UNIVERSITY OF CALIFORNIA SAN DIEGO

Fast Measurements of Two-Point Statistics in Simulations of Supersonic Isothermal Turbulence

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Physics

by

Richard P. Wagner

Committee in charge:

Michael L. Norman, Chair
Michael Holst
Dušan Kereš
Alexei G. Kritsuk
David A. Meyer

2023

The dissertation of Richard P. Wagner is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

I am dedicated to finishing.

EPIGRAPH

However, like anything/everything related to turbulence it is not easy.

—A. Tsinober

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF FUNCTIONS

LIST OF CODE EXAMPLES

NOMENCLATURE

SFgen Method used to calculate the stastistics of two-point functions, initially named as a "structure function generator". See Chapter 3 xxvii, 8, 9, 10, 20, 21, 22, 25, 26, 41, 67, 77, 125, 128

Re Reynolds number, $Re = uL/\nu$, where u is the flow speed, L is a characteristic length scale, and ν is the viscosity. Re is used to describe the level of turbulence within a flow xiv, 2

c_s The speed of sound in a fluid, defined as $c_\mathrm{s} \equiv 1$ for most of the simulations described in this dissertation xiv, 2, 12

Mach number The root-mean-squared average speed of the fluid in units of the speed of sound, c_s, $\mathcal{M} = u_\mathrm{rms}/c_\mathrm{s}$ xiv, 2, 12

$\mathcal{M}$ See Mach number xiv, 2, 3, 5, 12

subsonic Flows with $\mathcal{M} \lesssim 1$. See also Mach number 120

transonic Flows with $\mathcal{M} \approx 1$. See also Mach number 120

supersonic Flows with $1 \lesssim \mathcal{M} \lesssim 10$. See also Mach number 122

$\mathbb{D}$ The simulation domain. $\mathbb{D}_i$ refers to the distance along axis i xvi, 12

grid A portion of the simulation. The term grid is used interchangeably to refer to the arrays holding the physical state variables and the object with methods used to manage the arrays xiv, xv, xvi, 11, 15, 30

G See grid 30

N_G The total number of grids xviii, 50, 57, 58, 62

$G_{l,i}$ The set of nested grids in an AMR simulation, where l is the level of refinement and i is the index of the grid at that leve; 12

$G_{0,0}$ The first grid in hierarchy, also referred to as the top, or root, grid 15, 56, 57

G.`Index` A unique integer attribute of each grid ordered by the rank of owning task xviii, 58

G.`PhysicalFields` Grid arrays representing the physical fields being modeled in the simulation xvii, 60

refinement factor The integer ratio between the cell widths grids at level l and $l+1$ xv, 13, 31

R See refinement factor 13, 31

hierarchy entry The triply linked list used to associate grids 15, 49

task A single computing process with its own memory domain and a unique MPI rank xv, xviii, 20, 27

rank A positive integer value assigned by MPI to identify tasks xv, 47, 125

T The MPI rank of the current task xv, xviii, 47, 49, 50, 56, 57, 58

N_T The total number of MPI tasks xvii, xviii, 28, 53, 57, 58, 59, 60, 61, 62, 66, 69, 129, 130

$\mathbb{M}$ The array of grids owned by the owned by the local MPI task, T 29, 49, 50, 51, 56, 57, 60

point A location in $\mathbb{D}$ to sample from xvi, xviii

point pair A set of points, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, separated by a fixed distance r where $\boldsymbol{x}_2 \equiv \boldsymbol{x}_1 + r\hat{\boldsymbol{r}}$, at which values for analysis are taken xvi, xviii, 28

separation The distance, r, between the points in a point pair xvi, xvii, 28

r See separation xvi, 22, 24, 41, 69

$\boldsymbol{x}_1$ The first point in a point pair xvi, 41, 68, 121

$\boldsymbol{x}_2$ The second point in a point pair xvi, 41, 45, 69, 121

$\hat{\boldsymbol{r}}$ The unit vector along the line from point $\boldsymbol{x}_1$ to $\boldsymbol{x}_2$ xvi, 21, 29, 41, 42, 43, 45, 46, 68

$\mathbb{L}$ A set of separation distances xvi, xvii, 22, 28, 30

N_L The number of separation distances in $\mathbb{L}$ 28, 62, 66

$\mathbb{F}$ The FastFind lookup array used to optimize the search for the grid containing a point xvi, 31, 32, 33, 35, 36, 37, 39, 40

L_F Maximum level of $\mathbb{F}$ 31, 32, 35, 36, 37

$\mathbb{G}$ The array of grids containing point pairs. $\mathbb{G}$ is indexed as $\mathbb{G}_{r,p,i}$ where r is the separation, p is the index of the pair, and i is the point (1 or 2) xvi, 29, 42, 46, 53, 57, 58, 62

$\mathbb{I}$ The array of cell indices for the grids containing each point pair. $\mathbb{I}$ is indexed similarly to $\mathbb{G}$ 29, 42, 46, 53, 58

$\mathbb{R}$ The array of unit vectors, $\hat{\boldsymbol{r}}$, for each point pair. $\mathbb{R}$ is indexed as $\mathbb{R}_{r,p}$ where r is the separation, p is the index of the pair 42, 46, 53

$\mathbf{\Phi}$ The arrays of physical field values at the point pairs (see G.`PhysicalFields`). $\mathbf{\Phi}$ is indexed as $\mathbf{\Phi}_{r,p,i,f}$ where r is the separation, p is the index of the pair, i is the point (1 or 2), and f is the field 51, 62, 63

N_ϕ The number of physical fields communicated between tasks (the same on every task) 60, 61, 62

m_i A one-dimensional integer array mapping the grid field arrays to the arrays used for communication (the same on every task) 60

N_M The number of passes of the MakePass loop xvii, 28, 29, 66, 69, 75

N_P The number of point pairs sampled at each separation distance in $\mathbb{L}$ during a MakePass loop (the same on every task) xvii, 28, 29, 30, 41, 46, 57, 58, 62, 64, 66, 69

N_Σ The total number of point pairs sampled across tasks and MakePass loops, $N_P \times N_M \times N_T$ xvii, 24, 65, 68, 69, 71, 72, 73, 74, 75

$\mu_{f,r}$ The (likely) uknown "true" or population mean of a two-point function 24, 71

$\sigma^2_{f,r}$ The (likely) uknown "true" or population variance of a two-point function 24, 71

$\hat{\mu}_{f,r}$ Mean of a two-point function from sampling, $\hat{\mu}_{f,r} = \mathbf{\Sigma}_{\mu,f,r}/N_\Sigma$ xvii, 24, 63, 71

$\hat{\sigma}^2_{f,r}$ Variance of a two-point function from sampling, calculated as $\hat{\sigma}^2_{f,r} = \mathbf{\Sigma}_{\sigma,f,r}/N_\Sigma - (\hat{\mu}_{f,r})^2$ xvii, xviii, 24, 63, 71, 74

$\hat{\sigma}_{f,r}$ Standard deviation of a two-point function from sampling, calculated as $\hat{\sigma}_{f,r} = \sqrt{\hat{\sigma}^2_{f,r}}$ xvii, 71

$\mathbf{\Sigma}_{\mu,f,r}$ Running sum of the values for a two-point function, used to calculate the sample mean, $\hat{\mu}_{f,r}$ xvii, 63, 64, 66

$\mathbf{\Sigma}_{\sigma,f,r}$ Running sum of the squared values for a two-point function, used to calculate the sample variance, $\hat{\sigma}^2_{f,r}$ xvii, 63, 64, 66

$\mathbb{P}$ The arrays of binned (discrete) PDF measurements. $\mathbb{P}$ is indexed as $\mathbb{P}_{f,r,b}$ where f is the two-point function, r is the separation, and b is the PDF bin xviii, 22, 24, 28, 29, 63, 64, 65, 66, 77

N_B The number of PDF bins for each two-point function xviii, 28, 63, 64, 66

$F_{f,\text{min}}$ Minimum bin value for the PDF $\mathbb{P}_f$ 63, 64

$F_{f,\text{max}}$ Maximum bin value for the PDF $\mathbb{P}_f$ xviii, 63, 64

$\mathbb{Q}$ The edges of the $\mathbb{P}$ arrays where the left edge is $\mathbb{Q}_{f,r,b}$ and the right is $\mathbb{Q}_{f,r,b+1}$ 24, 63, 65, 66

$\mathbb{O}$ Over unders counts for $\mathbb{P}$ 63, 64, 65, 66

Δ_f Width of the PDF bins, defined as $\Delta_f = \frac{1}{2}\log_{10}(F_{f,\text{max}}/\epsilon_f)N_B$ xviii, 64

ϵ_f Width of the $\log_{10}$ bin around zero for the PDF $\mathbb{P}_f$, see Δ_f xviii, 64

$\mathbb{J}$ An array of the cells of overlapping points owned by the owned by the local MPI task, T, ordered by G.`Index` 50, 53, 55, 58, 59

$\mathbb{K}$ An array of the cell indices requested by other tasks ordered by G.`Index` 49, 50, 53, 55, 59, 60, 61

$\mathbb{S}$ An integer array of length N_T where the total number of grids owned by task t is $\mathbb{S}_t$ 29, 47, 48, 49, 51, 53, 56, 57, 58, 59, 60, 61

$\mathbb{C}$ An integer array of length N_G indexed by G.`Index` with the count of cells from each grid containing a point from the local task's point pairs 50, 51, 57, 58, 59, 61, 62

$\mathbb{B}$ An integer array of length $\mathbb{S}_T \times N_T$ with the count of cells in grids owned by the current task containing points owned by other tasks 51

$\mathbb{U}$ A two-dimensional array of floating-point values holding the cell values of physical field requested by other tasks. Indexed by field and then cell value 50, 51

$\mathbb{V}$ A two-dimensional array of floating-point values holding the cell values of physical field sent by other grids. Indexed by field and then cell value 51

ACRONYMS

CUDA Compute Unified Device Architecture® 123

ESS extended self-similarity 119

HPC high-performance computing 123

ISM interstellar medium 1, 3, 6, 7, 117, 119

MHD magneto-hydrodynamics 7, 11

MIMD multiple instruction, multiple data 123

MPI Message Passing Interface 10, 11, 16, 17, 20, 27, 28, 65, 123

PDE partial differential equation 11, 17

PDF probability density function 22, 24, 26, 28–30, 63, 65

PPM piecewise parabolic method 11, 41

SAMR structured adaptive mesh refinement 12, 14

SIMD single instruction, multiple data 123

SMP symmetric multiprocessing 123

PREFACE

I will address something evident at the beginning, namely the time between when I am submitting this dissertation and when I did most of the research presented in it. Simply put, a dissertation takes a substantial amount of dedicated focus, and until recently, I chose other priorities. Some of these priorities were personal, some were professional, and some were less chosen than due to circumstances. Regardless, I am extremely grateful to have the opportunity to complete my doctorate.

Rather than ignore various timeframes at play, the delay will be used as a framing device to establish the outline of this dissertation. The introduction and method chapters will primarily lay out the state of the research when I was actively doing research and provide background for the rest of the dissertation. Then, I will describe the method I developed, followed by three results chapters published between 2007 and 2103. Finally, in the conclusions chapter, I will review the relevant science up to the present day. During that review, I will answer a question that few other doctoral candidates can, namely, how supported my contributions have been.

The fundamental conclusions of this dissertation (published a decade ago) have been reproduced repeatedly. While I am proud of my part in those publications, I am also aware that starting point for that research came from those around me. Therefore, my wisest action was to put myself where I would have the support to apply my capabilities in scientific computing to worthwhile problems first presented by my colleagues.

ACKNOWLEDGEMENTS

Many people. If you're reading this, you're probably one of them. Thank you.

Chapter 4, in full, is a reprint of "Scaling Laws and Intermittency in Highly Compressible Turbulence", which was published in 2007 in AIP Conference Proceedings, volume 932, page 393, by Alexei G. Kritsuk, Paolo Padoan, Rick Wagner, and Michael L. Norman. Reproduced with the permission of AIP Publishing. The dissertation author was the third investigator and author of this paper.

Chapter 5, in full, is a reprint of "Flux Correlations in Supersonic Isothermal Turbulence", which was published in 2012 in Journal of Fluid Mechanics, volume 713, page 482, by Rick Wagner, Gregory Falkovich, Alexei G. Kritsuk, and Michael L. Norman. Reprinted with permission. The dissertation author was the first investigator and author of this paper.

Chapter 6, in full, is a reprint of "Energy Cascade and Scaling in Supersonic Isothermal Turbulence", which was published in 2013 in Journal of Fluid Mechanics, volume 729, R1, by Alexei G. Kritsuk, Rick Wagner, and Michael L. Norman. Reprinted with permission. The dissertation author was the second investigator and author of this paper.

VITA

2023	Ph. D. in Physics University of California San Diego
2020-Present	Principal Research IT Systems Integration Engineer University of California San Diego
2016-2020	Globus Professional Services Manager University of Chicago
2012-2016	HPC Systems Manager San Diego Supercomputer Center
2010-2012	HPC Systems Engineer San Diego Supercomputer Center
2006-2010	Graduate Research Assistant University of California San Diego
2009	C. Phil. in Physics University of California San Diego
2005-2006	Graduate Teaching Assistant University of California San Diego
2006	M. S. in Physics University of California San Diego
2004	B. S. in Physics *magna cum laude* University of California San Diego
2003	A. A. in Transfer Studies *with honors* San Diego Mesa College

PUBLICATIONS

Charbonneau, A. L., Brady, A., Czajkowski, K., Aluvathingal, J., Canchi, S., Carter, R., Chard, K., Clarke, D. J. B., Crabtree, J., Creasy, H. H., D'Arcy, M., Felix, V., Giglio, M., Gingrich, A., Harris, R. M., Hodges, T. K., Ifeonu, O., Jeon, M., Kropiwnicki, E., Lim, M. C. W., Liming, R. L., Lumian, J., Mahurkar, A. A., Mandal, M., Munro, J. B., Nadendla, S., Richter, R., Romano, C., Rocca-Serra, P., Schor, M., Schuler, R. E., Tangmunarunkit, H., Waldrop, A., Williams, C., Word, K., Sansone, S., Ma'ayan, A., **Wagner, R.**, Foster, I., Kesselman, C., Brown, C. T., White, O., "Making Common Fund data more findable: catalyzing a data ecosystem", *GigaScience*, 11, giac105 (2022)

Petrie, R., Denvil, S., Ames, S., Levavasseur, G., Fiore, S., Allen, C., Antonio, F., Berger, K., Bretonnière, P.-A., Cinquini, L., Dart, E., Dwarakanath, P., Druken, K., Evans, B., Franchistéguy, L., Gardoll, S., Gerbier, E., Greenslade, M., Hassell, D., Iwi, A., Juckes, M., Kindermann, S., Lacinski, L., Mirto, M., Nasser, A. B., Nassisi, P., Nienhouse, E., Nikonov, S., Nuzzo, A., Richards, C., Ridzwan, S., Rixen, M., Serradell, K., Snow, K., Stephens, A., Stockhause, M., Vahlenkamp, H., **Wagner, R.**, "Coordinating an operational data distribution network for CMIP6 data", *Geoscientific Model Development*, 14, 629–644 (2021)

Brummel-Smith, C., Bryan, G., Butsky, I., Corlies, L., Emerick, A., Forbes, J., Fujimoto, Y., Goldbaum, N., Grete, P., Hummels, C., Kim, J., Koh, D., Li, M., Li, Y., Li, X., OShea, B., Peeples, M., Regan, J., Salem, M., Schmidt, W., Simpson, C., Smith, B., Tumlinson, J., Turk, M., Wise, J., Abel, T., Bordner, J., Cen, R., Collins, D., Crosby, B., Edelmann, P., Hahn, O., Harkness, R., Harper-Clark, E., Kong, S., Kritsuk, A., Kuhlen, M., Larrue, J., Lee, E., Meece, G., Norman, M., Oishi, J., Paschos, P., Peruta, C., Razoumov, A., Reynolds, D., Silvia, D., Skillman, S., Skory, S., So, G., Tasker, E., **Wagner, R.**, Wang, P., Xu, H., Zhao, F., "ENZO: An Adaptive Mesh Refinement Code for Astrophysics (Version 2.6)", *JOSS*, 4, 1636 (2019)

Kritsuk, A. G., **Wagner, R.**, Norman, M. L., "Energy Cascade and Scaling in Supersonic Isothermal Turbulence", *JFM*, 729, R1 (2013)

Wagner, R., Falkovich, G., Kritsuk, A. G., Norman, M. L., "Flux Correlations in Supersonic Isothermal Turbulence", *JFM*, 713, 482 (2012)

Lemze, D., **Wagner, R.**, Rephaeli, Y., Sadeh, S., Norman, M. L., Barkana, R., Broadhurst, T., Postman, M., Ford, H., "Profiles of Dark Matter Velocity Anisotropy in Simulated Clusters", *ApJ*, 752, 141 (2012)

Lemze, D., Rephaeli, Y., Barkana, R., Broadhurst, T., **Wagner, R.**, Norman, M. L., "Quantifying the Collisionless Nature of Dark Matter and Galaxies in A1689", *ApJ*, 728, 40 (2011)

Kritsuk, A. G., Norman, M. L., **Wagner, R.**, "On the Density Distribution in Star-forming Interstellar Clouds", *ApJL*, 727, L20 (2011)

Vazza, F., Brunetti, G., Kritsuk, A., **Wagner, R.**, Gheller, C., Norman, M., "Turbulent Motions and Shocks Waves in Galaxy Clusters simulated with AMR", *A&A*, 504, 33 (2009)

Kritsuk, A. G., Norman, M. L., Padoan, P., **Wagner, R.**, "The Statistics of Supersonic Isothermal Turbulence", *ApJ*, 665, 416 (2007)

Hallman, E. J. , O'Shea, B. W., Burns, J. O., Norman, M. L., Harkness, R., **Wagner, R.**, "The Santa Fe Light Cone Simulation Project: I. Confusion and the WHIM in Upcoming Sunyaev-Zel'dovich Effect Surveys", *ApJ*, 671, 27 (2007)

BOOK CHAPTERS & CONFERENCE PROCEEDINGS & PREPRINTS

Ananthakrishnan, R., Chard, K., D'Arcy, M., Kesselman, C., Foster, I., McCollam, B., Pruyne, J., Rocca-Serra, P., Schuler, R., **Wagner, R.**, "An Open Ecosystem for Pervasive Use of Persistent Identifiers", *PEARC '20*, 1, 99, (2020)

Allcock, W. E., Allen, B. S., Ananthakrishnan, R., Blaiszik, B., Chard, K., Chard, R., Foster, I., Lacinski, L., Papka, M. E., **Wagner, R.**, "Petrel: A Programmatically Accessible Research Data Service", *PEARC '19*, 49, 1 (2019)

Strande, S. M., Cai, H., Cooper, T., Flammer, K., Irving, C., von Laszewski, G., Majumdar, A., Mishin, D., Papadopoulos, P., Pfeiffer, W., Sinkovits, R. S., Tatineni, M., **Wagner, R.**, Wang, F., Wilkins-Diehr, N., Wolter, N., Norman, M. L., "Comet: Tales from the long tail: Two years in and 10,000 users later", *PEARC17*, 38, 1 (2017)

Cianfrocco, M. A., Wong-Barnum, M., Youn, C., **Wagner, R.**, Leschziner, A., "COSMIC2: A science gateway for cryo-electron microscopy structure determination", *PEARC17*, 22, 1 (2017)

Wagner, R., Papadopoulos, P., Mishin, D., Cooper, T., Tatineni, M., von Laszewski, G., Wang, F., "User Managed Virtual Clusters in Comet", *XSEDE '16*, (2016)

Kritsuk, A. G., **Wagner, R.**, Norman, M. L., "Scaling in Supersonic Isothermal Turbulence", *ASP Conference Series*, 498, 16 (2015)

Lockwood, G. K., Tatineni, M., **Wagner, R.**, "Storage utilization in the long tail of science", *XSEDE '15*, 32, 1 (2015)

Kreuzer, P., Hufnagel, D., Dykstra, D., Gutsche, O., Tadel, M., Sfiligoi, I., Letts, J., Würthwein, F., McCrea, A., Bockelman, B., Fajardo, E., Linares, L., **Wagner, R.**, Konstantinov, P., Blumenfeld, B., Bradley, D., The CMS Collaboration, "Opportunistic Resource Usage in CMS", *Journal of Physics: Conference Series*, 513, 6 (2014)

Moore, R. L., Baru, C., Baxter, D., Fox, G. C., Majumdar, A., Papadopoulos, P., Pfeiffer, W., Sinkovits, R. S., Strande, S., Tatineni, M., **Wagner, R.**, Wilkins-Diehr, N., Norman, M. L., "Gateways to Discovery: Cyberinfrastructure for the Long Tail of Science", *XSEDE '14*, 39, 1 (2014)

Lockwood, G. K., Tatineni, M., **Wagner, R.**, "SR-IOV: Performance Benefits for Virtualized Interconnects", *XSEDE '14*, 47, 1 (2014)

Wagner, R., Tatineni, M., Hocks, E., Yoshimoto, K., Sakai, S., Norman, M. L., Bockelman, B., Sfiligoi, I., Tadel, M., Letts, J., Würthwein, F., Bauerdick, L. A., "Using Gordon to accelerate LHC science", *XSEDE '13*, 59, 1 (2013)

Tatineni, M., Greenberg, J., **Wagner, R.**, Hocks, E., Irving, C., "Hadoop deployment and performance on Gordon data intensive supercomputer", *XSEDE '13*, 45, 1 (2013)

Wagner, R., Hallman, E. J., O'Shea, B. W., Burns, J. O., Norman, M. L., Harkness, R., So, G., "The Santa Fe Light Cone Simulation research project files", *UC San Diego Library Digital Collections*, Data version 1.0 (2013)

Strande, S. M., Cicotti, P., Sinkovits, R. S., Young, W. S., **Wagner, R.**, Tatineni, M., Hocks, E., Snavely, A., Norman, M., "Gordon: design, performance, and experiences deploying and supporting a data intensive supercomputer", *XSEDE '12*, 3, 1 (2012)

Hereld, M., Insley, J. A., Olson, E. C., Papka, M. E., Vishwanath, V., Norman, M. L., **Wagner, R.**, "Exploring large data over wide area networks", *Large Data Analysis and Visualization (LDAV)*, 2011, 133 (2011)

Wagner, R., "A RESTful catalog for simulations", *Mem. Soc. Astron. Ital.*, 80, 365 (2009)

Kritsuk, A. G., Padoan, P., **Wagner, R.**, Norman, M. L., "Scaling Laws and Intermittency in Highly Compressible Turbulence", *AIP Conference Proceedings*, 932, 393 (2007)

Wagner, R., Norman, M.L., "Theory SkyNode", *ASP Conference Series*, 382, 289 (2007)

Norman, M. L., Bryan, G. L., Harkness, R., Bordner, J., Reynolds, D., O'Shea, B., **Wagner, R.**, "Simulating Cosmological Evolution with Enzo" Petascale Computing: Algorithms and Applications, Ed. D. Bader, CRC Press LLC (2007)

Kritsuk, A. G., **Wagner, R.**, Norman, M. L., Padoan, P., "High resolution simulations of supersonic turbulence in molecular clouds", *ASP Conference Series*, 359, 84 (2006)

ABSTRACT OF THE DISSERTATION

Fast Measurements of Two-Point Statistics in Simulations of Supersonic Isothermal Turbulence

by

Richard P. Wagner

Doctor of Philosophy in Physics

University of California San Diego, 2023

Michael L. Norman, Chair

This dissertation describes my work developing and using SFgen: a fast, MPI-parallel method for measuring two-point statistics in simulation data produced by the AMR code Enzo. My research using SFgen focused on measuring several scaling relations in large-scale simulations modeling the supersonic isothermal turbulence observed in molecular clouds. This work contributed to research on three-dimensional compressible turbulence, mainly whether the exchange of energy between scales is local in Fourier space on average, similar to incompressible turbulence. A reliable model for energy flux between scales in supersonic turbulence is essential to understanding molecular clouds' lifecycles and predicting the

magnitude and scale of their driving sources. Furthermore, since molecular clouds are the birthplaces of stars, the energy cascade has further implications for several factors of star formation. The results of my research support a local energy cascade in supersonic turbulence and have been reproduced by others.

Chapter 1

Introduction

> Turbulence is the graveyard of theories.
>
> H. W. Liepmann

Stars, like tropical cyclones, only form under certain conditions, and in both cases, those conditions are only present in particular locations. For tropical cyclones, these locations are the seven tropical cyclone basins. For stars, formation happens in molecular clouds, cold ($10 - 100$K) and highly turbulent regions of the ISM (interstellar medium) filled with molecular hydrogen (H_2) (Hennebelle and Falgarone 2012). Molecular clouds are subjects of intense research because of their central role in the star formation process, similar to understanding the climate to improve predictions of extreme weather events (Almazroui et al. 2021). Molecular clouds and the climate are also both multiscale problems in time and space that include numerous coupled processes; advances in computing capabilities (hardware and software) have directly enabled in both areas (Brandenburg and Nordlund 2011; Eyring et al. 2016).

The role of turbulence within the ISM and molecular clouds has been studied for many reasons, including its impact on the star formation rate and the mass distribution of

stars (Hennebelle and Falgarone 2012; Krumholz and McKee 2005). Defining turbulence can be challenging (see, for example, Appendix A of Tsinober 2004), but the first sentence of Lele (1994) is appropriate for this context: "Turbulence is a macroscopic state of flow in which the instantaneous flow variables exhibit a seemingly random variation in time and in the three spatial coordinates." Characterizing turbulence can be similarly challenging, except that turbulent flows have reproducible statistical properties, similar to other chaotic systems.

The first value commonly used to describe a turbulent flow is the Reynolds number, $Re = uL/\nu$, where u and L are a characteristic flow speed and length scale, respectively, and ν is the viscosity (Reynolds 1883). Re is the ratio of the inertial and viscous forces and is used to describe the level of expected turbulence within a flow. Different flow configurations will begin to exhibit turbulence above critical Reynolds numbers. For a flow in a straight cylindrical pipe, this is around $Re \approx 2,000$ (Davidson 2004). Likewise, the level of turbulence decreases with the Reynolds number and below the critical Reynolds the flow will be smooth, or laminar. In molecular clouds $Re \sim 10^8$, provided viscosity is the dominant damping mechanism, rather than magnetic fields (Elmegreen and Scalo 2004; Hennebelle and Falgarone 2012). The characteristic flow speed in that estimate is a second value often used to describe compressible turbulence, the Mach number, $\mathcal{M} = u_{\rm rms}/c_{\rm s}$, where $u_{\rm rms}$ is the rms speed of the fluid and $c_{\rm s}$ is the sound speed. In molecular clouds the flow is supersonic with typical values for $\mathcal{M}$ of $1 \lesssim \mathcal{M} \lesssim 15$ (Elmegreen and Scalo 2004; Girichidis et al. 2020; Hennebelle and Falgarone 2012), although peaks of $\mathcal{M} \approx 50$ have been observed (Mac Low and Klessen 2004).

Without additional energy input, the kinetic energy in a turbulent fluid will dissipate through viscosity at small scales. For molecular clouds, this dissipation is estimated to occur on a timescale of $\approx$ 1Myr (Mac Low 1999). As the motion of the gas slows and becomes less chaotic, gravity will begin to dominate the kinematics, leading to collapse and potential

star formation. This dissipation timescale is at odds with the observed ages of molecular clouds of more than 20Myr (Larson 1981). Supersonic turbulence driven by large-scale external forces such as galactic shear or feedback from star formation (Hennebelle and Falgarone 2012) could support molecular clouds against collapse as the motion of the gas outpaces the work of gravity. Recent models predict that gravitational collapse sufficient to drive star formation does occur within a few Myr, after which the new stars disrupt the molecular cloud (Chevance et al. 2020). In this model, the large-scale gravitational collapse could be the energy source of the supersonic turbulence in the cloud. This model also requires the steady creation of molecular clouds, possibly through turbulence motion of the warm ISM.

The dynamics of supersonic turbulence are a substantial part of several other astrophysical systems, including Wolf-Rayet stars (Moffat and Robert 1994) and black holes (Hobbs et al. 2011); it also plays a role in inertial confinement fusion (Blue et al. 2005), volcanic eruptions (Ogden, Glatzmaier, and Wohletz 2008), and, unsurprisingly, aviation, including scramjet fueling (Ingenito and Bruno 2010) and controlling noise from jet aircraft[1] (Jordan and Colonius 2013). In all of these examples, the density of the gas varies significantly, and unlike incompressible turbulence the velocity, $\boldsymbol{u}$, has a non-zero dilational component; that is, $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$ is not a guaranteed property of the flow. These differences and others break assumptions used to define measurable properties of incompressible turbulence. In particular, kinetic energy density is not a quadratic invariant of inviscid compressible flows and is not a conserved quantity advected within the inertial range, where the inertial range is roughly defined as lengths smaller than the driving scale and larger than the dissipation scale. In other words, the inertial range is an upper and lower bound on scales where we may treat the driving (source) and dissipation (sink) terms

[1]This includes aircraft such as jet airliners that use turbofans. Thrust from turbofans comes from both the air moved by the fan and the hot engine exhaust. The engine exhaust velocity can reach $\approx 2\mathcal{M}$ while the flow from the fan is subsonic.

as zero in the scale-by-scale energy budget (Frisch 1995). However, as stated in Section 4.5 of Elmegreen and Scalo (2004):

> Virtually all the phenomenology associated with incompressible turbulence traces back to the inviscid global conservation of kinetic energy; unfortunately, compressible turbulence does not share this property.

Perhaps the most fundamental example of this is the Richardson-Kolmogorov energy cascade (Kolmogorov 1941a,b,c; Richardson 1922) and the specific predictions of Kolmogorov's four-fifths law (Kolmogorov 1941a). In three-dimensional turbulence, the Richardson-Kolmogorov model is based on the premise that, within the inertial range, structures in the flow, described as eddies, or "blobs of vorticity" (Davidson 2004), of similar size exchange energy in a cascade from large to small scales, until energy is dissipated by viscosity at the smallest scales. Richardson described the energy cascade qualitatively in 1922 and in 1941 Kolmogorov proposed the following relation for homogeneous turbulence,

$$\left\langle [\delta u_{\parallel}(r)]^3 \right\rangle = -\frac{4}{5}\epsilon r, \tag{1.1}$$

where

$$\delta u_{\parallel}(r) \equiv [\boldsymbol{u}(\boldsymbol{x}+\boldsymbol{r}) - \boldsymbol{u}(\boldsymbol{x})] \cdot \boldsymbol{r}/r \tag{1.2}$$

is the longitudinal velocity difference between two points separated by a vector $\boldsymbol{r}$, $r = |\boldsymbol{r}|$, $\langle \ldots \rangle$ denotes a spatial average over both $\boldsymbol{x}$ and the possible angles of $\boldsymbol{r}$, and ϵ is the mean energy dissipation rate (Kolmogorov 1941a). Equation 1.2 and left-hand side of Equation 1.1 are structure functions, where Equation 1.2 is the first-order longitudinal velocity structure function, and the left-hand side of Equation 1.1 is the third order version of the same (see Sayles and Thomas 1977, for a discussion of second-order structure functions in comparison to autocorrelation). Equation 1.1 can be derived from the von Kármán-Howarth relation (von Kármán and Howarth 1938) for turbulence that is both homogeneous and isotropic.

Equation 1.1 is known as Kolmogorov's four-fifths law, and it is difficult to overstate its significance: it is well-supported experimentally and any scaling relations for compressible turbulence must agree with Equation 1.1 in the limit $\mathcal{M} \to 0$. Kolmogorov's four-fifths law and the experimental evidence also emphatically support the locality of the energy cascade (i.e., energy transfer between structures of comparable scale) in incompressible turbulence. But, the derivation of Equation 1.1 is based on the inviscid global conservation of kinetic energy, therefore, an aspect of its closure model is not valid for compressible flows. Thus, the linear scaling of $\left\langle [\delta u_{\parallel}(r)]^3 \right\rangle$ in the inertial range of incompressible turbulence is not assured for $\mathcal{M} \gtrsim 1$.

Numerical modeling of transonic turbulence ($\mathcal{M} \approx 1$) showed a slight departure from the four-fifths law (Porter, Pouquet, and Woodward 2002), which was potentially attributable to the presence of dilational velocity modes. While further simulations at higher Mach numbers ($1 \lesssim \mathcal{M} \lesssim 3$) suggested that intermittency (variations in dissipation seen in all turbulent flows) could predict deviations from Equation 1.1 in supersonic turbulence (Padoan et al. 2004), simulations at $\mathcal{M} > 3$ showed that density variations mediate kinetic energy transfer between scales in supersonic turbulence (Boldyrev, Nordlund, and Padoan 2002; Kritsuk et al. 2006). Since within supersonic turbulence the density and velocity fields are not strongly coupled, and their power spectra scale differently (see Figures 4, 7, and 15 of Kritsuk et al. 2007b, and Figure 5.1 in Chapter 5), two questions arose, one very pragmatic and the other more foundational. For supersonic isothermal turbulence:

- Can Kolmogorov's four-fifths law be generalized to compressible turbulence by defining scaling relations using suitable density-velocity correlators?
- Is energy transfer local–i.e., due to interactions of structures of comparable scale–within the inertial range?

This dissertation describes my contribution to answering those questions, beginning

with the first and continuing through the second. The latter question is clearly more critical since a reliable model for the transfer of energy across scales in supersonic turbulence is essential to understanding the lifecycles of molecular clouds and to estimating the magnitude and scale of the driving sources (Elmegreen and Scalo 2004; Girichidis et al. 2020; McKee and Ostriker 2007). My research focused on measuring several scaling relations using data from large-scale simulations of the supersonic isothermal turbulence observed in molecular clouds (Kritsuk et al. 2007a; Kritsuk, Wagner, and Norman 2013; Kritsuk et al. 2007b; Wagner et al. 2012) in the ISM. Modeling, analyzing, and describing every possible dynamic of molecular clouds is far beyond the scope of this dissertation. Instead, this work was limited to models of molecular clouds using hydrodynamical simulations of isothermal turbulence; this was a practical choice, well described in Porter, Pouquet, and Woodward (1992):

> Many complex phenomena are at play in such clouds, besides hydrodynamics: coupling to a magnetic field, thermal and gravitational instabilities, radiative transfer, intricate chemistry within MHD shocks, ... We may consider a simpler subproblem and analyze what are the relevant observational features that can thus be recovered. In view of the ensemble of observations just mentioned and of the fact that the Reynolds numbers in such clouds are huge, a pure hydrodynamical description in terms of a supersonic flow may suffice in a range of intermediate scales.

Restricting the problem to isothermal fluids without magnetic fields or other coupled systems allows the work to bridge the well-established properties of incompressible fluids and supersonic fluids, including molecular clouds. The results support a local energy cascade in supersonic turbulence (in particular, see Section 6.5 and Kritsuk, Wagner, and Norman 2013), which has been reproduced by others (see Chapter 7).

1.1 Suggested Readings

The results chapters (Chapters 4, 5, and 6) each provide somewhat overlapping scientific backgrounds on:

- turbulence, both incompressible and compressible;
- various aspects of star formation;
- the observed and predicted physical properties of molecular clouds;
- and numerical modeling of turbulent fluids.

Rather than repeat or paraphrase that information, I would like to highlight a few references and suggest some additional ones.

The first suggestion is the *Space Science Reviews* article "Physical Processes in Star Formation" (Girichidis et al. 2020), which details several of the coupled systems (MHD, chemistry, stellar feedback, etc.) at work in star formation. The article's particular relevance for this dissertation is in Section 4, "Turbulence", which was largely written by A. Kritsuk, who, along with Prof. M. Norman, was a coauthor on each of the the publications included as my results chapters. That section gives a recent and very complete summary of compressible turbulence and its application to star formation, molecular clouds, and the ISM. Subsection 4.5, "Scaling Relations and Energy Cascades in Compressible Turbulence" describes the research timeline which my work was a part of, and includes the latest progress. I will discuss this article a bit more in Chapter 7.

Four other review articles also cover the relevant aspects of the ISM and star formation, from both observational and theoretical perspectives. Three are *Annual Review of Astronomy and Astrophysics* articles, namely "Interstellar Turbulence I: Observations and Processes" (Elmegreen and Scalo 2004), "Theory of Star Formation" (McKee and Ostriker 2007), and "Star Formation in the Milky Way and Nearby Galaxies" (Kennicutt

and Evans 2012). Section 5 of Elmegreen and Scalo (2004), "Simulations of Interstellar Turbulence", is an excellent background on the areas which numerical modeling continues to contribute to our knowledge of the ISM. The fourth review article is "Turbulent molecular clouds" (Hennebelle and Falgarone 2012), published in The Astronomy and Astrophysics Review.

For a background on turbulence, the modern canonical reference is *Turbulence: the Legacy of A. N. Kolmogorov*, by Frisch (1995), which provides a concise summary of the statistical methods used in turbulence research, the theoretical and experimental successes, and explicitly acknowledges the impact of Kolmogorov's 1941 work (Kolmogorov 1941a,b,c). I found several other books valuable. Two are general texts on turbulence, *Turbulent Flows* (Pope 2000) and *Turbulence: An Introduction for Scientists and Engineers* (Davidson 2004), both of which are broader than Frisch and go into more detail for particular types of flows. Personally, I preferred the narrative style of Davidson over Pope, which made Davidson more approachable (note that there is a second edition of Davidson's book, Davidson 2015). The next book is *Homogeneous Turbulence Dynamics* (Sagaut and Cambon 2018), an extensive monograph that includes several chapters on compressible turbulence.

1.2 Dissertation Outline

Simulations & Analysis Methods Chapter 2 gives a short description of the software used for the simulations, Enzo (Brummel-Smith et al. 2019; Bryan et al. 2014), to sufficiently describe the relevant Enzo data structures and how those structures are distributed across compute processes. Chapter 3 introduces SFgen, the method I developed to rapidly and efficiently measure high-order scaling relations in large-scale three-dimensional simulations. SFgen is part of a larger framework that I developed to integrate data analysis directly into Enzo for use at runtime. Other tools using the framework provide one-dimensional

statistics, projections of fields, and extracted fields along a ray. I first used these tools to build synthetic maps of the Sunyaev-Zel'dovich Effect (Hallman et al. 2007). SFgen itself was created for measuring structure functions as part of the research for Kritsuk et al. (2007a) and Kritsuk et al. (2007b). Later, the one-point statistics contributed to the analysis of star-forming molecular clouds undergoing gravitational collapse (Kritsuk, Norman, and Wagner 2011) and SFgen was also used to analyze turbulence within galaxy clusters (Vazza et al. 2009).

Results & Publications Chapters 4, 5, and 6 are part of a series of publications that highlight the use of the method and numerical support for a local compressible energy cascade. Chapter 4 was published as the conference proceeding "Scaling Laws and Intermittency in Highly Compressible Turbulence" (Kritsuk et al. 2007a). This research initiated the method development to explore density-weighted versions of Equation 1.1 motivated by the scaling relations proposed in Fleck (1996). Chapter 4 also provides additional background on compressible turbulence, structure functions, and the relevant statistics. Chapter 5 presents the evaluation of an analogue to the four-fifths law based on the currents and densities of conserved quantities proposed in Falkovich, Fouxon, and Oz (2010) and was published as "Flux Correlations in Supersonic Isothermal Turbulence" (Wagner et al. 2012). Chapter 6 evaluated a second relation proposed in Galtier and Banerjee (2011) and was published as "Energy Cascade and Scaling in Supersonic Isothermal Turbulence" (Kritsuk, Wagner, and Norman 2013). Combined with the results in Kritsuk et al. (2007b), these publications provide evidence that the energy cascade within compressible turbulence is local.

Conclusions I have a (hopefully) somewhat unique opportunity to see how worthwhile my research contributions were. Chapter 7 explores more recent research in supersonic turbulence in molecular clouds, and how these results compare to those in the Chapters 4,

5, and 6.

Nomenclature, Acronyms, & Appendices Lists of acronyms and nomenclature used are provided in the preliminary pages. Appendix A summarizes the syntax I have used to bridge between mathematical notation and the pseudocode used to describe the algorigthms in SFgen. Appendix B is a short description of the MPI (Message Passing Interface) programming library. Appendix C is a list of peer-reviewed journal articles and a book by other authors that cite one or more of the results chapters. Appendix D is the bibliogrpahy of works cited in this dissertation.

Chapter 2

Simulations & Enzo

> It is not sufficient to set up the code and let the computer zip along. It zips all right, but to where?
>
> L. Kadanoff

This chapter describes Enzo, an MPI-parallel structured adaptive mesh refinement code (Berger and Colella 1989; Brummel-Smith et al. 2019; Bryan et al. 2014; Norman et al. 2007). Enzo models fluids using several coupled PDE (partial differential equation) solvers that can evolve a simulation of ideal MHD (magneto-hydrodynamics), including gravity, in a comoving (cosmological) coordinate system. The following sections include a description of Enzo, focusing on its data structures related to handling the representing the grids representing the simulation domain and the algorithms used.

For simulations of supersonic turbulence, Enzo solves the Euler equations for density, ρ, and velocity, $\boldsymbol{u}$, driven by an external acceleration term $\boldsymbol{F}$, using the PPM (piecewise parabolic method) (Colella and Woodward 1984),

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho\, \boldsymbol{u}) = 0, \tag{2.1}$$

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} = -\boldsymbol{\nabla} \rho / \rho + \boldsymbol{F}. \tag{2.2}$$

To simulate an isothermal gas, the specific heat ratio in the ideal gas equation of state is set to close to unity, $\gamma = 1.001$.

The simulation domain, $\mathbb{D}$, is defined as a period cube with sides of length $L = 1$. The density field is initialized with a uniform distribution of $\rho(\boldsymbol{x}, t = 0) = 1$ and the sound speed is assumed to be $c_{\mathrm{s}} \equiv 1$. The acceleration field $\boldsymbol{F}$ is constructed with power limited to wavenumbers $k/k_{\mathrm{min}} \in [1, 2]$ where $k_{\mathrm{min}} = 2\pi$ (see Figure 5.2). The initial velocity field is scaled to $\boldsymbol{F}$, $\boldsymbol{u}(\boldsymbol{x}, t = 0) \propto \boldsymbol{F}(\boldsymbol{x}, t = 0)$, so that the root mean squared Mach number is $\mathcal{M} = 6$. $\boldsymbol{F}$ is spatially fixed such that $\boldsymbol{F}(\boldsymbol{x}, t) = \varepsilon(t)\boldsymbol{F}_0(\boldsymbol{x})$ and scaled by the initial energy injection rate at $t = 0$ to maintain a constant level of energy in the fluid (Mac Low 1999)

$$\varepsilon_0 = \langle \rho(t = 0)\boldsymbol{u}(t = 0) \cdot \boldsymbol{F}_0 \rangle \,, \tag{2.3}$$

$$\varepsilon(t) = \frac{\varepsilon_0}{\langle \rho(t)\boldsymbol{u}(t) \cdot \boldsymbol{F}_0 \rangle}. \tag{2.4}$$

This acceleration is a large-scale source of energy and balances the kinetic energy loss caused by numerical dissipation when modeling strong shocks (Porter, Pouquet, and Woodward 1992). The rate of energy injection, $\varepsilon(t)$, varied by $\approx 5\%$ in the simulations and ε_0 is the assumed average rate of energy transfer between scales used when evaluating scaling relations.

2.1 Data Structures

Enzo implements the SAMR (structured adaptive mesh refinement) strategy proposed by Berger and Colella (1989). In SAMR codes, $\mathbb{D}$ is broken into a set of nested grids, $G_{l,i}$, at multiple levels, $0 \leq l \leq l_{max}$, where i is the index of the grid on level l referred to here as the level index. All the grids at a particular level will have the same resolution, or

cell spacing, Δ_l. At level 0, the grids fill the simulation domain, and their boundaries are typically defined using recursive bisection so that each grid $G_{0,i}$ will have the same spatial dimensions and require similar time for each time step. Beyond level 0, the resolution of nested grids (subgrids) is defined by a positive integer refinement factor R

$$\Delta_{l+1} = \frac{\Delta_l}{R} \tag{2.5}$$

$$\Delta_l = \frac{\Delta_0}{R^{l-1}} \tag{2.6}$$

While Enzo supports multiple coordinate systems, non-uniform grid spacing, and various boundary conditions, this description will focus on Cartesian grids and periodic boundary conditions at the edges of the simulation domain with equal grid cell widths along each axis.

Figure 2.1 shows a 16×16 two-dimensional simulation domain broken into four level 0 grids of 8×8, $G_{0,i=1\ldots4}$. Except for $G_{0,1}$, each level 0 grid has subgrids, as does the level 1 grid $G_{0,2}$. Lines are drawn within each grid to represent the cell boundaries. The boundaries and cells shown do not include ghost zones. Ghost zones are additional cells beyond a grid's boundaries with data copied from neighboring grids and are required for the stencil of the solver.

The spatial locations of subgrids in Enzo are constrained in a few ways:

- a grid must be fully within its parent grid;
- its boundaries must align cell edges of its parent grid;
- and the subgrid must cover at least 2 of its parent grid's cells along each axis.

These conditions prevent a grid from "skipping" a level of refinement by ensuring that it overlaps with a grid one level higher. However, it can also cause a grid or refinement region to be broken across two coarser grids—look at grids $G_{1,0}$ and $G_{1,3}$ in Figure 2.1

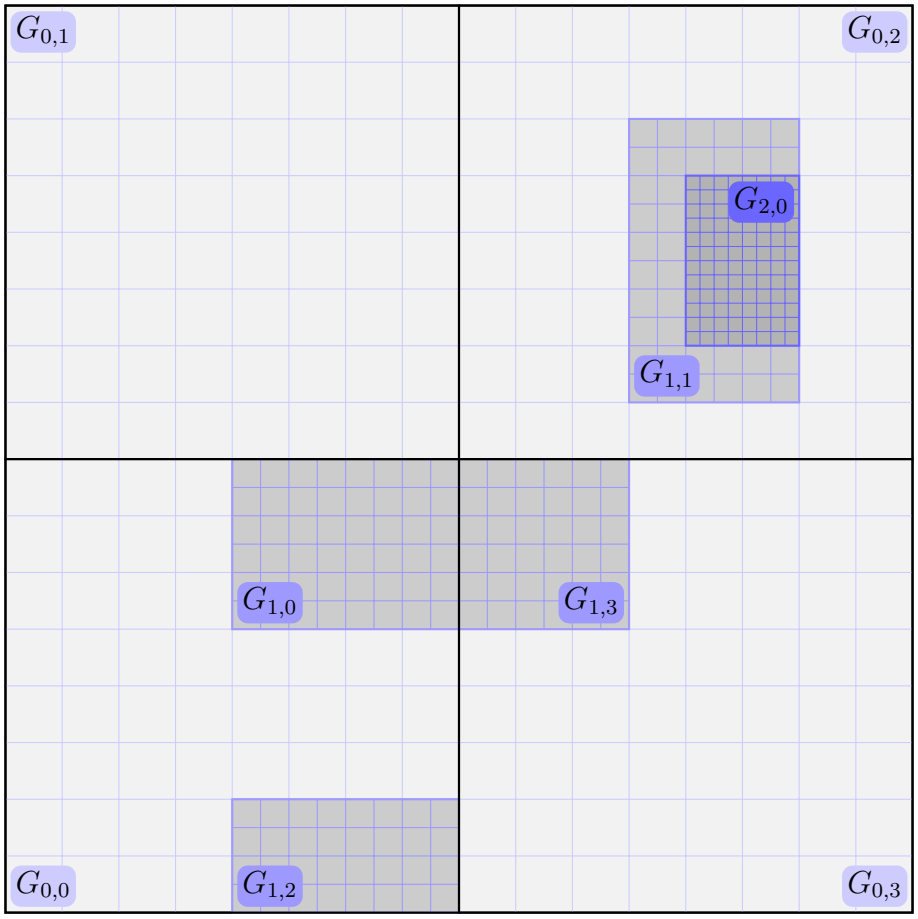

Figure 2.1: An example of Enzo grids with 3 levels of refinement.

for an example. This isn't required by SAMR but is a result of Enzo's parallel SAMR implementation[1]. These constraints help when locating subgrids in space for this work, as we will see in Section 3.4.

References to the grids are stored in a hierarchical tree structure and an array of linked lists indexed by level. The methods I developed only rely on the tree structure where each grid $G_{l,j}$ is a node on the tree and has pointers to:

- $G_{l-1,i}$, its parent grid at the higher (less refined) level;
- $G_{l,j+1}$, the next grid at the same level with the same parent grid;

[1]This can make it challenging when populating ghost zones and setting the boundary conditions of refined grids, see Collins (2009)

- $G_{l+1,k}$, the first grid at the next (more refined) level within the spatial boundaries of $G_{l,j}$;
- and $D_{l,j}$, the data structure with the grids' methods, spatial (left and right edges, grid spacing, etc.), and physical field data.

These pointers are named `ParentGrid`, `NextGridThisLevel`, `NextGridNextLevel`, and `GridData`, respectively. The pointers are set to `NULL` at initialization to determine the ends of the tree. Figure 2.2 shows the structure of a node in the hierarchy, and Figure 2.3 shows the hierarchy for the grids shown in Figure 2.1. Grid $G_{0,0}$ is usually referred to as the top, or root, grid.

Outside of the software, the terminology around grids can be a bit loose. Colloquially, the term "grid" is often interpreted to mean both the node in the hierarchy $G_{l,i}$ and the C++ object $D_{l,i}$ with computationally useful information and methods. I'm going to contribute a bit to this confusion: In the code, the attributes and methods of the grid object, $D_{l,j}$, are accessed from a hierarchy entry as:

```
HierarchyEntry *grid;
grid->GridData->Method();
```

However, I will use the following shorthand in pseudocode for brevity:

```
HierarchyEntry *grid;
grid->Method();
```

Terminology can also be confusing when discussing the hierarchy and its relation to level. *Down* describes going from coarser to more refined levels, even though the level l is increasing. Going "up a level" means going from l to $l-1$. This comes from the tendency to visualize the hierarchy starting with the coarsest grids at the top and more refined grids below. Finally, the number of levels may begin at 1 rather than 0.

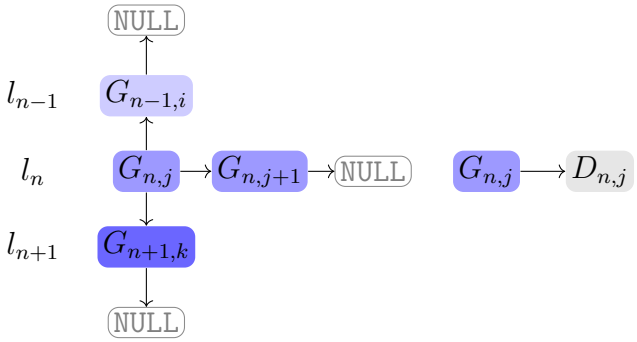

Figure 2.2: Grid hierarchy entry data structure. Some of the pointers from grids $G_{n-1,i}$, $G_{n,j+1}$, and $G_{n+1,k}$ are not shown.

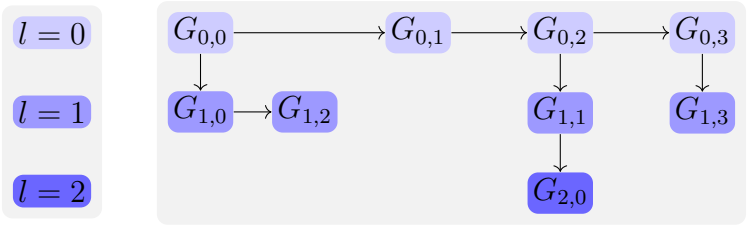

Figure 2.3: AMR hierarchy of the example grids from Figure 2.1. `NULL` references are not shown.

2.2 Parallelism in Enzo

The primary building block for Enzo's parallelization is the MPI (Message Passing Interface)[2] (Message Passing Interface Forum 2021). When running in parallel, each Enzo task has a copy of the hierarchy tree and basic attributes of each grid, like its position and dimensions of the arrays representing the physical fields. This allows each task to find neighboring or overlapping grids for boundary condition updates or to see if a subgrid already overlaps a region requiring refinement. The various physical fields (densities, velocity, gravitational potential, etc.) are only stored on the task that "owns" the grid to limit the amount of memory used by each task and avoid unnecessary communication.

A load balancing algorithm distributes the grids among the tasks roughly based on the total number of cells in a grid. This is because the amount of time to compute a

[2]Appendix B provides some basic information about MPI for those not familiar with it.

solution on each grid scales approximately with the number of cells, and one goal is to have all the tasks solve the PDEs for their grids at the same time. As described earlier, grids at $l = 0$ are typically the same size, and the norm is to use the same number of tasks as level one grids (this is not strictly necessary). Each taks is assigned a single level zero grid, and then the subgrids are distributed using the load balancing algorithm. There is no relationship between the spatial location of the subgrid and the owning task.

2.3 Example Parallel Grid Distribution

The following figures and table provide an example grid hierarchy that will be reused in Chapter 3:

- Figure 2.4 shows how the grids from Figure 2.1 might be assigned to four MPI tasks, $\{T_0, T_1, T_2, T_3\}$.
- Table 2.1 summarizes the distribution.
- Figure 2.5 show the hierarchy with the individual grids shaded by the owning task.
- Figure 2.6 shows the copies of the hierarchy on each task, where filled in grids represent ones with a portion of the simulation state.

The distribution of grid data amongst the tasks determines when MPI communications must occur. For example, if T_0 needs to compute a value based on cell data from a point in $G_{0,1}$, the values will need to be sent from T_1 to T_0. There may also need to be some synchronization point so that T_0 can request the data and T_1 can send it.

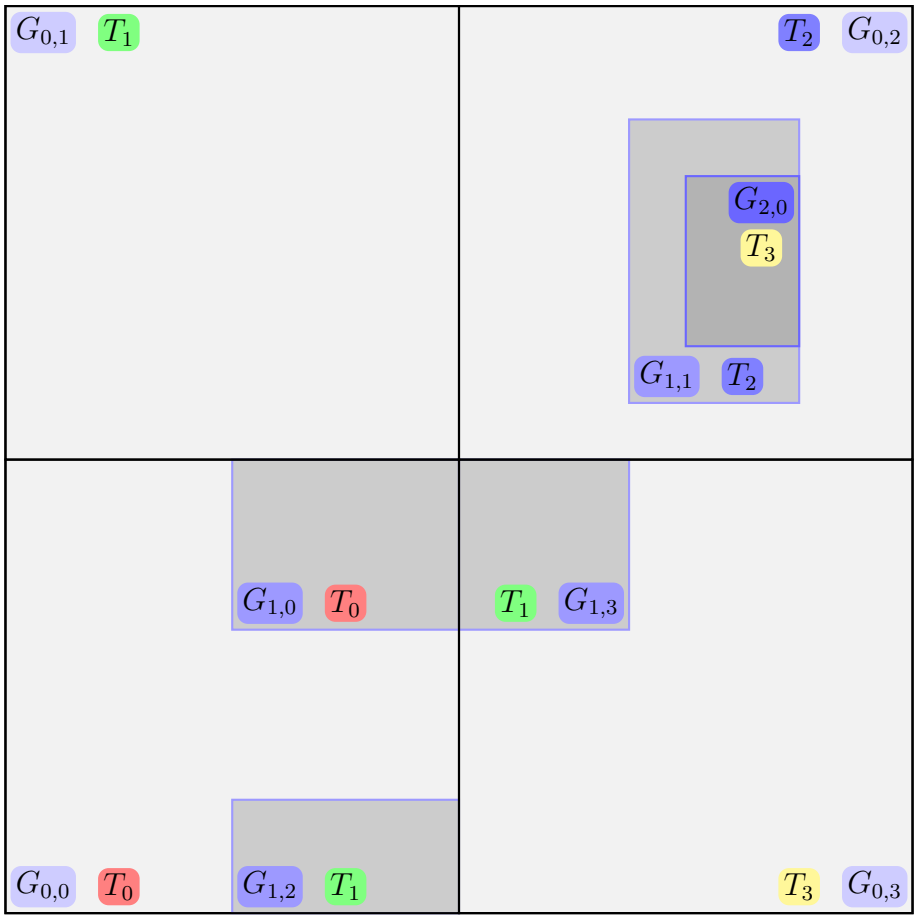

Figure 2.4: Example AMR grids labelled with the owning task.

Table 2.1: Tasks owning example grids. The left columns are ordered by the level and level index; the right columns are first ordered by task, then level and level index.

↓ Grid	Task T_i	Grid	Task T_i ↓
$G_{0,0}$	T_0	$G_{0,0}$	T_0
$G_{0,1}$	T_1	$G_{1,0}$	T_0
$G_{0,2}$	T_2	$G_{0,1}$	T_1
$G_{0,3}$	T_3	$G_{1,2}$	T_1
$G_{1,0}$	T_0	$G_{1,3}$	T_1
$G_{1,1}$	T_2	$G_{0,2}$	T_2
$G_{1,2}$	T_1	$G_{1,1}$	T_2
$G_{1,3}$	T_1	$G_{0,3}$	T_3
$G_{2,0}$	T_3	$G_{2,0}$	T_3

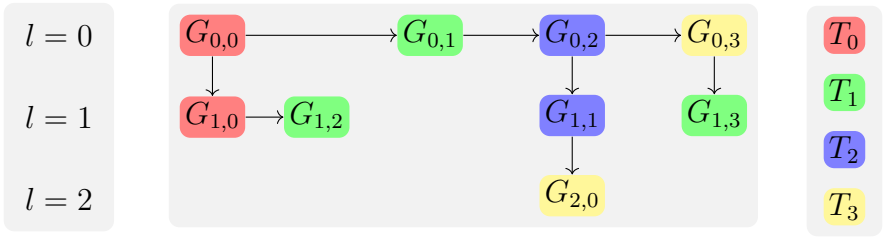

Figure 2.5: Example hierarchy with the grids colored by task.

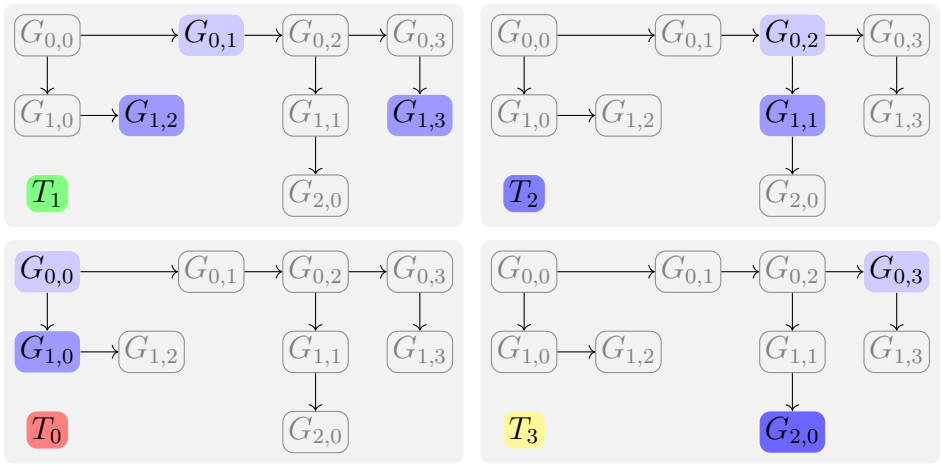

Figure 2.6: AMR hierarchy as owned by each task T_i in the example. Highlighted grids are ones where the owning task has the field data, i.e., ρ and $\boldsymbol{u}$.

Chapter 3

Method: SFgen

> Once you certify the code, it can go to work, and you really know that the answer is going to be true to a given accuracy
>
> J. Jiménez

This chapter describes the computational aspects (data structures, algorithms, and communications) of SFgen, the method used to measure scaling relations using the statistics of two-point[1] functions in Enzo simulations. While two-point statistics are frequently measured in simulations, only a few parallel applications have been published. One is the parallel structure function generator described in Skory (2010) and incorporated into `yt` (Turk et al. 2011). A very recent second example is `fastSF` (Sadhukhan, Bhattacharya, and Verma 2021). However, unlike SFgen and the method in Skory (2010), `fastSF` assumes that each MPI task has access to the fields in the full simulation domain and is merely orchestrating the work using MPI. As stated in Chapter 1, SFgen was designed to be run within Enzo as a simulation evolves to provide immediate analysis results; this required

[1] SFgen could be adapted to support an arbitrary number of points per sample.

working with the distributed memory model of the grid hierarchy and parallel performance (scaling with the number of tasks) to be on par with Enzo's.

3.1 Background

Measuring the expected (mean) values of scaling relations such as Equation 1.1 is fundamental to the study of turbulence. As an example, there are the density-weighted structure functions studied in Chapter 4,

$$\mathcal{S}_p(r) \equiv \left\langle [\delta v_\parallel(r)]^p \right\rangle , \tag{3.1}$$

where

$$\boldsymbol{v}(\boldsymbol{x}) \equiv \rho^{\frac{1}{3}}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}) \tag{3.2}$$

and Equation 3.1 uses the same definitions as Equations 1.1 and 1.2. $\mathcal{S}_3(r)$ is dimensionally appealing (Fleck 1996; Kritsuk et al. 2007b), even if not rigorously derived, and reduces to Kolmogorov's four-fifths law in the case of incompressibility. To measure two-point statistics, such as the different orders of structure functions Equation 3.1, SFgen samples data from many pairs of points to create ensemble averages. Using a set of N random points $\boldsymbol{x}_i$ and unit vectors $\hat{\boldsymbol{r}}_i$, the expected value of $\mathcal{S}_3(r)$ can be measured as

$$\mathcal{S}_3(r) = \frac{1}{N}\sum_{i=1}^{N} \left[\delta v_\parallel(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2})\right]^3 , \tag{3.3}$$

where

$$\delta v_\parallel(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}) = [\boldsymbol{v}(\boldsymbol{x}_{i,2}) - \boldsymbol{v}(\boldsymbol{x}_{i,1})] \cdot \hat{\boldsymbol{r}}_i \tag{3.4}$$

and

$$\boldsymbol{x}_{i,2} \equiv \boldsymbol{x}_{i,1} + r\hat{\boldsymbol{r}}_i. \tag{3.5}$$

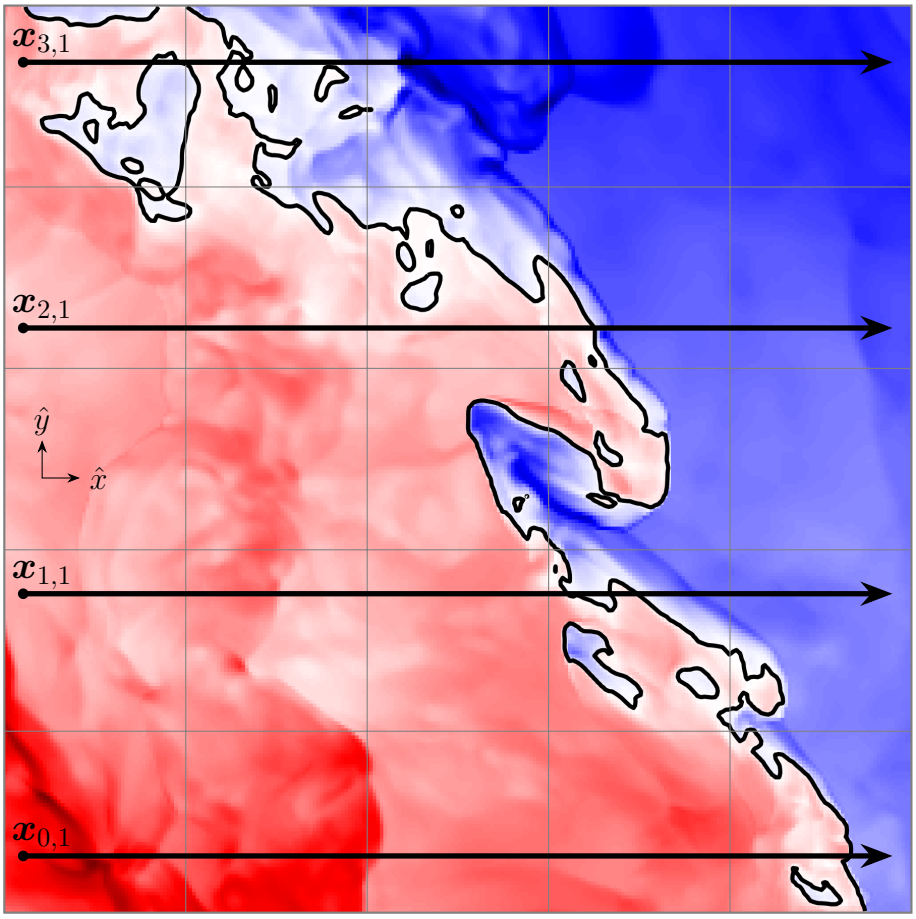

Figure 3.1: Diagram of extracting values along a path from a point. The background image is of the density-weighted field $\rho^{1/3}u_x$, and the black contour line represents $\rho^{1/3}u_x = 0$. The region to the left of the contour is moving to the right ($u_x > 0$) and vice versa.

As a simple example, Figure 3.1 shows paths along $\boldsymbol{r} = r\hat{x}$ originating at four points, $\boldsymbol{x}_{0,1} \ldots \boldsymbol{x}_{3,1}$, while Figure 3.2 shows the values of the two-point function $\delta v_x(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,1} + r\hat{x})$ along those paths. (Obviously, the points $\boldsymbol{x}_{i,1}$ in this example are not random.)

Starting from this and ignoring the realities of dealing with computers described in the rest of this chapter, I can describe the purpose of SFgen (somewhat) briefly: Use simple random sampling, with replacement, to measure the expected values, μ, variances, σ^2, and PDF (probability density function), $\mathbb{P}$, for a set of scaling relations, and separations, $r \in \mathbb{L}$. The individual samples are the output of a particular two-point function, F_f, using the values of the physical fields ($\boldsymbol{\phi}_{i,1}$ and $\boldsymbol{\phi}_{i,2}$) drawn from two points ($\boldsymbol{x}_{i,1}$ and $\boldsymbol{x}_{i,2}$) related

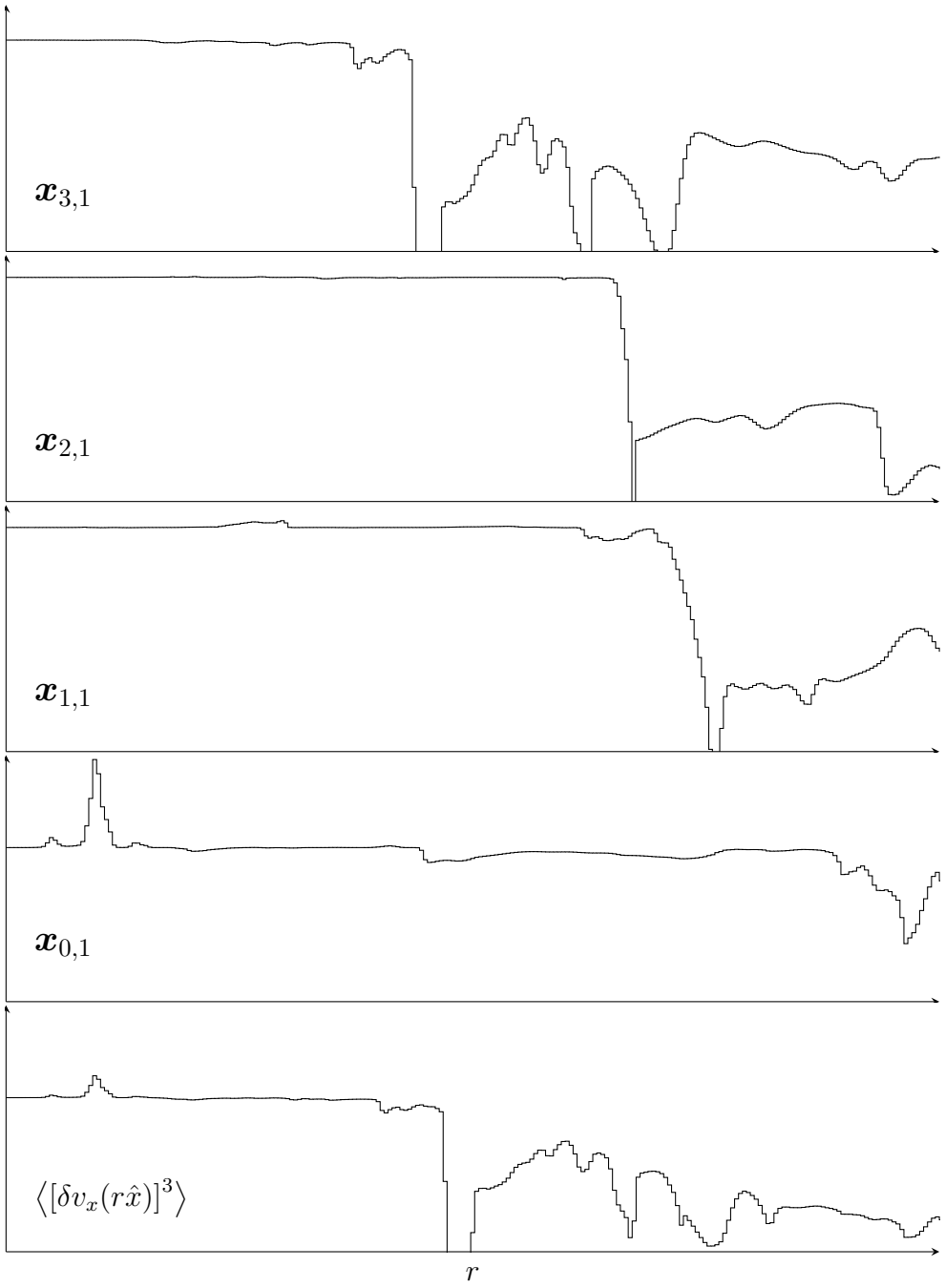

Figure 3.2: Extracted values and the average for $[\delta v_x(\boldsymbol{x}_1, \boldsymbol{x}_1 + \boldsymbol{r})]^3$ along four paths.

through Equation 3.5 as input. E.g.,

$$\boldsymbol{\phi}_{i,1} = \boldsymbol{\phi}(\boldsymbol{x}_{i,1}) = (\rho(\boldsymbol{x}_{i,1}), \boldsymbol{u}(\boldsymbol{x}_{i,1})) \tag{3.6}$$

$$\boldsymbol{\phi}_{i,2} = \boldsymbol{\phi}(\boldsymbol{x}_{i,2}) = (\rho(\boldsymbol{x}_{i,2}), \boldsymbol{u}(\boldsymbol{x}_{i,2})) \tag{3.7}$$

I denote $\mu_{f,r}$ and $\sigma^2_{f,r}$ as the "true", or population, expected value and variance, respectively, for a particular two-point function, F_f, and separation, r, while the sample mean and variances are denoted as $\hat{\mu}_{f,r}$ and $\hat{\sigma}^2_{f,r}$. The measurement of $\hat{\mu}_{f,r}$ is

$$\hat{\mu}_{f,r} = \frac{1}{N_\Sigma} \sum_{i=1}^{N_\Sigma} F_f(\boldsymbol{\phi}_{i,1}, \boldsymbol{\phi}_{i,2}), \tag{3.8}$$

where N_Σ is the total number of point pairs sampled at that separation (see Section 3.3.3). The variances are determined using

$$\hat{\sigma}^2_{f,r} = \left\{ \frac{1}{N_\Sigma} \sum_{i=1}^{N_\Sigma} \left[F_f(\boldsymbol{\phi}_{i,1}, \boldsymbol{\phi}_{i,2}) \right]^2 \right\} - \left(\hat{\mu}_{f,r} \right)^2 . \tag{3.9}$$

Each element of $\mathbb{P}_{f,r,b}$ represents the estimated likelihood of a value occurring between in the range

$$\mathbb{Q}_{f,r,b} \leq F_f(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \leq \mathbb{Q}_{f,r,b+1} \tag{3.10}$$

where b is the PDF bin with a left edge of $\mathbb{Q}_{f,r,b}$ and a right edge of $\mathbb{Q}_{f,r,b+1}$. Higher order moments for some two-point statistics may be calculated from the PDFs.

Restated more briefly: sample as many random points as necessary and bin the values for different functions using the data from those points as input. The two challenges to this approach are the a priori unknown variances in the PDFs of the two-point functions and the actual realities of dealing with computers.

3.2 The Need for Parallelism

Besides the requirement for SFgen to be incorporated within Enzo to provide in-line analysis, the number of samples needed to reduce the standard error of the mean to an acceptable level makes parallelism necessary. A deeper discussion around the number of samples needed to estimate statistical properties of the two-point functions of interest is covered in Section 3.9. For now, we'll use an example from one of the publications which used this method, Kritsuk, Wagner, and Norman (2013), described in Chapter 6. For that work, 2×10^9 point pairs were sampled at 16 different separations in each of 86 datasets. The two-point functions measured used 23 64-bit floating point numbers at every point pair. This analysis required communicating and processing 5.4 TB to analyze a single dataset and 460.5 TB across all of the datasets.

The datasets were uniform grids with 1024^3 uniform cubic cells totaling 56 GB after being read into memory. This may seem at odds with the two orders of magnitude greater data needed. To explain this, consider a measurement of point pairs at a separation of $1/16L$, where $L = 1$ is the length of the grid along one axis. To sample every possible set of values, one method would be to visit every cell, and then take samples at every cell intersecting a sphere of radius 1/16 from that cell, similar to the `fastSF` (Sadhukhan, Bhattacharya, and Verma 2021) algorithm. We can estimate the number of cells intersected by the sphere by looking at the surface areas of the sphere and the cells. The area of a sphere with a radius of 1/16 is $A = \pi/64 \approx 0.05$ and the cell faces are $a = (1/1024)^2 \approx 10^{-6}$ in area. The number of intersected cells would be around $A/a \approx 5 \times 10^4$. Using this method, the number of pairs would be $5 \times 10^4 \times 1024^3 \approx 5.4 \times 10^{13}$ and the amount of data processed would be $5 \times 10^4 \times 1024^3 \times 8 \times 23 \approx 8.8$ PB, where 8 is the precision in bytes of the 23 values needed per pair. Doing this for every separation and dataset would take this into exabytes.

Fortunately, robust measurements do not necessarily require all possible information.

Reasonable estimates of the probability density functions, expected values, and variances can be made from subsets of the total possible point pairs. Hence, the use of random sampling in SFgen. However, there is still a significant amount of data to compute, even for a single dataset. This leads to the use of parallelism and distributing the workload across many processors to complete the measurements quickly. The tradeoff with parallelism is the ensuing need to communicate data between computers. Section 3.6 gets into those details and describes how the tasks request and receive information.

3.3 Design & Overview

This section describes some design guidelines I used, followed by the parameters controlling SFgen and an overview of its outer and inner loops. For the algorithms, the pseudocode for the various functions are in their own subsections for reference.

3.3.1 Design Considerations

Similar to experimental science and engineering, efficient computational science is a compromise between expertise in computer science and expertise in theoretical science. Efficiency, in this case, is defined practically, in terms of a human's (my) time to develop the algorithms and the time to execute them and measure the values of interest. While the algorithms and overall structure of SFgen evolved over time, some stable principles guided it. Whenever possible, I tried to

- avoid traversing the hierarchy,
- operate on one-dimensional arrays from beginning to end,
- and limit the number of communication calls.

These principles came primarily from experience, both my own and those of other Enzo developers. In addition, there were the usual needs when creating a parallel application, like balancing the workload across the tasks so that they enter communication phases simultaneously.

Traversing the Hierarchy The grid hierarchy can contain hundreds of thousands or even millions of grids. And because it's implemented as a linked list, the entries are not guaranteed to be contiguous in memory. As I'll describe in Section 3.4.2, repeatedly traveling along its edges as part of an inner loop can add a significant number of operations. If it was necessary to use the hierarchy, I tried to do it a single time, e.g., part of an initialization, or to reduce the number of potential paths along its graph.

Operating on 1D Arrays Flexible data structures are convenient but can also be inefficient if the attributes are not local in memory (cache misses). One-dimensional arrays of a single datatype are also simpler to work with as part of MPI communications.

Limiting MPI Calls MPI calls have an up-front cost for each call due to the communication latency between tasks and the need for the library to track and manage the state of the call through its completion. These costs reduce the effective communication bandwidth between tasks. By sending as much data as possible in the fewest number of calls, a higher effective bandwidth rate is achieved, and there are fewer steps in an application to be blocked waiting for an MPI call to complete. In particular, all-to-all MPI communications provide a mechanism for all of the tasks to exchange data simultaneously in a single call. This principle leverages both the bidirectional networks and any optimizations in the MPI implementation (see Appendix B).

3.3.2 Two-Point Functions & Parameters

The code has flags to determine which of the two-point functions, F_f will be computed during a particular analysis run. In practice, this was usually "all of them" since the time to gather and communicate the physical fields from the point pairs was much greater than the time to compute a few more functions. Given the set of F_f to be used, the PDF arrays, $\mathbb{P}_f$, are instantiated on each task, using the following parameters:

F_f, the two-point function being measured for this array.

N_L The number of separation distances in $\mathbb{L}$.

N_B The number of PDF bins for each two-point function.

Section 3.7 describes how $\mathbb{P}$ is populated.

3.3.3 Sampling Parameters

The number of point pairs sampled is determined by the following parameters:

$\mathbb{L}$ The set of separations, r, between points, e.g., $r \in (L/32, L/16, \ldots)$.

N_L The count of separations in $\mathbb{L}$.

N_P The number of point pairs to use in each sampling iteration.

N_M The number of passes through MakePass (see Section 3.3.5).

N_T The number of MPI tasks.

The total number of point pairs sampled for each separation will be the product of the number of tasks, passes, and pairs: $N_T \times N_P \times N_M$. Maximizing N_P is largely based on memory constraints both the arrays allocated by the application and the buffers used by MPI. To achieve the total number of points needed, N_T or N_M can be increased. The

underlying analysis framework also supports setting the maximum depth of the hierarchy to use for analysis, which will limit the spatial resolution of the cells sampled. This will impact the results of the analysis, but does not change the overall method.

3.3.4 The Outer Loop

The broad outline of calculating the two-point functions is relatively simple: first initialize some necessary data structures; iterate many times sampling points and making individual measurements; summarize the data and write it out.

- Initialization
  - Compute the velocity divergence, $\boldsymbol{\nabla} \cdot \boldsymbol{u}$.
  - Initialize FastFind indices and data structures (see Section 3.4).
  - Allocate $\mathbb{G}$, $\mathbb{I}$, and $\hat{\boldsymbol{r}}$, the arrays for the point pairs (see Section 3.5).
  - Initialize the parallel random number generator (see Section 3.5).
  - Initialize GatherCellValues grid index, $\mathbb{M}$, and count, $\mathbb{S}$, arrays (see Section 3.6).
  - Allocate $\mathbb{P}$, the arrays for two-point function PDFs (see Section 3.7).
- MakePass N_M times.
- ReduceData, consolidate the measurements across the tasks (see Section 3.8.)
- Write out the measurements (see Section 3.8.)

3.3.5 The Inner Loop: MakePass

MakePass is called N_M times, where each time N_P random point pairs for each separation are sampled and the cell values at those points are used to compute individual values the two-point functions. Each loop through MakePass does the following steps:

- GetPointPairs: For each separation in $\mathbb{L}$, generate N_P random point pairs (see Section 3.5). The algorithm FastFind (see Section 3.4) is used to find the grid containing each point.
- GatherCellValues: Each task requests and receives cell values for its points (see Section 3.6).
- CalculateValues: Compute two-point function values for individual point pairs and build the PDFs (described Section 3.7).

3.4 FastFind

Given a point, $\boldsymbol{x}$, in the simulation volume, the FastFind algorithm returns the most refined grid G containing $\boldsymbol{x}$. Consider from Section 3.2 that each analysis of a single dataset required extracting the cell values at 1.3×10^{11} points. Because FastFind only uses the information available in a task's copy of the hierarchy, it does not require any communication steps. I developed the FastFind algorithm as a simple and efficient mechanism to allow the mapping of a point in the simulation domain to the most highly-refined overlapping grid and cell. It combines a coarse-grained search from a lookup table with a depth-first traversal of the hierarchy. Since the goal is to extract the physical values from the most refined grids, the challenge is to create a computationally efficient process that will find one of the highly-refined grids when necessary while traversing as few of the other grids as possible. (Once the containing grid is found, finding the index of the grid cell is only a few operations.)

FastFind is designed around Enzo's data structures, in particular, the AMR hierarchy structure and Grid class; it's very likely this algorithm could be adapted to other block-structured AMR codes, such as RAMSES (Teyssier 2002). Likewise, there's no dependency on Cartesian coordinates, only that the domain decomposition of the coarsest grids follows a

pattern amenable to constant time lookup and that finer grids are linked to the overlapping coarser grid. A similar algorithm using chaining meshes was developed as the fast sibling grid search (Bryan et al. 2014; Norman et al. 2007) to identify overlapping grids for boundary condition updates. The methods differ because of the different goals of identifying all the overlapping grids of a single grid (i.e., the motivation for a chaining mesh) versus identifying a single grid overlapping a point (i.e., FastFind). Estimating the computational (time and space) efficiency of FastFind is somewhat challenging as it's highly dependent on the distribution of grids in the hierarchy. But, there are practical constraints on the search based on how simulations are structured, and this somewhat bounded estimate is given in Section 3.4.2.

Initialization FastFind uses a 3D array of grids, $\mathbb{F}$, as a lookup table, where L_F is the maximum level of the grids referenced in the array. The array is allocated to fill the simulation domain with cells sized at level $L_F - 1$. This leverages the constraints regarding the placement of grids between levels l and $l + 1$, since a grid at level L_F or lower must fully overlap a cell in $\mathbb{F}$. After allocation, the hierarchy is traversed in a breadth-first order so that grids at level l are assigned to $\mathbb{F}$, followed by grids at level $l + 1$, stopping after completing level L_F. This simplifies the creation of $\mathbb{F}$ since each grid can populate its entire overlapping region, with more refined grids overwriting the subsets of their parent grids' regions. $\mathbb{F}$ increases in size (and memory usage) as R^3, where R is the refinement factor, for each additional level of coverage. In practice, I found that covering the level 1 or 2 grids was the appropriate balance between time efficiency and the memory required.

Algorithm FastFind starts by computing the index, j_i, of the $\mathbb{F}$ cell containing x_i. It then assigns grid $G = \mathbb{F}_{j_i}$ as the starting point of the search. It then traverses the down the hierarchy (increasing l), looking for any more highly refined grids containing $\boldsymbol{x}$. As it descends the hierarchy, FastFind uses the function ContainingGrid described in Function 3.2

Table 3.1: Grids containing points in example.

Point	Grid
$\boldsymbol{A}$	$G_{0,1}$
$\boldsymbol{B}$	$G_{1,1}$
$\boldsymbol{C}$	$G_{2,0}$

to check if any subgrids contain $\boldsymbol{x}$. The pseudocode is shown in Function 3.1.

3.4.1 Example

Reusing the AMR grids from the previous examples, Figure 3.3 shows three points (A, B, C) within grids at different levels. By looking at the figure, we can determine which grid the FastFind algorithm should return, as shown in Table 3.1. We can compare our visual inspection with the behavior of FastFindGrid for each point in our example.

Grid Lookup This example assumes that the maximum level of $\mathbb{F}$, $L_F = 1$. $\mathbb{F}$ will have the dimensions 16×16 and can reference the $l = 1$ grids, $G_{1,i}$, but not grid $G_{2,0}$ at $l = 2$. Figure 3.4 shows the array and reference values of $\mathbb{F}$ for this example.

FastFindGrid($\boldsymbol{A}$)

1. j_i is assigned with the index of the cell in $\mathbb{F}$ containing $\boldsymbol{A}$
2. $G_{0,1}$ will be assigned to G from the lookup of $\mathbb{F}_{j,i}$
3. $G_{0,1}$ does not have any subgrids, so the algorithm will not enter the while loop
4. algorithm returns $G_{0,1}$

FastFindGrid($\boldsymbol{B}$)

1. j_i is assigned with the index of the cell in $\mathbb{F}$ containing $\boldsymbol{B}$

2. $G_{1,1}$ will be assigned to G from the lookup of $\mathbb{F}_{j_i}$

3. $G_{1,1}$ does have subgrids, so the algorithm enters the while loop

4. ContainingGrid$(G, \boldsymbol{B})$ loops over the subgrids of $G_{1,1}$ and returns `NULL` because none of $G_{1,1}$'s subgrids overlap $\boldsymbol{x}$

5. S is set to `NULL`

6. S evaluates as false, so the while loop is broken

7. Algorithm returns $G_{1,1}$

FastFindGrid$(\boldsymbol{C})$

1. j_i is assigned with the index of the cell in $\mathbb{F}$ containing $\boldsymbol{C}$

2. $G_{1,1}$ will be assigned to G from the lookup of $\mathbb{F}_{j_i}$

3. $G_{1,1}$ does have subgrids, so the algorithm enters the while loop

4. ContainingGrid$(G, \boldsymbol{C})$ loops over the subgrids of $G_{1,1}$ and returns $G_{2,0}$ assigned to S

5. G is set to $G_{2,0}$

6. $G_{2,0}$ does not have any subgrids, so the algorithm exits the while loop

7. Algorithm returns $G_{2,0}$

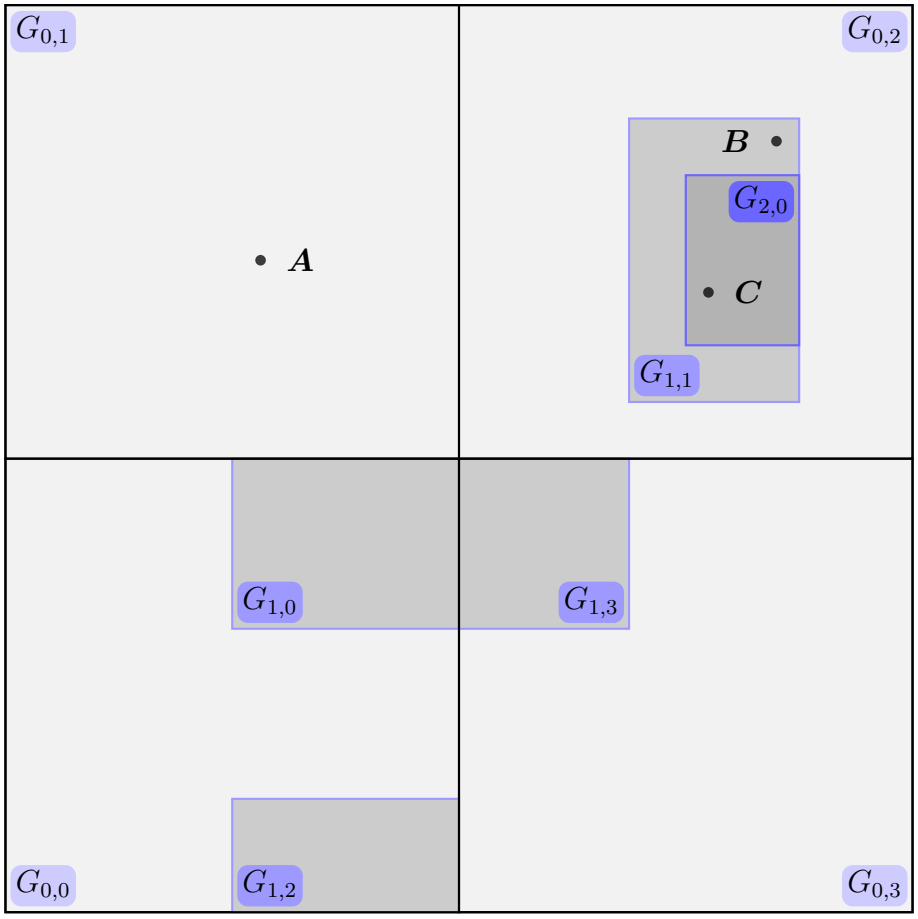

Figure 3.3: Points overlapping grids at different refinement levels.

$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$
$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$
$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{1,1}$	$G_{1,1}$	$G_{1,1}$	$G_{0,2}$	$G_{0,2}$
$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{1,1}$	$G_{1,1}$	$G_{1,1}$	$G_{0,2}$	$G_{0,2}$
$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{1,1}$	$G_{1,1}$	$G_{1,1}$	$G_{0,2}$	$G_{0,2}$
$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{1,1}$	$G_{1,1}$	$G_{1,1}$	$G_{0,2}$	$G_{0,2}$
$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{1,1}$	$G_{1,1}$	$G_{1,1}$	$G_{0,2}$	$G_{0,2}$
$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,1}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$	$G_{0,2}$
$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,3}$	$G_{1,3}$	$G_{1,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$
$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,3}$	$G_{1,3}$	$G_{1,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$
$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,0}$	$G_{1,3}$	$G_{1,3}$	$G_{1,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$
$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$
$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$
$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$
$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{1,2}$	$G_{1,2}$	$G_{1,2}$	$G_{1,2}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$
$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{0,0}$	$G_{1,2}$	$G_{1,2}$	$G_{1,2}$	$G_{1,2}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$	$G_{0,3}$

Figure 3.4: The grids referenced in the FastFind lookup array $\mathbb{F}$ using the example AMR grids and $L_F = 1$. Array regions with level 1 grids are called out with a darker border.

Table 3.2: Hierarchy statistics from a seven-level AMR simulation.

Level	$n_l = \Sigma_i G_{l,i}$	n_{l-1}/n_l	V_l	V_{l-1}/V_l
0	256		1	
1	153,680	600	0.21	0.21
2	98,976	0.64	0.012	0.058
3	70,833	0.72	8.2E-04	0.069
4	51,155	0.72	5.3E-05	0.065
5	17,094	0.33	1.6E-06	0.031
6	871	0.05	7.2E-09	0.0044

3.4.2 Performance & Limitations

FastFind relies on Enzo's spatial decomposition of the simulation by grids for a constant-time lookup as the first step. This can lead to an orders of magnitude performance improvement when compared to a simple traversal of the hierarchy. We can make a practical estimate of FastFind's efficiency using an AMR simulation with a modest number of possible refinement levels (7, although only 6 were used) and almost $400,000$ grids (Hallman et al. 2007). The root grid of the simulation was 512^3 cells, broken up into 256 level 1 grids of $64 \times 64 \times 128 = 524,288$ cells.

Table 3.2 has some basic statistics about the grids at each level of refinement for the final dataset at redshift $z = 0$. In the table, $\Sigma_i G_{l,i}$ is number of grids at a given refinement level, V_l is the volume covered by the grids at that level, and V_{l-1}/V_l is the relative fraction of the volume covered by the next level of refinement. From this we can see that $l = 1$ covers $\sim 20\%$ of the simulation volume and $l = 2$ covers $\sim 1\%$ of the full volume and $\sim 6\%$ of the $l = 1$ grids.

In this example dataset, there are $153,680$ total level 1 grids, with an average of 600 level 1 grids somewhat uniformly distributed amongst the level 0 grids. If $L_F = 0$, $\mathbb{F}$ will be allocated to cover a grid of $8 \times 8 \times 8 = 512$ cells. 79% of grid lookups would lead to a traversal of approximately 600 subgrids without a result. For the latter case of the other

21% of searches, this would lead to around 300 steps along the level 1 hierarchy, or roughly $0.79 \times 600 + 0.21 \times 300 = 537$ steps per point. The average number of subgrids past level 1 is than one, but there would be frequent traversals a few more steps beyond this level, with $\geq 94\%$ being missing. Instead, setting $L_F = 1$ requires 1GB of memory on a 64-bit system, but allows $\mathbb{F}$ to reference the level 2 grids, and results in a constant time lookup for $\sim 99\%$ of points searched for, a two orders of magnitude improvement.

Like the $L_F = 0$ example for the previous simulation, for unigrid simulations (ones with only level 0 grids), $\mathbb{F}$ can be set to the minimum size necessary based on how level 0 is partitioned.

An alternative version of FastFindGrid, is shown in Function 3.3. This checks a grid's `FlaggingField` array to see if a subgrid overlaps the cell containing $\boldsymbol{x}$. This avoids traversing the hierarchy when a grid has subgrids, but none of them overlap $\boldsymbol{x}$, like point $\boldsymbol{B}$ in the example. It could be more efficient in the case of many small highly-refined regions, like when modeling stellar collapse. However, it requires an additional setup step where every grid at level L_F or beyond allocates, initializes, and sets its `FlaggingField` based on the location of its subgrids. Figure 3.5 shows the field for grid $G_{0,0}$.

The cost in time of the initialization is relatively minimal, but the cost in memory of the `FlaggingField` could be substantial since it covers all or some of the cells in the simulation. If implemented, making a separate `FlaggingField` as an array of booleans (single bits per element) could overcome this drawback. By default, the `FlaggingField` is a set of integers so that it can be used to track not just if but why a cell requires refinement. Anecdotally, I can say that I tried both versions, but that FastFindGrid, as first described, was sufficient and required less work to handle the setup.

0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1

Figure 3.5: The flagging field $G_{0,0}$.`FlaggingField`.

3.4.3 Pseudocode

Function 3.1: FastFindGrid finds the most refined grid containing a point. Δ_F, is grid cell width of the lookup array $\mathbb{F}$. Note that ***floor*** or $\lfloor\rfloor$, is used to denote that the result of $x_i/\Delta_{F,i}$ is an integer. ContainingGrid is shown in Function 3.2.

```
Function FastFindGrid(x : point)
    j_i = ⌊ x_i / Δ_{F,i} ⌋
    G = 𝔽_{j_i}
    while G.NextGridNextLevel do
        S = ContainingGrid(G.NextGridNextLevel, x)
        if S then
            G = S
        else
            break
        end
    end
return G
```

Function 3.2: ContainingGrid returns the first sibling grid of G containing a point, or `NULL` if none do.

```
Function ContainingGrid(G : grid, x : point)
    while G do
        if x_i ∈ G then
            break
        end
        G = G.NextGridThisLevel
    end
    return G
```

Function 3.3: FastFindFlaggedGrid finds the most refined grid containing a point. Δ_F, is grid cell width of the lookup array $\mathbb{F}$. Note that ***floor*** or $\lfloor\rfloor$, is used to denote that the result of $x_i/\Delta_{F,i}$ is an integer. ContainingGrid is shown in Algorithm 3.2.

```
Function FastFindFlaggedGrid(x : point)
    j_i = ⌊x_i / Δ_{F,i}⌋
    G = 𝔽_{j_i}
    while G.NextGridNextLevel do
        k = G.GetGridCellIndex(x)
        if G.FlaggingField[k] then
            // There is a subgrid overlapping x
            // ContainingGrid will return a grid, not NULL
            G = ContainingGrid(G.NextGridNextLevel, x)
        else
            break
        end
    end
return G
```

3.5 GetPointPairs

Despite its name, GetPointPairs is used to generate paired sets of grids and cell indices, along with a unit vector. These pairs are selected by finding the most refined grid cells containing each of two points separated by a fixed distance r along a random direction. Only cell indices are used because the PPM fluid dynamics solver uses a finite volume method based on the Godunov (1959) scheme, and each cell has a constant value through its volume. Rather than make (likely incorrect) assumptions about the physical quantities at a point using interpolation, I chose to use the same value as the dynamical method. The time and space performance of GetPointPairs is linear. SFgen can make multiple passes to cap the memory usage. Like FastFind, GetPointPairs works with task-local data to avoid unnecessary communications.

The function is straightforward and its steps are shown using pseudocode in Functions 3.5 and 3.4. To summarize:

- A set of separation distances $\mathbb{L}$ and the number of pairs N_P is defined
- For each separation distance $r \in \mathbb{L}$, N_P pairs of points are created, each with the following vectors:
  - a random unit vector, $\hat{\boldsymbol{r}}$;
  - an initial point, $\boldsymbol{x}_1 \in \mathbb{D}$;
  - a second point, $\boldsymbol{x}_2 = \boldsymbol{x}_1 + r\hat{\boldsymbol{r}}$.
- The algorithm assumes periodic boundary conditions, and point $\boldsymbol{x}_2$ is wrapped back into the domain if it falls outside.
- The grids containing $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are found using the function FastFindGrid, described in Section 3.4

- The indices of the grid cells containing $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are calculated
- The grids, indices, and $\hat{\boldsymbol{r}}$ are stored in the arrays $\mathbb{G}$, $\mathbb{I}$, $\mathbb{R}$, respectively

3.5.1 Example

Figure 3.6 shows point pairs at different separations in the example AMR grids.

3.5.2 Selecting Points

Random Numbers Each point pair requires five random numbers (two angles and three coordinates) and a typical dataset required on the order of 10^9 random point pairs to estimate the PDF of the high-order two-point functions. This is on the edge of the period a random number function using 32-bit integers will provide, and some implementations of the ISO C `rand` function may be much less (32,767). Combined with the need to instantiate one or more independent streams of pseudorandom numbers on each MPI task, an MPI-aware pseudorandom number generator with periods much greater than 10^9 was desirable. Fortunately, SPRNG (Mascagni and Srinivasan 2000) is a parallel random generator that handles this issue. SPRNG provides several random number generators; the one I used was the 64-bit linear congruential generator (LCG). Each stream from this generator has a period of 2^{64}, and the total number of possible streams is more than 10^8. This is more than adequate for the number of points needed and MPI tasks used. Calls to SPRNG are shown as sprng() in Function 3.4.

Random Directions Note that there are several possible means to create a random unit vector[2], the important part is to sample the sphere by unit area, not angle (Cook 1957). A justification for spherical sampling is shown in Figure 3.7, as compared to sampling points

[2]A summary of various methods for this can be found at MathWorld http://mathworld.wolfram.com/SpherePointPicking.html

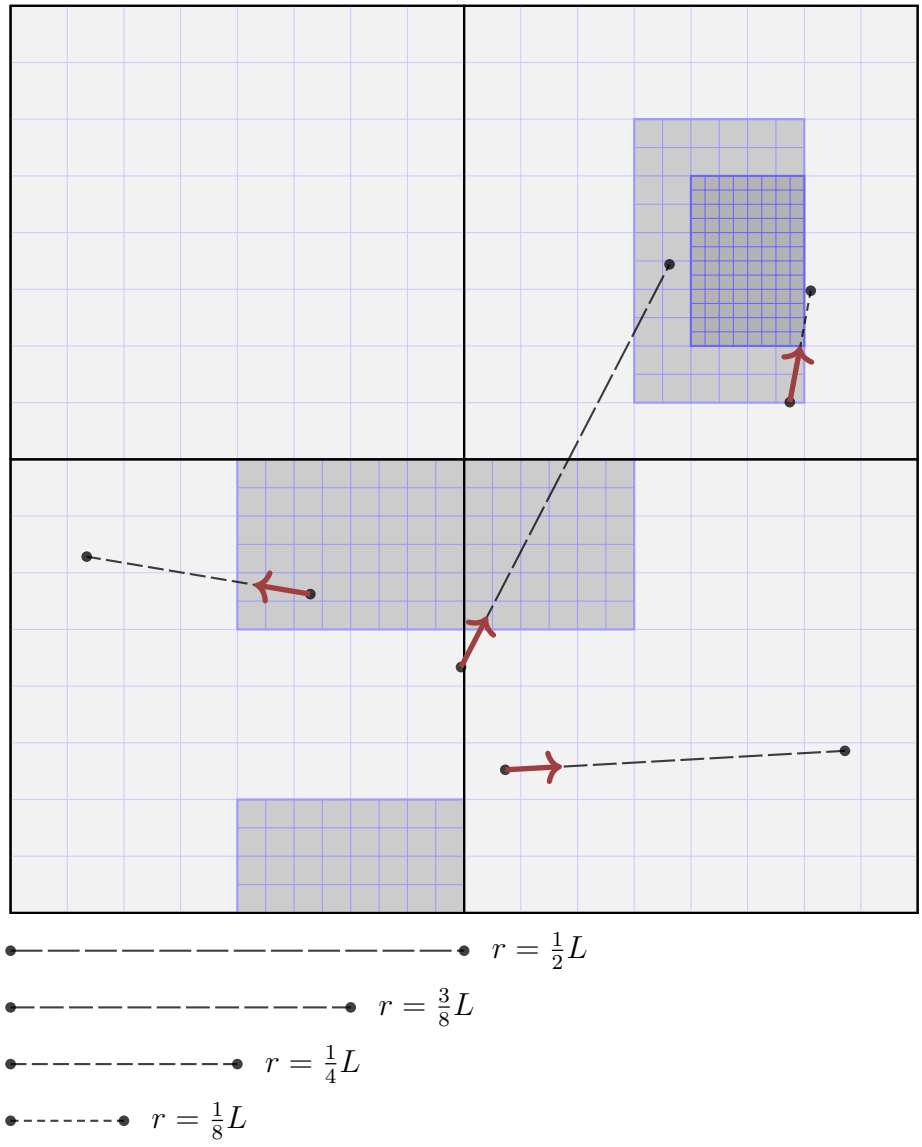

Figure 3.6: Random point pairs at four fractions of L, the length of a domain edge. Red arrows point from $\boldsymbol{x}_1$ along $\hat{\boldsymbol{r}}$. The pairs are overlayed on the example grids.

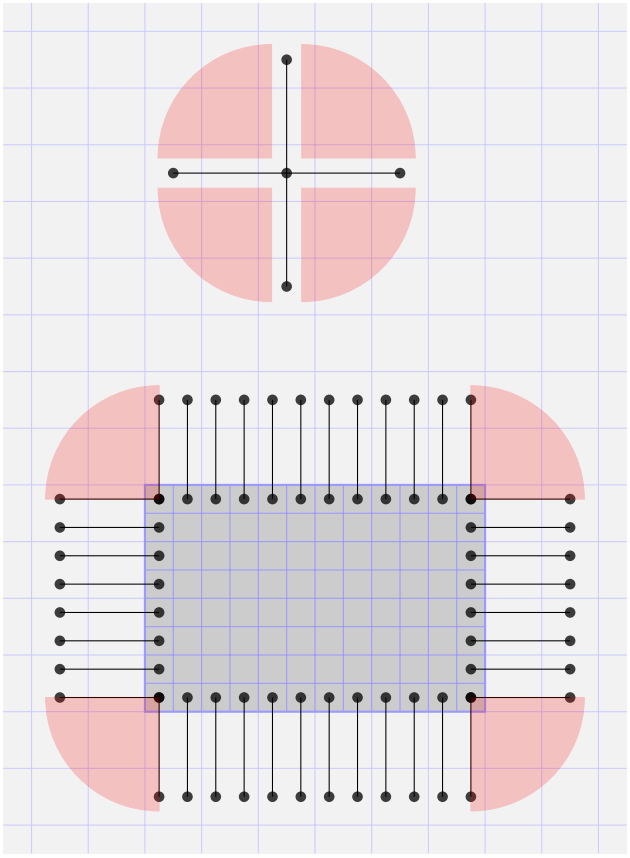

Figure 3.7: Point pairs sampled only along Cartesian axes. Top: A single grid cell compared to cells at the same refinement level. Bottom: Cells in a refined grid compared to cells at a higher level. Red shaded areas indicate regions not included in comparison.

only separated along the Cartesian axes, $\hat{\boldsymbol{x}}_i$. This method is convenient when dealing with uniform grids since it can be done by using arithmetic on array indices (Porter, Pouquet, and Woodward 2002). But, it does not include all possible points for comparison.

3.5.3 Pseudocode

Function 3.4: GetPointPair creates two random points within the simulation domain $\mathbb{D}$, separated by the distance r. It returns the grids containing the points, their respective cell indices, and the unit vector along their separation vector. G.GetCellIndex($\boldsymbol{x}$) is a grid method that computes the local cell index assuming a point $\boldsymbol{x} \in G$.

Function GetPointPair(r : distance)
  $\phi = 2\pi$ sprng()
  $z =$ sprng()
  $\theta = \cos^{-1}(2z - 1)$
  $\hat{\boldsymbol{r}} = (\sin(\theta)cos(\phi),\ \sin(\theta)\sin(\phi),\ \cos(\theta))$
  for $i \in (0, 1, 2)$ **do**
    $x_{1,i} =$ sprng()
  end
  $\boldsymbol{x}_2 = \boldsymbol{x}_1 + r\hat{\boldsymbol{r}}$
  `// Assuming periodic boundary conditions,`
  `// wrap or shift` $\boldsymbol{x}_2$ `back into` $\mathbb{D}$ `if it falls outside.`
  `// Can't use just modulo, in case a component of` $\boldsymbol{x}_2$ `is negative.`
  if $\boldsymbol{x}_2 \notin \mathbb{D}$ **then**
    for $i \in (0, 1, 2)$ **do**
      if $x_{2,i} < 0$ **then**
        $x_{2,i} += \mathbb{D}_i$
      else
        $x_{2,i} -= \mathbb{D}_i$
      end
    end
  end
  $G_1 =$ FastFindGrid($\boldsymbol{x_1}$)
  $j = G_1$.GetCellIndex($\boldsymbol{x_1}$)
  $G_2 =$ FastFindGrid($\boldsymbol{x_2}$)
  $k = G_2$.GetCellIndex($\boldsymbol{x_2}$)
return $(G_1, G_2, j, k, \hat{\boldsymbol{r}})$

Function 3.5: GetPointPairs populates the grid pair, grid cell indices, and unit vector arrays $\mathbb{G}$, $\mathbb{I}$, and $\mathbb{R}$, respectively.

Function GetPointPairs($\mathbb{L}$: set of distances, N_P : number of pairs)
  for $r \in \mathbb{L}$ **do**
    for $p \in (0, 1, \ldots, N_P - 1)$ **do**
      $(G_1, G_2, j, k, \hat{\boldsymbol{r}}) = \mathsf{GetPointPair}(r)$
      $\mathbb{R}_{r,p} = \hat{\boldsymbol{r}}$
      $\mathbb{G}_{r,p,1} = G_1$
      $\mathbb{G}_{r,p,2} = G_2$
      $\mathbb{I}_{r,p,1} = j$
      $\mathbb{I}_{r,p,2} = k$
    end
  end
return $(\mathbb{G}, \mathbb{I}, \mathbb{R})$

3.6 GatherCellValues

> Basically a hellacious amount of bookkeeping.
>
> R. Wagner

GatherCellValues orders the grids by task to allow a simultaneous exchange of information between all the tasks to distribute an arbitrary set of a grid's cell values to any set of tasks. Enzo does not maintain a global array index across the hierarchy since the hierarchy and level arrays are built around the AMR model, where grids will change over time with the refinement needs and the solvers work level-by-level. By assigning a unique integer to each grid, ordered by the task rank (T), grid data can be further ordered to be uniquely indexed in an array (essentially "grid-major order"). These arrays can then be used in MPI all-to-all communications. This indexing strategy is used throughout the communications process to pack data into large arrays.

3.6.1 Grid Indexing

The index is assigned to each grid as G.`Index` using a depth-first traversal of the hierarchy by following G.`NextGridNextLevel`, then following G.`NextGridThisLevel`. The indices for each task's grids are incremented with assignment (see Functions 3.8 and 3.9). The starting index, I_p, for a task of rank p's grids is, therefore, a sum of the total grids owned by tasks of rank $\leq T_p$.

$$I_p = \sum_{i=0}^{p-1} \mathbb{S}_i, \tag{3.11}$$

where $\mathbb{S}$ is an integer array of the count of grids per task; $\mathbb{S}$ is populated in Function 3.6. For the example grids, $\mathbb{S} = (2, 3, 2, 2)$ and $I_p = (0, 2, 5, 7)$. Note that 0 is assigned to $G_{1,0}$,

Table 3.3: Indices assigned to the example grids as part of GatherCellValues.

Ordered by level and level index			Ordered by $G.\texttt{Index}$		
Grid	$G.\texttt{Index}$	Task	Grid	$G.\texttt{Index}$	Task
$G_{0,0}$	1	T_0	$G_{1,0}$	0	T_0
$G_{0,1}$	4	T_1	$G_{0,0}$	1	T_0
$G_{0,2}$	6	T_2	$G_{1,2}$	2	T_1
$G_{0,3}$	8	T_3	$G_{1,3}$	3	T_1
$G_{1,0}$	0	T_0	$G_{0,1}$	4	T_1
$G_{1,2}$	2	T_1	$G_{1,1}$	5	T_2
$G_{1,1}$	5	T_2	$G_{0,2}$	6	T_2
$G_{1,3}$	3	T_1	$G_{2,0}$	7	T_3
$G_{2,0}$	7	T_3	$G_{0,3}$	8	T_3

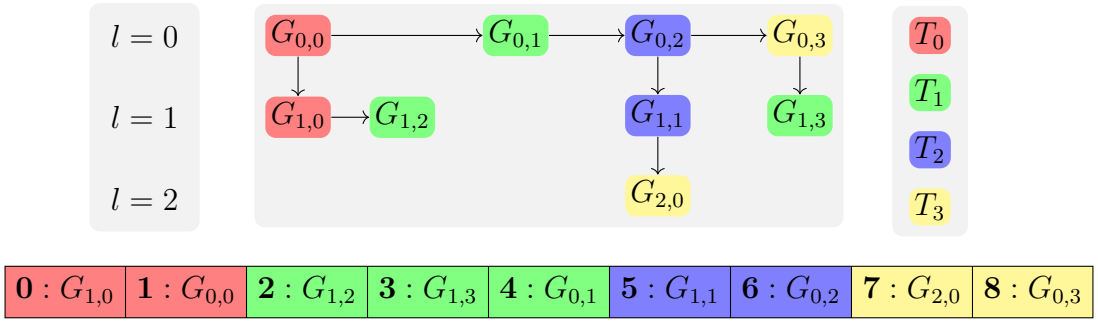

Figure 3.8: Top: The same as Figure 2.5, with the example hierarchy grids colored by task. Below: The grid index:grid mapping colored by task.

the subgrid owned by T_0, while

$$G_{1,2}.\texttt{Index} = I_1 = \mathbb{S}_0 = 2 \tag{3.12}$$

Table 3.3 shows the grid index as it would be assigned using the sample AMR grids and Figure 3.8 show indices of each grid ordered by task.

For an array ordered by grid index, computing the index of the first element for a given task depends on whether or not the count of elements per grid is uniform. For

uniform counts, where each grid has k elements, the starting index for task T_p is

$$I_{p,k} = k \sum_{i=0}^{p-1} \mathbb{S}_i. \tag{3.13}$$

For $k = 1$, this equivalent to Equation 3.11. When each grid has a distinct count of elements the index of element j for task T_p is

$$I_{p,j} = j + \sum_{i=0}^{p-1} \mathbb{K}_i \tag{3.14}$$

where $\mathbb{K}$ is an integer array with the counts of elements for each task. This is used in Functions 3.12 and 3.16, where the indices and values for requested cells are exchanged.

3.6.2 Process

GatherCellValues has several steps. The initialization phase is done once, and the remaining steps are executed in order during each pass of collecting cell values, after GetPointPairs. The individual algorithms are summarized in the following list, and each is detailed in its referenced pseudocode.

Initialization Prior to communications, GatherCellValues counts the number of grids per task, sets the grid indices (see Section 3.6.1), and creates an array of grids owned by the current task.

1. Allocate $\mathbb{S}$ and intialize its values to 0.
2. CountGridsPerTask: Count the grids per MPI task, populating the array $\mathbb{S}$ (Function 3.6).
3. Allocate $\mathbb{M}$ as an array of hierarchy entry pointers with length $\mathbb{S}_T$.

4. BuildMyGridArray: Populate $\mathbb{M}$ with the grids owned by T (Function 3.7).

5. IndexGridsByTask: Assign an index (G.`Index`) to each grid in the order of MPI task rank (Function 3.8). IndexGridsByTask uses a temporary array, $\mathbb{N}$, to track the last used index for each task's grids. The total number of grids, N_G, is found using the last index. This algorithm is run by all tasks rather than having one task communicate the results.

MakePass After GetPointPairs completes during each pass, GatherCellValues requests and receives the field values at each point's location.

1. Allocate $\mathbb{C}$, an integer array of length N_G, and intialize its values to 0.

2. CountGridCells: Populate $\mathbb{C}$ with a count of the cells in each grid containing points, ordered by grid index (Function 3.10). Cell indices may be repeated if multiple points are within a single cell.

3. CommunicateCellCounts: Share $\mathbb{C}$ to other MPI tasks. Also used to allocate send-receive arrays and calculate the counts and offsets used in communicating arrays (Function 3.11).

4. GetCellIndices: Build the array $\mathbb{J}$ of cell indices needed by the local task ordered by grid index and relative order of points (Function 3.12).

5. CommunicateCellIndices: Distribute $\mathbb{J}$ to other tasks, and receive requested cell indices into $\mathbb{K}$ (Function 3.13).

6. ReadInGrids: Iterate over the local task's grids, calling G.ReadCells (Function 3.15), to copy requested cell values into $\mathbb{U}$ (Function 3.14).

7. ReadCells: Iterate over cells, copying cell data into $\mathbb{U}$ (Function 3.15).

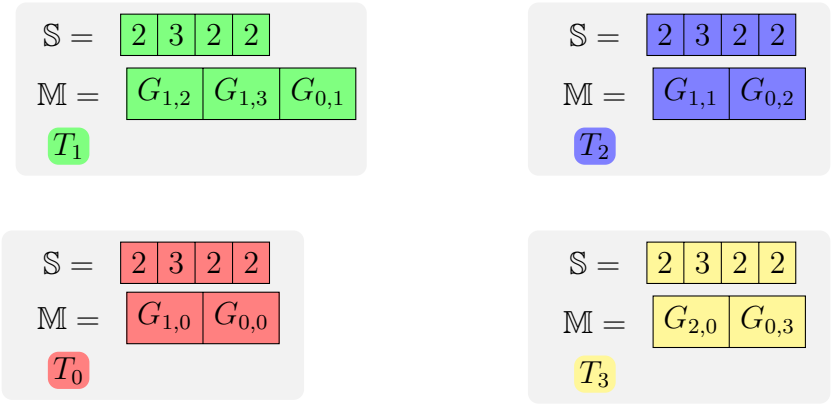

Figure 3.9: Grid count and local grid arrays after initialization.

8. CommunicateCellValues: Iterate over the grid's owned by the local tasks and populate $\mathbb{U}$ with the physical field cell values; communicate $\mathbb{U}$ into the receiving array, $\mathbb{V}$, on each task (Function 3.16).

9. ReadCellValues: Unpack the receiving arrays into $\mathbf{\Phi}$ (Function 3.17).

3.6.3 Example

This example goes through several of the GatherCellValues steps using the example grids and hierarchy. Figure 3.11 shows the grid count array, $\mathbb{S}$, and the array of grids owned by the local task, $\mathbb{M}$, after initialization before the start of MakePass and GatherCellValues. As a reminder, the grid indices are shown in Table 3.3.

For this example, we'll begin with a set of twelve random point pairs, one pair per task for each three separations $r/L \in (1/4, 3/8, 1/2)$, as shown in Figure 3.10. Table 3.4 lists the point pairs sorted by the owning task and separations, along with each point's containing grid and cell index.

Figure 3.11 shows the local cell count arrays, $\mathbb{C}$ and Figure 3.12 shows the requested cell counts $\mathbb{B}$ after the call to CommunicateCellCounts (Function 3.11). In CommunicateCellCounts, each task sends a number of elements from $\mathbb{C}$ equal to $\mathbb{S}_i$ to task T_i, so that

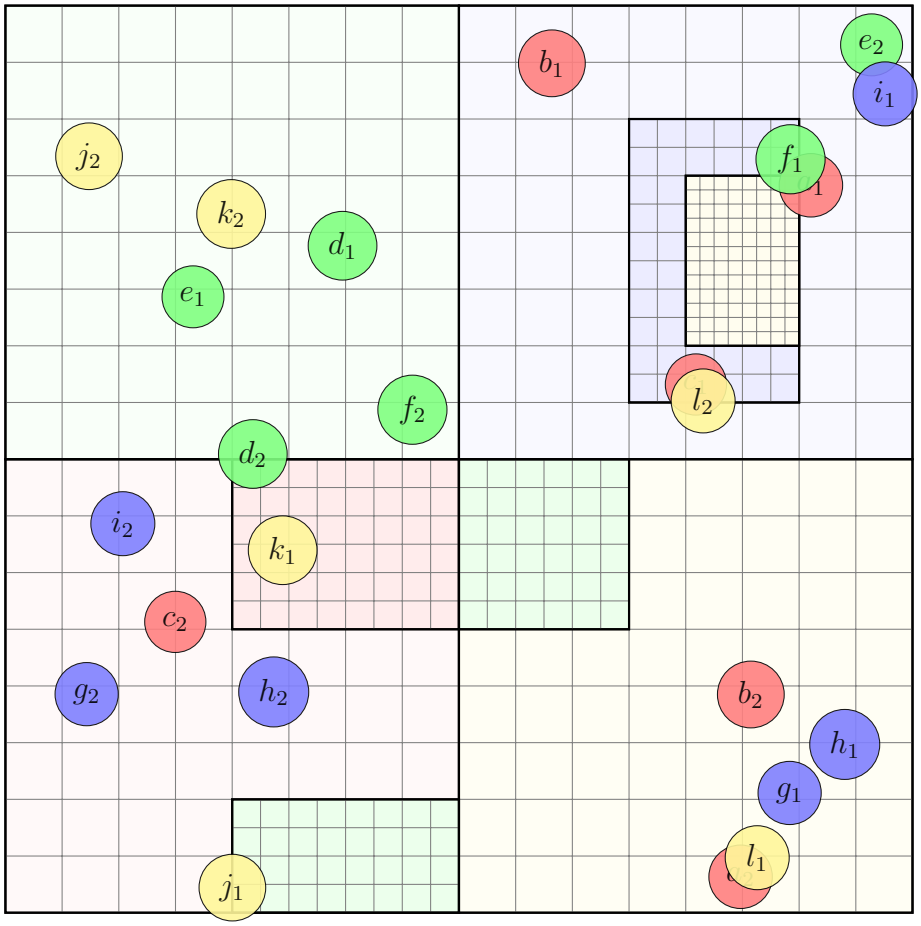

Figure 3.10: Twelve random point pairs, three for each of four tasks. Points are colored by owning task, and grids are colored by owning task and grid level, where coarser grids are lighter.

Table 3.4: List of points in Figure 3.10. Grids are indicated both by level and level index, as well as grid index. The unit vectors, $\mathbb{R}$, have been omitted.

Point	Task	r/L	$G_{r,1}$	$\mathbb{G}_{r,1}$	$\mathbb{I}_{r,1}$	$G_{r,2}$	$\mathbb{G}_{r,2}$	$\mathbb{I}_{r,2}$
a	T_0	1/4	$G_{0,2}$	6	22	$G_{0,3}$	8	3
b	T_0	3/8	$G_{0,2}$	6	31	$G_{0,3}$	8	15
c	T_0	1/2	$G_{1,1}$	5	1	$G_{0,0}$	1	20
d	T_1	1/4	$G_{0,1}$	4	16	$G_{0,1}$	4	3
e	T_1	3/8	$G_{0,1}$	4	14	$G_{0,2}$	6	35
f	T_1	1/2	$G_{1,1}$	5	28	$G_{0,1}$	4	5
g	T_2	1/4	$G_{0,3}$	8	10	$G_{0,0}$	1	13
h	T_2	3/8	$G_{0,3}$	8	17	$G_{0,0}$	1	15
i	T_2	1/2	$G_{0,2}$	6	29	$G_{0,0}$	1	31
j	T_3	1/4	$G_{1,2}$	2	0	$G_{0,1}$	4	25
k	T_3	3/8	$G_{0,0}$	1	27	$G_{0,1}$	4	20
l	T_3	1/2	$G_{0,3}$	8	3	$G_{1,1}$	5	1

each task T_i receives $N_T \times \mathbb{S}_i$ elements. The state of the grid cell index arrays, $\mathbb{J}$ and $\mathbb{K}$ are shown in Figure 3.13 after the call to CommunicateCellIndices (Function 3.13). The process to communicate the cell values in CommunicateCellValues (Function 3.16) is similar to CommunicateCellIndices and not shown.

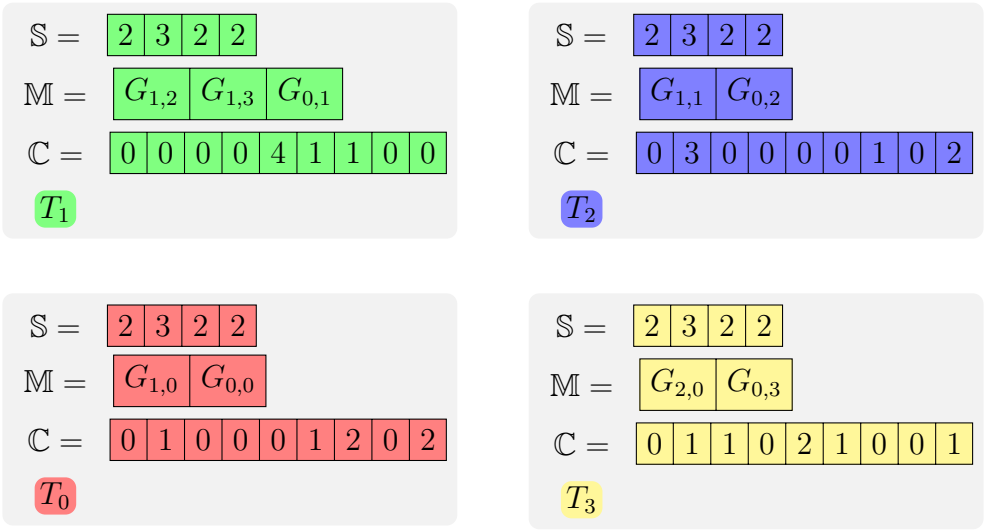

Figure 3.11: Arrays after grid cell count.

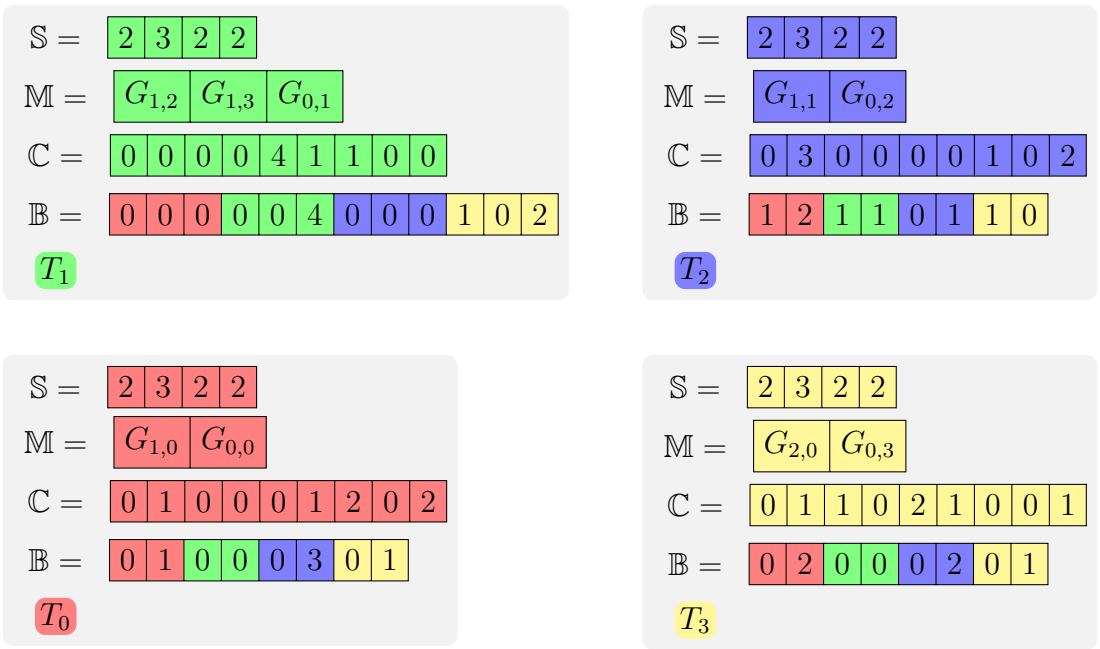

Figure 3.12: Arrays after communicating grid cell count.

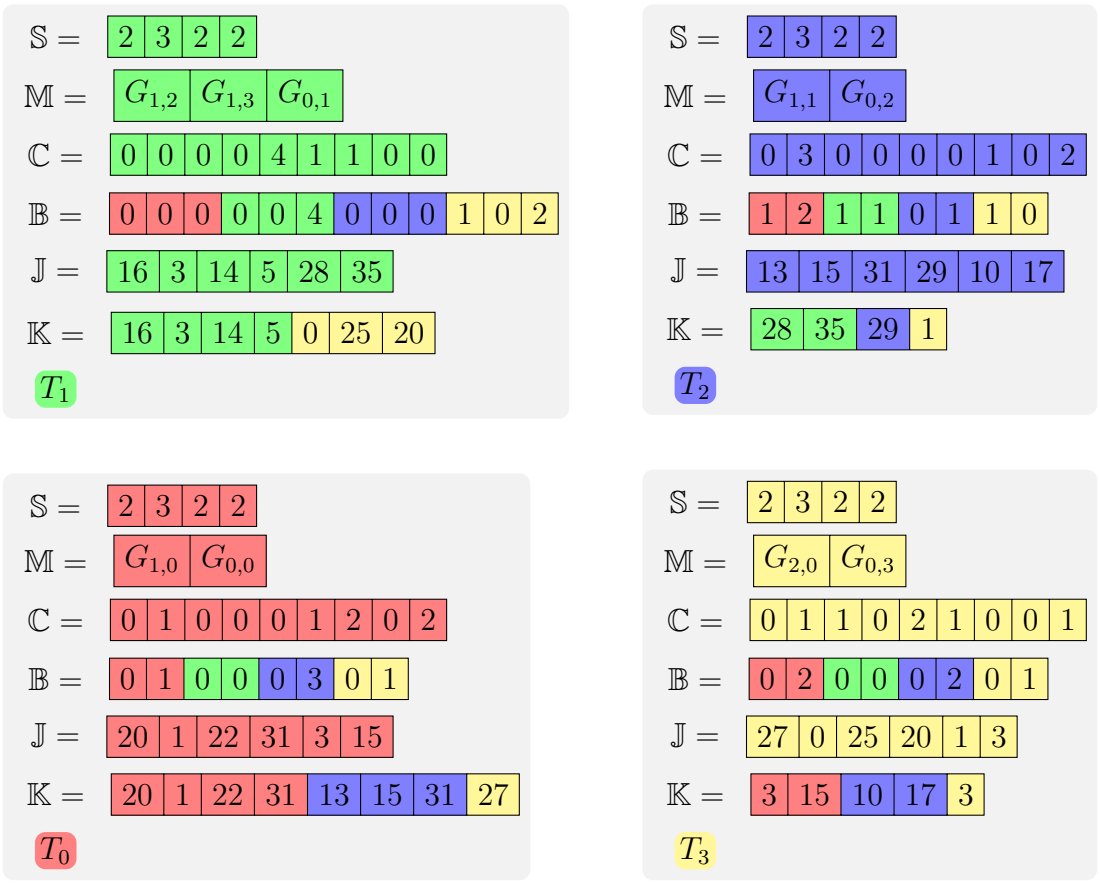

Figure 3.13: Arrays after communicating the grid cell indices, where $\mathbb{J}$ are the indices needed by each task, and $\mathbb{K}$ are the requested cell indices.

3.6.4 Pseudocode

Function 3.6: CountGridsPerTask recursively traverses the hierachy and populates the counting array $\mathbb{S}$ with the total number of grids owned by each task. The elements of $\mathbb{S}$ are intialized to 0 and the initial call is CountGridsPerTask($G_{0,0}, \mathbb{S}$).

```
Function CountGridsPerTask(G : grid, S : array of integers)
    while G do
        CountGridsPerTask(G.NextGridNextLevel, S)
        p = G.TaskNumber
        S_p++
        G = G.NextGridThisLevel
    end
```

Function 3.7: BuildMyGridArray populates the grid array $\mathbb{M}$ with the grids owned by the local task T. The initial call is BuildMyGridArray($G_{0,0}, \mathbb{M}, i = 0$).

```
Function BuildMyGridArray(G : grid, M : array of grids, i : integer)
    while G do
        BuildMyGridArray(G.NextGridNextLevel, M, i)
        if G.TaskNumber == T then
            M_i = G
            i++;
        end
        G = G.NextGridThisLevel
    end
```

Function 3.8: IndexGridsByTask assigns an index to each grid ordered by rank of the owning task. After allocation, the array $\mathbb{N}$ is initialized with a running total of the grids by task rank. Returns the total number of grids, N_G. The call is IndexGridsByTask($G_{0,0}, \mathbb{M}$).

Data: $\mathbb{N}$: an integer array of length N_T initialized to zero.
Function IndexGridsByTask(G : grid, $\mathbb{S}$: array of integers)
  for $t \in (1, 2, \ldots, N_T - 1)$ **do**
    $\mathbb{N}_t = \mathbb{N}_{t-1} + \mathbb{S}_t$
  end
  AssignGridIndices($G, \mathbb{N}$)
  $N_G = \mathbb{N}_{N_T}$
return N_G

Function 3.9: AssignGridIndices assigns an index to each grid from the array $\mathbb{N}$.

Function AssignGridIndices(G : grid, $\mathbb{N}$: array of integers)
  while G **do**
    AssignGridIndices(G.`NextGridNextLevel`, $\mathbb{N}$)
    $t = G$.`TaskNumber`
    G.`Index` $= \mathbb{N}_t$
    $\mathbb{N}_t++$
    $G = G$.`NextGridThisLevel`
  end

Function 3.10: CountGridCells counts the number of cells in each grid containing a point from the local task, T. Returns an array of the count indexed by grid, $\mathbb{C}$.

Function CountGridCells($\mathbb{G}$: array of grids, $\mathbb{C}$: array of integers)
  for $r \in \mathbb{L}$ **do**
    for $p \in (0, 1, \ldots, N_P - 1)$ **do**
      for $i \in (1, 2)$ **do**
        $G = \mathbb{G}_{r,p,i}$
        $g = G$.`Index`
        $\mathbb{C}_g++$
      end
    end
  end
return $\mathbb{C}$

Function 3.11: CommunicateCellCounts Returns $\mathbb{B}$.

Data: $\mathbb{B}$ an integer array of length $\mathbb{S}_T \times N_T$ with the count of cells in grids owned by the current task containing points owned by other tasks; t, u, v integer arrays of length N_T initialized to zero used to track the send counts, send offsets, receive counts, and receive offsets, respectively.

Function CommunicateCellCounts()

$u_1 = n = \mathbb{S}_T$

for $p \in (0, 1, \ldots, N_P - 1)$ **do**

$t_p = t_{p-1} + \mathbb{S}_{p-1}$

$u_p = pn$

$v_p = n$

end

MPI_Alltoallv($\mathbb{C}, \mathbb{S}, t, \ldots, \mathbb{B}, u, v, \ldots, \ldots$)

return $\mathbb{B}$

Function 3.12: GetCellIndices builds an array of the cells in each grid containing a point from the local task, T, ordered by the G.`Index`. Returns the array, $\mathbb{J}$.

Data: $\mathbb{J}$: an integer array of length $2N_P$ initialized to zero. O an integer array of length N_G initialized to zero

Function GetCellIndices()

for $i \in (1, 2, \ldots, N_G - 1)$ **do**

$O_i = O_{i-1} + \mathbb{C}_{i-1}$

end

for $r \in \mathbb{L}$ **do**

for $p \in (0, 1, \ldots, N_P - 1)$ **do**

for $i \in (1, 2)$ **do**

$G = \mathbb{G}_{r,p,i}$

$j = \mathbb{I}_{r,p,i}$

$g = G$.`Index`

$k = O_g$

$\mathbb{J}_k = j$

O_g++

end

end

end

return $\mathbb{J}$

Function 3.13: CommunicateCellIndices Returns $\mathbb{K}$.

Data: $\mathbb{K}$ an integer array of length $\Sigma\mathbb{B}_i$ with the indices of cells in grids owned by the current task containing points owned by other tasks; s, t, u, v integer arrays of length N_T initialized to zero used to track the send counts, send offsets, receive counts, and receive offsets, respectively.

Function CommunicateCellIndices()

```
m = n = 0                    // iterators over the grid indices G.Index
for i ∈ (0, 1, ..., N_T − 1) do
    s_i = 0                  // send count for task i starts at 0
    u_i = 0                  // recv count for task i starts at 0
    for j ∈ (0, 1, ..., 𝕊_i − 1) do
        s_i+ = ℂ_m  // add up all local cell counts for grids owned
         by task i
        m++                                                      //
    end
    for j ∈ (0, 1, ..., 𝕊_T − 1) do
        u_i+ = 𝔹_n
        n++
    end
end
for i ∈ (1, 2, ..., N_T − 1) do
    t_i = t_{i−1} + s_{i−1}
    v_i = u_{i−1} + v_{i−1}
end
MPI_Alltoallv(𝕁, s, t, ..., 𝕂, u, v, ..., ...)
return 𝕂
```

Function 3.14: ReadInGrids, copying N_ϕ from G.`PhysicalFields` to $\mathbb{U}$ using the field map m_i.

Data: $\mathbb{U}$ an array of floating point values with dimensions $(N_\phi, \Sigma\mathbb{B}_i)$ holding the request cell values; O an integer array of length $\mathbb{S}_t N_T$ initialized to zero used to track the offsets of the local task's grid data in $\mathbb{U}$.

Function ReadInGrids()

- $n = k = 0$
- **for** $i \in (0, 1, \ldots, N_T - 1)$ **do**
  - **for** $j \in (0, 1, \ldots, \mathbb{S}_T - 1)$ **do**
    - $O_k = n$
    - $n+ = \mathbb{B}_k$
    - $k++$
  - **end**
- **end**
- **for** $i \in (0, 1, \ldots, \mathbb{S}_T - 1)$ **do**
  - $G = \mathbb{M}_i$
  - **for** $j \in (0, 1, \ldots, N_T - 1)$ **do**
    - $g = j\mathbb{S}_T + i$
    - G.ReadCells$(\mathbb{U}, \mathbb{B}_g, \mathbb{K}, O_g)$
  - **end**
- **end**

return $\mathbb{U}$

Function 3.15: ReadCells populates the array of requested cells values, $\mathbb{U}$, from the physical fields of a grid.

Data: The grid's physical fields, G.`BaryonField`; n, the number of cells requested from this grid; o, the offset of this grid's data in the request array.

Function ReadCells($\mathbb{U}$: array of floating point numbers, n : integer, $\mathbb{K}$: array of integers, o : integer,)

- **for** $i \in (0, 1, \ldots, N_\phi - 1)$ **do**
  - $\phi = m_i$
  - **for** $j \in (0, 1 \ldots, n - 1)$ **do**
    - $k = \mathbb{K}_{o+j}$
    - $\mathbb{U}_{i,o+j} = G$.`BaryonField`$[\phi][k]$
  - **end**
- **end**

Function 3.16: CommunicateCellValues distributes the cell values from the owning tasks to the requesting tasks.

Data: $\mathbb{K}$ an integer array of length $\Sigma\mathbb{B}_i$ with the indices of cells in grids owned by the current task containing points owned by other tasks; s, t, u, v integer arrays of length N_T initialized to zero used to track the send counts, send offsets, receive counts, and receive offsets, respectively.

Function CommunicateCellValues()

```
m = n = 0
for i ∈ (0, 1, ..., N_T − 1) do
    s_i = 0
    u_i = 0
    for j ∈ (0, 1, ..., 𝕊_i − 1) do
        s_i += ℂ_m
        m++
    end
    for j ∈ (0, 1, ..., 𝕊_T − 1) do
        u_i += 𝔹_n
        n++
    end
end
for i ∈ (0, 1, ..., N_T − 1) do
    t_i = t_{i−1} + s_{i−1}
    v_i = u_{i−1} + v_{i−1}
end
for i ∈ (0, 1, ..., N_ϕ − 1) do
    MPI_Alltoallv(𝕌_i, s, t, ..., 𝕍_i, u, v, ..., ...)
end
```

Function 3.17: ReadCellValues, $\mathbf{\Phi}$.

Data: $\mathbf{\Phi}$: floating point array with dimensions $(N_L, N_T, 2, N_\phi)$ holding the physical field values at each point. O, an integer array containing the offsets of each grid's data in $\mathbb{V}$.

Function ReadCellValues()

- **for** $i \in (1, 2, \ldots, N_G - 1)$ **do**
  - $O_i = O_{i-1} + \mathbb{C}_{i-1}$
- **end**
- **for** $r \in \mathbb{L}$ **do**
  - **for** $p \in (0, 1, \ldots, N_P - 1)$ **do**
    - **for** $i \in (1, 2)$ **do**
      - $G = \mathbb{G}_{r,p,i}$
      - $g = G.\texttt{Index}$
      - $j = O_g$
      - O_g++
      - **for** $\phi \in (0, 1, \ldots, N_\phi - 1)$ **do**
        - $\mathbf{\Phi}_{r,p,i,\phi} = \mathbb{V}_{\phi,j}$
      - **end**
    - **end**
  - **end**
- **end**

return $\mathbf{\Phi}$

3.7 CalculateValues & CalculatePDF

First, the name of the function CalculatePDF is unintentionally misleading, along with the description of $\mathbb{P}$ as probability density functions. At best, these are simple binned counts until they are normalized by the total counts and bin widths as described in Section 3.8 and Function 3.19.

During initialization, the arrays $\mathbb{P}$ are created and the edges of the PDF bins, $\mathbb{Q}$, are determined by both N_B and a pre-defined upper and lower bound for each two-point function, $F_{f,\min}$ and $F_{f,\max}$. These bounds are difficult to determine a priori and were set using values from measures at $r = 0$ (see Table 3.5 in Section 3.9.1). To track the number of point pairs falling out of the range $(F_{f,\min}, F_{f,\max})$, integer counts of pairs greater or less than the range of a given function are kept in $\mathbb{O}_{o,f,r}$ and $\mathbb{O}_{u,f,r}$, respectively. Similarly, running sums of the values and values squared are kept in the arrays $\mathbf{\Sigma}_{\mu,f,r}$ and $\mathbf{\Sigma}_{\sigma,f,r}$, respectively, for each function and separation to be used to calculate the sample mean, $\hat{\mu}_{f,r}$, and variance, $\hat{\sigma}^2_{f,r}$, as part of the reduce step (see Section 3.8). By binning the results as a PDFs, and not just measuring the expected value and variance, we can later estimate higher-order moments of each function.

The functions CalculateValues and CalculatePDF are called at the end of MakePass (Section 3.3.5). CalculateValues builds an array of "diffs", $\boldsymbol{\delta}_{f,r,p}$ for each function, f, separation, r, and point pair, p, i.e.,

$$\boldsymbol{\delta}_{f,r,p} = F_f\left(\mathbf{\Phi}_{r,p,1}, \mathbf{\Phi}_{r,p,2}\right) \tag{3.15}$$

CalculatePDF then iterates over $\boldsymbol{\delta}$, symmetrically binning the values around 0 into $\mathbb{P}$, incrementing $\mathbb{O}$ to count any pairs outside the expected range of F_f, and updating $\mathbf{\Sigma}_{\mu,f,r}$ and $\mathbf{\Sigma}_{\sigma,f,r}$. The process is shown in Function 3.18 for functions binned in $\log_{10}$ space

around 0 using the values Δ_f and ϵ_f, where by default $\epsilon_f = 10^{-5}$ and

$$\Delta_f = \frac{1}{2}\frac{\log_{10}(F_{f,\max}/\epsilon_f)}{N_B}. \tag{3.16}$$

3.7.1 Pseudocode

Function 3.18: CalculatePDF, $\mathbb{P}$.

Function CalculatePDF()
- **for** $f \in F$ **do**
  - **for** $r \in \mathbb{L}$ **do**
    - **for** $p \in (0, 1, \ldots, N_P - 1)$ **do**
      - **if** $F_{f,\min} \leq \boldsymbol{\delta}_{f,r,p} \leq F_{f,\max}$ **then**
        - $\boldsymbol{\Sigma}_{\mu,f,r} \mathrel{+}= \boldsymbol{\delta}_{f,r,p}$
        - $\boldsymbol{\Sigma}_{\sigma,f,r} \mathrel{+}= (\boldsymbol{\delta}_{f,r,p})^2$
        - $b = \lfloor \log_{10}(|\boldsymbol{\delta}_{f,r,p}|/\epsilon_f)/\Delta_f \rfloor$
        - **if** $\boldsymbol{\delta}_{f,r,p} < 0$ **then**
          - $b = N_B/2 - b - 1$
        - **else**
          - $b \mathrel{+}= N_B/2$
        - **end**
        - $\mathbb{P}_{f,r,b} \mathrel{+}= 1$
      - **else**
        - **if** $\boldsymbol{\delta}_{f,r,p} < F_{f,\min}$ **then**
          - $\mathbb{O}_{u,f,r}{+}{+}$
        - **else**
          - $\mathbb{O}_{o,f,r}{+}{+}$
        - **end**
      - **end**
    - **end**
  - **end**
- **end**

return $\mathbb{P}$

3.8 Reduce & Write

Data reduction uses the MPI function MPI_Allreduce, which can perform common operations, such as summation, using input from all tasks. This is shown in Function 3.19, where the over-under, running sum, and PDF arrays are summed across all tasks and normalized by the total effective point count for each function and separation, $N^*_{f_r}$, where

$$N^*_{f,r} = N_\Sigma - (\mathbb{O}_{u,f,r} + \mathbb{O}_{o,f,r}). \tag{3.17}$$

The values in $\mathbb{P}$ are then normalized by the bin widths to convert them from probability mass functions to PDFs.

$$\frac{\mathbb{P}_{f,r,b}}{(\mathbb{Q}_{f,r,b+1} - \mathbb{Q}_{f,r,b})} \rightarrow \mathbb{P}_{f,r,b} \tag{3.18}$$

On task T_0, the PDFs, over and under counts, etc., for each function are written out to two files: a short text file with the parameters used, the over-under counts, and the sample means; and an HDF5 file with all of the data collected, including the full $\mathbb{P}$ arrays.

3.8.1 Pseudocode

Function 3.19: Reduce

Data: R: one-dimensional array of length N_L used to receive data; Q: one-dimensional array of length N_B used to receive data

Function Reduce()

$N = N_T \times N_P \times N_M$

for $f \in F$ **do**

MPI_Allreduce($\mathbb{O}_{u,f}, R, N_L, \ldots,$ MPI_SUM, $\ldots$)

$R \rightarrow \mathbb{O}_{u,f}$

MPI_Allreduce($\mathbb{O}_{o,f}, R, N_L, \ldots,$ MPI_SUM, $\ldots$)

$R \rightarrow \mathbb{O}_{o,f}$

MPI_Allreduce($\boldsymbol{Sigma}_{\mu,f}, R, N_L, \ldots,$ MPI_SUM, $\ldots$)

$R \rightarrow \mathbf{\Sigma}_{\mu,f}$

MPI_Allreduce($\mathbf{\Sigma}_{\sigma,f}, R, N_L, \ldots,$ MPI_SUM, $\ldots$)

$R \rightarrow \mathbf{\Sigma}_{\sigma,f}$

for $r \in \mathbb{L}$ **do**

MPI_Allreduce($\mathbb{P}_{f,r}, Q, N_B, \ldots$ MPI_SUM, $\ldots$)

$Q \rightarrow \mathbb{P}_{f,r}$

$N^* = N - (\mathbb{O}_{u,f,r} + \mathbb{O}_{o,f,r})$

$\mathbb{P}_{f,r} \mathrel{/}= N^*$

$\mathbf{\Sigma}_{\mu,f,r} \mathrel{/}= N^*$

$\mathbf{\Sigma}_{\sigma,f,r} \mathrel{/}= N^*$

for $b \in (0, 1, \ldots, N_B - 1)$ **do**

$\mathbb{P}_{f,r,b} \mathrel{/}= (\mathbb{Q}_{f,r,b+1} - \mathbb{Q}_{f,r,b})$

end

end

end

return

3.9 Validity

> A calculation that does not include a calculation of its predictive skill is not a legitimate scientific product
>
> H. Tennekes

There are three important points to make when discussing the validity of the method:

- Validity in this context is whether or not SFgen measures what is intended. It is not referring to any type of hypothesis testing, such as whether or not a scaling relation is supported by the measurements.
- The measurements are only intended to generalize to a particular dataset. More specifically, it does not address whether or not a particular measurement is stable over time (Galanti and Tsinober 2004). The measurements could be used to evaluate the stability or ergodicity of particular relations, but SFgen itself specifically works at a point in time.
- SFgen also cannot infer whether or not the results generalize beyond the simulation. That likelihood depends on how well the simulation models the desired physical system.

Within a single dataset, the physical fields and two-point functions have extreme variances and, except for density, do not follow well-defined distributions. However, while there is an infinite number of points within a simulation volume, the samples are selected from a finite population of grid cells and their pairs at a given r (see the estimates in Section 3.2). The samples are further constrained by the representation of the physical fields as floating-point numbers, bounded by the number of bits used to represent the exponent ("IEEE Standard for Floating-Point Arithmetic" 2019). Given this, the expected value,

variance, and distribution exist for the two-point functions. The central limit theorem ensures that any measurements through simple random sampling will approach the true values of each.

This method relies on ensuring that each point pair is an independent and identically distributed random variable so that an arbitrary level of precision can be achieved by increasing the number of pairs sampled. There are four key elements to ensuring sample independence in the method:

- the reliable determination of the most refined grid cell at a point (see Section 3.4);
- random selection of $\boldsymbol{x}_1$ (see Section 3.5.2);
- and the spherically uniform random selection of the angles defining $\hat{\boldsymbol{r}}$ (also in Section 3.5.2).

The first element ensures that grid cells at every level of refinement are represented proportionally to their volume minus the volume of their subgrids. The next two elements use the parallel random number application, SPRNG (Mascagni and Srinivasan 2000). The software uses SPRNG's 64-bit linear congruential generator with prime addend (LCG64) generator. This generator has a period of 2^{64} and supports up to 10^8 distinct streams, where a single stream is used on each task. This is more than sufficient for the number of point pairs required.

Outside of this, the remaining factors impacting the method's validity are the reliable retrieval of the physical field values from the point pairs as input to the two-point functions (see Section 3.6) and determining N_Σ to achieve a desirable range for the standard error of the mean, which is covered in the next section.

3.9.1 Determing the Number of Samples from Variances

There are qualitative indicators of whether or not the number of samples for a given two-point function was sufficient, the main one being that the measured relation should be smooth and continuous along r. Even though there are local extreme values in some fields, in particular, $\boldsymbol{\nabla} \cdot \boldsymbol{u}$, the average along the separation eliminates those jumps. If not enough points are sampled, however, the measured scaling may show sudden changes. Likewise, while there is no guarantee a particular relation is monotonic with r, their forms tend to are not suggestive of multiple extrema. In other words, the plots tend to go up or go down, but not up and down repeatedly.

As an example, Figure 3.14 shows a plot of the flux correlation from Equation 5.10 (see also Chapter 5)

$$\Phi(r) = \left\langle (\rho \boldsymbol{u} \cdot \boldsymbol{u}' \rho')\, u'_{\parallel} \right\rangle + \left\langle \rho u_{\parallel} \rho' \right\rangle \tag{3.19}$$

where is prime $'$ is used to denote values from $\boldsymbol{x}_2$. The plots of $\Phi(r)$ in Figure 3.14 were measured using different numbers of passes, $N_M = \mathrm{P} \in (4, 16, 64, 256, 1024)$, while the number of tasks ($N_T = 128$) and point pairs ($N_P = 2^{16}$) were held constant. The total number of samples per pass were $2^{23} \approx 8.4 \times 10^6$, for a range of $N_\Sigma \in (3.4 \times 10^7, 8.9 \times 10^{10})$. The data came from a single dataset of the same $1,024^3$ uniform grid simulation of supersonic turbulence used in Chapters 4, 5, and 6, and was part of the validation process for the publication of Chapter 5. The results from all $1,364$ passes were averaged to create a somewhat "true" value, noted as $\langle \Phi_t(r) \rangle$. The plots support a qualitative assessment that $\mathrm{P} = 4$ is insufficient, while $\mathrm{P} \geq 256$ may be adequate.

However, when considering a new scaling relation with an unknown distribution from its combination of fields, it would be better to start with an initial estimate of N_Σ and adjust the number of samples based on actual data. To begin, we can define a target level of precision, or confidence interval, based on the true (or population) mean for the

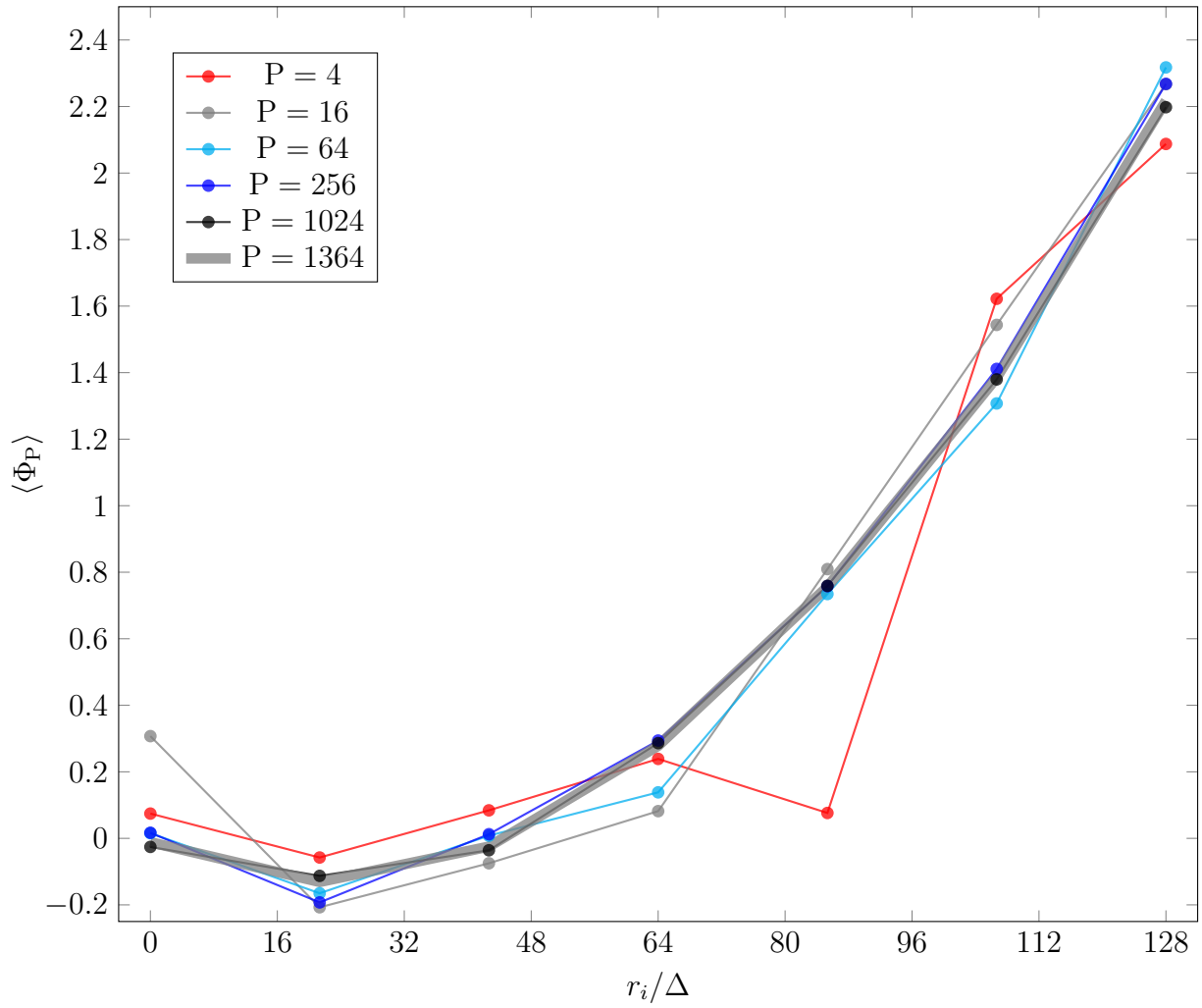

Figure 3.14: $\langle \Phi_{\mathrm{P}} \rangle$, for the different values of P, and the average over all 1364 passes. Δ is the width of a grid cell where $r = 1 = 1024\Delta$.

two-point function, $\mu_{f,r}$, and the true (or population) variation $\sigma^2_{f,r}$

$$\mu_{f,r} = \hat{\mu}_{f,r} \pm r_e \ \mu_{f,r} \tag{3.20}$$

where r_e is an arbitrary "radius of error" to scale the precision. This is certainly ad hoc, but when combined with the standard error of the mean and a standard score (z-score), it provides a mechanism to guide the number of samples, i.e.,

$$\mu_{f,r} = \hat{\mu}_{f,r} \pm z_p \ \frac{\hat{\sigma}_{f,r}}{\sqrt{N_\Sigma}}, \tag{3.21}$$

where

$$z_p = \sqrt{2} \ \mathrm{erf}^{-1}(2p - 1). \tag{3.22}$$

is the value from a normal distribution for a selected confidence level, p. The well-known example is for a 95% confidence level, $z_{0.95} = 1.96$. Using Equations 3.20 and 3.21 we now have an initial way to calculate a minimum for N_Σ

$$N_\Sigma \geq \left(\frac{z_p}{r_e}\right)^2 \left(\frac{\hat{\sigma}_{f,r}}{\hat{\mu}_{f,r}}\right)^2 \tag{3.23}$$

This is not ideal since it is not valid as $\mu_{f,r} \to 0$, but it provides a starting point, which can be used iteratively as the measurements of $\hat{\mu}_{f,r}$ and $\hat{\sigma}^2_{f,r}$ become more precise. It also shows that the number of samples depends on the ratio between the variance and the mean squared.

This is practically useful for estimating the number of samples needed by substituting the known means and variances for dimensionally similar single-point statistics. For many scaling relations, this provides an upper limit on the number of samples needed for a given level of precision since as the correlation between the inputs generally decrease with scale, the magnitude of the variance from their products also decrease. Single-point statistics for

Table 3.5: Single-point statistics for different fields and functions. Blanks are where the expected value of the mean is 0.

Variable	μ	σ^2	$\frac{\sigma^2}{\mu^2}$	min	max	median	κ
ρ	1	3	7	0.006	4×10^2	0.5	5×10^1
ρ^2	4	2×10^3	4×10^5	4×10^{-5}	2×10^5	0.3	6×10^7
$\log(\rho)$	-0.6	1	4	-5	6	-0.6	-0.08
$\log(\rho^2)$	-1	5	2×10^1	-1×10^1	1×10^1	-1	-9
u_x	0	2×10^1		-1×10^1	1×10^1	-0.6	4×10^1
u_y	0	6		-9	8	-0.04	-0.3
u_z	0	2×10^1		-1×10^1	1×10^1	-0.8	4×10^1
$u = \lvert\boldsymbol{u}\rvert$	6	1×10^1	3	0.004	2×10^1	5	2×10^2
u^2	4×10^1	2×10^3	2×10^3	1×10^{-5}	4×10^2	3×10^1	5×10^5
u^3	4×10^2	4×10^5	1×10^6	5×10^{-8}	7×10^3	1×10^2	1×10^9
$\rho\ u$	6	1×10^2	3×10^2	0.0006	2×10^3	3	1×10^4
$\rho\ u^2$	4×10^1	9×10^3	5×10^4	6×10^{-6}	2×10^4	1×10^1	1×10^7
$\rho\ u^3$	4×10^2	1×10^6	9×10^6	3×10^{-8}	3×10^5	7×10^1	2×10^{10}
$\rho^2\ u^3$	1×10^3	3×10^8	8×10^{10}	6×10^{-9}	3×10^7	3×10^1	1×10^{15}
$\boldsymbol{\nabla} \cdot \boldsymbol{u}$	0	1×10^5		-2×10^4	4×10^3	6×10^1	-1×10^8
$\rho \boldsymbol{\nabla} \cdot \boldsymbol{u}$	-3×10^1	2×10^6	4×10^9	-2×10^6	2×10^5	2×10^1	-4×10^{11}
$\rho u^2 \boldsymbol{\nabla} \cdot \boldsymbol{u}$	-1×10^3	5×10^9	1×10^{13}	-8×10^7	1×10^7	4×10^2	-3×10^{16}

several variables are shown in Table 3.5, including the ratio σ^2/μ^2. Table 3.6 shows the order of magnitude for the same terms. When combined with the magnitude of confidence and precision term using a 95% confidence level and $r_e = 0.1$, $\mathcal{O}(z_p^2/r_e^2) \approx 10^2$, column $\mathcal{O}(\sigma^2/\mu^2)$ in Table 3.6 provide a reasonable scale of N_Σ.

As an example, the two-point function for the third order density-weighted structure function $\mathcal{S}_3(r)$ shown in Equation 3.1 has the same dimensions as ρu^3. Referring to Table 3.6 the quick estimate for the number of samples would be $N_\Sigma \geq 10^8$. For completeness, the actual parameters for ρu^3 are

$$\mu_{\rho u^3} = \langle \rho u^3 \rangle = 365 \tag{3.24}$$

$$\sigma^2_{\rho u^3} = \left\langle \left(\rho u^3\right)^2 \right\rangle - \mu^2_{\rho u^3} = 1.11 \times 10^6 \tag{3.25}$$

Entering these statistics into Equation 3.23, keeping the 95% confidence level and $r_e = 0.1$,

Table 3.6: The same as Table 3.5, as order-of-magnitude estimates.

Variable	$\mathcal{O}(\lvert\mu\rvert)$	$\mathcal{O}(\sigma^2)$	$\mathcal{O}(\frac{\sigma^2}{\mu^2})$	$\mathcal{O}(\lvert\text{min}\rvert)$	$\mathcal{O}(\lvert\text{max}\rvert)$	$\mathcal{O}(\lvert\text{median}\rvert)$	$\mathcal{O}(\lvert\kappa\rvert)$
ρ	10^0	10^0	10^0	10^{-3}	10^2	10^{-1}	10^1
ρ^2	10^0	10^3	10^5	10^{-5}	10^5	10^{-1}	10^7
$\log(\rho)$	10^{-1}	10^0	10^0	10^0	10^0	10^{-1}	10^{-2}
$\log(\rho^2)$	10^0	10^0	10^1	10^1	10^1	10^0	10^0
u_x	0	10^1		10^1	10^1	10^{-1}	10^1
u_y	0	10^0		10^0	10^0	10^{-2}	10^{-1}
u_z	0	10^1		10^1	10^1	10^{-1}	10^1
$u = \lvert\boldsymbol{u}\rvert$	10^0	10^0	10^0	10^{-3}	10^1	10^0	10^2
u^2	10^1	10^3	10^3	10^{-5}	10^2	10^1	10^5
u^3	10^2	10^5	10^6	10^{-8}	10^3	10^2	10^9
$\rho\ u$	10^0	10^2	10^2	10^{-4}	10^3	10^0	10^4
$\rho\ u^2$	10^1	10^3	10^4	10^{-6}	10^4	10^1	10^7
$\rho\ u^3$	10^2	10^6	10^6	10^{-8}	10^5	10^1	10^{10}
$\rho^2\ u^3$	10^3	10^8	10^{10}	10^{-9}	10^7	10^1	10^{15}
$\boldsymbol{\nabla}\cdot\boldsymbol{u}$	0	10^4		10^4	10^3	10^1	10^8
$\rho\boldsymbol{\nabla}\cdot\boldsymbol{u}$	10^1	10^6	10^9	10^6	10^5	10^1	10^{11}
$\rho u^2\boldsymbol{\nabla}\cdot\boldsymbol{u}$	10^3	10^9	10^{13}	10^7	10^6	10^2	10^{16}

gives

$$N_\Sigma \geq \left(\frac{z_p}{r_e}\right)^2 \left(\frac{\sigma_{\rho u^3}}{\mu_{\rho u^3}}\right)^2 \tag{3.26}$$

$$N_\Sigma \geq \left(\frac{1.96}{0.1}\right)^2 \frac{1.11 \times 10^6}{365^2} \tag{3.27}$$

$$N_\Sigma \geq 384 \times 9.23 \times 10^6 \tag{3.28}$$

$$N_\Sigma \geq 3.55 \times 10^9 \tag{3.29}$$

where an order of magnitude was gained from the product of the coefficients. Hopefully this exercise and the tables provide more justification for the number of samples described in Section 3.2 and the results chapters.

I'll note two other aspects of the number of samples. First, r_e is the somewhat free parameter to adjust the number of samples. However, the measurements of the scaling relations are inputs to other analysis steps, like fitting to the predicted relations

and calculating higher-order moments. Minimizing (within practical limits) the standard error from the sampling process reduces the propagated error. Second, the method is constrained by the total number of point pairs, not the number of scaling relations or two-point functions. So N_Σ is set to handle the worst-case scenario, that is, the products with large variances compared to their mean, such as $\boldsymbol{\nabla} \cdot \boldsymbol{u}$. The benefit of this is that inputs with less extreme ranges have even greater precision.

3.9.2 Convergence Study

To finish the discussion of the method's validity, let's return to the qualitative assessment of Figure 3.14. Dimensionally, $\Phi(r)$ is $\rho^2 u^3$, and from Table 3.6 an initial guess for N_Σ is 10^{12} at $z_{0.95}$ and $r_e = 0.1$, while Figure 3.14 show that the sampling may have been sufficient at $N_\Sigma \approx 10^{10}$. As confirmation that we are within the intended precision, we can define a measurement of the error at each separation and sample count, $E_\mathrm{P}(r_i)$, using the absolute deviation, as shown in Equation 3.30.

$$E_\mathrm{P}(r_i) = |\langle \Phi_\mathrm{P}(r_i) \rangle - \langle \Phi_t(r_i) \rangle| \tag{3.30}$$

This can be extended to an average over all of the separations to give a rough guide to how well the number of point pairs is capturing the true value of Φ (see Equation 3.31).

$$\epsilon_\mathrm{P} = \frac{1}{7} \sum_{i=1}^{7} E_\mathrm{P}(r_i) \tag{3.31}$$

Finally, Equation 3.32 is the average error radius for Φ across the separations and the target level of precision.

$$\epsilon_t = \frac{1}{7} \sum_{i=1}^{7} |r_e \langle \Phi_t(r_i) \rangle| = 0.069 \tag{3.32}$$

Depending on the accuracy and precision of $\langle \Phi_P \rangle$, ϵ_P should diminish with $\hat{\sigma}^2_{f,r}$ like

Table 3.7: Sampling convergence. Horizontal line indicates the radius of error goal of $\epsilon_{\mathrm{P}} \leq \epsilon_t = 0.069$

N_M	ϵ_{P}	$\epsilon_{4\mathrm{P}}/\epsilon_{\mathrm{P}}$
4	0.479	
16	0.297	0.62
64	0.133	0.45
256	0.055	0.42
1024	0.030	0.55

$1/\sqrt{N_\Sigma}$, or $1/2$ for each factor of four increase of N_Σ used in this example. Indeed, this is shown in Table 3.7, using the ratio of $\epsilon_{4\mathrm{P}}$ and ϵ_{P}. As noted above, the target precision threshold, ϵ_t was reached at $\mathrm{P} = 256$. To complete the convergence study, Figure 3.15 shows both $E_{\mathrm{P}}(r_i)$ and ϵ_{P}.

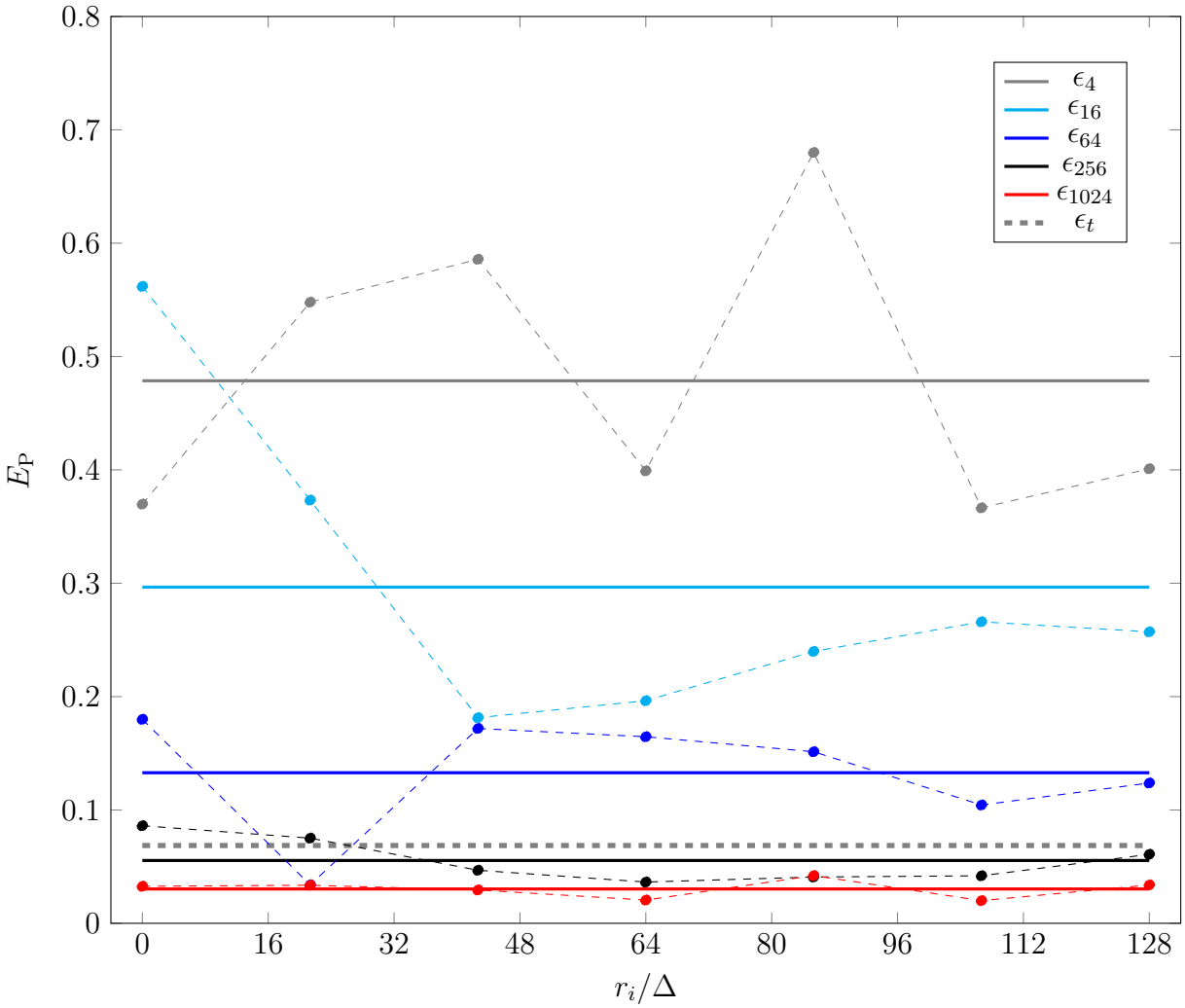

Figure 3.15: Average absolute deviation from $\langle \Phi_t \rangle$. The solid lines show the average of the error over r_i.

3.9.3 Summary

In summary, the sampling method in SFgen follows the central limit theorem for the finite population of values drawn from a simulation dataset. Upper bounds on the number of samples required to achieve an arbitrary level of precision can be estimated using the variances from single-point statistics. Evaluations of the standard errors from sampling show the expected $1/\sqrt{n}$ behavior, where n is the total number of samples.

3.10 Performance & Scaling

I'll conclude this chapter with a summary of some performance data. Overall, the method spends $\gtrsim 65\%$ of its time generating points, calculating values for each two-point function, and binning values in $\mathbb{P}$, while $\approx 10\%$ of the time is used for communications. The remainder of the time is used to handle data, like building the ordered arrays of cell indices for the requests, and $\approx 15\%$ going to data wrangling in Functions ReadCellValues (3.17) and ReadInGrids (3.14). Weak scaling was assessed by comparing the performance for 64 and 512 tasks, which showed an $\approx 4\%$ increase in the average time per pass. This shows that the method is well suited for generating large numbers of samples through increased parallelism and for inclusion in large-scale parallel applications.

Chapter 4

Scaling Laws and Intermittency in Highly Compressible Turbulence

Abstract

We use large-scale three-dimensional simulations of supersonic Euler turbulence to study the physics of a highly compressible cascade. Our numerical experiments describe non-magnetized driven turbulent flows with an isothermal equation of state and an rms Mach number of 6. We find that the inertial range velocity scaling deviates strongly from the incompressible Kolmogorov laws. We propose an extension of Kolmogorov's K41 phenomenology that takes into account compressibility by mixing the velocity and density statistics and preserves the K41 scaling of the density-weighted velocity $v \equiv \rho^{1/3}u$. We show that low-order statistics of v are invariant with respect to changes in the Mach number. For instance, at Mach 6 the slope of the power spectrum of v is -1.69 and the third-order structure function of v scales linearly with separation. We directly measure the mass dimension of the "fractal" density distribution in the inertial subrange, $D_m \approx 2.4$, which is similar to the observed fractal dimension of molecular clouds and agrees well with

the cascade phenomenology.

4.1 Introduction

In the late 1930's, Kolmogorov clearly realized that chances to develop a closed purely mathematical theory of turbulence are extremely low (Kolmogorov 1985).[1] Therefore, the basic approach in Kolmogorov (1941a,c) (usually referred to as the K41 theory) was to rely on physical intuition and formulate two general statistical hypotheses which describe the universal equilibrium regime of small-scale fluctuations in arbitrary turbulent flow at high Reynolds number. Following the Landau (1944) remark on the lack of universality in turbulent flows (Landau and Lifshitz 1987), and with information extracted from new experimental data, the original similarity hypotheses of K41 were then revisited and refined to account for intermittency effects (Dubrulle 1994; Kolmogorov 1962; She and Leveque 1994). While the K41 phenomenology became the cornerstone for all subsequent developments in incompressible turbulence research (e.g., Frisch 1995), there was no similar result established for compressible flows yet (Friedrich 2007; Lele 1994). Historically, compressible turbulence research, preoccupied with a variety of specific engineering applications, was generally lagging behind the incompressible developments.[2] The two major reasons for this time lag were an additional complexity of analytical treatment of compressible flows and a shortage in experimental data for super- and hypersonic turbulence. In this respect, although limited to relatively low Reynolds numbers, direct numerical simulations (DNS) of turbulence (pioneered by Orszag and Patterson 1972) have occupied the niche of

[1] "An understanding of solutions to the [incompressible] Navier-Stokes equations" yet remains one of the six unsolved grand challenge problems nominated by the Clay Mathematics Institute in 2000 for a $1M *Millennium Prize* [http://www.claymath.org/millennium/].

[2] A reasonable measure of the delay is 60+ years passed between the appearance of incompressible Reynolds averaging (Reynolds 1895) and mass-weighted Favre averaging for fluid flows with variable density (Favre 1958), although see Lumley and Yaglom (2001) for references to a few earlier papers that dealt with density-weighted averaging.

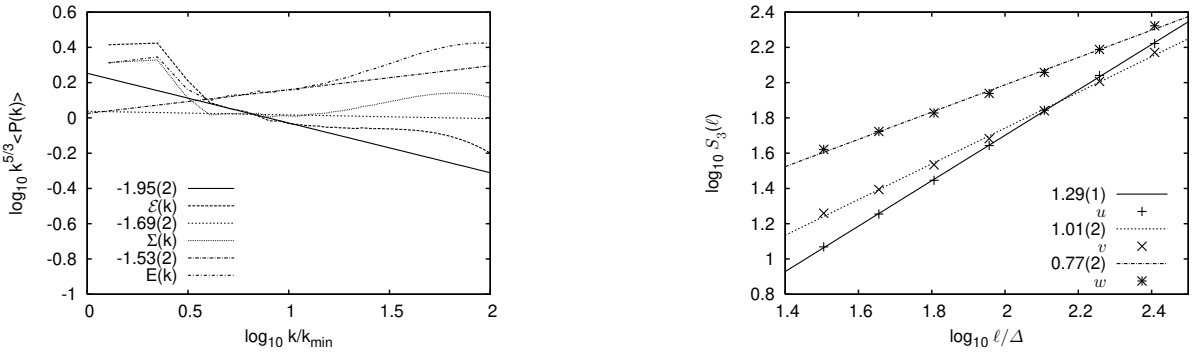

Figure 4.1: Time average compensated power spectra (*left*) and third-order transverse structure functions (*right*) for velocity u and mass-weighted velocities $v \equiv \rho^{1/3}u$ and $w \equiv \rho^{1/2}u$. The statistics of v clearly demonstrate a K41-like scaling. Notice strong bottleneck contamination in the spectra at high wavenumbers.

experiments at least for the most simple flows. One particularly important advantage of DNS is an easy access to variables that are otherwise difficult to measure in the laboratory or treat analytically.

A traditionally straightforward approach to data analysis from DNS of compressible turbulence includes computation of the "standard" statistics of velocity fluctuations. In addition, the diagnostics for density fluctuations are also computed and discussed as the direct measures of compressibility. Quite naturally, both density and velocity statistics demonstrate strong dependence on the Mach number $\mathcal{M}$ in supersonic ($\mathcal{M} \in [1,3]$) and hypersonic ($\mathcal{M} > 3$) regimes, while the variations in turbulent diagnostics at sub- or transonic Mach numbers are rather small. For instance, at $\mathcal{M} \approx 1$ the velocity power spectrum closely follows the K41 scaling and the third order velocity structure functions scale roughly linearly with separation (Porter, Pouquet, and Woodward 2002). The density power spectrum in weakly compressible isothermal flows scales as $\sim k^{-7/3}$ (Bayly, Levermore, and Passot 1992), at $\mathcal{M} \approx 1$ it scales as $\sim k^{-1.7}$ (Kritsuk et al. 2007b), and at $\mathcal{M} \approx 6$ the slope is -1.07 (Kritsuk et al. 2007b).

Based on the data from numerical experiments, it is well established that: (i) the velocity power spectra tend to get steeper as the Mach number increases, reaching the

Burgers slope of -2 asymptotically Biskamp 2003, and references therein; (ii) the density power spectra instead get shallower at high Mach numbers, approaching a slope of -1 or even shallower (Kritsuk, Norman, and Padoan 2006); (iii) the density PDF in isothermal turbulent flows is well represented by a lognormal distribution Biskamp 2003, and references therein; (iv) the dimensionality of the most singular velocity structures increases from $D_{s,u} \sim 1$ in a subsonic regime to $D_{s,u} \sim 2$ in highly supersonic (Padoan et al. 2004); (v) the mass dimension of the turbulent structures decreases from $D_m = 3$ in weakly compressible flows to $D_m \sim 2.5$ in highly compressible (Kritsuk et al. 2007b).

How can we combine these seemingly disconnected pieces of information into a coherent physical picture to improve our understanding of compressible turbulence? One way to do this is to consider a phenomenological concept of a *lossy* compressible turbulent cascade that would asymptotically match the incompressible Kolmogorov-Richardson energy cascade (Kolmogorov 1941c; Richardson 1922) in the limit of very low Mach numbers. Since incompressible turbulence represents a degenerate case where the density is uncorrelated with the velocity, the phenomenology of the compressible cascade must include this correlation. This essentially means that instead of velocity u, which is a single key ingredient of the K41 laws, one needs to consider a set of mixed variables, $\rho^{1/\eta}u$, where ρ is the density and η can take values 1, 2, or 3, depending on the statistical measure of interest (Kritsuk et al. 2007b). For instance, if one is studying the scale-by-scale kinetic energy budget in a compressible turbulent flow, a mixed variable power spectrum with $\eta = 2$ would be an appropriate choice. To deal with the kinetic energy flux through the hierarchy of scales within the inertial range, the key mixed variable would be the one with $\eta = 3$.

How will these mixed statistics scale in the inertial range of highly compressible turbulent flows? Will their scaling depend on the Mach number? Can the K41 phenomenology be extended to cover hypersonic turbulent flows? These and other related questions are in

detail discussed in Kritsuk et al. (2007b) based on Euler simulations of driven isotropic supersonic turbulence with the Piecewise Parabolic Method Colella and Woodward (1984) and with resolution up to 2048^3 grid points. In this paper we present the highlights of the compressible cascade phenomenology verified in Kritsuk et al. (2007b).

4.2 Scaling, Structures, and Intermittency

Nonlinear interactions transfer kinetic energy supplied to the system at large scales through the inertial range with little dissipation. Let us assume that the mean *volume* energy transfer rate in a compressible fluid, $\rho u^2 u/\ell$, is constant in a statistical steady state e.g., Lighthill 1955. If this is true, then

$$v^p \equiv (\rho^{1/3} u)^p \sim \ell^{\,p/3} \tag{4.1}$$

for an arbitrary power p and, with the standard assumption of self-similarity of the cascade, the structure functions (SFs) of mixed variable v for compressible flows should scale in the inertial range as

$$\mathcal{S}_p(\ell) \equiv \langle |v(r+\ell) - v(r)|^p \rangle \sim \ell^{\,p/3}. \tag{4.2}$$

In the limit of weak compressibility, the scaling laws (4.2) will reduce to the K41 results for the velocity structure functions. The scaling laws $\mathcal{S}_p(\ell) \sim \ell^{\zeta_p}$, where $\zeta_p = p/3$ are not necessarily exact. As the incompressible K41 scaling, they are subject to "intermittency corrections", e.g. $\zeta_p = p/3 + \tau_{p/3}$ (Kolmogorov 1962). The only exception is, perhaps, the third order relation for the longitudinal velocity SFs, which is exact in the incompressible case and is known as the *four-fifth law* (Kolmogorov 1941a). Our focus here is mostly on the low order statistics ($p \leq 3$) for which the corrections are small. Since the power spectrum slope is related to the exponent of the second order structure function, the K41

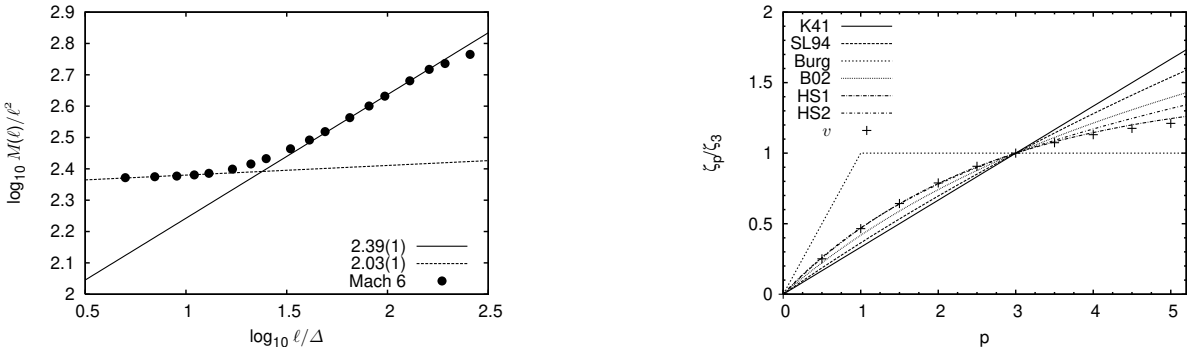

Figure 4.2: Gas mass $M(\ell)$ as a function of the box size ℓ (*left*). The mass dimension D_m is defined as the log-log slope of $M(\ell)$, see eq. (4). Relative exponents for structure functions of the transverse modified velocities v versus order p and two hierarchical structure models with different parameters HS1 & HS2, She and Leveque 1994 that fit the data for $p \in [0, 3]$ (*right*). Also shown are model predictions for the Kolmogorov-Richardson cascade (K41, Kolmogorov 1941a,c), for intermittent incompressible turbulence SL94, She and Leveque 1994, for "burgulence" Burg, Bec and Khanin 2007, and for the velocity fluctuations in supersonic turbulence B02, Boldyrev 2002.

slope of 5/3 is expected to hold for $v \equiv \rho^{1/3}u$ in the compressible case.

Figure 4.1 shows the power spectra of u, v, and $w \equiv \rho^{1/2}u$ and the corresponding third-order transverse structure functions based on the simulations at Mach 6 (Kritsuk et al. 2006; Kritsuk et al. 2007b). The power spectrum $\Sigma(k)$ and the structure function of v clearly follow the K41 scaling: $\Sigma \sim k^{-1.69}$ and $\mathcal{S}_3 \sim \ell^{1.01}$ (Kritsuk et al. 2007b), while the velocity power spectrum $\mathcal{E}(k)$ and structure function have substantially steeper-then-K41 slopes: -1.95 and 1.29 (Kritsuk et al. 2006). At the same time, the kinetic energy spectrum $E \sim k^{-1.53}$ is shallow and both solenoidal and dilatational components of w have the same slope implying a single compressible energy cascade with strong interaction between the two components (Kritsuk et al. 2007b). These results based on the high dynamic range simulations lend strong support to the scaling relations described by eq. (4.2) and to the conjecture from which they were inferred. Previous simulations at lower resolution did not allow to measure the absolute exponents reliably due to insufficient dynamic range and due to the bottleneck contamination (Zakharov, L'Vov, and Falkovich 1992).

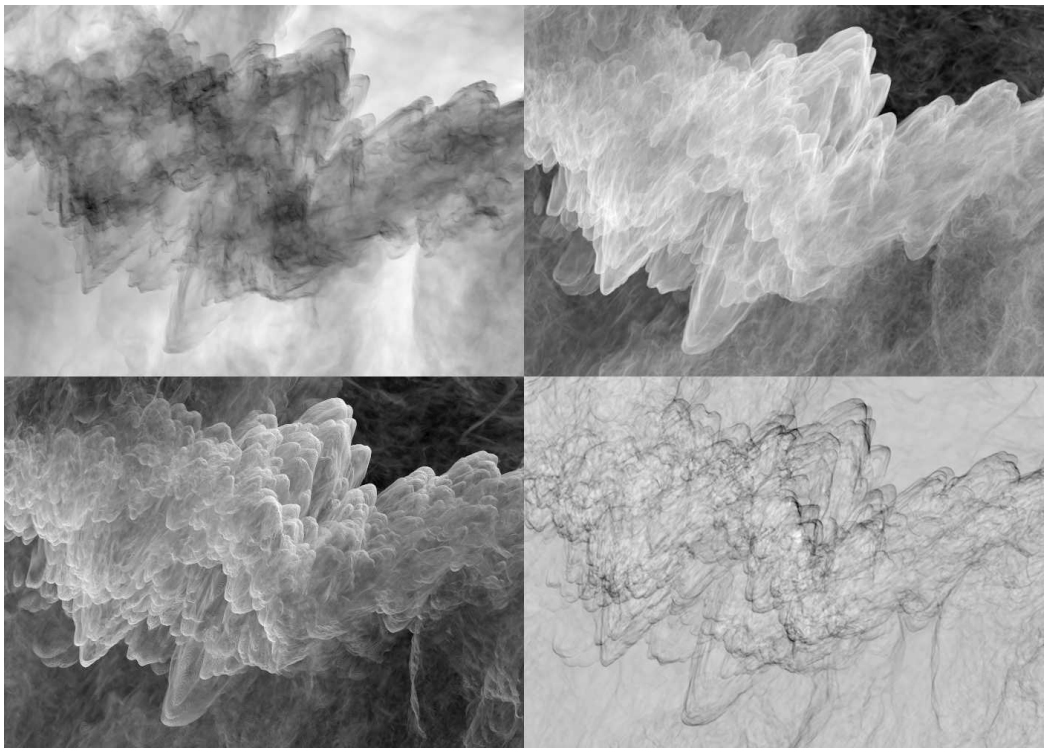

Figure 4.3: Coherent structures in Mach 6 turbulence at resolution of 1024^3. Projections along the minor axis of a subvolume of $700 \times 500 \times 250$ zones for the density (*upper left*), the enstrophy (*upper right*), the dissipation rate (*lower left*), and the dilatation (*lower right*). The logarithmic grey-scale ramp shows the lower values as dark in all cases except for the density. The inertial subrange structures correspond to scales between 40 and 250 zones and represent a fractal with $D_m \approx 2.4$. The dominant structures in the dissipation range ($\ell < 30\Delta$) are shocks with $D_m = 2$. [Reprinted from Kritsuk et al. (2007b).]

In 1951, von Weiszäker (von Weizsäcker 1951) introduced a phenomenological model for scale-invariant hierarchy of density fluctuations in compressible turbulence described by a simple equation that relates the mass density at two successive levels to the corresponding scales through a universal measure of the degree of compression, α,

$$\rho_n/\rho_{n-1} = (\ell_n/\ell_{n-1})^{-3\alpha}\,. \tag{4.3}$$

The geometric factor α takes the value of 1 in a special case of isotropic compression in three dimensions, 1/3 for a perfect one-dimensional compression, and zero in the incompressible limit. From equations (4.1) and (4.3), assuming mass conservation, Fleck (1996) derived a set of scaling relations for the velocity, specific kinetic energy, density, and mass:

$$u \sim \ell^{1/3+\alpha},\;\; \mathcal{E}(k) \sim k^{-5/3-2\alpha},\;\; \rho \sim \ell^{-3\alpha},\;\; M(\ell) \sim \ell^{D_m} \sim \ell^{3-3\alpha}, \tag{4.4}$$

where all the exponents depend on the compression measure α which is in turn a function of the rms Mach number of the turbulent flow. We can now use the data from numerical experiments to verify the scaling relations (4.4). Since the first-order velocity structure function scales as $\ell^{0.54}$ (Kritsuk et al. 2007b), we can estimate α for the Mach 6 flow, $\alpha \approx 0.21$. Using the last relation in (4.4), we can calculate the mass dimension for the density distribution, $D_m \approx 2.38$. It is indeed consistent with our direct measurement of the mass dimension for the same range of scales, $D_m \approx 2.39$, see Fig. 4.2.

In strongly compressible turbulence at Mach 6, the density contrast between supersonically moving blobs and their more diffuse environment can be as high as 10^6. The most common structural elements in such highly fragmented flows are nested bow-shocks (Kritsuk, Norman, and Padoan 2006). Figure 4.3 shows an extreme example of structures formed by a collision of counter-propagating supersonic flows. On small scales within the dissipation range, these structures are characterized by $D_m = 2$, while within the inertial

range $D_m \approx 2.4$ (Fig. 2, *left*). The hierarchical structure (HS) model

$$\zeta_p/\zeta_3 = \gamma p + C(1-\beta^p) \tag{4.5}$$

She and Leveque (1994) provides good fits to the data for the mass-weighted velocity v (see Fig. 2, *right*). Here the codimension of the support of the most singular dissipative structures

$$C \equiv 3 - D_{s,v} = (1-3\gamma)/(1-\beta^3). \tag{4.6}$$

If the fit is limited to $p \in [0, 3]$, two sets of model parameters β and γ are formally acceptable (models HS1 and HS2 in Fig. 2). The best-fit parameters of the HS1 model: $\beta_1^3 = 1/3$ (a measure of intermittency), $\gamma_1 = 0$ (a measure of singularity of structures), and $C_1 = 1.5$ correspond to a hybrid between the B02 model for the velocity fluctuations ($\beta_{B02}^3 = 1/3$, $\gamma_{B02} = 1/9$) (Boldyrev 2002) and the Burgers' model ($\beta_{Burg} = 0$, $\gamma_{Burg} = 0$) (Bec and Khanin 2007). The HS2 model ($\beta_2^3 = 1/6$, $\gamma_2 = 1/9$, and $C_2 = 0.8$) provides a fit of roughly the same quality for $p \in [0, 3]$, but overestimates the scaling exponents ζ_p at $p > 4$. Since the level of uncertainty in the high order statistics remains high even at a resolution of 1024^3 grid points, larger dynamic range simulations are needed to distinguish between the two options.

If the HS1 option is confirmed, then Mach 6 turbulence is more intermittent than incompressible turbulence ($\beta_1^3 < \beta_{SL94}^3 = 2/3$) and has the same degree of singularity of structures as burgulence. The singular dissipative structures with fractal dimension $D_{s,v} = 1.5$ can be conceived as perforated sheets reminiscent of the Sierpinski sieve. If the HS2 option is justified, then turbulence at Mach 6 is even more intermittent, but has the same degree of singularity of structures as incompressible turbulence ($\gamma_2 = \gamma_{SL94}$). In this case the fractal dimension of the most singular structures, $D_{s,v} = 2.2$, is slightly higher than in the B02 model (Boldyrev 2002). Formally, it is also possible that both types of

structures are present in highly compressible turbulence, implying multiple modulation defects and a compound nature of Poisson statistic cf. She and Waymire 1995. In this case, a linear combination of HS1 and HS2 models would describe the high order exponents best.

4.3 Conclusion

Using large-scale Euler simulations of supersonic turbulence at Mach 6 we have demonstrated that there exists an analogue of the K41 scaling laws valid for both weakly and highly compressible flows. The mass-weighted velocity $v \equiv \rho^{1/3}u$ – the primary variable governing the energy transfer through the cascade – should replace the velocity u in intermittency models for compressible flows at high Mach numbers.

Chapter 4, in full, is a reprint of "Scaling Laws and Intermittency in Highly Compressible Turbulence", which was published in 2007 in AIP Conference Proceedings, volume 932, page 393, by Alexei G. Kritsuk, Paolo Padoan, Rick Wagner, and Michael L. Norman. Reproduced with the permission of AIP Publishing. The dissertation author was the third investigator and author of this paper.

Chapter 5

Flux Correlations in Supersonic Isothermal Turbulence

Abstract

Using data from a large-scale three-dimensional simulation of supersonic isothermal turbulence, we have tested the validity of an exact flux relation derived analytically from the Navier–Stokes equation by Falkovich, Fouxon and Oz [2010 New relations for correlation functions in Navier–Stokes turbulence. *J. Fluid Mech.* **644**, 465]. That relation, for compressible barotropic fluids, was derived assuming turbulence generated by a large-scale force. However, compressible turbulence in simulations is usually initialized and maintained by a large-scale acceleration, as in gravity-driven astrophysical flows. We present a new approximate flux relation for isothermal turbulence driven by a large-scale acceleration, and find it in reasonable agreement with the simulation results.

5.1 Introduction

An important rigorously derived result in the statistical theory of incompressible turbulence is Kolmogorov's four-fifths law (Kolmogorov 1941a)

$$\left\langle [\delta u_{\parallel}(r)]^3 \right\rangle = -\frac{4}{5}\epsilon r, \tag{5.1}$$

where

$$\delta u_{\parallel}(r) \equiv [\boldsymbol{u}(\boldsymbol{r}) - \boldsymbol{u}(0)] \cdot \boldsymbol{r}/r \tag{5.2}$$

is the longitudinal velocity difference between two points, 0 and $\boldsymbol{r}$, separated by a distance $r = |\boldsymbol{r}|$, and ϵ is the mean energy dissipation rate. The four-fifths law is well supported experimentally and has been traditionally interpreted as a signature of a direct kinetic energy cascade within the inertial range of scales. Recent work by Falkovich, Fouxon, and Oz (2010, hereafter FFO), reinterprets the Kolmogorov relation in terms of currents and densities of the conserved quantities. From this new perspective, FFO derived an analytic scaling relation for barotropic flows with an arbitrary degree of compressibility, based on the correlation of currents and conserved fluxes in the inertial range.

There are few analytic relations proposed for compressible turbulence, e.g., the relation for an isothermal gas based on an effective energy transfer rate (Galtier and Banerjee 2011), and the lack of experimental results at high turbulent Mach numbers makes verification of such relations challenging. While supersonic turbulence is observed in molecular clouds (Heyer and Brunt 2004), and is believed to play an important role in star formation (McKee and Ostriker 2007), observations are limited by available angular resolution, projection and finite optical depth effects (Elmegreen and Scalo 2004). At present, numerical experiments provide the best route for testing theories of compressible turbulence (Benzi et al. 2008; Brandenburg and Nordlund 2011; Kitsionas et al. 2009;

Kritsuk et al. 2011; Sytine et al. 2000). Three-dimensional numerical simulations show that even at very high Mach numbers, a compressible energy cascade can be recovered with a proper density weighting, $\rho^{1/3}\boldsymbol{u}$ (Kritsuk et al. 2007a; Kritsuk et al. 2007b; Pan, Padoan, and Kritsuk 2009). Aluie, Li, and Li (2012) have recently presented evidence from simulations for pressure-dilatation action at large scales, and kinetic energy transfer to the dissipation scale via a conservative cascade, supporting an earlier analytical proof of the locality of kinetic energy transfer in compressible turbulence (Aluie 2011).

In this article, we use data from a simulation of supersonic isothermal turbulence to analyze the exact relation for compressible fluids from FFO. We are particularly interested in evaluating the role of the properties of the force that acts as a source of momentum and energy to compensate for the dissipation losses. We discuss the limitations implied by the assumptions used in the FFO derivation and present a new approximate scaling relation appropriate for the driving commonly used in numerical experiments on compressible fluid turbulence (e.g., Kritsuk et al. 2007b; Porter, Pouquet, and Woodward 2002; Schmidt, Federrath, and Klessen 2008; Wang et al. 2010).

5.2 Flux Correlations

A general relation is derived in FFO, along with a particular case for compressible turbulence in a barotropic fluid, both of which are based on two-point statistics in the inertial range. The lower limit of r where the relations are expected to hold is the scale where dissipation is negligible, while the upper limit is the correlation length of the force. The correlations involve the densities q^a, currents $\boldsymbol{j}^a$, fluxes F_i^a, and sources (forcing) f^a; the sources are assumed to be random, statistically stationary, spatially homogeneous, and isotropic. Beginning with the governing equations, and eliminating terms tied to

dissipation, a relation involving the densities, fluxes, and forcing is derived:

$$\nabla_i \left\langle q^a(0,t) F_i^a(\boldsymbol{r},t) \right\rangle = \left\langle q^a(0,t) f^a(\boldsymbol{r},t) \right\rangle, \tag{5.3}$$

see FFO for more detail.

In order to eliminate the scale dependence on the right-hand side of (5.3), the authors assume that on scales much smaller than the correlation length ($r \ll L$), the force is approximately constant,

$$f^a(0,t) \approx f^a(\boldsymbol{r},t). \tag{5.4}$$

Therefore, in the inertial range, the correlation of a density and its source is also approximately constant,

$$\langle q^a(0,t) f^a(\boldsymbol{r},t) \rangle \approx \langle q^a(0,t) f^a(0,t) \rangle \equiv \bar{\epsilon}_a. \tag{5.5}$$

Note that this ansatz is essentially carried over from the incompressible case, where there is no distinction between the large-scale acceleration and large-scale force since the density is constant. (While we recognize that the assumption of large-scale force per unit volume is possibly the only viable path to a rigorously derived relation–Galtier and Banerjee (2011) also relied on the same constraint to come up with an alternative relation for a compressible case–we emphasize that driving variable-density flows with a large-scale force would have a nontrivial effect on the turbulence statistics and eliminate the inertial range in the traditional sense, with respect to the fluid velocity.) Substituting $\bar{\epsilon}_a$ into (5.3) gives the correlation function

$$\nabla_i \left\langle q^a(0) F_i^a(\boldsymbol{r}) \right\rangle = \bar{\epsilon}_a, \tag{5.6}$$

and assuming isotropy, FFO derive the exact scaling relation in a vector form:

$$\langle q^a(0) F_i^a(\boldsymbol{r}) \rangle = \frac{\bar{\epsilon}_a r_i}{d}, \tag{5.7}$$

where d is the number of spatial dimensions.

In the case of a barotropic fluid, the densities are $\boldsymbol{q} = (\rho\boldsymbol{u}, \rho)$, and the fluxes are $F_i^j = \rho u_i u_j + p(\rho)\delta_{ij}$ for $i, j = 1, \ldots, d$, while $F_i^{d+1} = \rho u_i$. When dissipation is neglected, the governing equations are the forced Euler equations:

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho\, \boldsymbol{u}) = 0, \tag{5.8}$$

$$\partial_t(\rho u_i) + \partial_j(\rho u_i u_j + p\delta_{ij}) = f^i. \tag{5.9}$$

Substituting the densities, fluxes, and forcing into (5.7) gives a relation for isothermal turbulence when $p = \rho$,

$$\Phi(r) \equiv \left\langle [\rho(0)\boldsymbol{u}(0) \cdot \boldsymbol{u}(\boldsymbol{r})\rho(\boldsymbol{r})]\, u_\parallel(\boldsymbol{r}) \right\rangle + \left\langle \rho(0) u_\parallel(0)\rho(\boldsymbol{r}) \right\rangle = \frac{\bar{\epsilon} r}{d}, \tag{5.10}$$

where $u_\parallel = \boldsymbol{u} \cdot \boldsymbol{r}/r$ is the longitudinal velocity,

$$\bar{\epsilon} = \langle \rho(0)\boldsymbol{u}(0) \cdot \boldsymbol{f}(0) \rangle\,, \tag{5.11}$$

and we have taken the scalar product of both sides of (5.7) with $\boldsymbol{r}/r$ to get (5.10) in a more convenient scalar form. Unlike ϵ in (5.1), $\bar{\epsilon}$ is the injection rate of momentum squared, not energy. However, in the incompressible limit, the vector form of (5.10) reduces to a third-order velocity correlation function implying (5.1).

In numerical experiments of compressible turbulence, the flow is traditionally driven by a large-scale acceleration $\boldsymbol{a}$, such that $\boldsymbol{f} = \rho\boldsymbol{a}$, while the exact flux relation (5.10) was derived under the assumption that $\boldsymbol{f}$ is a smooth, large-scale, field. As noted in FFO, physically, $\boldsymbol{f}$ may come from a gradient of potential (e.g., external gravitational acceleration, i.e. $\boldsymbol{f} = \rho\nabla\phi$ or $\boldsymbol{a} = \nabla\phi$). The decorrelation and small-scale variations of the density at high Mach number make this an important point, as acceleration-driven

turbulence is important in astrophysics and other areas, such as turbulent convection.

Let us suggest a new flux relation appropriate for this case. Now, instead of the constant $\bar{\epsilon}$ in (5.11) we shall have the two-point fourth-order correlation function $\langle\rho(0)\rho(\boldsymbol{r})\boldsymbol{u}(0)\cdot\boldsymbol{a}(\boldsymbol{r})\rangle$, which is scale-dependent. The character of this scale-dependence can be found in the particular case of an acceleration short-correlated in time, when $\langle a_i(t)a_j(0)\rangle = \varepsilon_{ij}\delta(t)$. The average of the product of any quantity $U\{\boldsymbol{a}\}$ and a white noise $\boldsymbol{a}$ is expressed by the formula of Gaussian integration via a variational derivative: $\langle U\{\boldsymbol{a}\}a_i\rangle = \varepsilon_{ij}\langle\delta U/\delta a_j\rangle$. In the correlation function we consider, it is the velocity field which is related to the acceleration (by the equation $du_i/dt - u_i\nabla\cdot\boldsymbol{u} = a_i$) so that $\langle\delta u_i(t)/\delta a_j(t')\rangle = \delta_{ij}\theta(t-t')$ where the last factor is the step function. We now obtain

$$\langle\rho(0)\rho(\boldsymbol{r})\boldsymbol{u}(0)\cdot\boldsymbol{a}(\boldsymbol{r})\rangle = \langle\rho(0)\rho(\boldsymbol{r})\rangle\,\bar{\varepsilon} \tag{5.12}$$

where $\bar{\varepsilon} = \langle\boldsymbol{u}(0)\cdot\boldsymbol{a}(0)\rangle$. Since the acceleration is a large-scale field, then a two-point velocity-acceleration correlation function can be replaced by a single-point one. Let us stress that the constant $\bar{\varepsilon}$ is neither the total energy input nor the work of the external acceleration.

Of course, in the physical situations of interest, as well as in simulations, the acceleration cannot be considered short-correlated in time. In this case, the effective decoupling of the fourth moment into the product of second moments expressed by (5.12) is not an exact relation but can be suggested as a plausible approximation. We thus come to the generalization of the flux relation in the following form (up to an order-unity constant):

$$\nabla_r\Phi(r) \simeq \langle\rho(0)\rho(\boldsymbol{r})\rangle\bar{\varepsilon}. \tag{5.13}$$

For isothermal turbulence driven by a large-scale acceleration, (5.13) provides an approximate relation connecting the scaling of $\Phi(r)$ with that of $\langle\rho(0)\rho(\boldsymbol{r})\rangle$ in the intertial

range.

5.3 Experimental Verification

To appraise the relations from FFO, we have used data from a simulation designed to study the inertial range statistics of highly compressible turbulence (Kritsuk et al. 2007b). The simulation was performed with an implementation of the piecewise parabolic method (Colella and Woodward 1984) in the Enzo code (Norman et al. 2007). The forced Euler equations for an isothermal gas, (5.8) and (5.9), were solved numerically in a cubic periodic domain with a linear size of $L = 1$ along each axis, and covered by a uniform Cartesian grid of 1024^3 zones, each having sides of length $\Delta = L/1024$. The turbulent rms Mach number, $M = 6$, and a steady rate of energy injection were maintained by a random acceleration field with power limited to wavenumbers $k/k_{min} \in [1, 2]$, where $k_{min} = 2\pi/L$. The spatially fixed acceleration field was normalized at every time step so that $\langle \rho \boldsymbol{u} \cdot \boldsymbol{a} \rangle$ was constant; the normalization factor had a standard deviation of $\sim 5\%$ during the simulation. For this work, we used 43 full data snapshots evenly distributed in time in the range $t/\tau \in [6, 10]$, where the flow crossing (large eddy turnover) time, $\tau \equiv L/2M$, assumes the sound speed of unity. For each snapshot, we computed $\bar{\epsilon}$, $\bar{\varepsilon}$, and evaluated $\Phi(r)$ for 16 discrete r-values in the range $r/\Delta \in [8, 128]$, using $2^{32} \approx 4 \times 10^9$ random point pairs for each value of r.

To evaluate the role of various assumptions concerning the forcing and assess the hypotheses employed by FFO, we compare Φ based on the exact forms (5.6) and (5.10) with an approximate expression based on (5.13). We begin by examining the spatial distribution of the force and acceleration in the lower right-hand panels of figure 5.1, which show the values of the x component of $\boldsymbol{a}$ and $\boldsymbol{f}'$, in a slice through the simulation volume at $x = 0$, where $\boldsymbol{f}' \equiv \rho/\bar{\rho}\boldsymbol{a}$ is the force normalized by the average density. A more quantitative

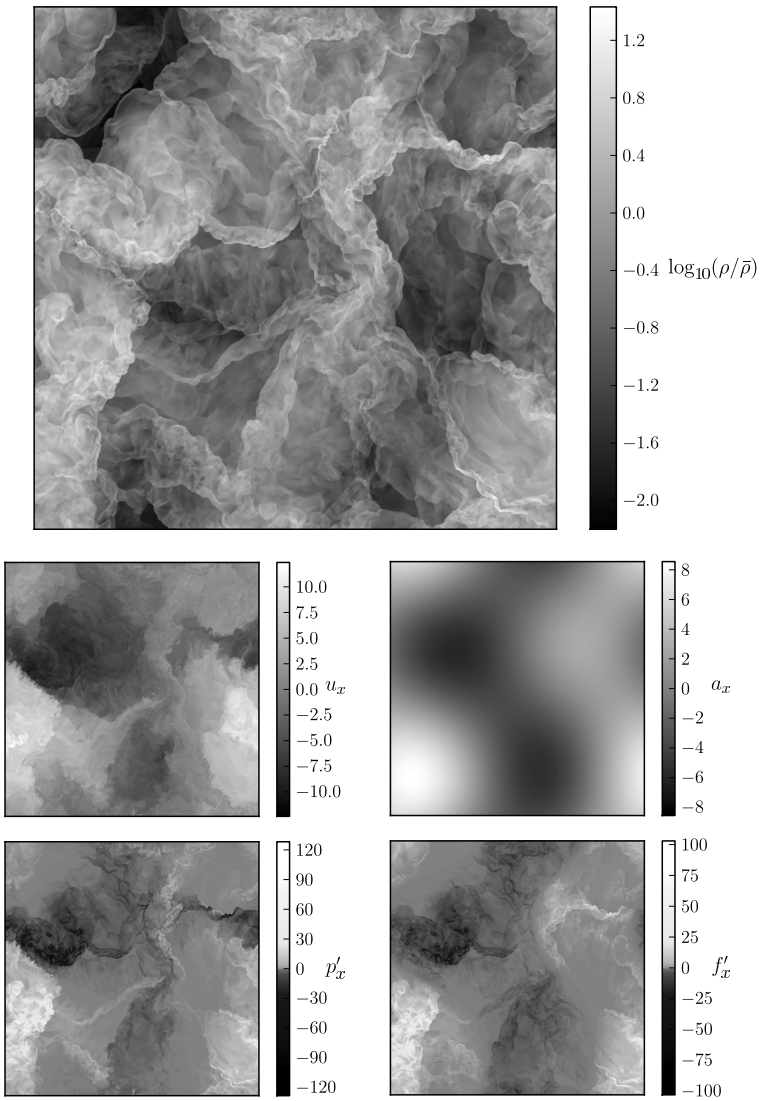

Figure 5.1: Slices of several instantaneous dynamical fields through the plane $x = 0$: (a) logarithm of the normalized density field $\log_{10}(\rho/\bar{\rho})$; (b) velocity field u_x (velocity is normal to the image plane); (c) normalized momentum $p'_x = \rho/\bar{\rho}\ u_x$; (d) acceleration field a_x; (e) normalized force $f'_x = \rho/\bar{\rho}\ a_x$. N.B.: To account for the distribution of the density field, the colour maps used for p'_x and f'_x are highly compressed around 0.

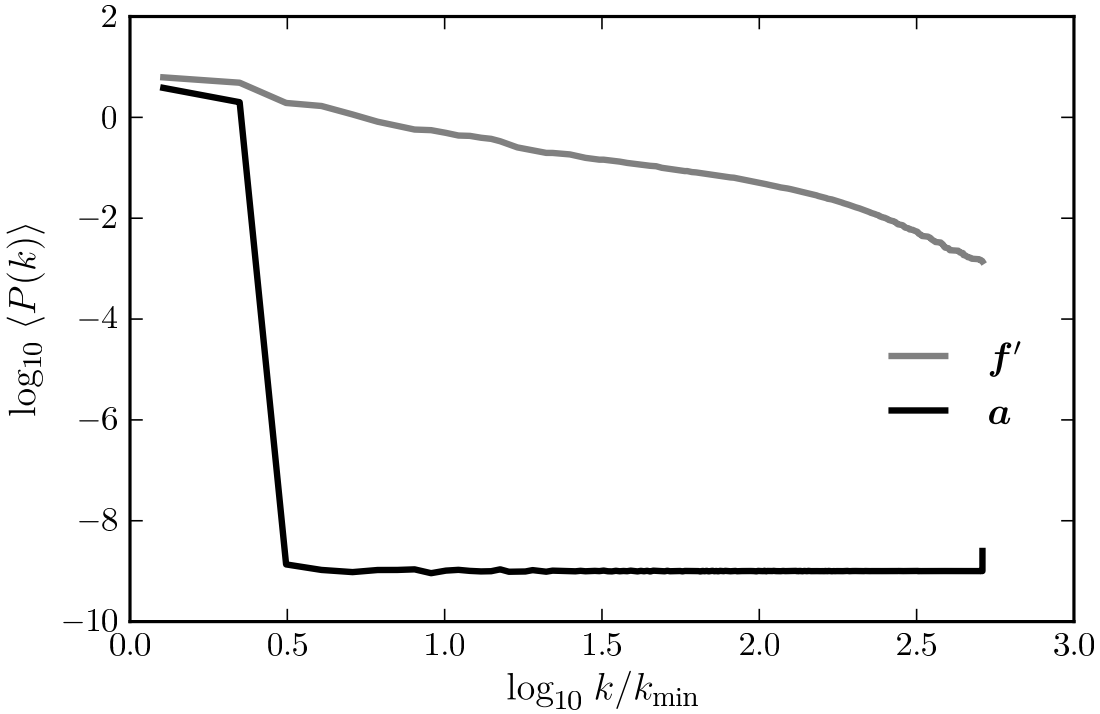

Figure 5.2: Power spectra of the driving acceleration, $\boldsymbol{a}$ (solid black line), and the normalized force, $\boldsymbol{f}' = \rho\boldsymbol{a}/\bar{\rho}$ (solid grey line), from a single flow snapshot.

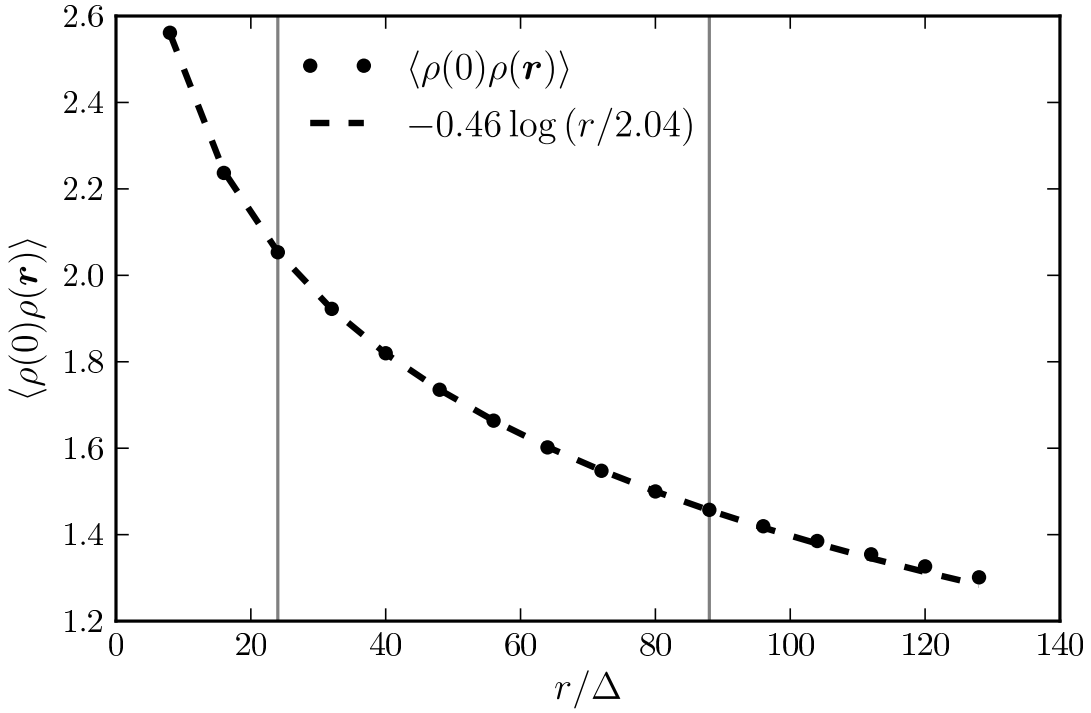

Figure 5.3: Dots: time-averaged density correlation; the standard error of the mean of $\langle\rho(0)\rho(\boldsymbol{r})\rangle$ is < 0.02. Dashed line: best fit to the data points with $a\log(r/b)$ in the range $r/\Delta \in [20, 90]$, as indicated by the vertical grey lines.

assessment can be found in figure 5.2, which shows the power spectra of both $\boldsymbol{a}$ and $\boldsymbol{f}'$. The power spectrum of $\boldsymbol{a}$ falls off at $k/k_{min} > 2$, while the spectrum of $\boldsymbol{f}'$ slowly decreases with k, and the two are separated by several orders of magnitude above $k/k_{min} \approx 2$. Further, we measure $\bar{\epsilon} = 589$, $\bar{\varepsilon} = 149$, $\langle\rho^2\rangle = 3.74$, and find that

$$\langle\rho^2\rangle\,\bar{\varepsilon} = 0.95\;\bar{\epsilon}, \tag{5.14}$$

supporting the decoupling suggested in (5.12). This decoupling, combined with the features of $\boldsymbol{f}$ shown by figures 5.1 and 5.2, indicates that the assumption of an approximately constant force does not hold in the simulation, as expected.

Next, we will compare Φ to the scaling from (5.13). We find that the density

correlation is well approximated by a logarithmic form:

$$\langle \rho(0)\rho(\boldsymbol{r})\rangle \simeq a \log\left(\frac{r}{b}\right) \tag{5.15}$$

with $a = -0.46 \pm 0.001$ and $b = 2.04 \pm 0.01$ when fitted in the range $r/\Delta \in [20, 90]$ (see figure 5.3). This is expected from the density power spectrum approximately following $\sim k^{-1}$ at this particular turbulent Mach number (Kritsuk et al. 2007b). Substituting (5.15) into (5.13) and integrating radially from 0 to r assuming isotropy, we have

$$\Phi(r) \simeq \frac{a\bar{\varepsilon}}{9}\left[3\log\left(\frac{r}{b}\right) - 1\right] r. \tag{5.16}$$

Note that (5.16) is a good approximation only for turbulence at $M = 6$. Simulations with different Mach numbers will have density power spectra with different slopes, and their density correlations will have different forms.

Figure 5.4 shows the time-averaged $\Phi(r)$, the expected scaling for a large-scale force using (5.10), and the expected scaling based on a large-scale acceleration and our approximation (5.16). The accuracy of the approximate flux relation can be assessed in figure 5.5, where we show $\Phi(r)$ normalized by the right-hand side of (5.16), using the values fitted to the density correlation. The right-hand side of (5.16) follows the actual measured $\Phi(r)$ within $\pm 5\%$ in the range $r/\Delta \in [10, 100]$, somewhat underestimating Φ at the lower end of the range, and overestimating it as r increases; this behaviour can be predicted as the assumed approximations break down for two different reasons. First, from (5.14), the density correlation and injection rate of specific kinetic energy are not totally decoupled; this may account for (5.13) underestimating Φ at small scales. Second, as r approaches $L/2$, $\langle \boldsymbol{u}(0) \cdot \boldsymbol{a}(\boldsymbol{r})\rangle \to 0$, and as (5.4) breaks down, a constant $\bar{\varepsilon}$ on the right-hand side of (5.13) causes it to excede $\Phi(r)$. This departure by $\Phi(r)$ from (5.16) at large scales is determined by the numerical resolution and driving scales; given a similar simulation with

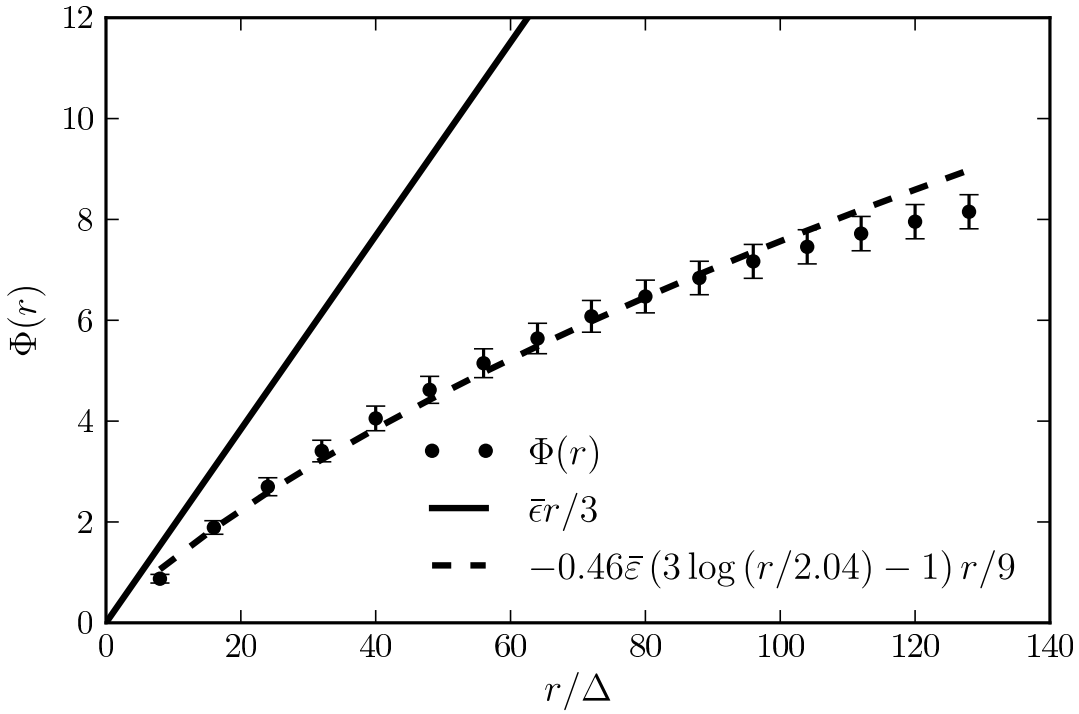

Figure 5.4: Dots: time-averaged measure of $\Phi(r)$; the error bars show the standard error of the mean. Solid line: expected scaling based on a large-scale force. Dashed line: expected approximate scaling based on the density correlation. Note that the dashed line is not a fit to the $\Phi(r)$ points, but shows the right-hand side of (5.16) with parameters determined by the best fit to the density correlation (5.15) shown in figure 5.3.

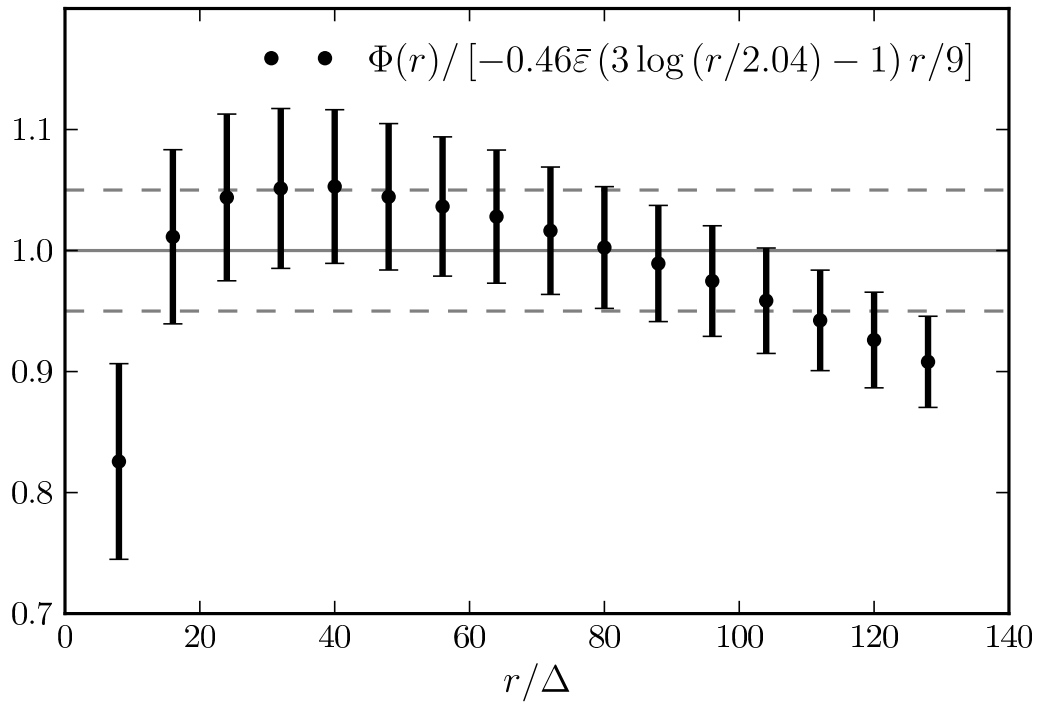

Figure 5.5: Time-averaged measure of $\Phi(r)$ normalized by the expected approximate scaling (5.16). The dashed grey lines show $\pm 5\%$ tolerance levels.

higher numerical resolution, we would expect the agreement to scale relative to the smallest driving scale.

5.4 Conclusion

We find that the approximate relation we derived for supersonic isothermal turbulence driven by a large-scale acceleration field (5.13) agrees reasonably well with the simulation results within a sufficiently wide range of scales consistent with the inertial range presence established in Kritsuk et al. (2007b). This implies support for the original FFO exact relation (5.3), so it can be anticipated that in a system where (5.5) holds, the exact relation (5.10) will be satisfied as well. Therefore, figures 4 and 5 demonstrate the strong sensitivity of the flux correlations in supersonic isothermal turbulence to the nature of forcing supporting the statistically stationary state. Such sensitivity has been observed in a similar context of the forced one-dimensional Burgers equation, where an invasive random force with a $\propto k^{-1}$ spectrum (compare with the grey line in figure 5.2) would produce a non-universal bifractal scaling (Mitra et al. 2005). Likewise, Bahraminasab et al. (2008) showed that the same Burgers system driven by a force with large-scale correlation in space and with a Wiener scaling in time exhibits a non-universal scaling for time increments larger than the correlation time of the force. Comparison with other relations and results addressing energy transfer in compressible turbulence (Aluie 2011; Aluie, Li, and Li 2012; Galtier and Banerjee 2011) will help us to understand whether non-universality, observed here due to the driving mechanism, is a more general property of highly compressible flows.

Chapter 5, in full, is a reprint of "Flux Correlations in Supersonic Isothermal Turbulence", which was published in 2012 in Journal of Fluid Mechanics, volume 713, page 482, by Rick Wagner, Gregory Falkovich, Alexei G. Kritsuk,and Michael L. Norman. Reprinted with permission. The dissertation author was the first investigator and author

of this paper.

Chapter 6

Energy Cascade and Scaling in Supersonic Isothermal Turbulence

Abstract

Supersonic turbulence plays an important role in a number of extreme astrophysical and terrestrial environments, yet its understanding remains rudimentary. We use data from a three-dimensional simulation of supersonic isothermal turbulence to reconstruct an exact fourth-order relation derived analytically from the Navier–Stokes equations (Galtier and Banerjee, *Phys. Rev. Lett.*, vol. 107, 2011, p. 134501). Our analysis supports a Kolmogorov-like inertial energy cascade in supersonic turbulence previously discussed on a phenomenological level. We show that two compressible analogues of the four-fifths law exist describing fifth- and fourth-order correlations, but only the fourth-order relation remains 'universal' in a wide range of Mach numbers from incompressible to highly compressible regimes. A new approximate relation valid in the strongly supersonic regime is derived and verified. We also briefly discuss the origin of bottleneck bumps in simulations of compressible turbulence.

6.1 Introduction

Supersonic turbulence is believed to play a key role in a wide range of extreme astrophysical and terrestrial environments; for example, regulating star formation in molecular clouds (Hennebelle and Falgarone 2012), feeding supermassive black holes (Hobbs et al. 2011), creating clumpy structure in hot winds from Wolf-Rayet stars (Moffat and Robert 1994), controlling air entrainment in high-pressure volcanic eruptions (Ogden, Glatzmaier, and Wohletz 2008), and affecting fuel mixing and combustion efficiency in scramjets (Ingenito and Bruno 2010).

Compared to incompressible turbulence, highly compressible turbulent flows are more complex due to nonlinear coupling of the velocity, density and pressure fields. Shock waves and vortex sheets change the topology of intermittent dissipative structures in supersonic turbulence (Pan, Padoan, and Kritsuk 2009). A 'universal' scaling of the mass-weighted velocity $\boldsymbol{v} \equiv \rho^{1/3}\boldsymbol{u}$ was demonstrated in numerical experiments (Kritsuk et al. 2007a; Kritsuk et al. 2007b) and independently verified in a number of numerical studies (Federrath et al. 2010; Kowal and Lazarian 2007; Price and Federrath 2010; Schmidt, Federrath, and Klessen 2008; Schwarz et al. 2010a), suggesting, by dimensional arguments, the presence of an inertial cascade. More recently, analytical scaling relations for compressible turbulence were derived and analyzed (Banerjee and Galtier 2013; Falkovich, Fouxon, and Oz 2010; Galtier and Banerjee 2011; Wagner et al. 2012) and the existence of an intermediate scaling range dominated by inertial dynamics was demonstrated rigorously based on coarse-graining (Aluie 2013; Aluie 2011; Aluie, Li, and Li 2012). This contribution reports on the verification of the new relation presented in Galtier and Banerjee (2011) with data from a Mach 6 simulation (Kritsuk et al. 2007b) and on the phenomenology that follows from these results.

6.2 A fourth-order relation

Consider a system of Navier–Stokes equations for an isothermal compressible fluid in three dimensions

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0, \tag{6.1}$$

$$\partial_t (\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}) + \boldsymbol{\nabla} p = \eta \Delta \boldsymbol{u} + \frac{\eta}{3} \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{u}) + \boldsymbol{f}, \tag{6.2}$$

where $p = c_\mathrm{s}^2 \rho$ is the pressure, c_s is the speed of sound, $\eta > 0$ is the dynamic viscosity (η is constant in space and time in isothermal flows) and $\boldsymbol{f}(\boldsymbol{x}, t)$ is a random force. Besides the usual conservation of mass and momentum expressed by (6.1) and (6.2), let us mention two additional ideal integral invariants in unforced isothermal fluids: (i) the total energy density,

$$\langle E \rangle \equiv \langle \rho u^2/2 + \rho e \rangle, \tag{6.3}$$

where $e = c_\mathrm{s}^2 \ln(\rho/\rho_0)$ is the specific isothermal compressive potential energy, ρ_0 is the mean density of the fluid and angle brackets $\langle \ldots \rangle$ in this context indicate average over the volume of the fluid, $\frac{1}{V} \int_V (\ldots) \mathrm{d}V$; and (ii) the mean kinetic helicity,

$$\langle H \rangle \equiv \langle \boldsymbol{u} \cdot \boldsymbol{\omega} \rangle / 2, \tag{6.4}$$

where $\boldsymbol{\omega} \equiv \boldsymbol{\nabla} \times \boldsymbol{u}$ is the vorticity. In the forced system, the evolution of total energy density is determined by the balance between the action of large-scale force and small-scale viscous dissipation

$$\partial_t \langle E \rangle = \langle \epsilon \rangle - \eta \left\langle \omega^2 - 4d^2/3 \right\rangle, \tag{6.5}$$

where $\epsilon \equiv \boldsymbol{u} \cdot \boldsymbol{f}$ is the local energy injection rate and $d \equiv \boldsymbol{\nabla} \cdot \boldsymbol{u}$ is the dilatation.

Assuming that a statistical steady state exists at $Re \gg 1$, the following relation for

homogeneous turbulence in the inertial interval can be derived (Galtier and Banerjee 2011)

$$\langle d'\left(R-2E-p\right)+d\left(R'-2E'-p'\right)\rangle+\boldsymbol{\nabla}_{\boldsymbol{r}}\cdot\left\langle\left[\delta(\rho\boldsymbol{u})\cdot\delta\boldsymbol{u}+2\delta\rho\delta e\right]\delta\boldsymbol{u}+\tilde{\delta}e\delta(\rho\boldsymbol{u})\right\rangle=-4\varepsilon, \tag{6.6}$$

note a different pressure-dilatation term placement (Banerjee and Galtier 2013). Here $\delta\boldsymbol{q}(r)\equiv\boldsymbol{q}(\boldsymbol{x}')-\boldsymbol{q}(\boldsymbol{x})$ is the increment in quantity $\boldsymbol{q}$ corresponding to the increment $\boldsymbol{r}=\boldsymbol{x}'-\boldsymbol{x}$, $r\equiv|\boldsymbol{r}|$, $R\equiv\rho\boldsymbol{u}\cdot\boldsymbol{u}'+2\rho e'$, $R'\equiv\rho'\boldsymbol{u}'\cdot\boldsymbol{u}+2\rho' e$, $\tilde{\delta}q\equiv q'+q$, $\boldsymbol{\nabla}_{\boldsymbol{r}}\equiv\hat{\mathbf{e}}_i\partial/\partial r_i$ denotes partial derivatives with respect to the increment $\boldsymbol{r}$, $\hat{\mathbf{e}}=\{\hat{\mathbf{e}}_1,\hat{\mathbf{e}}_2,\hat{\mathbf{e}}_3\}$ is an orthonormal basis,

$$\varepsilon=\langle\boldsymbol{u}'\cdot\boldsymbol{f}+\boldsymbol{u}\cdot\boldsymbol{f}'+\boldsymbol{u}'\cdot\boldsymbol{f}\rho'/\rho+\boldsymbol{u}\cdot\boldsymbol{f}'\rho/\rho'\rangle/4 \tag{6.7}$$

is the mean energy density injection rate and $\langle\ldots\rangle$ denote an ensemble-average.

Equation (6.6) can be written in symbolic form as

$$S(r)+\boldsymbol{\nabla}_{\boldsymbol{r}}\cdot\boldsymbol{F}=-4\varepsilon, \tag{6.8}$$

where $\boldsymbol{F}(r)$ is the total energy flux vector and $S(r)$ represents 'source' terms on the left-hand side of (6.6) that depend on the potential component of the velocity and can be expressed via increments

$$S(r)=\left\langle\left[\delta(d\rho\boldsymbol{u})-\tilde{\delta}d\delta(\rho\boldsymbol{u})\right]\cdot\delta\boldsymbol{u}+2\left[\delta(d\rho)-\tilde{\delta}d\delta\rho\right]\delta e+\delta d\delta p-2dp\right\rangle. \tag{6.9}$$

On the basis of empirical evidence (see § 3 below), in supersonic turbulence the pressure-dilatation contribution to the source at $r=0$ is positive, $S(0)=-2\langle dp\rangle>0$. Since both e and p are proportional to $c_{\rm s}^2$, only the first term in (6.9) will contribute to $S(r)$ at high

Mach numbers ($c_\mathrm{s} \to 0$)

$$S(r) \approx \mathscr{S}(r) \equiv \left\langle [\delta(d\rho\boldsymbol{u}) - \tilde{\delta}d\,\delta(\rho\boldsymbol{u})] \cdot \delta\boldsymbol{u} \right\rangle \tag{6.10}$$

(cf. equation (16) in Galtier and Banerjee 2011). Vector relation (6.8) should be compared to a primitive form of Kolmogorov's (1941) exact and nontrivial four-fifths law,

$$\rho_0 \boldsymbol{\nabla}_{\boldsymbol{r}} \cdot \langle (\delta\boldsymbol{u})^2 \delta\boldsymbol{u} \rangle = -4\varepsilon \tag{6.11}$$

(e.g. Frisch 1995, equation (6.8)), which follows from (6.6), assuming incompressibility.

If the force in (6.2) is expressed in terms of the external acceleration, $\boldsymbol{f} \equiv \rho\boldsymbol{a}$, then

$$\varepsilon = \langle \rho\boldsymbol{u}' \cdot \boldsymbol{a} + \rho'\boldsymbol{u} \cdot \boldsymbol{a}' + \rho'\boldsymbol{u}' \cdot \boldsymbol{a} + \rho\boldsymbol{u} \cdot \boldsymbol{a}' \rangle / 4. \tag{6.12}$$

In isotropic turbulence,

$$\varepsilon = \langle \rho\boldsymbol{u}' \cdot \boldsymbol{a} + \rho'\boldsymbol{u}' \cdot \boldsymbol{a} \rangle / 2 = \langle \tilde{\delta}\rho\,(\boldsymbol{u}' \cdot \boldsymbol{a}) \rangle / 2. \tag{6.13}$$

For incompressible fluids, it can be shown that $\varepsilon(r) = \rho_0 \langle \boldsymbol{u}' \cdot \boldsymbol{a} \rangle \approx \rho_0 \langle \boldsymbol{u} \cdot \boldsymbol{a} \rangle \equiv \rho_0 \bar{\varepsilon}$, if the acceleration $\boldsymbol{a}(\boldsymbol{x}, t)$ (and hence the force $\rho_0\boldsymbol{a}$) operate at large scales only. Here, $\bar{\varepsilon}$ denotes the (constant) average energy injection rate per unit mass.

This conventional technique, however, cannot be carried over to compressible fluid turbulence. In the limit of large correlation length L_a of the acceleration $\boldsymbol{a}$, for $r \ll L_a$ the second term in (6.13) can be reduced to a constant $\langle \rho'\boldsymbol{u}' \cdot \boldsymbol{a} \rangle \approx \langle \rho\boldsymbol{u} \cdot \boldsymbol{a} \rangle$, while the first cannot. If the force $\boldsymbol{f}(\boldsymbol{x}, t)$ had a large correlation length L_f instead, the first term in (6.13) would reduce to the same combination $\langle \rho\boldsymbol{u}' \cdot \boldsymbol{a} \rangle = \langle \boldsymbol{u}' \cdot \boldsymbol{f} \rangle \approx \langle \boldsymbol{u} \cdot \boldsymbol{f} \rangle = \langle \rho\boldsymbol{u} \cdot \boldsymbol{a} \rangle$ for $r \ll L_f$, but the second could not be decoupled simultaneously. On the basis of (6.13), in

supersonic turbulence the large-scale external force assumption is incompatible with the presence of inertial interval due to strong density variations on all scales (see also Wagner et al. 2012). If the large-scale acceleration is also short-correlated in time, then the density can be decoupled for r in the inertial interval $\langle \rho \boldsymbol{u}' \cdot \boldsymbol{a} \rangle \approx \langle \rho \rangle \langle \boldsymbol{u} \cdot \boldsymbol{a} \rangle$ (e.g. Wagner et al. 2012). In this case, (6.13) would reduce to

$$\varepsilon \approx \rho_0 \langle \boldsymbol{u} \cdot \boldsymbol{a} \rangle = \rho_0 \bar{\varepsilon} = \text{const.} \tag{6.14}$$

In reality, the large-scale acceleration (e.g. the free-fall acceleration in turbulent convection) is often not short-correlated in time. Also in simulations, forcing is routinely employed to mimic the energy cascade incoming from scales larger than the box size; thus the correlation time cannot be shorter than the large eddy turnover time and it is hard to expect the decoupling in the form of (6.14).

Nevertheless, the density field in supersonic turbulence has a very short correlation length $L_\rho \ll L_u \ll L_a$. Since $\boldsymbol{u}$ and $\boldsymbol{a}$ are larger-scale fields, while ρ related to the velocity gradient is a small-scale quantity controlled by the nonlinearity of governing equations (6.1) and (6.2), one can expect that $\langle \rho \boldsymbol{u}' \cdot \boldsymbol{a} \rangle \approx \langle \rho' \boldsymbol{u}' \cdot \boldsymbol{a} \rangle \approx \langle \rho \boldsymbol{u} \cdot \boldsymbol{a} \rangle$. Hence, even for $\boldsymbol{a}$ with a finite correlation time, the approximation

$$\varepsilon(r) \approx \langle \rho \boldsymbol{u} \cdot \boldsymbol{a} \rangle = \varepsilon_0 \tag{6.15}$$

is justified for $L_\rho \lesssim r \ll L_a$. Using a different approach based on coarse-graining, Aluie (2013) rigorously proved that the energy injection rate is constant at scales sufficiently separated from the injection scale, if the external acceleration used to support a statistical steady state is restricted to large scales. We will show how well (6.14) and (6.15) hold in § 6.3.

For now, however, let us assume $\varepsilon(r) = \varepsilon_0$ and integrate (6.8) over a ball of radius

r, to obtain an approximate scalar relation for isotropic turbulence in symbolic form:

$$Q(r) + F_{\parallel}(r) \simeq -\frac{4}{3}\varepsilon_0 r, \tag{6.16}$$

where the source function

$$Q(r) \equiv \frac{1}{r^2}\int_0^r S(r) r^2 \mathrm{d}r \tag{6.17}$$

and the longitudinal flux of total energy

$$F_{\parallel}(r) \equiv \boldsymbol{F}\cdot\boldsymbol{r}/r = \left\langle \left[\delta(\rho\boldsymbol{u})\cdot\delta\boldsymbol{u} + 2\delta\rho\delta e\right]\delta u_{\parallel} + \tilde{\delta} e \delta(\rho u_{\parallel})\right\rangle. \tag{6.18}$$

The inertial part of the flux dominates at high Mach numbers ($c_{\mathrm{s}} \to 0$):

$$F_{\parallel}(r) \approx \mathcal{F}_{\parallel}(r) \equiv \left\langle \delta(\rho\boldsymbol{u})\cdot\delta\boldsymbol{u}\,\delta u_{\parallel}\right\rangle \tag{6.19}$$

– see also (6.10). While approximation (6.19) does not include the flux of compressive energy density ρe, compressibility is still partly accounted for by the momentum difference $\delta(\rho\boldsymbol{u})$. Also note that $F_{\parallel}(0) = 0$, $\mathcal{F}_{\parallel}(0) = 0$ and $Q(0) = 0$.

6.3 Numerical verification

To evaluate (6.16), we shall use data from a numerical experiment designed to study the inertial range statistics of supersonic homogeneous isotropic turbulence (Kritsuk et al. 2007b). The simulation was carried out following the traditional implicit large eddy simulation (ILES) approach to the modeling of turbulent flows with strong shocks (Fernando F. Grinstein and Rider 2007), using an implementation of the piecewise parabolic method (PPM, Colella and Woodward 1984) in the Enzo code (O'Shea et al. 2004). The forced Euler equations for an isothermal fluid with the mean density $\rho_0 = 1$ were

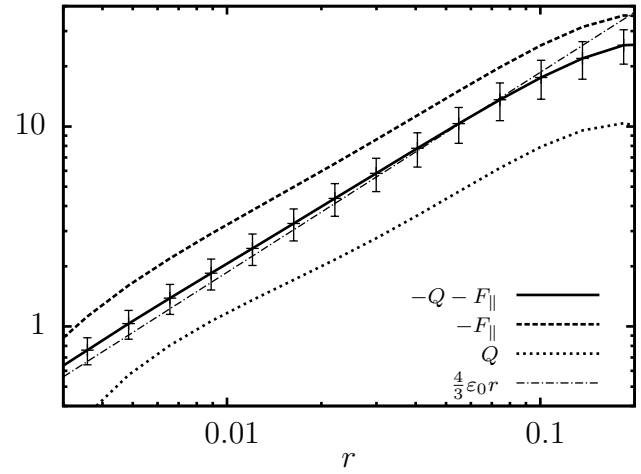

Figure 6.1: Time-average scaling for the left-hand side and right-hand side of (6.16), including individual contributions from the flux and source terms. Note that $F_\parallel(r) < 0$ (direct energy cascade) and $S(r) > 0$ (effective additional energy injection). Error bars indicate $\pm 1\sigma$ variation of the left-hand side of (6.16) in our sample of 86 flow snapshots.

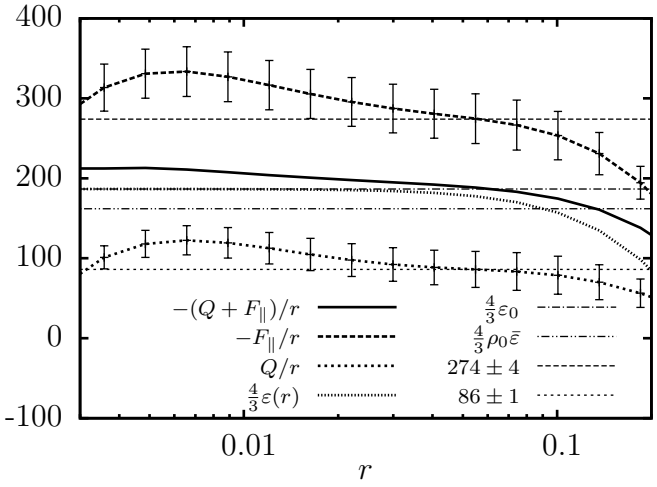

Figure 6.2: Compensated scaling for the flux and source terms from (6.16). Also shown are various proxies for the energy injection rate, including the two-point momentum–acceleration and velocity–force correlation functions, as well as single-point averages with and without density-weighting. Error bars indicate $\pm 1\sigma$ variation of the flux and source terms in our sample.

numerically integrated in a cubic periodic domain with a linear size $L = 1$ covered by a uniform Cartesian grid of 1024^3 cells. The turbulent r.m.s. Mach number, $M \sim 6$, and a steady rate of the kinetic energy injection were maintained by a random large-scale acceleration field, $\boldsymbol{a}(\boldsymbol{x}, t) = C(t)\boldsymbol{a}_0(\boldsymbol{x})$, with power limited to wavenumbers $k/k_{\rm min} \in [1, 2]$, where $k_{\rm min} = 2\pi/L$. The spatially fixed external acceleration $\boldsymbol{a}_0(\boldsymbol{x})$ was normalized at every time step to keep the energy injection rate approximately constant in time, $\varepsilon_0 = 140$; the normalization factor $C(t)$ had a standard deviation of $\sim 5\%$ during the simulation. For this work, we used a subset of 86 full data snapshots evenly distributed in the range $t/\tau \in [6, 10]$, where the flow crossing time $\tau \equiv L/(2c_{\rm s}M) \simeq 0.08$. For each snapshot, we computed $\bar{\varepsilon}$ and evaluated $S(r)$, $Q(r)$, $F_\parallel(r)$, and $\varepsilon(r) = \langle\tilde{\delta}\rho\,(\boldsymbol{u}' \cdot \boldsymbol{a})\rangle/2$ for 16 discrete values of increment r from the interval $r/\Delta \in [8, 128]$, where Δ is the grid spacing, using $2^{31} \approx 2 \times 10^9$ randomly selected point pairs for each value of r.

Figure 6.1 compares the scaling of $-(Q + F_\parallel)$ with the analytical prediction $4\varepsilon_0 r/3$, indicating that the approximate relation (6.16) holds reasonably well. Also shown are individual contributions for Q and $F_\parallel$. As expected for direct energy cascade, the flux is negative across the inertial interval. The source function is positive and a factor of ≈ 3.2 smaller than the flux. It represents the net effect of mean dilatation at scale r (conditioned on the energy density at this scale) on the associated energy flux. The source can be understood as a (positive) correction to ε_0, associated with the evolving metric in parts of the volume that are subject to compression. In a different context, such 'adiabatic heating' of compressible turbulent fluids was recently considered by Robertson and Goldreich (2012).

Figure 6.2 provides more detail by showing compensated scaling of various terms in (6.16); note that the inertial sub-range is limited to $r \in [0.03, 0.1]$ (Kritsuk et al. 2007b). On smaller scales, a bump indicating possible bottleneck contamination is clearly visible; on larger scales, the action of force is felt directly and $\varepsilon(r)$ starts to decline. The energy injection rate ε_0 defined in (6.15) gives an accurate measure for $\varepsilon(r) \approx 140$ in the inertial

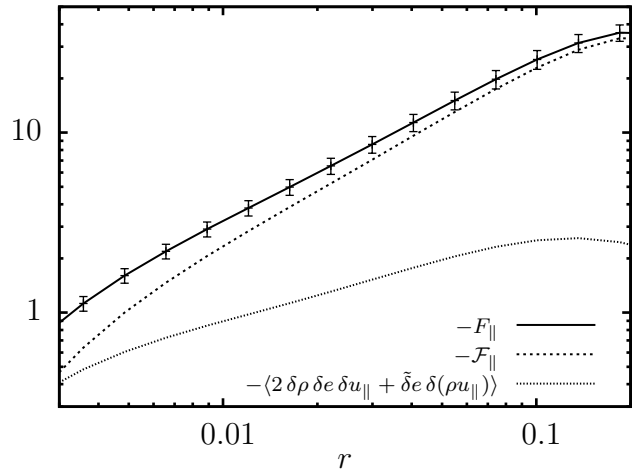

Figure 6.3: Scaling of the longitudinal flux of total, kinetic and compressive energy. At $M \simeq 6$, the kinetic energy flux strongly dominates over compressive flux terms. The exponents of the least-squares fits for $r \in [0.03, 0.08]$ are 0.91 ± 0.01, 0.99 ± 0.01 and 0.44 ± 0.02 for the total, kinetic and compressive fluxes, respectively. Error bars indicate $\pm 1\sigma$ variation of the total flux.

interval, while (6.14), which ignores the density–velocity correlation, underestimates the injection rate by $\sim 15\%$. The inertial range levels of the flux and source terms, -274 ± 4 and 86 ± 1 respectively, are estimated from the least-squares fits. The source terms play a relatively minor role across the inertial range; both individual flux and source contributions in (6.16) scale roughly linearly with r, indicating that $S(r) \approx$ const. This in turn implies that the kinetic energy cascades conservatively without substantial scale-dependent 'leakage' to the compressive potential energy (see also Aluie, Li, and Li 2012).

Figure 6.3 presents an analysis of different components of the flux in (6.18). A contribution from the inertial term $\mathcal{F}_\parallel$ dominates strongly on all scales, as predicted by (6.19). Two other terms related to the compressive energy flux are subdominant in the inertial range. The last term in (6.18) is ~ 1.6 dex smaller than the second.

Figure 6.4 shows various constituents of $S(r)$ listed in (6.9). As suggested by (6.10), the dominant contribution comes from dynamic-pressure-dilatation terms proportional to $\delta \boldsymbol{u}$; it is positive in a wide range of r and approximately constant ($\pm 4\%$) at $r \in [0.006, 0.1]$.

Source terms associated with the specific compressive energy difference δe contribute positively on small scales $r \lesssim 0.03$ and act as a sink at $r \gtrsim 0.03$. While their effect on the inertial range is minimal, they are responsible for an $\approx 40\%$ excess in $S(r)$ centered around $r \simeq 0.004$. The resulting small-scale excess in positive $S(r)$ is in turn responsible for the excess in negative $F_{\parallel}(r)$ (see figure 6.2). The impact of pressure-dilatation terms $\langle \delta d \delta p - 2dp \rangle$ on $S(r)$ is minor and they can be ignored in the inertial range. As $r \to 0$, the source is finite and positive: $S(r) \to -2\langle dp \rangle \simeq 73$. The average pressure dilatation may depend on various factors such as the Mach number, adopted equation of state, numerical resolution and so on. While for isothermal turbulence at $M \sim 6$ we obtain $\langle dp \rangle / (\gamma M^2) \approx -1$, also negative but substantially smaller ($\lesssim 0.04$) absolute values are reported by Aluie, Li, and Li (2012) and Wang et al. 2012 for ideal gas models at $M \lesssim 1$. At high Mach numbers, the p.d.f. of dilatation is strongly skewed towards negative values, making the average pressure dilatation finite and negative.

Since approximations (6.10) and (6.19) hold at $M \gg 1$, (6.16) can be reduced to

$$\mathcal{Q}(r) + \mathcal{F}_{\parallel}(r) \simeq -\frac{4}{3}\mathcal{C}\varepsilon_0 r, \tag{6.20}$$

where the source function $\mathcal{Q}(r) \equiv r^{-2} \int_0^r \mathscr{S}(r) r^2 \mathrm{d}r$ is analogous to (6.17) and $\mathcal{C} < 1$ is a constant of order unity accounting for the fraction of injected kinetic energy that goes into excitation of compressive modes that were ignored in (6.10) and (6.19). Figure 6.5 illustrates the quality of approximations at $M \simeq 6$, where $\mathcal{C} \simeq 0.84$, i.e. $\approx 16\%$ of the energy input supports modes related to the compressive energy and pressure dilatation. Note that at $M \simeq 6$ the right-hand side of (6.20) can be replaced by the incompressible expression $-4\rho_0 \bar{\varepsilon}/3$ without substantial loss of accuracy: see (6.14) for the definition of $\bar{\varepsilon}$. As $\mathscr{S}(r) \approx \mathscr{S}_0$ is nearly constant in the inertial interval, $\mathcal{F}_{\parallel}(r) \simeq -4\varepsilon_{\text{eff}} r/3$, where $\varepsilon_{\text{eff}} = \mathcal{C}\varepsilon_0 + \mathscr{S}_0/4$.

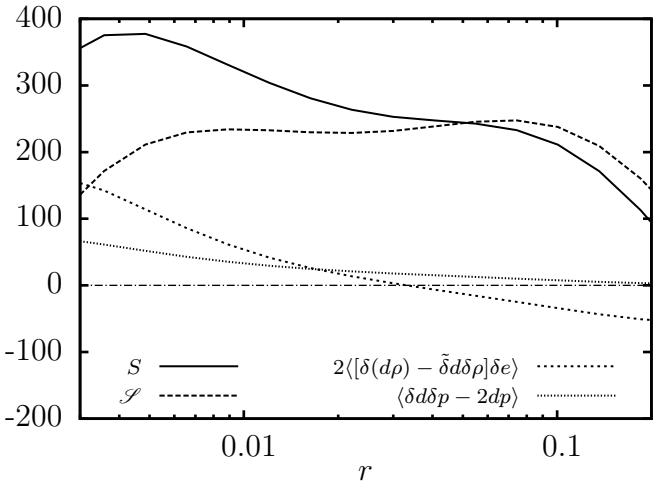

Figure 6.4: Time-average scaling of various source terms in (6.9) and (6.10).

6.4 Discussion

In practical astrophysical applications, where order-of-magnitude estimates are considered sufficient, relation (6.20) can be more convenient than (6.16). We therefore explored the scaling of several proxies to $\mathcal{F}_\parallel$, which are more closely related to observables and to structure functions previously measured numerically. Figure 6.6 illustrates the scaling of $\mathcal{F}_\parallel(r)$ as well as that of the transverse $\langle|\delta v_\perp|^3\rangle$ and longitudinal $\langle|\delta v_\parallel|^3\rangle$ structure functions of the mass-weighted velocity and compares these with theoretical expectations based on (6.16) and (6.20). The inertial range slopes of these proxies are close to linear, so they provide a convenient way of estimating the value of the total flux $F_\parallel$ and thereby the kinetic energy injection rate ε_0.

Figure 6.6 shows that $\mathcal{F}_\parallel(r) \sim 1.7\varepsilon_0$. The total energy flux $F_\parallel$ would have an 18% larger offset, corresponding to a factor of 2. The proxies based on the longitudinal and transverse structure functions of $\boldsymbol{v}$ overestimate ε_0 by factors 3.5 and 4.1, respectively. These values can be used to estimate the energy injection rate required to sustain turbulent cascade in the interstellar medium of the Milky Way (Hennebelle and Falgarone 2012).

An interesting feature to notice in figures 6.5 and 6.6 is the extended ($\sim$ 1 dex)

linear scaling range of $\mathcal{F}_\parallel$, which continues down to $r \simeq 0.008 \simeq 8\Delta$. In ILES carried out with a PPM-based code, all scales below $\sim 16\Delta$ are strongly affected by numerical dissipation (Porter and Woodward 1994). Scales shorter than $\sim 32\Delta$ are usually identified with the so-called 'bottleneck bump' (Falkovich 1994; Porter, Pouquet, and Woodward 1994), where the energy piles up in the near-dissipation part of the inertial range due to a steep wavenumber dependence of numerical diffusivity in the dissipation range ($\propto k^{4-5}$ for PPM, see Porter, Pouquet, and Woodward 1992). Similar bumps are present in the spectra of velocity, density and various mixed quantities in supersonic turbulence (Kritsuk et al. 2007b). In structure functions, the bottleneck is expected to be more pronounced at higher orders (Falkovich 1994); it is also less localized than in power spectra due to mixing of small- and large-scale information (Davidson and Pearson 2005; Dobler et al. 2003, see, e.g., a plot of $\langle|\delta v_\perp|^3\rangle$ in figure 6.6 and note that we take an absolute value). The inertial flux $\mathcal{F}_\parallel$, however, does not show a bump, as expected in the inertial cascade, when an absolute value operation is not applied.

As we discussed above, the bump in the total flux is associated with the compressive energy flux contribution $\langle 2\delta\rho\delta e\delta u_\parallel\rangle$, which becomes comparable to the kinetic energy flux and also somewhat flattens below the sonic scale at $r \lesssim r_\mathrm{s}$: $\delta u_\parallel(r_\mathrm{s}) = c_\mathrm{s}$. This behaviour may depend on the details of the shock-capturing scheme. Indeed, a PPM implementation in the Enzo code (O'Shea et al. 2004) produces a growing fraction of dilatational modes in the velocity power spectrum on scales below 32Δ (see figure 1b in Kritsuk et al. 2010), which could potentially contribute to the bump build-up.

6.5 Conclusions and final remarks

We verified a relation for correlation functions in compressible isothermal turbulence (Galtier and Banerjee 2011) with data from a numerical simulation at Mach 6 (Kritsuk et al.

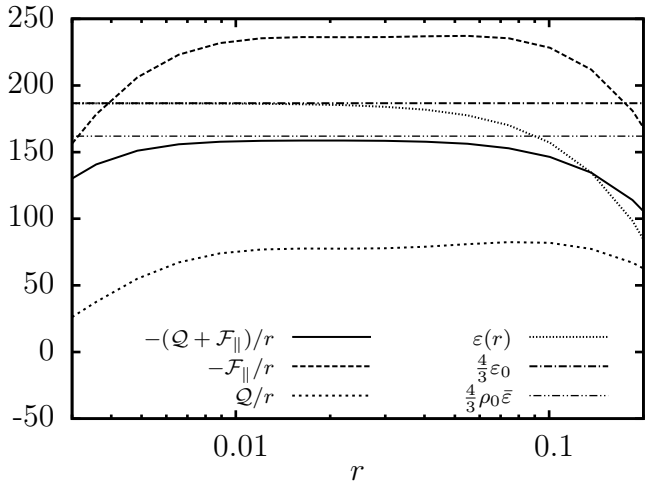

Figure 6.5: As figure 6.2, but for the flux and source terms from an approximate relation (6.20).

2007b). While an isotropic version of the relation is not strictly exact, it provides a good approximation to numerical results. Our analysis of different terms in (6.16) supports a Kolmogorov-like picture of the energy cascade in supersonic turbulence previously discussed on a phenomenological level (Kritsuk et al. 2007b) and recently supported theoretically (Aluie 2013; Aluie 2011; Aluie, Li, and Li 2012). A non-trivial new approximate relation (6.20) that holds at high turbulent Mach numbers is proposed. The relation represents an important step beyond phenomenology, as it sheds light on the problem of universality in compressible turbulence and provides a way to quantitatively predict the energy injection rate from the scaling of certain combinations of observables. This result can have important implications for interstellar turbulence, as approximately constant energy transfer rates are observed in the ISM over more than four decades in length scale (Hennebelle and Falgarone 2012).

The fourth-order scaling relation (6.16) traditionally formulated in terms of the energy flux is not the only compressible analogue of Kolmogorov's four-fifths law. Another approximate relation for homogeneous isothermal turbulence, formulated in terms of fluxes

and densities of conserved quantities,

$$\boldsymbol{\nabla}_{\boldsymbol{r}} \cdot \left\langle \rho\rho' \left[(\boldsymbol{u} \cdot \boldsymbol{u}')\, \boldsymbol{u}' + c_{\mathrm{s}}^2 \boldsymbol{u} \right] \right\rangle \sim - \left\langle \rho\rho' \right\rangle \bar{\varepsilon}, \tag{6.21}$$

cf. (6.6), has recently been obtained and verified with the same numerical data (Wagner et al. 2012). More strictly, this fifth-order flux relation should be viewed as an anisotropic analogue of the von Kármán–Howarth relation, as it involves correlation functions, but (6.21) can also be reduced to the four-fifths law in the incompressible limit (Falkovich, Fouxon, and Oz 2010). Note that the dependence of the density autocorrelation function in the right-hand side of (6.21) on the increment r varies with the Mach number, as does the slope of the density power spectrum (Kim and Ryu 2005). Unlike (6.16), an isotropic version of (6.21) does not have a trivial right-hand side universally linear in r. In a particular case at $M \simeq 6$, the density autocorrelation function has a logarithmic dependence on r and a closed-form analytical representation of the isotropic flux relation is feasible (see (3.3) in Wagner et al. 2012).

We thus conclude that at least two compressible analogues of Kolmogorov's four-fifths law exist, consistent with the extension of the turbulent energy cascade picture to supersonic regimes. Only the fourth-order energy cascade relation (6.16) is 'universal' in the sense that its right-hand side remains approximately linear in the inertial range at all Mach numbers. It is worth noting that the fourth-order relation exploits the conservation of total energy (which is an inviscid invariant), while the fifth-order one follows from conservation of momentum and involves the momentum density and flux.

We thank Hussein Aluie, Gregory Falkovich and Sébastien Galtier for stimulating discussions. This research is supported in part by NSF grants AST-0908740 and AST-1109570. The simulation utilized TeraGrid computer time allocations MCA98N020 and

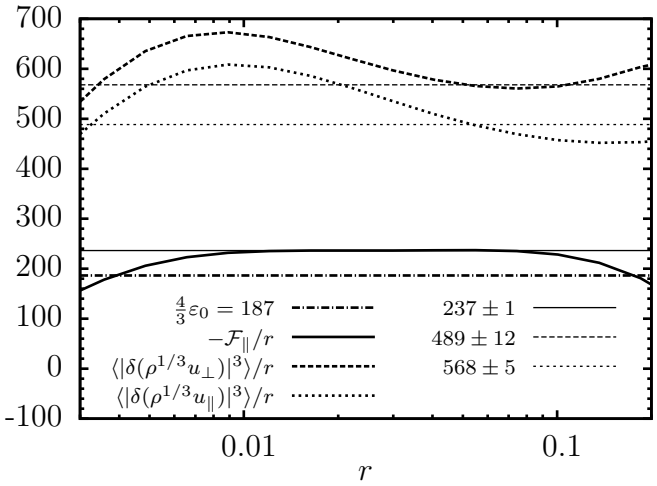

Figure 6.6: Compensated scaling for various proxies of the energy injection rate ε_0. Thin lines show horizontal least-squares fits at $r \in [0.03, 0.1]$; numbers indicate the best-fit value and its standard deviation.

MCA07S014 at SDSC. The analysis was performed on the XSEDE resource Gordon at SDSC under a Director's Discretionary Allocation.

Chapter 6, in full, is a reprint of "Energy Cascade and Scaling in Supersonic Isothermal Turbulence", which was published in 2013 in Journal of Fluid Mechanics, volume 729, R1, by Alexei G. Kritsuk, Rick Wagner, and Michael L. Norman. Reprinted with permission. The dissertation author was the second investigator and author of this paper.

Chapter 7

Conclusions

> The wonderful thing about scaling is
> that you can get everything right
> without understanding anything
>
> R. Kraichnan per L. Kadanoff

7.1 Support for the Local Engergy Cascade in Compressible Turbulence

Appendix C contains a list of the 60 peer-reviewed journal articles and one book–by authors other than myself or my original co-authors–that cite one or more of the results chapters. Several researchers (Ingenito and Bruno 2010; Kowal and Lazarian 2007; Price and Federrath 2010; Schmidt, Federrath, and Klessen 2008; Schmidt and Grete 2019) reproduced the density-weighted results from Chapter 4 and Kritsuk et al. (2007b), to the point that including density-weighted measures of kinetic energy power spectra and structure functions have become the norm in articles regarding simulations of turbulence in the ISM. Other publications (Sekundov and Yakubovskii 2019) were appropriately skeptical

of the initial density-weighted $\delta\rho^{1/3}u$ structure functions. Section 4.5.2 of Girichidis et al. (2020) describes this timeline:

> This was a clear indication that the energy transfer in supersonic regime is mediated by density fluctuations and hence ignoring the density velocity correlations would lead to large errors in the kinetic energy flux across scales (Fleck 1996; Henriksen 1991; Lighthill 1955; von Weizsäcker 1951). A very naïve check to make was to consider the third-order structure function of a density-weighted velocity, $\boldsymbol{v} \equiv \rho^{1/3}\boldsymbol{u}$, which happened to be linear (Kritsuk et al. 2007b). The linear scaling at high Mach numbers was then independently confirmed in (Kowal and Lazarian 2007; Schmidt, Federrath, and Klessen 2008; Schwarz et al. 2010b; Zrake and MacFadyen 2012). These results were consistent with simple Richardson-Kolmogorov-like energy cascade phenomenology modified to account for density fluctuations, $\Pi(r) \sim \rho_r u_r^3/r \sim \rho_0\epsilon$, and hence triggered a quest for exact relations describing the energy cascade in compressible turbulence.

This quest is what led to the relations proposed in Falkovich, Fouxon, and Oz (2010) and Galtier and Banerjee (2011), and the results in Chapters 5 and 6, respectively. The Galtier and Banerjee (2011) relation was reproduced in Wang et al. (2018, 2013) for subsonic and transonic turbulence simulations.

Sections 4.5.4 through 4.5.6 of Girichidis et al. (2020) covers numerous other relations that have been put forth as analogues to Kolmogorov's 1941 theory. For example, Equation 33 in Banerjee and Kritsuk (2017)

$$\begin{aligned} -4\varepsilon =& \boldsymbol{\nabla}_{\boldsymbol{r}} \cdot \langle (\delta \boldsymbol{j} \cdot \delta \boldsymbol{u} + \delta\rho\delta e)\, \delta \boldsymbol{u} \rangle \\ &+ \langle \delta\left(\boldsymbol{j} \cdot \boldsymbol{u} + \rho e\right) \delta\theta + \delta \boldsymbol{j} \cdot \delta\left(\nabla e - \boldsymbol{u}\theta\right) - \delta\rho\delta\left(e\theta\right) \rangle \\ &+ \langle \delta \boldsymbol{j} \cdot \delta \boldsymbol{g} - \delta \boldsymbol{u} \cdot \delta(\rho \boldsymbol{g}) \rangle \end{aligned} \tag{7.1}$$

where $\theta \equiv \nabla \cdot \boldsymbol{u}$, $\boldsymbol{j} \equiv \rho\boldsymbol{u}$, and $\boldsymbol{g}$ is the gravitational force. Equation 7.1 refines the relation in Galtier and Banerjee (2011) (see Equation 6.6 in that article). Ferrand et al. (2020) explored the scaling of relations based on Banerjee and Kritsuk (2017) and Galtier and Banerjee (2011) finding good agreement.

Other relations have been evaluated, such as the velocity structure functioned weighted by the density correlation used in Padoan et al. (2016)

$$S_p(r) \equiv \frac{\langle \rho(\boldsymbol{x}_1)\rho(\boldsymbol{x}_2)|\boldsymbol{u}(\boldsymbol{x}_2) - \boldsymbol{u}(\boldsymbol{x}_1)|^p \rangle}{\langle \rho(\boldsymbol{x}_1)\rho(\boldsymbol{x}_2) \rangle}, \tag{7.2}$$

which is based on the relation from Falkovich, Fouxon, and Oz (2010). Equation 7.2 was also used Chira et al. (2019), where they found the expect scaling broken down in the presence of shocks and small-scale structure. Padoan et al. (2016) and Chira et al. (2019) are also applying the ESS (extended self-similarity) model from Benzi et al. (1993), which normalizes structure functions by the third order one to study their scaling outside of the inertial range. My impression is that this may be an inappropriate application of the ESS model, since it attempts to define the inertial range based on the scaling of a third order structure function, which may not apply to supersonic turbulence. Instead, models such as Equation 7.1 define the inertial range as those scales undergoing a self-similar cascade of energy.

7.2 Observational Support

Direct measurement of turbulence statistics in the ISM is challenging since the multi-dimensional space of position, velocity, density, etc., is collapsed into two or three-dimensional static arrays (Boyden et al. 2018; Burkhart 2021; Girichidis et al. 2020; Koch et al. 2019). These difficulties are summarized in Haworth et al. (2018):

> *Observational limitations/complexity.* Since the optical depth of the ISM is highly frequency dependent and different lines probe different ranges of density and temperature, observations of only one or a few tracers do not provide us with all of the information available within a theoretical model. Furthermore, real observations are subject to other processes such as noise, resolution constraints and interferometric effects that may be significantly different to the limitations of a numerical model. The ISM is also geometrically complex and evolves on

timescales beyond the human experience. Observers are therefore limited to a restricted view of a single snapshot in time.

Despite these challenges, progress has been made, notably in the measure of compressive vs. solenoidal driving modes in Orion B (Orkisz et al. 2017), which agrees with the predictions from simulations (Kritsuk et al. 2010). Researchers are also consolidating work on statistical toolkits for measuring turbulence from observations (Koch et al. 2019) and decomposing spectroscopic observations into separate velocity and density components (Yuen, Ho, and Lazarian 2021).

7.3 Summary

I have described my work developing and using a fast method for measuring the statistics of two-point functions in simulations of supersonic isothermal turbulence. As described in Chapters 4, 5, and 6, the results support the locality of energy transfer within the inertial range. Several publications from other researchers have reported similar results from simulations using different codes. Many practical limitations hamper observational evidence, but experimental support is emerging.

Appendix A

Syntax Reference

Describing computational methods can be frustrating. One either feels they are doing a disservice to the mathematical notation or the code. The code is exact in what it does but lacks the elegance of the mathematical notation. A vector stored as the variable `x` in code could also be an integer, a real number, or even the letter "x". Being able show a vector as x_i, $\boldsymbol{x}$, or $\vec{x}$ communicates what "x" is in context. Mathematical notation does well when describing the scientific goals of the methods in this dissertation. I can succinctly present the individual two-point functions, as in Equation 1.1. However, mathematical notation becomes convoluted when describing functions, especially complex data structures that may have both attributes and methods. Since the goal of Chapters 2 and 3 is to describe computational methods, the following notes summarize the syntax conventions I have used in this dissertation. Also, the list of nomenclature includes the variables defined for many data structures.

- CamelCase is used to name complex things (functions, data structures, instances of classes).
- Functions, ClassMethods, and Algorithms are indicated by SansSerifFont.
- Instances of `DataStructures` (arrays, instance attributes, etc.) are named using a `fixed-width`, or `typewriter`, font.
- Array indices may either be in subscripts or brackets. Subscript was used mostly for physcal variables, e.g., x_i for a point in 3D space and brackets for data structures, like `SomeArray`$[k]$. The choice is largely to maintain clarity when reading equations or pseudocode. It is also relevant because many of the multidimensional arrays are allocated in memory as one-dimensional arrays for efficiency.
- Arrays are index beginning at 0, mostly because the code was written in C and C++, and it makes computing array offsets a bit clearer.
- Instance attributes or methods may be referenced using dot notation, e.g., B.`SomeArray`$[i]$ or C.SomeMethod(x, y).

- With some exceptions, assignment in pseudocode is shown using the '=' operator, not $\rightarrow$.
- As a result, equality in pseudocode is shown via '=='.
- Parameters passed to functions are essentially global. That is, changes to them persist when the function exits.
- Some Enzo code elements (attributes, methods, etc.) are renamed for clarity (e.g., *task* vs. *processor*).
- Likewise, class attributes may be referenced directly in pseudocode rather than through get or set methods.
- Some function parameters may be replaced by ..., such as the repetitive MPI datatype and communicator arguments.

Appendix B

A Brief Introduction to MPI

The analysis framework I developed that underlies SFgen is integrated with Enzo and therefore follows the same parallelization strategy. This allows for analysis to be done in-line during a simulation. It also enables the creation of stand-alone parallel applications that perform rapid analysis of Enzo datasets outside of simulation runs. In the out-of-band case, the level of parallelism can be reduced. This may seem like an odd choice, but by requesting fewer resources the analysis is more likely to be scheduled on HPC (high-performance computing) systems.

The basic design pattern for MPI applications is to assume that each MPI task (typically a distinct POSIX process, referred as the rank of the MPI task) is running in its own memory space, and cannot access the memory of other tasks. Each task guides its processing based on its rank. This is known as the SIMD (single instruction, multiple data) model (Flynn 1972), where multiple instances of an application (Enzo and SFgen, in this case) are started, but each process operates on different data based on its rank to parallelize the workload.

MPI can support other parallelization models, including MIMD (multiple instruction, multiple data), where different applications are integrated into the MPI communications domain. This can be complex to implement and is unsurprisingly less common. There are also hybrid mechanism that balance number of tasks within a single memory domain (e.g., a compute node or a GPU) with the amount of memory available in that domain. Those mechanisms will parallelize some operations using a shared-memory model, like SMP (symmetric multiprocessing) or NVIDIA's CUDA (Compute Unified Device Architecture®), and use MPI when necessary, like when coordinating with tasks on other compute nodes.

The MPI Forum defines the MPI API standard (Message Passing Interface Forum 2021), which is then implemented by different libraries (Gabriel et al. 2004; Gropp et al. 1996; Panda et al. 2021), and specific implementation may be better adapted to different hardware architectures, particularly the network layer. The basic MPI pattern is for tasks to exchange information through function explicit calls.

Code Example B.1 shows the function prototypes for the basic send and receive operations MPI_Send and MPI_Recv. The following description of the functions is from the

```
MPI_Send(
    void* data,
    int count,
    MPI_Datatype datatype,
    int destination,
    int tag,
    MPI_Comm communicator)

MPI_Recv(
    void* data,
    int count,
    MPI_Datatype datatype,
    int source,
    int tag,
    MPI_Comm communicator,
    MPI_Status* status)
```

Code Example B.1: MPI send and receive function prototypes.

MPI Tutorials site[1]:

> The first argument is the data buffer. The second and third arguments describe the count and type of elements that reside in the buffer. MPI_Send sends the exact count of elements, and MPI_Recv will receive at most the count of elements (more on this in the next lesson). The fourth and fifth arguments specify the rank of the sending/receiving process and the tag of the message. The sixth argument specifies the communicator and the last argument (for MPI_Recv only) provides information about the received message.

Code Example B.2 shows a code snippet with one task sending the value of an integer to a second task, adapted from the same site[2].

[1] https://mpitutorial.com/about/

[2] https://mpitutorial.com/tutorials/mpi-send-and-receive/

```
/* Find out rank, size */
int my_rank;
MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

int number;
if (my_rank == 0) {
    number = -1;
    MPI_Send(&number, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
} else if (my_rank == 1) {
    MPI_Recv(&number, 1, MPI_INT, 0, 0, MPI_COMM_WORLD,
             MPI_STATUS_IGNORE);
    printf(``Process 1 received number %d from process 0\n",
           number);
}
```

Code Example B.2: Minimal example of using MPI send and receive functions.

B.1 Key Concepts

There are two MPI concepts that are important to understand in the context of applications like SFgen : blocking and non-blocking communication; and collective operations.

Blocking and Non-Blocking Communications MPI functions are either blocking, where the calling task does not continue until the function returns, or non-blocking, where the task may continue and then check the status of the function call later. (The examples above, and all of the functions used in SFgen are blocking.) Using non-blocking calls is more complex, but may allow tasks to perform other operations while communication is handle by the network libraries and hardware. Note that tasks *may* not be blocked by the function call. The MPI implementation may not actually implement a non-blocking version of the function, and instead simply wrap the blocking function.

Collective Operations Collective operations are useful functions to address common use cases, such as

- broadcasting a value from one tasks other,
- distributing portions of an array across other tasks (scattering),
- gathering portions of an array from other tasks,
- of finding the minimum of or summing the elements of an array distributed across several tasks (reducing).

These collectives are defined as one-to-many, many-to-one, and many-to-many operations. In the one-to-many case, a single task is the source or destination, while for many-to-many each task is the source and destination for different portions of the data.

As shown in Section 3.6, SFgen relies heavily on collective operations, specifically all-to-all and what's referred to as *personalized* exchange, or all-to-allv. All-to-all sends a fixed number of elements from an array between each tasks, so that the send and receive arrays are the same size, while all-to-allv distributes an arbitrary number of elements from each tasks send arrays to the other tasks. Each collective operation has its own costs depending on the network topology, such as a ring, mesh, or hypercube, and this will vary based the amount of data (message size) being echanged between tasks. An advantage of using these collective operations is the chance to leverage the expertise that was part of developing the MPI implementation on a particular system (Faraj and Yuan 2005; Pješivac-Grbović et al. 2007; Sur, Jin, and Panda 2004). The disadvantage is that SFgen "has been known to break supercomputers due to its heavy communication load", as I told my colleague, Stephen Skory, while he was developing the structure function code for `yt` (Skory 2010). My recollection is that the specific reason for the creation of the MakePass loop (see Section 3.3.5) was to reduce the size of the message buffers used in the calls to MPI all-to-allv operations.

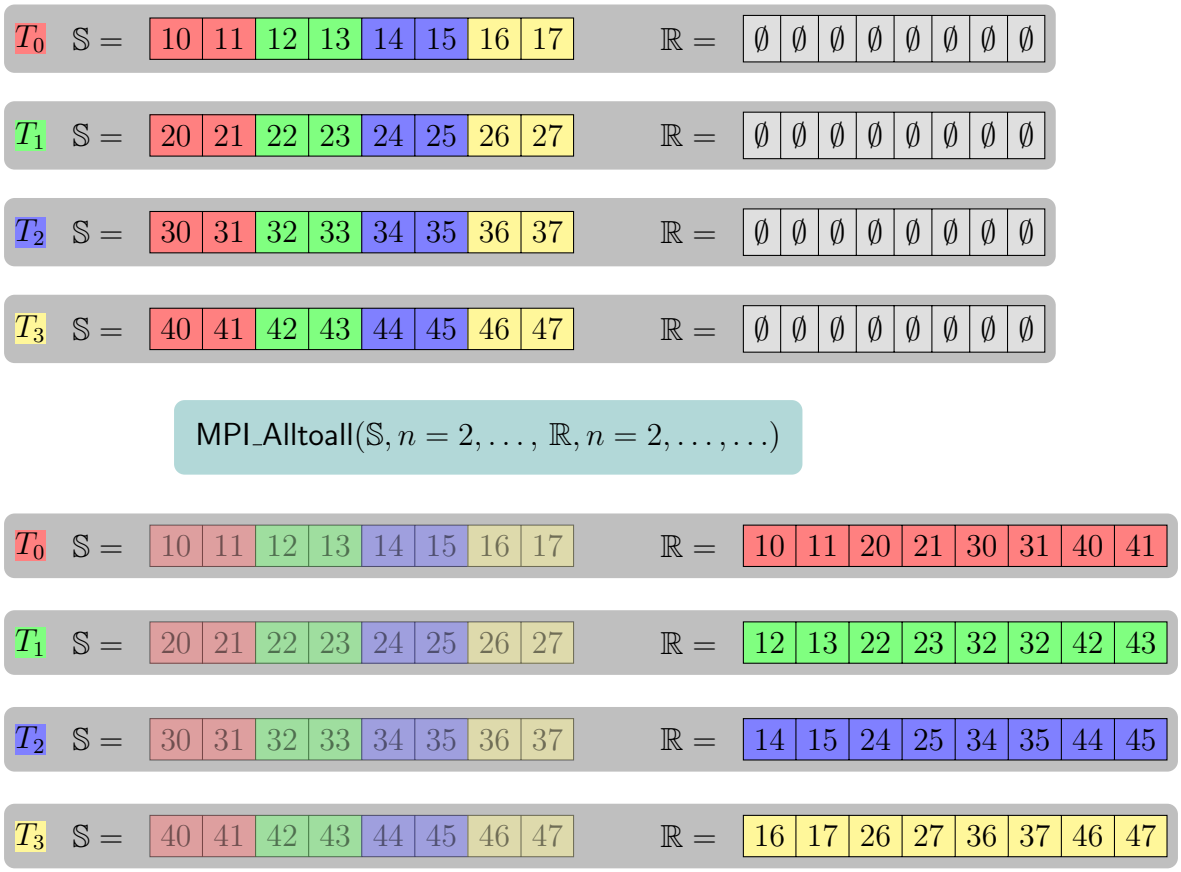

Figure B.1: Example of an MPI all-to-all collective operation, where four tasks distribute two integers to each other task. $\mathbb{S}$ and $\mathbb{R}$ are the send and receive buffers, respectively. In the call to MPI_Alltoall, the parameter $n = 2$ is the count of elements to be sent to each task. $\mathbb{S}$ and $\mathbb{R}$ must be of size $2 \times N_T$

B.2 MPI_Alltoall & MPI_Alltoallv

Figures B.1 and B.2 are diagrams of calls to all-to-all and all-to-allv, respectively. TODO: add function prototypes. Tie in motivation for grid index and bookkeeping.

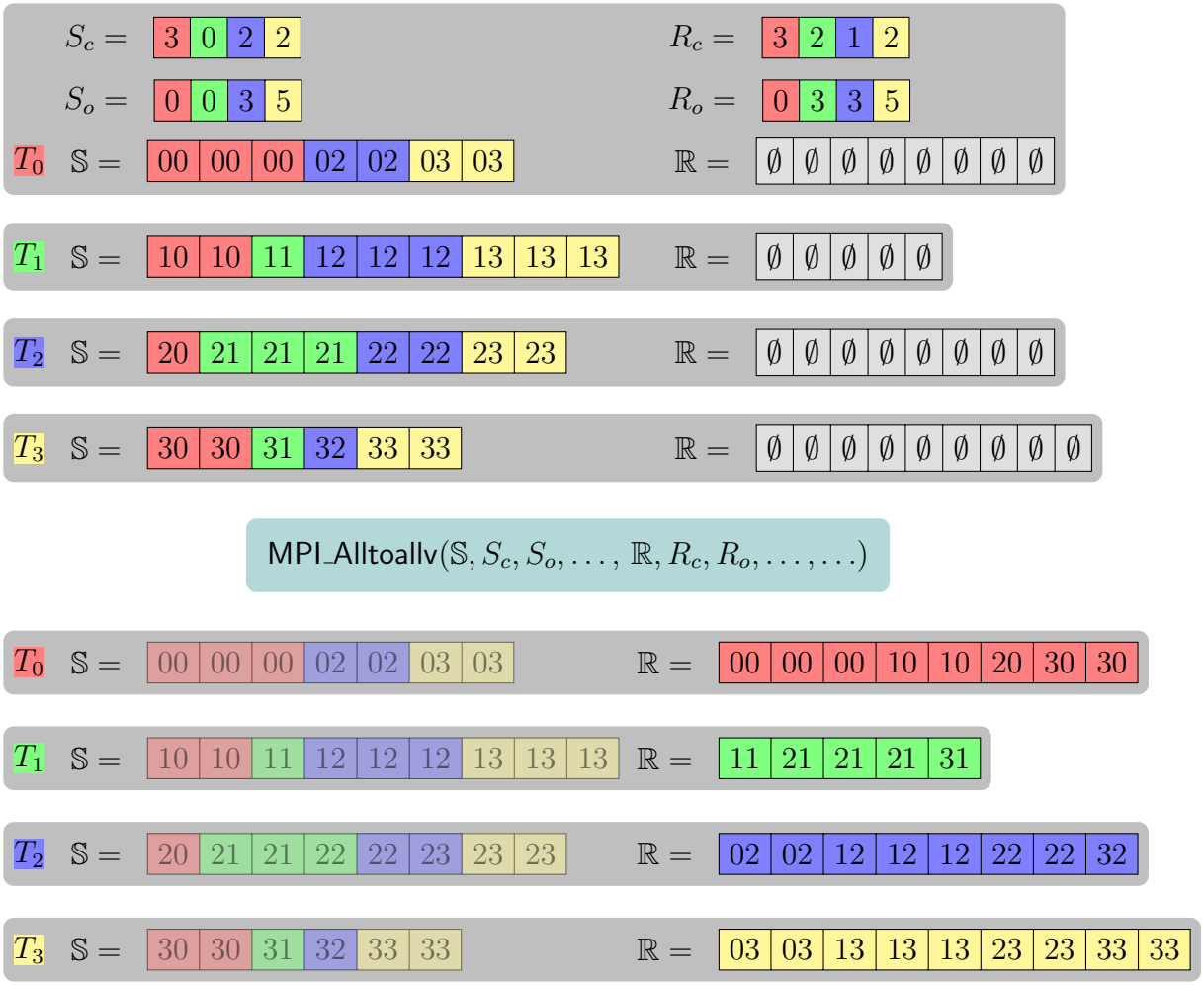

Figure B.2: Example of an MPI all-to-allv collective operation, where four tasks distribute an arbitrary count of integers to each other task. $\mathbb{S}$ and $\mathbb{R}$ are the send and receive buffers, respectively. The send counts and offsets, S_c and S_o, and receive counts and offsets, R_c and R_o, for task T_0 are shown at the top. The counts and offsets are integer arrays of size N_T and are used as part of the call to MPI_Alltoallv. The size of $\mathbb{S}$ on each task must equal the sum of S_c and the size of $\mathbb{R}$ must equal the sum of R_c.

Appendix C

Citations of Results Chapters

Aluie, Hussein (Mar. 2013). "Scale decomposition in compressible turbulence." In: *Physica D Nonlinear Phenomena* 247.1, pp. 54–65. DOI: 10.1016/j.physd.2012.12.009. arXiv: 1012.5877 [physics.flu-dyn].

Andrés, N. and F. Sahraoui (Nov. 2017). "Alternative derivation of exact law for compressible and isothermal magnetohydrodynamics turbulence." In: Phys. Rev. E 96.5, 053205, p. 053205. DOI: 10.1103/PhysRevE.96.053205. arXiv: 1707.00749 [physics.plasm-ph].

Andrés, N., F. Sahraoui, S. Galtier, L. Z. Hadid, P. Dmitruk, and P. D. Mininni (Aug. 2018). "Energy cascade rate in isothermal compressible magnetohydrodynamic turbulence." In: *Journal of Plasma Physics* 84.4, 905840404, p. 905840404. DOI: 10.1017/S0022377818000788. arXiv: 1802.05503 [physics.plasm-ph].

Andrés, N., F. Sahraoui, S. Galtier, L. Z. Hadid, R. Ferrand, and S. Y. Huang (Dec. 2019). "Energy Cascade Rate Measured in a Collisionless Space Plasma with MMS Data and Compressible Hall Magnetohydrodynamic Turbulence Theory." In: Phys. Rev. Lett. 123.24, 245101, p. 245101. DOI: 10.1103/PhysRevLett.123.245101. arXiv: 1911.09749 [physics.plasm-ph].

Andrés, N., F. Sahraoui, L. Z. Hadid, S. Y. Huang, N. Romanelli, S. Galtier, G. DiBraccio, and J. Halekas (Sept. 2021). "The Evolution of Compressible Solar Wind Turbulence in the Inner Heliosphere: PSP, THEMIS, and MAVEN Observations." In: ApJ 919.1, 19, p. 19. DOI: 10.3847/1538-4357/ac0af5. arXiv: 2102.11781 [physics.space-ph].

Antonuccio-Delogu, V. and J. Silk (Oct. 2008). "Active galactic nuclei jet-induced feedback in galaxies - I. Suppression of star formation." In: MNRAS 389.4, pp. 1750–1762. DOI: 10.1111/j.1365-2966.2008.13663.x. arXiv: 0806.4570 [astro-ph].

Banerjee, S., L. Z. Hadid, F. Sahraoui, and S. Galtier (Oct. 2016). "Scaling of Compressible Magnetohydrodynamic Turbulence in the Fast Solar Wind." In: ApJ 829.2, L27, p. L27. DOI: 10.3847/2041-8205/829/2/L27. arXiv: 1609.00598 [physics.space-ph].

Banerjee, Supratik and Nahuel Andrés (Apr. 2020). "Scale-to-scale energy transfer rate in compressible two-fluid plasma turbulence." In: Phys. Rev. E 101.4, 043212, p. 043212. DOI: 10.1103/PhysRevE.101.043212. arXiv: 2004.14447 [physics.plasm-ph].

Banerjee, Supratik and Sébastien Galtier (Mar. 2014). "A Kolmogorov-like exact relation for compressible polytropic turbulence." In: *Journal of Fluid Mechanics* 742, pp. 230–242. DOI: 10.1017/jfm.2013.657. arXiv: 1308.4051 [physics.flu-dyn].

Beresnyak, Andrey (Sept. 2019). "MHD turbulence." In: *Living Reviews in Computational Astrophysics* 5.1, 2, p. 2. DOI: 10.1007/s41115-019-0005-8. arXiv: 1910.03585 [astro-ph.GA].

Chefranov, S. G. and A. S. Chefranov (May 2019). "Exact solution of the compressible Euler-Helmholtz equation and the millennium prize problem generalization." In: Phys. Scr 94.5, 054001, p. 054001. DOI: 10.1088/1402-4896/aaf918. arXiv: 1703.07239 [physics.gen-ph].

Chira, R. -A., J. C. Ibáñez-Mejía, M. -M. Mac Low, and Th. Henning (Oct. 2019). "How do velocity structure functions trace gas dynamics in simulated molecular clouds?" In: A&A 630, A97, A97. DOI: 10.1051/0004-6361/201833970. arXiv: 1908.03951 [astro-ph.GA].

Colbrook, Matthew J., Xiangcheng Ma, Philip F. Hopkins, and Jonathan Squire (May 2017). "Scaling laws of passive-scalar diffusion in the interstellar medium." In: MNRAS 467.2, pp. 2421–2429. DOI: 10.1093/mnras/stx261. arXiv: 1610.06590 [astro-ph.GA].

Davidovits, Seth and Nathaniel J. Fisch (Apr. 2017). "A Lower Bound on Adiabatic Heating of Compressed Turbulence for Simulation and Model Validation." In: ApJ 838.2, 118, p. 118. DOI: 10.3847/1538-4357/aa619f. arXiv: 1612.00063 [astro-ph.IM].

Dong, Yufeng and Guowei He (June 2017). "Scaling of energy spectra in weakly compressible turbulence." In: *Acta Mechanica Sinica* 33.3, pp. 500–507. DOI: 10.1007/s10409-017-0663-x.

Eyink, Gregory L. and Theodore D. Drivas (Feb. 2018). "Cascades and Dissipative Anomalies in Compressible Fluid Turbulence." In: *Physical Review X* 8.1, 011022, p. 011022. DOI: 10.1103/PhysRevX.8.011022. arXiv: 1704.03532 [physics.flu-dyn].

Federrath, Christoph (Dec. 2013). "On the universality of supersonic turbulence." In: MNRAS 436.2, pp. 1245–1257. DOI: 10.1093/mnras/stt1644. arXiv: 1306.3989 [astro-ph.SR].

Ferrand, R., S. Galtier, F. Sahraoui, and C. Federrath (Dec. 2020). "Compressible Turbulence in the Interstellar Medium: New Insights from a High-resolution Supersonic Turbulence Simulation." In: ApJ 904.2, 160, p. 160. DOI: 10.3847/1538-4357/abb76e.

Ferrand, R., F. Sahraoui, S. Galtier, N. Andrés, P. Mininni, and P. Dmitruk (Mar. 2022). "An In-depth Numerical Study of Exact Laws for Compressible Hall Magnetohydrodynamic Turbulence." In: ApJ 927.2, 205, p. 205. DOI: 10.3847/1538-4357/ac517a. arXiv: 2201.10819 [physics.plasm-ph].

Folini, Doris and Rolf Walder (Mar. 2016). "Accuracy requirements to test the applicability of the random cascade model to supersonic turbulence." In: A&A 587, A120, A120. DOI: 10.1051/0004-6361/201527148. arXiv: 1602.02121 [astro-ph.CO].

Folini, Doris, Rolf Walder, and Jean M. Favre (Feb. 2014). "Supersonic turbulence in 3D isothermal flow collision." In: A&A 562, A112, A112. DOI: 10.1051/0004-6361/201322482. arXiv: 1401.0816 [astro-ph.SR].

Fouxon, Itzhak and Michael Mond (Aug. 2019). "Density and tracer statistics in compressible turbulence: Phase transition to multifractality." In: Phys. Rev. E 100.2, 023111, p. 023111. DOI: 10.1103/PhysRevE.100.023111. arXiv: 1807.01167 [physics.flu-dyn].

Grisdale, Kearn, Oscar Agertz, Alessandro B. Romeo, Florent Renaud, and Justin I. Read (Apr. 2017). "The impact of stellar feedback on the density and velocity structure of the interstellar medium." In: MNRAS 466.1, pp. 1093–1110. DOI: 10.1093/mnras/stw3133. arXiv: 1608.08639 [astro-ph.GA].

Hadid, L. Z., F. Sahraoui, and S. Galtier (Mar. 2017). "Energy Cascade Rate in Compressible Fast and Slow Solar Wind Turbulence." In: ApJ 838.1, 9, p. 9. DOI: 10.3847/1538-4357/aa603f. arXiv: 1612.02150 [astro-ph.SR].

Hellinger, Petr, Andrea Verdini, Simone Landi, Emanuele Papini, Luca Franci, and Lorenzo Matteini (Apr. 2021). "Scale dependence and cross-scale transfer of kinetic energy in compressible hydrodynamic turbulence at moderate Reynolds numbers." In: *Physical Review Fluids* 6.4, 044607, p. 044607. DOI: 10.1103/PhysRevFluids.6.044607. arXiv: 2103.12005 [physics.flu-dyn].

Ingenito, A., D. Cecere, and E. Giacomazzi (Sept. 2013). "Large Eddy simulation of turbulent hydrogen-fuelled supersonic combustion in an air cross-flow." In: *Shock Waves* 23.5, pp. 481–494. DOI: 10.1007/s00193-013-0454-7.

Ingenito, Antonella and Claudio Bruno (Mar. 2010). "Physics and Regimes of Supersonic Combustion." In: *AIAA Journal* 48.3, pp. 515–525. DOI: 10.2514/1.43652.

Krumholz, Mark R. (June 2014). "The big problems in star formation: The star formation rate, stellar clustering, and the initial mass function." In: Phys. Rep. 539, pp. 49–134. DOI: 10.1016/j.physrep.2014.02.001. arXiv: 1402.0867 [astro-ph.GA].

Lees, Aarne and Hussein Aluie (May 2019). "Baropycnal Work: A Mechanism for Energy Transfer across Scales." In: *Fluidika* 4.2, p. 92. DOI: 10.3390/fluids4020092. arXiv: 1905.03581 [physics.flu-dyn].

Magnani, Loris and Steven N. Shore (2017). *A Dirty Window*. Vol. 442. Springer Berlin, Heidelberg. DOI: 10.1007/978-3-662-54350-4.

Mandal, Ankush, Christoph Federrath, and Bastian Körtgen (Apr. 2020). "Molecular cloud formation by compression of magnetized turbulent gas subjected to radiative cooling." In: MNRAS 493.3, pp. 3098–3113. DOI: 10.1093/mnras/staa468. arXiv: 1910.13762 [astro-ph.GA].

Marchuk, Alexander A., Anton A. Smirnov, Aleksandr V. Mosenkov, Vladimir B. Il'in, George A. Gontcharov, Sergey S. Savchenko, and Javier Román (Dec. 2021). "Fractal dimension of optical cirrus in Stripe82." In: MNRAS 508.4, pp. 5825–5841. DOI: 10.1093/mnras/stab2846. arXiv: 2109.14034 [astro-ph.GA].

Mouraya, Sukhdev and Supratik Banerjee (Nov. 2019). "Determination of energy flux rate in homogeneous ferrohydrodynamic turbulence using two-point statistics." In: Phys. Rev. E 100.5, 053105, p. 053105. DOI: 10.1103/PhysRevE.100.053105. arXiv: 1907.00604 [physics.flu-dyn].

Pan, Liubin and Evan Scannapieco (Oct. 2010). "Mixing in Supersonic Turbulence." In: ApJ 721.2, pp. 1765–1782. DOI: 10.1088/0004-637X/721/2/1765. arXiv: 1008.0665 [astro-ph.GA].

Price, Daniel J. and Christoph Federrath (Aug. 2010). "A comparison between grid and particle methods on the statistics of driven, supersonic, isothermal turbulence." In: MNRAS 406.3, pp. 1659–1674. DOI: 10.1111/j.1365-2966.2010.16810.x. arXiv: 1004.1446 [astro-ph.GA].

Riols, A., H. Latter, and S. -J. Paardekooper (Oct. 2017). "Gravitoturbulence and the excitation of small-scale parametric instability in astrophysical discs." In: MNRAS 471.1, pp. 317–336. DOI: 10.1093/mnras/stx1548. arXiv: 1706.06537 [astro-ph.EP].

Sahraoui, Fouad, Lina Hadid, and Shiyong Huang (Feb. 2020). "Magnetohydrodynamic and kinetic scale turbulence in the near-Earth space plasmas: a (short) biased review." In: *Reviews of Modern Plasma Physics* 4.1, 4, p. 4. DOI: 10.1007/s41614-020-0040-2.

Saury, E., M. -A. Miville-Deschênes, P. Hennebelle, E. Audit, and W. Schmidt (July 2014). "The structure of the thermally bistable and turbulent atomic gas in the local interstellar medium." In: A&A 567, A16, A16. DOI: 10.1051/0004-6361/201321113. arXiv: 1301.3446 [astro-ph.GA].

Schmidt, W., C. Byrohl, J. F. Engels, C. Behrens, and J. C. Niemeyer (Sept. 2017). "Viscosity, pressure and support of the gas in simulations of merging cool-core clusters." In: MNRAS 470.1, pp. 142–156. DOI: 10.1093/mnras/stx1274. arXiv: 1705.06933 [astro-ph.CO].

Schmidt, W., F. Ciaraldi-Schoolmann, J. C. Niemeyer, F. K. Röpke, and W. Hillebrandt (Feb. 2010). "Turbulence in a Three-Dimensional Deflagration Model For Type Ia Supernovae. II. Intermittency and the Deflagration-to-Detonation Transition

Probability." In: ApJ 710.2, pp. 1683–1693. DOI: 10.1088/0004-637X/710/2/1683. arXiv: 0911.4345 [astro-ph.HE].

Schmidt, W., C. Federrath, M. Hupp, S. Kern, and J. C. Niemeyer (Jan. 2009). "Numerical simulations of compressively driven interstellar turbulence. I. Isothermal gas." In: A&A 494.1, pp. 127–145. DOI: 10.1051/0004-6361:200809967. arXiv: 0809.1321 [astro-ph].

Schmidt, Wolfram (Oct. 2015). "Large Eddy Simulations in Astrophysics." In: *Living Reviews in Computational Astrophysics* 1.1, 2, p. 2. DOI: 10.1007/lrca-2015-2. arXiv: 1404.2483 [astro-ph.CO].

Schmidt, Wolfram, Christoph Federrath, and Ralf Klessen (Nov. 2008). "Is the Scaling of Supersonic Turbulence Universal?" In: Phys. Rev. Lett. 101.19, 194505, p. 194505. DOI: 10.1103/PhysRevLett.101.194505. arXiv: 0810.1397 [astro-ph].

Schmidt, Wolfram and Philipp Grete (Oct. 2019). "Kinetic and internal energy transfer in implicit large-eddy simulations of forced compressible turbulence." In: Phys. Rev. E 100.4, 043116, p. 043116. DOI: 10.1103/PhysRevE.100.043116. arXiv: 1906.12228 [physics.flu-dyn].

Sekundov, A. N. and K. Ya. Yakubovskii (Mar. 2019). "Analysis of Kolmogorov's Hypotheses in a Compressible Turbulent Flow." In: *Fluid Dynamics* 54.2, pp. 184–193. DOI: 10.1134/S0015462819020113.

Teissier, Jean-Mathieu and Wolf-Christian Müller (Aug. 2021). "Inverse transfer of magnetic helicity in direct numerical simulations of compressible isothermal turbulence: helical transfers." In: *Journal of Fluid Mechanics* 921, A7, A7. DOI: 10.1017/jfm.2021.496. arXiv: 2012.10855 [physics.plasm-ph].

Wang, Jianchun, Toshiyuki Gotoh, and Takeshi Watanabe (May 2017a). "Scaling and intermittency in compressible isotropic turbulence." In: *Physical Review Fluids* 2.5, 053401, p. 053401. DOI: 10.1103/PhysRevFluids.2.053401.

— (Jan. 2017b). "Spectra and statistics in compressible isotropic turbulence." In: *Physical Review Fluids* 2.1, 013403, p. 013403. DOI: 10.1103/PhysRevFluids.2.013403.

Wang, Jianchun, Minping Wan, Song Chen, and Shiyi Chen (Apr. 2018a). "Kinetic energy transfer in compressible isotropic turbulence." In: *Journal of Fluid Mechanics* 841, pp. 581–613. DOI: 10.1017/jfm.2018.23.

Wang, Jianchun, Minping Wan, Song Chen, Chenyue Xie, and Shiyi Chen (Apr. 2018b). "Effect of shock waves on the statistics and scaling in compressible isotropic turbulence." In: Phys. Rev. E 97.4, 043108, p. 043108. DOI: 10.1103/PhysRevE.97.043108.

Wang, Jianchun, Minping Wan, Song Chen, Chenyue Xie, Lian-Ping Wang, and Shiyi Chen (May 2019). "Cascades of temperature and entropy fluctuations in compressible

turbulence." In: *Journal of Fluid Mechanics* 867, pp. 195–215. DOI: 10.1017/jfm.2019.116.

Wang, Jianchun, Yantao Yang, Yipeng Shi, Zuoli Xiao, X. T. He, and Shiyi Chen (May 2013a). "Cascade of Kinetic Energy in Three-Dimensional Compressible Turbulence." In: Phys. Rev. Lett. 110.21, 214505, p. 214505. DOI: 10.1103/PhysRevLett.110.214505.

— (June 2013b). "Statistics and structures of pressure and density in compressible isotropic turbulence." In: *Journal of Turbulence* 14.6, pp. 21–37. DOI: 10.1080/14685248.2013.831989.

Xie, Chenyue, Jianchun Wang, Hui Li, Minping Wan, and Shiyi Chen (June 2018). "A modified optimal LES model for highly compressible isotropic turbulence." In: *Physics of Fluids* 30.6, 065108, p. 065108. DOI: 10.1063/1.5027754.

— (Aug. 2019). "Artificial neural network mixed model for large eddy simulation of compressible isotropic turbulence." In: *Physics of Fluids* 31.8, 085112, p. 085112. DOI: 10.1063/1.5110788.

Yang, Yan, William H. Matthaeus, Yipeng Shi, Minping Wan, and Shiyi Chen (Mar. 2017). "Compressibility effect on coherent structures, energy transfer, and scaling in magnetohydrodynamic turbulence." In: *Physics of Fluids* 29.3, 035105, p. 035105. DOI: 10.1063/1.4979068.

Yang, Yan, Minping Wan, William H. Matthaeus, and Shiyi Chen (June 2021). "Energy budget in decaying compressible MHD turbulence." In: *Journal of Fluid Mechanics* 916, A4, A4. DOI: 10.1017/jfm.2021.199.

Yang, Yantao, Jianchun Wang, Yipeng Shi, Zuoli Xiao, X. T. He, and Shiyi Chen (Jan. 2016). "Intermittency caused by compressibility: a Lagrangian study." In: *Journal of Fluid Mechanics* 786, R6, R6. DOI: 10.1017/jfm.2015.681.

Zhao, Dongxiao, Riccardo Betti, and Hussein Aluie (Jan. 2022). "Scale interactions and anisotropy in Rayleigh–Taylor turbulence." In: *Journal of Fluid Mechanics* 930, A29, A29. DOI: 10.1017/jfm.2021.902. arXiv: 2006.04301 [physics.flu-dyn].

Zwick, Lorenz, Andrea Derdzinski, Mudit Garg, Pedro R. Capelo, and Lucio Mayer (Apr. 2022). "Dirty waveforms: multiband harmonic content of gas-embedded gravitational wave sources." In: MNRAS 511.4, pp. 6143–6159. DOI: 10.1093/mnras/stac299. arXiv: 2110.09097 [astro-ph.HE].

Appendix D

Bibliography

Almazroui, Mansour, Fahad Saeed, Sajjad Saeed, Muhammad Ismail, Muhammad Azhar Ehsan, M. Nazrul Islam, Muhammad Adnan Abid, Enda O'Brien, Shahzad Kamil, Irfan Ur Rashid, and Imran Nadeem (Aug. 2021). "Projected Changes in Climate Extremes Using CMIP6 Simulations Over SREX Regions." In: *Earth Systems and Environment*. DOI: 10.1007/s41748-021-00250-5.

Aluie, H. (Mar. 2013). "Scale decomposition in compressible turbulence." In: *Physica D Nonlinear Phenomena* 247, pp. 54–65. DOI: 10.1016/j.physd.2012.12.009.

Aluie, Hussein (Apr. 2011). "Compressible Turbulence: The Cascade and its Locality." In: *Phys. Rev. Lett.* 106.17, p. 174502.

Aluie, Hussein, Shengtai Li, and Hui Li (June 2012). "Conservative Cascade of Kinetic Energy in Compressible Turbulence." In: ApJ 751.2, L29, p. L29. DOI: 10.1088/2041-8205/751/2/L29. arXiv: 1107.5771 [nlin.CD].

Bahraminasab, A., M. D. Niry, J. Davoudi, M. Reza Rahimi Tabar, A. A. Masoudi, and K. R. Sreenivasan (June 2008). "Taylor's frozen-flow hypothesis in Burgers turbulence." In: Phys. Rev. E 77.6, 065302, p. 065302. DOI: 10.1103/PhysRevE.77.065302.

Banerjee, S. and S. Galtier (Jan. 2013). "Exact relation with two-point correlation functions and phenomenological approach for compressible magnetohydrodynamic turbulence." In: Phys. Rev. E 87.1, 013019, p. 013019. DOI: 10.1103/PhysRevE.87.013019. arXiv: 1301.2470 [physics.flu-dyn].

Banerjee, Supratik and Alexei G. Kritsuk (Nov. 2017). "Exact relations for energy transfer in self-gravitating isothermal turbulence." In: Phys. Rev. E 96.5, 053116, p. 053116. DOI: 10.1103/PhysRevE.96.053116. arXiv: 1711.05503 [physics.flu-dyn].

Bayly, B. J., C. D. Levermore, and T. Passot (May 1992). "Density variations in weakly compressible flows." In: *Physics of Fluids* 4, pp. 945–954. DOI: 10.1063/1.858275.

Bec, Jérémie and Konstantin Khanin (Aug. 2007). "Burgers turbulence." In: Phys. Rep. 447.1-2, pp. 1–66. DOI: 10.1016/j.physrep.2007.04.002. arXiv: 0704.1611 [nlin.CD].

Benzi, R., L. Biferale, R. T. Fisher, L. P. Kadanoff, D. Q. Lamb, and F. Toschi (June 2008). "Intermittency and Universality in Fully Developed Inviscid and Weakly Compressible Turbulent Flows." In: *Physical Review Letters* 100.23, pp. 234503–+. arXiv: 0709.3073 [nlin.CD].

Benzi, R., S. Ciliberto, R. Tripiccione, C. Baudet, F. Massaioli, and S. Succi (July 1993). "Extended self-similarity in turbulent flows." In: Phys. Rev. E 48, p. 29. DOI: 10.1103/PhysRevE.48.R29.

Berger, M. J. and P. Colella (May 1989). "Local Adaptive Mesh Refinement for Shock Hydrodynamics." In: *Journal of Computational Physics* 82.1, pp. 64–84. DOI: 10.1016/0021-9991(89)90035-1.

Biskamp, D. (July 2003). *Magnetohydrodynamic Turbulence*. Cambridge University Press.

Blue, B. E., S. V. Weber, S. G. Glendinning, N. E. Lanier, D. T. Woods, M. J. Bono, S. N. Dixit, C. A. Haynam, J. P. Holder, D. H. Kalantar, B. J. MacGowan, A. J. Nikitin, V. V. Rekow, B. M. van Wonterghem, E. I. Moses, P. E. Stry, B. H. Wilde, W. W. Hsing, and H. F. Robey (Jan. 2005). "Experimental Investigation of High-Mach-Number 3D Hydrodynamic Jets at the National Ignition Facility." In: Phys. Rev. Lett. 94.9, 095005, p. 095005. DOI: 10.1103/PhysRevLett.94.095005.

Boldyrev, S. (Apr. 2002). "Kolmogorov-Burgers Model for Star-forming Turbulence." In: ApJ 569, pp. 841–845. DOI: 10.1086/339403. eprint: arXiv:astro-ph/0108300.

Boldyrev, S., Å. Nordlund, and P. Padoan (July 2002). "Scaling Relations of Supersonic Turbulence in Star-forming Molecular Clouds." In: ApJ 573, pp. 678–684. DOI: 10.1086/340758. eprint: arXiv:astro-ph/0111345.

Boyden, Ryan D., Stella S. R. Offner, Eric W. Koch, and Erik W. Rosolowsky (June 2018). "Assessing the Impact of Astrochemistry on Molecular Cloud Turbulence Statistics." In: ApJ 860.2, 157, p. 157. DOI: 10.3847/1538-4357/aac76d. arXiv: 1805.09775 [astro-ph.GA].

Brandenburg, A. and Å. Nordlund (Apr. 2011). "Astrophysical turbulence modeling." In: *Reports on Progress in Physics* 74.4, 046901, p. 046901. DOI: 10.1088/0034-4885/74/4/046901. arXiv: 0912.1340 [astro-ph.SR].

Brummel-Smith, Corey, Greg Bryan, Iryna Butsky, Lauren Corlies, Andrew Emerick, John Forbes, Yusuke Fujimoto, Nathan Goldbaum, Philipp Grete, Cameron Hummels, Ji-hoon Kim, Daegene Koh, Miao Li, Yuan Li, Xinyu Li, Brian OShea, Molly Peeples, John Regan, Munier Salem, Wolfram Schmidt, Christine Simpson, Britton Smith, Jason Tumlinson, Matthew Turk, John Wise, Tom Abel, James Bordner, Renyue Cen, David Collins, Brian Crosby, Philipp Edelmann, Oliver Hahn, Robert Harkness, Elizabeth Harper-Clark, Shuo Kong, Alexei Kritsuk, Michael Kuhlen, James Larrue, Eve Lee, Greg Meece, Michael Norman, Jeffrey Oishi, Pascal Paschos, Carolyn Peruta, Alex Razoumov, Daniel Reynolds, Devin Silvia, Samuel Skillman, Stephen

Skory, Geoffrey So, Elizabeth Tasker, Rick Wagner, Peng Wang, Hao Xu, and Fen Zhao (Oct. 2019). "ENZO: An Adaptive Mesh Refinement Code for Astrophysics (Version 2.6)." In: *The Journal of Open Source Software* 4.42, 1636, p. 1636. DOI: 10.21105/joss.01636.

Bryan, Greg L., Michael L. Norman, Brian W. O'Shea, Tom Abel, John H. Wise, Matthew J. Turk, Daniel R. Reynolds, David C. Collins, Peng Wang, Samuel W. Skillman, Britton Smith, Robert P. Harkness, James Bordner, Ji-hoon Kim, Michael Kuhlen, Hao Xu, Nathan Goldbaum, Cameron Hummels, Alexei G. Kritsuk, Elizabeth Tasker, Stephen Skory, Christine M. Simpson, Oliver Hahn, Jeffrey S. Oishi, Geoffrey C. So, Fen Zhao, Renyue Cen, Yuan Li, and Enzo Collaboration (Apr. 2014). "ENZO: An Adaptive Mesh Refinement Code for Astrophysics." In: ApJS 211.2, 19, p. 19. DOI: 10.1088/0067-0049/211/2/19. arXiv: 1307.2265 [astro-ph.IM].

Burkhart, Blakesley (Oct. 2021). "Diagnosing Turbulence in the Neutral and Molecular Interstellar Medium of Galaxies." In: PASP 133.1028, 102001, p. 102001. DOI: 10.1088/1538-3873/ac25cf. arXiv: 2106.02239 [astro-ph.GA].

Chevance, Mélanie, J. M. Diederik Kruijssen, Enrique Vazquez-Semadeni, Fumitaka Nakamura, Ralf Klessen, Javier Ballesteros-Paredes, Shu-ichiro Inutsuka, Angela Adamo, and Patrick Hennebelle (Apr. 2020). "The Molecular Cloud Lifecycle." In: Space Sci. Rev. 216.4, 50, p. 50. DOI: 10.1007/s11214-020-00674-x. arXiv: 2004.06113 [astro-ph.GA].

Chira, R. -A., J. C. Ibáñez-Mejía, M. -M. Mac Low, and Th. Henning (Oct. 2019). "How do velocity structure functions trace gas dynamics in simulated molecular clouds?" In: A&A 630, A97, A97. DOI: 10.1051/0004-6361/201833970. arXiv: 1908.03951 [astro-ph.GA].

Colella, P. and P. R. Woodward (Sept. 1984). "The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations." In: *Journal of Computational Physics* 54, pp. 174–201.

Collins, David C. (2009). "Star formation with adaptive mesh refinement and magnetohydrodynamics." PhD dissertation. La Jolla, CA: University of California, San Diego.

Cook, J. M. (1957). "Technical Notes and Short Papers." In: *Mathematical Tables and Other Aids to Computation* 11.58, pp. 81–82. ISSN: 08916837. URL: http://www.jstor.org/stable/2002156.

Davidson, P. A. (2004). *Turbulence: An Introduction for Scientists and Engineers*. eng. 1st ed. Oxford, UK: Oxford University Press. ISBN: 019852949X.

Davidson, P. A. and B. R. Pearson (Nov. 2005). "Identifying Turbulent Energy Distributions in Real, Rather than Fourier, Space." In: *Physical Review Letters* 95.21, 214501, p. 214501. DOI: 10.1103/PhysRevLett.95.214501.

Davidson, Peter (June 2015). *Turbulence: An Introduction for Scientists and Engineers*. 2nd ed. Oxford University Press. ISBN: 9780198722588. DOI: 10.1093/acprof:oso/9780198722588.001.0001. URL: https://doi.org/10.1093/acprof:oso/9780198722588.001.0001.

Dobler, W., N. E. Haugen, T. A. Yousef, and A. Brandenburg (Aug. 2003). "Bottleneck effect in three-dimensional turbulence simulations." In: Phys. Rev. E 68.2, 026304, p. 026304. DOI: 10.1103/PhysRevE.68.026304. eprint: arXiv:astro-ph/0303324.

Dubrulle, B. (Aug. 1994). "Intermittency in fully developed turbulence: Log-Poisson statistics and generalized scale covariance." In: *Physical Review Letters* 73, pp. 959–962. DOI: 10.1103/PhysRevLett.73.959.

Elmegreen, B. G. and J. Scalo (Sept. 2004). "Interstellar Turbulence I: Observations and Processes." In: ARA&A 42, pp. 211–273. DOI: 10.1146/annurev.astro.41.011802.094859. eprint: arXiv:astro-ph/0404451.

Eyring, Veronika, Sandrine Bony, Gerald A. Meehl, Catherine A. Senior, Bjorn Stevens, Ronald J. Stouffer, and Karl E. Taylor (May 2016). "Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization." In: *Geoscientific Model Development* 9.5, pp. 1937–1958. DOI: 10.5194/gmd-9-1937-2016.

Falkovich, G. (Apr. 1994). "Bottleneck phenomenon in developed turbulence." In: *Physics of Fluids* 6, pp. 1411–1414.

Falkovich, G., I. Fouxon, and Y. Oz (Feb. 2010). "New relations for correlation functions in Navier-Stokes turbulence." In: *Journal of Fluid Mechanics* 644, pp. 465–+. arXiv: 0909.3404 [nlin.CD].

Faraj, A. and X. Yuan (June 2005). "An empirical approach for efficient all-to-all personalized communication on Ethernet switched clusters." In: *2005 International Conference on Parallel Processing (ICPP'05)*, pp. 321–328. DOI: 10.1109/ICPP.2005.19.

Favre, A (1958). "Équations statistiques des gaz turbulents masse, quantité de mouvement." In: *Comptes rendus hebdomadaires des séances de l'Académie des sciences.* 246, pp. 2576, 2723, 2839, 3216.

Federrath, C., J. Roman-Duval, R. S. Klessen, W. Schmidt, and M.-M. Mac Low (Mar. 2010). "Comparing the statistics of interstellar turbulence in simulations and observations. Solenoidal versus compressive turbulence forcing." In: A&A 512, A81, A81. DOI: 10.1051/0004-6361/200912437. arXiv: 0905.1060 [astro-ph.SR].

Fernando F. Grinstein, Len G. Margolin and William J. Rider (2007). *Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics*. Cambridge University Press.

Ferrand, R., S. Galtier, F. Sahraoui, and C. Federrath (Dec. 2020). "Compressible Turbulence in the Interstellar Medium: New Insights from a High-resolution Supersonic Turbulence Simulation." In: ApJ 904.2, 160, p. 160. DOI: 10.3847/1538-4357/abb76e.

Fleck Jr., R. C. (Feb. 1996). "Scaling Relations for the Turbulent, Non–Self-gravitating, Neutral Component of the Interstellar Medium." In: ApJ 458, p. 739. DOI: 10.1086/176853.

Flynn, Michael J. (1972). "Some Computer Organizations and Their Effectiveness." In: *IEEE Transactions on Computers* C-21, pp. 948–960.

Friedrich, R. (Mar. 2007). "Compressible turbulent flows: Aspects of prediction and analysis." In: *Zeitschrift Angewandte Mathematik und Mechanik* 87.3, pp. 189–211. DOI: 10.1002/zamm.200610312.

Frisch, U. (1995). *Turbulence: the Legacy of A. N. Kolmogorov*. Cambridge University Press.

Gabriel, Edgar, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall (Sept. 2004). "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation." In: *Proceedings, 11th European PVM/MPI Users' Group Meeting*. Budapest, Hungary, pp. 97–104.

Galanti, B. and A. Tsinober (Sept. 2004). "Is turbulence ergodic?" In: *Physics Letters A* 330.3-4, pp. 173–180. DOI: 10.1016/j.physleta.2004.07.009.

Galtier, Sébastien and Supratik Banerjee (Sept. 2011). "Exact Relation for Correlation Functions in Compressible Isothermal Turbulence." In: *Physical Review Letters* 107.13, 134501, p. 134501. arXiv: 1108.4529 [astro-ph.SR].

Girichidis, Philipp, Stella S. R. Offner, Alexei G. Kritsuk, Ralf S. Klessen, Patrick Hennebelle, J. M. Diederik Kruijssen, Martin G. H. Krause, Simon C. O. Glover, and Marco Padovani (Jan. 2020). "Physical Processes in Star Formation." In: Space Sci. Rev. 216.4, 68, p. 68. DOI: 10.1007/s11214-020-00693-8. arXiv: 2005.06472 [astro-ph.GA].

Godunov, Sergei Konstantinovich (1959). "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics." In: *Matematicheskii Sbornik* 89.3, pp. 271–306.

Gropp, William, Ewing Lusk, Nathan Doss, and Anthony Skjellum (1996). "A high-performance, portable implementation of the MPI message passing interface standard." In: *Parallel Computing* 22.6, pp. 789–828. ISSN: 0167-8191. DOI: https://doi.org/10.1016/0167-8191(96)00024-5. URL: https://www.sciencedirect.com/science/article/pii/0167819196000245.

Hallman, E. J., B. W. O'Shea, J. O. Burns, M. L. Norman, R. Harkness, and R. Wagner (Dec. 2007). "The Santa Fe Light Cone Simulation Project. I. Confusion and the Warm-Hot Intergalactic Medium in Upcoming Sunyaev-Zel'dovich Effect Surveys." In: ApJ 671, pp. 27–39. arXiv: 0704.2607.

Haworth, Thomas J., Simon C. O. Glover, Christine M. Koepferl, Thomas G. Bisbas, and James E. Dale (Aug. 2018). "Synthetic observations of star formation and the interstellar medium." In: New A Rev. 82, pp. 1–58. DOI: 10.1016/j.newar.2018.06.001. arXiv: 1711.05275 [astro-ph.GA].

Hennebelle, P. and E. Falgarone (Nov. 2012). "Turbulent molecular clouds." In: A&A Rev. 20, p. 55. DOI: 10.1007/s00159-012-0055-y. arXiv: 1211.0637 [astro-ph.GA].

Henriksen, R. N. (Aug. 1991). "On Molecular Cloud Scaling Laws and Star Formation." In: ApJ 377, p. 500. DOI: 10.1086/170379.

Heyer, M. H. and C. M. Brunt (Nov. 2004). "The Universality of Turbulence in Galactic Molecular Clouds." In: ApJ 615, pp. L45–L48. eprint: arXiv:astro-ph/0409420.

Hobbs, A., S. Nayakshin, C. Power, and A. King (June 2011). "Feeding supermassive black holes through supersonic turbulence and ballistic accretion." In: MNRAS 413, pp. 2633–2650. DOI: 10.1111/j.1365-2966.2011.18333.x. arXiv: 1001.3883 [astro-ph.HE].

"IEEE Standard for Floating-Point Arithmetic" (2019). In: *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84. DOI: 10.1109/IEEESTD.2019.8766229.

Ingenito, A. and C. Bruno (Mar. 2010). "Physics and Regimes of Supersonic Combustion." In: *AIAA Journal* 48, pp. 515–525. DOI: 10.2514/1.43652.

Jordan, Peter and Tim Colonius (Jan. 2013). "Wave Packets and Turbulent Jet Noise." In: *Annual Review of Fluid Mechanics* 45.1, pp. 173–195. DOI: 10.1146/annurev-fluid-011212-140756.

Kennicutt, Robert C. and Neal J. Evans (Sept. 2012). "Star Formation in the Milky Way and Nearby Galaxies." In: ARA&A 50, pp. 531–608. DOI: 10.1146/annurev-astro-081811-125610. arXiv: 1204.3552 [astro-ph.GA].

Kim, J. and D. Ryu (Sept. 2005). "Density Power Spectrum of Compressible Hydrodynamic Turbulent Flows." In: ApJ 630, pp. L45–L48. DOI: 10.1086/491600. eprint: arXiv:astro-ph/0507591.

Kitsionas, S., C. Federrath, R. S. Klessen, W. Schmidt, D. J. Price, L. J. Dursi, M. Gritschneder, S. Walch, R. Piontek, J. Kim, A.-K. Jappsen, P. Ciecielag, and M.-M. Mac Low (Dec. 2009). "Algorithmic comparisons of decaying, isothermal, supersonic turbulence." In: A&A 508, pp. 541–560. arXiv: 0810.4599.

Koch, Eric W., Erik W. Rosolowsky, Ryan D. Boyden, Blakesley Burkhart, Adam Ginsburg, Jason L. Loeppky, and Stella S. R. Offner (July 2019). "TURBUSTAT: Turbulence Statistics in Python." In: AJ 158.1, 1, p. 1. DOI: 10.3847/1538-3881/ab1cc0. arXiv: 1904.10484 [astro-ph.IM].

Kolmogorov, A. N. (1941a). "Dissipation of Energy in the Locally Isotropic Turbulence." In: *Dokl. Akad. Nauk SSSR* 32, p. 19. eprint: http://rspa.royalsocietypublishing.org/content/434/1890/15.full.pdf+html.

— (1941b). "On Degeneration of Isotropic Turbulence in an Incompressible Viscous Liquid." In: *Dokl. Akad. Nauk SSSR* 31, p. 538.

— (1941c). "The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers." In: *Dokl. Akad. Nauk SSSR* 30, p. 299.

— (1962). "A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high Reynolds number." In: *Journal of Fluid Mechanics* 13, pp. 82–85. DOI: 10.1017/S0022112062000518.

— (Jan. 1985). "Selected works: Mathematics and mechanics." In: *Moscow Izdatel Nauka*.

Kowal, G. and A. Lazarian (Sept. 2007). "Scaling Relations of Compressible MHD Turbulence." In: ApJ 666, pp. L69–L72. DOI: 10.1086/521788. arXiv: 0705.2464.

Kritsuk, A. G., Å. Nordlund, D. Collins, P. Padoan, M. L. Norman, T. Abel, R. Banerjee, C. Federrath, M. Flock, D. Lee, P. S. Li, W.-C. Müller, R. Teyssier, S. D. Ustyugov, C. Vogel, and H. Xu (Aug. 2011). "Comparing Numerical Methods for Isothermal Magnetized Supersonic Turbulence." In: ApJ 737, 13, p. 13. DOI: 10.1088/0004-637X/737/1/13. arXiv: 1103.5525 [astro-ph.SR].

Kritsuk, A. G., M. L. Norman, and P. Padoan (Feb. 2006). "Adaptive Mesh Refinement for Supersonic Molecular Cloud Turbulence." In: ApJ 638, pp. L25–L28. DOI: 10.1086/500688. eprint: arXiv:astro-ph/0411626.

Kritsuk, A. G., P. Padoan, R. Wagner, and M. L. Norman (Aug. 2007a). "Scaling Laws and Intermittency in Highly Compressible Turbulence." In: *Turbulence and Nonlinear Processes in Astrophysical Plasmas*. Ed. by D. Shaikh and G. P. Zank. Vol. 932. American Institute of Physics Conference Series, pp. 393–399. DOI: 10.1063/1.2778991. arXiv: 0706.0739.

Kritsuk, A. G., S. D. Ustyugov, M. L. Norman, and P. Padoan (Sept. 2010). "Self-organization in Turbulent Molecular Clouds: Compressional Versus Solenoidal Modes." In: *Numerical Modeling of Space Plasma Flows, Astronum-2009*. Ed. by N. V. Pogorelov, E. Audit, and G. P. Zank. Vol. 429. Astronomical Society of the Pacific Conference Series, p. 15. arXiv: 0912.0546 [astro-ph.GA].

Kritsuk, A. G., R. Wagner, and M. L. Norman (Aug. 2013). "Energy Cascade and Scaling in Supersonic Isothermal Turbulence." In: *Journal of Fluid Mechanics* 729, R1, p. 1. DOI: 10.1017/jfm.2013.342.

Kritsuk, A. G., R. Wagner, M. L. Norman, and P. Padoan (Dec. 2006). "High resolution simulations of supersonic turbulence in molecular clouds." In: *Numerical Modeling of*

Space Plasma Flows. Ed. by G. P. Zank and N. V. Pogorelov. Vol. 359. Astronomical Society of the Pacific Conference Series, p. 84. eprint: astro-ph/0607634.

Kritsuk, Alexei G., Michael L. Norman, Paolo Padoan, and Rick Wagner (Aug. 2007b). "The Statistics of Supersonic Isothermal Turbulence." In: ApJ 665, pp. 416–431. arXiv: 0704.3851.

Kritsuk, Alexei G., Michael L. Norman, and Rick Wagner (Jan. 2011). "On the Density Distribution in Star-forming Interstellar Clouds." In: ApJ 727.1, L20, p. L20. DOI: 10.1088/2041-8205/727/1/L20. arXiv: 1007.2950 [astro-ph.GA].

Krumholz, Mark R. and Christopher F. McKee (Sept. 2005). "A General Theory of Turbulence-regulated Star Formation, from Spirals to Ultraluminous Infrared Galaxies." In: ApJ 630.1, pp. 250–268. DOI: 10.1086/431734. arXiv: astro-ph/0505177 [astro-ph].

Landau, L. D. and E. M. Lifshitz (1987). *Fluid Mechanics*. Pergamon.

Larson, R. B. (Mar. 1981). "Turbulence and star formation in molecular clouds." In: MNRAS 194, pp. 809–826.

Lele, S. K. (1994). "Compressibility effects on turbulence." In: *Annual Review of Fluid Mechanics* 26, pp. 211–254. DOI: 10.1146/annurev.fl.26.010194.001235.

Lighthill, M. J. (1955). "The Effect of Compressibility on Turbulence." In: *Gas Dynamics of Cosmic Clouds*. Vol. 2. IAU Symposium, p. 121.

Lumley, John L. and Akiva M. Yaglom (May 2001). "A Century of Turbulence." In: *Flow, Turbulence and Combustion* 66.3, pp. 241–286. DOI: 10.1023/A:1012437421667. URL: https://doi.org/10.1023/A:1012437421667.

Mac Low, M.-M. (Oct. 1999). "The Energy Dissipation Rate of Supersonic, Magnetohydrodynamic Turbulence in Molecular Clouds." In: ApJ 524, pp. 169–178. eprint: arXiv:astro-ph/9809177.

Mac Low, M.-M. and R. S. Klessen (Jan. 2004). "Control of star formation by supersonic turbulence." In: *Reviews of Modern Physics* 76, pp. 125–194. DOI: 10.1103/RevModPhys.76.125. eprint: arXiv:astro-ph/0301093.

Mascagni, Michael and Ashok Srinivasan (Sept. 2000). "Algorithm 806: SPRNG: a scalable library for pseudorandom number generation." In: *ACM Trans. Math. Softw.* 26.3, pp. 436–461. ISSN: 0098-3500. DOI: 10.1145/358407.358427. URL: http://doi.acm.org/10.1145/358407.358427.

McKee, C. F. and E. C. Ostriker (Sept. 2007). "Theory of Star Formation." In: ARA&A 45, pp. 565–687. arXiv: 0707.3514.

Message Passing Interface Forum (June 2021). *MPI: A Message-Passing Interface Standard Version 4.0*. URL: https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf.

Mitra, D., J. Bec, R. Pandit, and U. Frisch (May 2005). "Is Multiscaling an Artifact in the Stochastically Forced Burgers Equation?" In: *Physical Review Letters* 94.19, 194501, p. 194501. DOI: 10.1103/PhysRevLett.94.194501. eprint: arXiv:nlin/0406049.

Moffat, A. F. J. and C. Robert (Jan. 1994). "Clumping and mass loss in hot star winds." In: ApJ 421, pp. 310–313. DOI: 10.1086/173648.

Norman, M. L., G. L. Bryan, R. Harkness, J. Bordner, D. Reynolds, B. O'Shea, and R. Wagner (2007). "Simulating Cosmological Evolution with Enzo." In: *Petascale Computing: Algorithms and Applications*. Ed. by D. A. Bader. Chapman & Hall/CRC Computational Science. CRC Press, 83–+.

O'Shea, B. W., G. Bryan, J. Bordner, M. L. Norman, T. Abel, R. Harkness, and A. Kritsuk (Mar. 2004). "Introducing Enzo, an AMR Cosmology Application." In: *ArXiv Astrophysics e-prints*. eprint: arXiv:astro-ph/0403044.

Ogden, D. E., G. A. Glatzmaier, and K. H. Wohletz (Apr. 2008). "Effects of vent overpressure on buoyant eruption columns: Implications for plume stability." In: *Earth and Planetary Science Letters* 268, pp. 283–292. DOI: 10.1016/j.epsl.2008.01.014.

Orkisz, Jan H., Jérôme Pety, Maryvonne Gerin, Emeric Bron, Viviana V. Guzmán, Sébastien Bardeau, Javier R. Goicoechea, Pierre Gratier, Franck Le Petit, François Levrier, Harvey Liszt, Karin Öberg, Nicolas Peretto, Evelyne Roueff, Albrecht Sievers, and Pascal Tremblin (Mar. 2017). "Turbulence and star formation efficiency in molecular clouds: solenoidal versus compressive motions in Orion B." In: A&A 599, A99, A99. DOI: 10.1051/0004-6361/201629220. arXiv: 1701.00962 [astro-ph.GA].

Orszag, Steven A. and G. S. Patterson (Jan. 1972). "Numerical Simulation of Three-Dimensional Homogeneous Isotropic Turbulence." In: Phys. Rev. Lett. 28.2, pp. 76–79. DOI: 10.1103/PhysRevLett.28.76.

Padoan, P., R. Jimenez, Å. Nordlund, and S. Boldyrev (May 2004). "Structure Function Scaling in Compressible Super-Alfvénic MHD Turbulence." In: *Physical Review Letters* 92.19, 191102, p. 191102. DOI: 10.1103/PhysRevLett.92.191102. eprint: arXiv:astro-ph/0301026.

Padoan, Paolo, Liubin Pan, Troels Haugbølle, and Åke Nordlund (May 2016). "Supernova Driving. I. The Origin of Molecular Cloud Turbulence." In: ApJ 822.1, 11, p. 11. DOI: 10.3847/0004-637X/822/1/11. arXiv: 1509.04663 [astro-ph.GA].

Pan, L., P. Padoan, and A. G. Kritsuk (Jan. 2009). "Dissipative Structures in Supersonic Turbulence." In: *Physical Review Letters* 102.3, pp. 034501–+. arXiv: 0808.1330.

Panda, Dhabaleswar Kumar, Hari Subramoni, Ching-Hsiang Chu, and Mohammadreza Bayatpour (2021). "The MVAPICH project: Transforming research into high-performance MPI library for HPC community." In: *Journal of Computational Science* 52. Case Studies in Translational Computer Science, p. 101208. ISSN: 1877-

7503. DOI: https://doi.org/10.1016/j.jocs.2020.101208. URL: https://www.sciencedirect.com/science/article/pii/S1877750320305093.

Pješivac-Grbović, Jelena, Thara Angskun, George Bosilca, Graham E Fagg, Edgar Gabriel, and Jack J Dongarra (2007). "Performance analysis of MPI collective operations." In: *Cluster Computing* 10.2, pp. 127–143.

Pope, Stephen B. (2000). *Turbulent Flows*. Cambridge University Press.

Porter, D., A. Pouquet, and P. Woodward (Aug. 2002). "Measures of intermittency in driven supersonic flows." In: Phys. Rev. E 66.2, 026301, p. 026301.

Porter, D. H., A. Pouquet, and P. R. Woodward (Nov. 1992). "A numerical study of supersonic turbulence." In: *Theoretical and Computational Fluid Dynamics* 4, pp. 13–49. DOI: 10.1007/BF00417962.

— (June 1994). "Kolmogorov-like spectra in decaying three-dimensional supersonic flows." In: *Physics of Fluids* 6, pp. 2133–2142. DOI: 10.1063/1.868217.

Porter, D. H. and P. R. Woodward (July 1994). "High-resolution simulations of compressible convection using the piecewise-parabolic method." In: ApJS 93, pp. 309–349.

Price, D. J. and C. Federrath (Aug. 2010). "A comparison between grid and particle methods on the statistics of driven, supersonic, isothermal turbulence." In: MNRAS 406, pp. 1659–1674. DOI: 10.1111/j.1365-2966.2010.16810.x. arXiv: 1004.1446 [astro-ph.GA].

Reynolds, Osborne (1883). "An Experimental Investigation of the Circumstances Which Determine Whether the Motion of Water Shall Be Direct or Sinuous, and of the Law of Resistance in Parallel Channels." In: *Philosophical Transactions of the Royal Society of London* 174, pp. 935–982. ISSN: 02610523. URL: http://www.jstor.org/stable/109431 (visited on 11/06/2022).

— (Jan. 1895). "On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion." In: *Philosophical Transactions of the Royal Society of London Series A* 186, pp. 123–164. DOI: 10.1098/rsta.1895.0004.

Richardson, L. G. (1922). *Weather Prediction by Numerical Process*. Cambridge University Press.

Robertson, Brant and Peter Goldreich (May 2012). "Adiabatic Heating of Contracting Turbulent Fluids." In: ApJ 750, L31, p. L31. DOI: 10.1088/2041-8205/750/2/L31. arXiv: 1203.4815 [astro-ph.GA].

Sadhukhan, Shubhadeep, Shashwat Bhattacharya, and Mahendra Verma (Jan. 2021). "fastSF: A parallel code for computing the structure functions of turbulence." In: *The Journal of Open Source Software* 6.57, 2185, p. 2185. DOI: 10.21105/joss.02185.

Sagaut, Pierre and Claude Cambon (2018). *Homogeneous Turbulence Dynamics*. eng. 2nd ed. Cham: Springer International Publishing. ISBN: 3319731610.

Sayles, R.S. and T.R. Thomas (1977). "The Spatial Representation of Surface Roughness by Means of the Structure Function: A Practical Alternative to Correlation." In: *Wear* 42.2, pp. 263–276. ISSN: 0043-1648. DOI: https://doi.org/10.1016/0043-1648(77)90057-6. URL: https://www.sciencedirect.com/science/article/pii/0043164877900576.

Schmidt, W., C. Federrath, and R. Klessen (Nov. 2008). "Is the Scaling of Supersonic Turbulence Universal?" In: *Physical Review Letters* 101.19, 194505, p. 194505. DOI: 10.1103/PhysRevLett.101.194505. arXiv: 0810.1397.

Schmidt, Wolfram and Philipp Grete (Oct. 2019). "Kinetic and internal energy transfer in implicit large-eddy simulations of forced compressible turbulence." In: Phys. Rev. E 100.4, 043116, p. 043116. DOI: 10.1103/PhysRevE.100.043116. arXiv: 1906.12228 [physics.flu-dyn].

Schwarz, C., C. Beetz, J. Dreher, and R. Grauer (Feb. 2010a). "Lyapunov exponents and information dimension of the mass distribution in turbulent compressible flows." In: *Physics Letters A* 374, pp. 1039–1042. DOI: 10.1016/j.physleta.2009.12.036.

Schwarz, Ch., Ch. Beetz, J. Dreher, and R. Grauer (Feb. 2010b). "Lyapunov exponents and information dimension of the mass distribution in turbulent compressible flows." In: *Physics Letters A* 374.8, pp. 1039–1042. DOI: 10.1016/j.physleta.2009.12.036.

Sekundov, A. N. and K. Ya. Yakubovskii (Mar. 2019). "Analysis of Kolmogorov's Hypotheses in a Compressible Turbulent Flow." In: *Fluid Dynamics* 54.2, pp. 184–193. DOI: 10.1134/S0015462819020113.

She, Z.-S. and E. Leveque (Jan. 1994). "Universal scaling laws in fully developed turbulence." In: *Physical Review Letters* 72, pp. 336–339. DOI: 10.1103/PhysRevLett.72.336.

She, Zhen-Su and Edward C. Waymire (Jan. 1995). "Quantized Energy Cascade and Log-Poisson Statistics in Fully Developed Turbulence." In: Phys. Rev. Lett. 74.2, pp. 262–265. DOI: 10.1103/PhysRevLett.74.262.

Skory, Stephen (2010). "Investigating a method of producing "red and dead" galaxies." PhD dissertation. La Jolla, CA: University of California, San Diego.

Sur, S., H. W. Jin, and D. K. Panda (Aug. 2004). "Efficient and scalable all-to-all personalized exchange for InfiniBand-based clusters." In: *Parallel Processing, 2004. ICPP 2004. International Conference on*, 275–282 vol.1. DOI: 10.1109/ICPP.2004.1327932.

Sytine, I. V., D. H. Porter, P. R. Woodward, S. W. Hodson, and K.-H. Winkler (Mar. 2000). "Convergence Tests for the Piecewise Parabolic Method and Navier-Stokes Solutions for Homogeneous Compressible Turbulence." In: *Journal of Computational Physics* 158, pp. 225–238.

Teyssier, R. (Apr. 2002). "Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES." In: A&A 385, pp. 337–364. DOI: 10.1051/0004-6361:20011817. eprint: astro-ph/0111367.

Tsinober, Arkady (2004). *An Informal Introduction to Turbulence*. 1st ed. Vol. 63. Fluid Mechanics and Its Applications. Springer Dordrecht. DOI: 10.1007/0-306-48384-X.

Turk, Matthew J., Britton D. Smith, Jeffrey S. Oishi, Stephen Skory, Samuel W. Skillman, Tom Abel, and Michael L. Norman (Jan. 2011). "yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data." In: ApJS 192.1, 9, p. 9. DOI: 10.1088/0067-0049/192/1/9. arXiv: 1011.3514 [astro-ph.IM].

Vazza, F., G. Brunetti, A. Kritsuk, R. Wagner, C. Gheller, and M. Norman (Sept. 2009). "Turbulent motions and shocks waves in galaxy clusters simulated with adaptive mesh refinement." In: A&A 504.1, pp. 33–43. DOI: 10.1051/0004-6361/200912535. arXiv: 0905.3169 [astro-ph.CO].

von Kármán, T. and L. Howarth (Jan. 1938). "On the Statistical Theory of Isotropic Turbulence." In: *Royal Society of London Proceedings Series A* 164, pp. 192–215. DOI: 10.1098/rspa.1938.0013.

von Weizsäcker, C. F. (Sept. 1951). "The Evolution of Galaxies and Stars." In: ApJ 114, p. 165. DOI: 10.1086/145462.

Wagner, R., G. Falkovich, A. G. Kritsuk, and M. L. Norman (Dec. 2012). "Flux Correlations in Supersonic Isothermal Turbulence." In: *Journal of Fluid Mechanics* 713, pp. 482–490. DOI: 10.1017/jfm.2012.470. arXiv: 1209.2164 [physics.flu-dyn].

Wang, J., L.-P. Wang, Z. Xiao, Y. Shi, and S. Chen (July 2010). "A hybrid numerical simulation of isotropic compressible turbulence." In: *Journal of Computational Physics* 229, pp. 5257–5279. DOI: 10.1016/j.jcp.2010.03.042.

Wang, Jianchun, Yipeng Shi, Lian-Ping Wang, Zuoli Xiao, X. T. He, and Shiyi Chen (Dec. 2012). "Effect of compressibility on the small-scale structures in isotropic turbulence." In: *Journal of Fluid Mechanics* 713, pp. 588–631. DOI: 10.1017/jfm.2012.474.

Wang, Jianchun, Minping Wan, Song Chen, and Shiyi Chen (Apr. 2018). "Kinetic energy transfer in compressible isotropic turbulence." In: *Journal of Fluid Mechanics* 841, pp. 581–613. DOI: 10.1017/jfm.2018.23.

Wang, Jianchun, Yantao Yang, Yipeng Shi, Zuoli Xiao, X. T. He, and Shiyi Chen (May 2013). "Cascade of Kinetic Energy in Three-Dimensional Compressible Turbulence." In: Phys. Rev. Lett. 110.21, 214505, p. 214505. DOI: 10.1103/PhysRevLett.110.214505.

Yuen, Ka Ho, Ka Wai Ho, and Alex Lazarian (Apr. 2021). "Technique for Separating Velocity and Density Contributions in Spectroscopic Data and Its Application to Studying Turbulence and Magnetic Fields." In: ApJ 910.2, 161, p. 161. DOI: 10.3847/1538-4357/abe4d4. arXiv: 2012.15776 [astro-ph.GA].

Zakharov, V. E., V. S. L'Vov, and G. Falkovich (1992). *Kolmogorov spectra of turbulence I: Wave turbulence*. Springer Berlin, Heidelberg.

Zrake, Jonathan and Andrew I. MacFadyen (Jan. 2012). "Numerical Simulations of Driven Relativistic Magnetohydrodynamic Turbulence." In: ApJ 744.1, 32, p. 32. DOI: 10.1088/0004-637X/744/1/32. arXiv: 1108.1991 [astro-ph.HE].