Instituto Nacional de Matemática Pura e Aplicada

Data-based approach for time-correlated closures of turbulence models

by

Júlia Domingues Lemos

Rio de Janeiro-RJ
October 2022

Instituto Nacional de Matemática Pura e Aplicada

Data-based approach for time-correlated closures of turbulence models

by

Júlia Domingues Lemos

under the supervision of Prof. Alexei A. Mailybaev

Thesis presented to the Postgraduate Program in Mathematics at Instituto Nacional de Matemática Pura e Aplicada-IMPA as partial fulfillment of the requirements for the degree of Doctor of Mathematics.

Rio de Janeiro-RJ
October 2022

Thesis committee

André Nachbin, IMPA

Fabio Pereira dos Santos, UFRJ

Massimo Cencini, CNR

Theodore D. Drivas, SBU

Yulia Petrova, IMPA

Alexei Mailybaev, IMPA

October 2022

Agradecimentos

Gostaria de agradecer aos meus pais, Cristina e Marco, pelo apoio e suporte incondicional, e à minha irmã Larissa, por resolver inúmeros pepinos diariamente. Agradeço também a toda a minha família, por entender que não posso sempre estar presente, mas que não passo um dia sequer sem sentir saudades.

Aos meus familiares que não estão mais aqui, sou grata pelo tempo que passamos juntos. Aos meus amigos que não estão mais aqui, sou grata pelos objetivos que compartilhamos desde crianças, e que hoje realizo, infelizmente, sem vocês.

Aos meus colegas de laboratório e de pesquisa, perto ou longe, por todas as discussões enriquecedoras, acadêmicas ou não. Em especial, quero agradecer meu companheiro Hugo, pela confiança, pelo suporte, pelo carinho e por nunca desistir de mim, mesmo nos momentos mais sombrios. Sou grata por compartilhar este período da minha vida com alguém tão especial.

A todas as mulheres cientistas, especialmente as brasileiras, que lutam diariamente por um país melhor e que são exemplos para milhares de meninas. Quero agradecer também a todas as minhas amigas, que entendem a dificuldade de ser mulher em um ambiente predominantemente masculino e que me ajudaram a navegar com clareza até as situações mais loucas.

Ao meu psicólogo Guilherme, por manter minha saúde mental sempre em dia e minhas paranóias sempre em cheque.

Ao meu orientador Alexei, figura vital na elaboração deste trabalho, pela paciência, pela compreensão e por ter me ensinado muito mais do que dinâmica de fluidos. Agradeço especialmente a humanidade e o respeito com que sempre me tratou. Sua postura profissional é uma referência para mim.

A todas as pessoas que colaboraram, contribuíram, ou escutaram áudios com mais de três minutos de duração a respeito deste período da minha vida, deixo aqui minha graditão.

Acknowledgments

I would like to thank my parents, Cristina and Marco, for the unconditional love and support, and my sister Larissa, for handling all sorts of pickles on a daily basis. I want to thank my family, for understanding that I cannot always be there, but that not a day goes by where I do not miss them.

To my family members who are no longer here, I am grateful for the time we spent together. To my friends who are no longer here, I am grateful for the goals we shared since we were children, one of them I am accomplishing today, unfortunately, without you.

To my lab & research mates, near or far, thank you for all the enriching discussions, being them of academic nature or not. Especially, I want to thank my partner Hugo, for the trust, for the support, for always caring and for never giving up on me, not even in the darkest of times. I am grateful for sharing these years of my life with someone so special.

I would like to acknowledge all women in science, especially the Brazilian ones, who fight every day for a better country and are role models for thousands of girls. I would like to thank all my female friends who understand how difficult it is to be a woman in a predominantly male environment. You all helped me navigate even the wildest of situations with clarity.

I would also like to thank my therapist Guilherme, for the astounding work in keeping my mental health up to date and my paranoia always in check.

To my advisor Alexei, a vital character in the making of this work, thank you for the patience, for always understanding, and for teaching me so much more than fluid dynamics. I am especially thankful for the humane and respectful treatment I always received. Your professional attitude is a reference to me.

To everyone who collaborated, contributed or listened to audio messages that lasted longer than three minutes about this period of my life, I hereby express my gratitude.

The meanest dog you'll ever meet
He ain't the hound dog in the street
He bares some teeth and tears some skin
But brother that's the worst of him
The dog you really got to dread
Is the one that howls inside your head
It's him whose howling drives men mad
And a mind to its undoing.

Anaïs Mitchell, *Hadestown*

List of Figures

Symbols

$\mathbf{x}$	spatial variable
t	original time variable
$\mathbf{u}$	velocity field
p	pressure
ν	viscosity
ρ	density
$\mathbf{f}$	forcing
ω	vorticity
$\mathcal{L}$	period of solution
i	imaginary unit
$\mathbf{k}$	wave vector
k	wave number
J	filter scale
$\mathbf{u}^<$	the lesser of velocity
$\mathbf{u}^>$	the greater of velocity
$\mathcal{E}$	cumulative energy spectrum
Π	energy flux
Ω	cumulative enstrophy
$\mathcal{F}$	cumulative energy injected by forcing
Re	Reynolds number
U	characteristic velocity
L	characteristic length
l	scale of motion
δu	velocity increment

$\bar{\varepsilon}$	mean energy disspation
S_p	structure function of order p
E	energy spectrum
H	helicity
u	shell velocity
θ, α	phase of shell velocity
n	Sabra shell index
s	cut-off shell
λ	shell spacing
$\mathcal{U}$	rescaled shell velocity
N	rescaled shell index
m	reference shell
T_m	local temporal scale
τ	intrinsic time
z	multipliers
w	absolute value of multipliers
δ, Δ	phase of multiplier
ξ	derivative of local temporal scale with respect to original time
K	amount of Gaussian components
$\mathcal{N}$	normal distribution
μ_k	mean of k-th component
Σ_k	variance of k-th component
π_k	weight of k-th component
$\mathbf{z}$	latent variable in GMM formulation
$\mathcal{Q}$	log-likelihood
$\boldsymbol{\theta}$	set of parameters
γ	responsibility
f, g	density estimation
$\Delta\tau$	time delay

Resumo

Apesar de mais de um século de pesquisas ativas, não há uma descrição analítica para turbulência desenvolvida. Atualmente, ideias fenomenológicas são amplamente utilizadas em aplicações práticas como fechamento de pequenas escalas para simulação numérica de escoamentos turbulentos. Neste trabalho utilizamos um modelo de turbulência do tipo shell na tentativa de de construir um fechamento que possua fundamentos teóricos sólidos e que capture propriedades probabilístices intrínsecas a escoamentos turbulentos. Modelos de turbulência do tipo shell são uma classe de sistemas dinâmicos determinísticos usados para modelar cascata de energia e outros aspectos importantes da equação de Navier-Stokes. Alguns, como o modelo Sabra, também apresentam intermitência. Redimensionamos as variáveis do Sabra de modo que observamos simetrias ocultas e estatísticas universais. Em seguida, utilizamos as boas propriedades destas estatísticas para escrever modelos de fechamento, ou seja, expressões faltantes para algumas das variáveis do Sabra. A base de nossos modelos de fechamento é a aproximação de funções densidade de probabilidade utilizando um Modelo de Mistura Gaussiana, o que os torna naturalmente probabilísticos e permite que escrevamos fechamentos temporalmente condicionados. Também fornecemos uma estrutura na qual podemos empregar outras ferramentas de aprendizado de máquina com aspecto caixa-preta reduzido.

Palavras-chave: turbulência, modelos shell, fechamento probabilístico, condicionamento ao histórico, simetria oculta.

Abstract

Developed turbulent motion of fluid still lacks an analytical description despite more than a century of active research. Nowadays phenomenological ideas are widely used in practical applications, such as small-scale closures for numerical simulations of turbulent flows. In the present work, we use a shell model of turbulence to construct a closure intended to have a solid theoretical background and to capture intrinsic probabilistic features of turbulence. Shell models of turbulence are dynamical deterministic systems used to model energy cascade and other key aspects of the Navier-Stokes equation. Some, such as the Sabra model, also present intermittency. We rescale the variables of the Sabra model in a way which leads to hidden symmetries and universal distributions. We then use such fine distributions to write closures, i.e., missing expressions for some of the Sabra variables. Our closures rely on approximating probability density functions using a Gaussian Mixture Model, which makes them probabilistic by nature and allows us to write time-correlated closures. We also provide a framework where other Machine Learning tools can be employed with reduced black-box aspects.

Keywords: turbulence, shell models, probabilistic closure, time-correlation, hidden symmetry.

Contents

Chapter 1

Introduction

Let us begin by writing the Navier-Stokes equation (NSE) [11] for incompressible fluids.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho} + \nu \Delta \mathbf{u} + \mathbf{f}, \tag{1.1}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{1.2}$$

In this equation we are computing the components of a velocity field $\mathbf{u} = (u_1(\mathbf{x}, t), u_2(\mathbf{x}, t), u_3(\mathbf{x}, t))$, with spatial coordinates $\mathbf{x} = (x_1, x_2, x_3)$, $\mathbf{x} \in \mathbb{R}^3$, and time t, of an incompressible fluid with a certain viscosity $\nu \in \mathbb{R}_+$ under a pressure $p(\mathbf{x}, t)$ and an external forcing $\mathbf{f}(\mathbf{x}, t)$. The incompressibility condition in equation (1.2) tells us that density of fluid ρ remains constant. We are also assuming periodic boundary conditions, meaning $\mathbf{u}(x_1 + q\mathcal{L}, x_2 + r\mathcal{L}, x_3 + s\mathcal{L}, t) = \mathbf{u}(x_1, x_2, x_3, t)$ for all x_1, x_2, x_3, $q, r, s \in \mathbb{Z}$ and given period $\mathcal{L} > 0$.

Now, let us write the velocity as a Fourier series

$$\mathbf{u}(\mathbf{x}, t) = \sum_{\mathbf{k}} \hat{\mathbf{u}}_{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{x}} \tag{1.3}$$

with wave vectors $\mathbf{k} \in (2\pi/\mathcal{L})\mathbb{Z}^3$ and $k = |\mathbf{k}|$. Choosing a parameter $J > 0$ we can define low/high-pass filters, respectively, as [53]

$$\mathbf{u}_J^<(\mathbf{x}) = \sum_{k \leq J} \hat{\mathbf{u}}_\mathbf{k} e^{i\mathbf{k}\cdot\mathbf{x}} \tag{1.4}$$

$$\mathbf{u}_J^>(\mathbf{x}) = \sum_{k > J} \hat{\mathbf{u}}_\mathbf{k} e^{i\mathbf{k}\cdot\mathbf{x}} \tag{1.5}$$

We can associate to each parameter J a length $l = 2\pi/J$, called the scale of the filter. With this filter, we can write the following decomposition

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_J^<(\mathbf{x}) + \mathbf{u}_J^>(\mathbf{x}) \tag{1.6}$$

This decomposition allow us to think of $\mathbf{u}_J^<$ and $\mathbf{u}_J^>$ as parts of $\mathbf{u}$, where $\mathbf{u}_J^<$ contains information about scales larger than l, while $\mathbf{u}_J^>$ contains information about scales smaller than l. They are typically called *lesser* and *greater*, respectively.

We would like to better understand what is the energy budget as we look at NSE scale by scale. The procedure here, described in detail in [20], consists in splitting the velocity as in equation (1.6), taking the scalar product with $\mathbf{u}_J^<(\mathbf{x})$, averaging in time, using the incompressibility condition from equation (1.2) and simplifying the terms. We then have the energy balance equation

$$\frac{\partial \mathcal{E}_J}{\partial t} + \Pi_J = -2\nu\Omega_J + \mathcal{F}_J \tag{1.7}$$

The first term of the left-hand side of equation (1.7) involves the cumulative energy spectrum, defined as

$$\mathcal{E}_J = \frac{1}{2}\langle|\mathbf{u}_J^<(\mathbf{x})|^2\rangle \tag{1.8}$$

where $\langle\cdot\rangle$ means time averaging. The second term on the left-hand side of equation (1.7) is the energy flux through a wavenumber J, given by

$$\Pi_J = \langle\mathbf{u}_J^< \cdot (\mathbf{u}_J^< \cdot \nabla\mathbf{u}_J^>)\rangle + \langle\mathbf{u}_J^< \cdot (\mathbf{u}_J^> \cdot \nabla\mathbf{u}_J^>)\rangle. \tag{1.9}$$

The second term on the right hand side of equation (1.7) is cumulative energy injected by the forcing term, defined below as

$$\mathcal{F}_J = \langle \mathbf{f}_J^{>} \cdot \mathbf{u}_J^{<} \rangle \tag{1.10}$$

and, lastly, considering the vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, the cumulative enstrophy is given by

$$\Omega_J = \frac{1}{2} \langle |\boldsymbol{\omega}_J^{>}|^2 \rangle \tag{1.11}$$

One way to interpret equation (1.7) is to think that the energy injected by the forcing term adds to the rate of change of the energy at scales larger than l, but the viscous term and the flux of energy to smaller scales both subtract from it.

1.1 A quick word on turbulence

Before we start discussing shell models, we must introduce some aspects of turbulent flow, which will lay the groundwork for later chapters. Firstly, let us define a characteristic velocity U and a characteristic length L. The NSE has a scale invariance [20], which means that if we do the following changes

$$\tilde{\mathbf{x}} = L\mathbf{x}, \tag{1.12}$$

$$\tilde{\mathbf{u}} = U\mathbf{u}, \tag{1.13}$$

$$\tilde{t} = \frac{L}{U} t, \tag{1.14}$$

then equation (1.1) for the new variables becomes

$$\frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{t}} + \tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}} = -\nabla \tilde{p} + \mathrm{Re}^{-1} \Delta \tilde{\mathbf{u}} + \tilde{\mathbf{f}}, \tag{1.15}$$

where $\mathrm{Re} = \frac{UL}{\nu}$ is called the Reynolds number [74] and $\tilde{p}$ and $\tilde{\mathbf{f}}$ are analogously scaled.

We can think of the Reynolds number as a controlling parameter for the flow. The transition from laminar to turbulent flow has been observed in several experiments [16], including by Reynolds himself [63], and is associated with the increase of the Reynolds number. For a flow inside a long pipe, for example, the flow that was once laminar, turns turbulent at $\mathrm{Re} = 2500$, with U being the mean flow velocity and L being the pipe diameter [23]. In this work we are always in a high Reynolds number regime.

If we are talking about an unforced system with no injection of energy in any scale, say, a cup of coffee that has been stirred and left alone, then the turbulence is decaying and all energy will eventually be dissipated by the viscous term. We would like to work in a situation where, on average, the energy injected in the turbulent flow on large scales is balanced by the dissipation through viscous action.

The above paragraph sheds light into something that equation (1.9) had already hinted: that turbulence is a multiscale phenomenon. Indeed, there is a myriad of scales between the integral scale, where energy is being injected, and smaller scales where energy is being dissipated, and what happens between those two is of utmost importance. The energy cascade was first described by Richardson in [41] as big whirls splitting into smaller whirls, that split into even smaller whirls, until the whirls are so tiny that the viscosity smooths them out. This is a symptom of the fact that the energy injected in the larger scales is transferred to smaller scales until it reaches a scale so small, known as the Kolmogorov's scale, that the viscous term becomes the dominant term in the dynamic and the energy is dissipated as heat. The set of scales where energy is being transferred is called the inertial interval, while the set of scales where energy is being dissipated is called the dissipation range.

Any symmetries the NSE could present in particular flows are successively broken as Reynolds number keeps increasing. It was conjectured by Kolmogorov in [37] that these broken symmetries are recovered in a statistical sense in the stage of fully developed turbulence [20]. In particular, that the turbulent flow is homogeneous and isotropic, which is the same as saying it is invariant under translations and rotations, respectively.

To help us see things more clearly, let us define a longitudinal velocity increment, or the velocity characteristic of scale $\mathbf{l}$, of length $l = |\mathbf{l}|$, as

$$\delta u(\mathbf{l}) = |\mathbf{u}(\mathbf{x}+\mathbf{l}) - \mathbf{u}(\mathbf{x})| \cdot \frac{\mathbf{l}}{l}. \tag{1.16}$$

Since we are in the context of isotropic homogeneous turbulence, the probability distribution of velocity increment only depends on l.

If we think of an eddy of size l, scales of size larger than l will essentially sweep the eddy along the dominant direction of the flow. However, if this eddy contains a smaller eddy, its effect on the smaller eddy is the same as the larger scales on itself, which indicates self-similarity, since there is nothing particular about this choice of l [61]. Given two scales $\mathbf{l}_1$ and $\mathbf{l}_2$, such that $l_2 < l_1 \leq L$, self similarity translates as existing an universal function f such that

$$\delta u(l_2) = f\left(\frac{l_1}{l_2}\right)\delta u(l_1). \tag{1.17}$$

The Kolmogorov theory, known as K41 [37], considers a functional relation of the type

$$\delta u(l) = f\left(l\bar{\varepsilon}\right). \tag{1.18}$$

and denote by $\bar{\varepsilon}$ the mean dissipation of energy per unit mass. Now, under the International System of Units, δu is measured in m/s, while l is measured in m and $\bar{\varepsilon}$ is measured in m^2/s^3. Comparing the dimensions of both sides we see that an exponent of $1/3$ is required on the right-hand side, and considering that the left-hand side should remain unchanged under change of units, we see that the only option is f being proportional to the cubic root of $l\bar{\varepsilon}$, meaning

$$\delta u(l) \sim (\bar{\varepsilon} l)^{1/3}. \tag{1.19}$$

To have an estimation on how big is the Kolmogorov scale, where viscous dissipation takes place, and which we will call η, let us first see that

$$\bar{\varepsilon} \sim \nu \mathbf{u} \Delta \mathbf{u} \sim \frac{\nu \delta u(\eta)^2}{\eta^2}, \tag{1.20}$$

where the second term estimates the energy dissipation by viscous forces. Using (1.19) we get

$$\eta \sim \left(\frac{\bar{\varepsilon}}{\nu^3}\right)^{-1/4}. \tag{1.21}$$

At the integral scale, by a dimensional argument, we have

$$\bar{\varepsilon} \sim \frac{U^3}{L} \tag{1.22}$$

which, along with 1.21, leads to

$$\frac{\eta}{L} \sim \mathrm{Re}^{-3/4}. \tag{1.23}$$

We will also need to define other quantities, such as the structure functions of order p

$$S_p(l) = \langle \delta u(l)^p \rangle, \tag{1.24}$$

which are time averages of moments of velocities increment of order p.

Under the assumption of finite mean dissipation, an exact law, known as the Four-fifths law, for the third order structure function was derived in [36], namely

$$S_3(l) = -\frac{4}{5}\bar{\varepsilon} l. \tag{1.25}$$

This law was deduced in detail, for example, in [20]. It leads, along with the self similarity hypothesis, to a theoretical prediction for all structure functions, given by

$$S_p(l) \sim l^{\frac{p}{3}}. \tag{1.26}$$

However, experimental and numerical data, as well as subsequent works such as [3, 44, 55] suggest that structure functions do scale with length, only not with exponent $p/3$, but with a nonlinear exponent $\zeta(p)$ instead. This anomalous scaling is caused by intermittent fluctuations of small scales of motion [82, 73]. Intermittency in small scales was first reported in [33] and works such as [57, 38, 21] presented different approaches to describe it, arguing, amongst other things, that very small fluctuations in energy dissipation are dramatically amplified in the dissipation range, or that intermittent behaviour is related to complex singularities in some specific equations. In fact, intermittency is closely related to the fact that NSE's symmetries are broken at high Reynolds number [61].

From equation (1.8), in the context of isotropic homogeneous stationary turbulence, the energy spectrum is given by

$$E(k) = \frac{d\mathcal{E}(k)}{dk}. \tag{1.27}$$

Then $E(k)dk$ is the energy contained in the interval of wave numbers $[k, k + dk]$. Also,

$$E(k) \sim \bar{\varepsilon}^{2/3} k^{-5/3} \tag{1.28}$$

is the phenomenological prediction given in [37].

1.2 The closure problem

If we want to run a simulation of the NSE, we need to discretize all our variables, in phase or in physical space, and from this discretization we can assemble a computational grid. The spatial fineness of this grid determines how many scales of motion will be contemplated by our simulation. From equation (1.9) we see that energy flux across each wave number indeed involves both larger and smaller scales than the one in question.

In a numerical scheme, the scales of motion that are smaller than the grid spacing are not taken into account. However, if the grid spacing is at least as fine as the Kolmogorov's scale, our simulation can contemplate even the smallest vortices that are formed before energy starts dissipating as heat. The secondary problem comes along when we note that, for our grid to attain such a level of fineness, the number of points on the grid must scale with the Reynolds number as $\mathrm{Re}^{9/4}$ [20].

Considering that turbulent flows present a Reynolds number of at least $\mathrm{Re} = 10^4$, we see that a fully resolved simulation is unfeasible for most flows. In a sense, dealing with closure problems is essentially handling the fact that there are more degrees of freedom than there are equations we are able to solve. This signals that we need to model small scales somehow.

One popular approach is to filter out the smalls scales of motion through a convolution with a filtering kernel and solve the remaining equations only for the large scales [12]. This is usually referred to as Large Eddy Simulation (LES). Solving these equations, however, requires modelling the interactions between scales, including filtered and unfiltered ones. Many models were written along the years since LES was first proposed, some of them based on Bousinessq approximations, as in [71, 19], while others rely on solving an additional scalar transport equation for kinetic energy of small scales, as in [68]. More recent models started incorporating different types of neural networks and other machine learning frameworks into building closures for LES, as reported in [4, 39].

Another way of approaching this problem is to use the Reynolds decomposition of the velocity field as a time averaged velocity (as opposed to LES's spatial filtering) plus some fluctuation [64]. This is known as Reynolds Averaged Navier-Stokes (RANS), and the equation for the averaged velocity is the same as the one for the filtered velocity produced by LES, but these two velocity fields present different properties, as well as different stress tensors [31]. A recent tendency of RANS has also been to introduce machine learning techniques to improve performance and prediction of flows, as reported in [83, 42].

A series of studies, such as [43, 40, 67, 18, 10], and, more recently, [56, 47, 48, 7, 76, 15] indicate that solutions for the NSE at high Reynolds number are expected to evolve as stochastic processes triggered by a small noise felt by small scales, i.e., solutions are spontaneously stochastic. Coupled with the intent of achieving a more realistic modelling of small scales, this motivates the search for a probabilistic approach of the closure problem.

1.3 Motivation and strategy for this work

Writing a closure for Navier-Stokes is already difficult. A thorough survey was conducted in [84], where the author articulates that, even though much has advanced in the theory of fluid equilibria, and in the fundamentals of turbulence, analytical treatment of probabilistic approaches is still challenging. A time-correlated probabilistic closure for NSE would present quite the challenge.

For these reasons, we turn to shell models of turbulence. Shell models are a class of infinite dimensional dynamical systems. They attempt to model the non-linearity in spectral NSE in a friendlier, more tractable way, but at the same time retaining key aspects, such as anomalous scaling and energy cascade. Understanding the closure problem in shell models from a more fundamental point of view may help us solve the analogous problem for the NSE in the future.

When we are working with shell models the closure problem is much more straightforward. The Sabra model [24, 81], which is detailed in chapter 2, is governed by the following equations

$$\frac{du_n}{dt} = i\big(k_{n+1}u_{n+2}u^*_{n+1} - \frac{1}{2}k_n u_{n+1}u^*_{n-1} + \frac{1}{2}k_{n-1}u_{n-1}u_{n-2}\big) - \nu k_n^2 u_n + f_n \tag{2.6}$$

for $n = 1, 2, 3, \dots$ and u^* is the complex conjugate of u.

Each equation describes the velocity fluctuation $u_n \sim \delta u(l_n)$ associated with a scale of motion $l_n = 1/k_n$, for a geometric progression of $k_n = k_0\lambda^n$ with interscale factor $\lambda = 2$. As we can see, given any cut-off shell $n = s$, a closure for the Sabra model simply becomes expressions for the closure variables u_{s+1} and u_{s+2} in terms of variables $u_1, \dots, u_s$. If s is large enough, so that $1/s$ is comparable to the Kolmogorov scale, where energy dissipates, we can consider this a fully resolved model and $u_{s+1} = u_{s+2} = 0$ is a reasonable assumption. If s is in the inertial range, however, energy cascade is still at play and the viscous term is not yet strong enough, so u_{s+1} and u_{s+2} need modelling.

One important result is the work done in [8], where the authors formulate an optimal model for closing shell models, a probabilistic spatially correlated model. Their approach relies on Kolmogorov's universality hypothesis for multipliers, which are ratios between velocity increments of consecutive scales [35]. Now, the model presented in [8], despite being optimal, required approximations to be computable. One of the conclusions was that, even if the approximation required improves significantly, the closure not necessarily performs better. This raised the hypothesis that what was missing is time conditioning, by which we mean introducing a dependence of the closure variables on the flow history. This turns this reference into the first of the two foundational rocks of the present work.

Some shell models, like Sabra, present intermittency, which prevents the velocities' statistics from being universal across scales. The universality of statistics is a crucial element to guarantee correct and consistent results when choosing different cut-off shells. This brings us to the second foundational rock of this work, which is the spatio-temporal rescaling presented in [50], detailed in chapter 3. These rescaled variables, which we call $\mathcal{U}_N$, present universal statistics across scales and no intermittent behavior.

When working with rescaled variables we do not need to resort to the universality of multipliers statistics because shell velocities $\mathcal{U}_N$ already have universal statistics. We want to write probabilistic closures for these variables that also involve time conditioning. We do so by rethinking the probabilistic approach presented in [8], including how the spatial conditioning was performed, under the light of the rescaling presented in [50]. Combining these two works leaves us with a more treatable problem.

All we need now is to understand how to use these fine universal statistics for the rescaled variables to write a time-correlated closure. From data of one fully resolved simulation of the rescaled variables we can obtain estimations of the probability density functions for the closure variables (u_{s+1} and u_{s+2} in the original Sabra system, $\mathcal{U}_0$ and $\mathcal{U}_1$ in the rescaled system). These estimations will be computed as sums of weighted Gaussian components, called Gaussian Mixture Models (GMM), which easily turns the estimated PDF into a generator of new instances of data. Moreover, this allows us to generate samples conditioned to the pre-history of the rescaled variables.

The core idea is to use these new instances of data, sampled from the estimated densities, perhaps conditioned to other values, to close a reduced model that was cut off somewhere in the inertial range, and evolve it in time. We then compare moments and other statistics of the reduced model against the fully resolved model. We have found that our closures work well in a qualitative sense, but as they get more and more elaborated, their ability to accurately reproduce statistics from the fully resolved model is somewhat degraded. The reason why this happens could be some yet unknown phenomenon, but it could also be related to the fact that Gaussian Mixture Models have strong limitations for high dimensional data, which is the case of more elaborate closures.

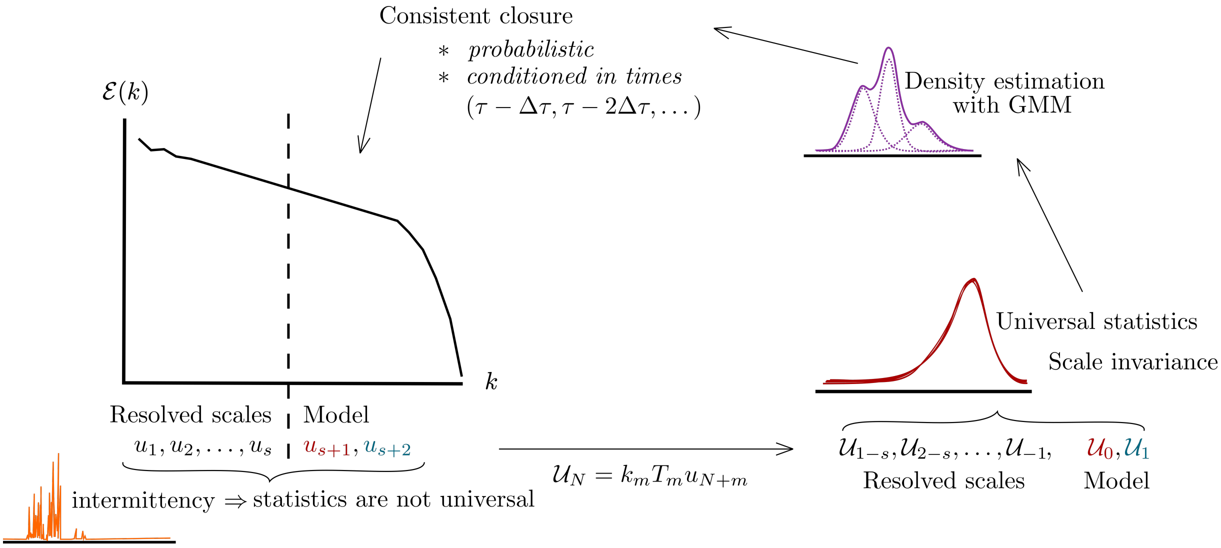

Consistent closure
$*$ *probabilistic*
$*$ *conditioned in times*
$(\tau - \Delta\tau, \tau - 2\Delta\tau, \dots)$
Density estimation with GMM
$\mathcal{E}(k)$
k
Universal statistics
Scale invariance
Resolved scales
Model
$\underbrace{u_1, u_2, \dots, u_s \mid u_{s+1}, u_{s+2}}$
intermittency ⇒ statistics are not universal
$\mathcal{U}_N = k_m T_m u_{N+m}$
$\overbrace{\mathcal{U}_{1-s}, \mathcal{U}_{2-s}, \dots, \mathcal{U}_{-1}, \quad \mathcal{U}_0, \mathcal{U}_1}$
Resolved scales
Model

Figure 1.1: A road map for the steps described above.

Figure 1.1 shows us a map of all the steps we took in the making of this work. We start with the need to model small scales in Sabra variables, but intermittency makes it so that our statistics are not universal. Going through a rescaling, however, we recover universality and a scale invariance. The PDFs from the closure variables are then estimated via a GMM, from which we sample values to evolve the reduced models, which are probabilistic and have time conditioning. Our closures are derived from fundamental properties of turbulence, where unknown functional relations are approximated from numerical statistics.

1.4 Structure

We have introduced above the Navier-Stokes equation, as well as important aspects of turbulence theory. We have also swept over our problem, its motivation and how we will approach it.

In chapter 2 we talk about the Sabra model in more detail, and in chapter 3 we explain how the rescaling of variables works and what it brings to the table.

In chapter 4 we reproduce the derivation for Gaussian Mixture Models and run some numerical tests.

Chapters 5 and 6 are numerical results of a wide variety of probabilistic, time-correlated, closures we formulated and tested against a fully resolved model.

Chapter 7 summarizes our work and offers some comments and discussions on our numerical results.

Chapter 2

Shell models

Considering the discussion about turbulence we conducted in chapter 1, we motivate and introduce shell models of turbulence. We will discuss two models in particular, their definitions and some aspects of phenomenological and numerical nature. The numerical settings we lay down in this chapter are valid for the entirety of this work.

Ideally we would like to model the non-linear term in (1.1) in such a way that it preserves most of Navier-Stokes key aspects while being dimensionally consistent, mathematically simpler and more tractable. They should also take into account that homogeneous isotropic stationary turbulence is a multiscale phenomenon. The energy cascade gives us an indication that, if we want a good description of what is happening in a turbulent flow, we must look closely at all scales, even the ones past the start of the dissipation range, as impractical as that might be.

We would also like to model the non-linearity in (1.1) in a way that small scales present intermittency. The occurrence of rare events still contributes significantly for statistics of structure functions, leading to anomalous scaling of exponents.

Many models were written along the years [14]. We would like to talk about shell models, which rely on a discretization of the phase space that considers a sequence $|\mathbf{k}| = k^n$ as a geometric progression $k_n = k_0\lambda^n$. It may help to think that each scale of motion is associated to one k_n, meaning, small n corresponds to large scales and large n corresponds to small scales. In this work, we will always have $\lambda = 2$ and $k_0 = 1$.

Let us first write a generic form for a shell model.

$$\frac{du_n}{dt} = k_n G_n(u,u) - \nu k_n^2 u_n + f_n, \tag{2.1}$$

for $n = 1, 2, 3, \ldots$. It is important to note that shell model is written for all n, so it is an infinite-dimensional system of coupled ordinary differential equations. The nonlinear coupling $G_n(u,u)$ is what ties different scales together. As in the NSE, we would like the coupling to be quadratic in order but we would like to limit the shell interactions to only the closest ones, so correlations are short ranged. Different shell models will have different nonlinear couplings. In fact, even if we add the existence of inviscid invariants like energy and helicity to the constraints mentioned above, we still can not uniquely determine $G_n(u,u)$ [6].

2.1 Sabra

Now let us look at some specific shell models, proposed by Gledzer [24] and studied in depth by Ohkitani and Yamada [81]

$$\frac{du_n}{dt} = i\big(k_{n+1}u_{n+2}u_{n+1} + bk_n u_{n+1}u_{n-1} + ck_{n-1}u_{n-1}u_{n-2}\big)^* - \nu k_n^2 u_n + f_n. \tag{2.2}$$

Equation (2.2) indeed has two inviscid invariants, namely E the energy, and H a dimensional equivalent of helicity.

$$E = \sum |u_n|^2, \tag{2.3}$$

$$H = \sum (-1)^n k_n |u_n|^2. \tag{2.4}$$

The expressions for the inviscid invariants look like this provided $b = c - 1$. Also, equations (2.2) are invariant under a phase transformation

$$u_n \to u_n e^{i\theta_n}, \tag{2.5}$$

provided all phases satisfy $\theta_{n+2} + \theta_{n+1} + \theta_n = 0$ [77]. This phase invariance, however, will imply that all quadratic forms of the type $\langle u_n, u_{n+3j}\rangle$, or $\langle u_n, u_n^*\rangle$, will have a mean value different than zero. This leads to some issues like long range correlations, but this can be improved by adjusting the complex conjugation of the non-linear part, as was done by [45], which gives us

$$\frac{du_n}{dt} = i\big(k_{n+1}u_{n+2}u^*_{n+1} + bk_n u_{n+1}u^*_{n-1} + ck_{n-1}u_{n-1}u_{n-2}\big) - \nu k_n^2 u_n + f_n. \tag{2.6}$$

Equations (2.6) are called the Sabra model and it is the main model studied in this work. It has the same two inviscid invariants as the model (2.2), energy and helicity, and the choice of parameters will be $b = -1/2$ and $c = 1/2$ from here on. Sabra also has the same invariance (2.5), but the phases are now under the constraint of $\theta_{n+2} - \theta_{n+1} - \theta_n = 0$. Now, the only quadratic form with non zero mean value is $\langle u_n, u_n^* \rangle$ [6].

The equivalent of structure function of order p, denoted $S_p(r)$ and defined in chapter 1, for shell models is

$$S_p(k_n) = \langle |u_n|^p \rangle, \tag{2.7}$$

which we also address here as moments. We can also write an energy budget analogue to equation (1.7). The energy flux due to the non linearity across a shell n is given by

$$\Pi_n = \mathrm{Im}(k_{n+1}u_{n+2}u^*_{n+1}u^*_n + \frac{1}{2}k_n u_{n+1}u^*_n u^*_{n-1}), \tag{2.8}$$

given our choice of parameters for b and c [8]. The energy dissipated by the viscous term is given by

$$D_J = \nu \sum_{n=1}^{J} k_n^2 |u_n|^2 \tag{2.9}$$

Lastly, the energy injected into the system by the forcing term is

$$F_J = \sum_{n=1}^{J} \mathrm{Re}(f_n u_n^*). \tag{2.10}$$

Then, the energy balance equation for the Sabra model is given by

$$\frac{dE_J}{dt} + \Pi_J = -D_J + F_J, \tag{2.11}$$

for $E_J = \sum_{n=1}^{J} |u_n|^2$. Equation (2.11) is the analogue to equation (1.7) we were looking for, as described in [45].

2.2 Numerical settings

Now we may be wondering how can we solve, at least numerically, something like the Sabra model. First of all, we need to decide on boundary conditions for u_0 and u_{-1}. Thinking of shell 1 as the integral scale, the natural boundary condition is to set $u_0 = u_{-1} = 0$. This only means that there is no motion going on in scales larger than the integral. Equation (2.6) also requires a viscosity and a forcing term, so let us set those at $\nu = 10^{-8}$ and $f_1 = 1 + i$, meaning the forcing term is only active in the integral scale. We also need an initial data, say, $u_n(0) = k_n^{-1/3} e^{iq_n}$ for $n = 1, 2$ and q_n being random phases. In practice, an initial simulation with this initial data is run for a long time, namely $t = 50$, until Sabra reaches stationary state, and the values of the solution at the last time step are the initial data for all simulations in this work. Our fully resolved simulation is run, from the stationary state, until $t = 50$ with time step 10^{-6}, using an Adams-Bashforth integration scheme [28], which is the same integration scheme used in all the simulations.

Computationally speaking, we need to settle for a finite quantity of shells. Given the multiscale nature of this model, we need to take into account that energy will cascade through the inertial range before reaching Kolmogorov's scale to, then, be dissipated in smaller scales. If we take into account, say, s shells, and the s-th shell is still in the inertial range, then energy is still cascading and the viscous term is not yet strong enough to dissipate it. Instead of cascading down, energy will just accumulate in the last shell available. In this scenario, not only setting $u_{s+1} = u_{s+2} = 0$ is incorrect, it will also lead to numerical instabilities. How to close the Sabra model mid-inertial range is the core of this work and will be discussed in more detail in later chapters. For now, the important message is that, for a reliable simulation, one must take into account at least enough shells to reach the dissipation range.

For our choice of viscosity, 30 shells are enough to see both the inertial and the dissipation range well represented in the energy spectrum of figure 2.1, where the inertial range starts around shell 5 and ends around shell 18, while the dissipation range covers shells 19 through 30.

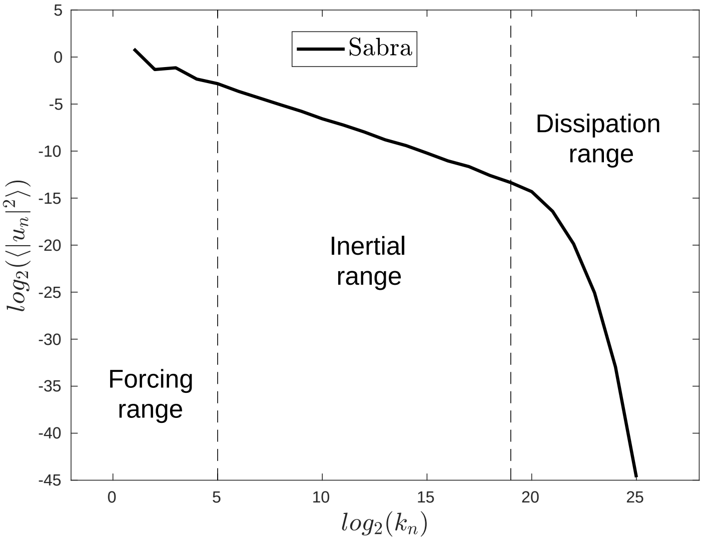


Figure 2.1: Energy spectrum for a Sabra simulation with the aforementioned parameters, until $t = 30$.

We can also see how the absolute values of the shell velocities behave in figure 2.2.

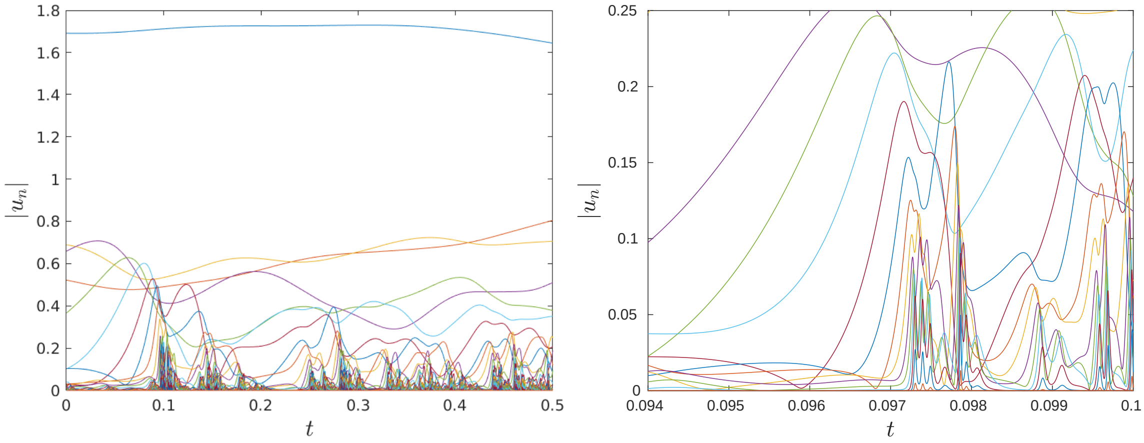


Figure 2.2: Absolute value of the time evolution of the solution for the Sabra model, with 30 shells, up to $t = 0.5$ and a zoom.

Larger scales, meaning u_1 and the closest ones, go by slowly varying. Small scales, however, present long stretches of time where not a lot happens, followed by sudden bursts of activity, as can be seen in figure 2.3. This is called intermittency. We can see in figure 2.2 that, when a large scale starts growing, so do the smaller ones, but with a slight delay in time. Due to the nonlinear coupling we are using, this increase trickles down to smaller scales, causing them to grow more and more abruptly. In [46] we see that any increase in large scales actually follows through all infinite amount of shells in finite time, so this is not some purely numerical consequence of time discretization, this is indeed an aspect of the Sabra model.

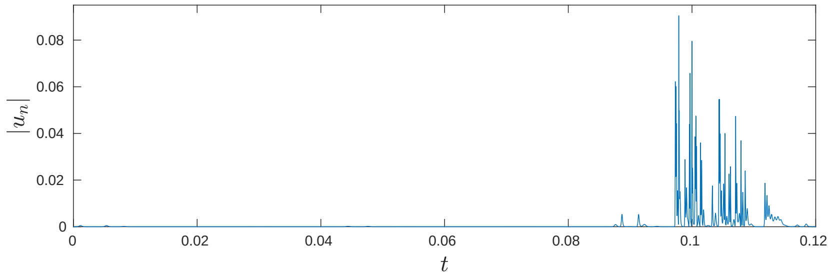
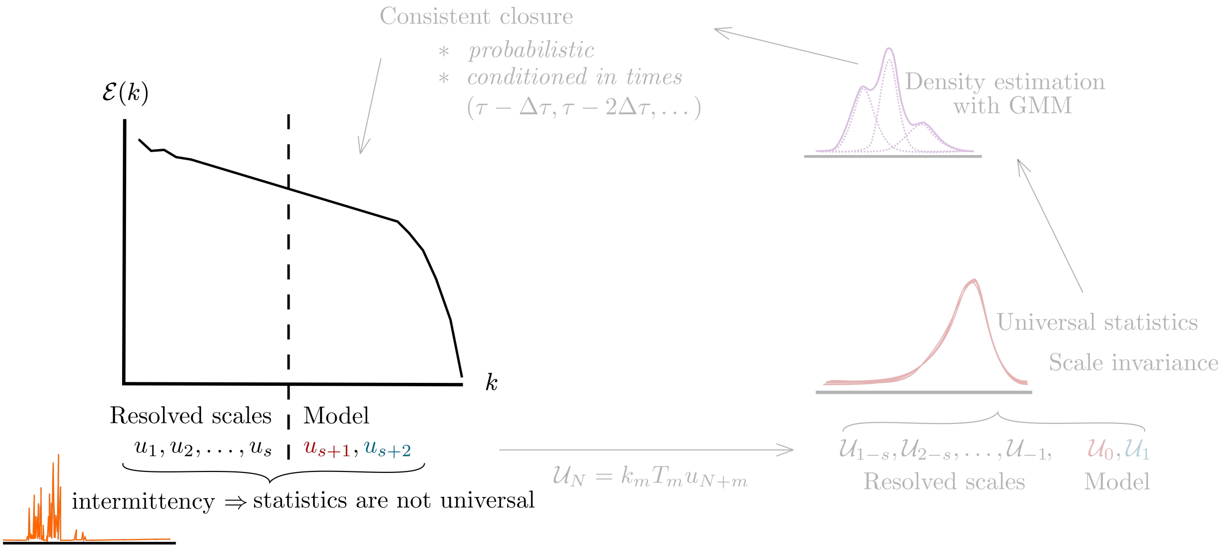


Figure 2.3: Absolute value of u_{21} up to $t = 0.12$.

2.3 In this chapter

So far we have defined the Sabra shell model and laid down the numerical settings for what is considered here a fully resolved simulation. Figure 2.4 shows our progression so far.



Consistent closure
* *probabilistic*
* *conditioned in times*
$(\tau - \Delta\tau, \tau - 2\Delta\tau, \dots)$
Density estimation with GMM
$\mathcal{E}(k)$
k
Universal statistics
Scale invariance
Resolved scales
Model
$u_1, u_2, \dots, u_s$
u_{s+1}, u_{s+2}
intermittency ⇒ statistics are not universal
$\mathcal{U}_N = k_m T_m u_{N+m}$
$\mathcal{U}_{1-s}, \mathcal{U}_{2-s}, \dots, \mathcal{U}_{-1},$
$\mathcal{U}_0, \mathcal{U}_1$
Resolved scales
Model

Figure 2.4: We are here.

Chapter 3

Rescaled model and hidden symmetries

In this chapter we will talk about how we can write a rescaled version of the Sabra model, presented in chapter 2, and the good properties of this rescaled version. We will write the rescaled system in its viscous forced version.

Intermittency is a significant point to be considered when we are talking about turbulence. It is what causes the breaking of scale invariance in NSE, and, in the case of Sabra, such scale invariance is not recovered statistically. It is also responsible for the velocity moments, as computed in equation (2.7), anomalous scaling exponents. In this situation, $S_p(k_n) \sim k_n^{-\zeta_p}$ and ζ_p has a nonlinear dependence on p. In this chapter we will see that there is a set of spatiotemporal rescaling relations which bring out symmetries that were previously inaccessible. These symmetries allow us to see intermittent dynamics as scale invariant.

The idea behind this rescaling is to adjust time scales according to the intensity of motion and this can be done for each scale. Let us start by choosing a reference shell m in the inertial range. To each reference shell we can compute a local temporal scale, which we will call T_m, at shell m and time t and has been introduced in [50] as

$$T_m(t) = \left(k_0^2 U^2 + \sum_{n<m} k_n^2 |u_n|^2 \right)^{-1/2}, \tag{3.1}$$

where U is a constant characteristic velocity. First, let us note that if $m = 1$, then $T_m = 1/k_0 U$ is constant. The terms we have before taking the $-1/2$ power can be seen as accumulated enstrophy of shells up to (but not including) m. This temporal scale T_m dynamically changes as shell m presents intermittent fluctuations.

Now let us introduce the intrinsic time at shell m, called τ, defined in [50] as

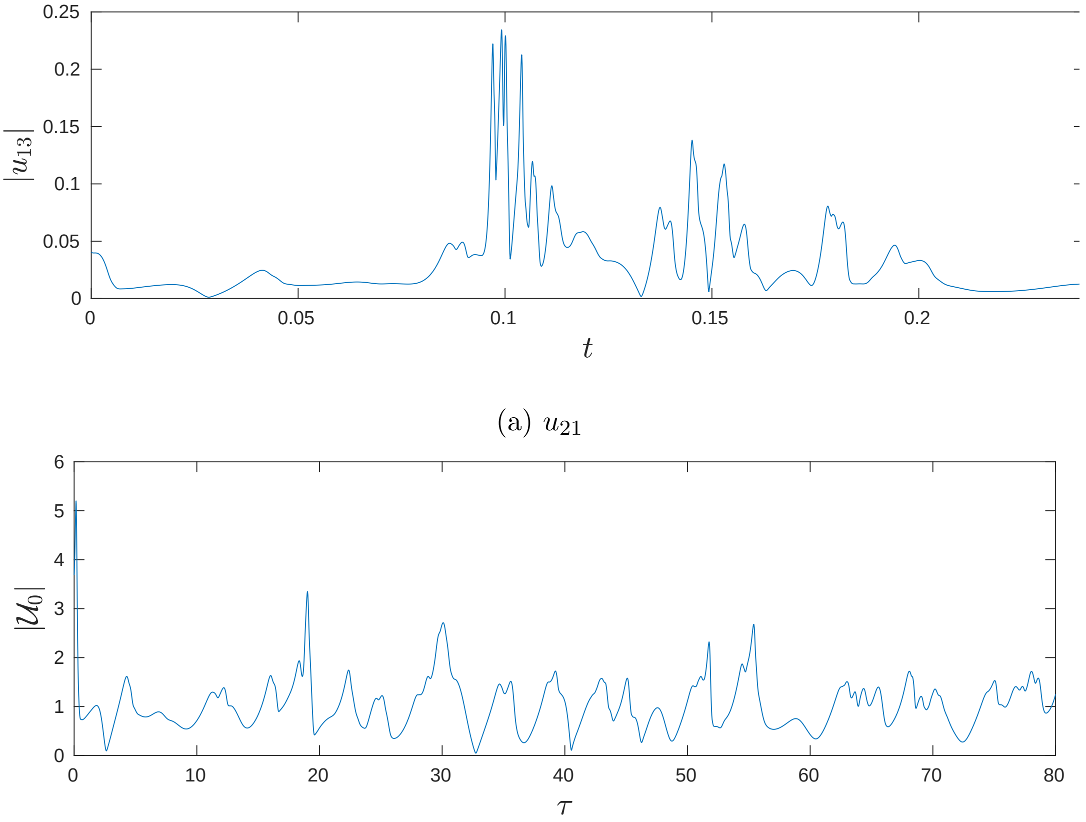
$$\tau = \int_0^t \frac{dt'}{T_m(t')}. \tag{3.2}$$

This nonlinear change from time t to time τ acts in such a way that, at shell m, long "laminar" stretches of time (where not much was happening) will be shortened, and, likewise, short "turbulent" ranges of time (where velocity was rapidly varying) will be stretched out. This notion becomes clearer when we think of time increments $d\tau$ as a rescaling of time increments dt by a factor of $1/T_m$.

We are now ready to introduce the rescaling of shell velocities. We will call the rescaled velocities $\mathcal{U}_N$, for $N > -m$, and we will do it according to the following expression.

$$\mathcal{U}_N = k_m T_m(t) u_{N+m}(t). \tag{3.3}$$

Along with equation (3.2), equation (3.3) implicitly defines $\mathcal{U}_N(\tau)$. Because of the way intrinsic times are defined, the rescaled variables vary in a quite uniform pace and with much less dramatic peaks. Also, we can compute τ and $\mathcal{U}_N$ if we have u_n. We can see one example of how a rescaled variable behaves in figure 3.1 for $m = 21$.



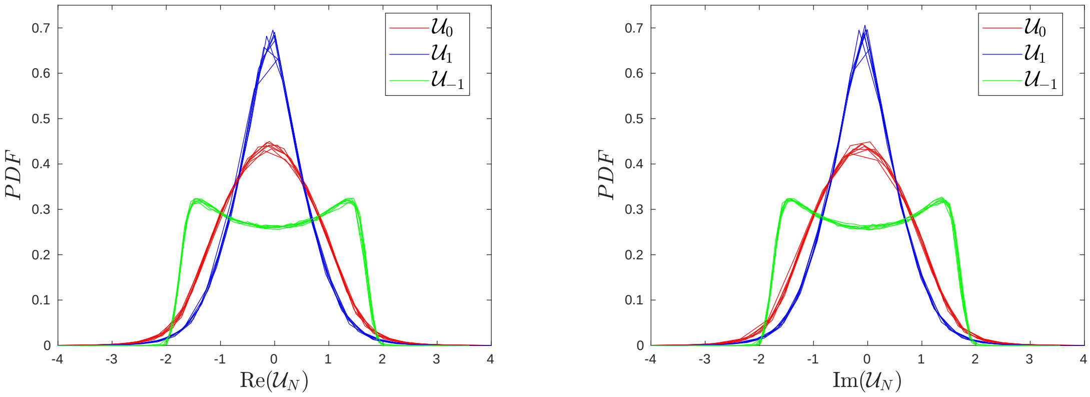
(a) u_{21}

(b) $\mathcal{U}_0$

Figure 3.1: Absolute value of u_{13} and of $\mathcal{U}_0$ with $m = 13$, the 13th shell on the rescaled system.

Now assume that u_n is known and we want to look at the rescaled variables for several different reference shells m. We want to choose an m, apply the change of variables and then look at some statistics in time. One relevant bit about the indices is to note that N starts at $1 - m$. This means that $\mathcal{U}_0$ is always the m-th shell, so, if we apply the change of variables for different m, what we are calling $\mathcal{U}_0$ will be a different shell for each m.

The statistics we would like to check are the evolution of real and imaginary parts in time, and also absolute values and phases. First, let us take a look on how the probability density functions of real and imaginary parts are, for $m = 8, \ldots, 14$ and statistics collected in time [49].



(a) Real parts

(b) Imaginary parts

Figure 3.2: PDFs for real and imaginary parts of $\mathcal{U}_{-1}, \mathcal{U}_0, \mathcal{U}_1$ and $m = 8, \ldots, 14$

To see statistics on absolute values and phases we will need a definition. Let us define, for the Sabra model, the complex multipliers

$$z_n = w_n e^{i\delta_n}, \tag{3.4}$$

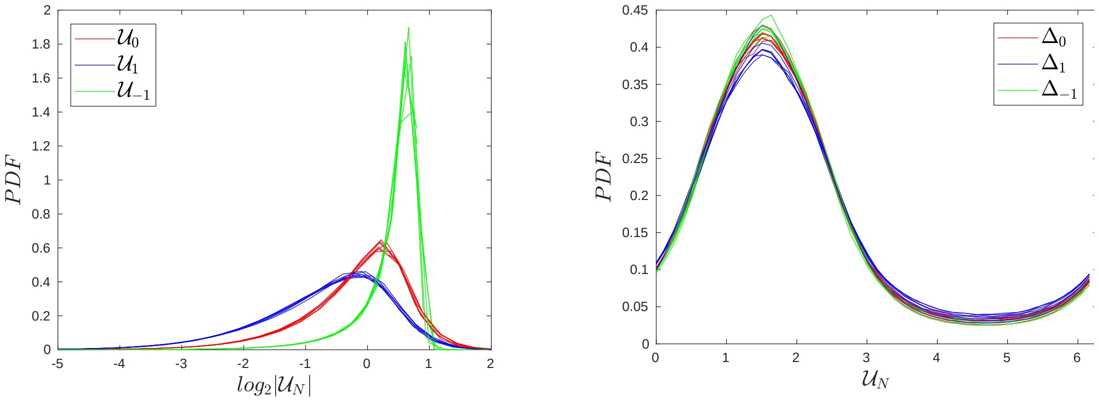
$$w_n = \left| \frac{u_n}{u_{n-1}} \right|, \tag{3.5}$$

$$\delta_n = \arg(u_n) - \arg(u_{n-1}) - \arg(u_{n-2}), \tag{3.6}$$

with the phases δ_n are chosen according to Sabra's phase invariance. Because the change of variables does not affect phases, we can write

$$\Delta_N = \arg(\mathcal{U}_N) - \arg(\mathcal{U}_{N-1}) - \arg(\mathcal{U}_{N-2}). \tag{3.7}$$

The reason why we are looking at these particular phases Δ_N will become clearer when we talk more about closures. Meanwhile, let us take a look at their PDFs for $m = 8, \ldots, 14$ and statistics collected in time.



(a) Absolute values

(b) Phases

Figure 3.3: PDFs for absolute values and phases of $\mathcal{U}_{-1}$,$\mathcal{U}_0$,$\mathcal{U}_1$ and $m = 8, \ldots, 14$

What we see in figures 3.2 and 3.3 is a scaling symmetry that has been restored in the statistical sense in the inertial range. Even though $\mathcal{U}_0$ corresponds to different scales with different choices of m, they all show the same PDFs, i.e., all the curves in red. Much of why this symmetry can be recovered comes from the fact that intermittency is now encoded in intrinsic times τ.

In [50] it was shown that the inviscid unforced version of the transformed Sabra system did not depend on m, which means that the rescaled inviscid unforced system actually has this scaling symmetry for an arbitrary choice of m. If we want to know whether the rescaled viscous forced system depends on m or not (spoiler: it does not), we need to perform the calculations.

The proposition below is one of the results of this work. It extends the analogous result from [50], where the calculations only considered the inviscid unforced system, but not the viscous forced case.

Proposition 1. *The rescaled viscous, forced system is given by*

$$\frac{d\mathcal{U}_N}{d\tau} = i(k_{N+1}\mathcal{U}_{N+2}\mathcal{U}^*_{N+1} - \frac{1}{2}k_N\mathcal{U}_{N+1}\mathcal{U}^*_{N-1} + \frac{1}{2}k_{N-1}\mathcal{U}_{N-1}\mathcal{U}_{N-2})$$
$$+ \left(\xi_{total} - \nu k^2_{N+m}T_m\right)\mathcal{U}_N + T^2_m k_m f_{N+m}, \qquad (3.8)$$

where

$$\xi_{total} = \xi + \xi_\nu + \xi_f, \tag{3.9}$$

$$\xi = \sum_{N<0} k_N^3 \operatorname{Im}\left(2\mathcal{U}_N^* \mathcal{U}_{N+1}^* \mathcal{U}_{N+2} - \frac{1}{2}\mathcal{U}_{N-1}^* \mathcal{U}_N^* \mathcal{U}_{N+1} - \frac{1}{4}\mathcal{U}_{N-1}^* \mathcal{U}_N \mathcal{U}_{N-2}^*\right), \tag{3.10}$$

$$\xi_\nu = \nu T_m k_m^2 \sum_{N<0} k_N^4 |\mathcal{U}_N|^2, \tag{3.11}$$

$$\xi_f = -T_m^2 \sum_{N<0} k_{N+m} k_N \operatorname{Re}\left(\mathcal{U}_N^* f_{N+m}\right), \tag{3.12}$$

$$T_m = \frac{1}{k_0 U}\left(1 - \sum_{N<0} k_N^2 |\mathcal{U}_N|^2\right)^{1/2}. \tag{3.13}$$

Proof. What we are looking for are the equations of motion in terms of intrinsic time τ for the rescaled variables $\mathcal{U}_N$. From the definition of τ in equation (3.2), we have

$$\frac{d\mathcal{U}_N}{d\tau} = \frac{d\mathcal{U}_N}{dt}\frac{dt}{d\tau}. \tag{3.14}$$

where the first factor can be computed deriving equation (3.3) with respect to t, finding

$$\frac{d\mathcal{U}_N}{dt} = \frac{dT_m}{dt} k_m u_{N+m} + T_m k_m \frac{du_{N+m}}{dt}. \tag{3.15}$$

and the second can be computed rewriting equation (3.2) as

$$\frac{dt}{d\tau} = T_m. \tag{3.16}$$

Equation (3.14) then becomes

$$\frac{d\mathcal{U}_N}{d\tau} = \left(\frac{dT_m}{dt} k_m u_{N+m} + T_m k_m \frac{du_{N+m}}{dt}\right) T_m. \tag{3.17}$$

To make things simpler, we will separately compute the derivative of T_m with respect to t.

$$\frac{dT_m}{dt} = -\frac{1}{2}\left(k_0^2 U^2 + \sum_{n<m} k_n^2 |u_n|^2\right)^{-3/2} \cdot \sum_{n<m} k_n^2 2 \operatorname{Re}\left(u_n^* \frac{du_n}{dt}\right). \tag{3.18}$$

Now, using equations (3.1) and (2.6), we find

$$\frac{dT_m}{dt} = -T_m^3 \sum_{n<m} k_n^2 \operatorname{Re}(u_n^*[i(k_{n+1}u_{n+2}u_{n+1}^* - \frac{1}{2}k_n u_{n+1}u_{n-1}^* + \frac{1}{2}k_{n-1}u_{n-1}u_{n-2}) - \nu k_n^2 u_n + f_n]), \tag{3.19}$$

which is the same as

$$\begin{aligned}\frac{dT_m}{dt} = -\sum_{n<m} \operatorname{Re}\Big(&i(k_n T_m u_n^* \cdot k_n T_m u_{n+2} \cdot k_{n+1} T_m u_{n+1}^* \\ &-\frac{1}{2}k_n T_m u_n^* \cdot k_n T_m u_{n+1} \cdot k_n T_m u_{n-1}^* \\ &+\frac{1}{2}k_n T_m u_n^* \cdot k_{n-1} T_m u_{n-1} \cdot k_n T_m u_{n-2}) \\ &-\nu k_n^4 T_m^3 u_n^* u_n + k_n^2 T_m^3 u_n^* f_n\Big).\end{aligned} \tag{3.20}$$

Now note that we need some k_m to appear in each parcel in order to use the change of variables defined in equation (3.3), so we write $k_n = k_{n-m}k_m$ and that leads to

$$\begin{aligned}\frac{dT_m}{dt} = -\sum_{n<m} \operatorname{Re}\Big(&i(k_{n-m}^2 k_{n+1-m} \cdot k_m T_m u_n^* \cdot k_m T_m u_{n+2} \cdot k_m T_m u_{n+1}^* \\ &-\frac{1}{2}k_{n-m}^3 \cdot k_m T_m u_n^* \cdot k_m T_m u_{n+1} \cdot k_m T_m u_{n-1}^* \\ &+\frac{1}{2}k_{n-m}^2 k_{n-1-m} \cdot k_n T_m u_n^* \cdot k_{n-1} T_m u_{n-1} \cdot k_n T_m u_{n-2}) \\ &-\nu k_n^2 k_{n-m}^2 k_m^2 T_m^3 u_n^* u_n + k_n k_{n-m} k_m T_m^3 u_n^* f_n\Big).\end{aligned} \tag{3.21}$$

We can now go ahead and do the change of variables from equation (3.3), finding

$$\begin{aligned}\frac{dT_m}{dt} = -\sum_{n<m} \operatorname{Re}\Big(&i(k_{n-m}^3 . k_1 \cdot \mathcal{U}_{n-m}^* \cdot \mathcal{U}_{n+2-m} \cdot \mathcal{U}_{n+1-m}^* \\ &-\frac{1}{2}k_{n-m}^3 \cdot \mathcal{U}_{n-m}^* \cdot \mathcal{U}_{n+1-m} \cdot \mathcal{U}_{n-1-m}^* \\ &+\frac{1}{2}k_{n-m}^3 k_{-1} \cdot \mathcal{U}_{n-m}^* \cdot \mathcal{U}_{n-1-m} \cdot \mathcal{U}_{n-2-m}) \\ &-\nu k_n^2 k_{n-m}^2 T_m |\mathcal{U}_{n-m}|^2 + k_n k_{n-m} T_m^2 \mathcal{U}_{n-m}^* f_n\Big).\end{aligned} \tag{3.22}$$

Now note that $k_1 = 2$, $k_{-1} = 1/2$ and let us re-write the indices, calling $N = n-m$ and seeing that $n < m$ is the same as $N < 0$. We then have

$$\begin{aligned}\frac{dT_m}{dt} = -\sum_{N<0} \mathrm{Re}\Big(& i(k_N^3 2 \cdot \mathcal{U}_N^* \cdot \mathcal{U}_{N+2} \cdot \mathcal{U}_{N+1}^* \\ & -\frac{1}{2}k_N^3 \cdot \mathcal{U}_N^* \cdot \mathcal{U}_{N+1} \cdot \mathcal{U}_{N-1}^* \\ & +\frac{1}{4}k_N^3 \cdot \mathcal{U}_N^* \cdot \mathcal{U}_{N-1} \cdot \mathcal{U}_{N-2}) \\ & -\nu k_{N+m}^2 k_N^2 T_m |\mathcal{U}_N|^2 + k_{N+m}k_N T_m^2 \mathcal{U}_N^* f_{N+m}\Big). \end{aligned} \tag{3.23}$$

Now see that $\mathrm{Re}(iz) = -\,\mathrm{Im}(z)$ and that $|\mathcal{U}_{n-m}|^2$ is a real number.

$$\begin{aligned}\frac{dT_m}{dt} = & \sum_{N<0} k_N^3 \,\mathrm{Im}\left(2\mathcal{U}_N^*\mathcal{U}_{N+1}^*\mathcal{U}_{N+2} - \frac{1}{2}\mathcal{U}_{N-1}^*\mathcal{U}_N^*\mathcal{U}_{N+1} - \frac{1}{4}\mathcal{U}_{N-1}^*\mathcal{U}_N\mathcal{U}_{N-2}^*\right) \\ & +\nu T_m \sum_{N<0} k_{N+m}^2 k_N^2 |\mathcal{U}_N|^2 - T_m^2 \sum_{N<0} k_{N+m}k_N \,\mathrm{Re}\left(\mathcal{U}_N^* f_{N+m}\right). \end{aligned} \tag{3.24}$$

Then we can split the sum above in several pieces, as

$$\frac{dT_m}{dt} = \xi + \xi_\nu + \xi_f = \xi_{total}, \tag{3.9}$$

$$\xi = \sum_{N<0} k_N^3 \,\mathrm{Im}\left(2\mathcal{U}_N^*\mathcal{U}_{N+1}^*\mathcal{U}_{N+2} - \frac{1}{2}\mathcal{U}_{N-1}^*\mathcal{U}_N^*\mathcal{U}_{N+1} - \frac{1}{4}\mathcal{U}_{N-1}^*\mathcal{U}_N\mathcal{U}_{N-2}^*\right), \tag{3.10}$$

$$\xi_\nu = \nu T_m k_m^2 \sum_{N<0} k_N^4 |\mathcal{U}_N|^2, \tag{3.11}$$

$$\xi_f = -T_m^2 \sum_{N<0} k_{N+m}k_N \,\mathrm{Re}\left(\mathcal{U}_N^* f_{N+m}\right). \tag{3.12}$$

We can now go back to computing the main system, plugging eqs. (3.9) to (3.12) back into (3.17).

$$\frac{d\mathcal{U}_N}{d\tau} = \left(\xi_{total} k_m u_{N+m} + T_m k_m \frac{du_{N+m}}{dt}\right) T_m \tag{3.25}$$

We then use equation (2.6), with $b = -1/2$ and $c = 1/2$, to write

$$\begin{aligned}\frac{d\mathcal{U}_N}{d\tau} = \xi_{total} T_m k_m u_{N+m} + T_m^2 k_m \Big(i(k_{N+m+1} u_{N+m+2} u^*_{N+m+1} \\ - \frac{1}{2} k_{N+m} u_{N+m+1} u^*_{N+m-1} \\ + \frac{1}{2} k_{N+m-1} u_{N+m-1} u_{N+m-2}) - \nu k^2_{N+m} u_{N+m} + f_{N+m} \Big),\end{aligned} \tag{3.26}$$

which becomes

$$\begin{aligned}\frac{d\mathcal{U}_N}{d\tau} = \xi_{total} T_m k_m u_{N+m} + i(k_{N+1} T_m k_m u_{N+m+2} T_m k_m u^*_{N+m+1} \\ - \frac{1}{2} k_N T_m k_m u_{N+m+1} T_m k_m u^*_{N+m-1} \\ + \frac{1}{2} k_{N-1} T_m k_m u_{N+m-1} T_m k_m u_{N+m-2}) \\ - \nu k^2_{N+m} T_m^2 k_m u_{N+m} + T_m^2 k_m f_{N+m}.\end{aligned} \tag{3.27}$$

Now let us do the change of variables from (3.3) again, to find

$$\begin{aligned}\frac{d\mathcal{U}_N}{d\tau} = \xi_{total} \mathcal{U}_N + i(k_{N+1} \mathcal{U}_{N+2} \mathcal{U}^*_{N+1} \\ - \frac{1}{2} k_N \mathcal{U}_{N+1} \mathcal{U}^*_{N-1} \\ + \frac{1}{2} k_{N-1} \mathcal{U}_{N-1} \mathcal{U}_{N-2}) - \nu k^2_{N+m} T_m \mathcal{U}_N + T_m^2 k_m f_{N+m}.\end{aligned} \tag{3.28}$$

Reorganizing the terms, we have

$$\begin{aligned}\frac{d\mathcal{U}_N}{d\tau} = i(k_{N+1} \mathcal{U}_{N+2} \mathcal{U}^*_{N+1} - \frac{1}{2} k_N \mathcal{U}_{N+1} \mathcal{U}^*_{N-1} + \frac{1}{2} k_{N-1} \mathcal{U}_{N-1} \mathcal{U}_{N-2}) \\ + \left(\xi_{total} - \nu k^2_{N+m} T_m\right) \mathcal{U}_N + T_m^2 k_m f_{N+m}.\end{aligned} \tag{3.8}$$

We are now only one step away from finishing this ordeal. In order to write this system with no dependence on t, we must also write T_m as a function of $\mathcal{U}$ instead of u. Recall from equation (3.1) that

$$T_m = \left(k_0^2 U^2 + \sum_{n<m} k_n^2 |u_n|^2 \right)^{-1/2}.$$

Looking at the expanded sum, we see that

$$T_m = \left(k_0^2 U^2 + k_1^2|u_1|^2 + k_2^2|u_2|^2 + \cdots + k_{m-1}^2|u_{m-1}|^2\right)^{-1/2}. \tag{3.29}$$

Now doing the change of variables defined in (3.3),

$$T_m = \left(k_0^2 U^2 + \frac{k_1^2}{k_m^2 T_m^2}|\mathcal{U}_{1-m}|^2 + \frac{k_2^2}{k_m^2 T_m^2}|\mathcal{U}_{2-m}|^2 + \cdots + \frac{k_{m-1}^2}{k_m^2 T_m^2}|\mathcal{U}_{-1}|^2\right)^{-1/2}. \tag{3.30}$$

Dividing by k_m^2 where indicated,

$$T_m = \left(k_0^2 U^2 + \frac{k_{1-m}^2}{T_m^2}|\mathcal{U}_{1-m}|^2 + \frac{k_{2-m}^2}{T_m^2}|\mathcal{U}_{2-m}|^2 + \cdots + \frac{k_{-1}^2}{T_m^2}|\mathcal{U}_{-1}|^2\right)^{-1/2}. \tag{3.31}$$

Regrouping the sum,

$$\frac{1}{T_m^2} = k_0^2 U^2 + \frac{1}{T_m^2}\sum_{N<0} k_N^2 |\mathcal{U}_N|^2. \tag{3.32}$$

It then follows that

$$1 - k_0^2 U^2 T_m^2 = \sum_{N<0} k_N^2 |\mathcal{U}_N|^2, \tag{3.33}$$

$$T_m^2 = \frac{1}{k_0^2 U^2}\left(1 - \sum_{N<0} k_N^2 |\mathcal{U}_N|^2\right), \tag{3.34}$$

$$T_m = \frac{1}{k_0 U}\left(1 - \sum_{N<0} k_N^2 |\mathcal{U}_N|^2\right)^{1/2}. \tag{3.13}$$

To summarize, we have the rescaled viscous, forced system given by

$$\begin{aligned}\frac{d\mathcal{U}_N}{d\tau} &= i(k_{N+1}\mathcal{U}_{N+2}\mathcal{U}_{N+1}^* - \frac{1}{2}k_N\mathcal{U}_{N+1}\mathcal{U}_{N-1}^* + \frac{1}{2}k_{N-1}\mathcal{U}_{N-1}\mathcal{U}_{N-2}) \\ &\quad + \left(\xi_{total} - \nu k_{N+m}^2 T_m\right)\mathcal{U}_N + T_m^2 k_m f_{N+m},\end{aligned} \tag{3.8}$$

where

$$\xi_{total} = \xi + \xi_\nu + \xi_f, \tag{3.9}$$

$$\xi = \sum_{N<0} k_N^3 \,\mathrm{Im}\left(2\mathcal{U}_N^* \mathcal{U}_{N+1}^* \mathcal{U}_{N+2} - \frac{1}{2}\mathcal{U}_{N-1}^* \mathcal{U}_N^* \mathcal{U}_{N+1} - \frac{1}{4}\mathcal{U}_{N-1}^* \mathcal{U}_N \mathcal{U}_{N-2}^*\right), \tag{3.10}$$

$$\xi_\nu = \nu T_m k_m^2 \sum_{N<0} k_N^4 |\mathcal{U}_N|^2, \tag{3.11}$$

$$\xi_f = -T_m^2 \sum_{N<0} k_{N+m} k_N \,\mathrm{Re}\left(\mathcal{U}_N^* f_{N+m}\right), \tag{3.12}$$

$$T_m = \frac{1}{k_0 U}\left(1 - \sum_{N<0} k_N^2 |\mathcal{U}_N|^2\right)^{1/2}, \tag{3.13}$$

as we wanted.

□

In this chapter we presented a rescaling of the Sabra model around a reference shell m that works to absorb intermittency, therefore restoring a quite convenient scale invariance. Finally, we saw that, from equation (3.8), the choice of reference shell m is arbitrary. Figure 3.4 shows our progress so far.

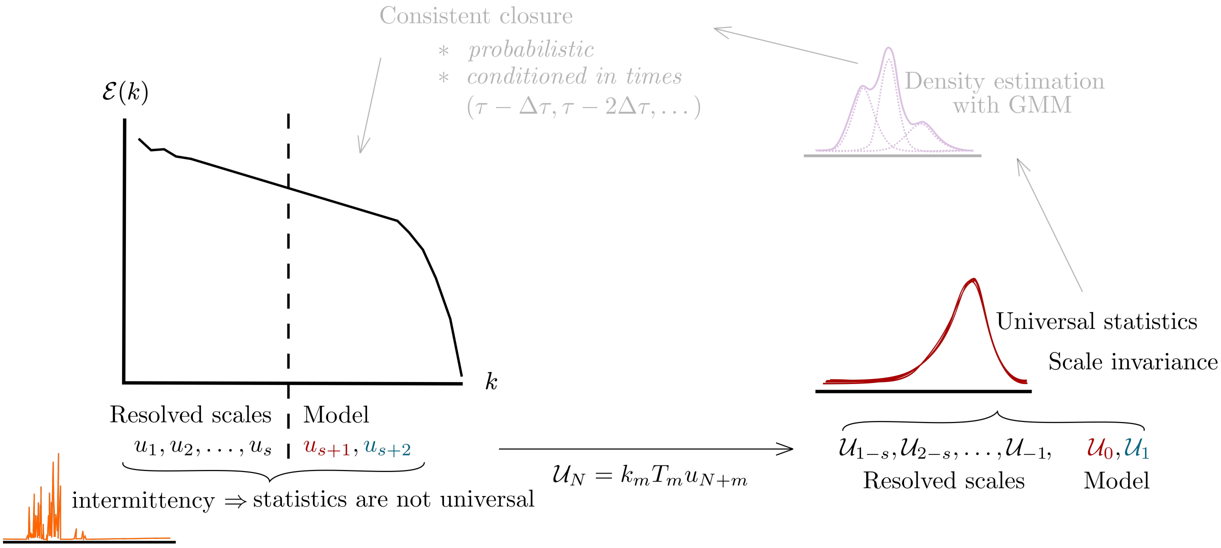

Consistent closure
∗ probabilistic
∗ conditioned in times
$(\tau - \Delta\tau, \tau - 2\Delta\tau, \dots)$
Density estimation
with GMM
$\mathcal{E}(k)$
k
Universal statistics
Scale invariance
Resolved scales
Model
$u_1, u_2, \dots, u_s$
u_{s+1}, u_{s+2}
$\mathcal{U}_N = k_m T_m u_{N+m}$
$\mathcal{U}_{1-s}, \mathcal{U}_{2-s}, \dots, \mathcal{U}_{-1},$
$\mathcal{U}_0, \mathcal{U}_1$
Resolved scales
Model
intermittency ⇒ statistics are not universal

Figure 3.4: We are here.

Chapter 4

Density estimation

In this chapter we study how to approximate probability density functions from data alone using a weighted sum of Gaussian densities. This is called a Gaussian Mixture Model (GMM) [9] and we present the derivation of its equations, as well as an iterative algorithm. We also discuss, with the aid of some examples and numerical testing, the limitations and functionalities of this model.

Density estimation consists in providing an estimate for the probability density function from which a large sample of data has been drawn. The PDF itself is inaccessible, but the data drawn from it must be widely available. The most common form of density estimation is a normalized histogram [59]. Histograms are built dividing the range of values assumed by the data in bins of width h, counting how many samples fall into each bin and then normalizing so it integrates to unity. The process of building a histogram is simple, but choosing the parameter h may be challenging. Let us look at the extreme values, say, if h has the same size of the whole range of values, we only have one bin that contains all the data samples. On the other hand, if h is small enough so that each sample sits in its own bin, we also gain no more information than we had before. There are rules of thumb for choosing h that perform better or worse in different contexts, and these will surely depend on the amount of samples on the data set [69].

One of the most popular class of density estimation methods are Kernel Density Estimators (KDE) [65, 58]. Say we want to write an estimate $\hat{f}$ for a probability density function f. Kernel methods are non-parametric and provide the following estimation

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} g\left(\frac{x - x_i}{h}\right) \tag{4.1}$$

for a data set $\{x_i\}$ of size n, and with a choice of window size h. Here, the kernel g is a positive function that integrates to unity. Different options of kernel functions, as well as different values of h, will perform better or worse depending on how the data is distributed and some computational implementations use a Gaussian kernel as default [60]. It is worth noting that KDE is a sum with as many parcels as there is data.

Gaussian Mixture Models, on the other hand, rely on choosing a number of Gaussian components and estimating their parameters as a maximum likelihood estimate. More details on how this is done are provided later in this chapter, but there is an important distinction to be made between these two methods. While KDE uses the window size h to determine how much smoothing should take place, GMM does so using the amount of Gaussian components involved in the approximation. Also, GMM has to estimate variances, weights and means to each of the Gaussian components, while KDE, being non-parametric, centers identical copies of the same kernel function around data samples. If the kernel is Gaussian, they all have variance h^2 and weights $1/(nh)$ [75].

The fact that GMM provides an approximation that is easily stored, even in high dimensions, plays an important role in the choice of this method. More importantly, the fact that the components are Gaussian makes generating new samples a very easy task. That includes generating samples that are conditioned to other values, which is a crucial aspect of the closures we write in chapter 5.

4.1 Gaussian mixture models

In this section we go into more detail as to how GMM works. First we discuss how the amount of components affects the approximation, then we write a well known latent variable approach to GMM and we end by writing an expectation-maximization algorithm for the iterative process.

We can write an approximation p to a density function as the weighted sum of $K \in \mathbb{N}$ Gaussians densities in the following manner:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4.2}$$

with

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{4.3}$$

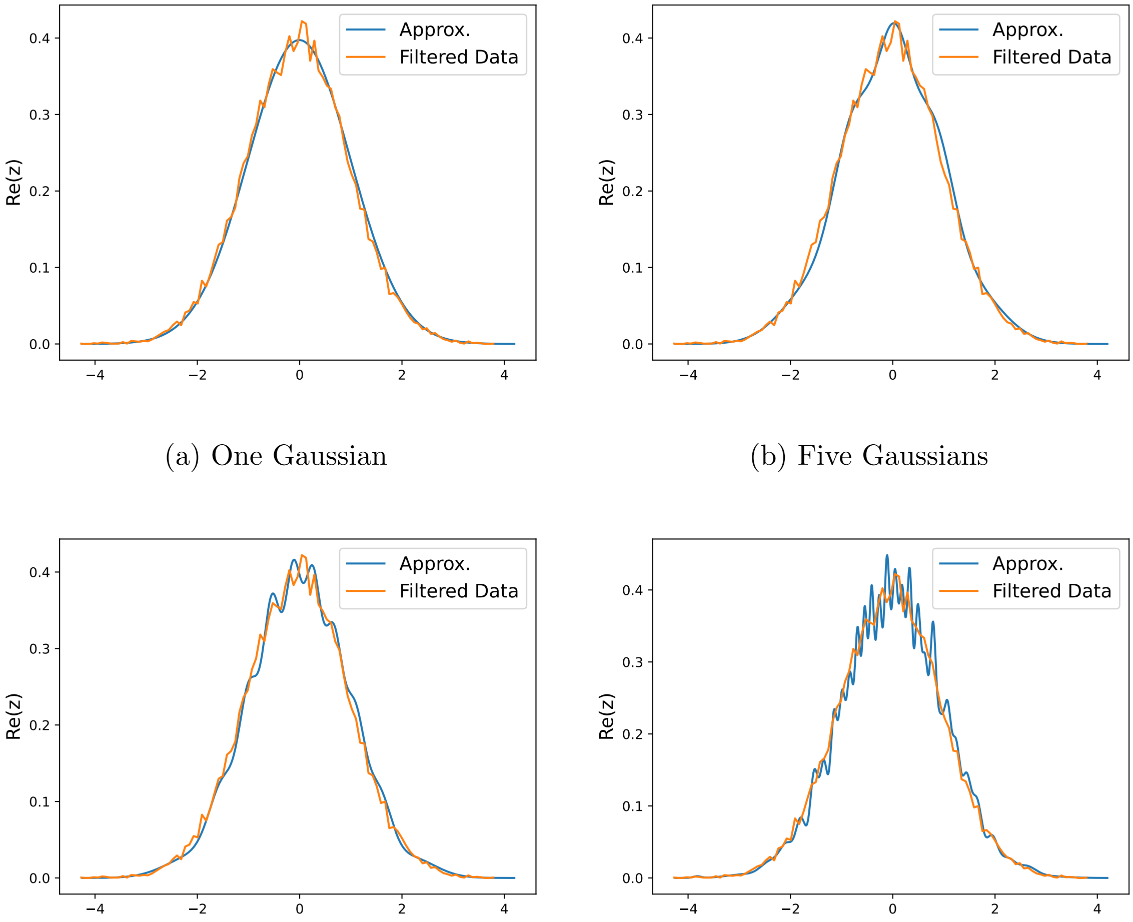
$$\sum_{n=1}^{k} \pi_k = 1. \tag{4.4}$$

Here $\mathbf{x}$ is a vector in $\mathbb{R}^d$, which means it has d entries, equation (4.3) is a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ and equation (4.4) is a restriction imposed on the mixing coefficients $\pi_k \geq 0$ to ensure that the approximation will integrate to one. The number K of Gaussian components has to be chosen beforehand and it controls how well the approximation conforms to the data set.

We can think of a somewhat artificial example but that will illustrate how the parameter K works. Let us think of a data set of 5000 real values sampled from a normal distribution $\mathcal{N}(0,1)$. With one Gaussian component, we expect it to recover the parameters of zero mean and variance one, and, indeed, a numerical experiment yields $\boldsymbol{\mu} = 0.001$ and $\boldsymbol{\Sigma} = 1.001$. This can be seen in figure 4.1a. However, as we increase the amount of Gaussian components, we can see the approximation conforming better and better to specific traits of the data set, and at the same time losing track of the general tendency of the data.

Results with several different amounts of Gaussian components can be seen in figure 4.1. This is related to the trade-off between variance and bias, which is in general connected to the complexity of the model [34, 79]. A simpler model, say, with one Gaussian component, may have higher bias for not fitting the training data set as well as, say, 10 Gaussian components, which hugs the data set very closely. On the other hand, a simpler model may do a better job at generalizing, fitting unseen data as well as it did in the training phase, while the more complex model would fit the training data extremely well but would make crass errors when confronted with unseen data.

However, since we are trying to estimate the density of the data in order to generate more samples, we do not want the estimation to capture fine details of one specific data set because this implies some absorption of noise by the approximation, which will contaminate the generated samples. Instead we want the approximation to capture the overall tendency of the data in order to generate more suitable new instances of data.



(a) One Gaussian

(b) Five Gaussians

(c) 10 Gaussians

(d) 30 Gaussians

Figure 4.1: GMM approximations for different amounts of Gaussian components

The process of density estimation and generation of new samples we do here can be broken down in three steps.

The first is to generate and treat the data we wish to mimic. Our data comes from one long simulation of the full Sabra model as in (2.6), from which we use can compute the rescaled variables as in eqs. (3.2) and (3.3).

The second step is to compute an estimation of the density of the rescaled variables. We do that by assuming the probability density function of the data can be approximated by a weighted sum of Gaussian densities.

The third step is to sample, from the estimated density, the desired values. These values can then be used in a number of manners to evolve the reduced models. The GMM allows us to perform the estimation of a joint probability density, which will later allow us to easily sample values from conditional distributions.

4.1.1 Formulation

Our goal in this method is to find parameters π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma_k}$, for $k = 1, \dots, K$, that best fit our data set and we can recall that the approximation is written as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{4.2}$$

Before we start looking at the data set, a slightly different, although equivalent, formulation will be useful. It has been written in details in, for example, [9]. Let us define a latent variable $\mathbf{z}$ that is a K-dimensional binary random variable in which only the entry z_k is equal to one and all the others are zero.

We would like to work with a joint probability distribution of $\mathbf{x}$ and $\mathbf{z}$. The introduction of the latent variable $\mathbf{z}$ works to tie $\mathbf{x}$ to the assumption we made in equation (4.2). It will be the variable that maps each of our data samples into one of the Gaussian components.

Let us first note that the marginal distribution of $\mathbf{z}$ can be written in terms of the weights π_k, as

$$p(z_k = 1) = \pi_k. \tag{4.5}$$

Because only one of the entries of $\mathbf{z}$ is equal to one and all the others are zero, we have

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{4.6}$$

and, using equation (4.2),

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{4.7}$$

This is equivalent to

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}. \tag{4.8}$$

Now, noting that $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, and that $\delta_{j,k} = 1$ if $j = k$ and 0 otherwise, we can write the marginal distribution for $\mathbf{x}$.

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) = \sum_{\mathbf{z}} \prod_{k=1}^{K} (\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))^{z_k} = \tag{4.9}$$

$$= \sum_{j=1}^{K} \prod_{k=1}^{K} (\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))^{\delta_{jk}} = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j). \tag{4.10}$$

This formulation, writing $p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$, helps us see that, for every observation $\mathbf{x}_n$ (in a data set of N samples, each one in $\mathbb{R}^d$) there is a corresponding latent variable $\mathbf{z}_n$. This follows up on the idea that, according to our assumption that the density is a sum of weighted Gaussian components, each data sample came from one Gaussian component.

This correspondence induces a natural way of sampling from the Gaussian mixture, first generating a value $\hat{\mathbf{z}}$ from the marginal distribution $p(\mathbf{z})$, and then generating a sample $\hat{\mathbf{x}}$ from $p(\mathbf{x}|\hat{\mathbf{z}})$. Generators for both these samples are widely available and this process is detailed in section 4.1.2.

It will also be convenient to compute $p(z_k = 1|\mathbf{x})$, according to Bayes rule

$$p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{z}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{k} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \gamma(z_k) \tag{4.11}$$

This quantity $\gamma(z_k)$ is sometimes called *responsibility*, in a reference to how much the k-th Gaussian component contributed to the observation $\mathbf{x}$. One can also think about π_k as a prior probability of $z_k = 1$, while $\gamma(z_k)$ corresponds to a posterior probability after $\mathbf{x}$ is observed.

Now let us assume we have a data set of samples $\mathbf{x}_n$ with N samples. We will call the data set $\mathbf{X}$ and view it as an $N \times d$ matrix, each row being one observation $\mathbf{x}_n^\top$. Similarly, we can view the latent variables as an $N \times K$ matrix $\mathbf{Z}$, each row being one $\mathbf{z}_n^\top$. For this data set, assuming all samples are independently drawn, we can write the log of the likelihood function as

$$\ln\left(p(\mathbf{X}|\boldsymbol{\theta})\right) = \ln\left(\sum_{\mathbf{z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})\right) = \sum_{n=1}^{N} \ln\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{\boldsymbol{k}}, \boldsymbol{\Sigma}_{\boldsymbol{k}})\right). \tag{4.12}$$

The general idea is to find a set of parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ that maximizes (4.12). Intuitively, we would like to differentiate equation (4.12) with respect to the parameters, set it to zero and solve for the parameters analytically. That, however, is not simple. Direct computation of these parameters is often not possible because the latent data $\mathbf{Z}$ are unobserved, so, in a sense, is like we only possessed an incomplete data set because we only observed $\mathbf{X}$. All the information we have on $\mathbf{Z}$ is given by its posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$.

Because the logarithm of the likelihood is somewhat inaccessible, given our incomplete data set, we can consider instead its expected value, given by

$$\mathcal{Q}(\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^*). \tag{4.13}$$

where $\boldsymbol{\theta}^*$ is another set of the same parameters. It is interesting to think of this in two steps; The first consists of, given an estimate $\boldsymbol{\theta}$ for the parameters, finding $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$, which can then be used to evaluate (4.13). The second is computing a new estimate set of parameters $\widehat{\boldsymbol{\theta}}$ by doing

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}^*} \mathcal{Q}(\boldsymbol{\theta}^*, \boldsymbol{\theta}). \tag{4.14}$$

The optimal set of parameters $\boldsymbol{\theta}$ can now be computed using these two steps as base for an iterative scheme, which is in fact an Expectation-Maximization (or EM) algorithm. EM algorithms are a broad class of algorithms that tend to many purposes, and eqs. (4.13) and (4.14) are already more general than we need. We will now describe these two steps in more detail for our specific context.

First of all, this iterative scheme needs an initial estimate. The Expectation step (or E step) consists of evaluating the quantity $p(z_k|\mathbf{x}_n) = \gamma(z_{nk})$ with the old (or the initial) estimate for $\boldsymbol{\theta}$, using equation (4.11), so we have

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{k} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{4.15}$$

The Maximization step (or M step) requires us to write the objective function $\mathcal{Q}(\boldsymbol{\theta}^*, \boldsymbol{\theta})$ in more details. Let us substitute (4.15) into (4.13), to find

$$\mathcal{Q}(\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} \gamma(z_{nk}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^*). \tag{4.16}$$

Now let us write the logarithm of the joint probability of $\mathbf{X}$ and $\mathbf{Z}$ as

$$\ln(p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^*)) = \ln\left(\prod_{n=1}^{N}\prod_{k=1}^{K} (\pi_k^* \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*))^{z_{nk}}\right) = \tag{4.17}$$

$$= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}(\ln \pi_k^* + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*)). \tag{4.18}$$

We can then plug (4.18) into (4.16), and recall that z_{nk} is equal to 1 only once per summation, to find

$$\mathcal{Q}(\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk})(\ln \pi_k^* + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*)). \tag{4.19}$$

We only need to clear one more element in the objective function, which is to remember that we added a constraint on the mixing coefficients π_k^* in equation (4.4). To comply with such constraint, we will use a Lagrange multiplier as follows.

$$\mathcal{Q}(\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk})(\ln \pi_k^* + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*)) + \lambda\left(\sum_{k=1}^{K} \pi_k^* - 1\right). \tag{4.20}$$

Now that our objective function is clear, we can think of ways to find optimal parameters. First we will derive $\mathcal{Q}(\boldsymbol{\theta}^*, \boldsymbol{\theta})$ with respect to $\boldsymbol{\mu}_k^*$, $\boldsymbol{\Sigma}_k^*$ and π_k^*.

For the sake of clarity, let us recall the form of the normal distribution in equation (4.3), and take its logarithm.

$$\ln(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})) = -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln(\Sigma) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})^\top \tag{4.21}$$

With equation (4.21) in mind, we can compute the derivatives.

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_k^*} = -\sum_{n=1}^{N}\gamma(z_{nk})\boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k^*), \tag{4.22}$$

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\Sigma}_k^*} = \sum_{n=1}^{N}\gamma(z_{nk})(\boldsymbol{\Sigma}_k^* - (\mathbf{x}_n - \boldsymbol{\mu}_k^*)(\mathbf{x}_n - \boldsymbol{\mu}_k^*)^\top). \tag{4.23}$$

If we set eqs. (4.22) and (4.23) to zero, we get

$$\boldsymbol{\mu}_k^* = \frac{\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n}{\sum_{n=1}^{N}\gamma(z_{nk})} \tag{4.24}$$

$$\boldsymbol{\Sigma}_k^* = \frac{\sum_{n=1}^{N}\gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k^*)(\mathbf{x}_n - \boldsymbol{\mu}_k^*)^\top}{\sum_{n=1}^{N}\gamma(z_{nk})} \tag{4.25}$$

For the weights, we have

$$\frac{\partial \mathcal{Q}}{\partial \pi_k^*} = \sum_{n=1}^{N}\frac{\gamma(z_{nk})}{\pi_k^*} + \lambda \tag{4.26}$$

If we set equation (4.26) equal to 0, we can do the following.

$$\sum_{n=1}^{N}\gamma(z_{nk}) = \lambda\pi_k^* \tag{4.27}$$

$$\sum_{k=1}^{K}\sum_{n=1}^{N}\gamma(z_{nk}) = \sum_{k=1}^{K}\lambda\pi_k^* \tag{4.28}$$

Using constraint (4.4), we have $\lambda = -N$. That gives us

$$\pi_k^* = \frac{\sum_{n=1}^{N}\gamma(z_{nk})}{N} \tag{4.29}$$

This concludes the M step. Now that we computed the maximizing parameters in eqs. (4.24), (4.25) and (4.29) we see one more reason why direct computation from (4.12) would not be an easy task, as all three expressions for the parameters depend on $\gamma(z_{nk})$, which depends on the parameters. The EM algorithm deals with this circular feedback problem by keeping $\gamma(z_{nk})$ delayed in comparision to the log-likelihood, meaning $\gamma(z_{nk})$ is always being evaluated with the old estimates while the log-likelihood always gets the new estimates.

4.1.2 The algorithm and sampling

We can now summarize the algorithm.

1. Give an initial estimate for $\boldsymbol{\theta} = \{\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

2. (The E step) Evaluate

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{k} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{4.15}$$

   using the old estimates.

3. (The M step) Use $\gamma(z_{nk})$ to compute new estimates $\widehat{\boldsymbol{\theta}}$

$$\widehat{\boldsymbol{\mu}_k} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})\mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})}, \tag{4.24}$$

$$\widehat{\boldsymbol{\Sigma}_k} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \widehat{\boldsymbol{\mu}_k})(\mathbf{x}_n - \widehat{\boldsymbol{\mu}_k})^\top}{\sum_{n=1}^{N} \gamma(z_{nk})}, \tag{4.25}$$

$$\widehat{\pi_k} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N}. \tag{4.29}$$

4. Evaluate

$$\ln\left(p(\mathbf{X}|\boldsymbol{\theta})\right) = \sum_{n=1}^{N} \ln\left(\sum_{k=1}^{K} \widehat{\pi_k}\mathcal{N}(\mathbf{x}_n|\widehat{\boldsymbol{\mu_k}}, \widehat{\boldsymbol{\Sigma_k}})\right). \tag{4.12}$$

   If convergence criterion achieved, algorithm stops. If not, $\widehat{\boldsymbol{\theta}}$ becomes the old estimate and the workflow returns to E step.

It was proven in [13] that the log-likelihood increases after the M step, unless it is already at a maximum, so the difference between the values in consecutive iterations falling below a certain threshold is a valid convergence criterion for the EM algorithm.

Furthermore, it was shown in [80] that, under the condition of the components not overlapping significantly, EM for mixture models converges at a better rate than gradient ascent.

There are options as to how initialize the parameters in step 1. Weights here are initialized as $\pi_k = 1/N$ for all k. Means and covariances can be randomly assigned, or they can be computed by a simpler and faster algorithm to provide a crude estimate, that will be refined by the GMM. Since EM converges to local maxima, different initial conditions may generate different approximations.

Now, once the EM algorithm has finished running, we are left with an approximation for the density of the data, from which we would like to draw more samples. To do so we use the weights of the approximation to choose one Gaussian component with probability π_k and then sample from the chosen component with a built-in Gaussian generator, with corresponding parameter μ_k and Σ_k. This reflects the idea that Gaussian components with heavier weights will be responsible for most of the generated samples.

Now, say we want to generate samples conditioned to another value. First, we must compute the approximation accordingly. For example, if we want to sample a quantity y_1 of dimension d_1 conditioned to another quantity y_2 of dimension d_2, we need to approximate the density of concatenated data $[y_1, y_2]$ of dimension $d_1 + d_2$.

For the sake of this argument, let us assume our approximation only has one Gaussian component. In this case we can think in terms of a concatenated random variable $Y = [y_1, y_2]$ whose density has been approximated by a normal distribution, i.e., $Y \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. It is convenient to divide the mean and the variance matrix in blocks of dimensions compatible with d_1 and d_2,

$$\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2], \tag{4.30}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}. \tag{4.31}$$

Because the conditional probability of something that has a Gaussian distribution also has a Gaussian distribution [17], we have

$$p(y_1|y_2) = \mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}). \tag{4.32}$$

Now we only need to recalculate $\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\Sigma}}$, to find

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(y_2 - \boldsymbol{\mu}_2), \tag{4.33}$$

$$\bar{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}, \tag{4.34}$$

as was also computed in [17].

Now, considering our approximation has several Gaussian components, what we do is first use the weights to select one of the Gaussian components with probability π_k. Then, once the Gaussian component is selected, we sample y_1 conditioned to y_2 by sampling from the selected Gaussian component but with mean $\bar{\boldsymbol{\mu}}_k$, and variance $\bar{\boldsymbol{\Sigma}}_k$, which are recalculated using eqs. (4.33) and (4.34).

4.2 Numerical testing

In this section we run a few numerical tests to assess how GMM performs in one and two dimensional densities. In these tests we are trying to approximate the Gumbel distribution, given in one dimension as

$$p(x) = \frac{1}{b} e^{-(x-m)/b - e^{-(x-m)/b}}, \tag{4.35}$$

with $m = 0$ and $b = 2$ [27, 26]. This distribution is asymmetric, has non zero skewness and kurtosis and decays exponentially.

Using the EM algorithm described in the previous section over 40 iterations and random initial conditions, we first perform approximations of one dimensional densities with five, 10 and 20 Gaussian components, on top of data sets of different sizes. We then do the same for two-dimensional densities. What we would like to observe in these tests is that the approximation improves as the training data set grows in size [62].

4.2.1 One-dimensional case

The data set used to compute the approximations was generated using the NumPy built in generator for the Gumbel distribution [29]. Since we know the expression of the true density p from equation (4.35), we can compare it with the approximation $\hat{p}$ by calculating the integrated square error (ISE) as

$$ISE = \int_{\mathbb{R}} ||p(x) - \hat{p}(x)||_2^2 dx. \tag{4.36}$$

In order to obtain more reliable values, for each amount of Gaussian components and for each sample size, we are generating six different data sets, computing six different approximations and evaluating six different ISEs to obtain a mean integrated square error (MISE). This is shown in figure 4.2

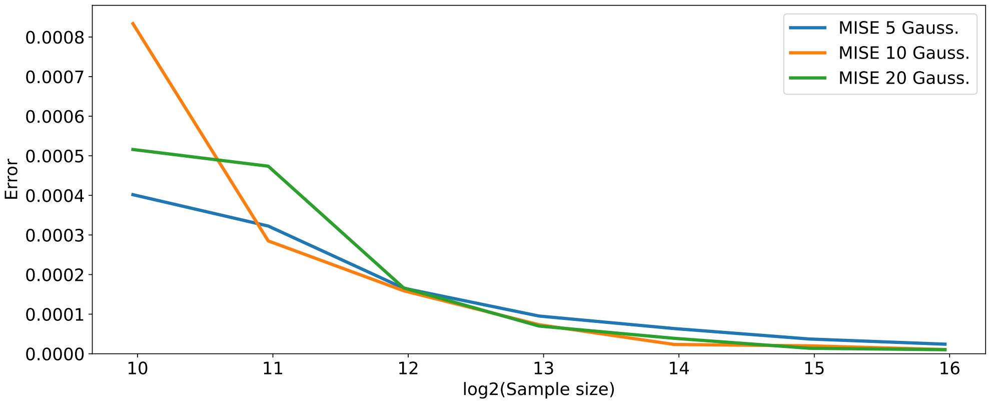


Figure 4.2: Mean integrated square errors for one-dimensional approximations using five, 10 and 20 Gaussians, for data sets of different sizes. The x-axis is in logarithmic scale.

We can also assess what the true density and the approximation look like for different sample sizes and see that, as indicated by figure 4.2, the approximation improves as the data set grows in size. We can see in figures 4.3, 4.4 and 4.5 that a lower MISE reflects on a tighter fit of the approximation with respect to the true density.

We must also consider that these approximations are going to be used to generate more data, so we need to assess the quality of the generated samples. In figure 4.6 we see the histograms of three samples compared against the true density. Each one

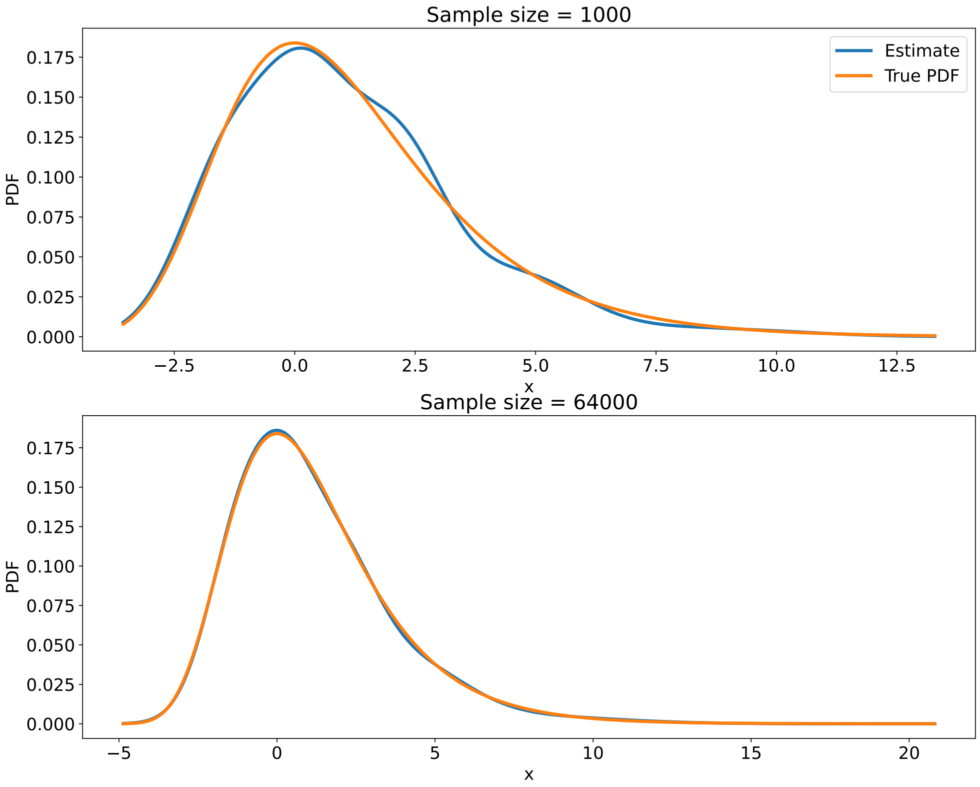


Figure 4.3: Approximation with five Gaussians computed on top of data sets of sizes 1000 and 64000

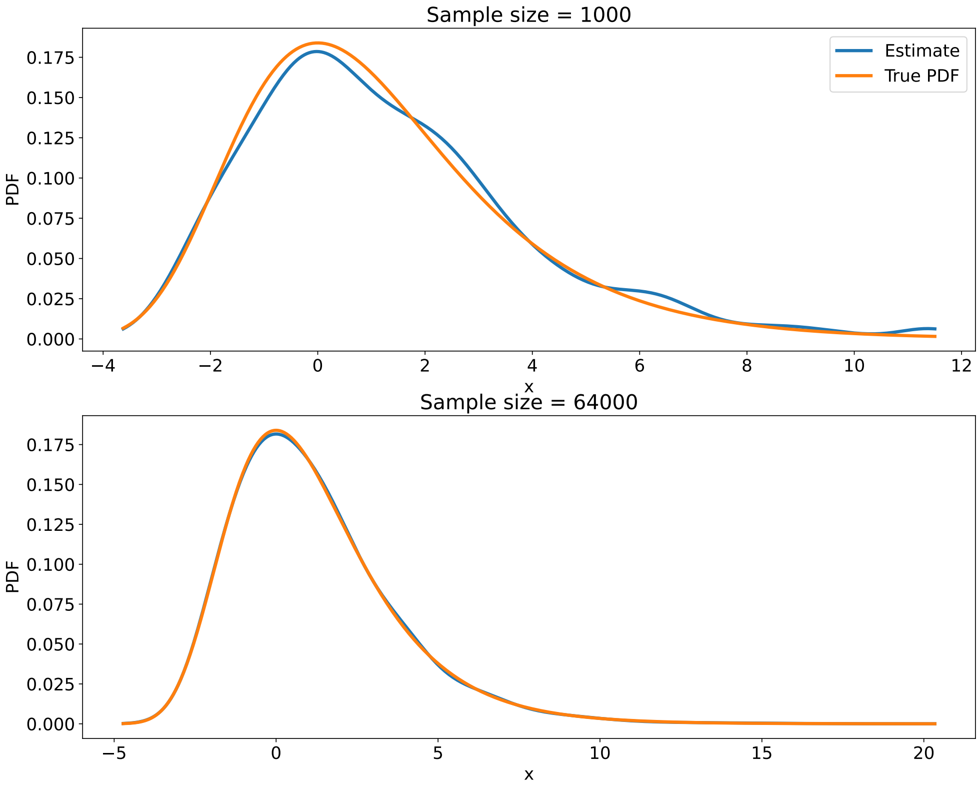


Figure 4.4: Approximation with 10 Gaussians computed on top of data sets of sizes 1000 and 64000

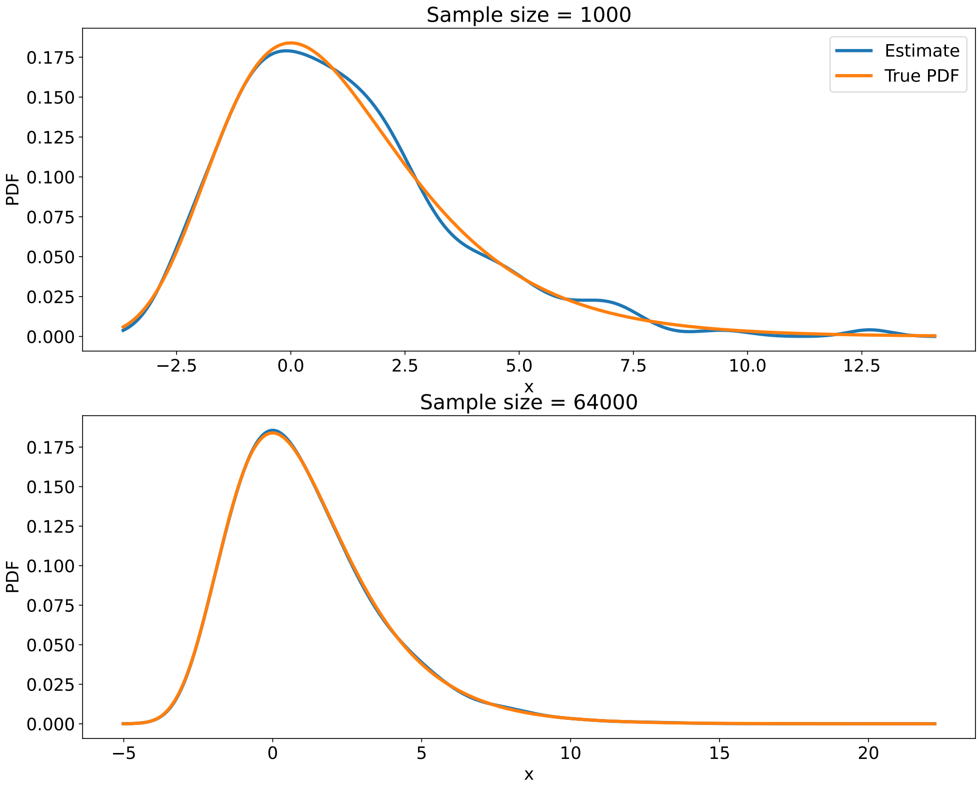


Figure 4.5: Approximation with 20 Gaussians computed on top of data sets of sizes 1000 and 64000

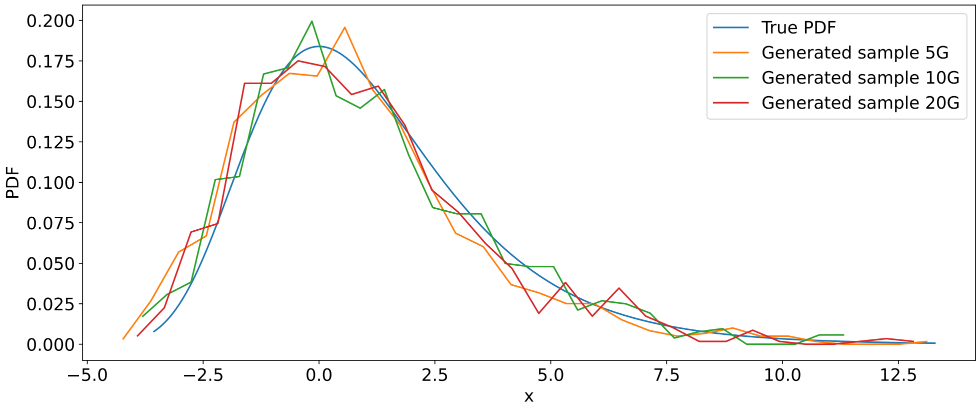


Figure 4.6: Histograms of samples generated from the 64000 points approximation, for five, 10 and 20 Gaussian components.

of these samples was generated using a different amount of Gaussian components, but the approximations that generated these samples were all computed using a data set of size 64000. The three generated samples in figure 4.6, as well as the reference sample from the Gumbel distribution, have size 1000.

In order to assess the quality of these samples we can compare them against a reference sample generated directly from the Gumbel distribution. We could compare directly to the density, since, in this case, we know the expression for the PDF, but this will not be the case in any other chapter of this work.

This comparison is made by running the Kolmogorov-Smirnov (KS) test with the implementation available in SciPy [78]. This test was developed in [72, 70] and asks whether two samples could have come from the same distribution. The test statistic D is the maximum value of the difference between the cumulative distribution functions of the samples and the p-value evaluates how likely we are to find such a value of D were the null hypothesis true. The null hypothesis in this test is that both samples came from the same distribution and can not be rejected in case of a small test statistic or a high p-value.

Specifically for the samples in figure 4.6, the one generated from the approximation with five Gaussians has a test statistic of $D = 0.031$ and a p-value of $p = 0.72$, the one generated from the approximation with 10 Gaussians has $D = 0.036$ and $p = 0.53$ and the one generated from 20 Gaussians has $D = 0.0027$ and $p = 0.86$. Under a signifi-

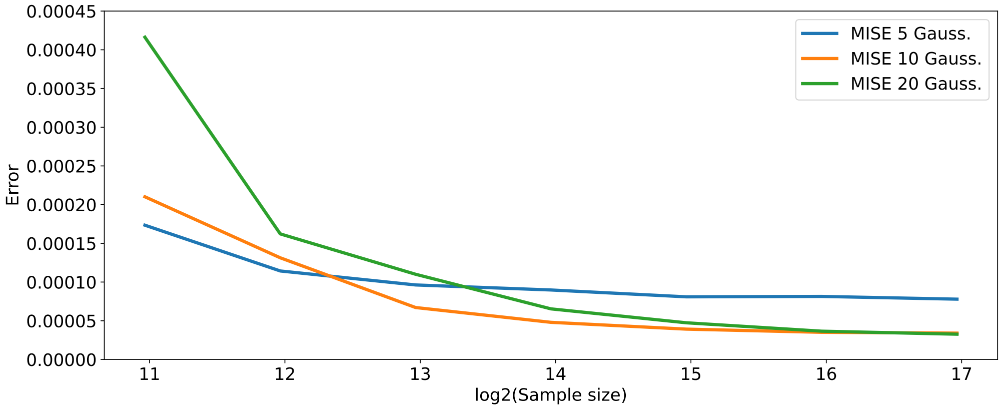


Figure 4.7: Mean integrated square errors for two-dimensional approximations using five, 10 and 20 Gaussians, for data sets of different sizes. The x-axis is in logarithmic scale.

cance level of 0.05, it is not possible to reject the hypothesis that the samples generated from the approximations and a sample generated from the Gumbel distribution came from the same distribution.

This, however, is one test performed on one sample. When we generate 1000 samples from each of the three approximations and run the KS test for all of them against the same reference sample, we see that 94.3%, 91.1% and 94.8% of the samples for the five, 10 and 20 Gaussians approximation, respectively, have a test statistic less than the significance value of 0.05. In fact, no sample produced a statistic higher than 0.08.

4.2.2 Two-dimensional case

In the two-dimensional case, each coordinate is sampled separately from the one-dimensional Gumbel generator. We are again computing approximations for data sets of different sizes and using different amounts of Gaussian components. The MISE is being computed as several instances of equation (4.36), and results can be seen in figure 4.7.

Even though visualization becomes more difficult as dimension increases, we can still try to understand what is happening with the true density and the estimation we computed. They can be seen in figures 4.11, 4.12 and 4.13.

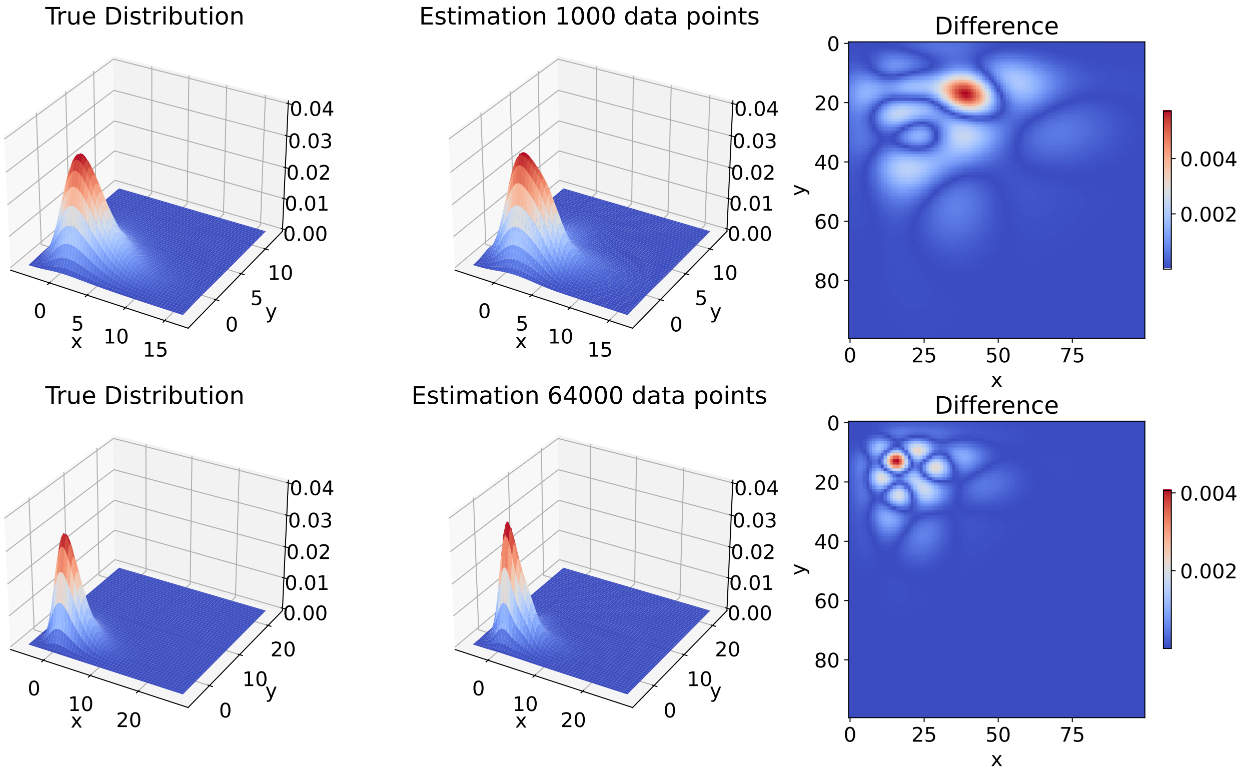


Figure 4.8: True PDFs and approximation with five Gaussians computed on top of data sets of sizes 1000 and 64000, followed by absolute error.

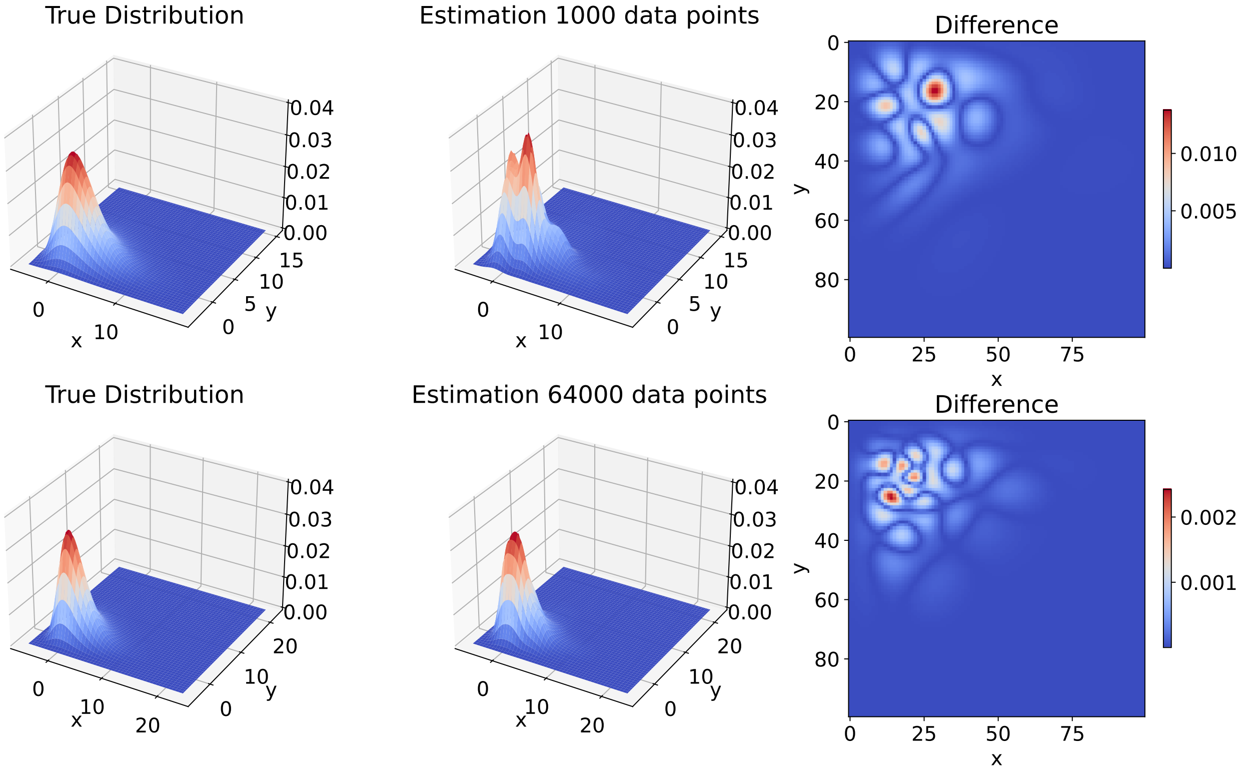


Figure 4.9: True PDFs and approximation with 10 Gaussians computed on top of data sets of sizes 1000 and 64000, followed by absolute error.

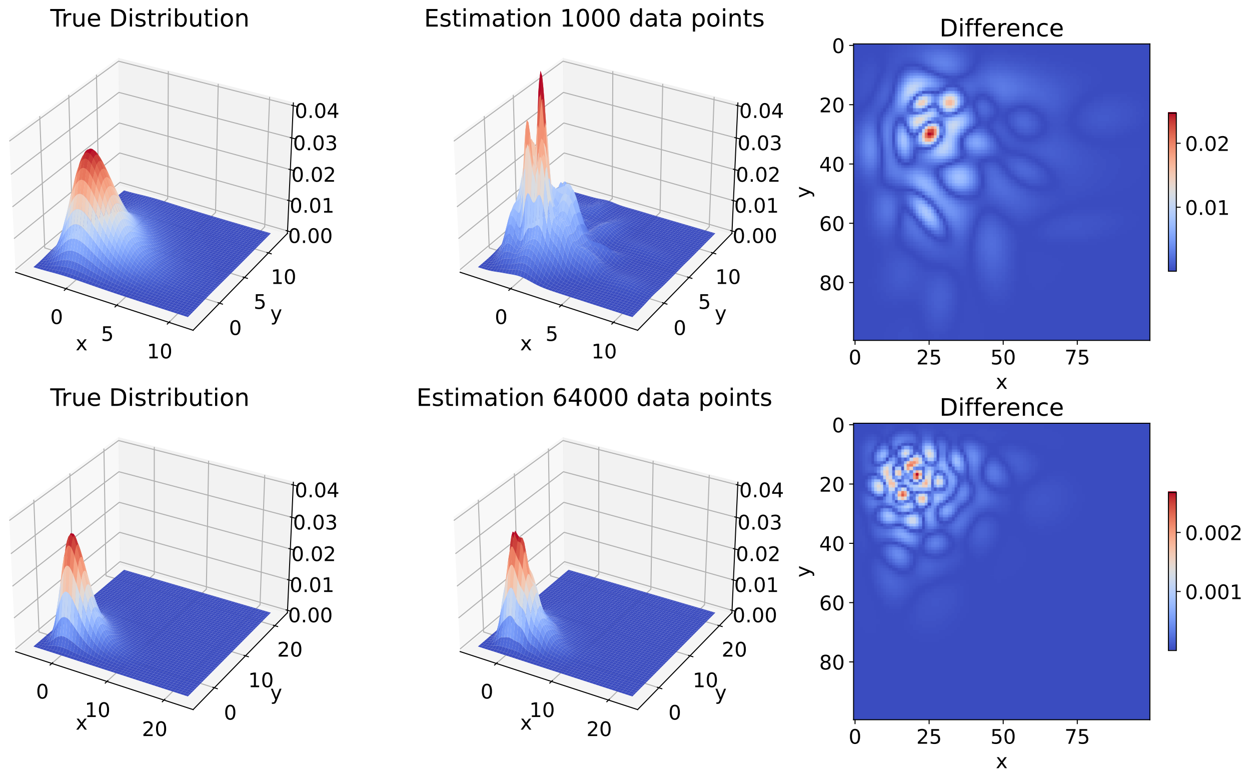
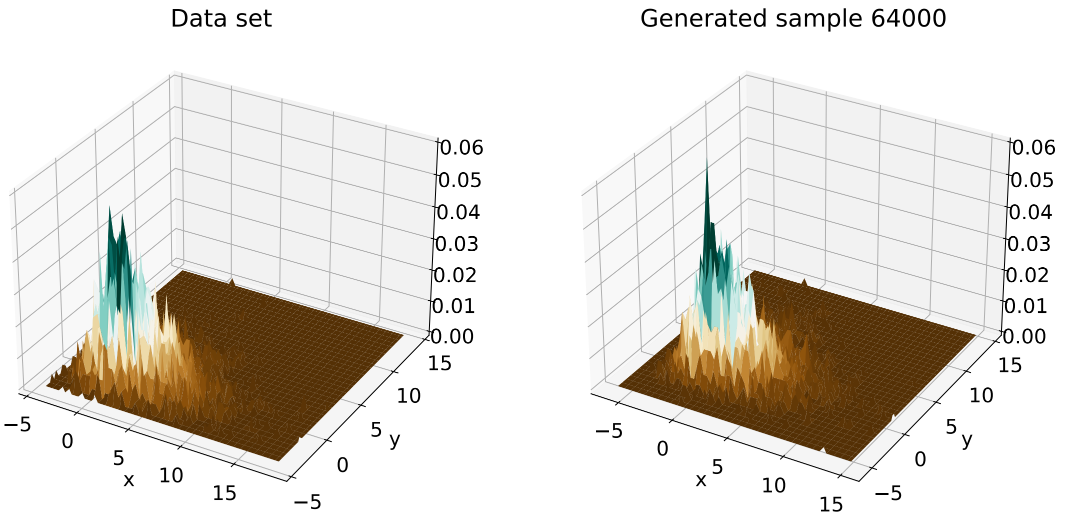


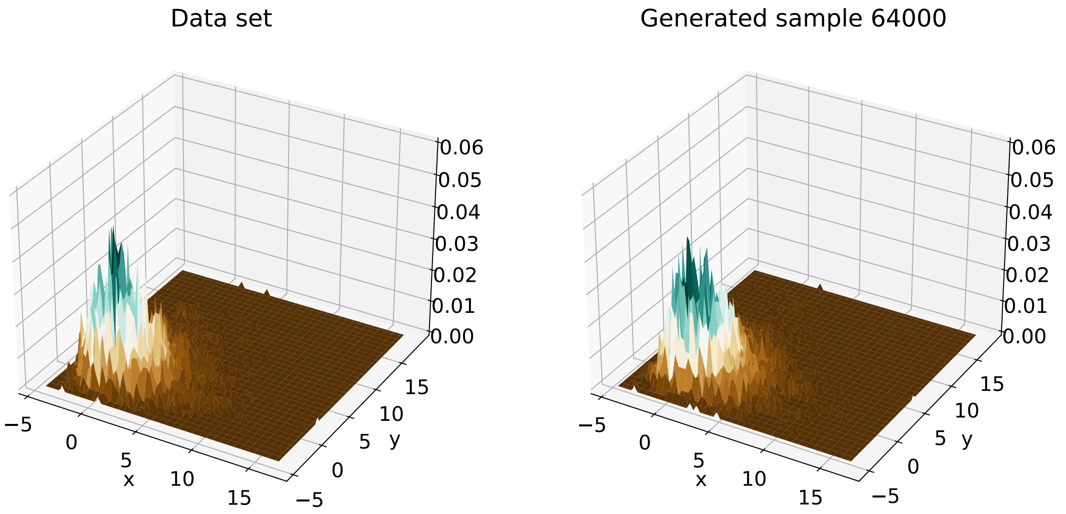
Figure 4.10: True PDFs and approximation with 20 Gaussians computed on top of data sets of sizes 1000 and 64000, followed by absolute error.

Testing the generated sample against a reference one in this case has some nuances. Because it is not clear how the points of each sample should be ordered, the distances between cumulative density functions can not be uniquely determined and the KS test is not easily applicable. What we do instead is apply the KS test to each coordinate and evaluate how they are behaving. Even though the reference sample is generated from the Gumbel distribution for each coordinate, it is not obvious that the generated sample will automatically present a similar density because it has been sampled directly from a two-dimensional Gaussian component of the approximation. The statistic and p-value of the KS test for coordinate x_1 are called D_1 and p_1, and for coordinate x_2 are called D_2 and p_2


Data set

Generated sample 64000

Figure 4.11: Histograms of the training data set and of the sample generated from the 64000 points approximation, for five Gaussian components. Test statistics are $D_1 = 0.02$, $p_1 = 0.23$, $D_2 = 0.01$, $p_2 = 0.6$.


Data set

Generated sample 64000

Figure 4.12: Histograms of the training data set and of the sample generated from the 64000 points approximation, for 10 Gaussian components. Test statistics are $D_1 = 0.01$, $p_1 = 0.66$, $D_2 = 0.02$, $p_2 = 0.34$.

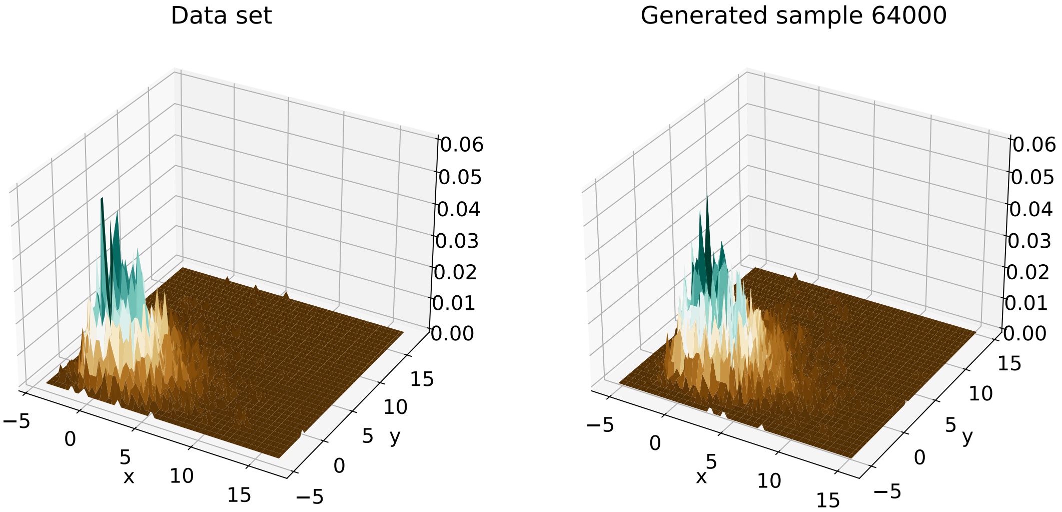


Figure 4.13: Histograms of the training data set and of the sample generated from the 64000 points approximation, for 20 Gaussian components. Test statistics are $D_1 = 0.014$, $p_1 = 0.71$, $D_2 = 0.034$, $p_2 = 0.005$.

After generating 1000 samples from each of the three approximations, with five, 10 and 20 Gaussian components, computed on top of a data set of size 64000, all the generated samples returned statistics D_1 and D_2 below the significance level of 0.05. Surely this should be taken with a grain of salt, since this testing procedure is not entirely complete, but it is still a reasonable indicator of how GMM is performing in two dimensions.

4.2.3 Dimensionality remark

We can see in figures 4.11, 4.12 and 4.13 that the approximation computed on top of a data set of size 1000 is significantly worse than the ones performed on a data set of same size in the one-dimensional case. This is closely related to the fact that Euclidean distances increase as dimension grows, which eventually causes data in high dimension to become sparse [5]. Combining this to the fact that Gaussian distributions behave unusually in high dimensions [1] we see that, even though the EM algorithm for GMM is written and formulated for any dimension, we must thread carefully when scaling up.

The question, though, seems to be how high the dimension of the data needs to be for this approach to become unfeasible. Let us generate samples of a normal distribution, of mean zero and identity variance and compute the 2-norm of these samples. We do this for Gaussians in higher and higher dimensions, then look at histograms of how these norms are distributed as dimension grows.

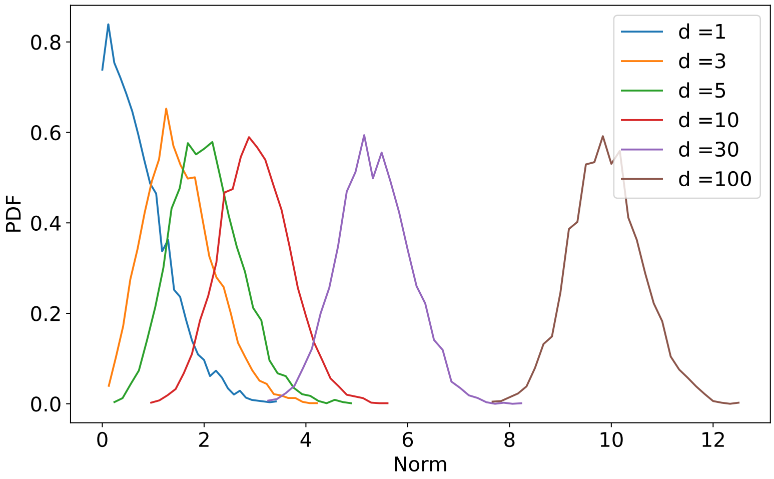


Figure 4.14: Histograms of norms of samples from Gaussian distributions of dimensions between one and 100.

From figure 4.14 we see that samples from Gaussians in 10 dimensions are already very different from samples from Gaussians in, say, two dimensions. Some of this issue can be remedied by increasing the size of the data set we are using to compute the approximation, but there is only so much RAM memory available. We will see at a later chapter that, for this work in particular, densities of dimension around 16 already present a significant challenge.

4.3 In this chapter

We studied how Gaussian Mixture Models approximate PDFs from data. We saw its formulation and studied an Expectation-Maximization algorithm that performs well under suitable conditions. With the aid of examples and extensive testing we understood that the GMM performs well in one and two dimensional scenarios. We also saw that it has limitations when it comes to high-dimensional data, but some, although not all, of it can be mitigated by increasing the size of the data set. Figure 4.15 shows our progress so far.

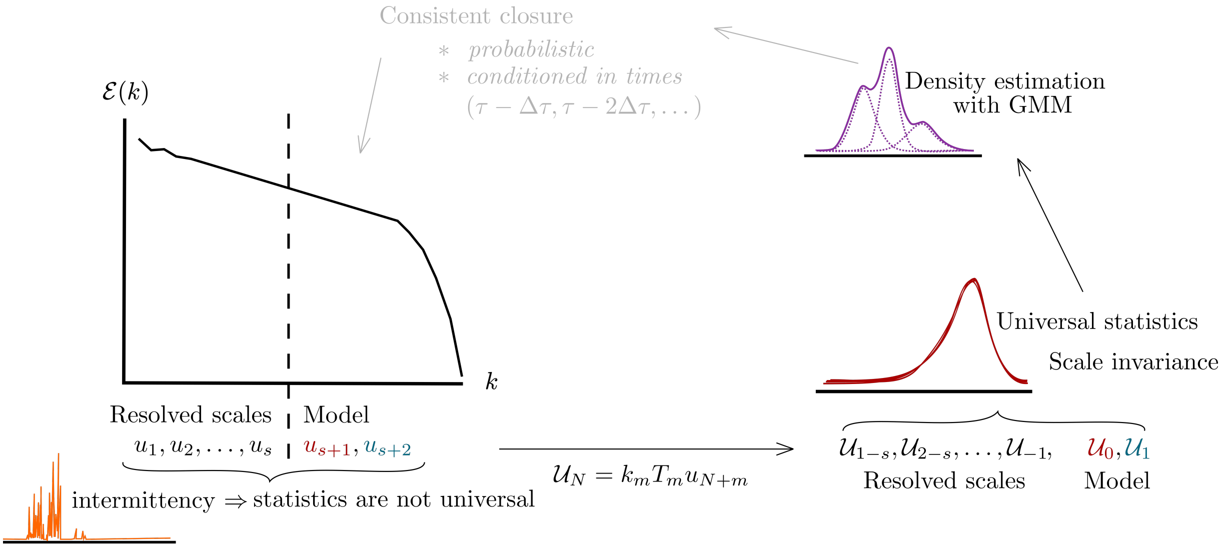


Figure 4.15: We are here.

Chapter 5

Closures with module modelling

In this chapter we formulate a wide variety of closures for the Sabra model and for the rescaled model, presented in chapters 2 and 3, respectively. We use a deterministic closure as a starting point to write our probabilistic, time correlated, closures. We also run numerical tests on how they are performing in terms of recovering the Sabra statistics, such as moments and energy flux PDFs. This chapter only contains closures that model absolute values. Phases will be treated in chapter 6.

5.1 General form of the truncated model

Recall the Sabra model, written for $n = 1, 2, \ldots,$

$$\frac{du_n}{dt} = i\big(k_{n+1}u_{n+2}u_{n+1}^* - \frac{1}{2}k_n u_{n+1}u_{n-1}^* + \frac{1}{2}k_{n-1}u_{n-1}u_{n-2}\big) - \nu k_n^2 u_n + f_n. \tag{2.6}$$

We would like to model the small scales of this system, which would allow us to compute only a finite number of shells $n = 1, \ldots, s$ with a fixed cut-off shell s in the inertial range. Because s is a shell in the inertial interval, the dissipative term's action in negligible. The reduced, but yet unclosed, Sabra system then becomes

$$\frac{du_n}{dt} = i\big(k_{n+1}u_{n+2}u_{n+1}^* - \frac{1}{2}k_n u_{n+1}u_{n-1}^* + \frac{1}{2}k_{n-1}u_{n-1}u_{n-2}\big) + f_n, \quad n = 1, \ldots, s, \tag{5.1}$$

In the context of shell models, the closure problem is reduced to finding expressions for u_{s+1} and u_{s+2}, which appear in the last two equations for $n = s-1$ and $n = s$. From equation (3.8), we can write the reduced, also yet unclosed, rescaled system as

$$
\begin{aligned}
\frac{d\mathcal{U}_N}{d\tau} &= i(k_{N+1}\mathcal{U}_{N+2}\mathcal{U}^*_{N+1} - \frac{1}{2}k_N\mathcal{U}_{N+1}\mathcal{U}^*_{N-1} + \frac{1}{2}k_{N-1}\mathcal{U}_{N-1}\mathcal{U}_{N-2}) \\
&\quad + (\xi + \xi_f)\,\mathcal{U}_N + T_m^2 k_m f_{N+m}, \quad N = -s, \dots, -1.
\end{aligned} \tag{5.2}
$$

In this expression, as we computed in chapter 3,

$$
\xi = \sum_{N<0} k_N^3 \operatorname{Im}\left(2\mathcal{U}^*_N\mathcal{U}^*_{N+1}\mathcal{U}_{N+2} - \frac{1}{2}\mathcal{U}^*_{N-1}\mathcal{U}^*_N\mathcal{U}_{N+1} - \frac{1}{4}\mathcal{U}^*_{N-1}\mathcal{U}_N\mathcal{U}^*_{N-2}\right), \tag{3.10}
$$

$$
\xi_f = -T_m^2 \sum_{N<0} k_{N+m}k_N \operatorname{Re}\left(\mathcal{U}^*_N f_{N+m}\right), \tag{3.12}
$$

$$
T_m = \frac{1}{k_0 U}\left(1 - \sum_{N<0} k_N^2 |\mathcal{U}_N|^2\right)^{1/2}. \tag{3.13}
$$

Once we have defined a cut-off shell s, we can define a value for the reference shell m. In theory any m in the inertial range is an appropriate value, but the choice of $m = s+1$ sets the temporal scale in u_{s+1} (or, equivalently, in $\mathcal{U}_0$). This leaves us with the task of computing $\mathcal{U}_0$ and $\mathcal{U}_1$ if we want to close the reduced model for the rescaled variables.

Different closures will be defined in the following sections. They are expressions for the missing variables $\mathcal{U}_0$ and $\mathcal{U}_1$ in the last two equations of (5.2). With the exception of a deterministic closure, reproduced here in section 5.2, all our closures are probabilistic and several of them are time correlated.

It is important to note, however, that even though we write all closures for the rescaled system, i.e., expressions for $\mathcal{U}_0$ and $\mathcal{U}_1$, all reduced models are actually simulated in Sabra variables but in rescaled time τ. This is done by noting that

$$
\frac{du_n}{d\tau} = \frac{du_n}{dt}\frac{dt}{d\tau} = \frac{du_n}{dt}T_m. \tag{5.3}
$$

Using equation (3.13) to write T_m, the right-hand side of (5.3) depends only on τ. Thus, in each time step, we must not only sample $\mathcal{U}_0$ and $\mathcal{U}_1$, but also use these values to compute $u_{s+1}(\tau)$ and $u_{s+2}(\tau)$. This is also convenient when it comes to computing statistics of the reduced models, since all we need to do to obtain statistics in t instead of τ is weigh each datum according to the evolution of $t(\tau)$. All simulations of reduced models are run until $\tau = 30000$ with time step 10^{-4}.

The reasons why we go through this somewhat convoluted process are two. The first is that the rescaled system, from equation (3.8), is quite stiff. In fact, it is even stiffer than Sabra and even the reduced model for the rescaled system proved to be a numerical challenge. Computing the Sabra variables in rescaled time is a much friendlier computational alternative. The second reason is that, even though Sabra variables are easier to simulate, the rescaled variables are the ones with universal statistics. The estimation of the densities we performed in chapter 4 is the same for any cut-off shell in the inertial range, meaning our closures should work without the need to re-learn such densities. This would not be possible in Sabra variables.

5.2 Kolmogorov's Closure

In this section we explore what we are calling Kolmogorov's closure. It was first presented in [8], along with other closures derived from a probabilistic approach, where it was thoroughly studied.

Let us first write it for the Sabra model. According to the Kolmogorov scaling law in equation (1.19), we see that shell velocities present a power law tail

$$u_n \sim k_n^{-1/3}. \tag{5.4}$$

For the absolute values of the multipliers, defined in eqs. (3.4) to (3.6), we have

$$\frac{|u_{s+2}|}{|u_{s+1}|} = \frac{|u_{s+1}|}{|u_s|} = \lambda^{-1/3}, \tag{5.5}$$

For the multipliers phases we have that

$$\delta_n = \arg(u_n) - \arg(u_{n-1}) - \arg(u_{n-2}) = \theta_n - \theta_{n-1} - \theta_{n-2}. \tag{5.6}$$

For $n = s+1$ and $n = s+2$ we fix the multipliers' phases in their most probable value,

$$\delta_{s+1} = \delta_{s+2} = \frac{\pi}{2}. \tag{5.7}$$

We then have two expressions for the phases of the closure variables,

$$\theta_{s+1} = \frac{\pi}{2} + \theta_s + \theta_{s-1}, \tag{5.8}$$

$$\theta_{s+2} = \frac{\pi}{2} + \theta_{s+1} + \theta_s. \tag{5.9}$$

Then, combining equations eqs. (5.5), (5.8) and (5.9), we have the Kolmogorov closure for the Sabra model, given by

$$u_{s+1} = |u_s|\lambda^{-1/3}e^{i(\frac{\pi}{2}+\theta_s+\theta_{s-1})}, \tag{5.10}$$

$$u_{s+2} = |u_{s+1}|\lambda^{-1/3}e^{i(\frac{\pi}{2}+\theta_{s+1}+\theta_s)}. \tag{5.11}$$

This is a deterministic closure written on top of phenomenological predictions. We can compute the moment of order two, as in equation (2.7)

$$S_2(k_n) = \langle |u_n|^2 \rangle, \tag{2.7}$$

for both Sabra and the closed model, displayed in logarithmic scale in figure 5.1, and see that it performs well in recovering this statistic.

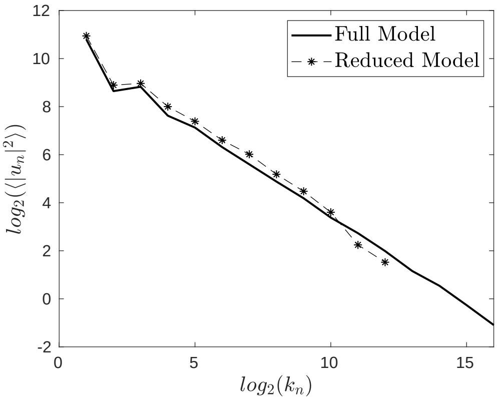


Figure 5.1: Sabra full model with $\nu = 10^{-8}$, 30 shells and a forcing on the first shell and a reduced model with Kolmogorov closure with $s = 12$

We can write an equivalent of this closure for the rescaled system. By multiplying eqs. (5.10) and (5.11) by $k_m T_m$ and applying the change of variables defined in eqs. (3.2) and (3.3), we find

$$k_m T_m u_{s+1} = k_m T_m |u_s| \lambda^{-1/3} e^{i(\frac{\pi}{2}+\theta_s+\theta_{s-1})}, \tag{5.12}$$

$$k_m T_m u_{s+2} = k_m T_m |u_{s+1}| \lambda^{-1/3} e^{i(\frac{\pi}{2}+\theta_{s+1}+\theta_s)}. \tag{5.13}$$

Because $k_m T_m$ is real and positive, it does not affect the phases of the shell velocities after going through the change of variables. Because we chose $m = s+1$, we have

$$\mathcal{U}_0 = |\mathcal{U}_{-1}| \lambda^{-1/3} e^{i(\frac{\pi}{2}+\alpha_{-1}+\alpha_{-2})}, \tag{5.14}$$

$$\mathcal{U}_1 = |\mathcal{U}_0| \lambda^{-1/3} e^{i(\frac{\pi}{2}+\alpha_0+\alpha_{-1})}, \tag{5.15}$$

where $\alpha_N = \arg(\mathcal{U}_N) = \theta_{N+m}$. We can see in figure 5.2 that this also performs well in recovering moment of order two.

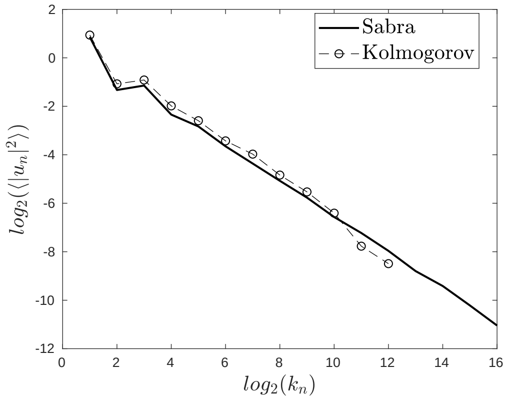


Figure 5.2: Rescaled variables computed Kolmogorov's closure with $\nu = 10^{-8}$, 30 shells and a forcing on the first shell and a reduced model with Kolmogorov closure with $s = 12$ and $m = s+1$

We can see that at the cut-off shell and the previous one, the energy spectrum has a slight dip. This is related to the fact that we kept phases of the closure variables fixed at $\pi/2$. Let us recall the equation for energy flux through a scale,

$$\Pi_n = \mathrm{Im}(k_{n+1} u_{n+2} u^*_{n+1} u^*_n + \frac{1}{2} k_n u_{n+1} u^*_n u^*_{n-1}). \tag{2.8}$$

Using the definition of δ_n, we have

$$\Pi_n = k_{n+1}|u_{n+2}u_{n+1}u_n|\sin(\delta_{n+2}) + \frac{1}{2}k_n|u_{n+1}u_n u_{n-1}|\sin(\delta_{n+1}). \tag{5.16}$$

From equation (5.16) we can see that for $\delta_{s+1}, \delta_{s+2} \in [\frac{\pi}{2}, \frac{3\pi}{2}]$, we have $\Pi_s > 0$. This means that any closure that keeps phases fixed at values from this interval will be strictly dissipative. On one hand, this guarantees that energy will be dissipated, which is necessary. On the other hand, as we shall see in this chapter, the Sabra dynamics involves negative flux in all inertial range scales, so strictly dissipative closures are somewhat lacking. The slight dipping in the energy spectrum we saw for Kolmogorov closure in figures 5.1 and 5.2 is a recurrent feature in all closures of this chapter. A more detailed modelling of phases is done in chapter 6.

This closure, despite working well, is deterministic and built on phenomenological predictions. Also, given the probabilistic nature of turbulent flows, we are more interested in understanding how a closure behaves when it ceases to be deterministic.

We will first only model the absolute values of $\mathcal{U}_0$ and $\mathcal{U}_1$ as a manner of understanding how these aspects work individually, while also building up to a more complete closure.

By running the Sabra model with 30 shells from a stationary state until $t = 50$ with time step $dt = 10^{-6}$, we compute a fully resolved set of the variables $u_n(t)$. Applying the change of variables we can compute $\mathcal{U}_N(\tau)$. We select the rescaled variables in which we are interested and feed them to the Gaussian Mixture Model to have their density estimated. Then we sample the required absolute values (and, in the next chapter, phases) from the estimated densities. The approximations for the densities of all the closures that only involve modelling absolute values are weighted sums of three Gaussian distribution built on 40000 data samples.

5.3 “Half” closure

Let us start by introducing a probabilistic component in one of the equations for the closure, while maintaining the other one as Kolmogorov’s closure. We can write the closure as,

$$\mathcal{U}_0 = 2^{z_0} e^{i(\frac{\pi}{2}+\alpha_{-1}+\alpha_{-2})}, \tag{5.17}$$

$$\mathcal{U}_1 = |\mathcal{U}_0| \lambda^{-1/3} e^{i(\frac{\pi}{2}+\alpha_0+\alpha_{-1})}, \tag{5.18}$$

$$z_0 \sim f. \tag{5.19}$$

Here, z_0 is a random variable with probability density function F. Because we do not have access to the actual density F, if we want to run a numerical simulation of the reduced system that uses the closure written in eqs. (5.17) to (5.19), at each time step we need to sample a new value of z_0 from an approximation f instead. This approximation was computed using GMM, as described in chapter 4.

All numerical simulations presented for reduced models, including the ones in chapter 6, ran until $\tau = 30000$ with time step $d\tau = 10^{-4}$. We have computed the moments of order between two and six in figure 5.3.

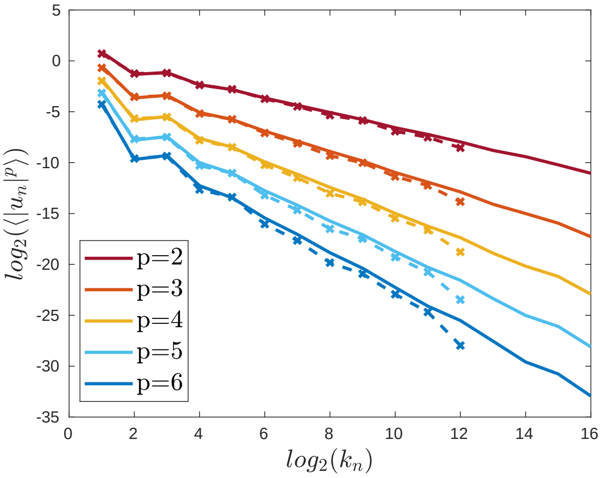
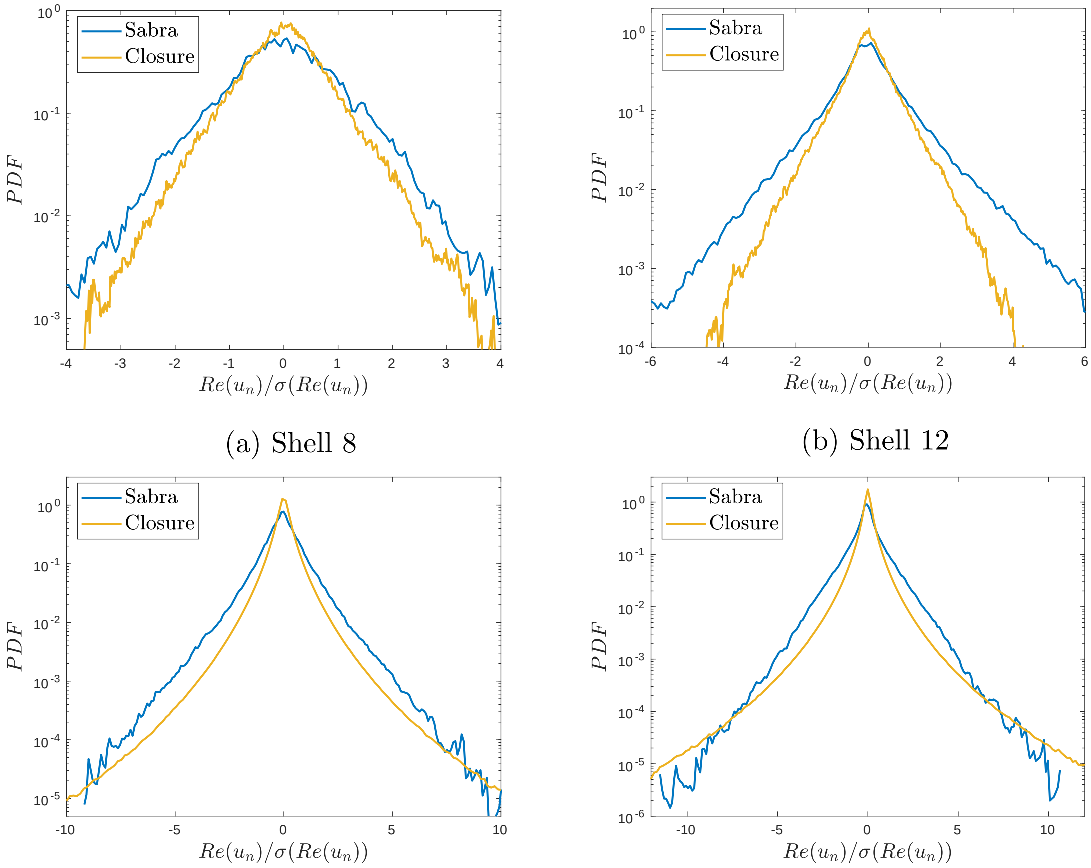


Figure 5.3: Half closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure. Cut-off shell is $s = 12$.

It is also important to evaluate if the closure is succeeding in recovering other aspects, like, for instance, PDFs of real parts of different shells. In particular, we would like to see if the closure is accurately recovering the statistics of $\mathcal{U}_0$ and $\mathcal{U}_1$. In this case, with the cut-off shell set at $s = 12$, $\mathcal{U}_0$ and $\mathcal{U}_1$ are the 13th and 14th shells, respectively. These are computed in figure 5.4.



(a) Shell 8

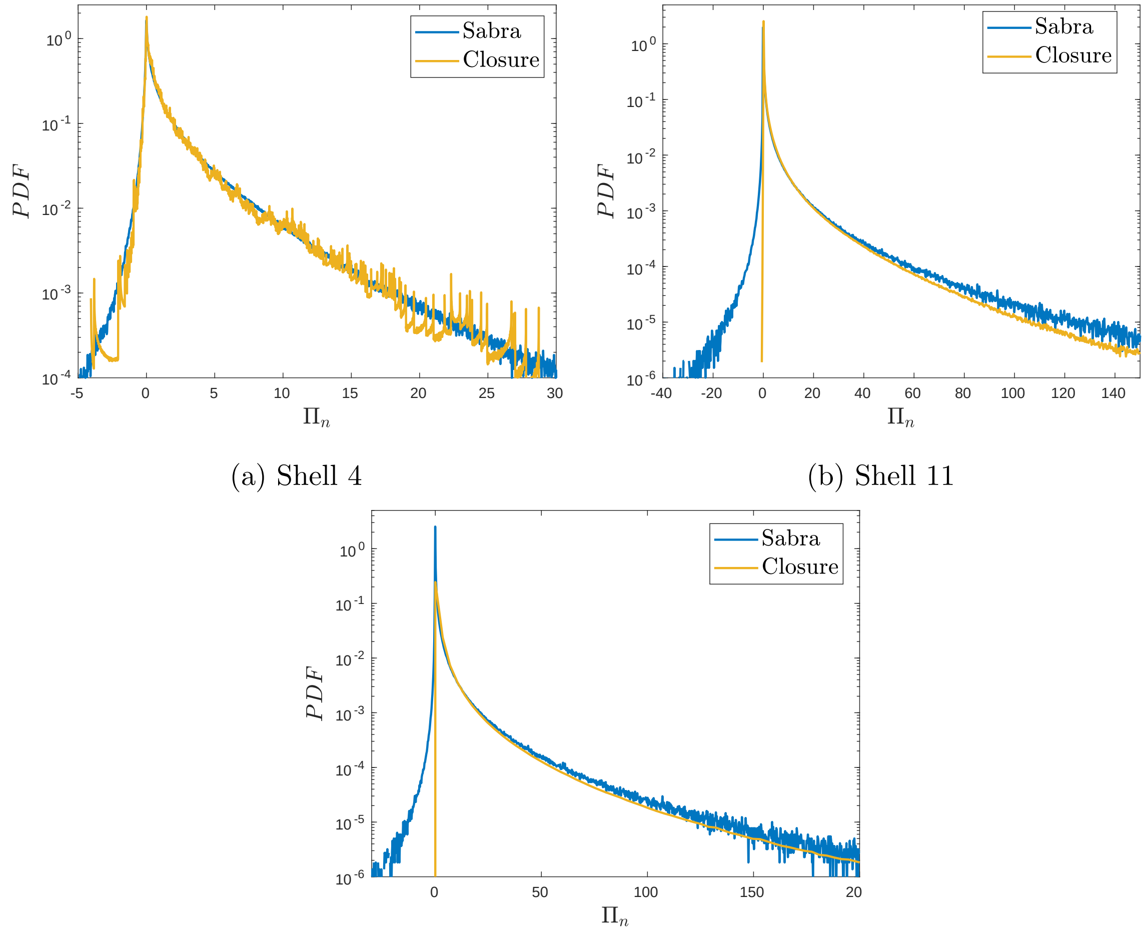
(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 5.4: Half closure: normalized PDFs of real part of different shells, including the ones computed by the closure.

How energy is flowing across scales is a key aspect in how a closure is performing. It is vital that energy dissipates correctly because the viscous term, which would naturally dissipate energy, is not active in the inertial range. We can also compute energy flow across scales according to equation (2.8), and the results can be seen in figure 5.5.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 5.5: Half closure: PDFs of energy flux across different shells.

5.4 Joint Probability Closure

We now intend to write a closure where we can jointly draw values for $|\mathcal{U}_0|$ and $|\mathcal{U}_1|$, instead of just the first one. To do so, we feed to the density estimator a data set consisting of pairs $(\log_2 |\mathcal{U}_0|, \log_2 |\mathcal{U}_1|) = (z_0, z_1)$. We estimate this joint density using three two-dimensional Gaussian components. Then, once we have an estimation f for the density of the random vector $z = (z_0, z_1)$, for each time step we can sample one z and use it to evolve the reduced model in time. This closure is written as follows.

$$\mathcal{U}_0 = 2^{z_0} e^{i(\frac{\pi}{2}+\alpha_{-1}+\alpha_{-2})}, \tag{5.20}$$

$$\mathcal{U}_1 = 2^{z_1} e^{i(\frac{\pi}{2}+\alpha_0+\alpha_{-1})}, \tag{5.21}$$

$$z = (z_0, z_1) \sim f. \tag{5.22}$$

It is worth noting that this is not the same as doing the half closure in eqs. (5.17) to (5.19) for $\mathcal{U}_0$ and $\mathcal{U}_1$ separately. Moments of order between two and six can be seen in figure 5.6.

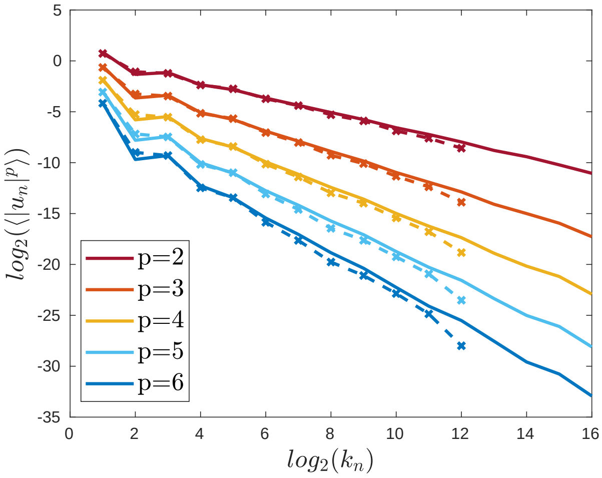
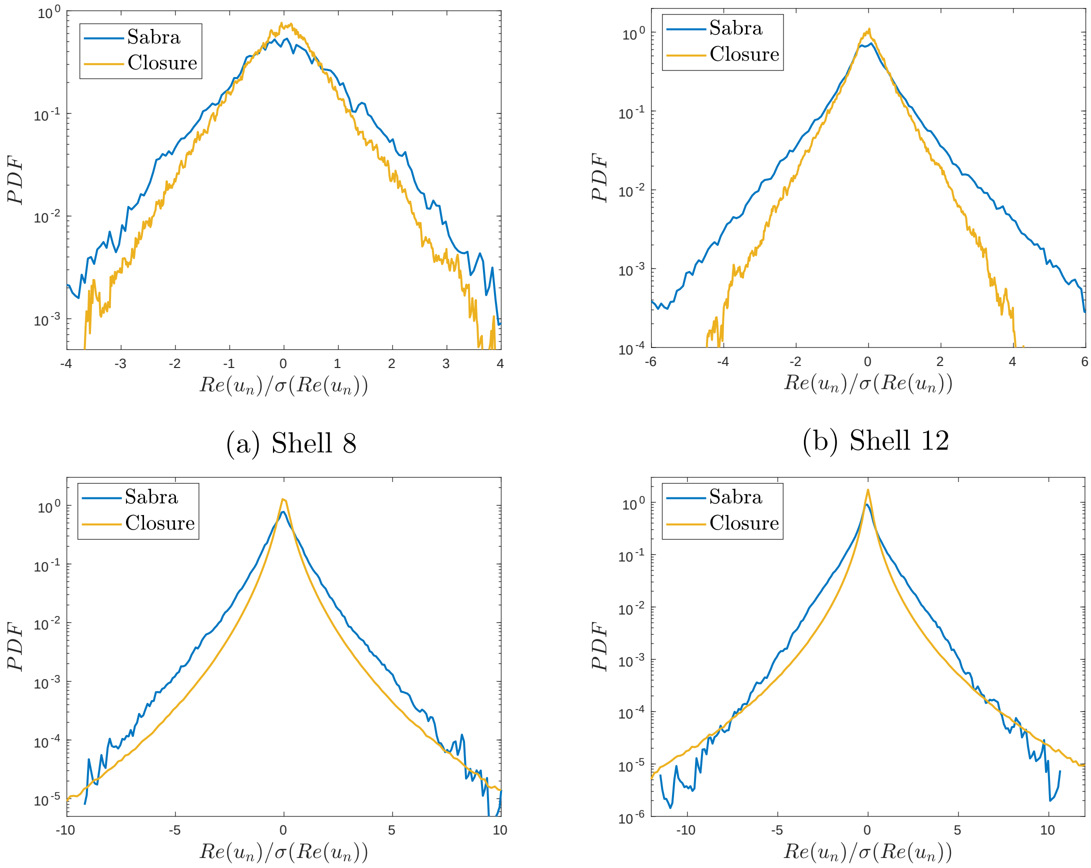


Figure 5.6: Joint probability closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure. Cut-off shell is $s = 12$.

We can also see how this closure performs in recovering statistics for real part of different shells. Figure 5.7 includes the shells computed by the closure, as well as two other shells.



(a) Shell 8

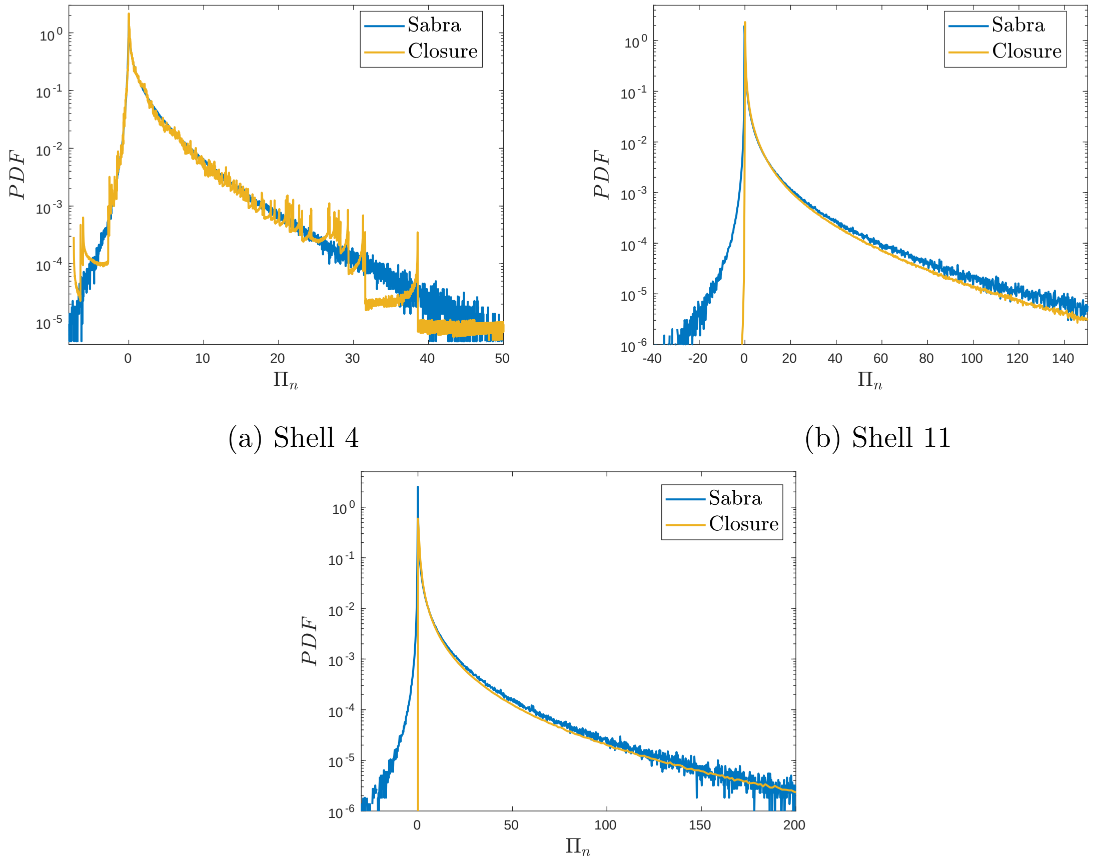
(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 5.7: Joint probability closure: normalized PDFs of real part of different shells, including the ones computed by the closure.

In figure 5.8 we can see PDFs for energy flux across scales.



(a) Shell 4

(b) Shell 11

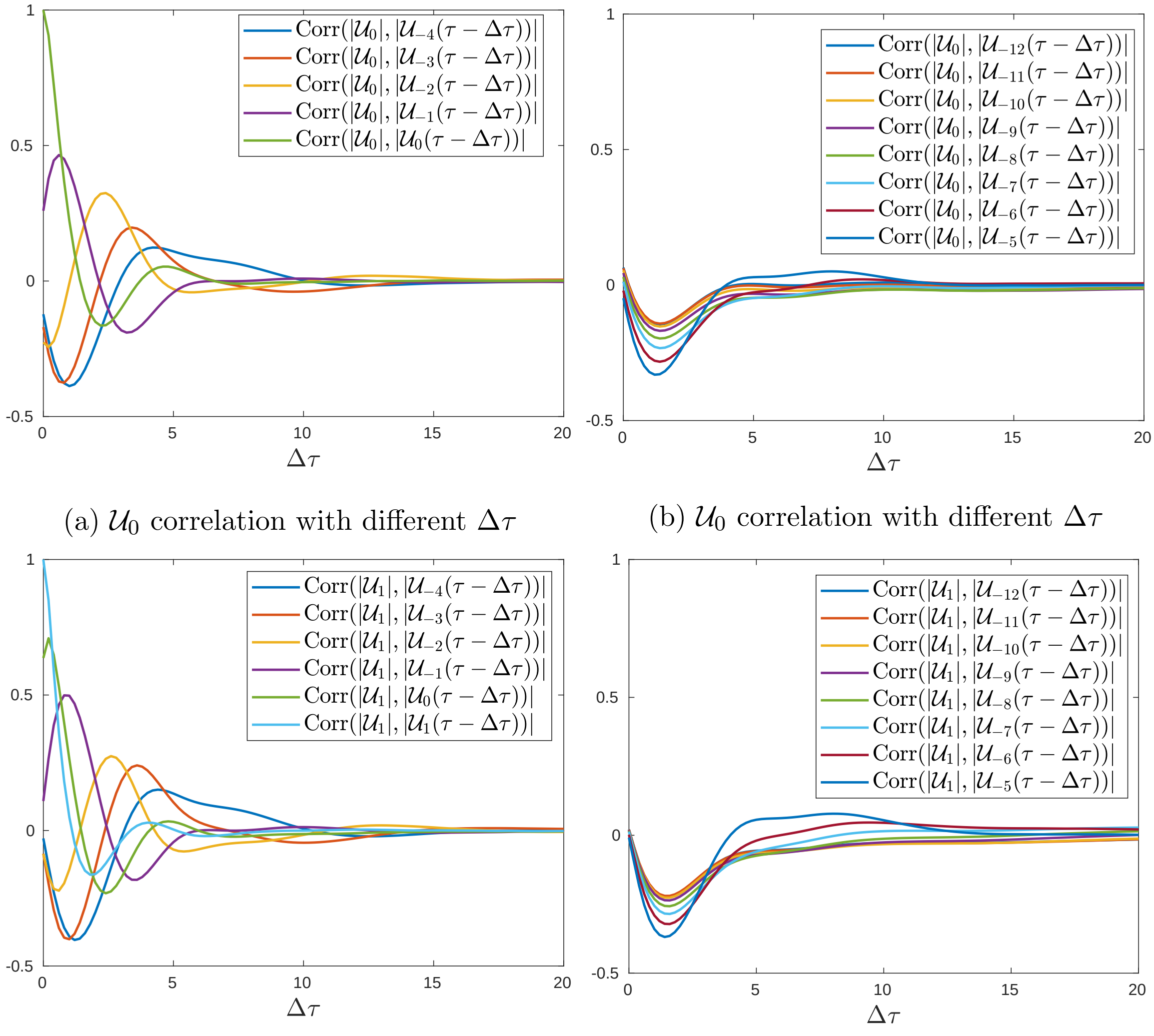
(c) Shell 12

Figure 5.8: Joint probability closure: PDFs of energy flux across different shells.

5.5 Conditional Probability Closure

The two closures we wrote so far rely solely on sampling directly from estimated densities. Now we want to draw samples conditioned to the time history of the system. We will start by conditioning only one of the closure shells to the last shell of the model, but delayed in time.

We build a data set consisting of pairs of the form $(\log_2 |\mathcal{U}_0(\tau)|, \log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|)$, for some $\Delta\tau > 0$. We can see in figure 5.9 that the correlation coefficients between $\mathcal{U}_0$ and all the other shells, as well as between $\mathcal{U}_1$ and all the other shells, decays in time. This means that there is a window of reasonable values for $\Delta\tau$. Conditioning $\mathcal{U}_0$ to $\mathcal{U}_{-1}$ with a long delay, i.e., with a large $\Delta\tau$, would not be effective because these quantities present such a low correlation coefficient.



(a) $\mathcal{U}_0$ correlation with different $\Delta\tau$

(b) $\mathcal{U}_0$ correlation with different $\Delta\tau$

(c) $\mathcal{U}_1$ correlation with different $\Delta\tau$

(d) $\mathcal{U}_1$ correlation with different $\Delta\tau$

Figure 5.9: Correlation between $\mathcal{U}_0$ and other delayed shells, and between $\mathcal{U}_1$ and other delayed shells for different values of $\Delta\tau$.

We see that the bulk of relevant values for $\Delta\tau$ is contained in, approximately, $0 < \Delta\tau \lesssim 8$. All the tests conducted in this work will contemplate values in this range. Let us write the closure

$$\mathcal{U}_0 = 2^{z_0} e^{i(\frac{\pi}{2} + \alpha_{-1} + \alpha_{-2})}, \tag{5.23}$$

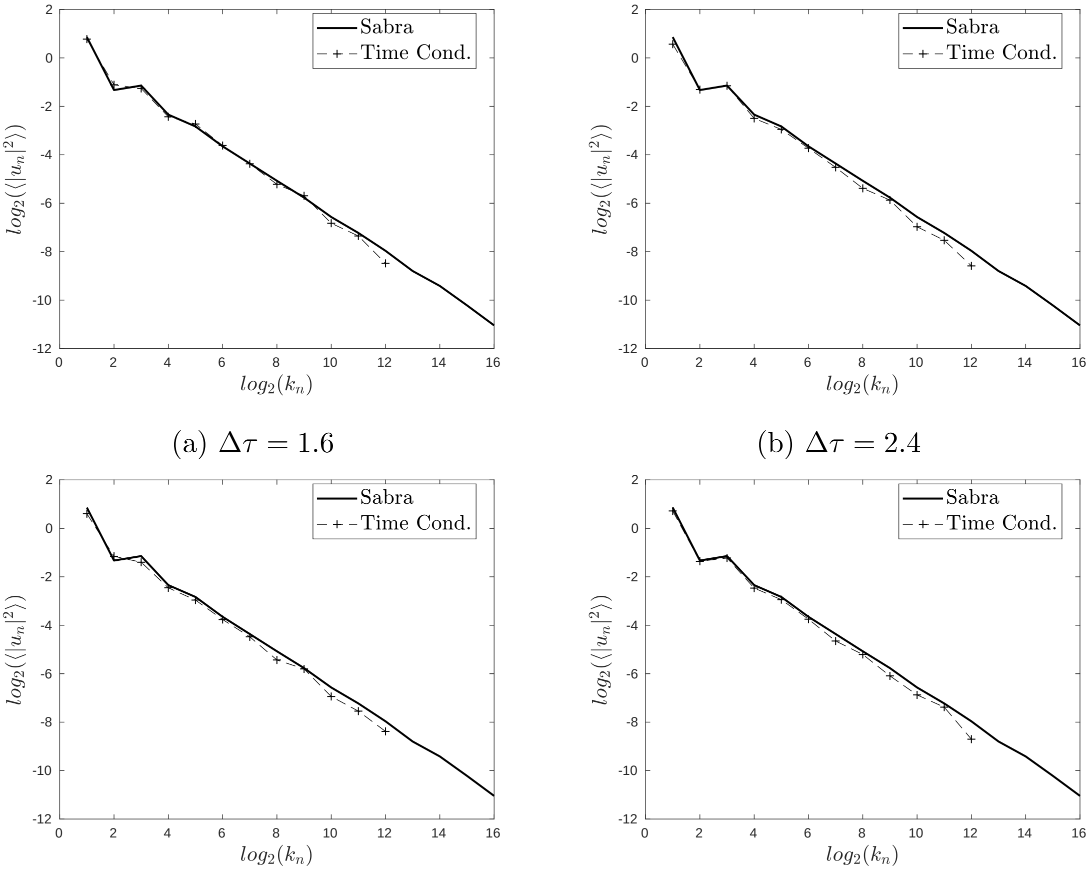
$$\mathcal{U}_1 = |\mathcal{U}_0| \lambda^{-1/3} e^{i(\pi + 2\alpha_{-1} + \alpha_{-2})}, \tag{5.24}$$

$$z_0 \sim f(z \big| \log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|). \tag{5.25}$$

Now our data set, comprised of pairs $(\log_2 |\mathcal{U}_0(\tau)|, \log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|)$, can be fed to the density estimator in order to obtain an estimation f for a two-dimensional joint probability density. This is also the weighted sum of three two-dimensional Gaussian components.

The value of $\mathcal{U}_{-1}(\tau-\Delta\tau)$ computed by the numerical method can be plugged into the second coordinate of the estimated density f and what is left is a one-dimensional slice, from which we can draw a sample for $\log_2|\mathcal{U}_0|$. This sampling process was described in section 4.1.2. Such value is, therefore, conditioned to whatever value $|\mathcal{U}_{-1}|$ had at time $\tau-\Delta\tau$.

While running a simulation of the closed model, we need to pay attention to the first values of $|\mathcal{U}_{-1}|$. For $\tau<\Delta\tau$, we are computing $\mathcal{U}_0$ and $\mathcal{U}_1$ with the Kolmogorov closure described in eqs. (5.14) and (5.15). Once τ reaches $\Delta\tau$ the closure changes to eqs. (5.23) to (5.25). Figure 5.10 shows $\log_2\langle|u_n|^2\rangle$ for some values of $\Delta\tau$.



(a) $\Delta\tau=1.6$

(b) $\Delta\tau=2.4$

(c) $\Delta\tau=3.2$

(d) $\Delta\tau=6.4$

Figure 5.10: Conditional probability closure: moment of order 2 for different values of $\Delta\tau$. Cut-off is $s=12$.

To show more detailed results, we choose $\Delta\tau=1.6$. In figure 5.11 we see moments of other orders.

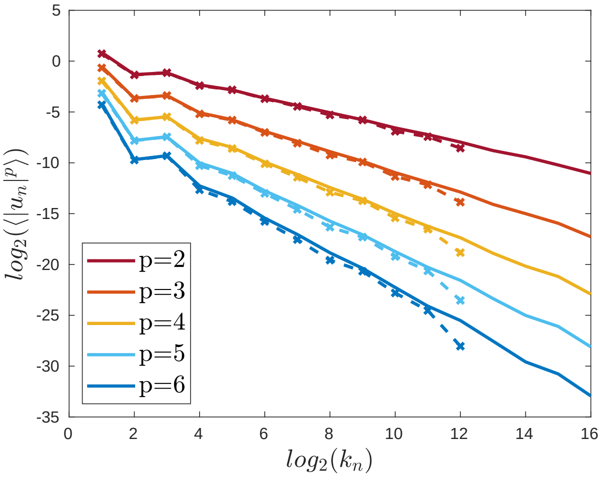
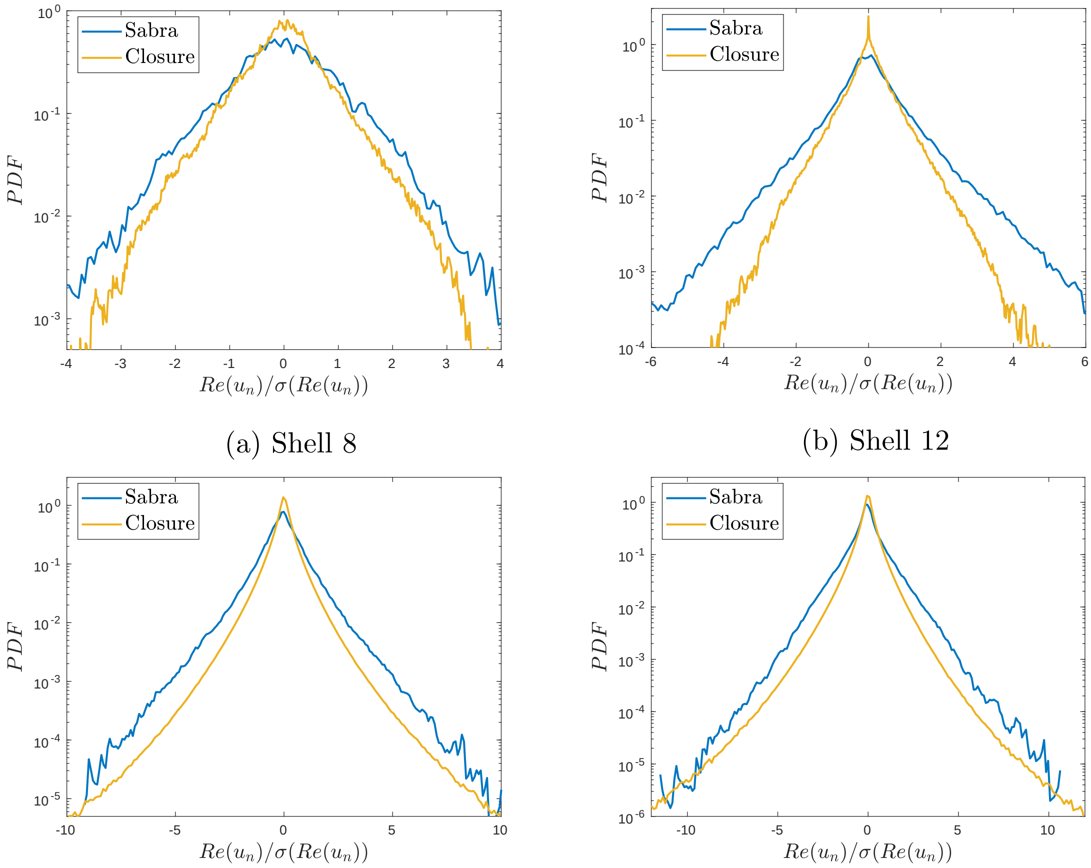


Figure 5.11: Conditional probability closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure.

Figures 5.12 and 5.13 show PDFs of real parts and energy flux across different shells.



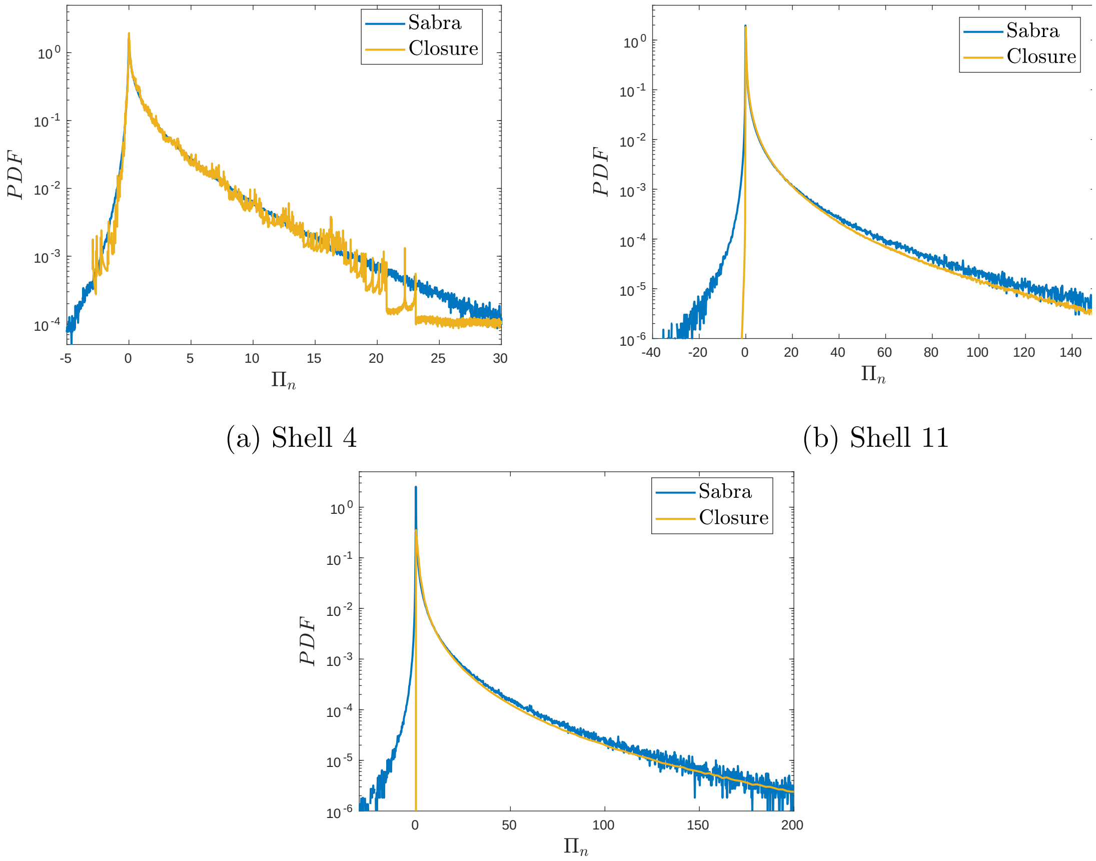
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 5.12: Conditional probability closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 5.13: Conditional probability closure: PDFs of energy flux across different shells.

5.6 Conditional Joint Probability Closure

Now, instead of conditioning only $|\mathcal{U}_0|$ to the value of $|\mathcal{U}_{-1}(\tau - \Delta\tau)|$ we can look at a joint probability for $|\mathcal{U}_0|$ and $|\mathcal{U}_1|$ conditioned to $|\mathcal{U}_{-1}(\tau - \Delta\tau)|$. The data fed to the density estimator is a triple of the form

$$(\log_2 |\mathcal{U}_0(\tau)|, \log_2 |\mathcal{U}_1(\tau)|, \log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|). \tag{5.26}$$

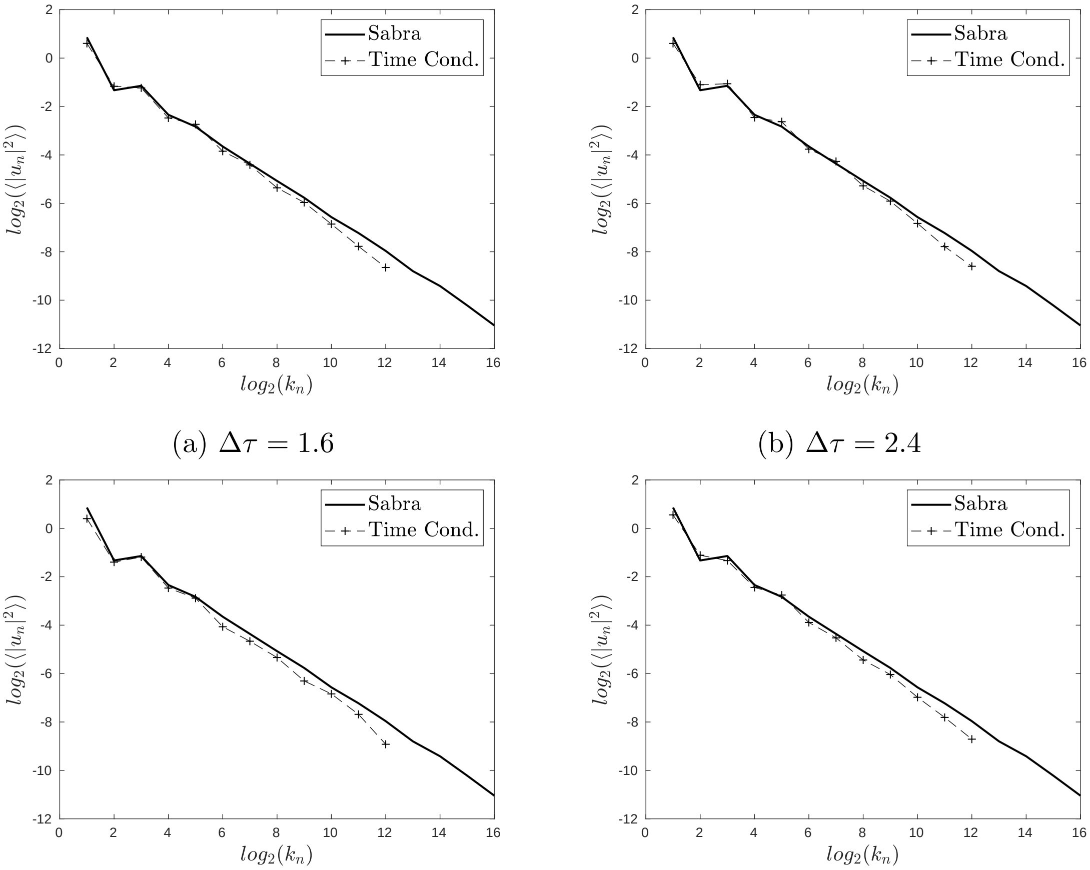
which shows we are estimating a three-dimensional density. Once a value for $|\mathcal{U}_{-1}(\tau - \Delta\tau)|$ is computed by the numerical method, we can again plug it into the third coordinate of the estimated density and we will end up with a two-dimensional slice, from which we can jointly sample z_0 and z_1 for the closure below.

$$\mathcal{U}_0 = 2^{z_0} e^{i(\frac{\pi}{2}+\alpha_{-1}+\alpha_{-2})}, \tag{5.27}$$

$$\mathcal{U}_1 = 2^{z_1} e^{i(\frac{\pi}{2}+\alpha_{0}+\alpha_{-1})}, \tag{5.28}$$

$$z = (z_0, z_1) \sim g(z|\log_2|\mathcal{U}_{-1}(\tau - \Delta\tau)|). \tag{5.29}$$

Again, different values of $\Delta\tau$ yield different results. Some are shown in figure 5.14.



(a) $\Delta\tau = 1.6$

(b) $\Delta\tau = 2.4$

(c) $\Delta\tau = 3.2$

(d) $\Delta\tau = 6.4$

Figure 5.14: Conditional joint probability closure: moment of order 2 for different values of $\Delta\tau$. Cut-off is $s = 12$.

For the detailed results shown below, we choose $\Delta\tau = 2.4$. In figure 5.15 we see moments of orders between two and six.

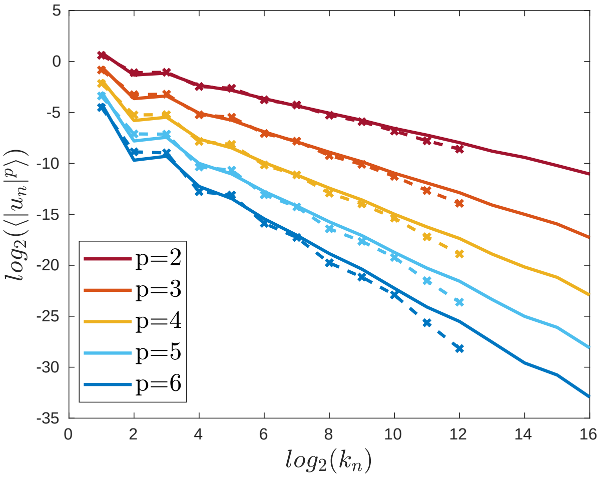
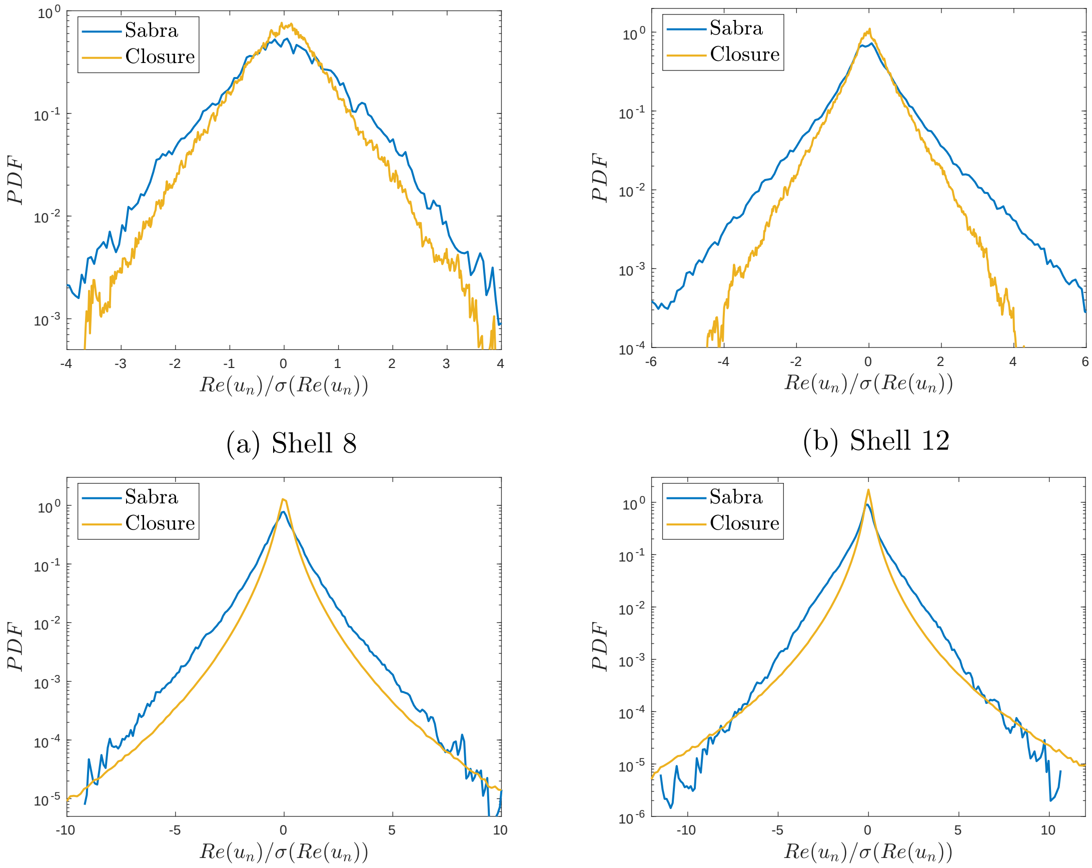


Figure 5.15: Conditional joint probability closure: moment of order 2 for $\Delta\tau = 2.4$. Cut-off is $s = 12$.

Figures 5.16 and 5.17 show PDFs of real parts and energy flux across different scales.



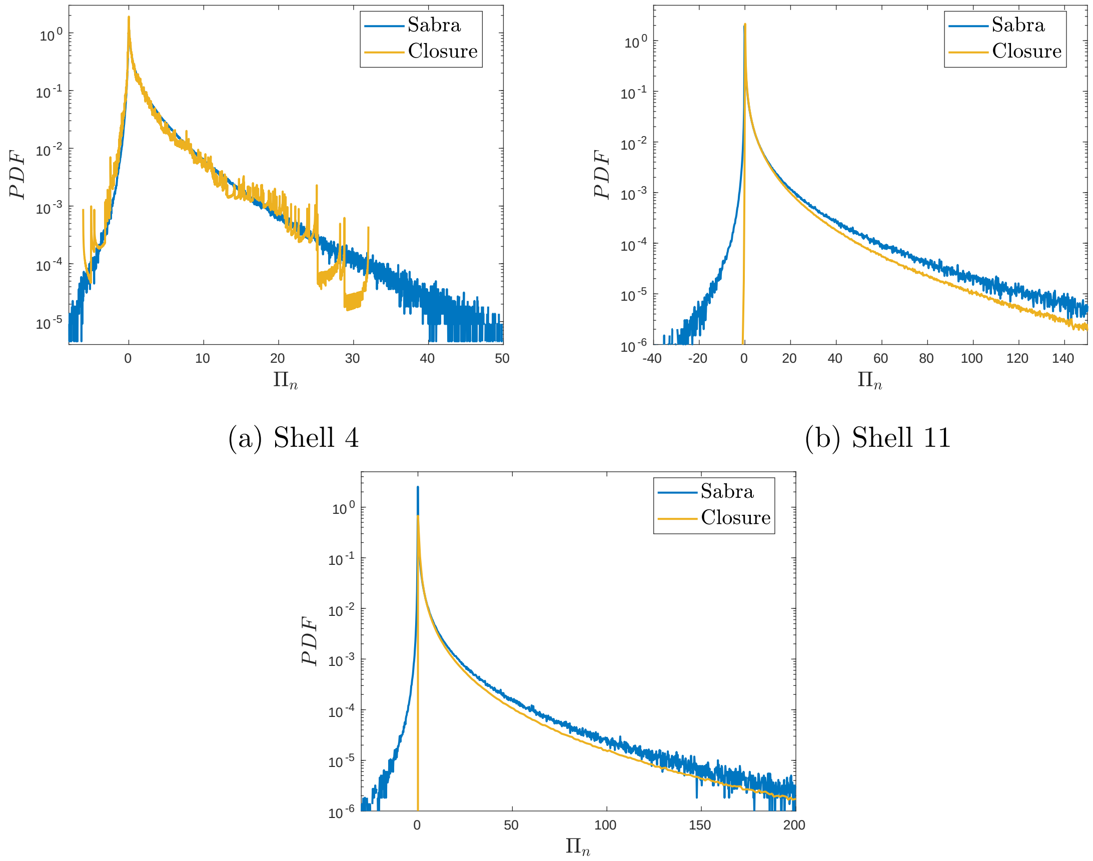
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 5.16: Conditional joint probability closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 5.17: Conditional joint probability closure: PDFs of energy flux across different shells.

5.7 Three-closest Conditional Probability Closure

One of the natural questions that rise from the last two closures is to how many shells can a pair $(z_0, z_1) = (\log_2 |\mathcal{U}_0(\tau)|, \log_2 |\mathcal{U}_1(\tau)|)$ be conditioned, and which ones will yield the best results. Let us look at how the closure behaves when (z_0, z_1) is conditioned to the last three shells delayed $\Delta\tau$ in time. This closure then has the form

$$\mathcal{U}_0 = 2^{z_0} e^{i(\frac{\pi}{2}+\alpha_{-1}+\alpha_{-2})}, \tag{5.30}$$

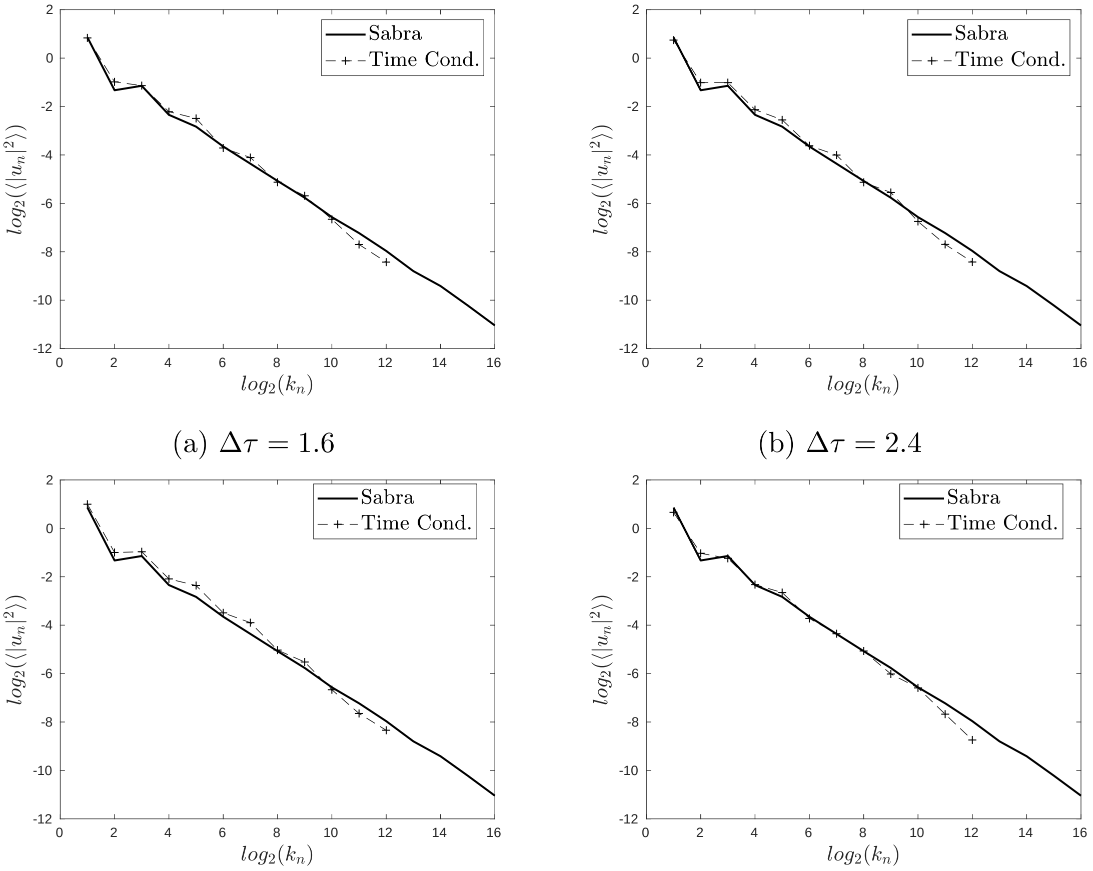
$$\mathcal{U}_1 = 2^{z_1} e^{i(\frac{\pi}{2}+\alpha_{0}+\alpha_{-1})}, \tag{5.31}$$

$$z = (z_0, z_1) \sim g(z| \log_2 |\mathcal{U}_{-3}(\tau - \Delta\tau)|, \log_2 |\mathcal{U}_{-2}(\tau - \Delta\tau)|, \log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|), \tag{5.32}$$

and the data set contains 5-tuples of the form

$$(\log_2|\mathcal{U}_0(\tau)|, \log_2|\mathcal{U}_1(\tau)|, \log_2|\mathcal{U}_{-3}(\tau-\Delta\tau)|, \log_2|\mathcal{U}_{-2}(\tau-\Delta\tau)|, \log_2|\mathcal{U}_{-1}(\tau-\Delta\tau)|). \tag{5.33}$$

Results for different values of $\Delta\tau$ can be seen in figure 5.18.



(a) $\Delta\tau = 1.6$

(b) $\Delta\tau = 2.4$

(c) $\Delta\tau = 3.2$

(d) $\Delta\tau = 6.4$

Figure 5.18: Three-closest closure: moment of order 2 for different values of $\Delta\tau$. Cut-off is $s = 12$.

We choose $\Delta\tau = 2.4$ to display more detailed results below. Figure 5.19 shows moments of order between two and six.

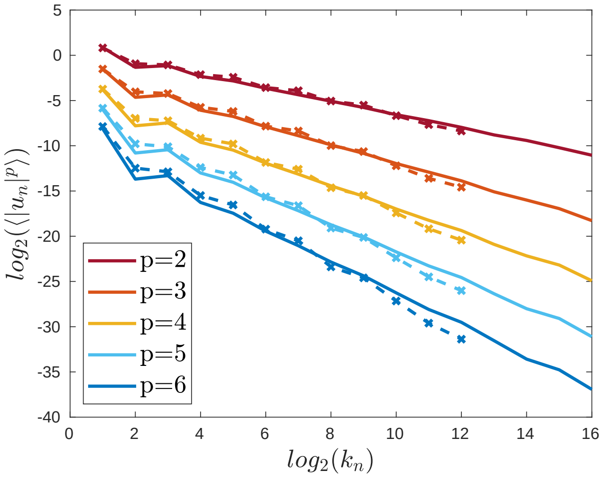


Figure 5.19: Three-closest closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure.

In figure 5.20 we see the solution computed by this closure, in Sabra variables and plotted in original time t.

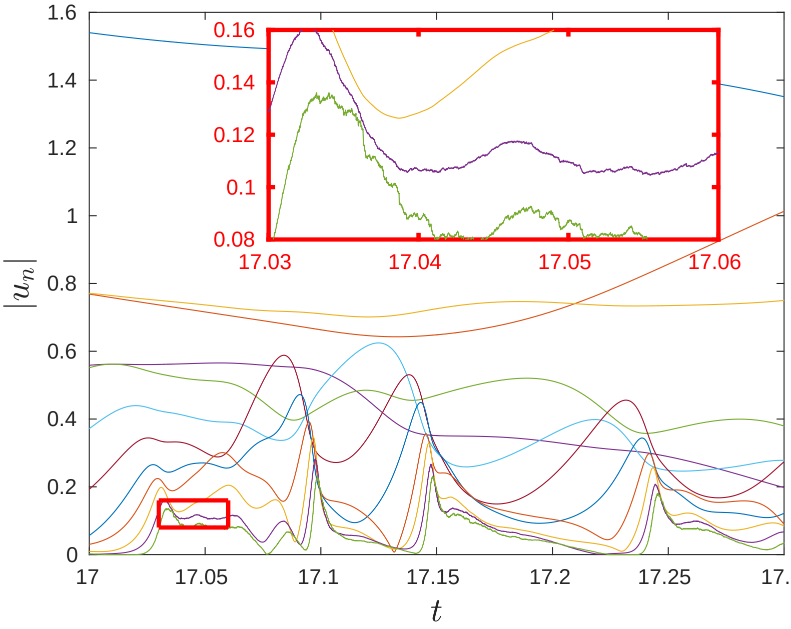
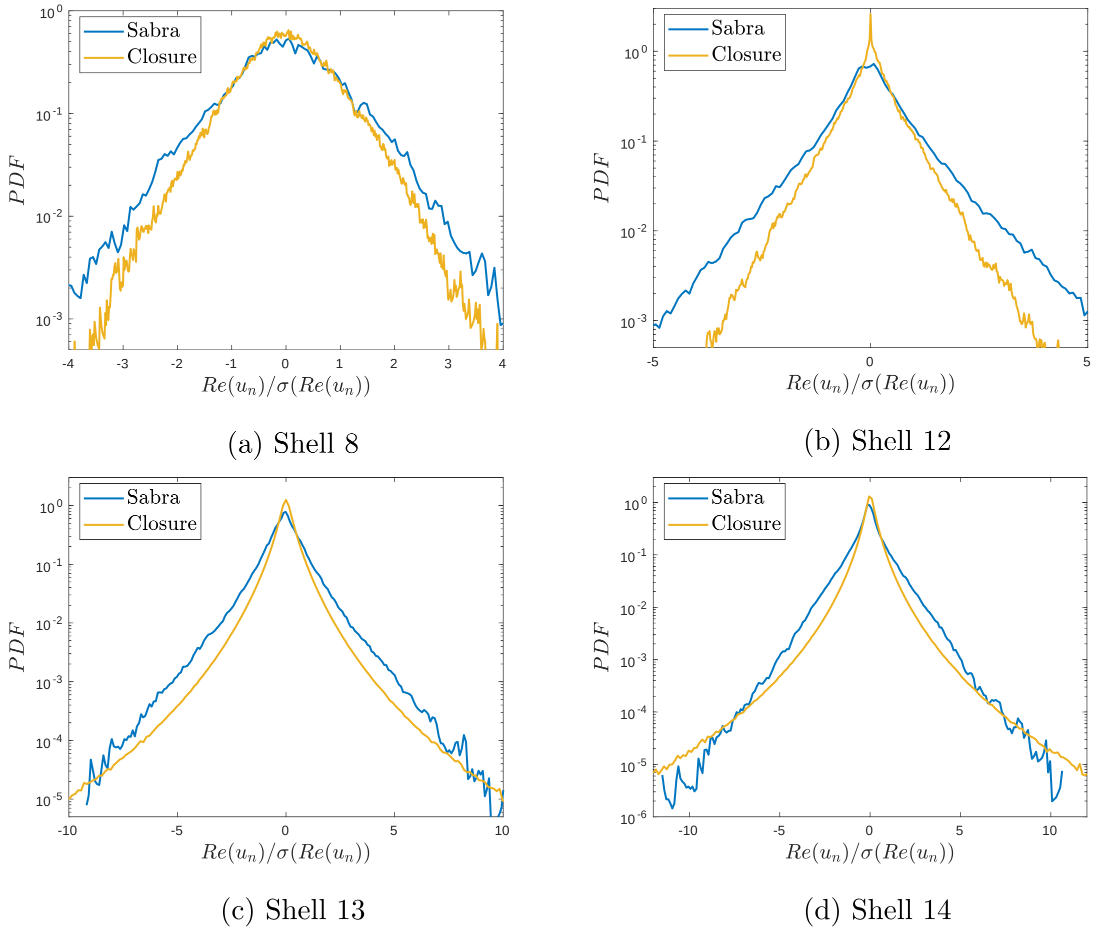


Figure 5.20: Three-closest closure: absolute value of the solution, with an inset of a zoom in.

We can see PDFs for real part and energy flux across different scales in figures 5.21 and 5.22.



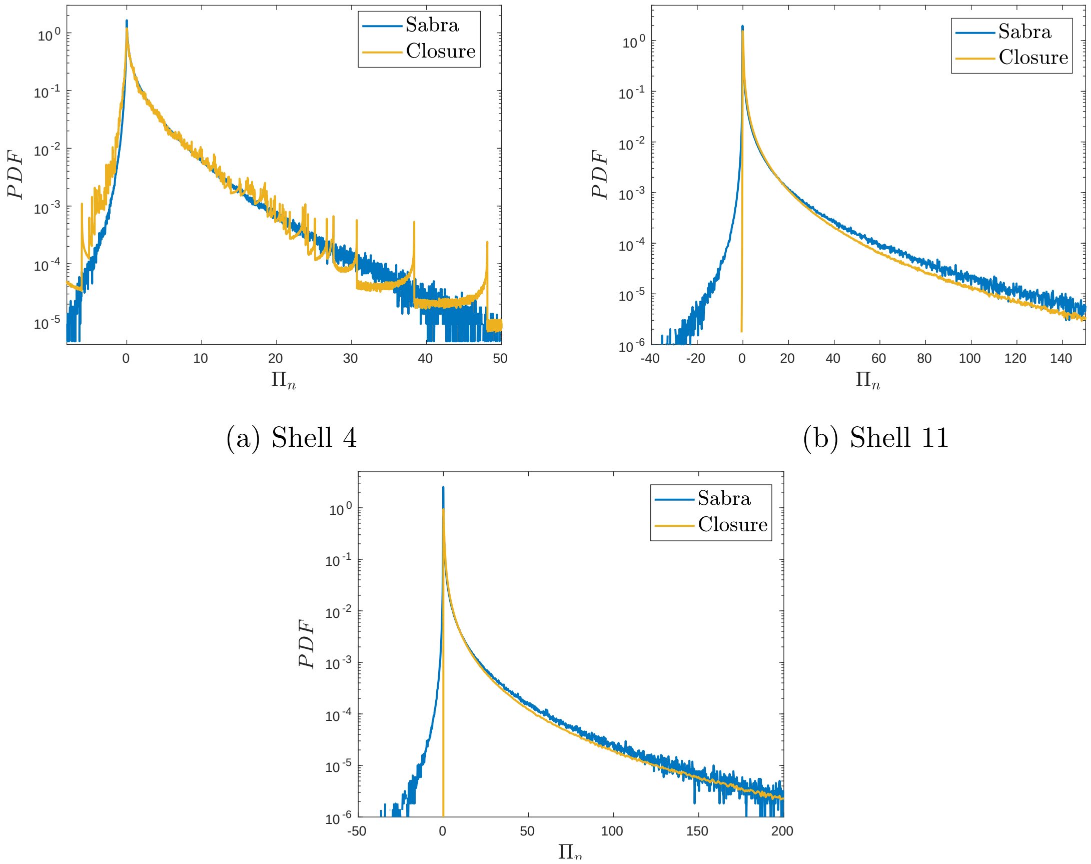
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 5.21: Three-closest closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 5.22: Three-closest closure: PDFs of energy flux across different shells.

5.8 Three-closest Conditional Probability Closure II

At this point someone may be wondering if the claim that our closures perform equivalently for any cut-off shell in the inertial range actually holds. For the same closure, as in eqs. (5.30) to (5.32), we run a second test with a cut-off shell $s = 9$ and $\Delta\tau = 2.4$.

In figure 5.23 we can observe moments of different orders.

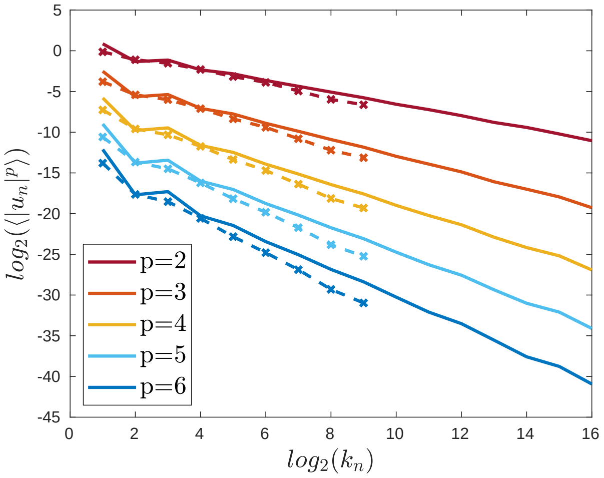


Figure 5.23: Three-closest closure II: Moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure. Cut-off shell is s=9.

In figure 5.24 we see the solution computed by this closure, in Sabra variables and plotted in original time t.

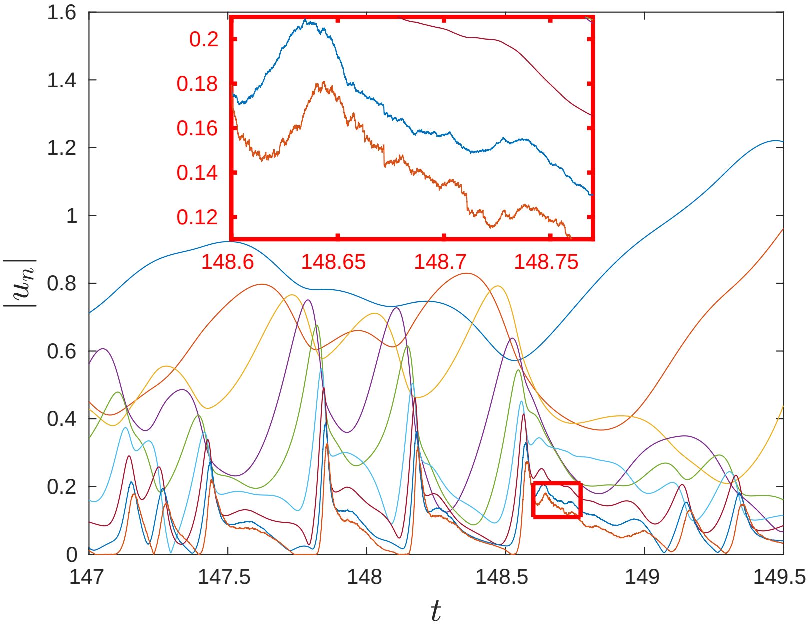
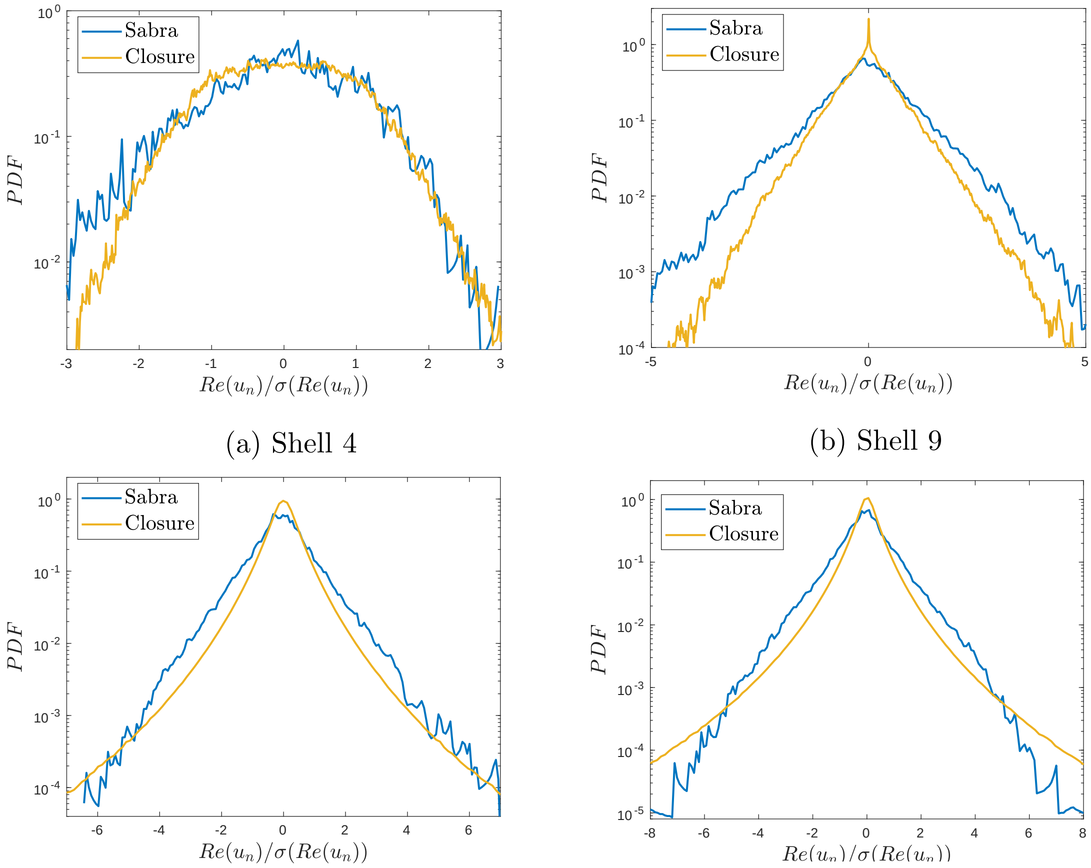


Figure 5.24: Three-closest closure II: absolute value of the solution, with an inset of a zoom in.

We can see how this closure performs by comparing PDFs of real parts in figure 5.25 and PDFs of energy flux across different scales in 5.26.



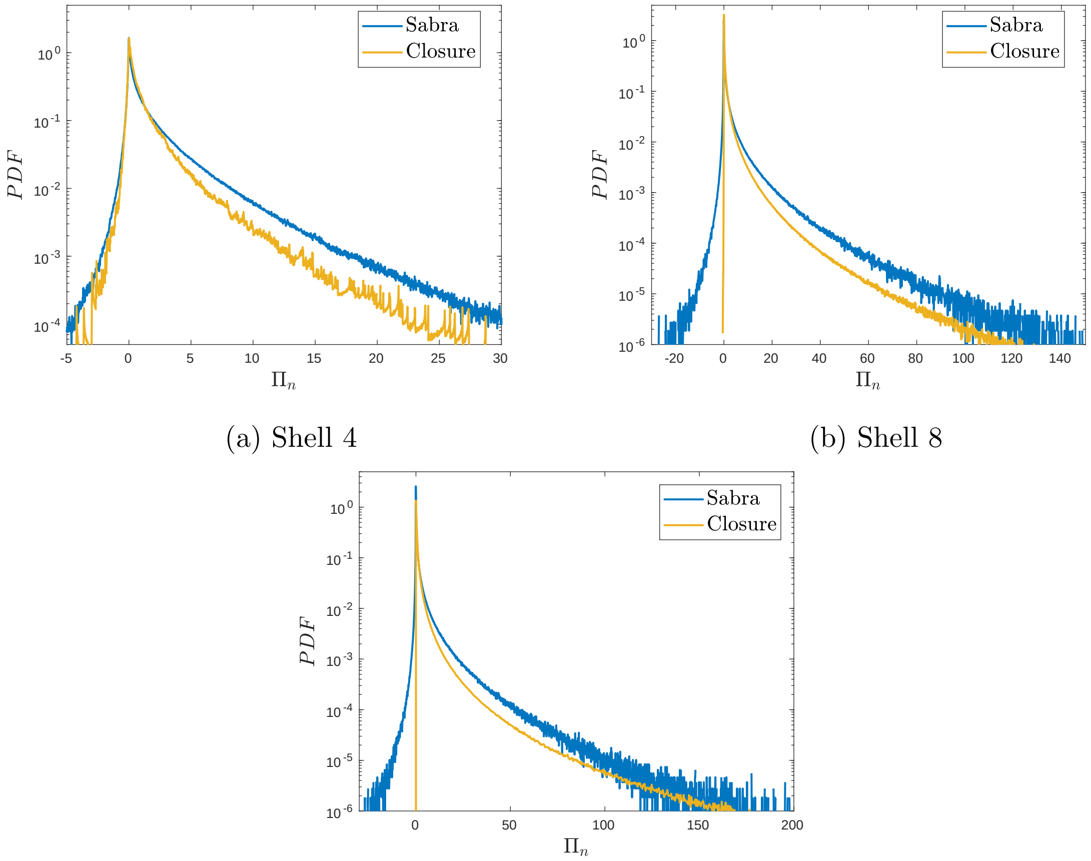
(a) Shell 4

(b) Shell 9

(c) Shell 10

(d) Shell 11

Figure 5.25: Three-closest closure II: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 8

(c) Shell 9

Figure 5.26: Three-closest closure II: PDFs of energy flux across different shells.

5.9 Long Conditional Probability Closure

To end this chapter, instead of conditioning (z_0, z_1) to only the three closest shells, we could simply condition to the prehistory of all previous shells. This gives us a closure of the form

$$\mathcal{U}_0 = 2^{z_0} e^{i(\frac{\pi}{2}+\alpha_{-1}+\alpha_{-2})}, \tag{5.34}$$

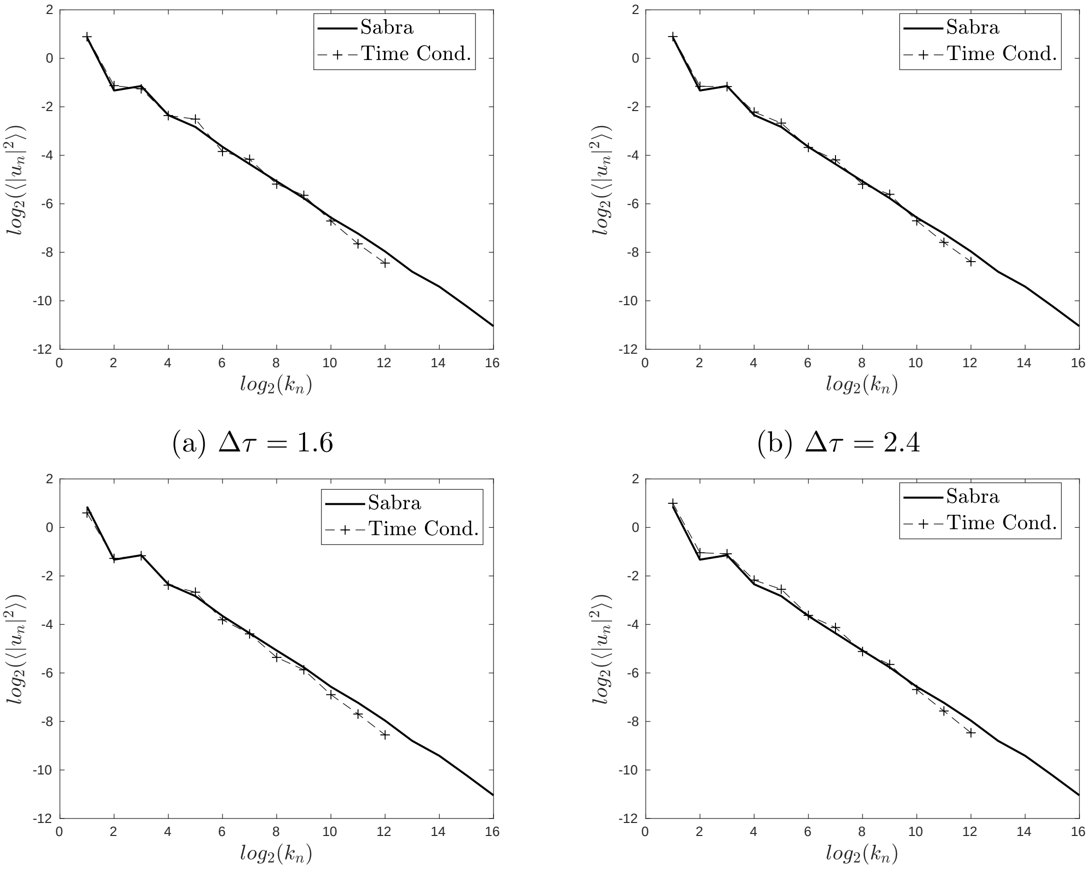
$$\mathcal{U}_1 = 2^{z_1} e^{i(\frac{\pi}{2}+\alpha_0+\alpha_{-1})}, \tag{5.35}$$

$$z = (z_0, z_1) \sim g(z| \log_2 |\mathcal{U}_{-s}(\tau - \Delta\tau)|, \cdots, \log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|). \tag{5.36}$$

and the data set is made of $(s+2)$-tuples of the form

$$(\log_2|\mathcal{U}_0(\tau)|, \log_2|\mathcal{U}_1(\tau)|, \log_2|\mathcal{U}_{-s}(\tau-\Delta\tau)|, \cdots, \log_2|\mathcal{U}_{-1}(\tau-\Delta\tau)|). \tag{5.37}$$

Results for different $\Delta\tau$ can be seen in figure 5.27. It is important to note that, if we want to choose another cut-off shell after the density has been estimated with $s+2$ dimensions, we must choose one that is larger than s. Otherwise we would not have enough information to adequately condition the sampling process, and would need to re-estimate the density for the desired cut-off. This is a specific trait of this closure.



(a) $\Delta\tau = 1.6$

(b) $\Delta\tau = 2.4$

(c) $\Delta\tau = 3.2$

(d) $\Delta\tau = 6.4$

Figure 5.27: Long closure: moment of order 2 for different values of $\Delta\tau$. Cut-off is $s = 12$.

In figure 5.28 we can see moments of orders up to six for $\Delta\tau = 2.4$.

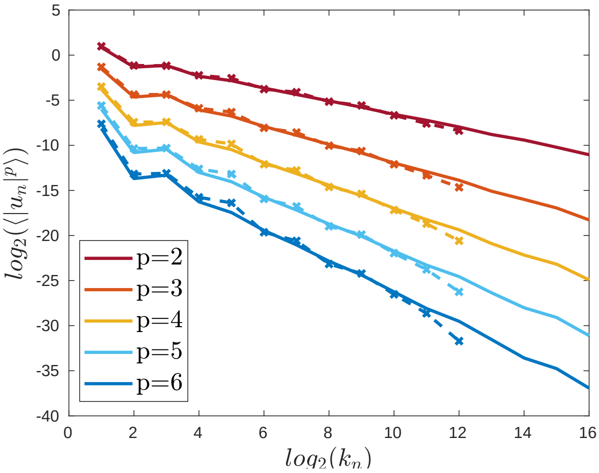
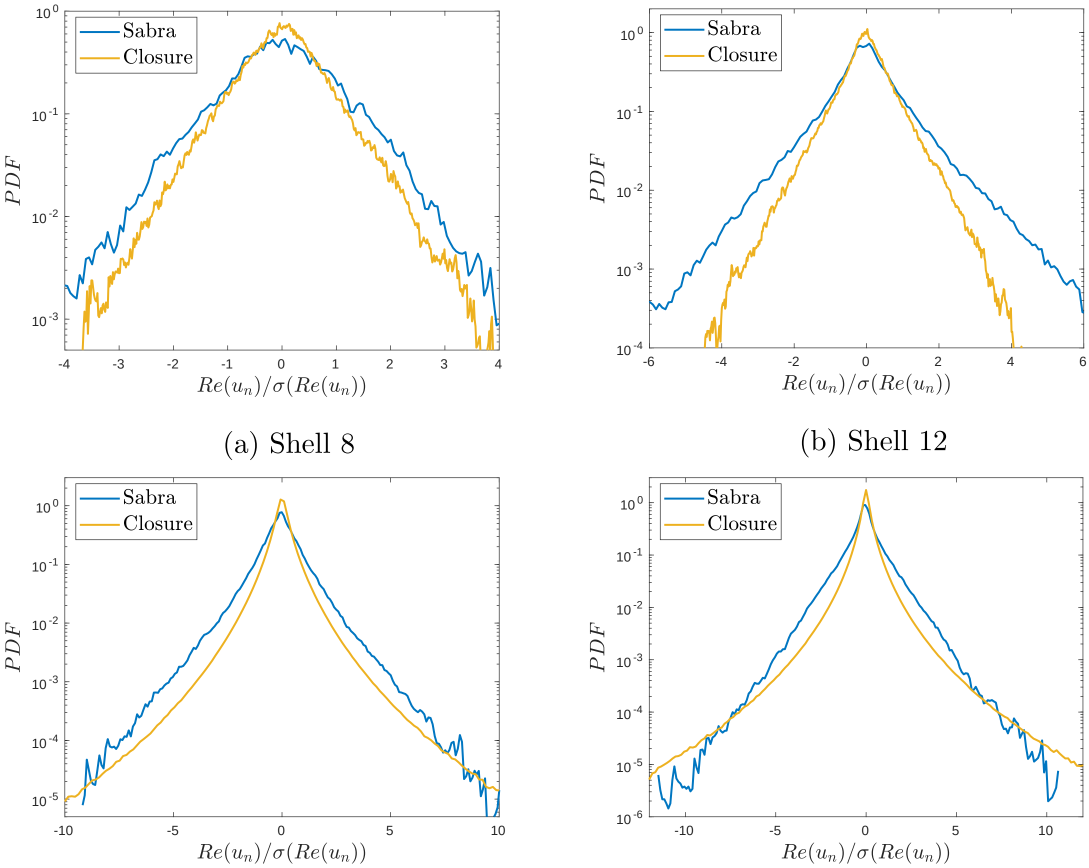


Figure 5.28: Long closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure.



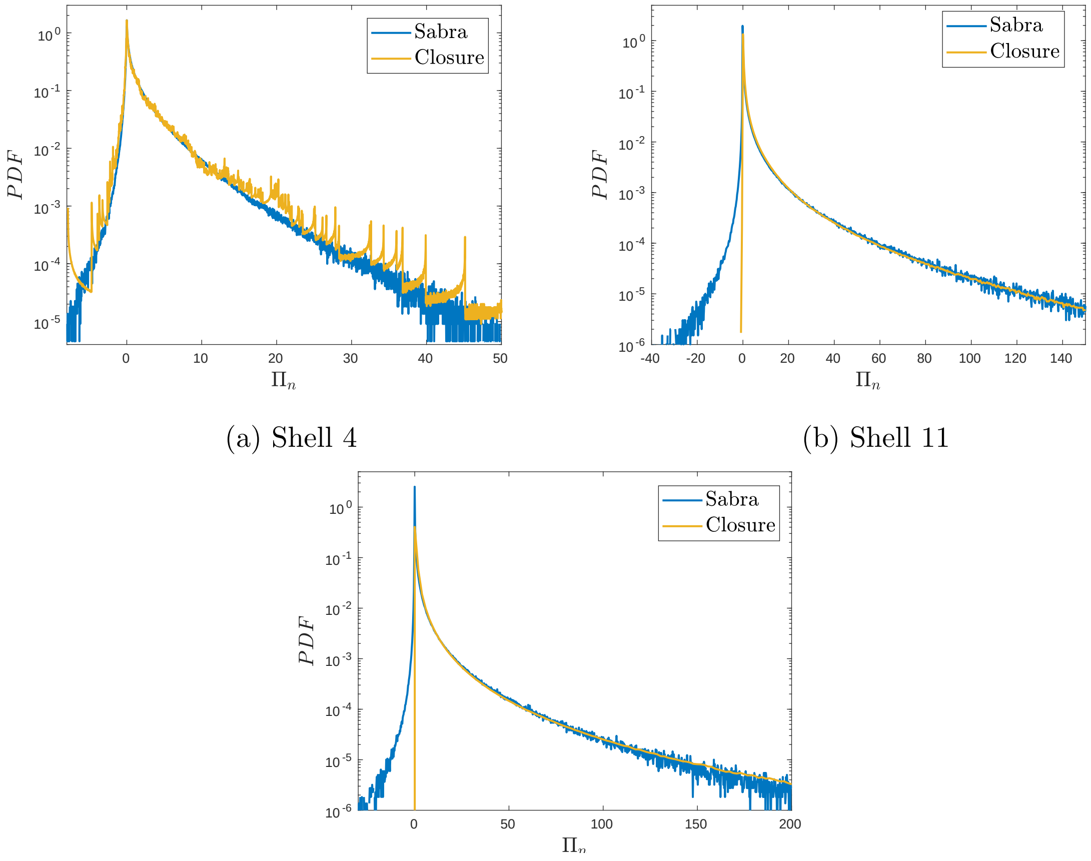
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 5.29: Long closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 5.30: Long closure: PDFs of energy flux across different shells.

5.10 In this chapter

We established the reduced models we want to simulate and wrote a variety of closures for such models. All the closures presented in this chapter only involved modelling absolute values, while phases are kept fixed at their most probable value. Qualitatively all closures are performing fine, although they are displaying some difficulty reproducing PDFs of energy flux and, at times, the tails of PDFs of real parts.

Chapter 6

Closures with phase modelling

We intend now to provide some treatment for the phases to go along with the closures we presented in chapter 5. We write several closures, many are conditioned in time and all are probabilistic. The closures presented in this chapter encompass substantially more information than the previous ones.

Before we start, we need to mention a small detail. The phases Δ_N are computed according to equation (3.7), which is reproduced below for convenience,

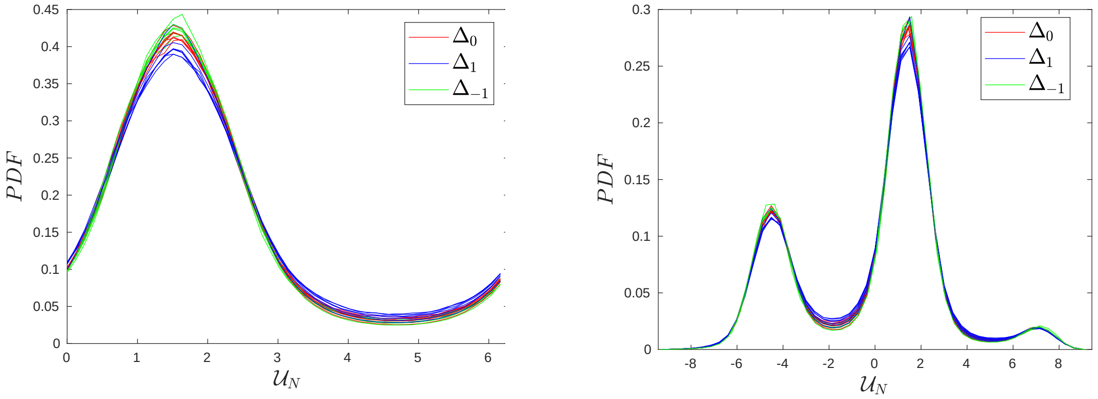
$$\Delta_N = \arg(\mathcal{U}_N) - \arg(\mathcal{U}_{N-1}) - \arg(\mathcal{U}_{N-2}). \tag{3.7}$$

For $\delta_n \in [\frac{\pi}{2}, \frac{3\pi}{2}]$ the energy flux through a shell n as computed by equation (2.8) is positive [8]. For this reason, all the closures in chapter 5 were strictly dissipative, which reflected directly in the flux PDFs for the cut-off shell (either 12 or 9) in the form of positive values only.

The energy flux in the Sabra system, however, is not strictly positive, in fact it displays some negative values. This means that a correct closure is not strictly dissipative, but needs to show some backscattering. The difficulty in modelling phases comes from the fact that closures with negative flux values at the cut-off shell can incur in numerical instabilities and inaccurate modelling of the phases can lead to blow-up.

Because we think of them as $2\pi-$periodic values, what we see in figures like 3.3b, reproduced in figure 6.1a, is a $2\pi-$periodic density. However, the GMM described in chapter 4 does not deal well with periodic densities, or with discontinuities. Therefore, the density we are considering here is the one where we compute each Δ_N as in equation

(3.7) but do not think of it as a $2\pi-$periodic value. Instead, we are thinking of it as a $6\pi-$periodic value, since each term on the right-hand side of equation (3.7) is $2\pi-$periodic and there are three of them. The two formulations, 2π and 6π periodic, are equivalent and both densities can be seen in figure 6.1b.



(a) $2\pi-$periodic phases

(b) $6\pi-$periodic phases

Figure 6.1: Probability densities of Δ_N

The probability densities in figure 6.1b present quite a feasible data set for GMM, while the same data set reduced to a 2π period is problematic. There are works in GMM for periodic densities [2, 66], but they do not scale easily to high dimensions, which we need.

Now, let us write two intermediate closures. In the first, $\log_2|\mathcal{U}_0| = z_0$ and the associated multiplier phase $\Delta_0 = z_1$, defined in equation (3.7), are both sampled from a joint probability distribution g, while the module and phase of $\mathcal{U}_1$ are given by Kolmogorov's closure.

$$\mathcal{U}_0 = 2^{z_0} e^{i(z_1+\alpha_{-1}+\alpha_{-2})}, \tag{6.1}$$

$$\mathcal{U}_1 = |\mathcal{U}_0|\lambda^{-1/3} e^{i(\frac{\pi}{2}+\alpha_0+\alpha_{-1})}, \tag{6.2}$$

$$z = (z_0, z_1) \sim g(z). \tag{6.3}$$

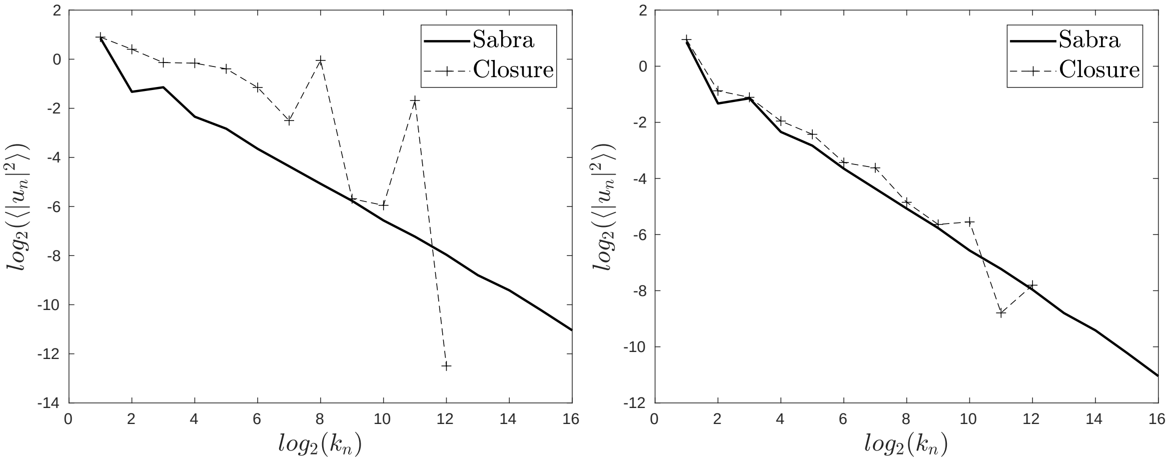
The second closure is an equivalent of the first one, but now $\log_2|\mathcal{U}_1| = z_2$ and the associated multiplier phase $\Delta_1 = z_3$ are sampled, while module and phase of $\mathcal{U}_0$ are given by Kolmogorov's closure.6

$$\mathcal{U}_0 = |\mathcal{U}_{-1}|\lambda^{-1/3}e^{i(\frac{\pi}{2}+\alpha_{-1}+\alpha_{-2})}, \tag{6.4}$$

$$\mathcal{U}_1 = 2^{z_2}e^{i(z_3+\alpha_0+\alpha_{-1})}, \tag{6.5}$$

$$z = (z_2, z_3) \sim g(z). \tag{6.6}$$

The combination of the two approaches does not seem do dissipate energy correctly. In fact, the moments of order 2 showed in figure 6.2 are extremely short simulations that encountered numerical instabilities very early on.



(a)

(b)

Figure 6.2: Intermediate closures with odd behavior: moments of order two. Cut-off is $s = 12$. In 6.2a $\mathcal{U}_0$ is given by closure, $\mathcal{U}_1$ is given by Kolmogorov. In 6.2b $\mathcal{U}_1$ is given by closure, $\mathcal{U}_0$ is given by Kolmogorov.

6.1 Joint probability closure

When both $\mathcal{U}_0$ and $\mathcal{U}_1$ have their module and phase sampled from a four-dimensional joint density, we can write the closure as follows.

$$\mathcal{U}_0 = 2^{z_0}e^{i(z_1+\alpha_{-1}+\alpha_{-2})}, \tag{6.7}$$

$$\mathcal{U}_1 = 2^{z_2}e^{i(z_3+\alpha_0+\alpha_{-1})}, \tag{6.8}$$

$$z = (z_0, z_1, z_2, z_3) \sim g(z). \tag{6.9}$$

The density estimation from here on, until said otherwise, is performed with six Gaussian components on top of 80000 data samples.

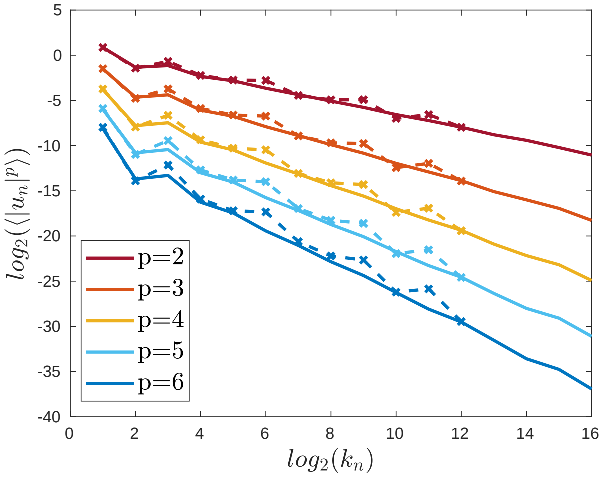
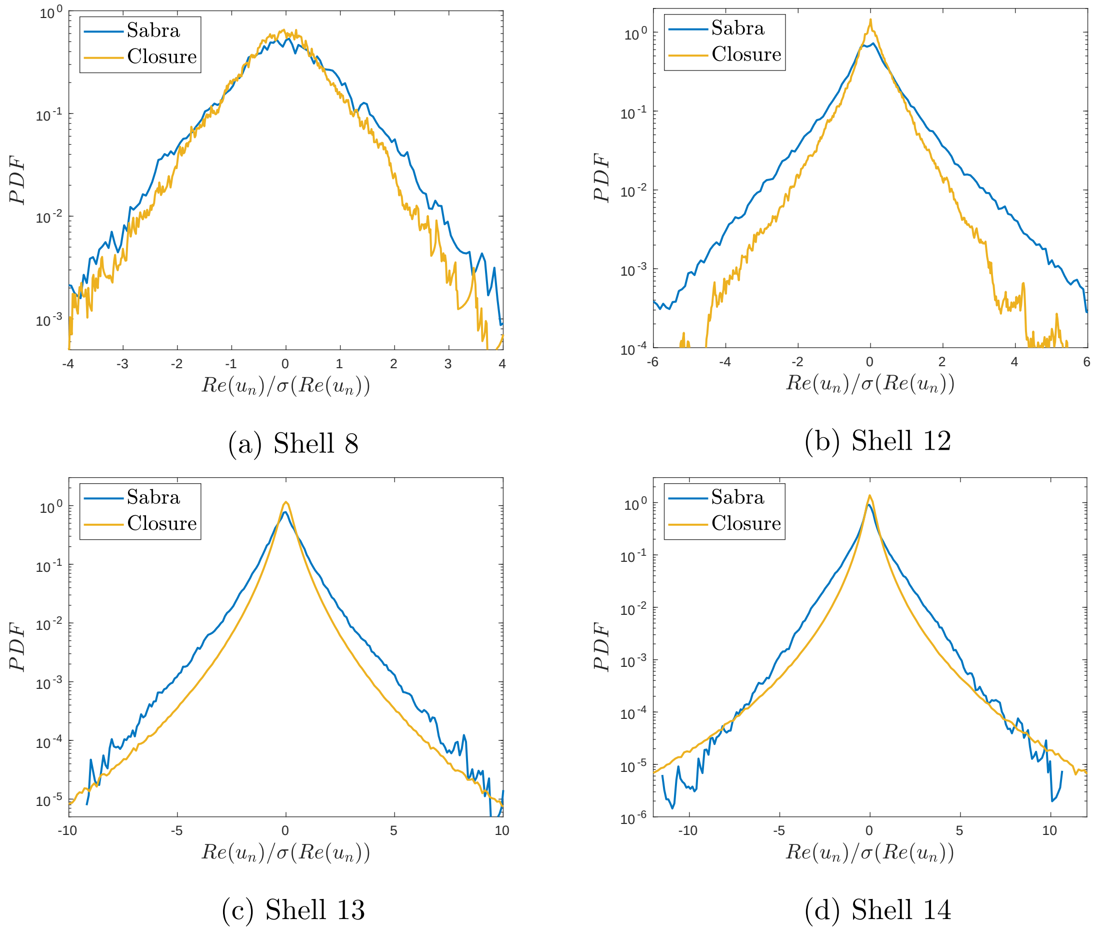


Figure 6.3: Joint probability closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure.

This is a preliminary closure in which none of the quantities are conditioned, but they are jointly estimated and sampled. Figures 6.4 and 6.5 show PDFs of real parts and energy flux.



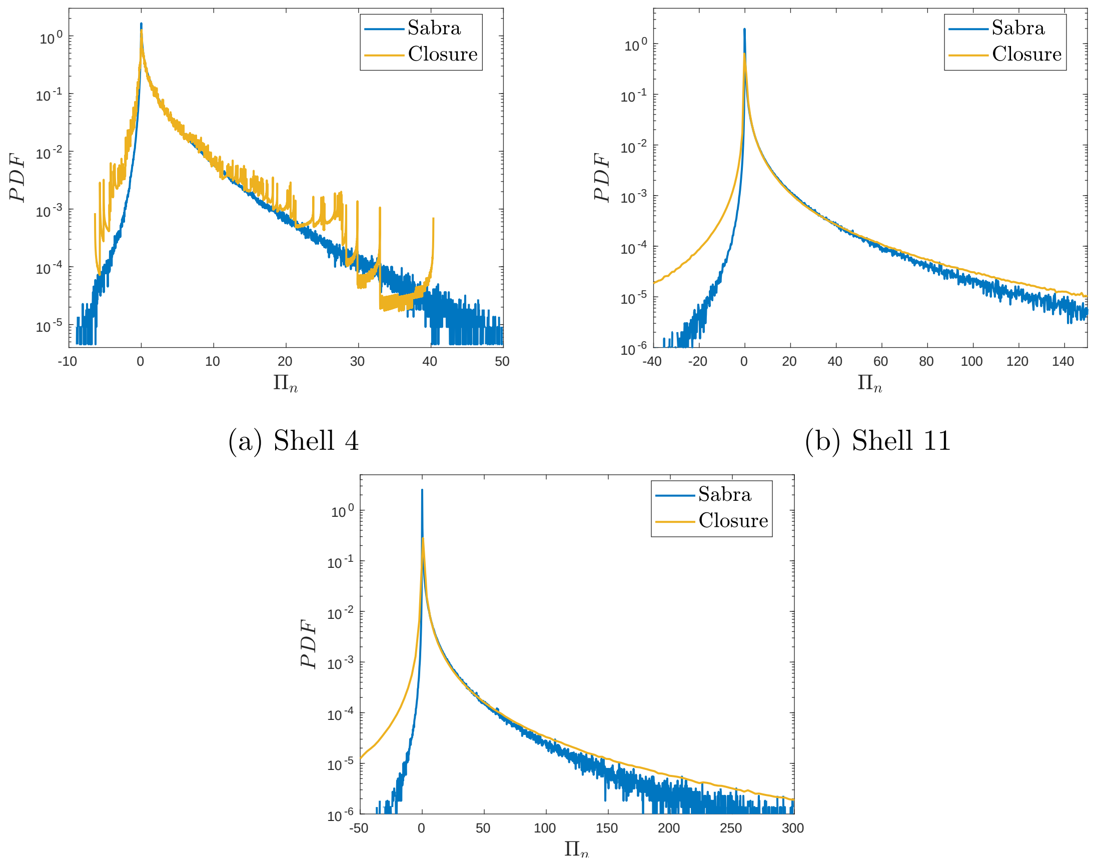
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 6.4: Joint probability closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 6.5: Joint probability closure: PDFs of energy flux across different shells.

We can see in figure 6.5 that this closure already presents a more realistic energy flux through the cut-off shell, albeit with much stronger backscattering than the actual11 dynamics.

6.2 Time conditioning

Now we can start writing closures that are conditioned in time. First, let us condition $\mathcal{U}_0$ and $\mathcal{U}_1$ to $\mathcal{U}_{-1}$. It is relevant to note that we are conditioning not only to the module of $\mathcal{U}_{-1}$, but also to the multiplier phase Δ_{-1}. We can write the closure as

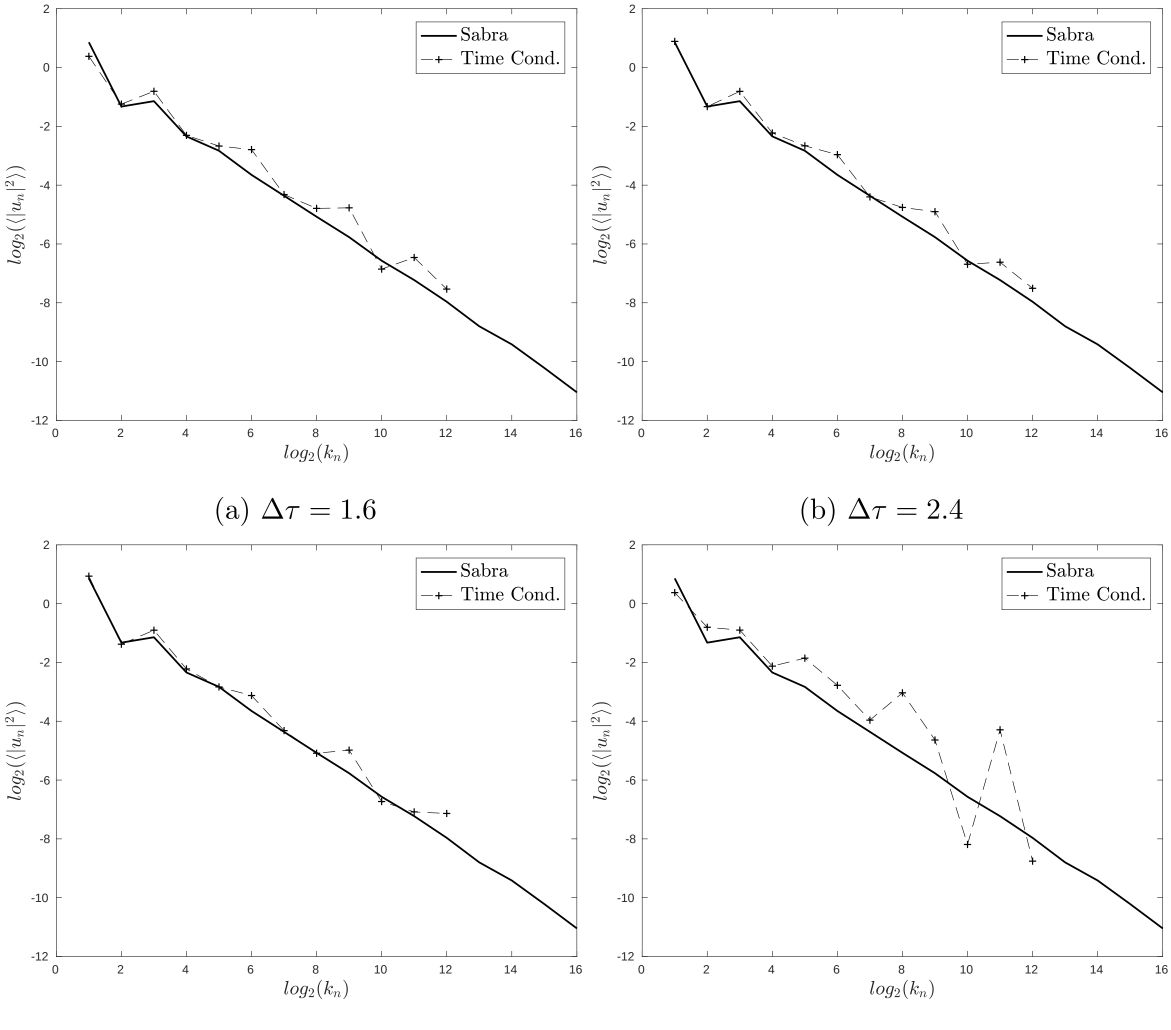
$$\mathcal{U}_0 = 2^{z_0} e^{i(z_1+\alpha_{-1}+\alpha_{-2})}, \tag{6.10}$$

$$\mathcal{U}_1 = 2^{z_2} e^{i(z_3+\alpha_0+\alpha_{-1})}, \tag{6.11}$$

$$z = (z_0, z_1, z_2, z_3) \sim g(z|z'). \tag{6.12}$$

$$z' = (\log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|, \Delta_{-1}) \tag{6.13}$$

Again, different values of $\Delta\tau$ yield different results. Moments of order two for some values of $\Delta\tau$ can be seen in figure 6.6.



(a) $\Delta\tau = 1.6$

(b) $\Delta\tau = 2.4$

(c) $\Delta\tau = 3.2$

(d) $\Delta\tau = 6.4$

Figure 6.6: Time conditioned closure: moment of order 2 for different values of $\Delta\tau$. Cut-off is $s = 12$.

We choose $\Delta\tau = 3.2$ to present more detailed results. In figure 6.7 we see moments of order between two and six.

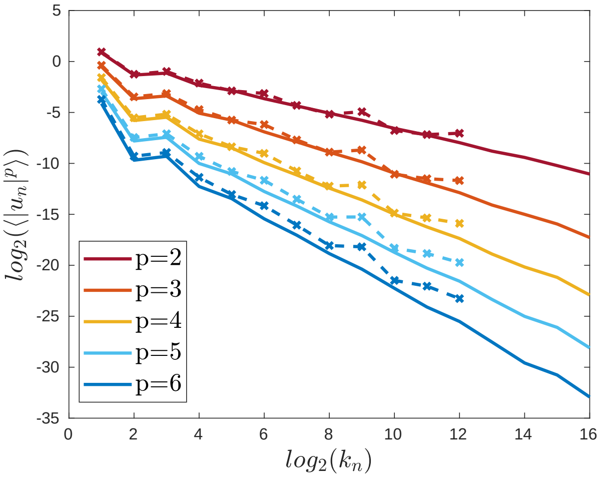
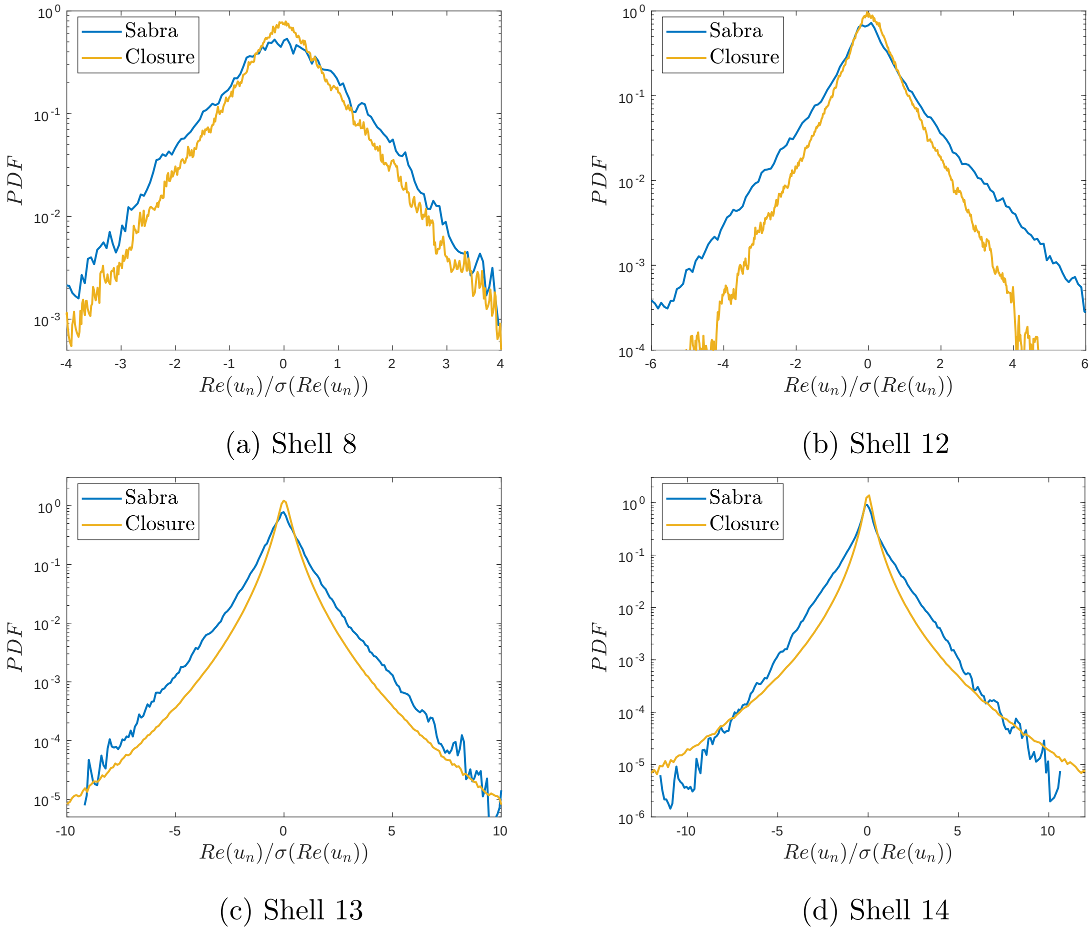


Figure 6.7: Time conditioned closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure.

Figures 6.8 and 6.9 show PDFs for real parts and energy flux.



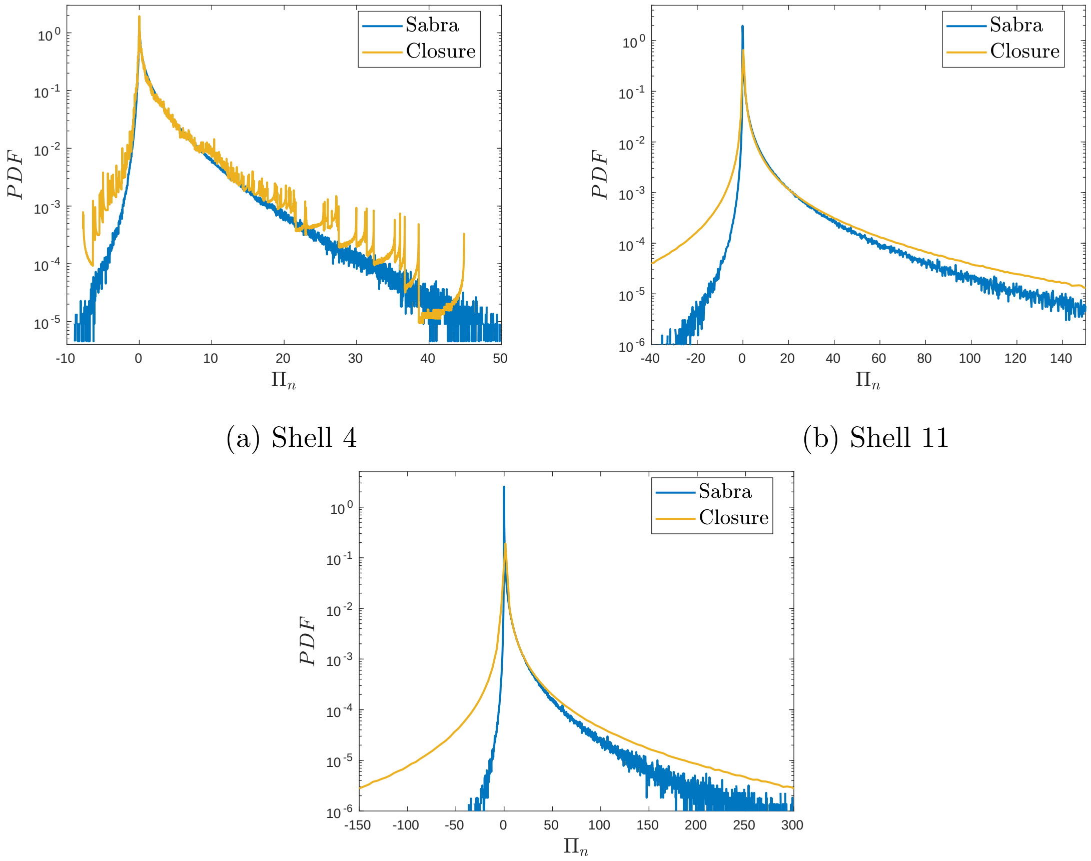
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 6.8: Time conditioned closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 6.9: Time conditioned closure: PDFs of energy flux across different shells.

6.3 Self conditioning

We can write different closures based on what the modules and phases of $\mathcal{U}_0$ and $\mathcal{U}_1$ are conditioned. We can condition, for example, to the modules of $\mathcal{U}_0(\tau - \Delta\tau)$ and $\mathcal{U}_1(\tau - \Delta\tau)$, and to their associated multipliers phases Δ_0 and Δ_1. This gives rise to the following closure

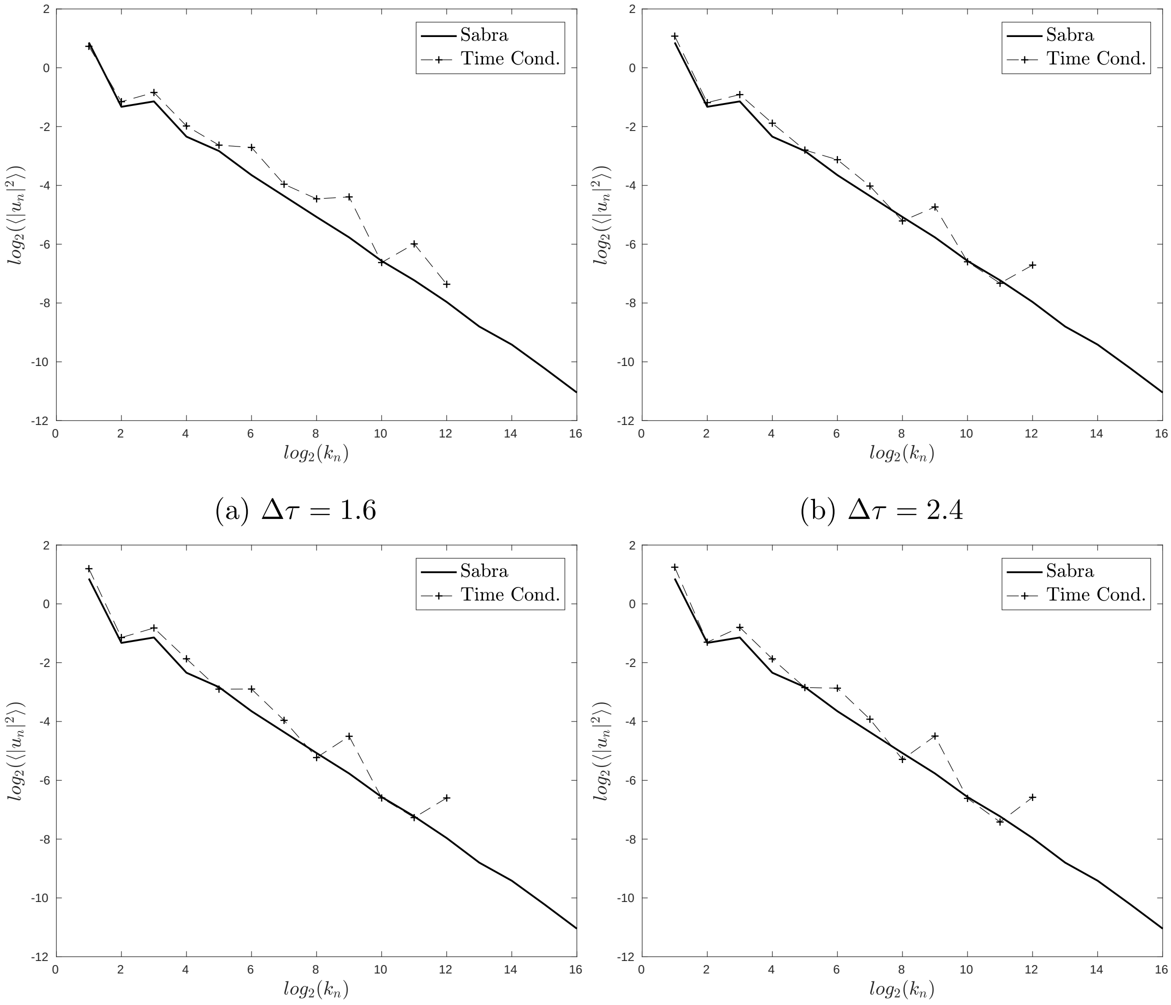
$$\mathcal{U}_0 = 2^{z_0} e^{i(z_1 + \alpha_{-1} + \alpha_{-2})}, \tag{6.14}$$

$$\mathcal{U}_1 = 2^{z_2} e^{i(z_3 + \alpha_0 + \alpha_{-1})}, \tag{6.15}$$

$$z = (z_0, z_1, z_2, z_3) \sim g(z|z'). \tag{6.16}$$

$$z' = (\log_2 |\mathcal{U}_0(\tau - \Delta\tau)|, \Delta_0, \log_2 |\mathcal{U}_1(\tau - \Delta\tau)|, \Delta_1). \tag{6.17}$$

Results for different values of $\Delta\tau$ can be seen in figure 6.10.



(a) $\Delta\tau = 1.6$

(b) $\Delta\tau = 2.4$

(c) $\Delta\tau = 3.2$

(d) $\Delta\tau = 6.4$

Figure 6.10: Self conditioned closure: moment of order 2 for different values of $\Delta\tau$. Cut-off is $s = 12$.

For detailed results we choose $\Delta\tau = 2.4$ and show moments of several orders in figure 6.11

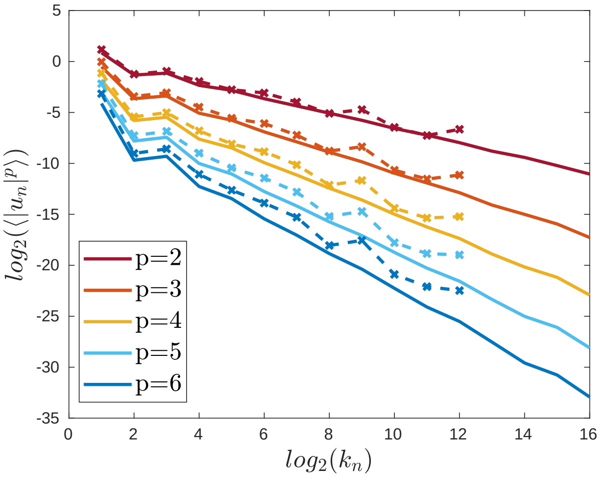


Figure 6.11: Self conditioned closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure.

In figure 6.16 we see the solution computed by this closure, in Sabra variables and plotted in original time t.

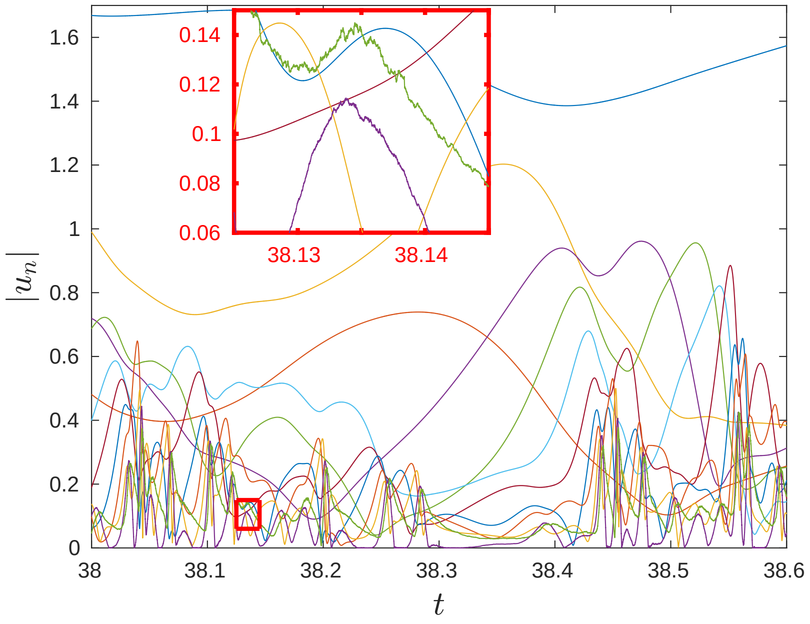
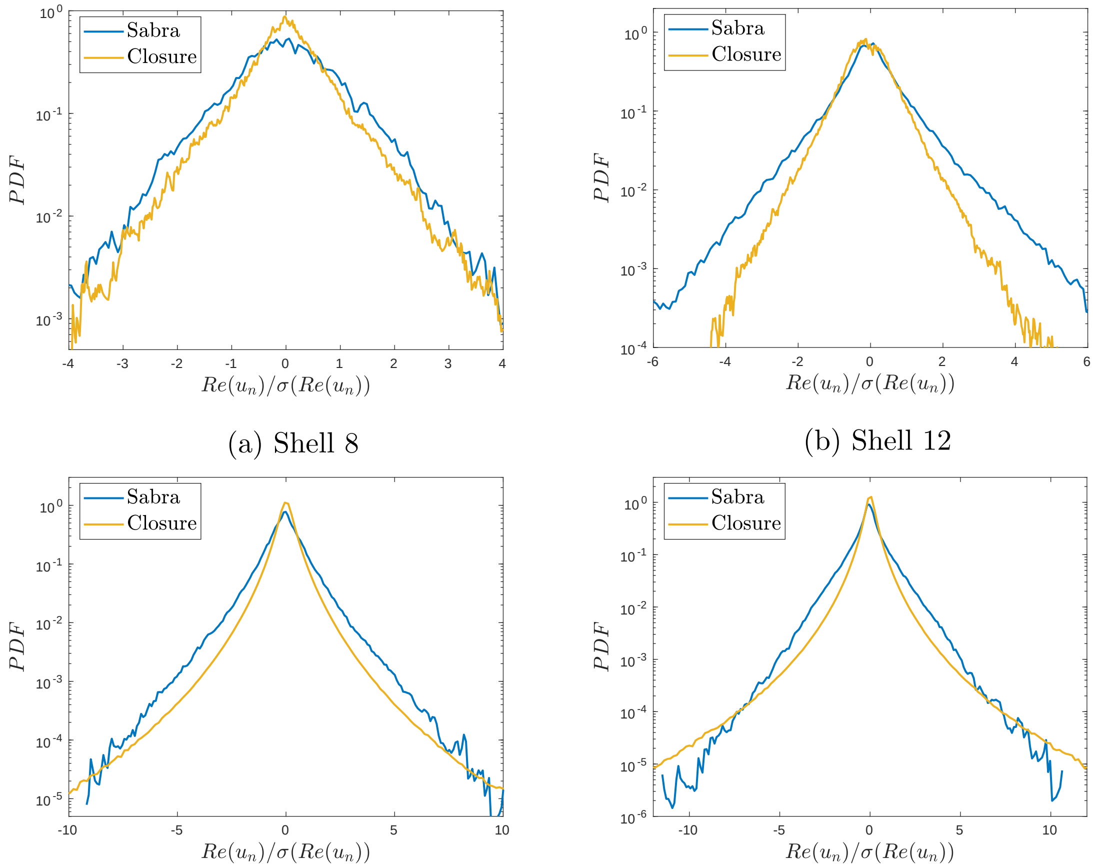


Figure 6.12: Three-closest closure: absolute value of the solution, with an inset of a zoom in.

We can see PDFs for real parts in figure 6.13 and for energy flux across different scales in 6.14.



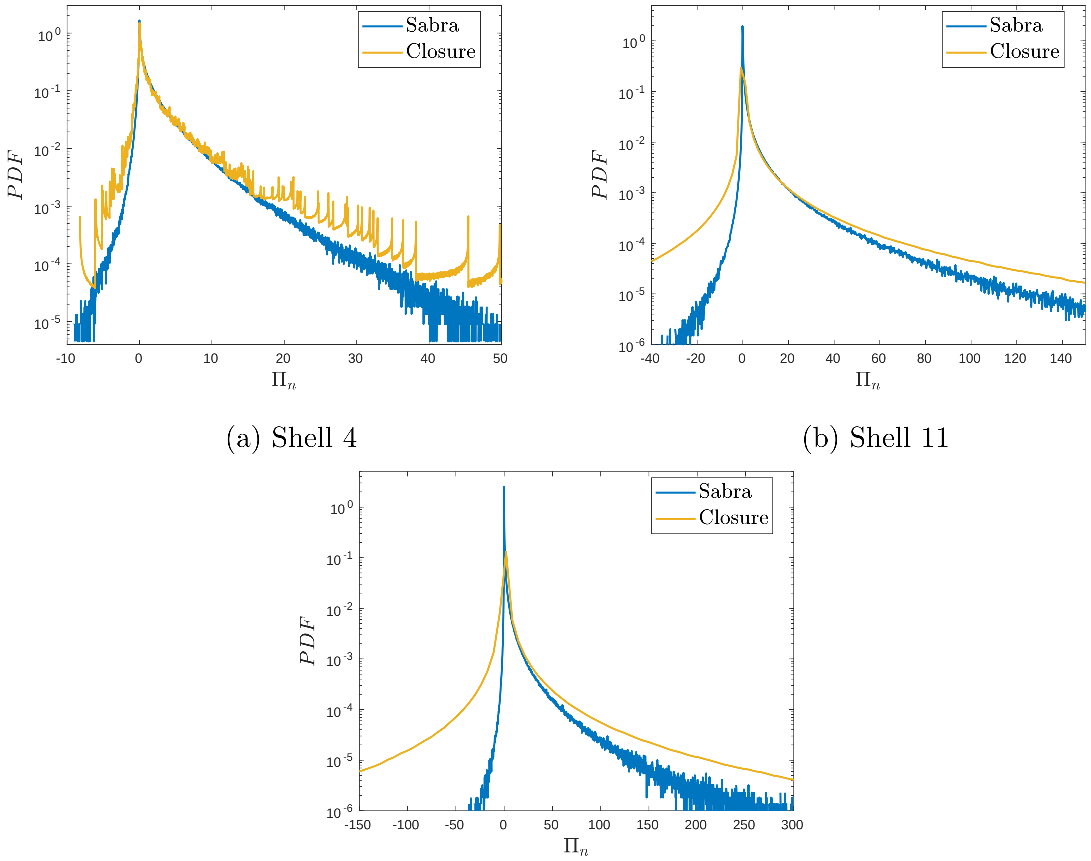
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 6.13: Self conditioned closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 6.14: Self conditioned closure: PDFs of energy flux across different shells.

6.4 Self conditioning II

It is possible that the same individual from section 5.8 is, again, wondering if these closures will perform equivalently with a different cut-off shell. For this purpose we once more run a test of the same self conditioned closure, but with a cut-off shell $s = 9$ and the same $\Delta\tau = 2.4$. Moments of different orders can be seen in figure 6.15 and similar bumps can be seen in figure 6.11

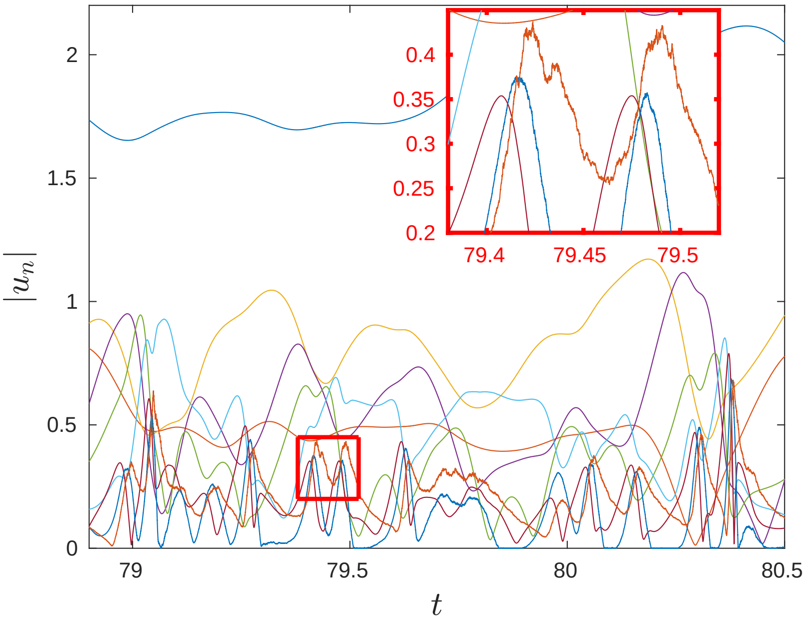


Figure 6.16: Self conditioned closure II: absolute value of the solution, with an inset of a zoom in.

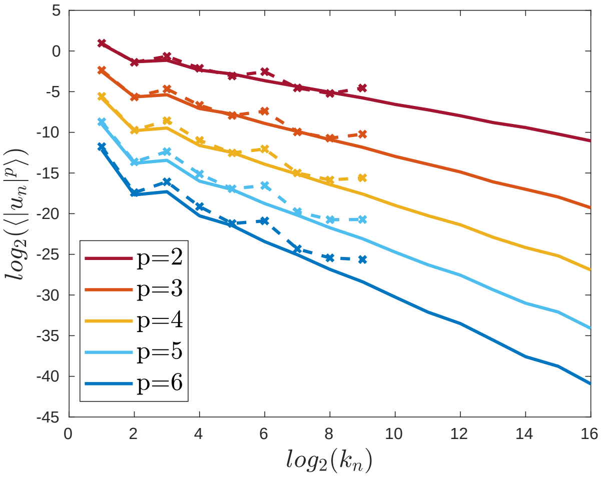
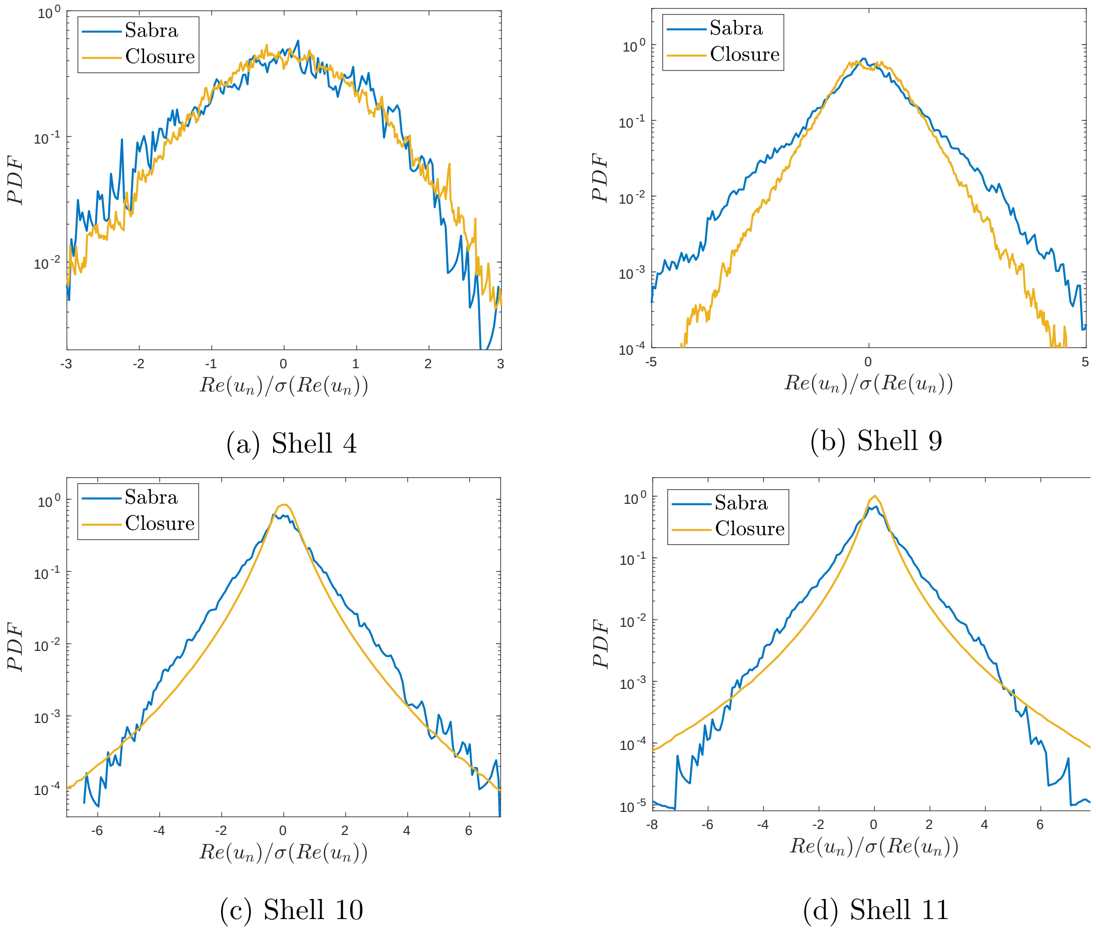


Figure 6.15: Self conditioned closure II: Moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure. Cut-off is $s = 9$.

In figure 6.16 we see the solution computed by this closure, in Sabra variables and plotted in original time t.

Figures 6.17 and 6.18 present PDFs for real parts and energy flux.



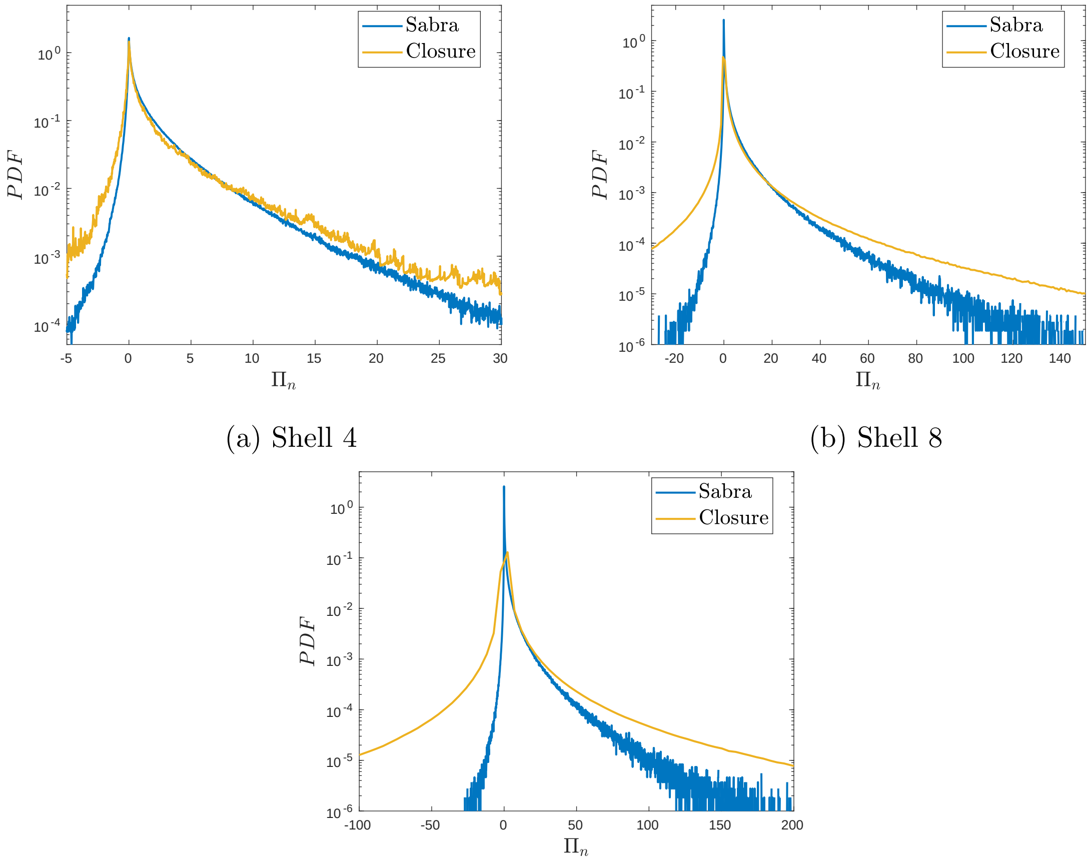
(a) Shell 4

(b) Shell 9

(c) Shell 10

(d) Shell 11

Figure 6.17: Self conditioned closure II: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 8

(c) Shell 9

Figure 6.18: Self conditioned closure II: PDFs of energy flux across different shells.

6.5 Global conditioning

In an attempt to write a more complete closure, we can increase the amount of shells in which $\mathcal{U}_0$ and $\mathcal{U}_1$ are conditioned, while also conditioning in shells at the same time τ. The density estimation for this closure contains 15 Gaussian components and required 120000 data samples.

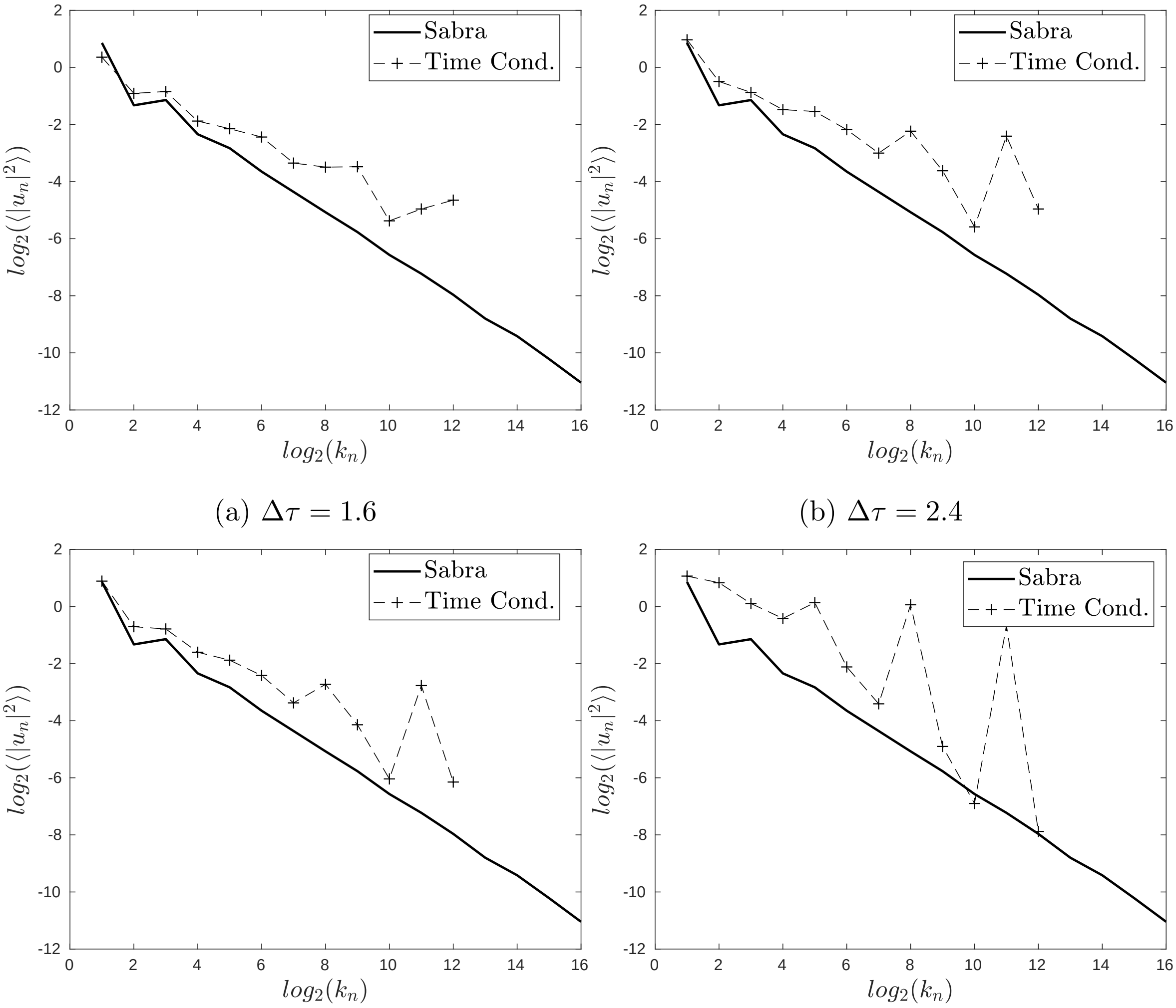
$$\mathcal{U}_0 = 2^{z_0} e^{i(z_1+\alpha_{-1}+\alpha_{-2})}, \tag{6.18}$$

$$\mathcal{U}_1 = 2^{z_2} e^{i(z_3+\alpha_0+\alpha_{-1})}, \tag{6.19}$$

$$z = (z_0, z_1, z_2, z_3) \sim g(z|z'). \tag{6.20}$$

$$\begin{aligned} z' = (&\log_2 |\mathcal{U}_{-1}(\tau)|, \Delta_{-1}(\tau), \\ &\log_2 |\mathcal{U}_{-2}(\tau)|, \Delta_{-2}(\tau), \\ &\log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|, \Delta_{-1}(\tau - \Delta\tau), \\ &\log_2 |\mathcal{U}_{-2}(\tau - \Delta\tau)|, \Delta_{-2}(\tau - \Delta\tau), \\ &\log_2 |\mathcal{U}_{0}(\tau - \Delta\tau)|, \Delta_{0}(\tau - \Delta\tau), \\ &\log_2 |\mathcal{U}_{1}(\tau - \Delta\tau)|, \Delta_{1}(\tau - \Delta\tau)) \end{aligned} \tag{6.21}$$

This closure retains information from the most recent state of the system and from $\Delta\tau$-time-units ago. In a sense, it is an attempt to understand how this closures accommodates these two pieces of information. Moments of order two can be seen in figure 6.19.



(a) $\Delta\tau = 1.6$

(b) $\Delta\tau = 2.4$

(c) $\Delta\tau = 3.2$

(d) $\Delta\tau = 6.4$

Figure 6.19: Global closure: moment of order 2 for different values of $\Delta\tau$. Cut-off is $s = 12$.

Figure 6.20 shows moments of different orders and the results detailed below correspond to $\Delta\tau = 4.8$.

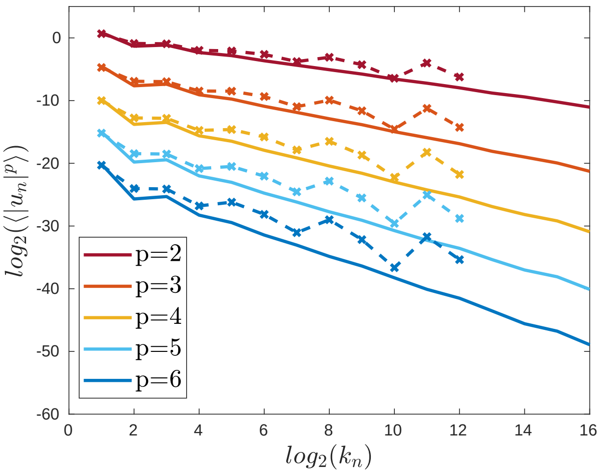
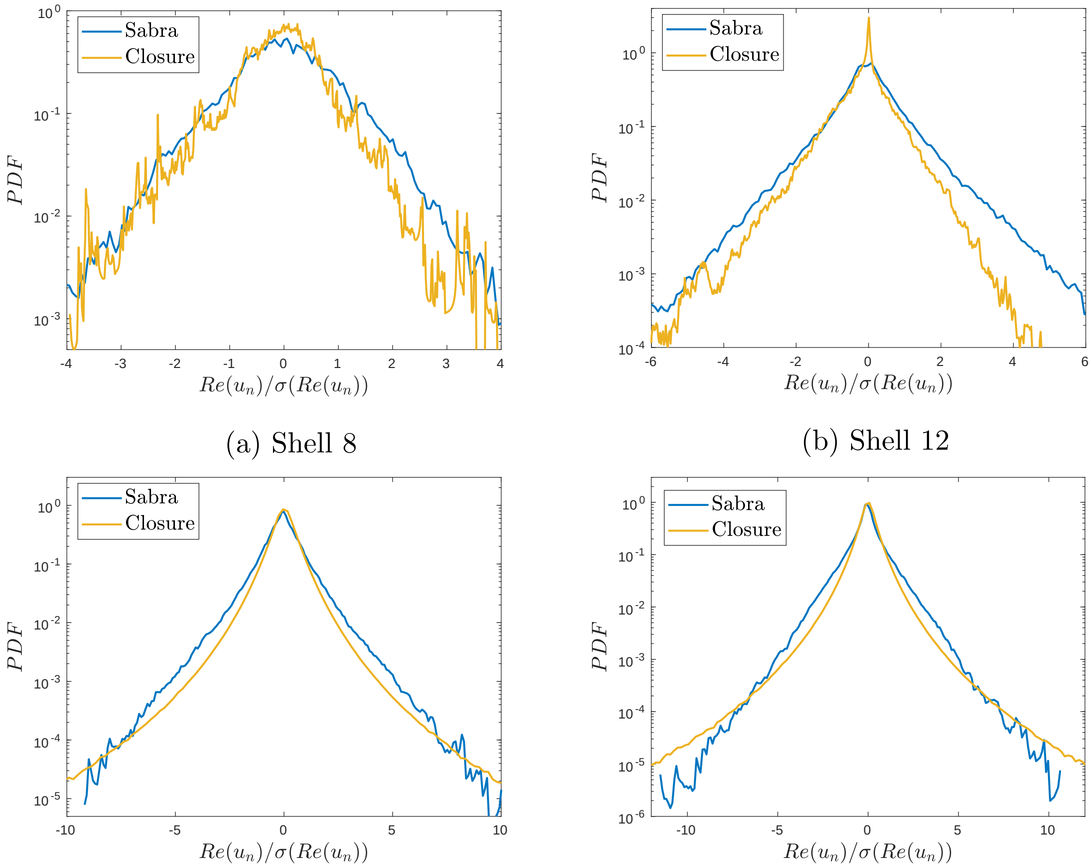


Figure 6.20: Global closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure.

It is important to note that the density estimation required for this closure is already a problem in 16 dimensions. As we remarked in section 4.2.3, Gaussian densities start to behave oddly in high dimensions. We can attest, from the difficulty we faced while adjusting this approximation, that 16 dimensions is already enough to shine a light on the limitations of GMM.

Figures 6.21 and 6.22 show PDFs of real parts and energy flux, across different scales.



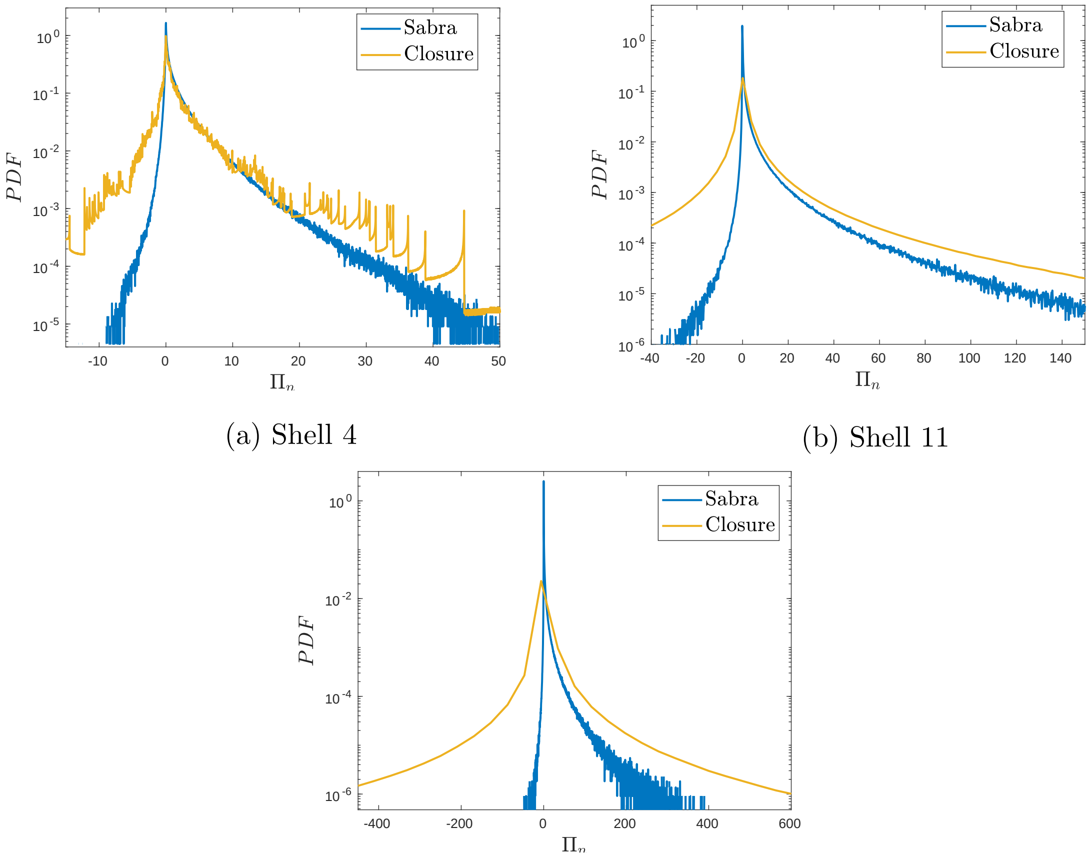
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 6.21: Global closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 6.22: Global closure: PDFs of energy flux across different shells.

6.6 Two-time history conditioning

For the last set of results of this work, we now focus on trying to understand how this closure behaves when $\mathcal{U}_0$ and $\mathcal{U}_1$ are conditioned in two different moments of its prehistory. We can write

$$\mathcal{U}_0 = 2^{z_0} e^{i(z_1+\alpha_{-1}+\alpha_{-2})}, \tag{6.22}$$

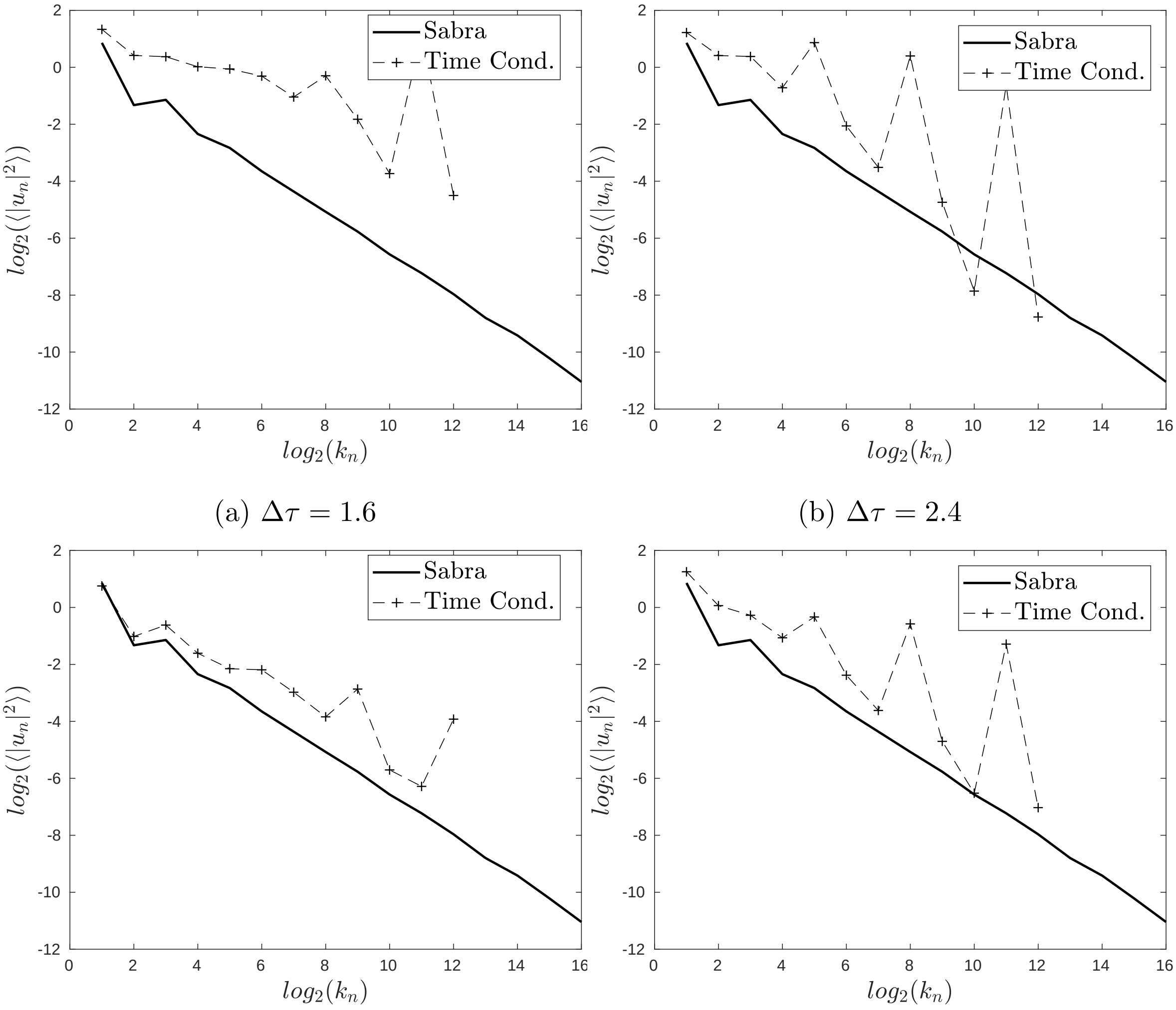
$$\mathcal{U}_1 = 2^{z_2} e^{i(z_3+\alpha_0+\alpha_{-1})}, \tag{6.23}$$

$$z = (z_0, z_1, z_2, z_3) \sim g(z|z'). \tag{6.24}$$

$$
\begin{aligned}
z' = (&\log_2 |\mathcal{U}_{-1}(\tau)|, \Delta_{-1}(\tau), \\
&\log_2 |\mathcal{U}_{-2}(\tau)|, \Delta_{-2}(\tau), \\
&\log_2 |\mathcal{U}_{-1}(\tau - \Delta\tau)|, \Delta_{-1}(\tau - \Delta\tau), \\
&\log_2 |\mathcal{U}_{-2}(\tau - \Delta\tau)|, \Delta_{-2}(\tau - \Delta\tau), \\
&\log_2 |\mathcal{U}_{0}(\tau - \Delta\tau)|, \Delta_{0}(\tau - \Delta\tau), \\
&\log_2 |\mathcal{U}_{1}(\tau - \Delta\tau)|, \Delta_{1}(\tau - \Delta\tau)), \\
&\log_2 |\mathcal{U}_{-1}(\tau - 2\Delta\tau)|, \Delta_{-1}(\tau - 2\Delta\tau), \\
&\log_2 |\mathcal{U}_{-2}(\tau - 2\Delta\tau)|, \Delta_{-2}(\tau - 2\Delta\tau), \\
&\log_2 |\mathcal{U}_{0}(\tau - 2\Delta\tau)|, \Delta_{0}(\tau - 2\Delta\tau), \\
&\log_2 |\mathcal{U}_{1}(\tau - 2\Delta\tau)|, \Delta_{1}(\tau - 2\Delta\tau)).
\end{aligned}
\tag{6.25}
$$

The previous closure, which involved an estimation in 16 dimensions, was already challenging. This is a density estimation in 24 dimensions. It required 160000 samples and 40 Gaussian components, and the results, albeit reasonable, are still not ideally smooth. In fact, we managed to use an entire 1TB of RAM memory and not reach ideal results in this density estimation problem.

Figure 6.23 shows moments of order two for different values of $\Delta\tau$.



(a) $\Delta\tau = 1.6$

(b) $\Delta\tau = 2.4$

(c) $\Delta\tau = 3.2$

(d) $\Delta\tau = 6.4$

Figure 6.23: Two-time history closure: moment of order 2 for different values of $\Delta\tau$. Cut-off is $s = 12$.

For $\Delta\tau = 4.8$ we show detailed results below. Figure 6.24 shows moments of several orders, while figures 6.25 and 6.26 show PDFs for real parts and energy flux.

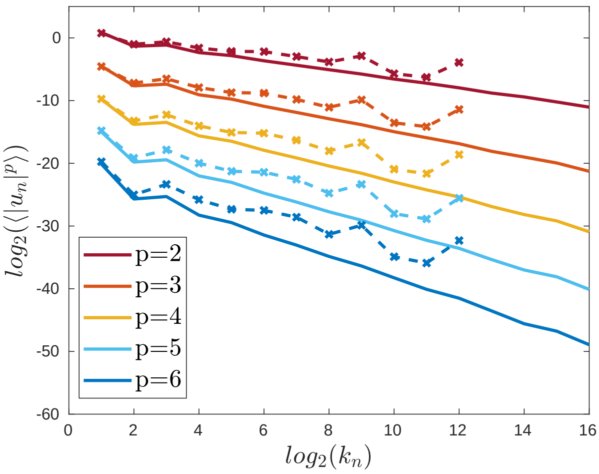
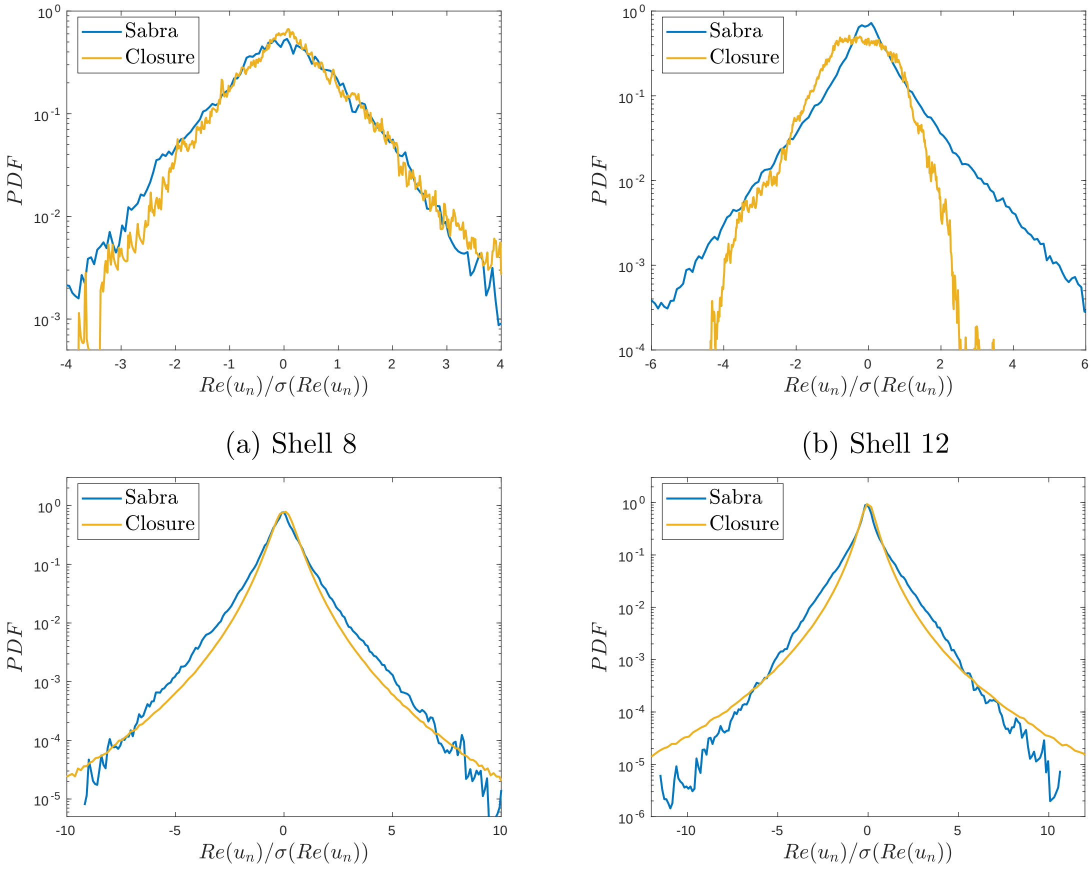


Figure 6.24: Two-time history closure: moments of order between 2 and 6, with a vertical shift for clarity. Solid lines are moments for Sabra (full model) and dashed lines are the current closure.



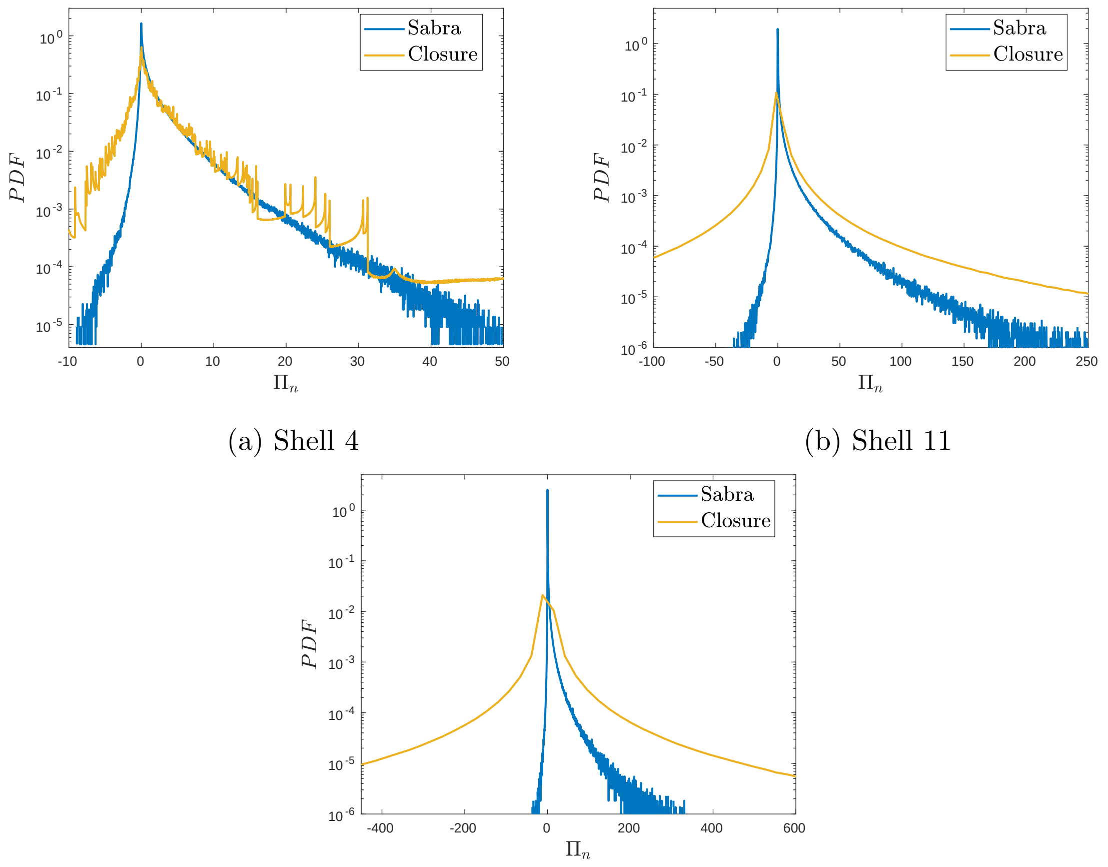
(a) Shell 8

(b) Shell 12

(c) Shell 13

(d) Shell 14

Figure 6.25: Two-time history closure: normalized PDFs of real part of different shells.



(a) Shell 4

(b) Shell 11

(c) Shell 12

Figure 6.26: Two-time history closure: PDFs of energy flux across different shells.

6.7 In this chapter

In this second chapter of results we studied closures that incorporated phase modelling, on top of the treatment we had already done for absolute values in chapter 5. We wrote more complex and more complete closures, which led to problems of density estimation in higher dimensions.

On one hand, the fact that these closures display backscattering and do not blow up despite presenting heavy negative fluxes is a positive thing. On the other hand, statistics were mostly not accurately recovered.

Figure 6.27 shows that we have completed our path.

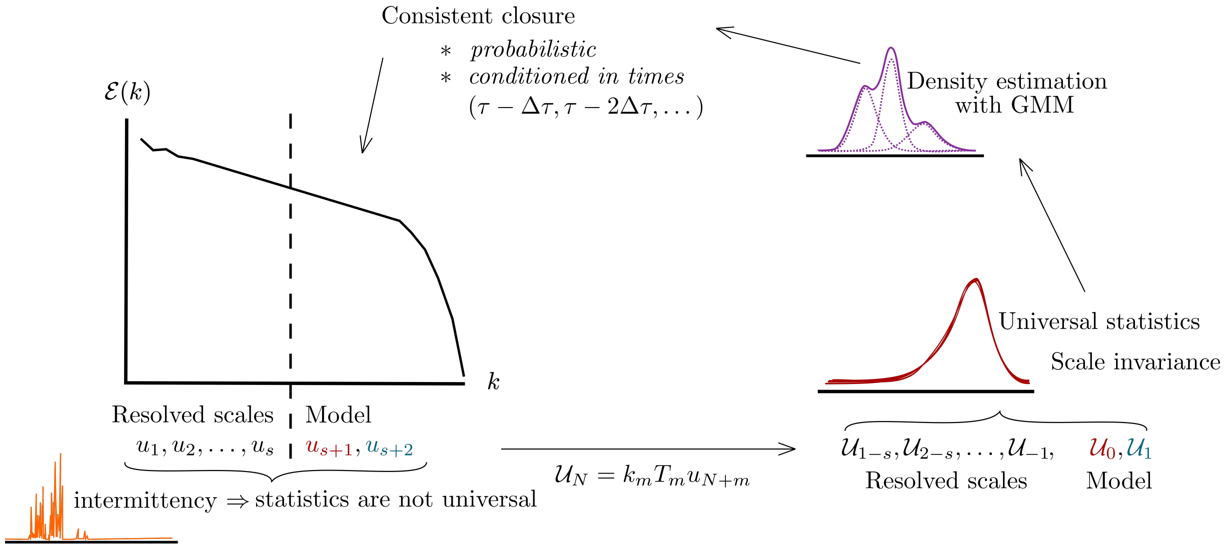


Figure 6.27: We have arrived.

Chapter 7

Conclusions

We are interested in writing probabilistic, time-correlated closures for small scales of motion in the context of turbulence in fluids. Working on a model of turbulence, the Sabra model, we identify which scales need modelling and which ones can be simulated when we do not wish to simulate the entire inertial range. However, velocity's statistics in Sabra are not universal across scales, so we apply a spatio-temporal rescaling that restores universality and reveals hidden symmetries. We then use a density estimation process to approximate the density of rescaled closure variables, use the approximation to (conditionally) sample new instances of data from this density, and, lastly, use these new instances to evolve reduced models in time.

The closures we wrote performed differently from each other. All closures from chapter 5, including Kolmogorov's closure in section 5.2, show a slight dipping in the moments of all orders for the last two shells of the model. This could mean some trouble in energy flux, given that none of the energy flux PDFs for the cut-off shell showed any energy backscattering events whatsoever. The cause for this lack of negative energy flux is the fact that phases were not included in any of the closures of chapter 5. Keeping phases fixed only provided us strictly dissipative closures, even when they were heavily conditioned in the system's history.

It is clear that including phases in the closures has a strong impact in the energy flux, given that all closures from chapter 6, even the simpler ones, without any conditioning, showed intense backscattering. It is important to note that reproducing energy flux through the cut-off shell with an adequate amount of negative values is extremely challenging. Our models in chapter 6 managed to qualitatively do so without running into numerical instabilities, but at the same time presenting results that do not match the Sabra dynamics exactly.

These inaccuracies, however, may also be related to the fact that Gaussian Mixture Models struggled to achieve reasonable approximations in higher dimensions. The closures in sections 6.5 and 6.6, which involve density estimations in dimensions 16 and 24 respectively, were barely feasible. It is important to note, though, that GMM made it easy to write closures with pre-history conditioning. They were also easy to implement and the density estimation themselves take minimum space in disk storage.

One indication that GMM is not performing optimally in chapter 6 is the fact that the PDFs for real parts, specially in shells 8 and 12 (or 4 and 9, in section 6.4), gradually grow distant from the fully resolved model. On the other hand, PDFs for closure variables are consistently recovered with heavier tails than the fully resolved model throughout chapters 5 and 6.

We can safely say that time conditioning brings more information to the closures. We saw differences between conditioning to few and to many shells with a time delay in sections 5.6 and 5.7. We also saw that conditioning closure variables to themselves brought new information to the closure, in sections 6.4 and 6.3. The reason why this is interesting is because closure variables are not exactly part of the reduced model, which only resolves scales up to s. In a sense, it is like the reduced model is evolving conditioned to the closure pre-history, instead of the model's pre-history.

One could argue that the high dimensionality of the density estimation problem could be mitigated by applying a Principal Component Analysis (PCA) [32], or other dimensionality reduction techniques. While this may very well be true, a PCA would also instantly kill the possibility of conditioning in dropped dimensions. In our framework, where the conditioning is done explicitly, we need such entries to exist in the computational arrays if we want to be able to condition the sampling to some specific variable.

It is also difficult to evaluate, out of all the closures presented here, which one performed the best. Even though works like [8] show no numerical evidence of such convergence, it would be reasonable to expect more elaborate closures to perform better, since they carry more information. There are at least two strong possibilities as to why this does not happen here. The first one is related to an unstable attractor, as discussed in [8], and the second one is related to a suboptimal performance of GMM due to high dimensionality. This introduces a (perhaps illusory) trade-off between complexity of the closures and how well they perform, which makes comparison between closures extremely tricky.

Even though we were unable to eliminate some of the discrepancies between our closures and the fully resolved model, our results are still quite sound. Moments have been recovered in a reasonable manner and PDFs for real parts mostly agree with Sabra statistics. This relative success of our approach is more evident when we note that all of these closures were built entirely on data, with little to no phenomenological predictions or additional assumptions. Moreover, it is remarkable that this approach produced equivalent closures for different cut-off shells without the need to re-learn the densities. These results are, in and of themselves, a significant contribution to the study of closure problems in the context of shell models.

Perhaps the most important contribution of this work is the clear and systematic framework for writing probabilistic, time correlated closures we have put together throughout the chapters. Our closures rely on universal statistics of a rescaled system that has no intermittent behavior. The step we called "Density estimation with GMM" in figure 6.27 can be replaced by any other more accurate, more robust, with better scalability, machine learning tool.

This is a step towards minimizing black box aspects that may come along when we introduce data science methods in our processes. When our statistics are universal there is much less doubt as to what, exactly, whatever machine learning tool we are using is learning from the data.

Generative Adversarial Networks [25], for example, are an incredibly powerful resource to estimate the underlying density of data sets and generating new instances of data. Long Short-Term Memory networks [30] are recurrent neural networks that can process time series, generating instances of data that are, at least implicitly, conditioned to its pre-history [54]. There are algorithms for finite mixtures of other types of distributions [52, 22]. Any of these tools could replace GMM in the density estimation step, and it would quite possibly yield different results providing different insights into this problem.

It may also be possible to extend this framework for closure problems for the Navier-Stokes equation, using hidden scale invariances, for example, like the ones described in [51]. This, combined with a suitable machine learning tool, has the potential for groundbreaking insight. We hope that our work will provide meaningful thoughts for future studies in closure problems, either in the context of shell models or for modelling small scales in the Navier-Stokes equation.

Bibliography

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, *On the surprising behavior of distance metrics in high dimensional space*, in Database Theory — ICDT 2001, J. Van den Bussche and V. Vianu, eds., Berlin, Heidelberg, 2001, Springer Berlin Heidelberg, pp. 420–434.

[2] Y. Agiomyrgiannakis and Y. Stylianou, *Wrapped gaussian mixture models for modeling and high-rate quantization of phase data of speech*, IEEE Transactions on Audio, Speech, and Language Processing, 17 (2009), pp. 775–786.

[3] F. Anselmet, Y. Gagne, E. J. Hopfinger, and R. A. Antonia, *High-order velocity structure functions in turbulent shear flows*, Journal of Fluid Mechanics, 140 (1984), p. 63–89.

[4] A. Beck, D. Flad, and C.-D. Munz, *Deep neural networks for data-driven les closure models*, Journal of Computational Physics, 398 (2019), p. 108910.

[5] R. Bellman, *Dynamic Programming*, Princeton University Press, 1972.

[6] L. Biferale, *Shell models of energy cascade in turbulence*, Annual Review of Fluid Mechanics, 35 (2003), pp. 441–468.

[7] L. Biferale, G. Boffetta, A. A. Mailybaev, and A. Scagliarini, *Rayleigh-taylor turbulence with singular nonuniform initial conditions*, Phys. Rev. Fluids, 3 (2018), p. 092601.

[8] L. Biferale, A. A. Mailybaev, and G. Parisi, *Optimal subgrid scheme for shell models of turbulence*, Physical Review E, 95 (2017).

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information science and statistics, Springer, 1st ed. 2006. corr. 2nd printing ed., 2006.

[10] G. Boffetta, M. Cencini, M. Falcioni, and A. Vulpiani, *Predictability: a way to characterize complexity*, Physics Reports, 356 (2002), pp. 367–474.

[11] M. J. Chorin A.J., *A Mathematical Introduction to Fluid Mechanics*, Springer, 3 ed., 1992.

[12] J. W. Deardorff, *A numerical study of three-dimensional turbulent channel flow at large reynolds numbers*, Journal of Fluid Mechanics, 41 (1970), p. 453–480.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the Royal Statistical Society. Series B (Methodological), 39 (1977), pp. 1–38.

[14] P. D. Ditlevsen, *Turbulence and Shell Models*, Cambridge University Press, 2010.

[15] T. D. Drivas, A. A. Mailybaev, and A. Raibekas, *Statistical determinism in non-lipschitz dynamical systems*, (2020).

[16] M. V. Dyke, *An Album of Fluid Motion*, Parabolic Press, Inc., 12th ed., 2008.

[17] M. L. Eaton, *Multivariate statistics: a vector space approach*, Inst of Mathematical Statistic, 2007.

[18] G. L. Eyink, *Turbulence noise*, Journal of Statistical Physics, 83 (1996), pp. 81–82.

[19] F. D. F. Nicoud, *Subgrid-scale stress modelling based on the square of the velocity gradient tensor*, Flow, Turbulence and Combustion, 62 (1999), pp. 183–200.

[20] U. Frisch, *Turbulence: The Legacy of A.N. Kolmogorov*, 11 1995.

[21] U. Frisch and R. Morf, *Intermittency in nonlinear dynamics and singularities at complex times*, Phys. Rev. A, 23 (1981), pp. 2673–2705.

[22] D. Gerogiannis, C. Nikou, and A. Likas, *The mixtures of student's t-distributions as a robust framework for rigid registration*, Image and Vision Computing, 27 (2009), pp. 1285–1294.

[23] E. O. H. S. H. Gersten, K.; Krause, *Boundary-Layer Theory*, Springer, 9ed. ed., 2017.

[24] E. B. Gledzer, *System of hydrodynamic type admitting two quadratic integrals of motion*, Soviet Physics Doklady, 18 (1973), p. 216.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds., vol. 27, Curran Associates, Inc., 2014.

[26] E. Gumbel, *Les valeurs extrêmes des distributions statistiques*, Annales de l'institut Henri Poincaré, 5 (1935), pp. 115–158.

[27] E. J. Gumbel, *The return period of flood flows*, The Annals of Mathematical Statistics, 5 (1941), pp. 115–158.

[28] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Springer, Berlin, 1996.

[29] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, *Array programming with NumPy*, Nature, 585 (2020), pp. 357–362.

[30] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Comput., 9 (1997), p. 1735–1780.

[31] S. Hoyas, A. Gil, J. Mompó-Laborda, and D. Khuong, *A large-eddy simulation of diesel-like gas jets*, International Journal of Vehicle Systems Modelling and Testing, 6 (2011), pp. 268–282.

[32] I. Jolliffe and J. Cadima, *Principal component analysis: A review and recent developments*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374 (2016), p. 20150202.

[33] B. G. Keith and T. A. Alan, *The nature of turbulent motion at large wavenumbers*, Proc. R. Soc. Lond. A, 199 (1949), pp. 238–255.

[34] R. Kohavi and D. Wolpert, *Bias plus variance decomposition for zero-one loss functions*, in Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, ICML'96, San Francisco, CA, USA, 1996, Morgan Kaufmann Publishers Inc., p. 275–283.

[35] A. N. Kolmogorov, *A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high reynolds number*, Journal of Fluid Mechanics, 13 (1962), p. 82–85.

[36] A. N. Kolmogorov, *Dissipation of energy in the locally isotropic turbulence*, Proc. R. Soc. Lond. A, 434 (1991), pp. 15–17.

[37] ——, *The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers*, Proc. R. Soc. Lond. A, 434 (1991), pp. 9–13.

[38] R. H. Kraichnan, *Intermittency in the very small scales of turbulence*, The Physics of Fluids, 10 (1967), pp. 2080–2082.

[39] M. Kurz and A. Beck, *A machine learning framework for les closure terms*, ETNA - Electronic Transactions on Numerical Analysis, 56 (2022), pp. 117–137.

[40] C. E. Leith and R. H. Kraichnan, *Predictability of turbulent flows*, Journal of Atmospheric Sciences, 29 (1972), pp. 1041 – 1058.

[41] P. L. Lewis Fry Richardson, *Weather prediction by numerical process*, Cambridge Mathematical Library, Cambridge University Press, 2 ed., 2007.

[42] W. Liu, J. Fang, S. Rolfo, C. Moulinec, and D. R. Emerson, *An iterative machine-learning framework for rans turbulence modeling*, International Journal of Heat and Fluid Flow, 90 (2021), p. 108822.

[43] E. N. Lorenz, *The predictability of a flow which possesses many scales of motion*, Tellus A, 21 (1969), pp. 289–307.

[44] V. S. L'vov, I. Procaccia, and A. L. Fairhall, *Anomalous scaling in fluid mechanics: The case of the passive scalar*, Phys. Rev. E, 50 (1994), pp. 4684–4704.

[45] V. S. L'vov, E. Podivilov, A. Pomyalov, I. Procaccia, and D. Vandembroucq, *Improved shell model of turbulence*, Phys. Rev. E, 58 (1998), pp. 1811–1822.

[46] A. A. Mailybaev, *Blowup as a driving mechanism of turbulence in shell models*, Phys. Rev. E, 87 (2013), p. 053011.

[47] A. A. Mailybaev, *Spontaneously stochastic solutions in one-dimensional inviscid systems*, Nonlinearity, 29 (2016), pp. 2238–2252.

[48] ——, *Toward analytic theory of the rayleigh–taylor instability: lessons from a toy model*, Nonlinearity, 30 (2017), pp. 2466–2484.

[49] A. A. Mailybaev, *Hidden spatiotemporal symmetries and intermittency in turbulence*, 2020.

[50] ——, *Hidden scale invariance of intermittent turbulence in a shell model*, Phys. Rev. Fluids, 6 (2021), p. L012601.

[51] A. A. Mailybaev and S. Thalabard, *Hidden scale invariance in navier–stokes intermittency*, Philosophical Transactions of the Royal Society A, 380 (2022).

[52] T. M. Nguyen and Q. M. J. Wu, *Multivariate student's-t mixture model for bounded support data*, in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 5548–5552.

[53] A. OBUKHOV, *Spectral energy distribution in a turbulent flow*, Izv. Akad. Nauk. SSSR. Ser.Geogr. i. Geofiz, 5 (1941), pp. 453–466.

[54] G. Ortali, A. Corbetta, G. Rozza, and F. Toschi, *Towards a numerical proof of turbulence closure*, 2022.

[55] G. Paladin and A. Vulpiani, *Anomalous scaling laws in multifractal objects*, Physics Reports, 156 (1987), pp. 147–225.

[56] T. N. Palmer, A. Döring, and G. Seregin, *The real butterfly effect*, Nonlinearity, 27 (2014), pp. R123–R141.

[57] G. Parisi and U. Frisch, *On the singularity structure of fully developed turbulence in turbulence and predictability in geophysical fluid dynamics and climate dynamics*, NTurbulence and Predictability of Geophysical Flows and Climate Dynamics, 88 (1985).

[58] E. Parzen, *On Estimation of a Probability Density Function and Mode*, The Annals of Mathematical Statistics, 33 (1962), pp. 1065 – 1076.

[59] K. Pearson, *Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material*, Philosophical Transactions of the Royal Society of London Series A, 186 (1895), pp. 343–414.

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

[61] S. B. Pope, *Turbulent Flows*, Cambridge University Press, 1 ed., 2000.

[62] J. V. Psutka and J. Psutka, *Sample size for maximum-likelihood estimates of gaussian model depending on dimensionality of pattern space*, Pattern Recognition, 91 (2019), pp. 25–33.

[63] O. Reynolds, *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*, Philosophical Transactions of the Royal Society of London, 174 (1883), pp. 935–982.

[64] ——, *On the dynamical theory of incompressible viscous fluids and the determination of the criterion*, Philosophical Transactions of the Royal Society of London A, (1895), pp. 123–164.

[65] M. Rosenblatt, *Remarks on Some Nonparametric Estimates of a Density Function*, The Annals of Mathematical Statistics, 27 (1956), pp. 832 – 837.

[66] A. Roy, S. K. Parui, and U. Roy, *Swgmm: A semi-wrapped gaussian mixture model for clustering of circular—linear data*, Pattern Anal. Appl., 19 (2016), p. 631–645.

[67] D. Ruelle, *Microscopic fluctuations and turbulence*, Physics Letters A, 72 (1979), pp. 81–82.

[68] U. Schumann, *Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli*, Journal of Computational Physics, 18 (1975), pp. 376–404.

[69] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*, Wiley Series in Probability and Statistics, Wiley, 1992.

[70] A. N. Shiryayev, *15. On The Empirical Determination of A Distribution Law*, Springer Netherlands, Dordrecht, 1992, pp. 139–146.

[71] J. Smagorinsky, *General circulation experiments with the primitive equations: I. the basic experiment*, Monthly Weather Review, 91 (1963), pp. 99 – 164.

[72] N. Smirnov, *On the estimation of discrepancy between empirical curves of distribution for two independent samples*, Bulletin Mathématique de LUniversité de Moscow, 2 (1939), pp. 279 – 281.

[73] A. Staicu, *Intermittency in turbulence*, PhD thesis, Applied Physics, 2002. Proefschrift.

[74] G. G. Stokes, *On the effect of the internal friction of fluids on the motion of pendulums*, Transactions of the Cambridge Philosophical Society, 9 (1851), p. 8.

[75] R. O. P. E. Stork, David G.;Duda, *Pattern classification and scene analysis. Part 1, Pattern classification*, Wiley, 2nd edition ed., 2000.

[76] S. Thalabard, J. Bec, and A. Mailybaev, *From the butterfly effect to spontaneous stochasticity in singular shear flows*, Communications Physics, 3 (2020).

[77] K. S. Thorne, *Modern classical physics: optics, fluids, plasmas, elasticity, relativity, and statistical physics*, Princeton University Press, 2017.

[78] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods, 17 (2020), pp. 261–272.

[79] U. von Luxburg and B. Schölkopf, *Statistical learning theory: Models, concepts, and results*, in Inductive Logic, D. M. Gabbay, S. Hartmann, and J. Woods, eds., vol. 10 of Handbook of the History of Logic, North-Holland, 2011, pp. 651–706.

[80] L. Xu and M. I. Jordan, *On convergence properties of the em algorithm for gaussian mixtures*, Neural Computation, 8 (1996), pp. 129–151.

[81] M. Yamada and K. Ohkitani, *Lyapunov spectrum of a model of two-dimensional turbulence*, Phys. Rev. Lett., 60 (1988), pp. 983–986.

[82] T. Yasuda and J. C. Vassilicos, *Spatio-temporal intermittency of the turbulent energy cascade*, Journal of Fluid Mechanics, 853 (2018), p. 235–252.

[83] Y. Zhao, H. D. Akolekar, J. Weatheritt, V. Michelassi, and R. D. Sandberg, *Rans turbulence model development using cfd-driven machine learning*, Journal of Computational Physics, 411 (2020), p. 109413.

[84] Y. Zhou, *Turbulence theories and statistical closure approaches*, Physics Reports, 935 (2021), pp. 1–117. Turbulence theories and statistical closure approaches.