Synthetic turbulence generation using statistical methods

Samy Braik

Supervisor: Aurélien Larcher[1], Jonathan Viquerat[1], Fabien Duval[2], Aubin Brunel[2]

[1]CEMEF, Mines Paris – PSL
[2]ANSR

September 29, 2025

Contents

1 Introduction

Turbulence describes the unpredictable and complex motion of a fluid, characterized by chaotic vortices spanning a wide range of scales. The behavior of a fluid is commonly quantified by the Reynolds number (Re): the higher the Re, the more turbulent the flow. Turbulence plays a crucial role in many natural and engineering systems, from atmospheric flows to industrial mixing, and its multiscale nature makes it challenging to model accurately.

In this work, we focus on an idealized type of turbulence known as Homogeneous and Isotropic Turbulence (HIT). HIT is widely used in fundamental turbulence studies because it isolates the intrinsic statistical properties of turbulence without the complications of boundaries or anisotropies. This flow exhibits two key properties: it is homogeneous, meaning statistical quantities (mean, variance, correlation functions) are independent of spatial position, and isotropic, meaning these statistics remain invariant under rotations or reflections of the coordinate system. These characteristics make HIT a convenient test case for validating computational models of turbulent velocity fields.
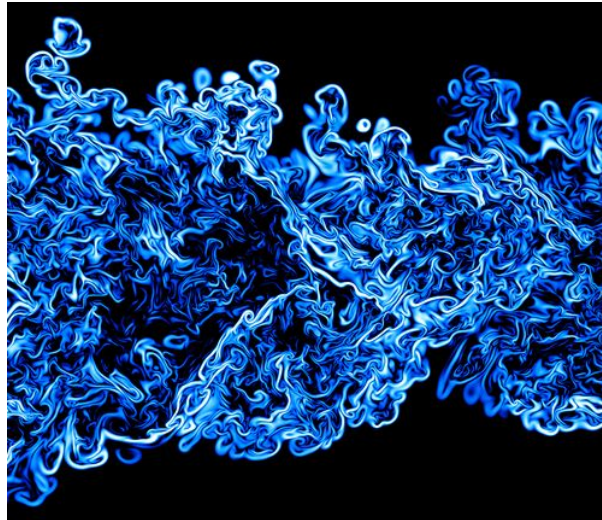

Figure 1: Slice through the field of the scalar dissipation reveals the small-scale structure of turbulence from CNRS UMR 6614 CORIA and JSC

When the goal is to use statistical methods to produce turbulent velocity field two paths are worth considering. The first one is to simply use a statistical method that generate the velocity field directly. This is done in [9],[8] or [6].
Here, generative models seem the most suitable. Two major difficulties could be encountered, the huge cost and the poor generalization of these approaches. Indeed, in order to train such a model, there is a need for highly resolved turbulence usually produced with Direct Numerical Simulation (DNS) which are costly numerical methods. Apart from that, there is the natural drawback of the method itself that can't be sidelined. The poor generalization of the method stem from the fact that usually, the models are trained on fields with a specific Reynolds number which leads to poor robustness if the turbulence parameters are changed.
The other path would be to start from an already existent method. For example in the case of Reynolds-averaged Navier–Stokes equations (RANS) technique (see [2] for details), we can use statistical methods to learn a closure model, like it is done in [1]. This allows

a better theoretical guarantee on the produced field and also get rid of the robustness problem from the previous case.

We opted for the second path, and tried to improve the random Fourier model developed in [4].

2 Model

2.1 Random Fourier

We set ourselves in the random Fourier model developed in [4] and briefly remind it. Some papers that study turbulence start from a velocity field generated by DNS and use Fourier transform to study the field properties in the spectral space. The motivation behind this model, is to perform an inverse Fourier transform to generate a synthetic turbulent field with a good choice of Fourier coefficient.
This leads to the following expression of the velocity field

$$u^s(x,t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\hat{u}(\kappa,\omega)e^{l\psi(\kappa,\omega)}\sigma(\kappa,\omega) \right] e^{l(\kappa \cdot x + \omega t)} d\kappa d\omega \tag{1}$$

Discretizing the expression in N random Fourier modes and taking into account that the field is real, it could be written

$$u^s(x,t) = 2 \sum_{i=1}^{N} \hat{u}_n \cos(\kappa^n \cdot x + \psi_n + \omega_n t)\sigma^n \tag{2}$$

Furthermore, we place ourselves in the following work in a froze turbulence which means that we ignore the effect of time. Therefore, the expression if simplified and we have

$$u^s(x,t) = 2 \sum_{n=1}^{N} \hat{u}_n \cos(\kappa^n \cdot x + \psi_n)\sigma^n \tag{3}$$

For each nth Fourier mode associated with the wave vector κ^n, $\hat{u}_n$ is the amplitude, ψ_n the phase, σ^n the direction and ω_n the time-frequency.
Like previously mentioned, a good choice of the Fourier coefficient is needed in order to produce a realistic velocity field. [4] features a discussion on the choice of $\hat{u}, \kappa, \psi_n$ and σ.
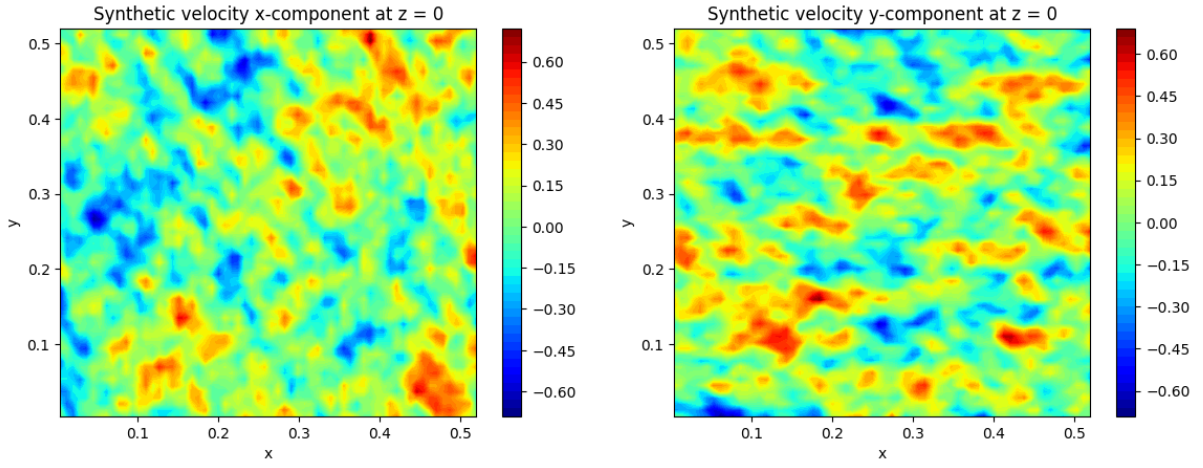
Figure 2: Synthetic velocity obtain using the Random Fourier model
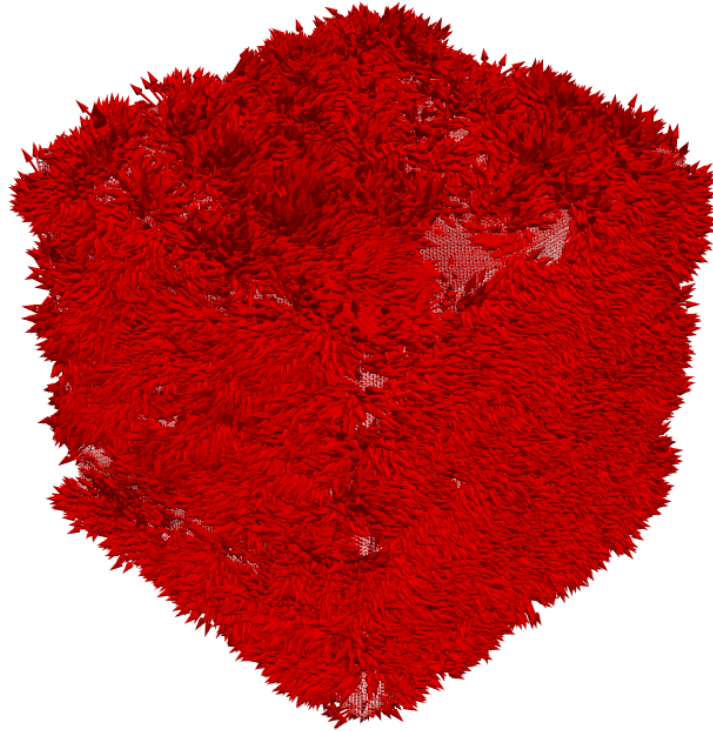

Figure 3: Synthetic velocity obtain using the Random Fourier model represented by arrows
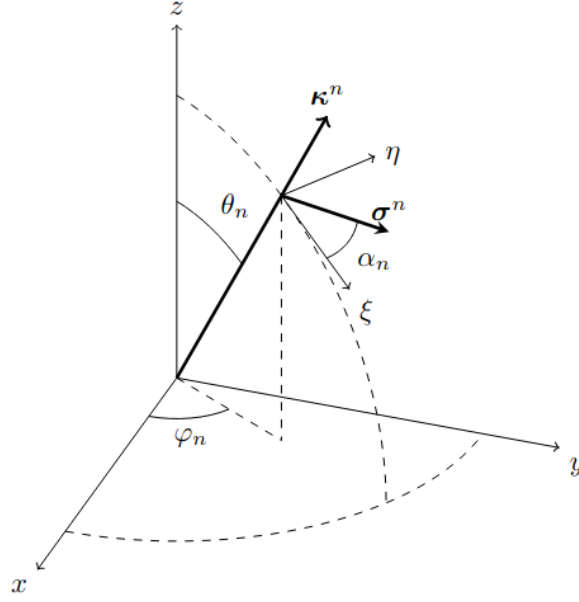
4

2.2 Coefficient choice

Figure 4: Wave vector geometry of the nth Fourier mode from [4]

To ensure the statistical isotropy of the generated field, the wave vector κ is chosen randomly on the half-sphere. This leads to the following components choices

$$\kappa_1 = \sin(\theta)\cos(\varphi) \tag{4}$$
$$\kappa_2 = \sin(\theta)\sin(\varphi) \tag{5}$$
$$\kappa_2 = \cos(\theta) \tag{6}$$

with the angles chosen randomly according to the following probability density functions $f_\theta(x) = \frac{\sin(x)}{2}$ and $f_\varphi(x) = \frac{1}{2\pi}\mathbb{1}_{[0,2\pi]}$.

Regarding the choice of σ, it is such that $\kappa \cdot \sigma = 0$ is verified, which is implied by the divergence-free condition $\nabla \cdot u^s = 0$.

Therefore,

$$\sigma_1 = \cos(\varphi)\cos(\theta)\cos(\alpha) - \sin(\varphi)\sin(\alpha) \tag{7}$$
$$\sigma_2 = \sin(\varphi)\cos(\theta)\cos(\alpha) + \cos(\theta)\sin(\alpha) \tag{8}$$
$$\sigma_3 = -\sin(\theta)\cos(\alpha) \tag{9}$$

with α chosen according to a uniform distribution on $[0, 2\pi]$ i.e. $f_\alpha(x) = \frac{1}{2\pi}\mathbb{1}_{[0,2\pi]}$. The phase coefficient ψ is randomly chosen according to $\mathcal{U}[0, 2\pi]$ to ensure spatial homogeneity.

Lastly, the amplitude $\hat{u}_n = \sqrt{E(\kappa_n)\delta\kappa_n}$ where $\delta\kappa_n = \frac{\log(\kappa_N) - \log(\kappa_1)}{N}$ and $\kappa_n = e^{(\log(\kappa_1) + n\delta\kappa_n)}$. The method to compute the prescribed energy spectrum $E(\kappa_n)$ is described in the next subsection.

2.3 Energy spectrum

Large eddies that carry energy distribute it through the energy cascade to smaller scales until it is dissipated by viscous effect at Kolmogorov scales. This energy transfer across

different scales is characterized by the energy spectrum, noted $E(\kappa)$. It describes how much kinetic energy is attached to a given wave number κ.
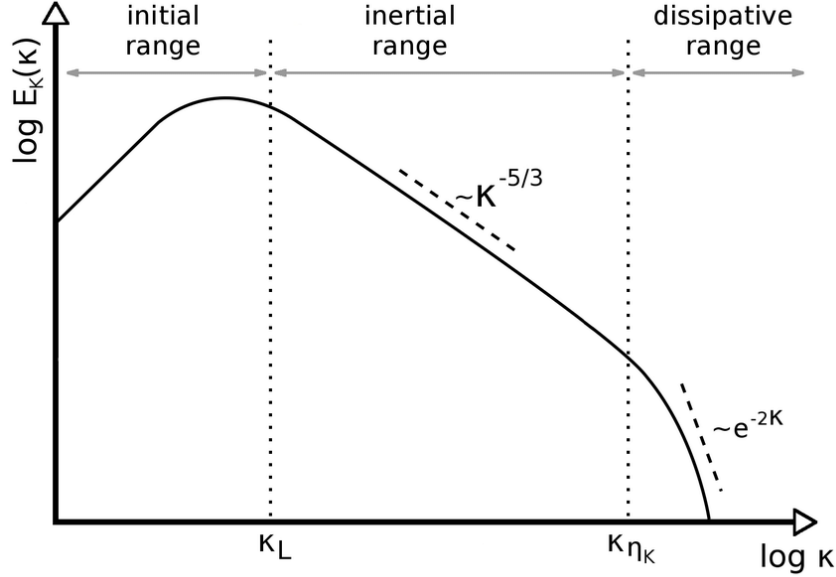

Figure 5: Energy spectrum of a turbulent flow from [7]

In the initial range, energy is injected into the flow, for instance, by an obstacle, rotating machinery, or another mechanism. The dynamics in this range are dominated by large eddies.
The inertial range is where the energy is transferred without significant dissipation. The large eddies are broken down into smaller eddies and so on. Finally, in the dissipative range, the viscous effects dominate and the remaining turbulent kinetic energy is dissipated into heat.

In the context of homogeneous isotropic turbulence, Kolmogorov gave the following formula

$$E(\kappa) = C_k \kappa^{-5/3} \varepsilon^{2/3} \tag{10}$$

where ε is the turbulent dissipation, κ the wave number and C_k the Kolmogorov constant.
Although this formula captures the behavior in the inertial range and introduces this famous $-\frac{5}{3}$ slop, it fails at reconstructing the spectrum in the initial and dissipative range i.e. at low and high wave numbers. To obtain a more complete recovery of the energy spectrum, [4] uses, among others, the von-Kármán Pao (VKP) energy spectrum.

It is defined by

$$E_{\mathrm{VKP}}(\kappa) = \frac{2}{3}\alpha_e k L_e \frac{(kL_e)^4}{[(kL_e)^2 + 1]^{17/6}} \exp(-2(\kappa L_\eta)^2) \tag{11}$$
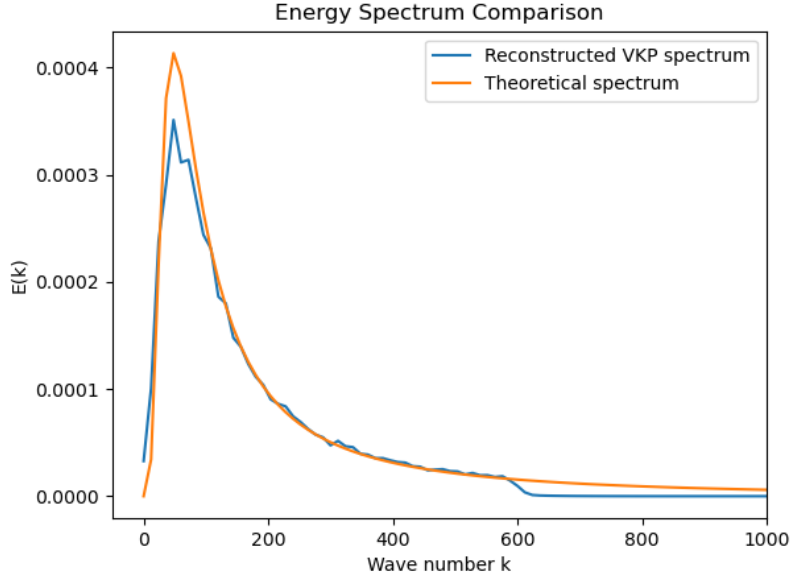
6

Figure 6: VKP energy spectrum vs theoretical spectrum

In order to accurately reconstruct the theoretical spectrum, a sufficient number of computational Fourier modes is required. In practice, you don't include infinitely many modes. We introduce a cutoff wave number which defines the smallest lengths scales that will be resolved. A trade-off naturally arises : if the cutoff is too low, we will miss information the inertial and dissipative range. On the other hand, if the cutoff is too high, we will capture the full inertial range and potentially the Kolmogorov's ones, but it will come at a significant computational cost. Moreover, the smallest scales may not always be physically meaningful, so resolving them is not necessarily worthwhile.
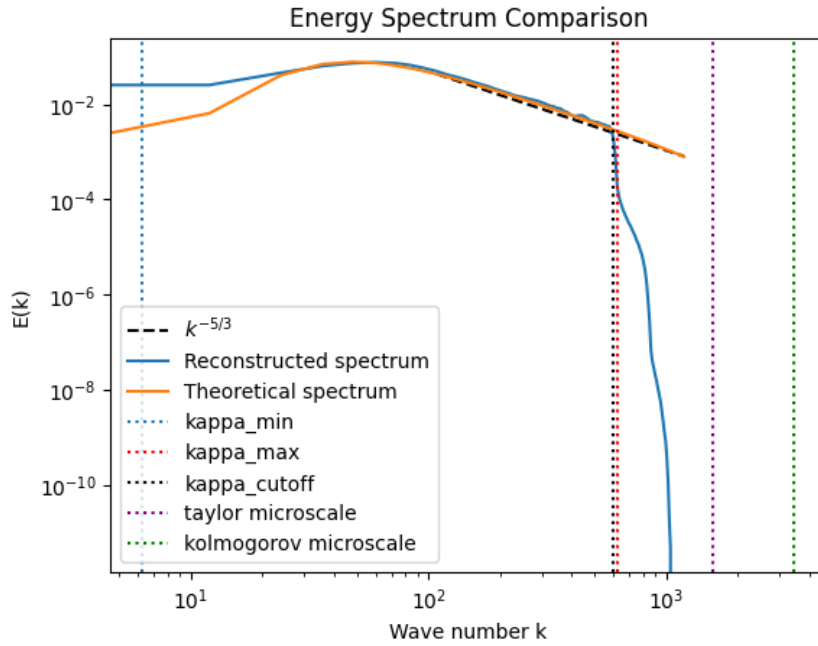

Figure 7: VKP energy spectrum vs theoretical spectrum with different scales and the cutoff

2.4 Limitations and goals

The assessment of the synthetic velocity field's quality can be assessed using several metrics. First, the reconstructed energy spectrum should match the thoretical spectrum as closely as possible. Then, for each component, the mean should be zero and the Root Mean Square (RMS) values should match the one used to build the turbulence in the first place. Lastly, the shape of the velocity increments distribution should be examined.

Definition 1 (RMS). Let $x = (x_1, \ldots x_n)$ a real vector. The RMS is defined by

$$x_{\mathrm{RMS}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \tag{12}$$

Definition 2 (Velocity increments). Let u be a velocity field. For $r > 0$ and a point x, the velocity increments are defined by

$$\delta_r u(x) = u(x+r) - u(x) \tag{13}$$

The random Fourier model gives us the following results for each metric.

Direction	Value
x	-0.0026918877847492695
y	-0.0001003775978460908
z	-0.0012576839653775096

Table 1: Velocity mean

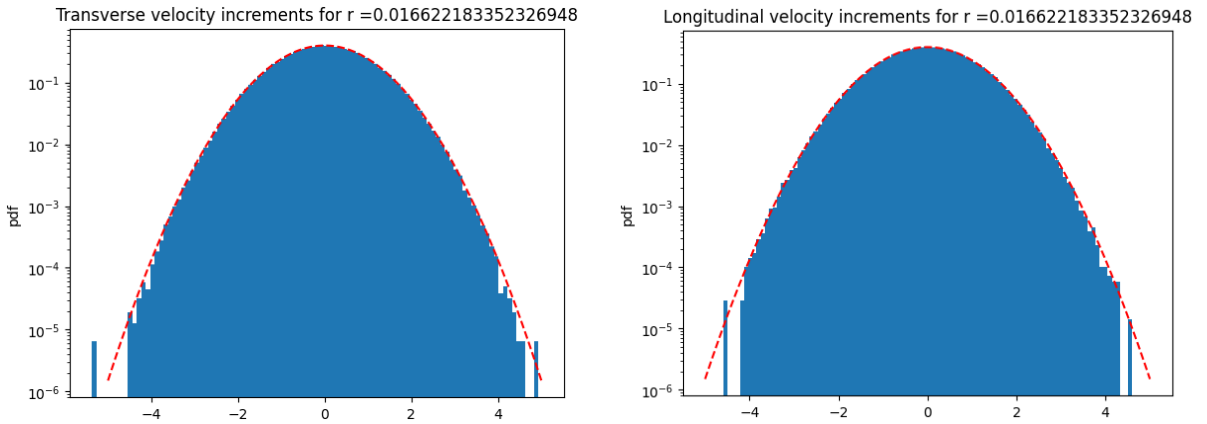| Direction	Value (expected : 0.222)
x	0.19583044946193695
y	0.17598901689052582
z	0.19306901097297668

Table 2: RMS speed

Figure 8: Velocity increments in blue with standard Gaussian in red dots

Although, the model retrieve a good energy spectrum (Figure 6) and limited anisotropy on the RMS speed, the velocity increments obtained are Gaussian. It is known that velocity increments appear to be non-Gaussian and more particularly heavy-tailed distribution. For instance [5] shows that velocity increments display heavy taildness when r is small in 13.

Furthermore, the following figures show that the more turbulent a flow is, the heavier the tails of its velocity increments are.
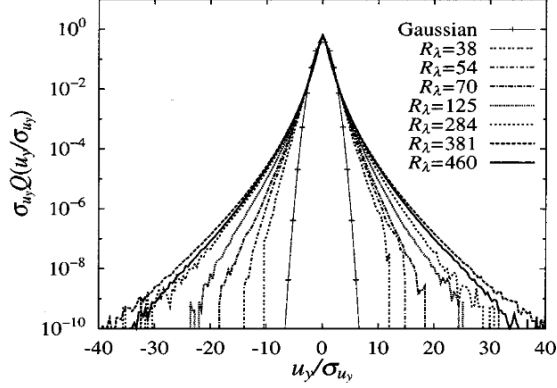

Figure 9: Variation of the transverse velocity derivative PDF with the Reynolds number from [3]
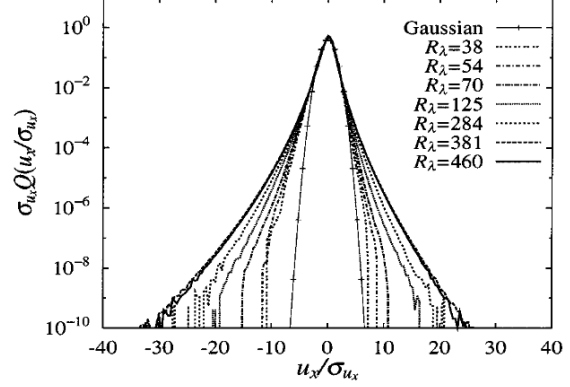

Figure 10: Variation of the longitudinal velocity derivative PDF with the Reynolds number from [3]

With the random Fourier model in mind, the goal is to identify which coefficients could produce more realistic velocity increments without disrupting the other desirable properties that are already satisfied.

3 Method

3.1 Approach

Since the goal is to retrieve heavy-tailed velocity increments, the logical criteria to look out is the kurtosis of those increments.

Definition 3 (Kurtosis). Let X be a real random variable, μ its mean and σ its standard deviation. Its kurtosis is defined by

$$\text{kurt}(X) = \mathbb{E}\left[\left(\frac{X - \mu}{\sigma}\right)\right]^2 \tag{14}$$

Remark 1. In practice, we look at the kurtosis excess or flatness which is simply $\text{kurt} - 3$.

Therefore, in order to push for heavy tailed velocity increments, we came up with a straight forward flatness loss.

Definition 4 (Flatness loss). Let F_{target} be the target flatness, $x = (x_1, \ldots, x_n)$ be the input vector.

$$\mathcal{L}_F(x, F_{\text{target}}) = \frac{1}{N}\sum_{i=1}^{N}\left(\text{kurt}(x) - 3 - F_{\text{target}}\right)^2 \tag{15}$$

To compute the energy spectrum as close as the theoretical spectrum, we simply look at the MSE with an added smoothness term. This gave us the following loss

Definition 5 (Energy spectrum loss). Let $\alpha > 0$, E_{rec} the reconstructed spectrum and E_{theory} the theoretical spectrum. We define a smoothness term by

$$S(E_{\text{rec}}, E_{\text{theory}}) = \frac{1}{N-2} \sum_{i=2}^{N-1} (E_{\text{rec},i-1} - 2E_{\text{rec},i} + E_{\text{rec},i+1}) \tag{16}$$

The loss is then defined by

$$\mathcal{L}_{ES}(E_{\text{rec}}, E_{\text{theory}}) = \frac{1}{N} \sum_{i=1}^{N} (E_{\text{rec},i} - E_{\text{theory},i})^2 + \alpha S(E_{\text{rec}}) \tag{17}$$

Definition 6 (Total loss). Let x_L be the longitudinal velocity increments, x_T the transverse velocity increments and E_{rec} the reconstructed energy spectrum.

$$\mathcal{L} = \mathcal{L}_F(x_L, F_{\text{target}}) + \mathcal{L}_F(x_T, F_{\text{target}}) + 1e3 \times \mathcal{L}_{ES}(E_{\text{rec}}, E_{\text{theory}}) \tag{18}$$

To tackle the problem, several methods were tested, but it turned out simply defining the desired parameter as `nn.Parameter` in PyTorch and directly optimizing them led to good results. From a computational perspective, this approach was also the most convenient (more on this in 4).

In order to optimize the models according to the loss 6 we use AdamW and a scheduler. The parameterization of the optimizer was mostly due to empirical evidence (see B).

Although training was typically run for 5000 epochs, convergence was usually achieved after around 2000 epochs. We worked with a target flatness of 15 and $\alpha = 0.5$.

3.2 Preliminary work

In order to check the robustness of the random Fourier model in the first place, we conducted a simple experiment. We set the target flatness as 0, meaning we look for Gaussian velocity increments. Since it is the base model velocity increments, we want to know if we will retrieve the parameterization introduce in [4] or will it move away. We look up two situations : starting from Gaussian velocity increments and heavy-tailed velocity increments.
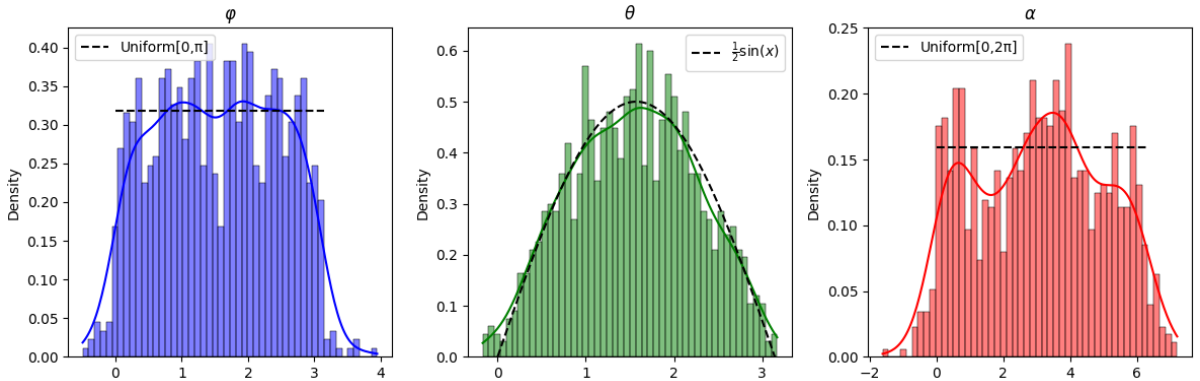

Figure 11: Angles distributions starting from Gaussian velocity increments

Here, the angles distributions stay sensibly the same, with a slight change in α. This is expected since we already satisfy the target flatness.
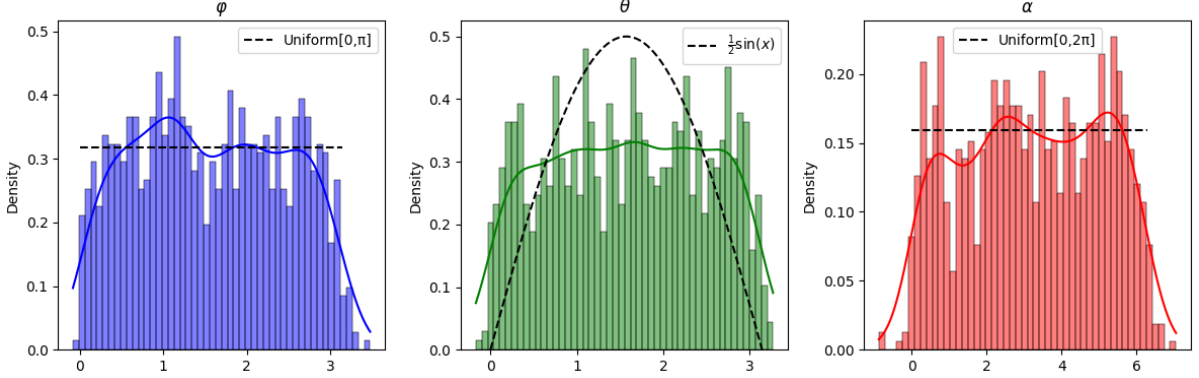

Figure 12: Angles distributions starting from heavy-tailed velocity increments

In this case, φ and α stay relatively stable but θ is drawn to a more spread distribution.

In both cases Gaussian velocity increments are indeed recovered, the RMS speed are in the same range as in 2 and the generated velocity field does average to 0.

3.3 Phase coefficient

To improve the situation, our first idea was to work on the phase parameter ψ in 3, with the idea of finding a better distribution than $\mathcal{U}([0, 2\pi])$. A simple experiment consisting of changing $\mathcal{U}[0, 2\pi]$ to arbitrary distribution like $\mathcal{N}(-10, 130)$ or $\mathcal{E}(1)$ showed that it has no effect on the shape of the velocity increments distribution. Indeed, the reason is pretty straight forward : the Central Limit Theorem (CLT).
Sampling I.I.D. values of ψ according to any of these distributions and plugging it in 3 with sufficiently large N (which is expected in our situation) leads directly to the conditions of the CLT.

Applying the approach described in 3 didn't produce satisfying results and focusing solely on ψ was ultimately abandoned.

3.4 Phase, wave vector and direction

The natural follow-up was to look at all the coefficient all together, i.e. (ψ, κ, σ), without changing the objectives nor the method. It led to heavy tailed velocity increments indeed as shown in figure 13.
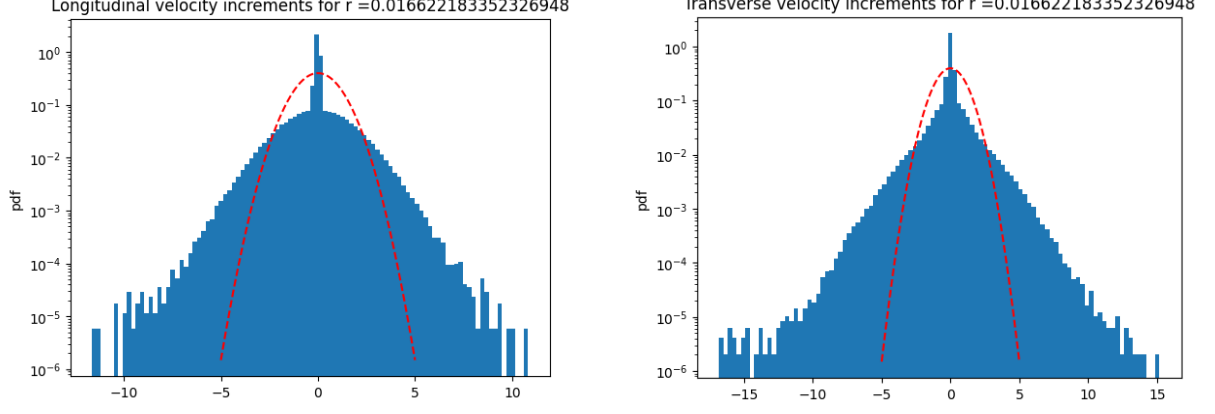
Figure 13: Velocity increments with learned coefficient (ψ, κ, σ) and standard Gaussian in red dots

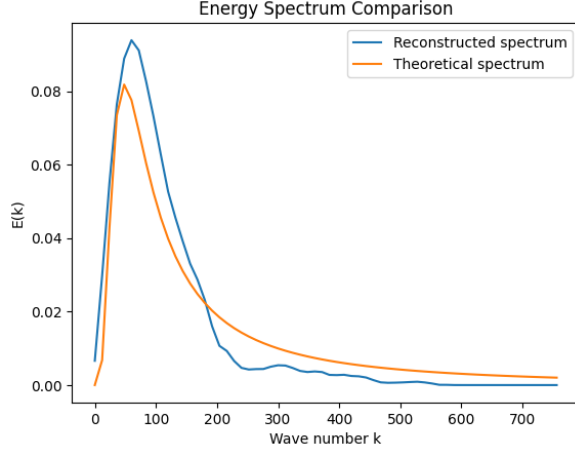The reconstructed spectrum shows a considerable bias and doesn't capture the small scales correctly.

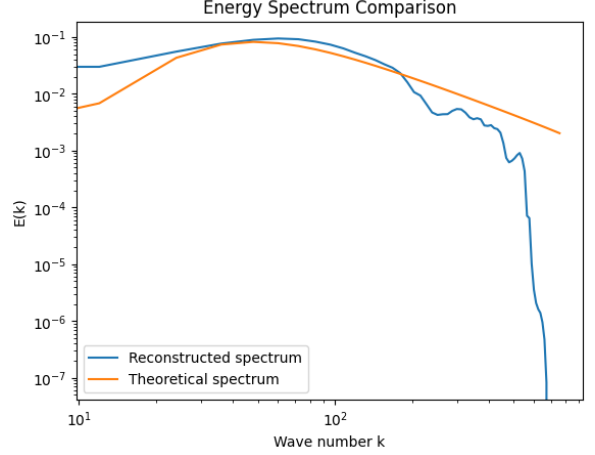

Figure 14: Energy spectrum with the learned parameters (ψ, κ, σ)

Figure 15: Energy spectrum with the learned parameters (ψ, κ, σ) with log scale

Direction	Value
x	7.787999493302777e-05
y	5.701697227777913e-05
z	-0.030603401362895966

Table 3: Velocity mean

Direction	Value (expected : 0.222)
x	0.0017423529643565416
y	0.0006628130213357508
z	0.889664888381958

Table 4: RMS speed

Although, this approach works as it produces heavy tailed velocity increments, it exhibits serious anisotropy and fails to recover a good energy spectrum.

Furthermore, there is no guarantee that the obtained κ and σ are orthogonal to each other as stated in 2.2.

Therefore, we shift our focus on the angles $(\varphi, \theta, \alpha)$ composing the κ and σ component. We aim at learning the three angles while using 4, 5,6 and 7,8,9 to preserve the identity $\kappa \cdot \sigma = 0$.

3.5 Angles

Since we want to work on the angles, we need to make sure the angles are always coherent with the geometry (see 4). Therefore, to make sure that φ, θ and α lives in their respective domain $[0, \pi], [0, \pi]$ and $[0, 2\pi]$.

We train the model on 500 modes with 100^3 points, with the turbulence parameter described in A and retrieve great velocities increments.
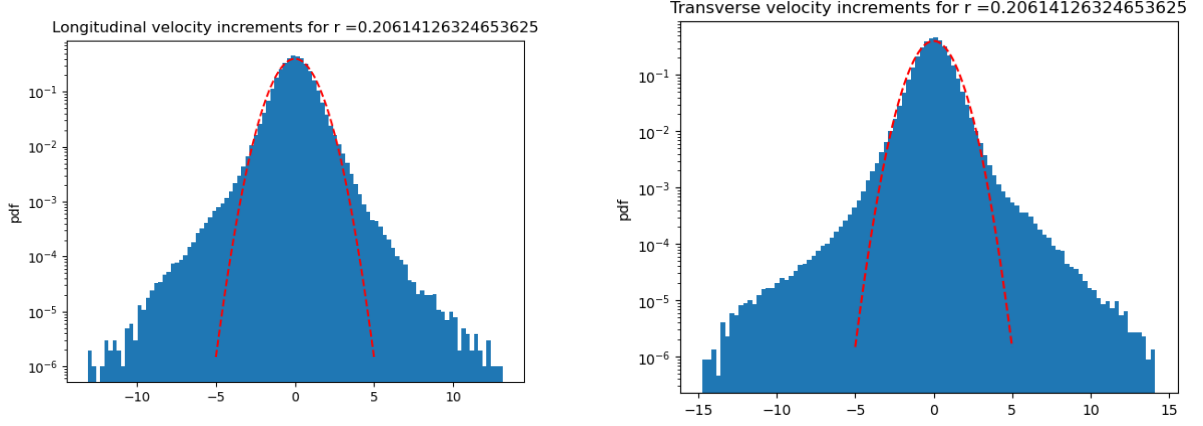

Figure 16: Velocity increments with learned angles $(\varphi, \theta, \alpha)$ and standard Gaussian in red dots

Unlike in 15, here the energy spectrum is accurately reconstructed but nowhere near the one from 4.
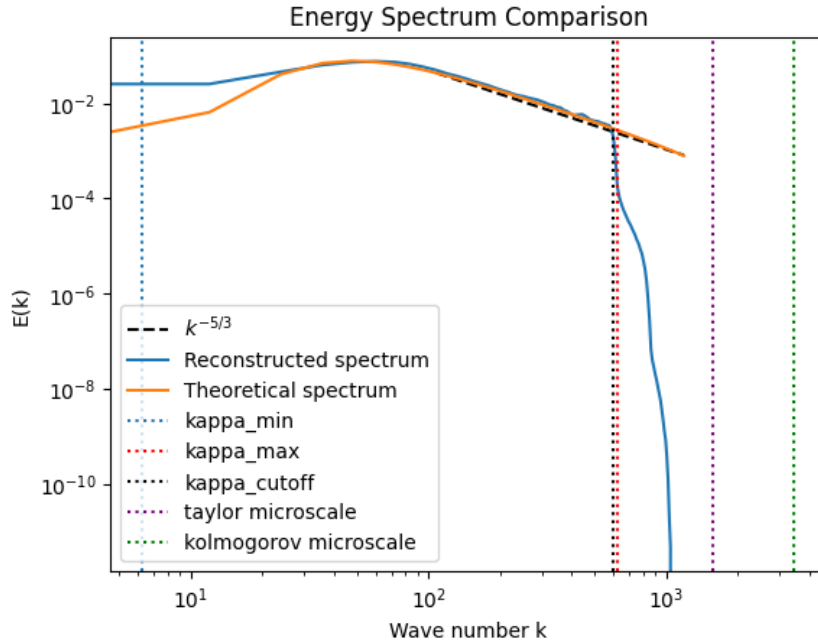
Figure 17: Energy spectrum reconstructed after training

However, a slight anisotropy is displayed in the RMS speed obtained.

We obtain the following marginal distribution.

13

Direction	Value
x	0.0025640586391091347
y	-0.0065282071009278
z	-0.0009377452079206705

Table 5: Velocity mean

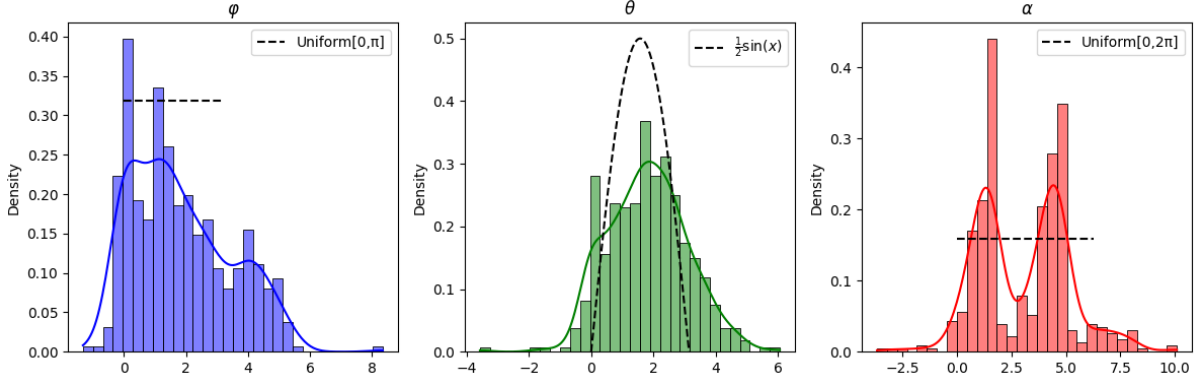| Direction	Value (expected : 0.222)
x	0.12904468178749084
y	0.3165183663368225
z	0.11406964063644409

Table 6: RMS speed

Figure 18: Marginal distribution of the learned angles with the base distribution in black dots

3.5.1 Robustness

Additionally, these angles exhibit some robustness to changes in the turbulence characteristics. When the kinetic energy is doubled (by multiplying the RMS speed in A by $\sqrt{2}$) or quadrupled. The metric obtained are comparable to those at the end of the training step. However, when the length of the periodic box is doubled, the precision on the RMS speed is lost. This effect could be due to a dilution of the resolution when the number of points is still.

4 Discussion

Different architectures were explored to address the problem and also bring new generalization possibilities, such as generating new instances of angles for a specific turbulent case. Unfortunately, significant computational challenges were encountered.

At the time of writing, I got my hands on an Nvidia A10 GPU which I am working with to try more complex solution that won't be featured in this work.

In reality, the complexity and the computational bottleneck arises mainly with the number of points considered. A low-resolution grid leads to under-resolved fields, which directly impacts the accuracy of velocity increments computed via the discretization in 13.

The code used to generate all results presented in this work will be made available at: GitHub repository

5   Conclusion

In this work, we focused on improving the generation of synthetic turbulent velocity fields using the random Fourier model, with the specific goal of producing velocity increments exhibiting heavy-tailed statistics, as observed in real turbulence. We reviewed the limitations of the standard random Fourier approach, which, while accurately reconstructing the energy spectrum and maintaining near-zero mean velocities, produces Gaussian velocity increments that fail to capture the intermittency characteristic of turbulence.

Our approach centered on directly optimizing the coefficients of the Fourier modes, and more specifically the angles defining the wave vectors and directions, to control the statistical properties of the velocity increments. By introducing a flatness-based loss in combination with an energy spectrum loss, we were able to generate heavy-tailed velocity increments while preserving the overall energy distribution. The learned angle distributions proved robust to changes in turbulence parameters and maintained reasonable isotropy, demonstrating the generality of the approach.

While optimizing the full set of coefficients (ψ, κ, σ) produced heavy tails, it led to strong anisotropy and a biased energy spectrum. Focusing on the angle parameterization allowed us to overcome these limitations, balancing the statistical accuracy of velocity increments with the physical consistency of the field.

Overall, our work highlights the potential of targeted parameter optimization in synthetic turbulence generation, providing a method for producing velocity fields that better reflect the non-Gaussian features of real turbulence. Future work could extend this approach to unfrozen turbulence, incorporating temporal effects and ultimately quit the HIT framework to look at anisotropic turbulence to get closer to realistic scenarios.

A   Turbulence parameters

To generate the turbulence several physical and spectral parameters have to be given as input for the model. Throughout the report the parameters that have been used are listed in the following table.

Parameter	Value	Unit
Number of points	64^3 or 100^3	–
Length of the periodic box	$\frac{\pi}{6}$	m
Number of modes	250 or 500	–
RMS speed	0.222	$\mathrm{m\,s^{-1}}$
Integral length scale	0.024	m
Kinetic energy	$1.5 \times (0.222^2)$	$\mathrm{m^2\,s^{-2}}$
Viscosity	1.8×10^{-5}	$\mathrm{m^2\,s^{-1}}$
Minimum wave number	$\frac{2\pi}{1.0}$	$\mathrm{m^{-1}}$
Maximum wave number	$\frac{2\pi}{0.01}$	$\mathrm{m^{-1}}$

B   Optimizer hyperparameters

```
optimizer = torch.optim.AdamW(
    model.parameters(),
    lr=1e-3,
    weight_decay=1e-4
)


scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(
    optimizer,
    mode='min',
    factor=0.5,
    patience=50,
    threshold=1e-3,
    min_lr=1e-6,
    cooldown=10
)
```

C   Variable selection

To address the question of which angles are the most influential and which one could be negligible, we simply perform a case analysis. In the same training context, we varied individual angles while keeping others froze and untrained. It gave us the following results.

Overall, the more modes the better it is. A lot of the cases are pretty close together and lead to sensibly the same losses and in practice the desired flatness and a good recovery of the energy spectrum.

RMS values are affected after the learning step, depending on which angles are considered. For the moment, adding a loss on those RMS values tend to lead to less flat tails on the output velocities increments. A trade-off between RMS and flat tails seems to arise. For

16

Table 7: Losses obtain for different parameter combination

(a) 250 modes

Parameters	Loss	RMS (x, y, z)
$(0, 0, \alpha)$	123.24456	(0.24, 0.08, 0.12)
$(0, \theta, 0)$	1.50244	(0.22, 0.11, 0.08)
$(0, \theta, \alpha)$	65.67394	(0.11, 0.22, 0.09)
$(\varphi, 0, 0)$	0.75362	(0.09, 0.24, 0.06)
$(\varphi, 0, \alpha)$	0.74764	(0.20, 0.10, 0.09)
$(\varphi, \theta, 0)$	0.75258	(0.12, 0.22, 0.04)
$(\varphi, \theta, \alpha)$	2.15906	(0.22, 0.11, 0.09)

(b) 500 modes

Parameters	Loss	RMS (x, y, z)
$(0, 0, \alpha)$	0.73864	(0.14, 0.20, 0.06)
$(0, \theta, 0)$	0.74741	(0.20, 0.11, 0.10)
$(0, \theta, \alpha)$	0.79737	(0.14, 0.21, 0.09)
$(\varphi, 0, 0)$	0.75514	(0.10, 0.22, 0.08)
$(\varphi, 0, \alpha)$	0.74716	(0.21, 0.11, 0.09)
$(\varphi, \theta, 0)$	0.74118	(0.22, 0.09, 0.11)
$(\varphi, \theta, \alpha)$	0.75355	(0.20, 0.13, 0.09)

instance, in the case where all the parameters are learned, like in the previous section, the RMS for the z direction is off (0.09 with an expected value of 0.222).

C.1 Marginal distributions
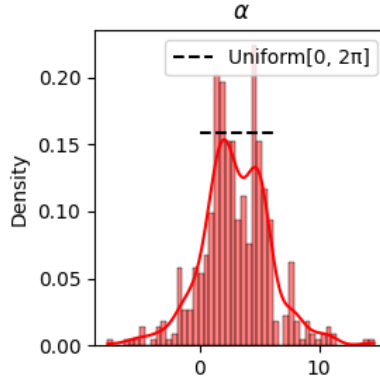
Figure 19: Marginal density of the learned parameters (α)
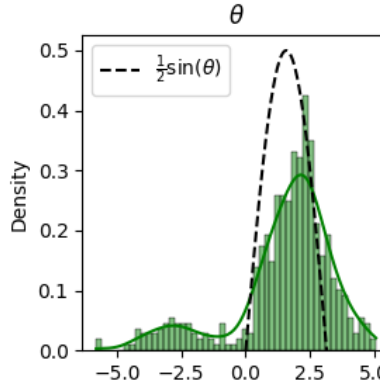
Figure 20: Marginal density of the learned parameters (θ)
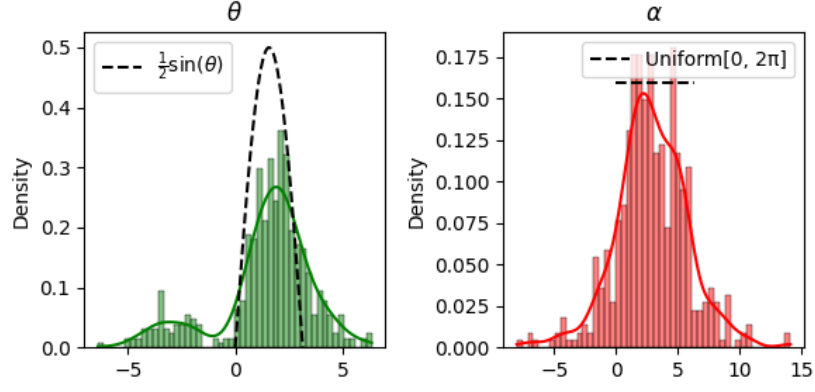
Figure 21: Marginal density of the learned parameters (θ, α)
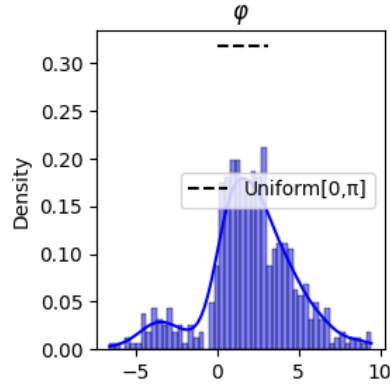
Figure 22: Marginal density of the learned parameters (φ)
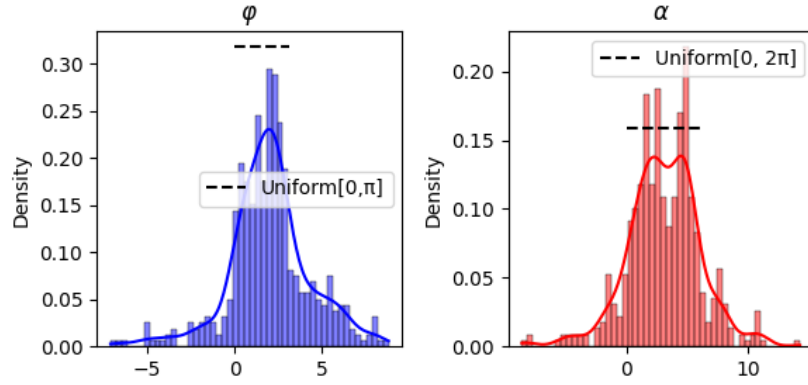
Figure 23: Marginal density of the learned parameters (φ, α)

18
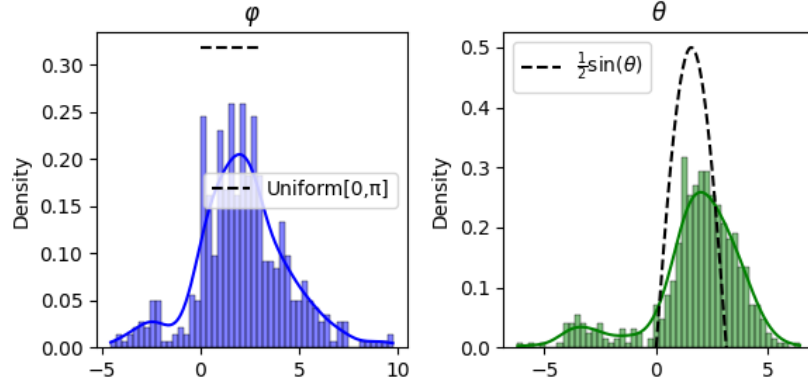
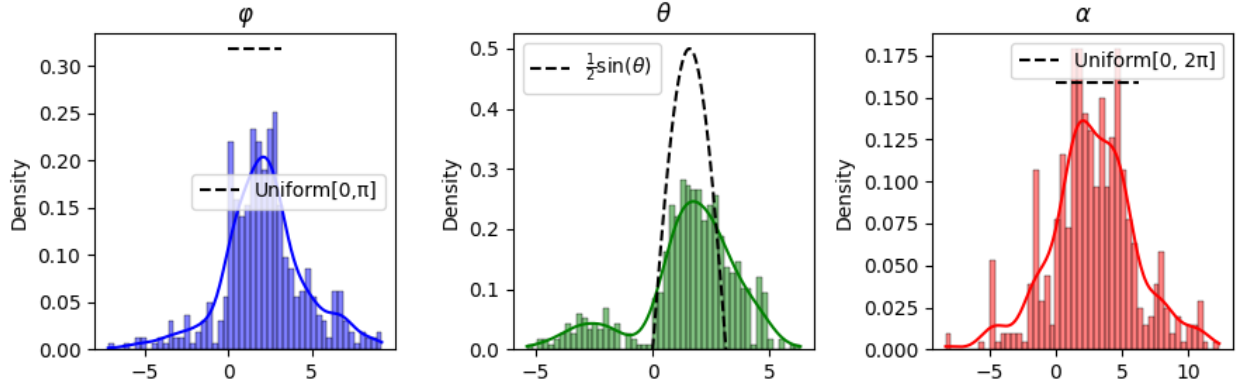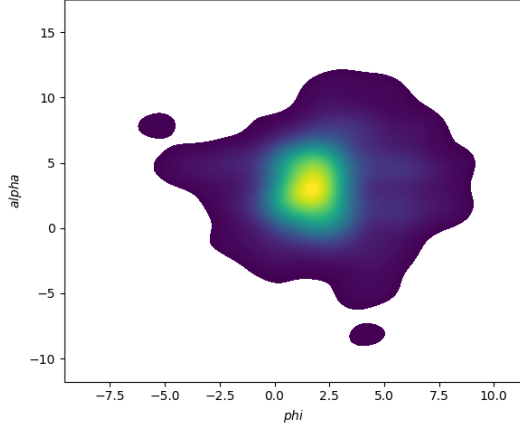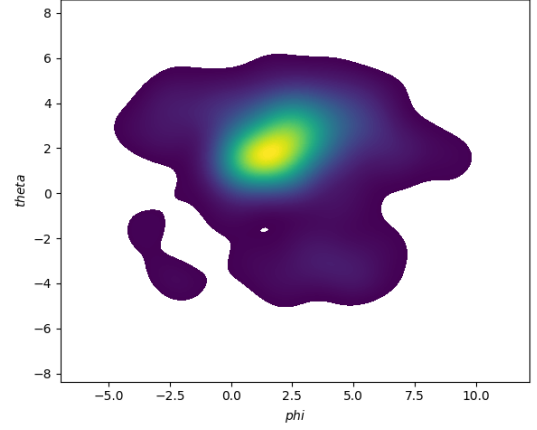Figure 24: Marginal density of the learned parameters (φ, θ)

Figure 25: Marginal density of the learned parameters $(\varphi, \theta, \alpha)$

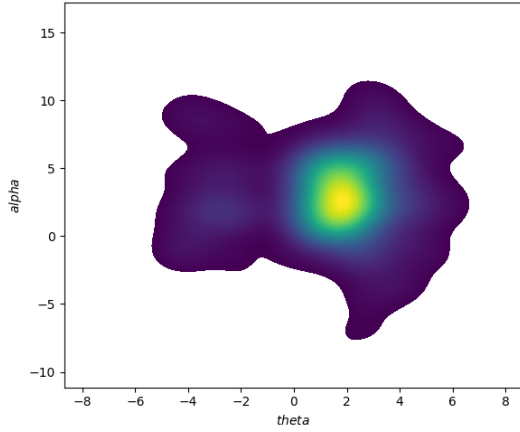We can see that among all the cases, the final marginal distributions seem pretty consistent throughout.

C.2 Joint distributions

(a) Learned parameters (φ, α)

(b) Joint distribution of (φ, θ)

(c) Joint distribution of (θ, α)
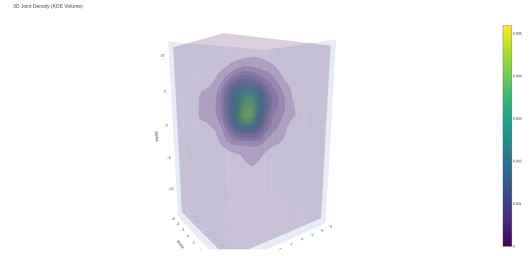
(d) Joint distribution of $(\varphi, \theta, \alpha)$

Figure 26: Joint distributions of parameter combination

D Variable selection with ψ

We repeat the previous experiment considering also the phase coefficient ψ.

Table 8: Losses obtain for different parameter combination

(a) 250 modes

(b) 500 modes

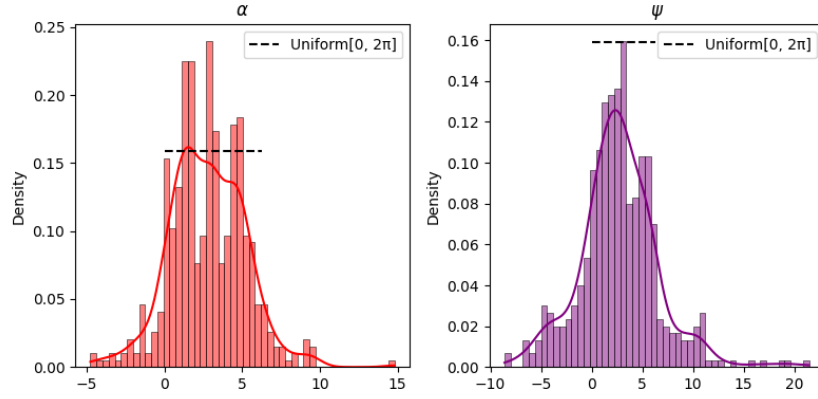| Parameters	Loss	RMS (x, y, z)	Parameters	Loss	RMS (x, y, z)
$(0, 0, \alpha, \psi)$	0.75522	(0.20, 0.11, 0.10)	$(0, 0, \alpha, \psi)$	0.74805	(0.11, 0.25, 0.07)
$(\varphi, 0, \alpha, \psi)$	0.74388	(0.09, 0.22, 0.04)	$(\varphi, 0, \alpha, \psi)$	0.74443	(0.21, 0.17, 0.06)
$(\varphi, 0, 0, \psi)$	0.73930	(0.11, 0.22, 0.08)	$(\varphi, 0, 0, \psi)$	0.74356	(0.09, 0.27, 0.06)
$(0, \theta, 0, \psi)$	24.56362	(0.12, 0.22, 0.06)	$(0, \theta, 0, \psi)$	0.74534	(0.20, 0.11, 0.10)
$(0, \theta, \alpha, \psi)$	0.75160	(0.21, 0.12, 0.10)	$(0, \theta, \alpha, \psi)$	0.74584	(0.20, 0.14, 0.10)
$(\varphi, \theta, 0, \psi)$	0.81957	(0.09, 0.21, 0.08)	$(\varphi, \theta, 0, \psi)$	0.73757	(0.20, 0.12, 0.11)
$(\varphi, \theta, \alpha, \psi)$	183.87051	(0.11, 0.27, 0.05)	$(\varphi, \theta, \alpha, \psi)$	3.38473	(0.23, 0.11, 0.09)

D.1 Marginal distributions

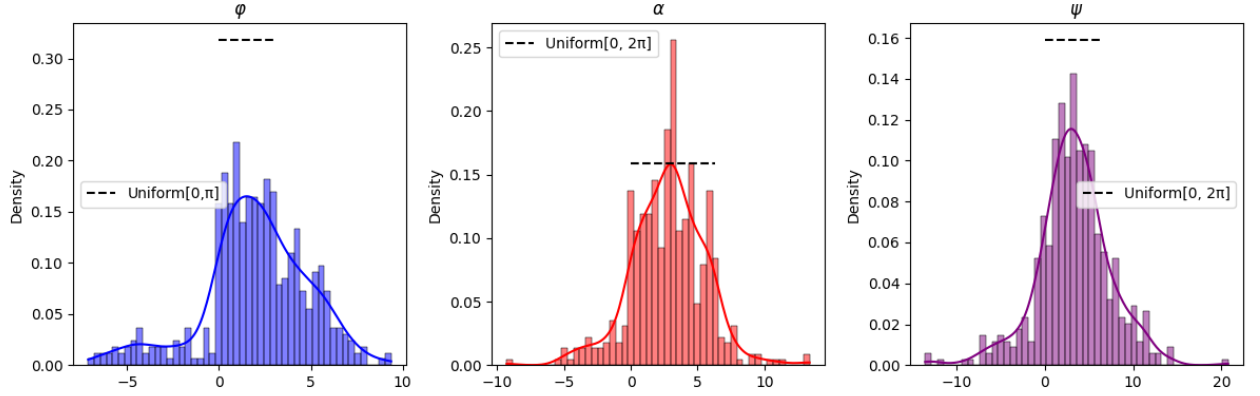Figure 27: Marginal density of the learned parameters (α, ψ)
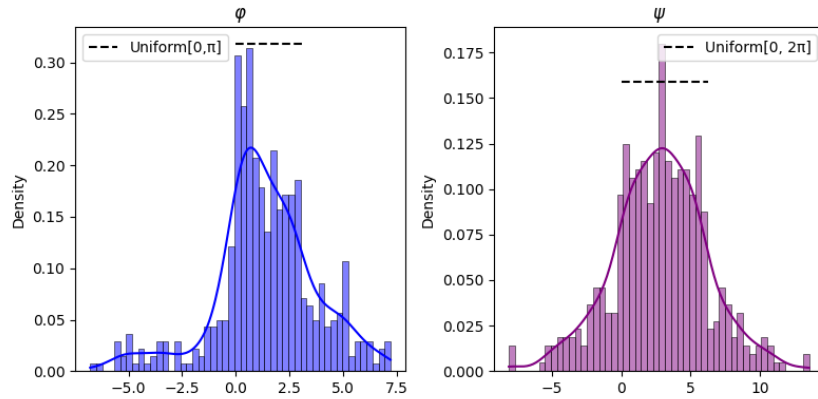

Figure 28: Marginal density of the learned parameters (φ, α, ψ)
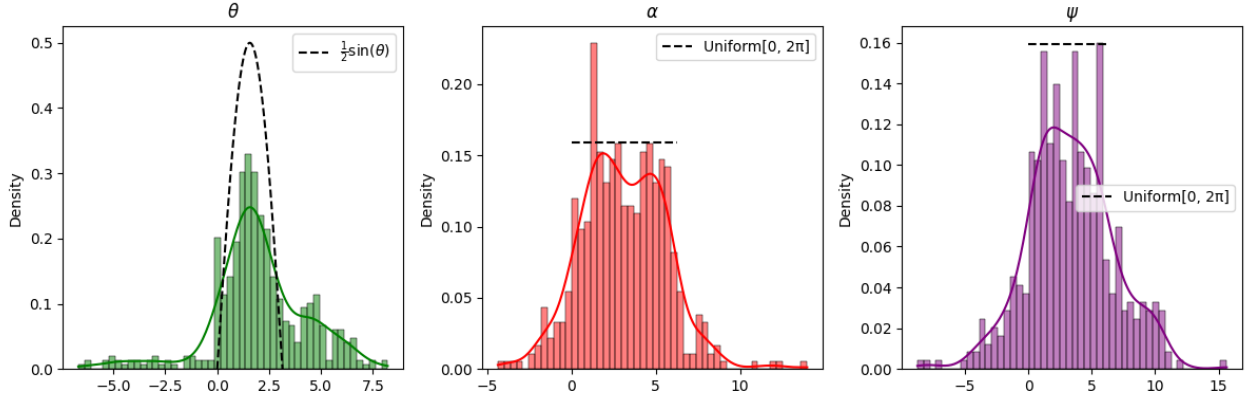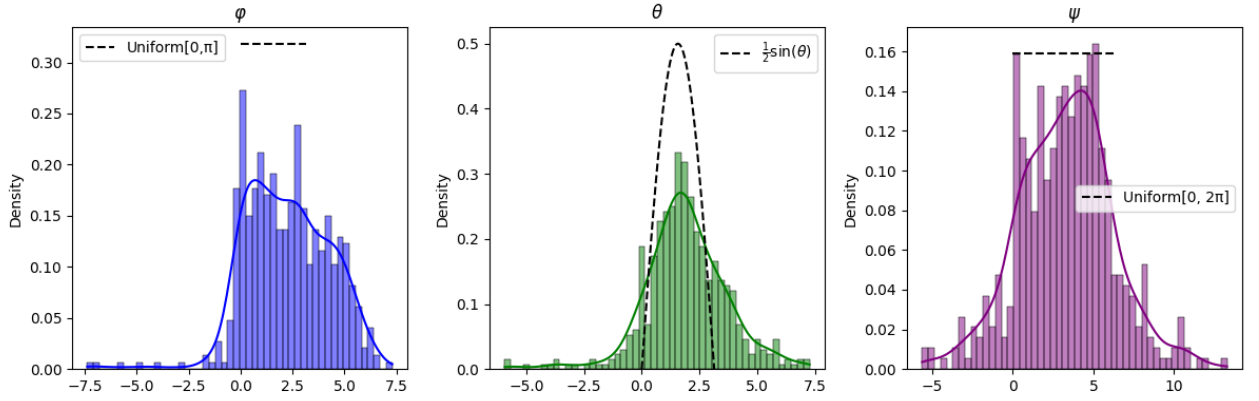

Figure 29: Marginal density of the learned parameters (φ, ψ)
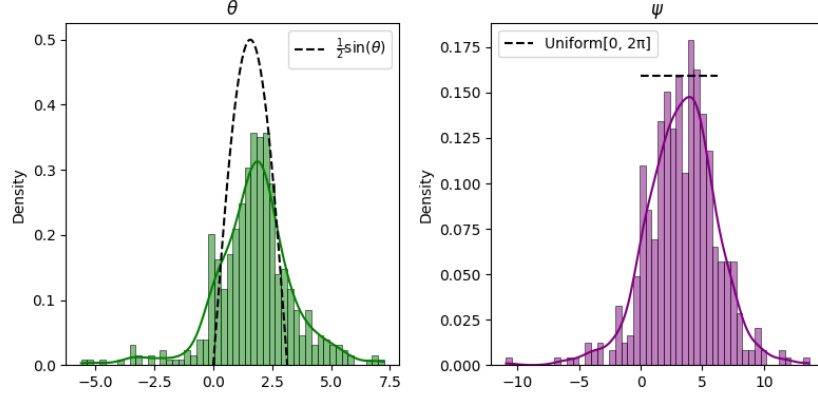
Figure 30: Marginal density of the learned parameters (θ, ψ)

Figure 31: Marginal density of the learned parameters (θ, α, ψ)

Figure 32: Marginal density of the learned parameters (φ, θ, ψ)
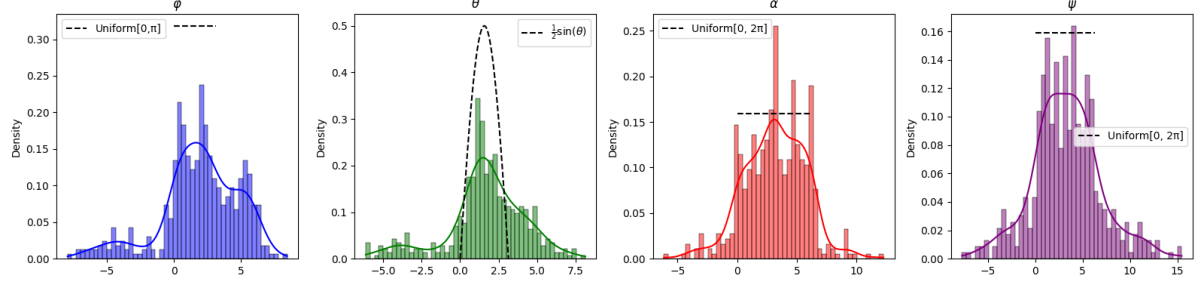
Figure 33: Marginal density of the learned parameters $(\varphi, \theta, \alpha, \psi)$

Overall, the distribution of the angles appears consistent with the previous case where ψ was not considered. The distribution of ψ itself remains pretty consistent across all cases.
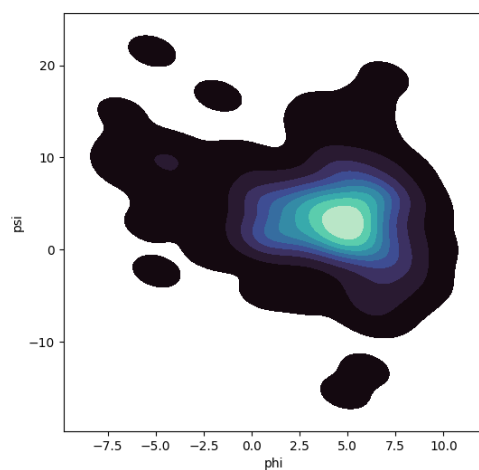
The distribution of α is the most similar to the base distribution $\mathcal{U}([0, 2\pi])$.

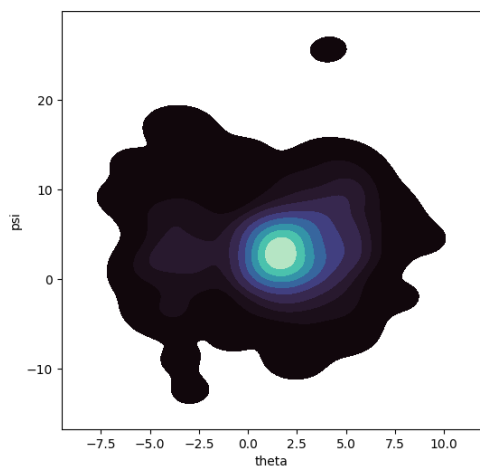However, these figures represent only the marginal densities and do not capture the dependencies between parameters.
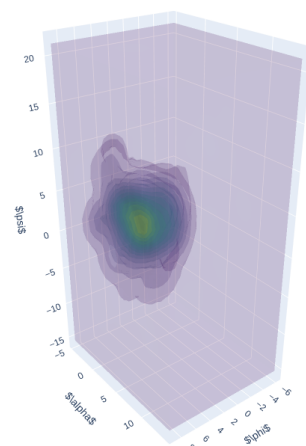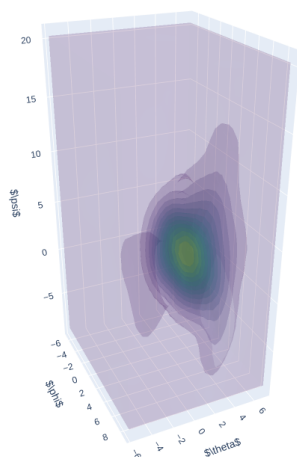
D.2 Joint distributions
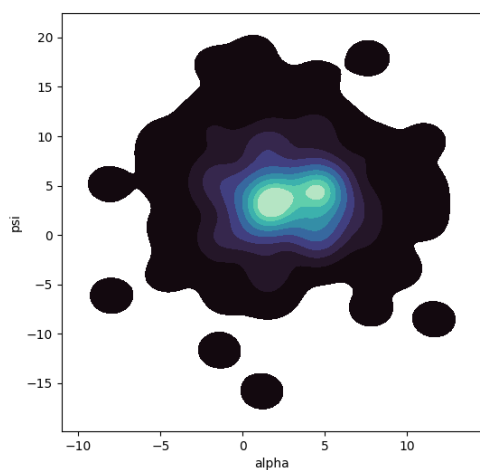
(a) Learned parameters (α, ψ)
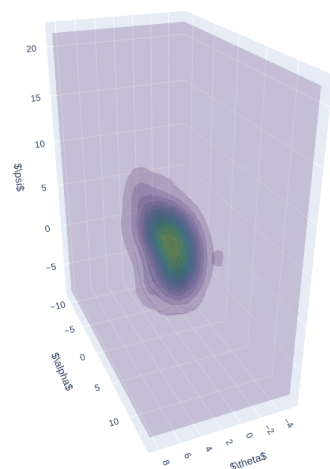

(b) Joint distribution of (φ, ψ)

(c) Joint distribution of (θ, ψ)

(d) Joint distribution of $(\varphi, \theta, \alpha)$

(e) Joint distribution of (θ, ψ)

(f) Joint distribution of $(\varphi, \theta, \alpha)$

Figure 34: Joint distributions of parameter combination

References

[1] Deniz Bezgin and Nikolaus Adams. "Normalizing Flows as a Novel PDF Turbulence Model". In: (Jan. 2021). DOI: 10.48550/arXiv.2101.03590.

[2] Nor Azwadi Che Sidik et al. "A Short Review on RANS Turbulence Models". In: *CFD Letters* 12 (Nov. 2020), pp. 83–96. DOI: 10.37934/cfdl.12.11.8396.

[3] Toshiyuki Gotoh, Daigen Fukayama, and Tohru Nakano. "Velocity field statistics in homogeneous steady turbulence obtained using a high-resolution direct numerical simulation". In: *Physics of Fluids* 14.3 (Mar. 2002), pp. 1065–1081. ISSN: 1070-6631. DOI: 10.1063/1.1448296. eprint: https://pubs.aip.org/aip/pof/article-pdf/14/3/1065/19111368/1065_1_online.pdf. URL: https://doi.org/10.1063/1.1448296.

[4] Jérémie Janin et al. "A new linear forcing method for isotropic turbulence with controlled integral length scale". In: *Physics of Fluids* 33 (Apr. 2021), p. 045127. DOI: 10.1063/5.0045818.

[5] J. Jiménez. "Small scale intermittency in turbulence". In: *European Journal of Mechanics - B/Fluids* 17.4 (1998), pp. 405–419. ISSN: 0997-7546. DOI: https://doi.org/10.1016/S0997-7546(98)80002-2. URL: https://www.sciencedirect.com/science/article/pii/S0997754698800022.

[6] Meet Hemant Parikh, Xiantao Fan, and Jian-Xun Wang. *Conditional flow matching for generative modeling of near-wall turbulence with quantified uncertainty.* 2025. arXiv: 2504.14485 [physics.flu-dyn]. URL: https://arxiv.org/abs/2504.14485.

[7] Florian Ries. "Numerical Modeling and Prediction of Irreversibilities in Sub- and Supercritical Turbulent Near-Wall Flows". PhD thesis. Nov. 2018.

[8] Haixin Wang et al. *FourierFlow: Frequency-aware Flow Matching for Generative Turbulence Modeling.* 2025. arXiv: 2506.00862 [cs.LG]. URL: https://arxiv.org/abs/2506.00862.

[9] Mustafa Z. Yousif, Linqi Yu, and HeeChang Lim. "Physics-guided deep learning for generating turbulent inflow conditions". In: *Journal of Fluid Mechanics* 936 (2022), A21. DOI: 10.1017/jfm.2022.61.