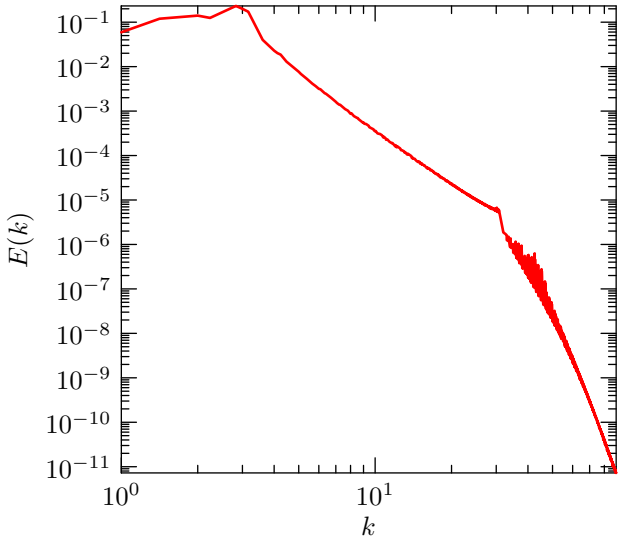

Figure 5.24: Energy spectrum from a forced-dissipative parallel-synchronized coincident-mode two-dimensional Navier–Stokes subgrid simulation with an oblique subgrid, using a grid and subgrid of dimension 63×32, along with the parameters of Figure 5.20. An energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$.

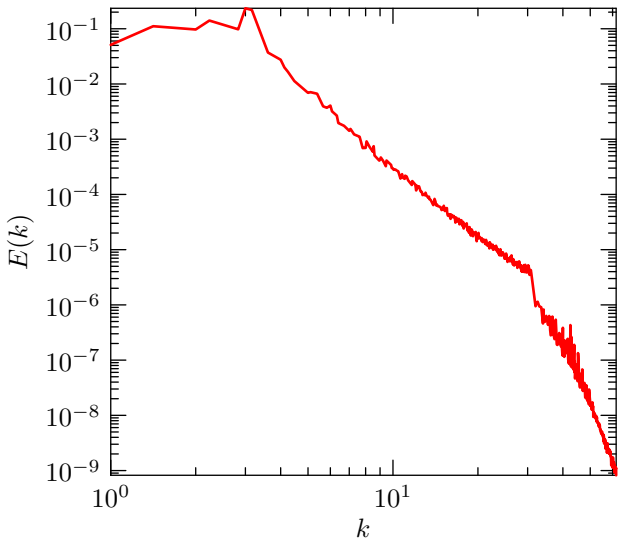

Figure 5.25: Energy spectrum from a forced-dissipative parallel-synchronized coincident-mode two-dimensional Navier–Stokes simulation with radix two, using the parameters of Figure 5.24. An increased energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$.

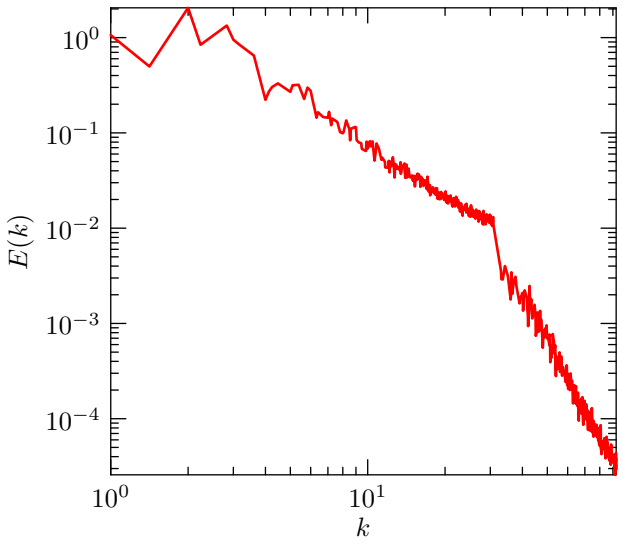

Figure 5.26: Energy spectrum from a forced-dissipative parallel-synchronized coincident-mode two-dimensional Navier–Stokes simulation with radix three, using the parameters of Figure 5.24. A large energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (32, 32\sqrt{2})$.

decimated by removing modes outside of a core set, according to a prescribed radix. While this method is nearly as computationally expensive as a full-resolution simulation and therefore not useful as a subgrid method, the behaviour of the system under this decimation scheme sheds light onto problems that we encounter with decimated systems. As such, the decimated DNS is useful both in developing multispectral methods and in their exposition.

Consider the decimation scheme shown in Figure 5.27, where modes have been removed in an oblique scheme in each dimension, creating a two dimensional radix-two scheme. In this system, the nonlinear source term is calculated on the entire $2m-1 \times 2m-1$ grid, but modes with k_x or k_y greater than $m/2$ are set to zero if either k_x or k_y is odd. This is also implemented with

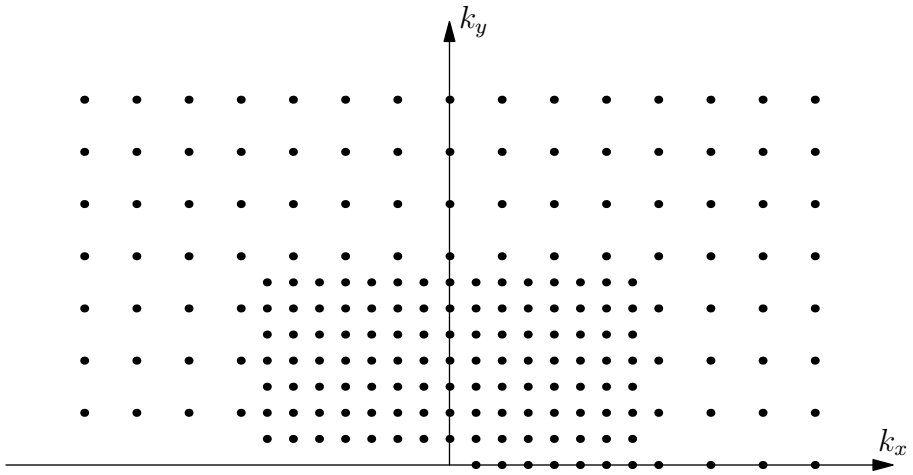

Figure 5.27: Fourier lattice modes in the radix-two decimated DNS. All modes are contained in one variably decimated grid.

modes removed in a checker-board pattern, where modes with modes with k_x or k_y greater than $m/2$ are removed if exactly one of k_x and k_y is odd, as shown in Figure 5.28. This is as an oblique scheme. This decimation can be made yet more severe, keeping all modes in the low-wavenumber box near the

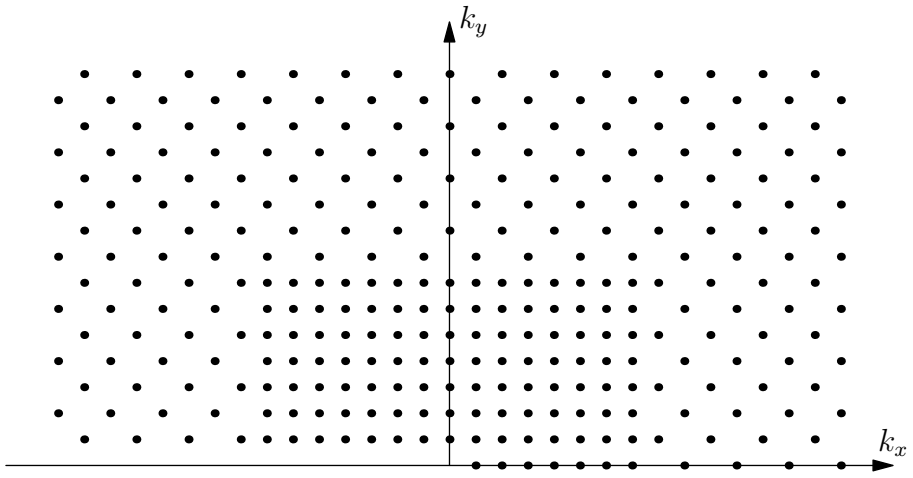

Figure 5.28: Fourier lattice modes in the oblique decimated DNS. All modes are contained in one variably decimated grid.

Fourier origin and removing as many modes as we wish when the wavenumber is higher. For sake of comparison, we also use the radix-three scheme, shown in Figure 5.29. The radix-three decimation scheme is similar to the radix-two decimation scheme, with the outer modes removed if either k_x or k_y are not multiples of three. The decimated systems described above contain all the interactions for all the retained modes, and they conserve the same invariants as does the undecimated system. Because of these two facts, and the fact that we have not removed so many interactions as to destroy the ergodicity of the system, the decimated DNS reproduces the correct equipartition spectrum regardless of the choice of radix, as shown in Figure 5.30. The defect in the forced-dissipative spectrum, which was first encountered with coincident-mode synchronization, is also present, as shown in Figure 5.31 and Figure 5.32.

Since the decimated DNS method uses only one grid, the defect in the forced-dissipative spectrum cannot be due to the choice of synchronization method, and is therefore due to decimation.

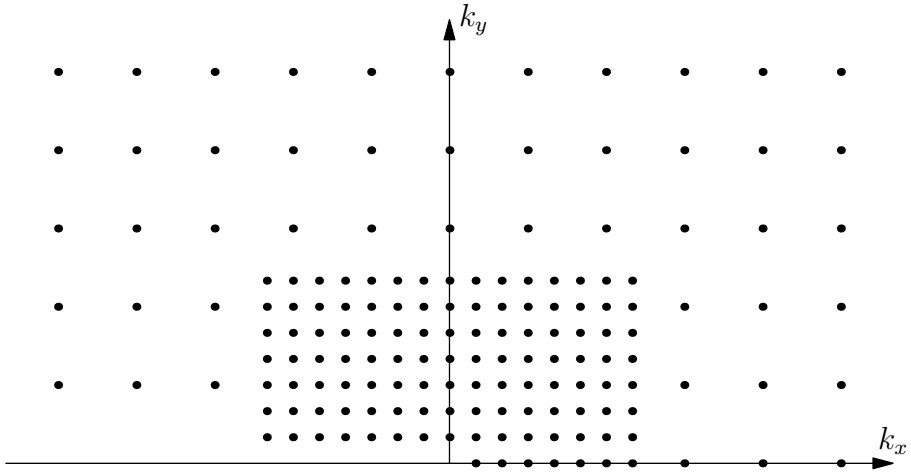

Figure 5.29: Fourier lattice modes in the radix-three decimated DNS. All modes are contained in one variably decimated grid.

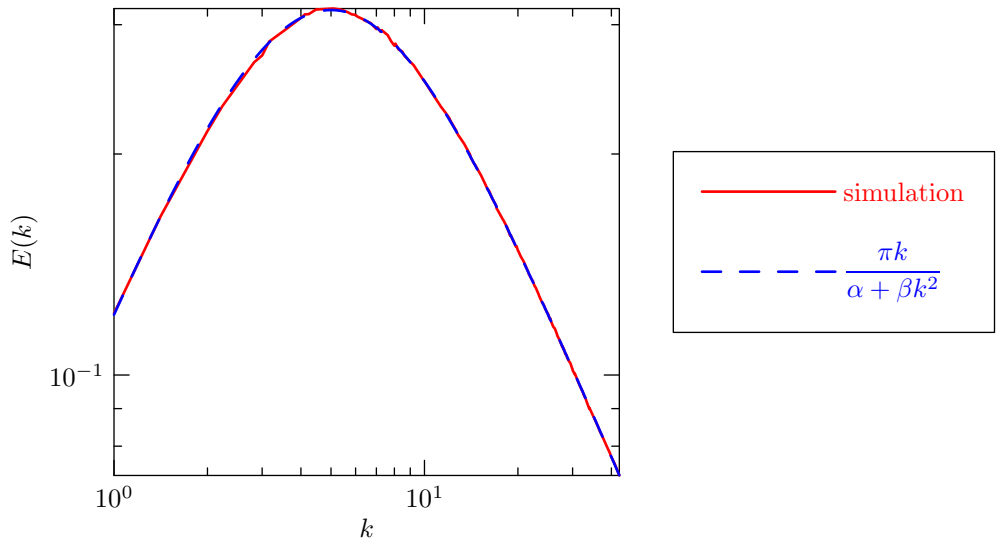

Figure 5.30: Energy spectrum from an inviscid unforced serially synchronized oblique decimated DNS simulation, with $\alpha = 25$ and $\beta = 1$, using a grid and subgrid of dimension 63×32. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$.

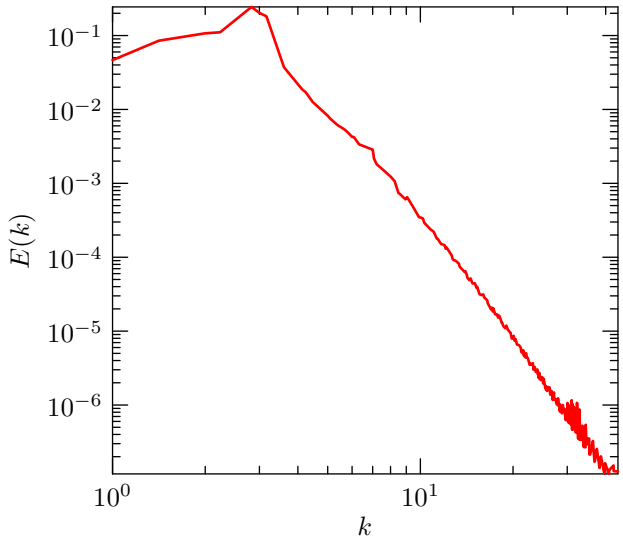

Figure 5.31: Energy spectrum from a forced-dissipative serially synchronized oblique decimated DNS simulation, using the parameters of Figure 5.20. An energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$.

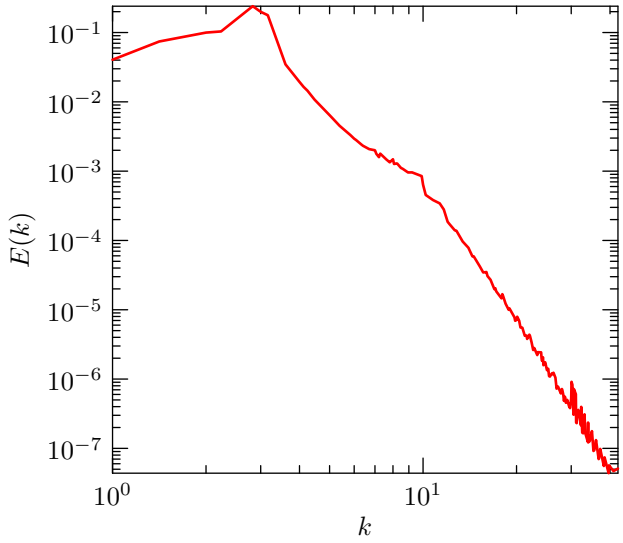

Figure 5.32: Energy spectrum from a forced-dissipative serially synchronized radix-two decimated DNS simulation, using the parameters of Figure 5.20. An increased energy pile-up at the grid/subgrid interface is observed. The grid/subgrid boundary occurs at $k \in (7, 7\sqrt{2})$.

A DNS simulation involves evolving modes on the full-resolution grid G_{DNS}. The nonlinear interactions are all triadic interactions included in the set

$$G_{\mathrm{DNS}} \times G_{\mathrm{DNS}} \to G_{\mathrm{DNS}}. \tag{5.20}$$

Moreover, the decimated DNS contains interactions that are not available with coincident-mode synchronization. Let G be the set of modes with $\boldsymbol{k} \in [-m+1, m-1] \times [-m+1, m-1]$, and SG denote the set containing all other modes for the decimated DNS. The union of the grid and the subgrid are a strict subset of the modes available in a DNS simulation, i.e. $G \cup SG \subsetneq G_{\mathrm{DNS}}$. The nonlinear interaction of the decimated DNS evolution equation contains all triadic interactions in the set

$$(G \cup SG) \times (G \cup SG) \to (G \cup SG), \tag{5.21}$$

which is strictly fewer than the nonlinear interactions in the DNS simulation. The coincident-mode multispectral method has even fewer interactions. Let G_c, G_{nc}, SG_c, and SG_{nc}, be as above. Note that G_c and SG_c have modes with identical wave vectors. The decimated DNS and multispectral modes are related by the fact that $G = G_c \cup G_{\mathrm{nc}}$ and $SG = SG_{\mathrm{nc}}$. The coincident-mode multispectral method includes the nonlinear triad contained in interactions in

$$(G \times G \to G) \cup [(G_c \cup SG_{\mathrm{nc}}) \times (G_c \cup SG_{\mathrm{nc}}) \to (G_c \cup SG_{\mathrm{nc}})], \tag{5.22}$$

which is a strict subset of the interactions present in Equation (5.21). This reduction of the nonlinear interaction exacerbates the already-depressed nonlinear energy/enstrophy transfer of the decimated DNS method, which is the cause of the artificially high energy spectrum at wavenumbers lower than the grid/subgrid boundary.

Since the decimated DNS method already shows an energy bottleneck due to its inability to transfer energy effectively from low to high wavenumbers,

there is no reason to expect that the coincident-mode synchronization scheme will be able to reproduce the energy transfer of the undecimated system. This will be an issue when applying the decimated DNS or coincident-mode synchronization method to any system where energy is injected at large scales and transferred by the nonlinear terms to small scales where it is dissipated by viscosity. For inviscid, unforced systems, the net energy flux is zero, so the system is still able to relax to the correct equipartition spectrum despite this bottleneck.

Another problem with the decimated DNS and coincident-mode synchronization is that the energy and enstrophy from the removed modes remains unaccounted, so the system will have less energy and enstrophy than the comparable full-resolution simulation. In light of these deficiencies, it is preferable to bin subgrid modes, and synchronize the grid and subgrid over this bin, instead of considering the behaviour of only those modes on the subgrid that coincide exactly with modes on the grid.

5.C.2 Binned Mode Synchronization

The mode-centric multispectral method and the decimated DNS method both remove modes from the system. These methods reproduce the inviscid, unforced spectra of the original model. However, the more modes that are removed, the worse the system is able to reproduce the behaviour of the undecimated system when forcing and viscosity are present.

Binned mode synchronization accounts for these missing modes and interactions by treating the subgrid as evolving bins of modes. This was the method applied to shell models of turbulence. Moreover, binning modes is the starting

point for spectral reduction. In order to synchronize the grid properly with the subgrid, the modes on the grid must be synchronized with the corresponding bins on the subgrid. In order to calculate the nonlinear source term on the subgrid efficiently, the subgrid bins are chosen to lie on a uniform rectangular grid. This places a restriction on the subgrid geometry, as we require that grid modes be associated with a unique subgrid bin. This is straightforward when the subgrid is decimated according to a radix-three decimation scheme, but not possible at all with oblique or radix-two decimation schemes. The radix-three grid/subgrid geometry is shown in Figure 5.33.

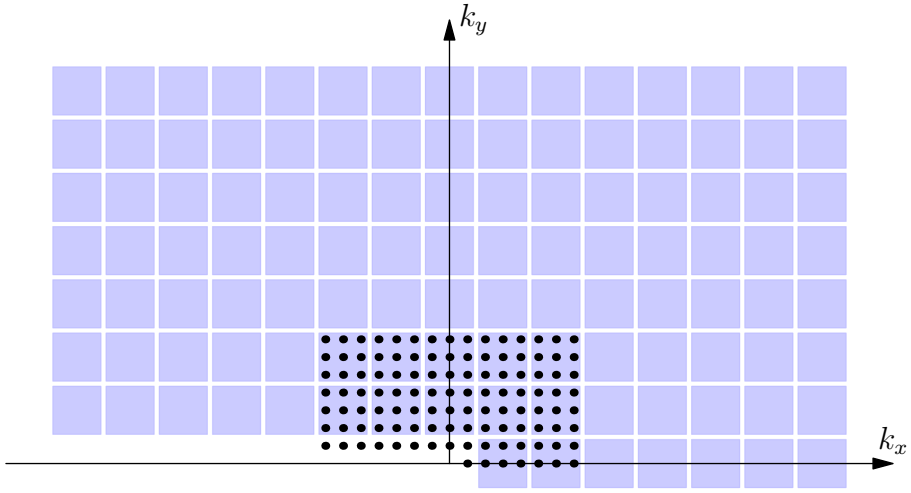

Figure 5.33: Grid/subgrid with radix-three decimation on the subgrid. Grid modes are represented by black dots, and subgrid bins are represented by blue squares.

Synchronizing modes with bins maps all modal interactions to bin interactions. Suppose that modes $\omega_{\boldsymbol{k}}$, $\omega_{\boldsymbol{p}}$, and $\omega_{\boldsymbol{q}}$ comprise an interacting triad in the undecimated system. Let $\Omega_{\boldsymbol{K}}$, $\Omega_{\boldsymbol{P}}$, and $\Omega_{\boldsymbol{Q}}$ be the corresponding bins. Since $\omega_{\boldsymbol{k}}$, $\omega_{\boldsymbol{p}}$, and $\omega_{\boldsymbol{q}}$ interact in the original system, the bins $\Omega_{\boldsymbol{K}}$, $\Omega_{\boldsymbol{P}}$, and $\Omega_{\boldsymbol{Q}}$ interact in the decimated system. If all three modes are on the grid, then

the interaction is included in the grid nonlinear term. If all the modes are on the subgrid, then the interaction is included in the subgrid nonlinear term. If only some of the modes are on the grid, say $\omega_{\boldsymbol{k}}$, then the synchronization operator will copy the effect of the interaction of $\omega_{\boldsymbol{k}}$, $\omega_{\boldsymbol{p}}$, and $\omega_{\boldsymbol{q}}$ onto $\omega_{\boldsymbol{k}}$, which includes the interaction in the grid/subgrid system without having to calculate $\omega_{\boldsymbol{p}}$ and $\omega_{\boldsymbol{q}}$ explicitly.

Since the subgrid consists of bins of modes, it is natural to apply pseudospectral reduction. As with the coincident-mode synchronization scheme, double counting of interactions is avoided by eliminating certain nonlinear couplings on the subgrid.

Binning modes on the subgrid changes the definition of the invariants. We account for the fact that each bin represents a number of modes by including a weighting factor in the definition of energy and enstrophy. The energy is defined as

$$\begin{aligned} E &= \frac{1}{2}\sum_{\omega_{\boldsymbol{k}}\in G}\frac{|\omega_{\boldsymbol{k}}|^2}{k^2} + \frac{1}{2}\lambda\sum_{\substack{\Omega_{\boldsymbol{K}}\in SG\\ \Omega_{\boldsymbol{K}}\notin G}}\frac{|\Omega_{\boldsymbol{K}}|^2}{K^2}, \\ Z &= \frac{1}{2}\sum_{\omega_{\boldsymbol{k}}\in G}|\omega_{\boldsymbol{k}}|^2 + \frac{1}{2}\lambda\sum_{\substack{\Omega_{\boldsymbol{K}}\in SG\\ \Omega_{\boldsymbol{K}}\notin G}}|\Omega_{\boldsymbol{K}}|^2, \end{aligned} \tag{5.23}$$

where $\lambda = 9$ is the square of the decimation radix. Unlike in the coincident-mode method, decimating with the binned multispectral method does not decrease the energy or enstrophy of the system.

Let $\{\omega_{\boldsymbol{k}_i}\}_{i=0}^{8}$ be grid modes corresponding to the subgrid bin $\Omega_{\boldsymbol{K}}$. The enstrophy for these modes is equal to

$$\frac{1}{2}\sum_{i\,=0}^{8}|\omega_{\boldsymbol{k}_i}|^2 \tag{5.24}$$

and the energy is

$$\frac{1}{2}\sum_{i=0}^{8}\frac{|\omega_{\boldsymbol{k}_i}|^2}{k_i^2}. \tag{5.25}$$

The subgrid bin has enstrophy

$$\frac{1}{2}\lambda\,|\Omega_{\boldsymbol{K}}|^2 \tag{5.26}$$

and energy

$$\frac{1}{2}\frac{\lambda\,|\Omega_{\boldsymbol{K}}|^2}{K^2}. \tag{5.27}$$

We require that the synchronization operator be able to make these energies and enstrophies equal to each other without changing the distribution of energy on the grid modes. If we take $\boldsymbol{K}$ to be the central wave number, this is not generally possible. Suppose, for example, that $\omega_{\boldsymbol{k}} = 1$ and all the other grid modes are zero, and that $|\boldsymbol{k}| \neq |\boldsymbol{K}|$. Enstrophy conservation requires that $\sum |\omega_{\boldsymbol{k}}|^2 = \lambda\,|\Omega_{\boldsymbol{K}}|^2$, but, since $|\boldsymbol{k}| \neq |\boldsymbol{K}|$, the grid modes and the subgrid bin cannot have the same energy.

This is not a problem with mode-centric synchronization, where the only modes that are synchronized have the same wavenumber. For binned synchronization, this problem can be dealt with the help of the following theorem.

Theorem 5.3: *Consider a system consisting of the modes* $\{\omega_{\boldsymbol{k}}\}, \boldsymbol{k} \in D$ *for some set* D*, with*

$$\frac{\partial \boldsymbol{\omega}_{\boldsymbol{k}}}{\partial t} = \sum_{\substack{\boldsymbol{p},\boldsymbol{q}\in D \\ \boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{0}}} \frac{\epsilon_{\boldsymbol{kpq}}}{f(\boldsymbol{q})}\boldsymbol{\omega}_{\boldsymbol{p}}^*\boldsymbol{\omega}_{\boldsymbol{q}}^* \tag{5.28}$$

where $f : \mathbb{R}^n \to \mathbb{R}$*, and*

$$\epsilon_{\boldsymbol{kpq}} = -\epsilon_{\boldsymbol{qpk}}, \tag{5.29}$$

then the system conserves the quantity

$$\frac{1}{2}\sum_{\boldsymbol{k}\in D}\frac{|\boldsymbol{\omega}_{\boldsymbol{k}}|^2}{f(\boldsymbol{k})}. \tag{5.30}$$

Proof. The rate of change of the quantity (5.30) is

$$\begin{aligned}\operatorname{Re}\sum_{\boldsymbol{k}\in D}\frac{\omega_{\boldsymbol{k}}^*}{f(\boldsymbol{k})}\frac{\partial\omega_{\boldsymbol{k}}}{\partial t} &= \operatorname{Re}\sum_{\boldsymbol{k}\in D}\frac{\omega_{\boldsymbol{k}}^*}{f(\boldsymbol{k})}\sum_{\substack{\boldsymbol{p},\boldsymbol{q}\in D\\ \boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{0}}}\frac{\epsilon_{\boldsymbol{kpq}}}{f(\boldsymbol{q})}\boldsymbol{\omega}_{\boldsymbol{p}}^*\boldsymbol{\omega}_{\boldsymbol{q}}^* \\ &= \operatorname{Re}\sum_{\substack{\boldsymbol{k},\boldsymbol{p},\boldsymbol{q}\in D\\ \boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{0}}}\frac{\epsilon_{\boldsymbol{kpq}}}{f(\boldsymbol{k})f(\boldsymbol{q})}\boldsymbol{\omega}_{\boldsymbol{k}}^*\boldsymbol{\omega}_{\boldsymbol{p}}^*\boldsymbol{\omega}_{\boldsymbol{q}}^* \\ &= 0\end{aligned} \tag{5.31}$$

since $\epsilon_{\boldsymbol{kpq}}$ is antisymmetric under exchange of the first and last indices. □

This is a generalization of Theorem 1.1, which is recovered in the case where $f(\boldsymbol{k}) = k^2$. Moreover, the quantity is conserved in a detailed fashion.

Theorem 5.3 gives us the freedom we need to synchronize the grid and the subgrid, and we are presented with two courses of action: we must either modify the wavenumbers for the modes on the grid, or we must modify the wavenumbers on the subgrid.

Effective subgrid wavenumbers

It is necessary to modify wavenumbers on either the grid or the subgrid in order to conserve energy and enstrophy simultaneously. It is simple to guarantee that the subgrid bin have the same energy and enstrophy as the corresponding grid modes by defining the effective subgrid wavenumber via

$$K_{\text{eff}}^2 \doteq \frac{E_{\text{bin}}}{Z_{\text{bin}}}. \tag{5.32}$$

At the beginning of each parallel-synchronized time step, we can set K_{eff}^2 for each bin using its current energy and enstrophy.

The mode amplitudes $\omega_i(t+\tau)$ for grid mode i at time $t+\tau$ is given by

$$\left|\omega_i(t+\tau)\right|^2 = \left|\widetilde{\omega}_i(t+\tau)\right|^2 + \frac{\Delta_{t\to t+\tau}\left|\Omega\right|^2}{\left|\Omega(t)\right|^2}\left|\omega_i(t)\right|^2 \tag{5.33}$$

where $\Omega(t)$ is the bin amplitude at the start beginning of the time step, $\Delta_{t\to t+\tau}\left|\Omega\right|^2$ is the change in the absolute value of the amplitude of the subgrid bin due to the subgrid evolution equation, and $\widetilde{\omega}_i(t+\tau)$ is the vorticity of the subgrid mode due the grid evolution equation. At the beginning of the time step, the enstrophy of the grid modes is equal to the enstrophy of the corresponding subgrid bin, i.e.

$$\frac{1}{2}\sum_i \left|\omega_i\right|^2 = \frac{1}{2}\left|\Omega\right|^2. \tag{5.34}$$

The energy of the grid modes is given by

$$\frac{1}{2}\sum_i \frac{\left|\omega_i\right|^2}{k_i^2}. \tag{5.35}$$

Insisting that the grid modes have the same energy as the corresponding subgrid bin implies that

$$E_{\text{bin}} = \frac{1}{2}\sum_i \frac{\left|\omega_i\right|^2}{k_i^2}, \tag{5.36}$$

which allows us to determine K_{eff}^2 via equation (5.32).

Theorem 5.4: *Equation* (5.33) *conserves energy and enstrophy locally.*

Proof. Let $\sum_i \left|\omega_i\right|^2 = 9\left|\Omega\right|^2$ and $\sum_i \frac{\left|\omega_i^2\right|}{k_i^2} = 9\frac{\left|\Omega^2\right|}{K^2}$ at time t. Then, advance w_i in time, so that

$$\omega_i \to \widetilde{\omega}_i, \qquad \Omega \to \widetilde{\Omega}. \tag{5.37}$$

Then, the new enstrophy is

$$\begin{aligned} 2Z(t+\tau) &= \sum_i \left|\omega_i(t+\tau)\right|^2 \\ &= \sum_i \left|\widetilde{\omega}_i(t+\tau)\right|^2 + \frac{\Delta_{t\to t+\tau}\left(\left|\Omega^2\right|\right)}{\left|\Omega^2(t)\right|}\left|\omega_i(t)\right|^2 \\ &= \sum_i \left|\widetilde{\omega}_i(t+\tau)\right|^2 + 9\Delta_{t\to t+\tau}\left(\left|\Omega^2\right|\right)\frac{\sum_i \left|\omega_i(t)\right|^2}{9\left|\Omega^2(t)\right|}. \end{aligned} \tag{5.38}$$

But $\frac{\sum_i \left|\omega_i(t)^2\right|}{9|\Omega(t)|^2} = 1$, so

$$\begin{aligned} 2Z(t+\tau) &= \sum_i \left|\widetilde{\omega}_i(t+\tau)^2\right| + 9\Delta_{t\to t+\tau}\left(\left|\Omega^2\right|\right) \\ &= \sum_i |\omega_i(t)|^2 + \sum_i \Delta_{t\to t+\tau}\left(\left|\omega_i^2\right|\right) + 9\Delta_{t\to t+\tau}\left(\left|\Omega^2\right|\right), \end{aligned} \tag{5.39}$$

which includes all the contributions from both the grid and the subgrid. Since the subgrid and the grid conserve energy on their own, any change from synchronization is balanced by a change of the opposite sign elsewhere in the system.

In terms of energy conservation, one has

$$\begin{aligned} 2E(t+\tau) &= \sum_i \frac{|\omega_i(t+\tau)|^2}{k_i^2} \\ &= \sum_i \frac{|\omega_i(t)|^2}{k_i^2} + \sum_i \frac{\Delta_{t\to t+\tau}\left(|\omega_i|^2\right)}{k_i^2} + 9\Delta_{t\to t+\tau}\left(|\Omega|^2\right)\frac{\sum_i \frac{|\omega_i(t)|^2}{k_i^2}}{9\,|\Omega(t)|^2}, \end{aligned} \tag{5.40}$$

but

$$\frac{\sum_i \frac{|\omega_i(t)|^2}{k_i^2}}{9\,|\Omega(t)|^2} = \frac{1}{K^2}, \tag{5.41}$$

so

$$2E(t+\tau) = \sum_i \frac{|\omega_i(t)|^2}{k_i^2} + \sum_i \Delta_{t\to t+\tau}\left(\frac{|\omega_i|^2}{k_i^2}\right) + 9\Delta_{t\to t+\tau}\left(\frac{|\Omega|^2}{K^2}\right), \tag{5.42}$$

which captures the changes in energy from the two grids. Again, conservation on each grid independently implies that this synchronization method conserves energy for the grids in conjunction. □

Figure 5.34 shows the energy spectrum of an inviscid, unforced binned subgrid wavenumber simulation. Notice that the grid boundary is visible, and that the spectrum on the subgrid follows a different curve than the predicted

equipartition spectrum. Upon increasing the source term on the subgrid, based on the assumption of non-correlated phase, this divergence becomes even more pronounced, as shown in Figure 5.35. However, *decreasing* the subgrid source term moves the spectrum closer to the theoretical prediction, as shown in Figure 5.36.

Decreasing the subgrid source term corrects for the shift in the equipartition spectrum on the subgrid. For shell model turbulence, rescaling the grid or subgrid source term had no effect on the energy spectrum of unforced inviscid simulations, and the reason why this plays an important rôle in two-dimensional Navier–Stokes turbulence is not entirely clear, and may be due to inaccuracies in the subgrid evolution equation stemming from approximations made in its derivation.

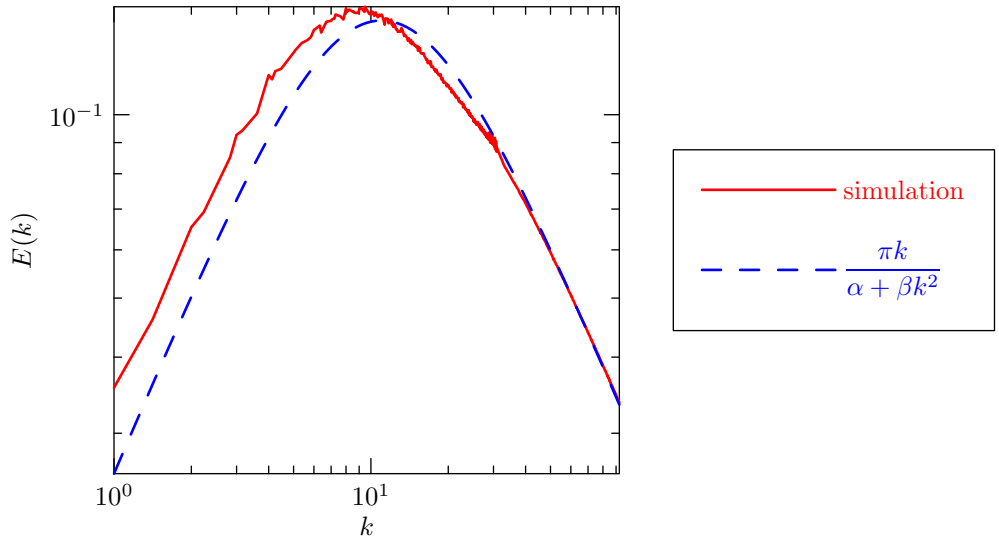

Figure 5.34: Energy spectrum from an unforced inviscid binned subgrid wavenumber simulation, with $\alpha = 121$, $\beta = 1$. The grid and subgrid have dimension 63×32. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$.

Scaling the subgrid source term continues to be important when we consider forced-dissipative simulations using a binned subgrid. Figure 5.37 shows the

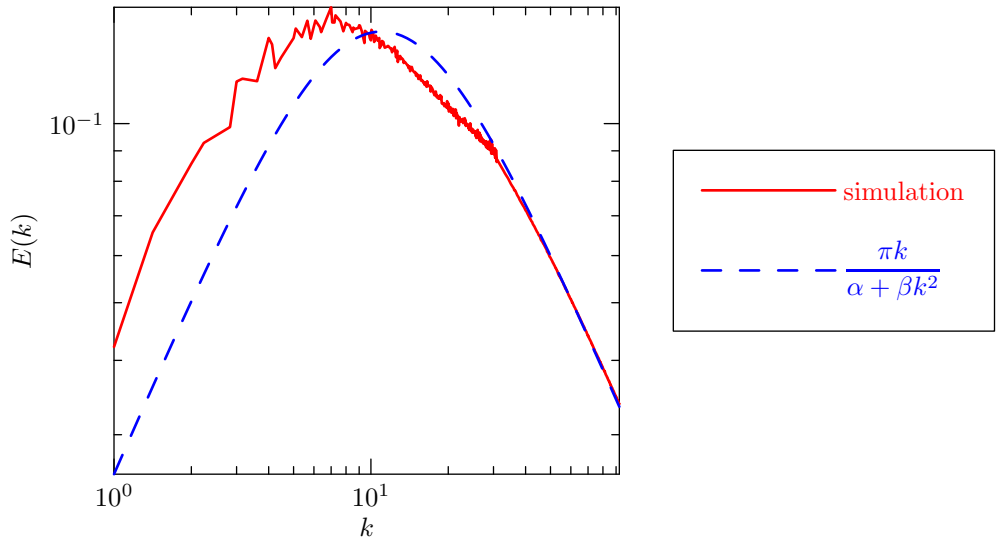

Figure 5.35: Energy spectrum from an unforced inviscid binned subgrid wavenumber simulation with increased subgrid nonlinearity, using the parameters of Figure 5.34. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$.

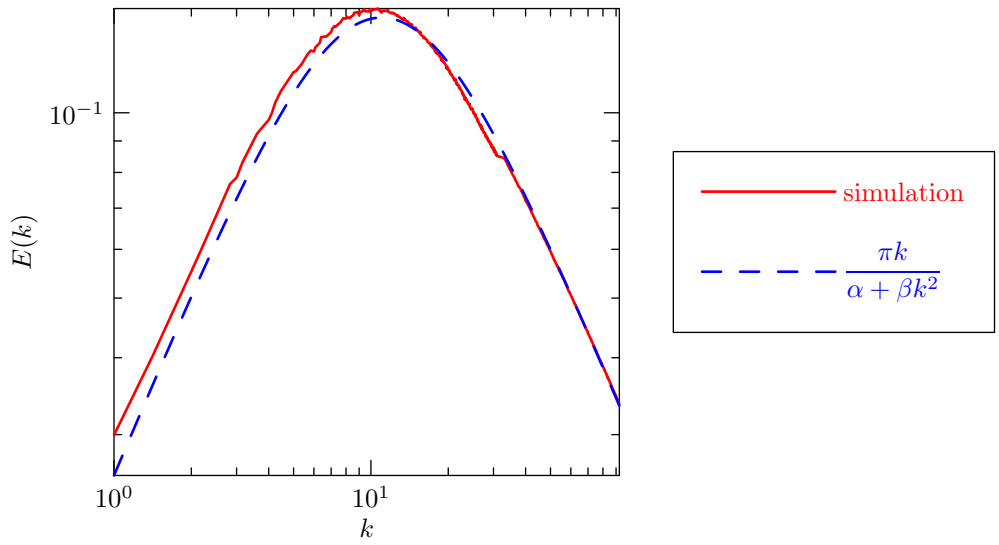

Figure 5.36: Energy spectrum from an unforced inviscid binned subgrid wavenumber simulation with decreased subgrid nonlinearity, using the parameters of Figure 5.34. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$.

energy spectrum of a forced-dissipative simulation with a grid and subgrid of equal size. The visible part of the subgrid is less energetic than the full-resolution DNS simulation would suggest. Upon increasing the subgrid source term, this divergence is exacerbated, as shown in Figure 5.38, but is partly corrected when the subgrid source term is diminished, as shown in Figure 5.39.

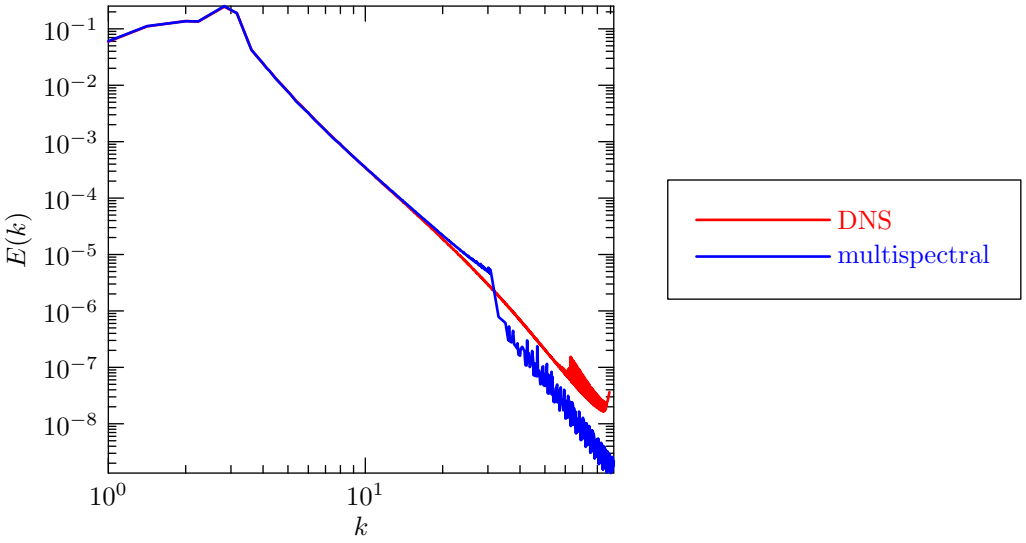

Figure 5.37: Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation compared to a full-resolution DNS simulation, using the parameters of Figure 5.20. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$.

The grid and the subgrid both have square geometries, and this lack of isotropy may be preventing the system from relaxing to the energy spectrum of the full-resolution simulation. This problem can be partially solved by eliminating modes on the grid in a circular pattern and synchronizing only the remaining grid modes, so that the grid is effectively circular. In addition to requiring a circular boundary, we also reduce the size of the grid. Since this allows for more interactions on the subgrid, it is more able to relax to the correct energy spectrum. Figure 5.40 shows the energy spectrum of such a simulation

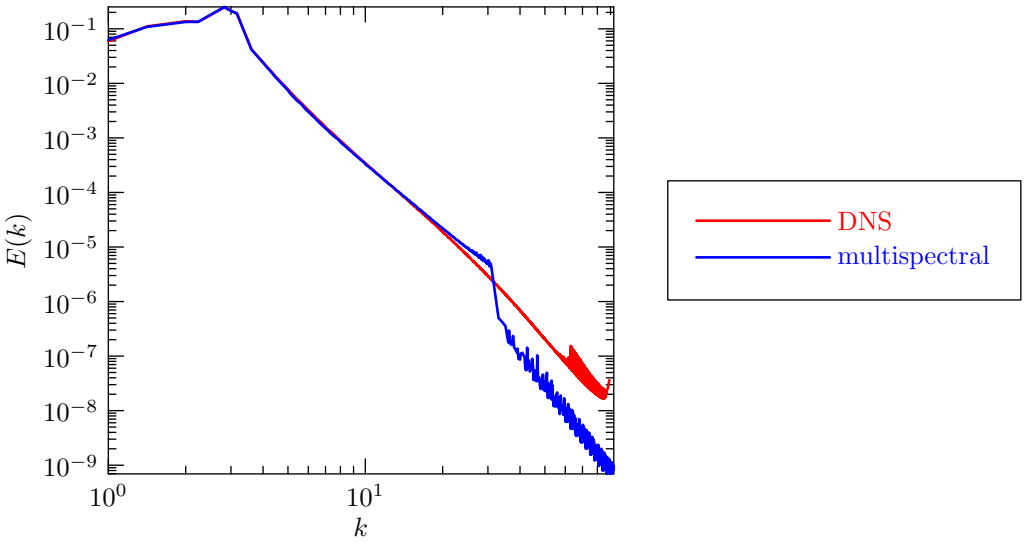

Figure 5.38: Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with increased subgrid source term compared to a full-resolution DNS simulation, using the parameters of Figure 5.20. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$.

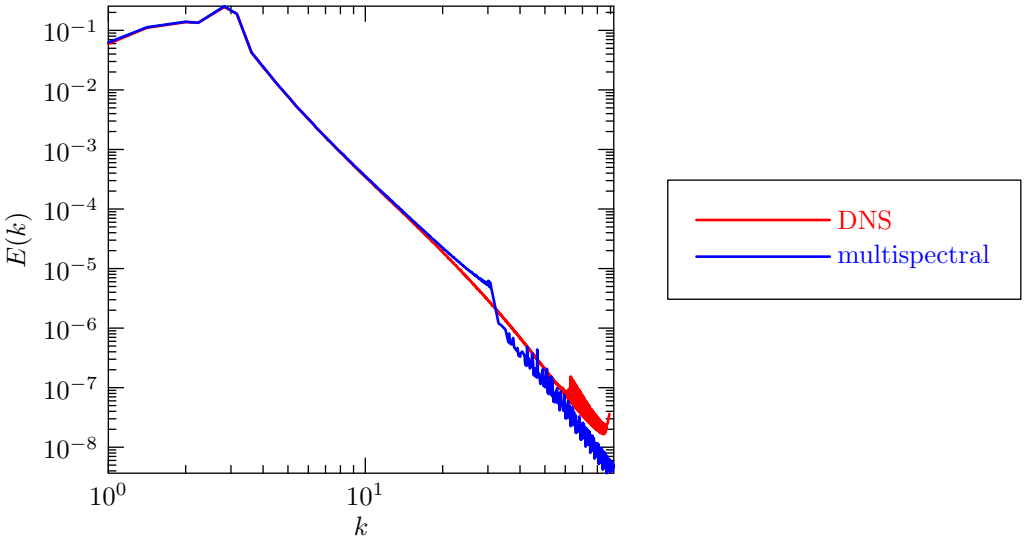

Figure 5.39: Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with decreased subgrid source term compared to a full-resolution DNS simulation, using the parameters of Figure 5.20. The grid/subgrid boundary occurs at $k \in (31, 31\sqrt{2})$.

with normal subgrid source. As before, increasing the subgrid nonlinear source increases the difference between the energy spectra of the multispectral simulation and the full-resolution DNS simulation, which is shown in Figure 5.41. Reducing the subgrid source term moves the energy spectrum more in line with the DNS energy spectrum, with only relatively minor oscillations at the grid/subgrid interface.

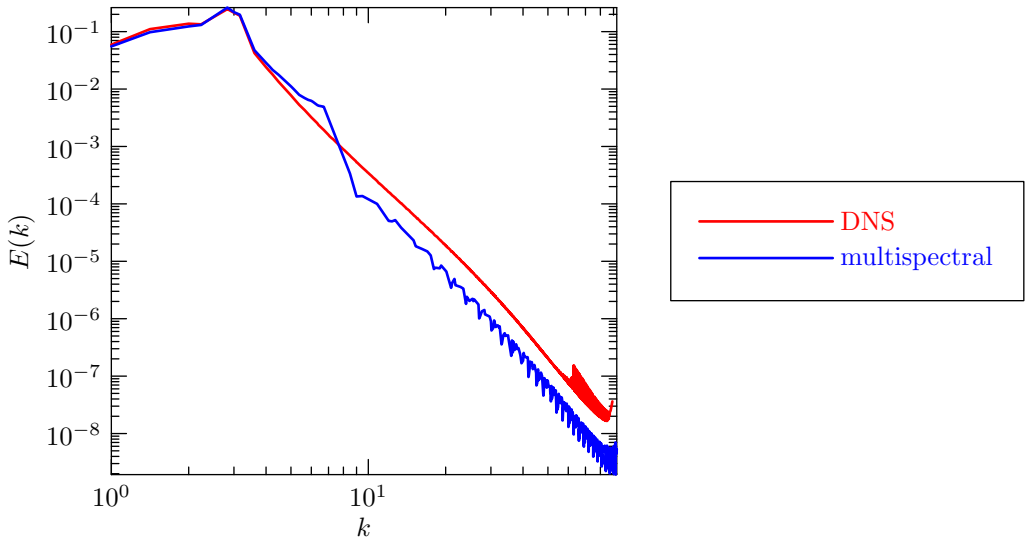

Figure 5.40: Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with a grid of size 15×8, a subgrid of size 63×32, and a circular grid boundary compared to a full-resolution DNS simulation. Other parameters are as in Figure 5.20. The grid/subgrid boundary occurs at $k = 7$.

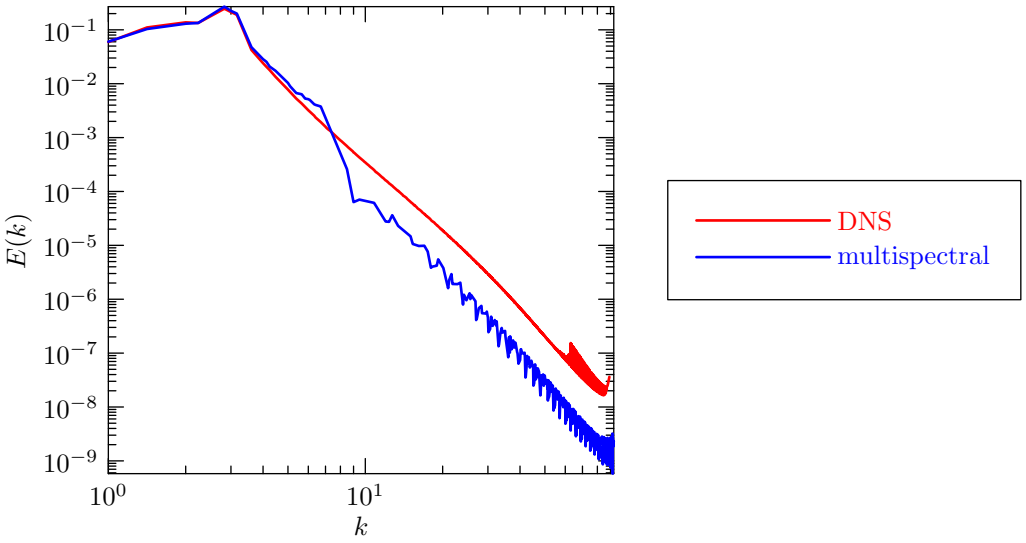

Figure 5.41: Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with a grid of size 15×8, a subgrid of size 63×32, a circular grid boundary, and an increased subgrid source compared to a full-resolution DNS simulation. Other parameters are as in Figure 5.20. The grid/subgrid boundary occurs at $k = 7$.

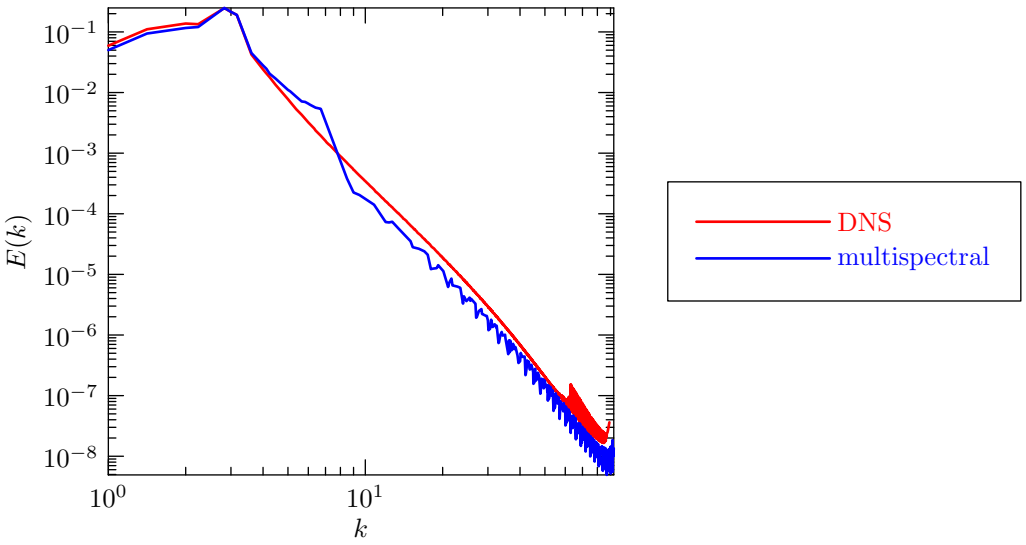

Figure 5.42: Energy spectrum from a forced-dissipative binned subgrid wavenumber simulation with a grid of size 15×8, a subgrid of size 63×32, a circular grid boundary, and a decreased subgrid source compared to a full-resolution DNS simulation. Other parameters are as in Figure 5.20. The grid/subgrid boundary occurs at $k = 7$.

Chapter 6

Conclusion

Wherein the results described in this monograph are summarized and open problems are considered.

Applied mathematicians face two fundamental problems: first, one must determine a mathematical model of the system under consideration and, second, one must use this model to help understand the system. In some sense, these two activities are in conflict with each other. On one hand, one would prefer a model which includes as many features as possible, but such complicated models are rarely solvable, and a simplified system can yield more insight, so long as it is still complex enough to capture the behaviour of the system.

6.A Summary of Results

In this monograph, I considered two models of fluid motion, namely the incompressible Navier–Stokes equations, a complicated model which is difficult

to solve, and shell models of turbulence, which are less complicated and easier to solve. I approached these systems using a variety of techniques.

Shell models of turbulence are by far the simpler type of models presented in this monograph; they are truly toy models of turbulence, which, while simpler to understand than their older and greater sibling the Navier–Stokes equations, solutions to shell models of turbulence exhibit complicated behaviour which is not entirely understood. In this work, I studied the effect of resolution on shell models of turbulence. In Chapter 2, I showed that the GOY, DN, and Sabra models of turbulence reproduced the same continuum shell model under the limit of infinite resolution. This continuum model reproduces aspects of classical Kolmogorov theory, i.e. it lacks anomalous scaling. This is suggestive that anomalous scaling in Navier–Stokes may also be due to an energy cascade that is non-local in Fourier space.

The Navier–Stokes equations are generally considered to provide a good description of the motion of Newtonian fluids. They seem straightforward when written down, with individual terms representing clearly defined effects which, individually, are not difficult to understand. However, solutions to the Navier–Stokes equations are available under very few circumstances, and, at high Reynolds number, their numerical solution is very difficult.

One method for numerically solving nonlinear partial differential equations such as the Navier–Stokes equations is the pseudospectral method, where most of the system is solved in Fourier space, except for the nonlinear term, which is solved in physical space. This is done for reasons of efficiency; since the nonlinearity is quadratic, it is transformed into a convolution in Fourier space. It is much faster to calculate this using the convolution theorem and FFTs. These convolutions require dealiasing, which was the subject of Chapter 3. With

the goal of producing a complete description of finite-length convolutions and their calculation, I considered convolutions on centered and non-centered data; nullary, unary, binary, and n-ary convolutions; and convolutions in one or many dimensions. In this work, I considered two methods for dealiasing such convolutions, namely phase-shift dealiasing and zero-padding. Moreover, the technique of implicit zero-padding was demonstrated to dealias multi-dimensional convolutions in less time and using less memory than explicit zero padding.

Implicit padding significantly decreases the difficulty of performing pseudospectral simulations of highly turbulent systems. Unfortunately, these systems are so complicated that much more must be done: even with the efficiencies gained from implicit padding, the vastness of the problem completely out-strips our computational ability. This can be addressed by considering a decimation scheme, which reduces the number of degrees of freedom of the problem by using some averaging technique. One such decimation scheme, spectral reduction, is based on evolving Fourier modes in groups known as bins. In the first part of Chapter 4, spectral reduction is applied to shell models of turbulence. It is shown that the GOY model reduces to the DN model, which, on further decimation, is a fixed point. Spectral reduction is based on the assumption that the individual modes may be approximated as constant over a bin. We attempt to improve on this approximation, accounting for the variation in amplitudes over a bin by interpolating the neighbouring bin amplitudes; however, this method is shown to be unstable. Spectral reduction is also applied to two-dimensional incompressible Navier–Stokes turbulence. Until recently, this required the direct calculation of a modified convolution, but this has now been shown to be amenable to FFT-based calculation, resulting in a dramatic reduction in computational effort.

The major drawback of spectral reduction is the requirement that the decimation be performed uniformly in Fourier space. In other words, the decimation must produce bins that represent the same number of modes in the original system. The problem becomes clear upon performing inviscid, unforced simulations of mixing dynamics; such simulations should move towards a statistical equipartition of a linear combination of the conserved quantities, and this equipartition should be an equipartition over the original modes. The decimated system will produce an equipartition over the bins; these two states are equivalent only when the decimation is uniform.

Performing a uniform spectral reduction reduces the information at both the small scales and the large scales alike, which can be a significant problem. The main focus in this monograph is the multispectral method, a multigrid-inspired numerical technique that uses a hierarchy of spectrally reduced grids to perform simulations of time-dependent problems.

The multispectral method was first developed for shell models of turbulence, and was applied in this work to the more difficult problem of two-dimensional incompressible Navier–Stokes flows. Various grid geometries and synchronization methods were examined, allowing the grids to be evolved in time and kept in sync with each other, while conserving the invariants of the system, i.e. energy and enstrophy. While shell models require that the subgrid evolution source term be increased in order to reproduce the energy spectrum of forced-dissipative turbulence, the multispectral Navier–Stokes simulations required that the source term for the subgrid evolution equation be decreased. This suggests that higher-order corrections to spectral reduction may perhaps be required in order to explain the difference in time scales between the grid and the subgrid. Moreover, it was found necessary to extend the subgrid,

which may indicate that discrete effects have a more prominent effect for the multispectral method than for direct simulation.

6.B Future Work

Implicitly padded convolutions are faster and require less memory than conventional techniques for dealiasing convolutions, and we expect that they will become a standard tool in many areas. Implicitly padded convolutions have been developed for non-centered, non-Hermitian data in one, two, and three dimensions; centered Hermitian data in one, two, and three dimensions; and centered, Hermitian ternary convolutions in one and two dimensions. All of these algorithms have been designed for independent inputs, and there are important special cases that can be optimized, such as the convolution of the array $f * f$; since the two input arrays are identical, one need only transform f once, allowing $f * f$ to be calculated using two Fourier transforms instead of three.

In addition to optimizing the special cases mentioned above, the calculation of n-ary convolutions merits more attention, and it may be possible to create an efficient code that can calculate convolutions or arbitrary order using implicit padding. It would also be worthwhile to determine the magnitude of error introduced by calculating such a convolution via binary convolutions, particularly if this technique is used for pseudospectral calculations of compressible Navier–Stokes turbulence.

As mentioned in section 3.D, it may be possible to parallelize implicitly padded convolutions efficiently. This could be accomplished by taking advantage of the fact that the implicitly padded Fourier transform divides the data

at the highest level, as do conventional FFTs. One could divide the problem into smaller pieces by increasing this division, and it may be possible to mitigate the high communication costs incurred by conventional explicitly padded convolution techniques.

The advent of implicitly padded convolutions presents an opportunity to create a new, more efficient three-dimensional pseudospectral Navier–Stokes solver. Using `TRIAD` for time stepping and `FFTW++` to compute the convolution, this should be a relatively straightforward exercise. The resulting code should be more efficient than existing serial pseudospectral codes.

In the multispectral method, the subgrid evolution equation had to be rescaled for both shell models of turbulence and the two-dimensional Navier–Stokes equations. This suggests that it would be worthwhile to consider higher-order approximations to spectral reduction. This may be in the form of the inclusion of phase information. Perhaps some more stable method for composing bins from modes of different amplitudes might be developed. With the advent of pseudospectral reduction, which allows the modified spectrally reduced convolution to be calculated via FFTs, the technique is again competitive and deserves renewed consideration.

Phase information may play an important rôle in the application of the multispectral method to two-dimensional Navier–Stokes flow. The synchronization operators were designed to conserve energy and enstrophy, which depend only on the modal amplitude, and not the phase. It would be worthwhile to consider synchronization operators that are able to transfer phase information between the grid and subgrid, as the importance of phase synchronization is not well understood at this time.

The two-dimensional multispectral method could also be studied at higher

resolution, as the removal of grid-mode interactions from the subgrid may exacerbate discrete effects. It is also interesting to consider what effect the choice of time stepping has on the multispectral method, and whether synchronizing only between time steps causes a drop in accuracy. This problem may be avoided by applying specially constructed Runge–Kutta integrators, where the accuracy of each sub-stage is at least as good as the previous sub-stage. With such integrators, one could possibly synchronize at each stage, instead of just once per time step.

The application of the multispectral method to two-dimensional Navier–Stokes turbulence was complicated by the presence of two sign-definite invariants, a problem that does not arise in three dimensions. The lessons learnt in developing the multispectral method for shell models of turbulence and the two-dimensional Navier–Stokes equations will allow us to avoid many pitfalls when applying the method to three-dimensional turbulence which, it must be noted, is a system of even greater physical interest.

Bibliography

[Batchelor 1969] G. K. Batchelor, Phys. Fluids, **12 II**:233, 1969.

[Bell & Nelkin 1977] T. Bell & M. Nelkin, Phys. Fluids, **20**:345, 1977.

[Benzi *et al.* 1996] R. Benzi, L. Biferale, R. M. Kerr, & E. Trovatore, Physical Review E, **35**:3541, 1996.

[Biferale *et al.* 1995] L. Biferale, A. Lambert, R. Lima, & G. Paladin, Physica D, **80**:105, 1995.

[Bowman & Roberts 2010] J. C. Bowman & M. Roberts, "`FFTW++`: A fast Fourier transform C^{++} header class for the `FFTW3` library," `http://fftwpp.sourceforge.net`, 2010.

[Bowman & Roberts 2011a] J. C. Bowman & M. Roberts, SIAM J. Sci. Comput., **33**:386, 2011.

[Bowman & Roberts 2011b] J. C. Bowman & M. Roberts, submitted to Communications in Nonlinear Science and Numerical Simulation, 2011.

[Bowman *et al.* 1989] J. C. Bowman, J. A. Krommes, & M. Ottaviani, Bull. Am. Phys. Soc., **34**:1926, 1989.

[Bowman *et al.* 1997] J. C. Bowman, B. A. Shadwick, & P. J. Morrison, "Exactly conservative integrators," in *15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, edited by A. Sydow, volume 2, pp. 595–600, Berlin, 1997, Wissenschaft & Technik.

[Bowman *et al.* 1999] J. C. Bowman, B. A. Shadwick, & P. J. Morrison, Phys. Rev. Lett., **83**:5491, 1999.

[Bowman *et al.* 2001] J. C. Bowman, B. A. Shadwick, & P. J. Morrison, "Numerical challenges for turbulence computation: Statistical equipartition and the method of spectral reduction," in *Scientific Computing and Applications*, edited by P. Minev, Y. S. Wong, & Y. Lin, volume 7 of *Advances in Computation: Theory and Practice*, pp. 171–178, Huntington, New York, 2001, Nova Science Publishers.

[Bowman *et al.* 2006] J. C. Bowman, C. R. Doering, B. Eckhardt, J. Davoudi, M. Roberts, & J. Schumacher, Physica D, **218**:1, 2006.

[Bowman 2011] J. C. Bowman, J. Fluid Mech, 2011, To be submitted.

[Canuto *et al.* 2006] C. Canuto, M. Hussaini, A. Quarteroni, & T. Zang, *Spectral Methods: Fundamentals in Single Domains*, Scientific Computation, Springer, Berlin, 2006.

[Carnevale *et al.* 1981] G. F. Carnevale, U. Frisch, & R. Salmon, J. Phys. A: Math. Gen., **14**:1701, 1981.

[Desnyansky & Novikov 1974] V. N. Desnyansky & E. A. Novikov, Prikl. Mat. Mekh., **38**:507, 1974.

[Eckhardt 2004] B. Eckhardt, private communication, 2004.

[Farge 1992] M. Farge, Annual Review of Fluid Mechanics, **24**:395, 1992.

[Fjørtoft 1953] R. Fjørtoft, Tellus, **5**:225, 1953.

[Frigo & Johnson 2005] M. Frigo & S. G. Johnson, Proceedings of the IEEE, **93**:216, 2005.

[Frisch 1995] U. Frisch, *Turbulence: The Legacy of A.N.Kolmogorov*, Cambdridge University Press, 1995.

[Gledzer 1973] E. B. Gledzer, Sov. Phys. Dokl., **18**:216, 1973.

[Grossmann *et al.* 1996] S. Grossmann, D. Lohse, & A. Reeh, Phys. Rev. Lett., **77**:5369, 1996.

[Hardy 1999] H. Hardy, *Ramanujan*, AMS Chelsea Pub, Providence, 1999.

[Holloway & Hendershott 1977] G. Holloway & M. C. Hendershott, J. Fluid Mech., **82**:747, 1977.

[Holmes *et al.* 1996] P. Holmes, J. L. Lumley, & G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems, and Symmetry*, University Press (1996), Cambridge, 1996.

[Jensen & Olesen 1998] M. H. Jensen & P. Olesen, Physica D: Nonlinear Phenomena, **111**:243 , 1998.

[Jensen *et al.* 1991] M. Jensen, G. Paladin, & A. Vulpiani, Phys. Rev. A, **43**:798, 1991.

[Johnson & Frigo 2007] S. Johnson & M. Frigo, IEEE Transactions on Signal Processing, **55**:111, 2007.

[Kadanoff *et al.* 1995] L. P. Kadanoff, D. Lohse, J. Wang, & R. Benzi, Phys. Fluids, **7**:617, 1995.

[Kolmogorov 1941] A. Kolmogorov, Dokl. Akad. Nauk SSSR, **30**:301, 1941, Reprinted in Proc. R. Soc. Lond. A434, 9–13,1991.

[Kraichnan & Chen 1989] R. H. Kraichnan & S. Chen, Physica D, **37**:160, 1989.

[Kraichnan 1958] R. H. Kraichnan, Phys. Rev., **109**:1407, 1958.

[Kraichnan 1961] R. H. Kraichnan, J. Math. Phys., **2**:124, 1961.

[Kraichnan 1967] R. H. Kraichnan, Phys. Fluids, **10**:1417, 1967.

[Kraichnan 1971] R. H. Kraichnan, J. Fluid Mech., **47**:513, 1971.

[Kraichnan 1975] R. H. Kraichnan, J. Fluid Mech., **67**:155, 1975.

[Kraichnan 1985] R. H. Kraichnan, in *Theoretical Approaches to Turbulence*, edited by D. L. Dwoyer, M. Y. Hussaini, & R. G. Voigt, volume 58 of *Applied Mathematical Sciences Series*, chapter V, p. 91, Springer, New York, 1985.

[Landau & Lifshitz 1959] L. D. Landau & E. M. Lifshitz, *Fluid Mechanics*, Addison-Wesley, Reading, MA, 1959.

[L'vov *et al.* 1998] V. S. L'vov, E. Podivilov, A. Pomyalov, I. Procaccia, & D. Vandembroucq, Phys. Rev. E, **58**:1811, 1998.

[Lee 1989] J. Lee, Physica D, **37**:417, 1989.

[Leith 1968] C. E. Leith, Phys. Fluids, **11**:671, 1968.

[Lorenz 1972] E. N. Lorenz, J. Fluid Mech., **55**:545, 1972.

[Lundy & Van Buskirk 2007] T. Lundy & J. Van Buskirk, Computing, **80**:23, 2007.

[Martin *et al.* 1973] P. C. Martin, E. D. Siggia, & H. A. Rose, Phys. Rev. A, **8**:423, 1973.

[Novikov 1964] E. A. Novikov, J. Exptl. Theoret. Phys. (U.S.S.R), **47**:1919, 1964.

[Orszag 1971] S. A. Orszag, Journal of the Atmospheric Sciences, **28**:1074, 1971.

[Orszag 1977] S. A. Orszag, "Lectures on the statistical theory of turbulence," in *Fluid Dynamics*, edited by R. Balian & J.-L. Peube, pp. 235–373, Gordon and Breach, London, 1977, (summer school lectures given at Grenoble University, 1973).

[Paladin & Vulpiani 1987] G. Paladin & A. Vulpiani, Phys. Rev. A, **35**:1971, 1987.

[Patterson Jr. & Orszag 1971] G. S. Patterson Jr. & S. A. Orszag, Physics of Fluids, **14**:2538, 1971.

[Roberts 2006] M. I. W. Roberts, "A multi-spectral decimation scheme for turbulence simulations," Master's thesis, University of Alberta, 2006, http://www.math.ualberta.ca/~bowman/group/roberts06.pdf.

[Shadwick *et al.* 1999] B. A. Shadwick, J. C. Bowman, & P. J. Morrison, SIAM J. Appl. Math., **59**:1112, 1999.

[She & Jackson 1993] Z.-S. She & E. Jackson, Phys. Rev. Lett., **70**:1255, 1993.

[Vázquez-Semadeni & Scalo 1992] E. Vázquez-Semadeni & J. Scalo, Phys. Rev. Lett., **68**:2921, 1992.

[Williams *et al.* 1987] T. Williams, E. R. Tracy, & G. Vahala, Phys. Rev. Lett., **59**:1922, 1987.

[Yamada & Ohkitani 1987] M. Yamada & K. Ohkitani, J. Phys. Soc. Jap., **56**:4210, 1987.

Index

p/q-rule, 63

advection, 3
aliased, 58
aliasing errors, 58
anomalous scaling, 10

bilinear map, 103
binned spectrum, 20

cascade, 9
Casimir invariants, 8
centered, 57
coincident, 130
coincident-mode synchronization, 130
continuity equation, 4
convolution, 56

dealiasing, 59
discrete cyclic convolution, 58
DN model, 27
dual cascade, 13

energy spectrum, 9
enstrophy, 7
equation of state, 4
equipartition, 15
explicit zero-padding, 61
external forces, 3

fast Fourier transform, 24
FFT, 24
Fourier transform, 13
fractional linear map, 103

Gibbs distribution, 16
grid, 117

helicity, 7
Hermitian symmetry, 13
Hermitian-symmetric, 58
high-resolution shell models, 38
hyperviscosity, 5

implicitly padded Fourier transform, 65
incompressible, 4
incompressible Navier–Stokes equations, 6
inertial range, 9

Kolmogorov dissipation scale, 10

Lagrangian derivative, 3
Liouville theorem, 15
low-resolution shell models, 36

molecular viscosity, 5

non-centered, 57

parallel synchronization, 118
phase-shift dealiasing, 59
pressure, 3
project, 118
prolongation, 119
pruning, 64
pruning transforms, 64
pseudospectral, 23

raw spectrum, 21
Reynolds number, 8

serial synchronization, 118
shell-model helicity, 30
spectral Navier–Stokes, 13
structure function exponent, 10
subgrid, 117

turbulence, 9

viscosity, 3
vorticity, 6

zero-padding, 59