The JHU Turbulence Databases (*JHTDB*)

FORCED ISOTROPIC TURBULENCE DATA SET ON A $32{,}768^3$ GRID

Data provenance: P.K. Yeung[1], R. Uma-Vaideswaran[1], D.L. Dotson[1], K. Ravikumar[1]

Database ingest and Web services: H. Yao[2], M. Schnaubelt [2], G. Lemson[2], T. Zaki[2], A. Szalay[2], C. Meneveau[2], and IDIES staff

[1] Georgia Institute of Technology, Atlanta, GA 30332

[2]Johns Hopkins University, Baltimore, MD 21218

1 Overview

The data provided here was obtained from direct numerical simulations of forced isotropic turbulence on a $32{,}768^3$ periodic grid, using a highly-efficient pseudo-spectral parallel code. The simulations were performed on the world's first exascale computer, called Frontier, at Oak Ridge Leadership Computing Facility. The details of the GPU algorithm have been documented in detail in Yeung et al. (2025). The first scientific results were presented by the Georgia Tech group at the 2023 and 2024 APS Fluid Dynamics Meetings. Time integration uses second-order Runge-Kutta. The simulation is de-aliased using phase-shifting and truncation. Energy is injected by keeping the energy density in the lowest wavenumber modes prescribed following the approach of (Donzis and Yeung, 2010). A single frame of data, which includes the 3 components of the velocity vector and the pressure, are generated and written in files that can be accessed directly by the database services (zarr format on a ceph cluster). The dataset is approximately 1/2 Petabytes in size.

2 Pseudo-spectral simulation of forced isotropic turbulence

Direct numerical simulations were performed of unsteady and 3D fluctuating velocity fields governed by the incompressible Navier Stokes equations with numerical forcing:

$$\partial \mathbf{u}/\partial t + (\mathbf{u}\cdot\nabla)\mathbf{u} = -(1/\rho)\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{f}. \tag{1}$$

As in Rogallo (1981) the numerical solution is advanced in time in wavenumber space, with exact integrating factor for the viscous terms, and a combination of phase shifting and truncation for the control of aliasing errors. The forcing scheme consists of adjusting the velocity Fourier coefficients in the lowest three wavenumber shells, such that values of the energy spectrum function in these shells are consistent with long-time averages from simulations with stochastic forcing. The solution domain was divided in two directions, which allows use of more than N parallel processes for an N^3 grid. Much effort was made to optimize data movement in the form of communication calls required for distributed-memory parallelism. Memory parity between the CPU and GPU on Frontier allows practically all operations, including both computation and

communication, to be performed on the GPU, thus avoiding any overhead associated with data movement between CPU and GPU. The code is demonstrated to achieve high scalability despite some inevitable limitations associated with smaller message sizes when pushing the limits of the communication network in simulations at near full-system size.

3 Simulation strategy: successive grid refinement and viscosity reduction

The primary objective of the simulations was to study small-scale physics accurately at high Reynolds number at the maximum grid resolution that can fit into the GPU memory of the exascale machine available. To make the best use of available resources, Georgia Tech group has devised a strategy of successive grid refinement which helps reduce resources needed to reach a state of statistical stationarity at high Reynolds numbers. The approach is essentially to start from a prior simulation at modest Reynolds number, and to reduce the fluid viscosity through a series of stages of successive grid refinements, while keeping the forcing parameters that govern the large scales the same. The viscosity is adjusted at each stage in a manner that results in a targeted ratio of grid spacing to Kolmogorov scales throughout this process. Since the small scales evolve on short time scales, a new stationary state at higher Reynolds number is usually reached within no more than 40-50 Kolmogorov time scales instead of multiple large-eddy turnover times which would be very expensive. At each Reynolds number we reach stationary state at modest resolution in time and space, but then reduce both time step and grid spacing for higher accuracy at readily manageable cost. Several initial snapshots at modest Reynolds number are taken through this process, leading to multiple short but highly-resolved simulations in the manner proposed by Yeung and Ravikumar (2020).

4 Description of field

The main statistical features of the final field stored in the database, computed over the entire stored snapshot (using Fourier-space representation), are listed in Table 2.

A radial kinetic energy spectrum and compensated spectrum for the snapshot, computed from the snapshot (in Fourier space) is shown in Fig. 1 and 2. The radial spectrum displays a small intermittency correction to the -5/3 scaling and shows indications of the bottleneck effect.

5 Data storage

We develop a method for storing data using chunked Zarr files (a scalable, compressible, and versatile array storage format (https://zarr.dev/)). This pivot to zarr files resulted in several benefits. First, it has simplified the processing code, making it more maintainable. Second, processing times became significantly faster with zarr files due to fewer distinct I/O operations when reading data. Third, adopting Zarr files facilitates a smoother and faster transition to Ceph (an object storage system) enhancing scalability and efficiency.

The 32768^3 dataset is stored using the Zarr format on a ceph cluster. The dataset (one snapshot) is organized within a single Zarr store, containing separate ‘velocity’ and ‘pressure’ directories for storing the

Table 1: Simulation parameters and resulting statistics for snapshot 1:

Quantity	Snapshot 1
Domain	$2\pi \times 2\pi \times 2\pi$
Grid	$32{,}768 \times 32{,}768 \times 32{,}768$
Viscosity ν	1.10555×10^{-5}
RMS velocities u', v', w'	1.3969, 1.6692, 1.6405
Kinetic energy	3.7143
Dissipation rate ε	1.2739
Resolution, $k_{\max}\eta$	2.7875
Kolmogorov length $(\nu^3/\varepsilon)^{1/4}$	1.8047×10^{-4}
Kolmogorov velocity $(\nu\varepsilon)^{1/4}$	6.1261×10^{-2}
Kolmogorov time-scale $(\nu/\varepsilon)^{1/2}$	2.9459×10^{-3}
u integral length scales L_{ux}, L_{uy}, L_{uz}	0.95576, 0.47324, 0.51600
v integral length scales L_{vx}, L_{vy}, L_{vz}	0.49598, 1.5480, 0.47430
w integral length scales L_{wx}, L_{wy}, L_{wz}	0.60514, 1.0460, 1.6205
Large eddy turnover times L_{ux}/u', L_{vy}/v', L_{wz}/w'	0.68422, 0.92743, 0.98778
Integral scale Reynolds # $Re = L_{ux}u'/\nu$, $L_{vy}v'/\nu$, $L_{wz}w'/\nu$	1.2076×10^5, 2.3372×10^5, 2.4046×10^5
Taylor microscales $\lambda_{u,v,w} = (u', v', w') \times (15\nu/\varepsilon)^{1/2}$	0.015936, 0.019046, 0.018719
Taylor scale Reynolds number	2,556

corresponding velocity and pressure fields. The data is further partitioned into chunks of size 64^3, which serve as the fundamental storage units and are arranged in Cartesian order (k, j, i). Consequently, the velocity and pressure directories each contain 3×512 subdirectories, with each subdirectory storing a corresponding 64^3 chunk in (k, j, i) orders.

6 Sample statistics from physical space data accessible via getdata tools:

As reference for users, we provide below some statistical features that can be computed efficiently using getdata functions from data in physical space. The statistics are computed over 64 equally distributed lines along x at every Δy_l and Δz_l intervals along the y and z axes (and similarly along the other two axes). As an example, for lines parallel to the x-direction, the coordinates of all the points for the data are generated as follows:

The coordinates for sampling lines are generated as shown in the snapshot in Fig. 3, where *points* array stores 3×64 lines that are perpendicular to $yz-, xz-, xy-$ planes. The totality of lines is shown in Fig. 4.

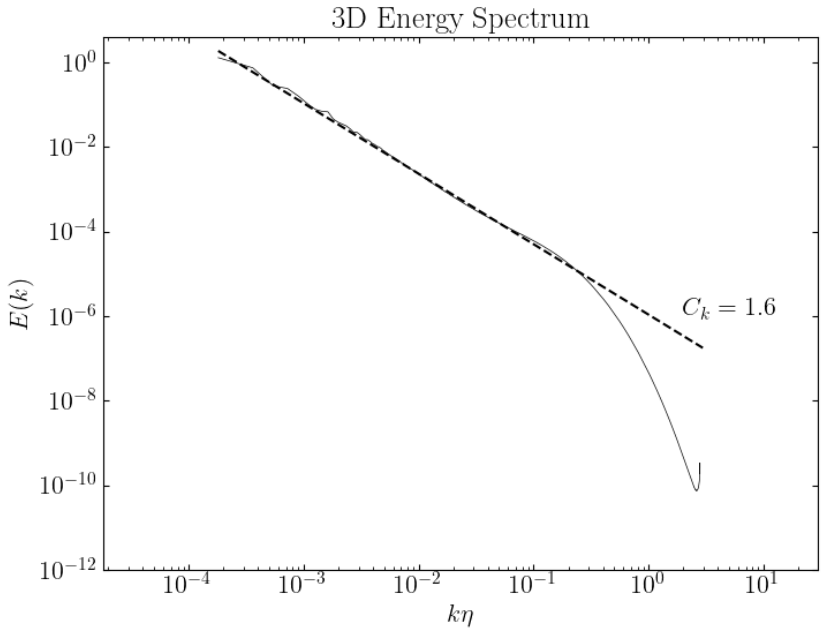

Figure 1: Radial 3D energy spectrum computed for the stored snapshot. The dashed line shows the expected Kolmogorov inertial range line $E(k) = 1.6 \times \varepsilon^{2/3} k^{-5/3}$.

The one-dimensional longitudinal energy spectrum $E_{11}(k)$ (computed as the average of the longitudinal spectra in each direction) is shown in Fig. 5 (solid blue line). The result is compared with the Kolmogorov spectrum $E_{11} = (18/55) \times C_K \times \varepsilon^{2/3} k_1^{-5/3}$ (dashed line) using $C_K = 1.6$. In Fig. 5 (solid orange line) we also show the transverse spectrum (averaged in the 2 transverse directions) and compare it to $E_{22} = (4/3) \times (18/55) \times C_K \times \varepsilon^{2/3} k_1^{-5/3}$. In Fig. 6, we show the compensated 1D longitudinal and transverse energy spectrum compared to coefficients $(18/55) \times 1.6$ and $(4/3) \times (18/55) \times 1.6$. The energy spectra and the compensated spectra in Kolmogorov units are shown in Fig. 7 and 8. As expected, 1D spectra also display a small intermittency correction and indications of the bottleneck effect.

The statistics of the various components of the velocity gradient (computed using the getdata function with the gradient spatial operator and the spline spatial operator, m2q8, option) are shown in Table 2. These parameters are averaged over the $3 \times 64 \times 32{,}768$ points on the lines with equal distance and parallel to the x, y, z directions used to compute the spectra shown. As such, they cannot be considered to be equivalent to those computed over all points, or fully statistically converged, but a good reference for potential users.

Finally, using these gradients, a check on the divergence-free condition using these gradient operators is performed by evaluating the quantity $\langle(\partial u/\partial x + \partial v/ + \partial w/\partial z)^2\rangle$ and comparing it to $\langle(\partial u/\partial x)^2\rangle + \langle(\partial v/\partial y)^2\rangle + \langle(\partial w/\partial z)^2\rangle$.

Note that the divergence-free condition in the simulation is enforced based on the spectral representation of the derivatives on horizontal planes while the gradients used in this analysis are based on finite differencing of various orders or splines as required for more efficient spatially localized data usage. Therefore, when evaluating the divergence using these spatially more localized derivative operators, a non-zero divergence

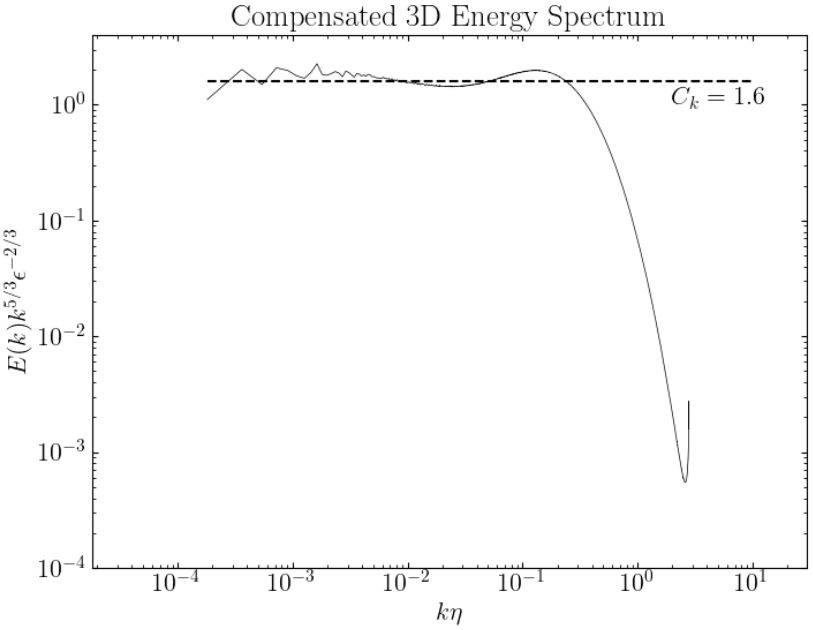

Figure 2: Compensated radial 3D energy spectrum computed for the stored snapshot. The dashed line shows the expected Kolmogorov coefficient $C_k = 1.6$.

should be expected. However, since the data are well resolved ($k_{\max}\eta 2.78$) the error is expected to be small. Indeed, averaging over the $3 \times 64 \times 32{,}768$ points, we obtain $\langle(\partial u/\partial x + \partial v/\partial y + \partial w/\partial z)^2\rangle = 1.84$. Compared to (e.g.) $\langle(\partial u/\partial x)^2\rangle + \langle(\partial v/\partial y)^2\rangle + \langle(\partial w/\partial z)^2\rangle = 23824.88$ or any of the individual gradient variances shown in Table 2 it is quite small and consistent with an error in individual gradient magnitudes of less than 1%.

Acknowledgements: We acknowledge funding by the National Science Foundation (Framework project NSF award number CSSI-2103874, with a JHU to Georgia Tech. subaward). We also gratefully acknowlegde support from the DOE Office of Science INCITE allocation program in the years 2023-2025. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

References

Donzis, D. A., and P. K. Yeung, 2010: Resolution effects and scaling in numerical simulations of passive scalar mixing in turbulence. *Physica D: Nonlinear Phenomena*, **239**, 1278–1287.

Rogallo, R. S., 1981: Numerical experiments in homogeneous turbulence. NASA TM 81315, NASA Ames Research Center, Moffett Field, CA.

Yeung, P. K., and K. Ravikumar, 2020: Advancing understanding of turbulence through extreme-scale computation: Intermittency and simulations at large problem sizes. *Physical Review Fluids*, **5**, 110 517.

```python
# Parameters
n_points = 32768
dx = 2 * np.pi / n_points
N_box = 8

# Coordinate arrays
x_points = np.linspace(0.0, 2 * np.pi, n_points, dtype=np.float64)
y_coords = np.linspace(512 * dx, 2 * np.pi - 512 * dx, N_box, dtype=np.float64)
z_coords = np.linspace(512 * dx, 2 * np.pi - 512 * dx, N_box, dtype=np.float64)

# Total number of lines = 3 * N_box * N_box
points = np.zeros((3 * N_box * N_box, n_points, 3), dtype=np.float64)

line_index = 0
for yy in y_coords:
    for zz in z_coords:
        # X-direction line
        points[line_index, :, 0] = x_points
        points[line_index, :, 1] = yy
        points[line_index, :, 2] = zz
        line_index += 1

        # Y-direction line
        points[line_index, :, 0] = yy
        points[line_index, :, 1] = x_points
        points[line_index, :, 2] = zz
        line_index += 1

        # Z-direction line
        points[line_index, :, 0] = yy
        points[line_index, :, 1] = zz
        points[line_index, :, 2] = x_points
        line_index += 1

print(points.shape)
```

```
(192, 32768, 3)
```

Figure 3: Python script to generate 64×3 lines in 3D physical space.

Yeung, P. K., K. Ravikumar, S. Nichols, and R. Uma-Vaideswaran, 2025: GPU-enabled extreme-scale turbulence simulations: Fourier pseudo-spectral algorithms at the exascale using OpenMP offloading. *Computer Physics Communications*, **306**, 109 364.

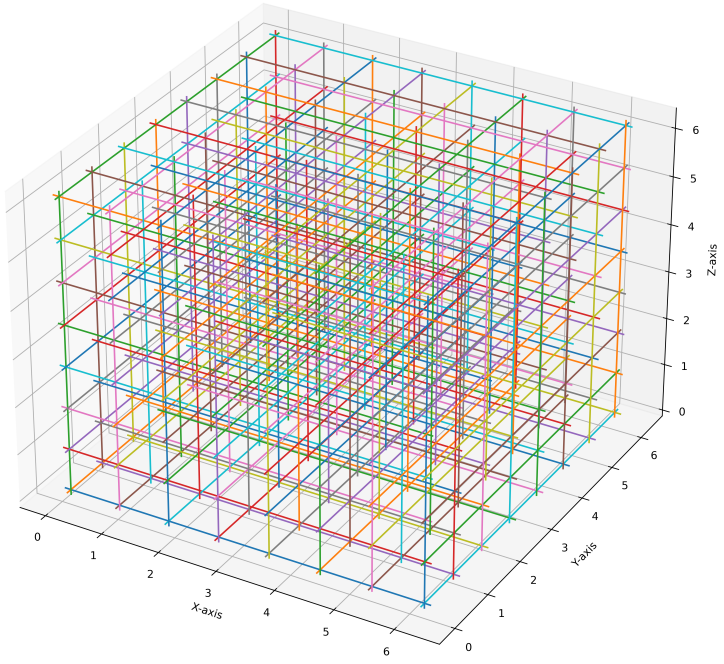

Figure 4: Lines (in 3D physical space) used to generate 1D energy spectra and evaluate velocity gradient statistics from a subset of the full data.

Table 2: Velocity gradient statistics from 3×64 lines ($192\times$ 32,768 = 6,291,456 sample points, generated using the code shown in Fig. 3). Since this sample represents only a tiny (0.1788×10^{-6}) fraction of all points in the domain, these values cannot be considered to be statistically converged moments.

Gradient component	Variance	Skewness factor	Flatness factor
$\partial u/\partial x$	7947.94	-0.7463	15.5863
$\partial u/\partial y$	15890.80	-0.0611	24.8250
$\partial u/\partial z$	15815.43	0.0029	22.4515
$\partial v/\partial x$	16026.68	-0.1300	29.1940
$\partial v/\partial y$	7940.95	-0.7540	16.2161
$\partial v/\partial z$	15826.85	0.0863	25.2166
$\partial w/\partial x$	15934.09	0.0898	24.1140
$\partial w/\partial y$	15826.62	0.1914	27.8905
$\partial w/\partial z$	7635.96	-0.7453	15.7545

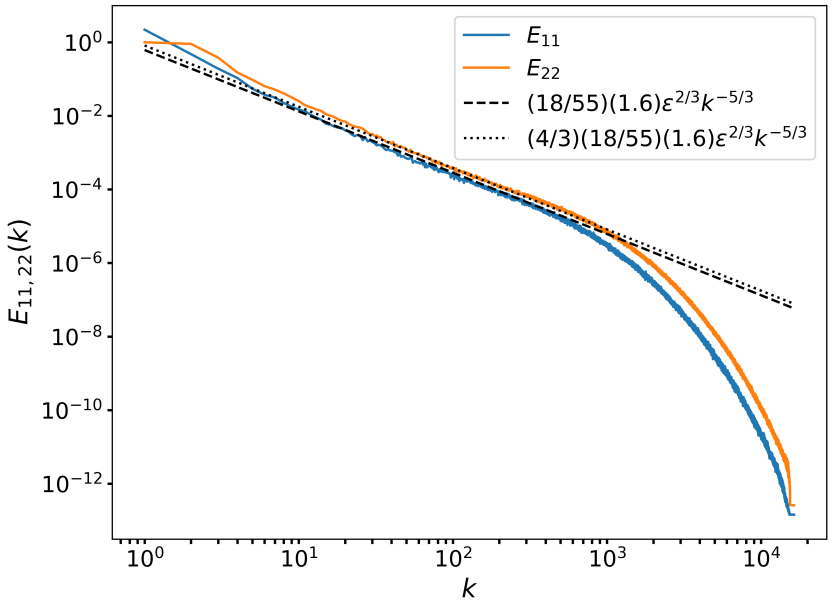

Figure 5: Longitudinal and transverse one-dimensional energy spectra computed from the 3×64 regularly spaced lines in the 3 Cartesian directions (Fig. 4). The dashed and dotted lines show the expected Kolmogorov inertial range longitudinal and transverse spectra ($E_{11} = (18/55) \times C_K \times \varepsilon^{2/3} k^{-5/3}$ and $E_{22} = (4/3)(18/55) \times C_K \times \varepsilon^{2/3} k^{-5/3}$, respectively, with $C_K = 1.6$).

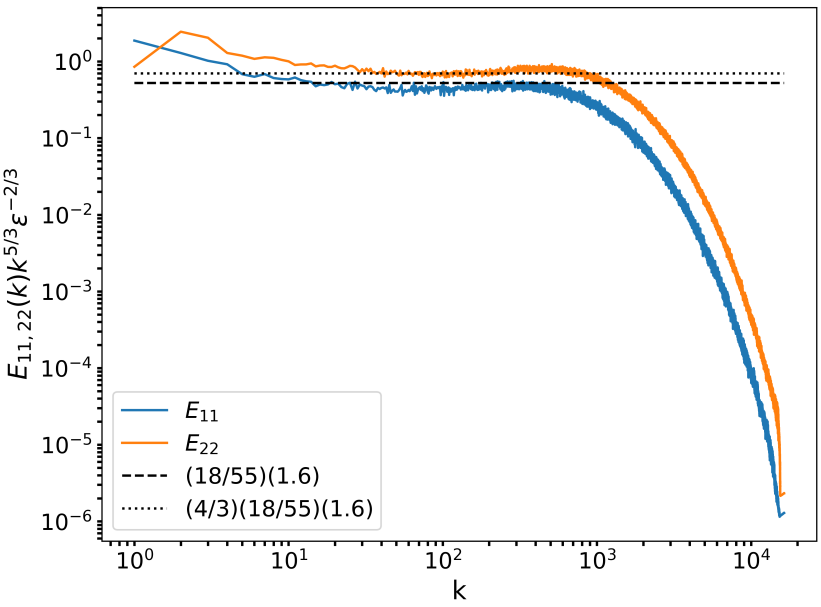

Figure 6: Same as Fig. 5 for compensated spectra.

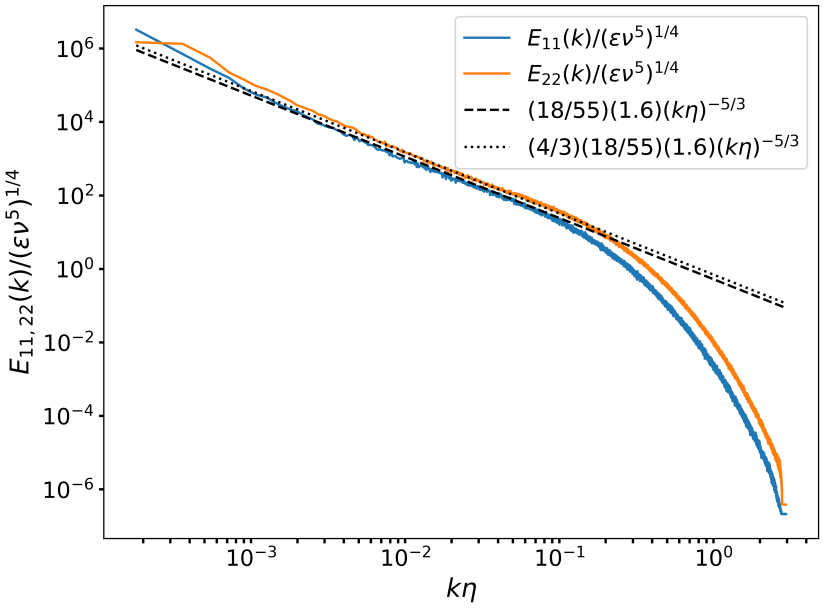

Figure 7: Same as Fig. 5 in Kolmogorov units.

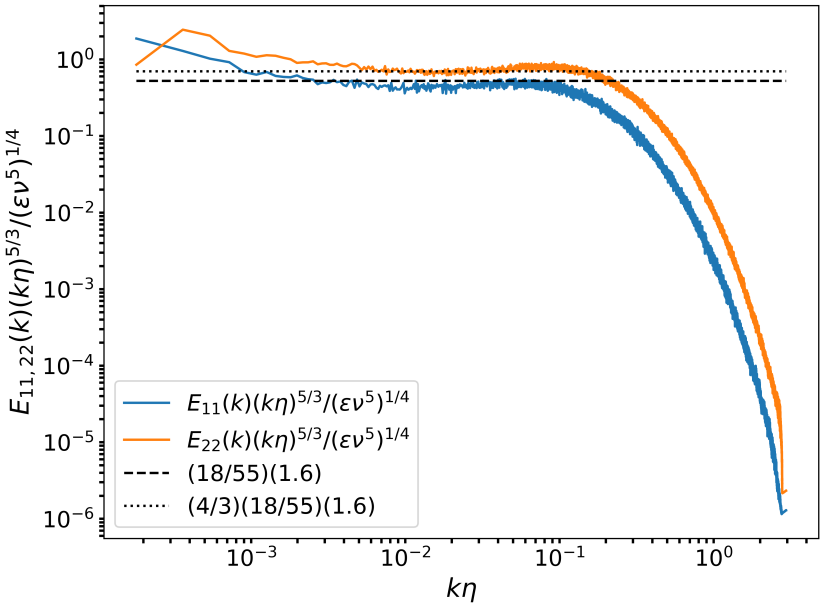

Figure 8: Same as Fig. 6 in Kolmogorov units.