Using Partial Fourier Transforms to Study Kolmogorov's Inertial-Range Flux

John C. Bowman and Zayd Ghoggali
Department of Mathematical and Statistical Sciences
University of Alberta

June 28, 2016

www.math.ualberta.ca/∼bowman/talks

Turbulence

Big whirls have little whirls that feed on their velocity, and little whirls have littler whirls and so on to viscosity... [Richardson 1922]

- In 1941, Kolmogorov conjectured that the energy spectrum of 3D incompressible turbulence exhibits a self-similar power-law scaling characterized by a uniform *cascade* of energy to molecular (viscous) scales:

$$E(k) = C\epsilon^{2/3}k^{-5/3}.$$

Turbulence

Big whirls have little whirls that feed on their velocity, and little whirls have littler whirls and so on to viscosity... [Richardson 1922]

- In 1941, Kolmogorov conjectured that the energy spectrum of 3D incompressible turbulence exhibits a self-similar power-law scaling characterized by a uniform *cascade* of energy to molecular (viscous) scales:

$$E(k) = C\epsilon^{2/3}k^{-5/3}.$$

- Here k is the Fourier wavenumber and $E(k)$ is normalized so that $\int E(k)\,dk$ is the total energy.

Turbulence

Big whirls have little whirls that feed on their velocity, and little whirls have littler whirls and so on to viscosity... [Richardson 1922]

- In 1941, Kolmogorov conjectured that the energy spectrum of 3D incompressible turbulence exhibits a self-similar power-law scaling characterized by a uniform *cascade* of energy to molecular (viscous) scales:

$$E(k) = C\epsilon^{2/3}k^{-5/3}.$$

- Here k is the Fourier wavenumber and $E(k)$ is normalized so that $\int E(k)\,dk$ is the total energy.
- Kolmogorov suggested that C might be a universal constant.

Turbulence

> *Big whirls have little whirls that feed on their velocity, and little whirls have littler whirls and so on to viscosity...* [Richardson 1922]

- In 1941, Kolmogorov conjectured that the energy spectrum of 3D incompressible turbulence exhibits a self-similar power-law scaling characterized by a uniform *cascade* of energy to molecular (viscous) scales:

$$E(k) = C\epsilon^{2/3}k^{-5/3}.$$

- Here k is the Fourier wavenumber and $E(k)$ is normalized so that $\int E(k)\,dk$ is the total energy.
- Kolmogorov suggested that C might be a universal constant.
- He hypothesized that the local energy flux in the inertial range is independent of wavenumber, presumably due to an underlying self-similarity.

2D Turbulence in Fourier Space

- Navier–Stokes equation for vorticity $\omega \doteq \hat{\boldsymbol{z}} \cdot \boldsymbol{\nabla} \times \boldsymbol{u}$ of an incompressible ($\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$) fluid:

$$\frac{\partial \omega}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \omega = \nu \nabla^2 \omega + f.$$

2D Turbulence in Fourier Space

- Navier–Stokes equation for vorticity $\omega \doteq \hat{\boldsymbol{z}}\boldsymbol{\cdot}\boldsymbol{\nabla}\times\boldsymbol{u}$ of an incompressible ($\boldsymbol{\nabla}\boldsymbol{\cdot}\boldsymbol{u} = 0$) fluid:

$$\frac{\partial \omega}{\partial t} + \boldsymbol{u}\boldsymbol{\cdot}\boldsymbol{\nabla}\omega = \nu\nabla^2\omega + f.$$

- In Fourier space:

$$\frac{\partial \omega_{\boldsymbol{k}}}{\partial t} + \nu_{\boldsymbol{k}}\omega_{\boldsymbol{k}} = \int d\boldsymbol{p} \int d\boldsymbol{q}\, \frac{\epsilon_{\boldsymbol{kpq}}}{q^2}\omega^*_{\boldsymbol{p}}\omega^*_{\boldsymbol{q}} + f_{\boldsymbol{k}},$$

where $\nu_{\boldsymbol{k}} \doteq \nu k^2$ and $\epsilon_{\boldsymbol{kpq}} \doteq (\hat{\boldsymbol{z}}\boldsymbol{\cdot}\boldsymbol{p}\times\boldsymbol{q})\,\delta(\boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q})$ is antisymmetric under permutation of any two indices.

$$\frac{\partial \omega_{\boldsymbol{k}}}{\partial t} + \nu_{\boldsymbol{k}} \omega_{\boldsymbol{k}} = \int d\boldsymbol{p} \int d\boldsymbol{q} \, \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \omega_{\boldsymbol{p}}^* \omega_{\boldsymbol{q}}^* + f_{\boldsymbol{k}},$$

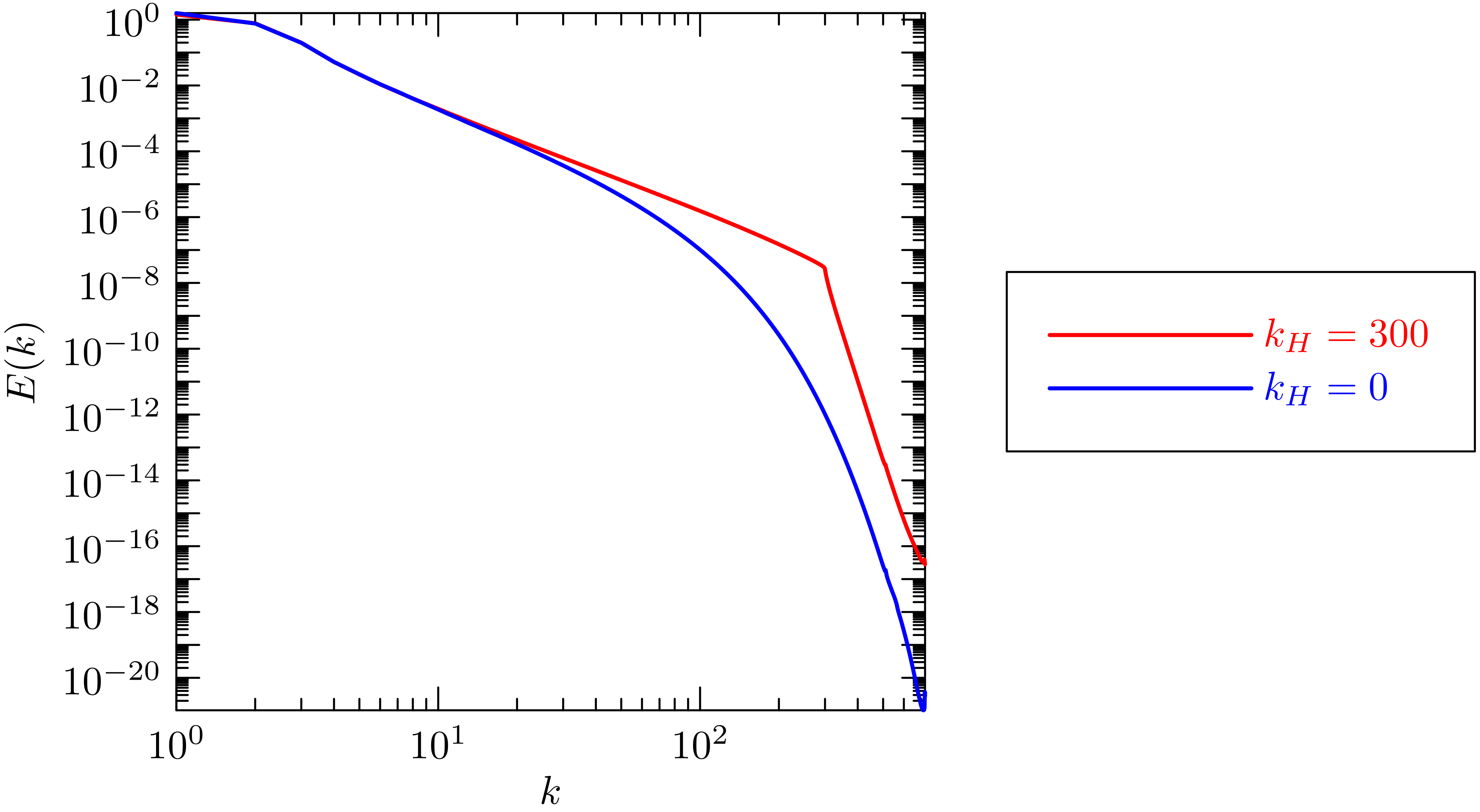
- When $\nu = f_{\boldsymbol{k}} = 0$,

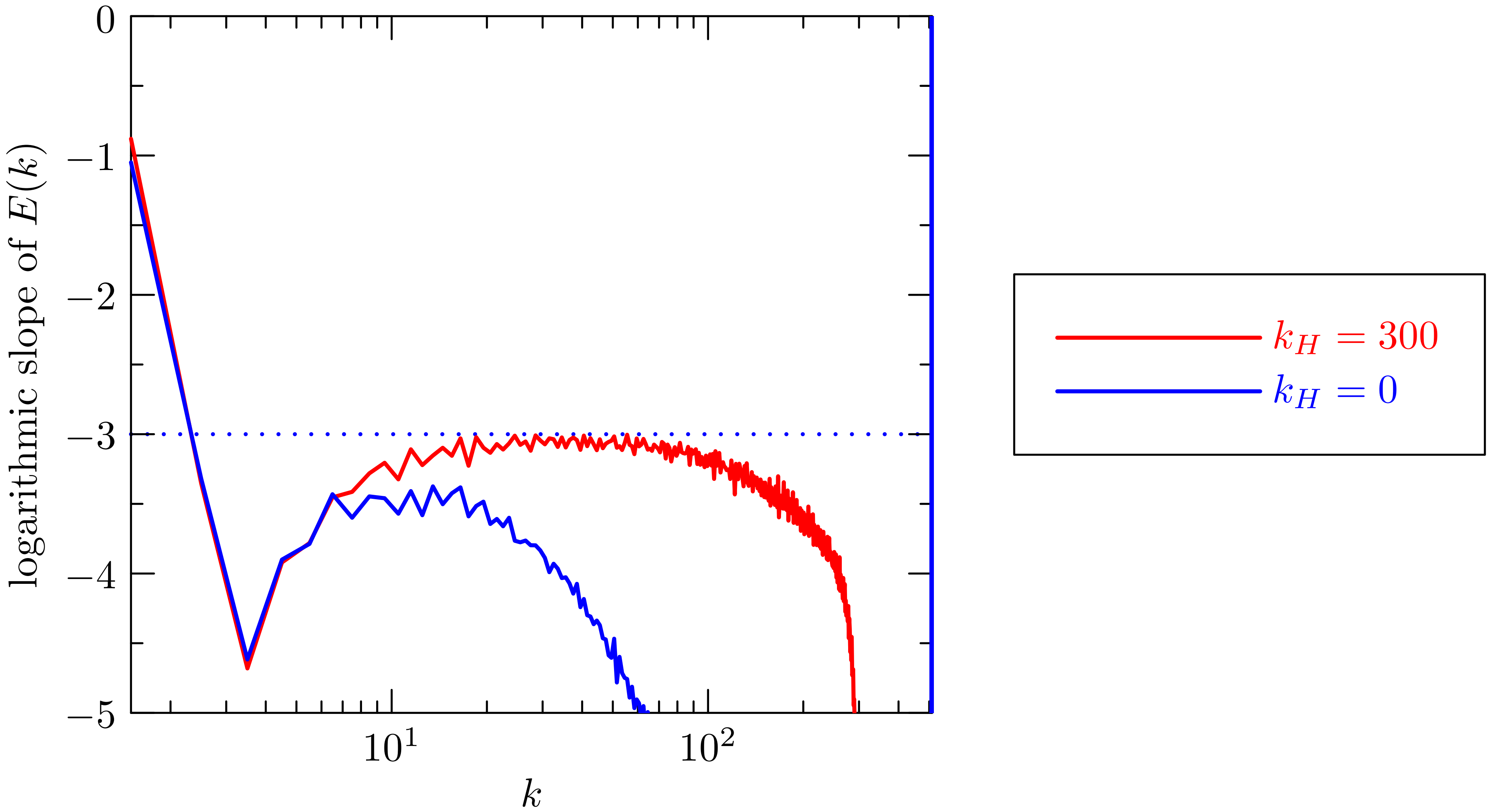
  enstrophy $Z = \frac{1}{2} \int |\omega_{\boldsymbol{k}}|^2 \, d\boldsymbol{k}$ and energy $E = \frac{1}{2} \int \frac{|\omega_{\boldsymbol{k}}|^2}{k^2} d\boldsymbol{k}$ are conserved:

$$\frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \qquad \text{antisymmetric in} \qquad \boldsymbol{k} \leftrightarrow \boldsymbol{p},$$

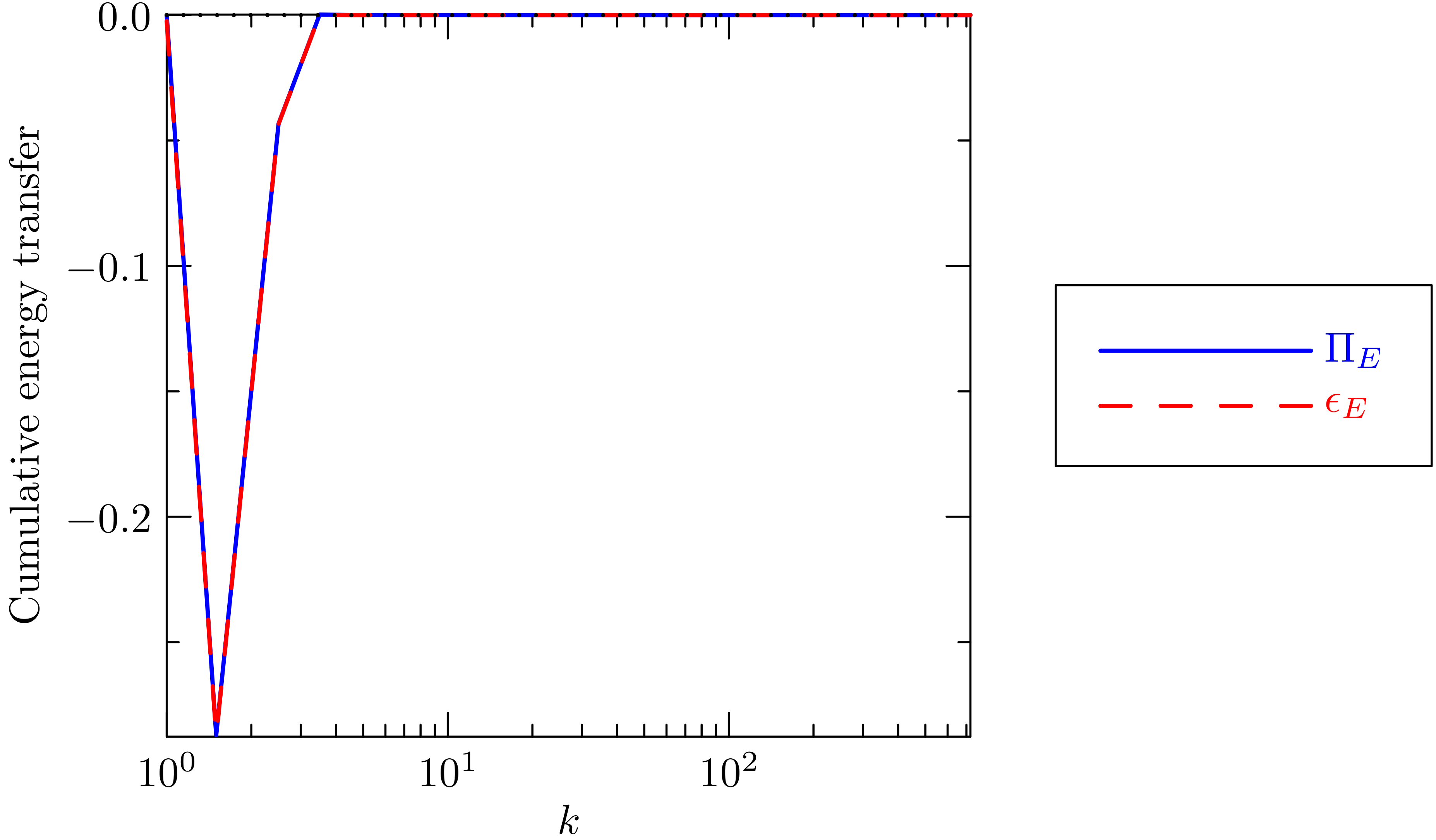
$$\frac{1}{k^2} \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \qquad \text{antisymmetric in} \qquad \boldsymbol{k} \leftrightarrow \boldsymbol{q}.$$

Forcing at $k = 2$, friction for $k < 3$, viscosity for $k \geq k_H = 300$ (1023 × 1023 dealiased modes)

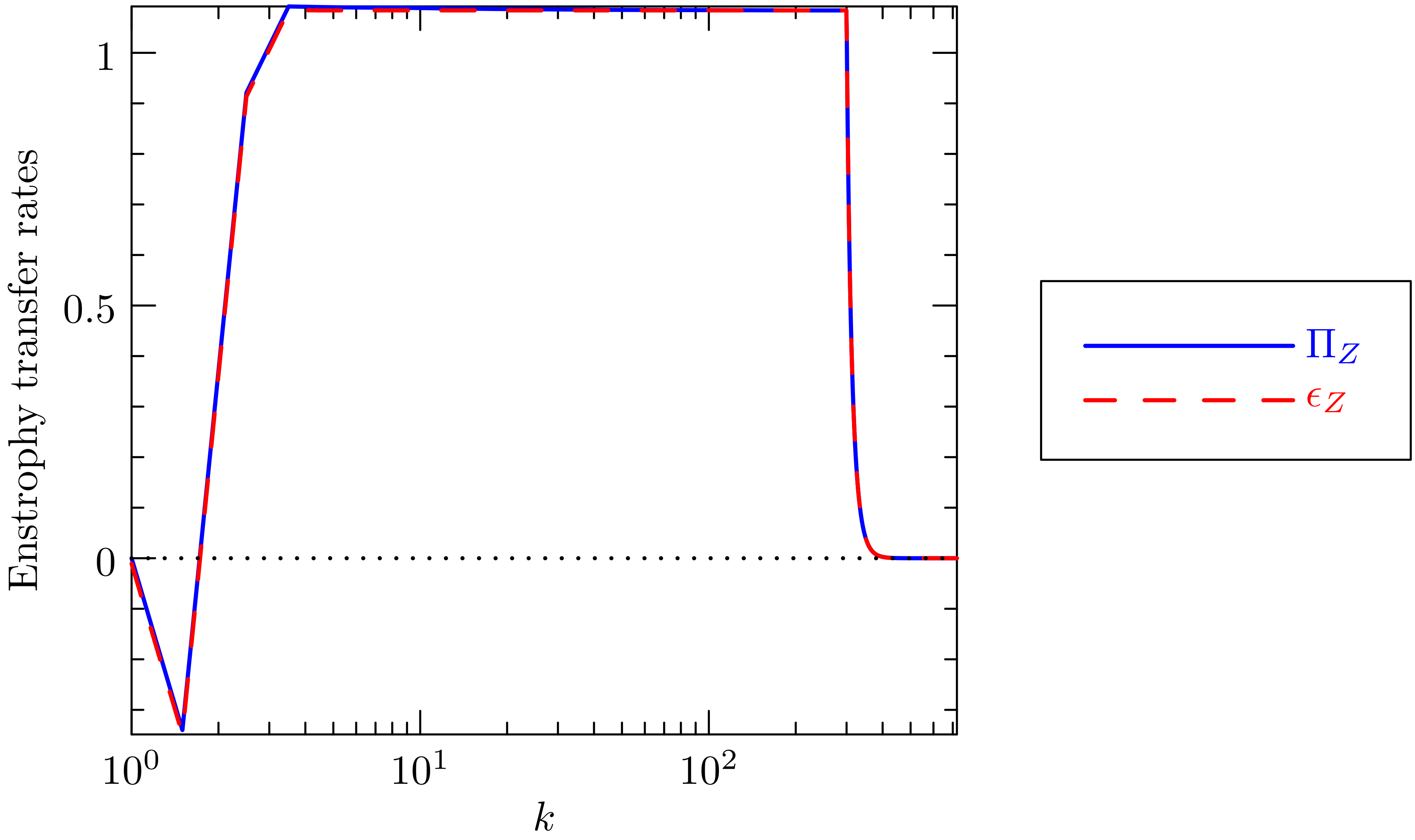




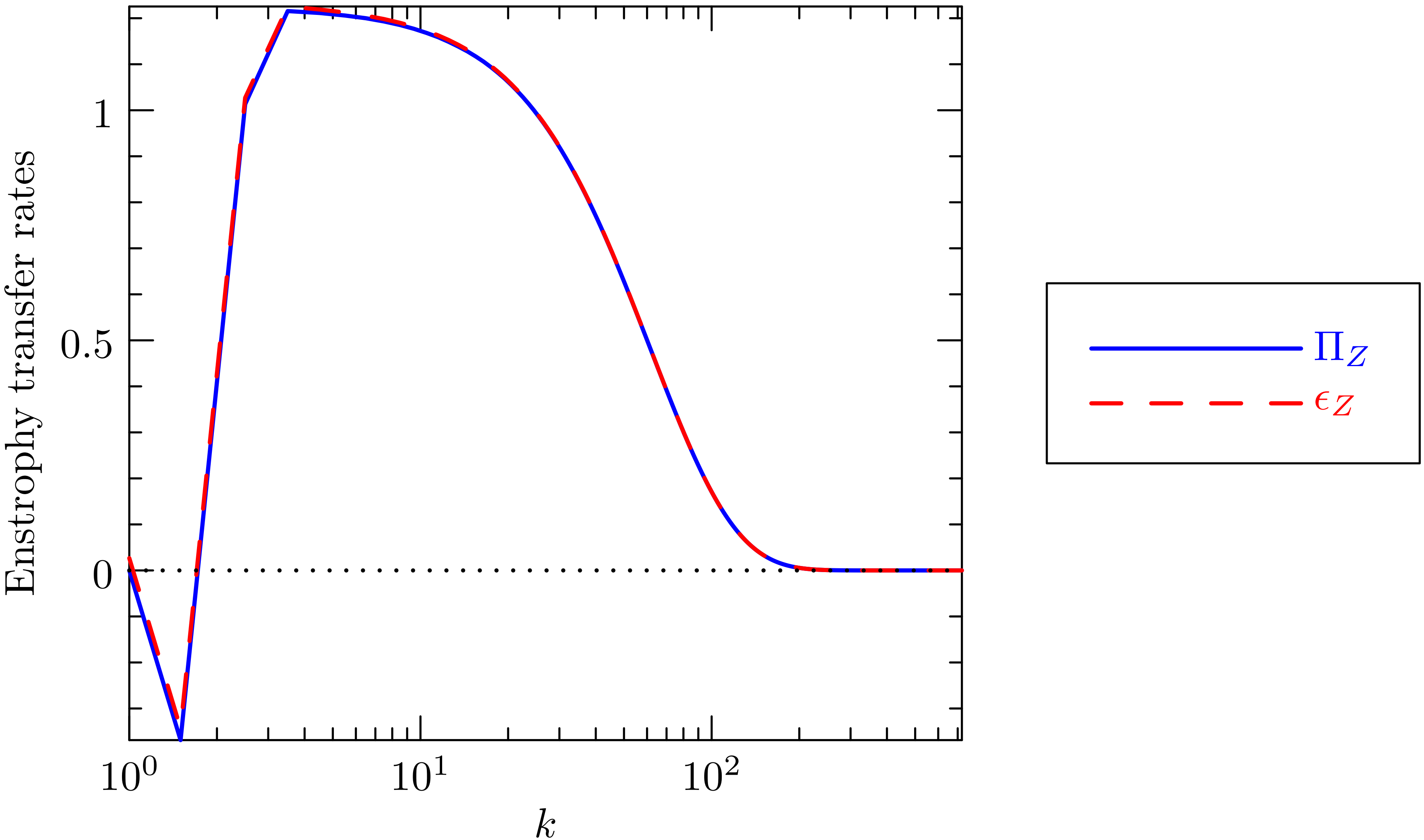
0
−1
−2
−3
−4
−5
logarithmic slope of $E(k)$
10^1
10^2
k
$k_H = 300$
$k_H = 0$



Cutoff viscosity ($k \geq k_H = 300$)



Cutoff viscosity $(k \geq k_H = 300)$



Molecular viscosity ($k \geq k_H = 0$)

Transfer vs. Flux

- Distinguish between transfer and flux.

Transfer vs. Flux

- Distinguish between transfer and flux.
- The mean rate of enstrophy transfer to $[k, \infty)$ is given by

$$\Pi(k) = \int_k^\infty T(k)\,dk = -\int_0^k T(k)\,dk.$$

Transfer vs. Flux

- Distinguish between transfer and flux.
- The mean rate of enstrophy transfer to $[k, \infty)$ is given by

$$\Pi(k) = \int_k^\infty T(k)\, dk = -\int_0^k T(k)\, dk.$$

- In a steady state, $\Pi(k)$ will trivially be constant within a true inertial range.

Transfer vs. Flux

- Distinguish between transfer and flux.

- The mean rate of enstrophy transfer to $[k, \infty)$ is given by

$$\Pi(k) = \int_k^\infty T(k)\, dk = -\int_0^k T(k)\, dk.$$

- In a steady state, $\Pi(k)$ will trivially be constant within a true inertial range.

- The statement of local wavenumber-independent inertial-range energy flux is fundamentally different than the trivial observation that the nonlocal energy transfer is independent of wavenumber in the inertial range.

Transfer vs. Flux

- Distinguish between transfer and flux.
- The mean rate of enstrophy transfer to $[k, \infty)$ is given by

$$\Pi(k) = \int_k^\infty T(k)\,dk = -\int_0^k T(k)\,dk.$$

- In a steady state, $\Pi(k)$ will trivially be constant within a true inertial range.
- The statement of local wavenumber-independent inertial-range energy flux is fundamentally different than the trivial observation that the nonlocal energy transfer is independent of wavenumber in the inertial range.
- In contrast, the enstrophy flux through a wavenumber k is the amount of enstrophy transferred to small scales *via* triad interactions involving mode k.

Uniform flux

- Although the independence of the local inertial-range energy flux with wavenumber is one of the key hypothesis underlying Kolmogorov's famous 5/3 power-law form for the kinetic energy spectrum, it has never been directly tested, either experimentally or numerically.

Uniform flux

- Although the independence of the local inertial-range energy flux with wavenumber is one of the key hypothesis underlying Kolmogorov's famous 5/3 power-law form for the kinetic energy spectrum, it has never been directly tested, either experimentally or numerically.

- To validate Kolmogorov's uniform flux hypothesis in a high-resolution pseudospectral code, detailed wavenumber constraints must be imposed on the convolution.

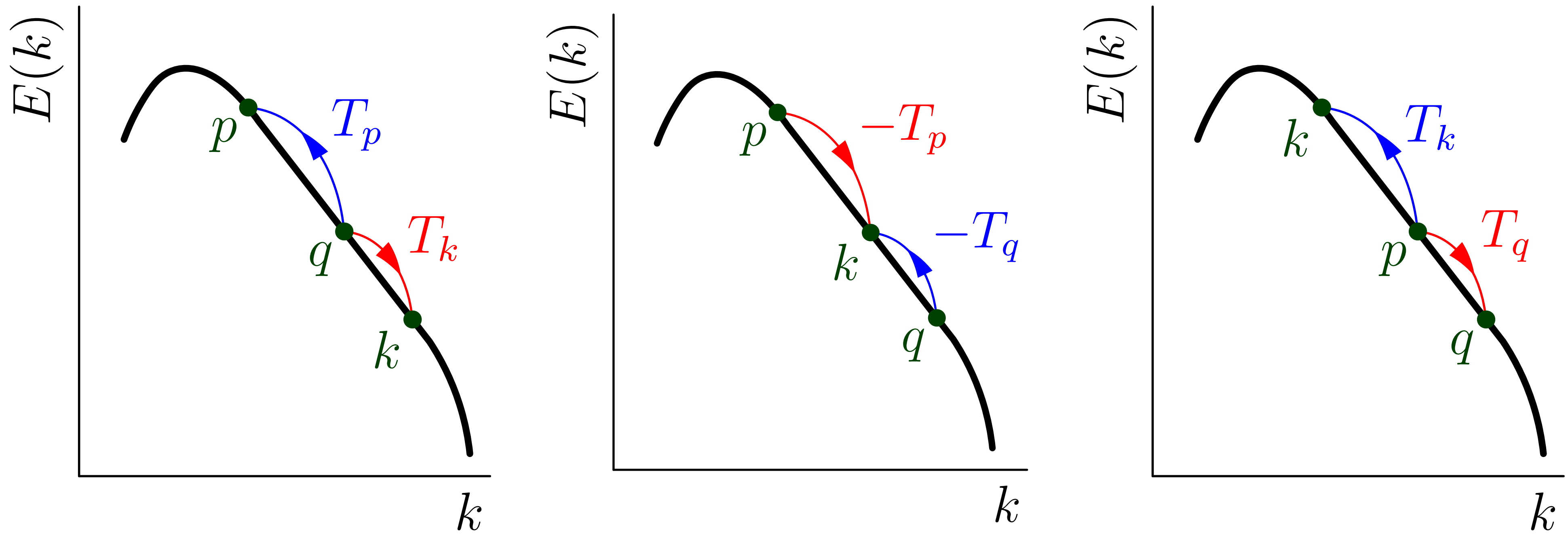
Uniform flux

- Although the independence of the local inertial-range energy flux with wavenumber is one of the key hypothesis underlying Kolmogorov's famous 5/3 power-law form for the kinetic energy spectrum, it has never been directly tested, either experimentally or numerically.
- To validate Kolmogorov's uniform flux hypothesis in a high-resolution pseudospectral code, detailed wavenumber constraints must be imposed on the convolution.
- The key tool needed is the partial fast Fourier transform, where the summation limits are restricted by a spatially-dependent constraint.

Uniform flux

- Although the independence of the local inertial-range energy flux with wavenumber is one of the key hypothesis underlying Kolmogorov's famous 5/3 power-law form for the kinetic energy spectrum, it has never been directly tested, either experimentally or numerically.
- To validate Kolmogorov's uniform flux hypothesis in a high-resolution pseudospectral code, detailed wavenumber constraints must be imposed on the convolution.
- The key tool needed is the partial fast Fourier transform, where the summation limits are restricted by a spatially-dependent constraint.
- To this end, we have improved on previous attempts [Ying 2009] to develop a partial FFT based on the fractional Fourier transform and Bluestein's algorithm [Bluestein 1970].

Flux Decomposition for a Single $(\boldsymbol{k}, \boldsymbol{p}, \boldsymbol{q})$ Triad



$L_k = T_k$, $S_k = 0$ | $L_k = -T_p$, $S_k = -T_q$ | $L_k = 0$, $S_k = T_k$

- Note that energy is conserved: $L_k + S_k = T_k = -T_p - T_q$. Thus

$$L_k = \mathrm{Re} \sum_{\substack{|\boldsymbol{k}|=k \\ |\boldsymbol{p}|<k \\ |\boldsymbol{k}-\boldsymbol{p}|<k}} M_{\boldsymbol{k},\boldsymbol{p}}\, \omega_{\boldsymbol{p}}\, \omega_{\boldsymbol{k}-\boldsymbol{p}}\, \omega_{\boldsymbol{k}}^* - \mathrm{Re} \sum_{\substack{|\boldsymbol{k}|=k \\ |\boldsymbol{p}|<k \\ |\boldsymbol{k}-\boldsymbol{p}|>k}} M_{\boldsymbol{p},\boldsymbol{k}-\boldsymbol{p}}\, \omega_{\boldsymbol{k}}\, \omega_{\boldsymbol{k}-\boldsymbol{p}}\, \omega_{\boldsymbol{p}}^*.$$

Discrete Cyclic Convolution

- The FFT provides an efficient tool for computing the *discrete cyclic convolution*

$$\sum_{p=0}^{N-1} F_p G_{k-p},$$

where the vectors F and G have period N.

Discrete Cyclic Convolution

- The FFT provides an efficient tool for computing the *discrete cyclic convolution*

$$\sum_{p=0}^{N-1} F_p G_{k-p},$$

where the vectors F and G have period N.

- Define the *Nth primitive root of unity:*

$$\zeta_N = \exp\left(\frac{2\pi i}{N}\right).$$

Discrete Cyclic Convolution

- The FFT provides an efficient tool for computing the *discrete cyclic convolution*

$$\sum_{p=0}^{N-1} F_p G_{k-p},$$

where the vectors F and G have period N.

- Define the *Nth primitive root of unity:*

$$\zeta_N = \exp\left(\frac{2\pi i}{N}\right).$$

- The fast Fourier transform (FFT) method exploits the properties that $\zeta_N^r = \zeta_{N/r}$ and $\zeta_N^N = 1$.

Discrete Cyclic Convolution

- The FFT provides an efficient tool for computing the *discrete cyclic convolution*

$$\sum_{p=0}^{N-1} F_p G_{k-p},$$

where the vectors F and G have period N.

- Define the *Nth primitive root of unity:*

$$\zeta_N = \exp\left(\frac{2\pi i}{N}\right).$$

- The fast Fourier transform (FFT) method exploits the properties that $\zeta_N^r = \zeta_{N/r}$ and $\zeta_N^N = 1$.

- However, the pseudospectral method requires a *linear convolution.*

- The unnormalized *backwards discrete Fourier transform* of $\{F_k : k = 0, \ldots, N\}$ is

$$f_j \doteq \sum_{k=0}^{N-1} \zeta_N^{jk} F_k \qquad j = 0, \ldots, N-1.$$

- The unnormalized *backwards discrete Fourier transform* of $\{F_k : k = 0, \ldots, N\}$ is

$$f_j \doteq \sum_{k=0}^{N-1} \zeta_N^{jk} F_k \qquad j = 0, \ldots, N-1.$$

- The corresponding *forward transform is*

$$F_k \doteq \frac{1}{N} \sum_{j=0}^{N-1} \zeta_N^{-kj} f_j \qquad k = 0, \ldots, N-1.$$

- The unnormalized *backwards discrete Fourier transform* of $\{F_k : k = 0, \ldots, N\}$ is

$$f_j \doteq \sum_{k=0}^{N-1} \zeta_N^{jk} F_k \qquad j = 0, \ldots, N-1.$$

- The corresponding *forward transform is*

$$F_k \doteq \frac{1}{N} \sum_{j=0}^{N-1} \zeta_N^{-kj} f_j \qquad k = 0, \ldots, N-1.$$

- The orthogonality of this transform pair follows from

$$\sum_{j=0}^{N-1} \zeta_N^{\ell j} = \begin{cases} N & \text{if } \ell = sN \text{ for } s \in \mathbb{Z}, \\ \dfrac{1-\zeta_N^{\ell N}}{1-\zeta_N^{\ell}} = 0 & \text{otherwise.} \end{cases}$$

Convolution Theorem

$$\sum_{j=0}^{N-1} f_j g_j \zeta_N^{-jk} = \sum_{j=0}^{N-1} \zeta_N^{-jk} \left(\sum_{p=0}^{N-1} \zeta_N^{jp} F_p \right) \left(\sum_{q=0}^{N-1} \zeta_N^{jq} G_q \right)$$
$$= \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} F_p G_q \sum_{j=0}^{N-1} \zeta_N^{(-k+p+q)j}$$
$$= N \sum_{s} \sum_{p=0}^{N-1} F_p G_{k-p+sN}.$$

- The terms indexed by $s \neq 0$ are *aliases;* we need to remove them!

Convolution Theorem

$$\sum_{j=0}^{N-1} f_j g_j \zeta_N^{-jk} = \sum_{j=0}^{N-1} \zeta_N^{-jk} \left(\sum_{p=0}^{N-1} \zeta_N^{jp} F_p \right) \left(\sum_{q=0}^{N-1} \zeta_N^{jq} G_q \right)$$
$$= \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} F_p G_q \sum_{j=0}^{N-1} \zeta_N^{(-k+p+q)j}$$
$$= N \sum_{s} \sum_{p=0}^{N-1} F_p G_{k-p+sN}.$$

- The terms indexed by $s \neq 0$ are *aliases;* we need to remove them!
- If only the first m entries of the input vectors are nonzero, aliases can be avoided by *zero padding* input data vectors of length m to length $N \geq 2m - 1$.

Convolution Theorem

$$\sum_{j=0}^{N-1} f_j g_j \zeta_N^{-jk} = \sum_{j=0}^{N-1} \zeta_N^{-jk} \left(\sum_{p=0}^{N-1} \zeta_N^{jp} F_p \right) \left(\sum_{q=0}^{N-1} \zeta_N^{jq} G_q \right)$$
$$= \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} F_p G_q \sum_{j=0}^{N-1} \zeta_N^{(-k+p+q)j}$$
$$= N \sum_s \sum_{p=0}^{N-1} F_p G_{k-p+sN}.$$

- The terms indexed by $s \neq 0$ are *aliases;* we need to remove them!
- If only the first m entries of the input vectors are nonzero, aliases can be avoided by *zero padding* input data vectors of length m to length $N \geq 2m - 1$.
- *Explicit zero padding* prevents mode $m - 1$ from beating with itself, wrapping around to contaminate mode $N = 0 \bmod N$.

Implicit Dealiasing

- Let $N = 2m$. For $j = 0, \ldots, 2m - 1$ we want to compute

$$f_j = \sum_{k=0}^{2m-1} \zeta_{2m}^{jk} F_k.$$

Implicit Dealiasing

- Let $N = 2m$. For $j = 0, \ldots, 2m-1$ we want to compute

$$f_j = \sum_{k=0}^{2m-1} \zeta_{2m}^{jk} F_k.$$

- If $F_k = 0$ for $k \geq m$, one can easily avoid looping over the unwanted zero Fourier modes by decimating in wavenumber:

$$f_{2\ell} = \sum_{k=0}^{m-1} \zeta_{2m}^{2\ell k} F_k = \sum_{k=0}^{m-1} \zeta_m^{\ell k} F_k,$$

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

Implicit Dealiasing

- Let $N = 2m$. For $j = 0, \ldots, 2m-1$ we want to compute

$$f_j = \sum_{k=0}^{2m-1} \zeta_{2m}^{jk} F_k.$$

- If $F_k = 0$ for $k \geq m$, one can easily avoid looping over the unwanted zero Fourier modes by decimating in wavenumber:

$$f_{2\ell} = \sum_{k=0}^{m-1} \zeta_{2m}^{2\ell k} F_k = \sum_{k=0}^{m-1} \zeta_m^{\ell k} F_k,$$

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

- This requires computing two subtransforms, each of size m, for an overall computational scaling of order $2m \log_2 m = N \log_2 m$.

- Parallelized multidimensional implicit dealiasing routines have been implemented as a software layer **FFTW++** (v 2.02) on top of the **FFTW** library under the Lesser GNU Public License:

  `http://fftwpp.sourceforge.net/`

Fast Variably Restricted Dealiased Convolution

- We need a practical algorithm for computing many *partial* Fourier transforms at once:

$$u_{\boldsymbol{j}} \doteq \sum_{|\boldsymbol{k}|<c(\boldsymbol{j})} \zeta_N^{\boldsymbol{k}\cdot\boldsymbol{j}} U_{\boldsymbol{k}}$$

where $\zeta_N = e^{2\pi i/N}$ is the Nth primitive root of unity.

Fast Variably Restricted Dealiased Convolution

- We need a practical algorithm for computing many *partial* Fourier transforms at once:

$$u_{\boldsymbol{j}} \doteq \sum_{|\boldsymbol{k}|<c(\boldsymbol{j})} \zeta_N^{\boldsymbol{k}\cdot\boldsymbol{j}} U_{\boldsymbol{k}}$$

  where $\zeta_N = e^{2\pi i/N}$ is the Nth primitive root of unity.

- Here $c(\boldsymbol{j})$ is a spatially-dependent constraint on the summation limits.

Fast Variably Restricted Dealiased Convolution

- We need a practical algorithm for computing many *partial* Fourier transforms at once:

$$u_{\boldsymbol{j}} \doteq \sum_{|\boldsymbol{k}|<c(\boldsymbol{j})} \zeta_N^{\boldsymbol{k}\cdot\boldsymbol{j}} U_{\boldsymbol{k}}$$

  where $\zeta_N = e^{2\pi i/N}$ is the Nth primitive root of unity.

- Here $c(\boldsymbol{j})$ is a spatially-dependent constraint on the summation limits.

- Goal: obtain a 'fast' computational scaling, following Ying & Fomel [2009] but with a smaller overall coefficient.

Partial 1D Fourier Transform

- Let $\zeta^{\alpha} \doteq \zeta_{1/a} = e^{2\pi i\alpha}$.

Partial 1D Fourier Transform

- Let $\zeta^\alpha \doteq \zeta_{1/a} = e^{2\pi i\alpha}$.

- The unnormalized backward discrete partial Fourier transform of a complex vector $\{F_k : k = 0, \ldots, N-1\}$ is defined as

$$f_j \doteq \sum_{k=0}^{c(j)} \zeta^{\alpha jk} F_k, \qquad j = 0, \ldots, N-1.$$

Special case of partial 1D FFT: $c(j) = j$

- Given inputs $\{F_k : k = 0, \ldots, N-1\}$,

$$f_j \doteq \sum_{k=0}^{j} \zeta^{\alpha jk} F_k, \qquad j = 0, \ldots, N-1.$$

Special case of partial 1D FFT: $c(j) = j$

- Given inputs $\{F_k : k = 0, \ldots, N-1\}$,

$$f_j \doteq \sum_{k=0}^{j} \zeta^{\alpha j k} F_k, \qquad j = 0, \ldots, N-1.$$

- Since $jk = \frac{1}{2}\left[j^2 + k^2 - (j-k)^2\right]$, [Bluestein 1970]

$$f_j = \sum_{k=0}^{j} \zeta^{\frac{\alpha}{2}\left[j^2+k^2-(j-k)^2\right]} F_k = \zeta^{\alpha j^2/2} \sum_{k=0}^{j} \zeta^{\alpha k^2/2} F_k \zeta^{-\alpha(j-k)^2/2},$$

Special case of partial 1D FFT: $c(j) = j$

- Given inputs $\{F_k : k = 0, \ldots, N-1\}$,

$$f_j \doteq \sum_{k=0}^{j} \zeta^{\alpha jk} F_k, \qquad j = 0, \ldots, N-1.$$

- Since $jk = \frac{1}{2}\left[j^2 + k^2 - (j-k)^2\right]$, [Bluestein 1970]

$$f_j = \sum_{k=0}^{j} \zeta^{\frac{\alpha}{2}\left[j^2+k^2-(j-k)^2\right]} F_k = \zeta^{\alpha j^2/2} \sum_{k=0}^{j} \zeta^{\alpha k^2/2} F_k \zeta^{-\alpha(j-k)^2/2},$$

- This can be written as the convolution of the two sequences $g_j = \zeta_2^{\alpha j^2}$ and $h_k = g_k F_k$:

$$f_j = g_j \sum_{k=0}^{j} h_k \overline{g}_{j-k}.$$

Partial FFT: Special Case $c(j) = (pj + s)/q$

- Here p, q, and s are integers, with $p \neq 0$ and

$$f_j \doteq \sum_{k=0}^{\lfloor (pj+s)/q \rfloor} \zeta^{\alpha jk} F_k, \qquad j = 0, \ldots, M-1.$$

Partial FFT: Special Case $c(j) = (pj + s)/q$

- Here p, q, and s are integers, with $p \neq 0$ and

$$f_j \doteq \sum_{k=0}^{\lfloor (pj+s)/q \rfloor} \zeta^{\alpha jk} F_k, \qquad j = 0, \ldots, M-1.$$

- Let $pj + s = qn + r$, with $n = 0, \ldots, N-1$. Then

$$\begin{aligned} f_j &= \sum_{k=0}^{n} \zeta_p^{\alpha(qn+r-s)k} F_k \\ &= \sum_{k=0}^{n} \zeta_{2p}^{\alpha q\left[n^2+k^2-(n-k)^2\right]} \zeta_p^{\alpha(r-s)k} F_k \\ &= \zeta_{2p}^{\alpha q n^2} \sum_{k=0}^{n} \zeta_{2p}^{-\alpha q(n-k)^2} \zeta_{2p}^{\alpha q k^2} \zeta_p^{\alpha(r-s)k} F_k \end{aligned}$$

- On setting $g_k = \zeta_{2p}^{\alpha q k^2}$ and $h_k = g_k \zeta_p^{\alpha(r-s)k} F_k$, the result can be written as a convolution of two sequences $\{h_k\}$ and $\{g_k\}$:

$$f_j = g_n \sum_{k=0}^{n} h_k \overline{g}_{n-k}, \qquad j = 0, \ldots, M-1.$$

- On setting $g_k = \zeta_{2p}^{\alpha q k^2}$ and $h_k = g_k \zeta_p^{\alpha(r-s)k} F_k$, the result can be written as a convolution of two sequences $\{h_k\}$ and $\{g_k\}$:

$$f_j = g_n \sum_{k=0}^{n} h_k \overline{g}_{n-k}, \qquad j = 0, \ldots, M-1.$$

- This general algorithm is only efficient when $p = 1$ or $q = 1$.

- On setting $g_k = \zeta_{2p}^{\alpha q k^2}$ and $h_k = g_k \zeta_p^{\alpha(r-s)k} F_k$, the result can be written as a convolution of two sequences $\{h_k\}$ and $\{g_k\}$:

$$f_j = g_n \sum_{k=0}^{n} h_k \overline{g}_{n-k}, \qquad j = 0, \ldots, M-1.$$

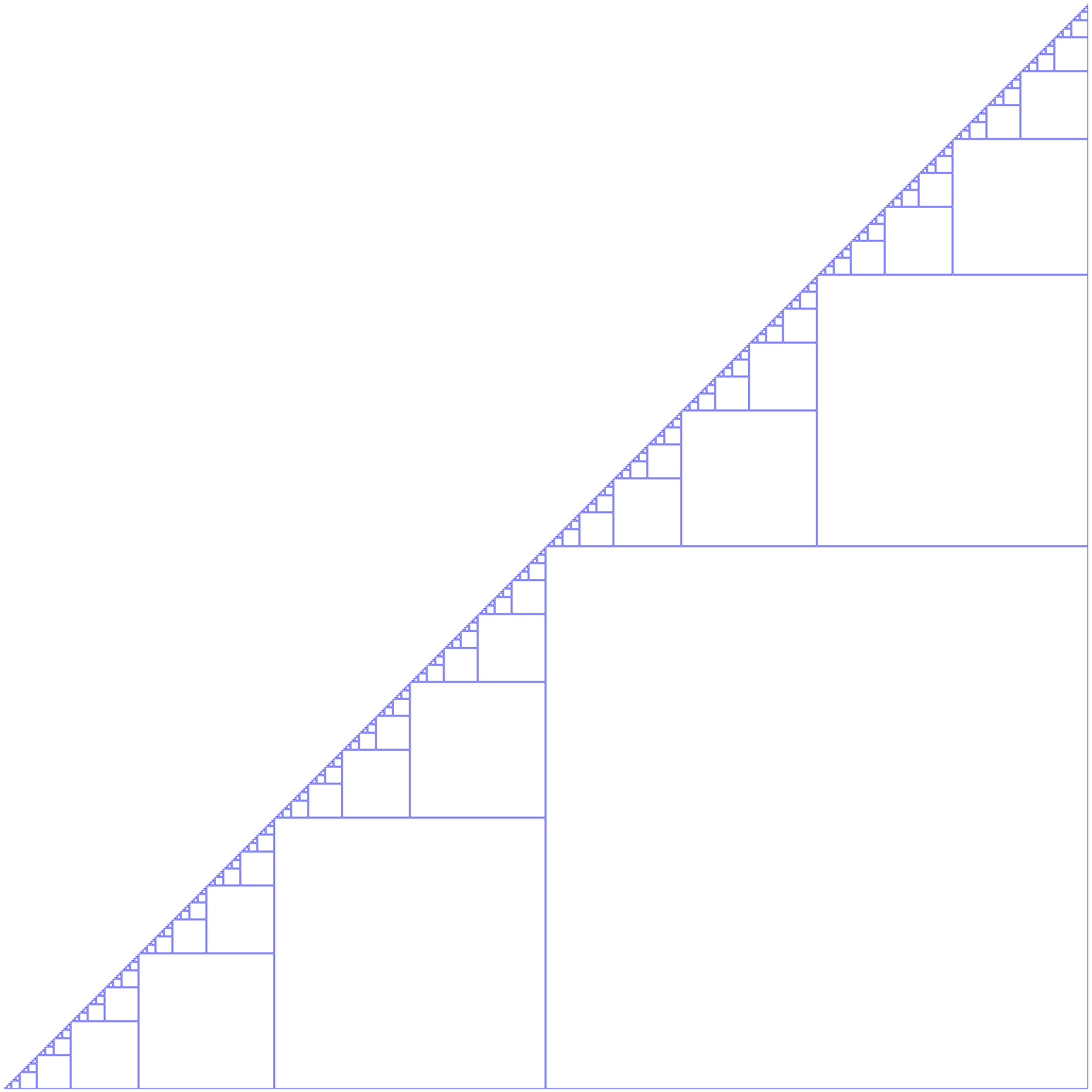
- This general algorithm is only efficient when $p = 1$ or $q = 1$.
- A similar procedure can be used to compute partial convolutions.

- On setting $g_k = \zeta_{2p}^{\alpha q k^2}$ and $h_k = g_k \zeta_p^{\alpha(r-s)k} F_k$, the result can be written as a convolution of two sequences $\{h_k\}$ and $\{g_k\}$:

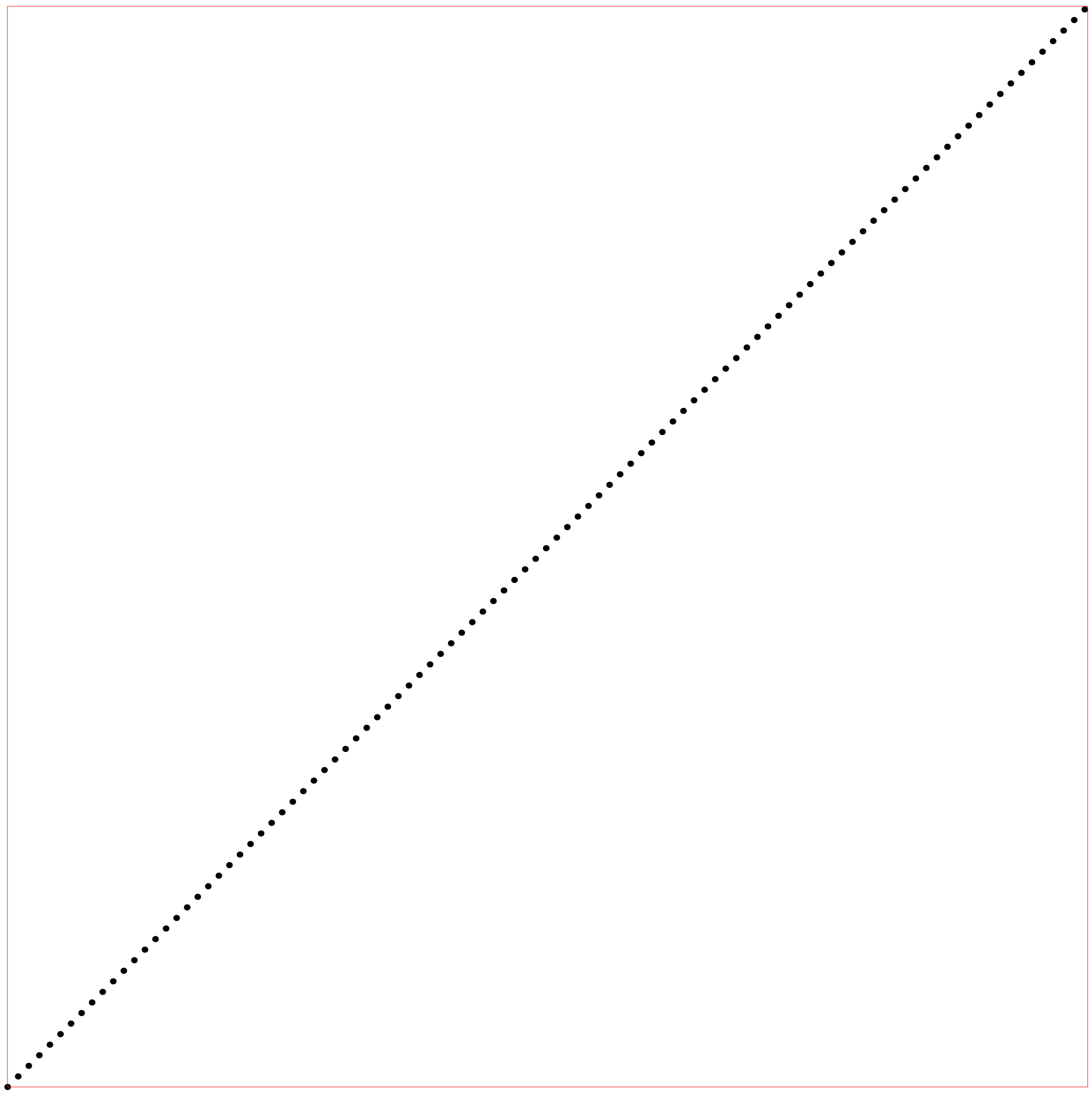
$$f_j = g_n \sum_{k=0}^{n} h_k \overline{g}_{n-k}, \qquad j = 0, \ldots, M-1.$$

- This general algorithm is only efficient when $p = 1$ or $q = 1$.
- A similar procedure can be used to compute partial convolutions.
- The technique can be readily extended to higher dimensions.

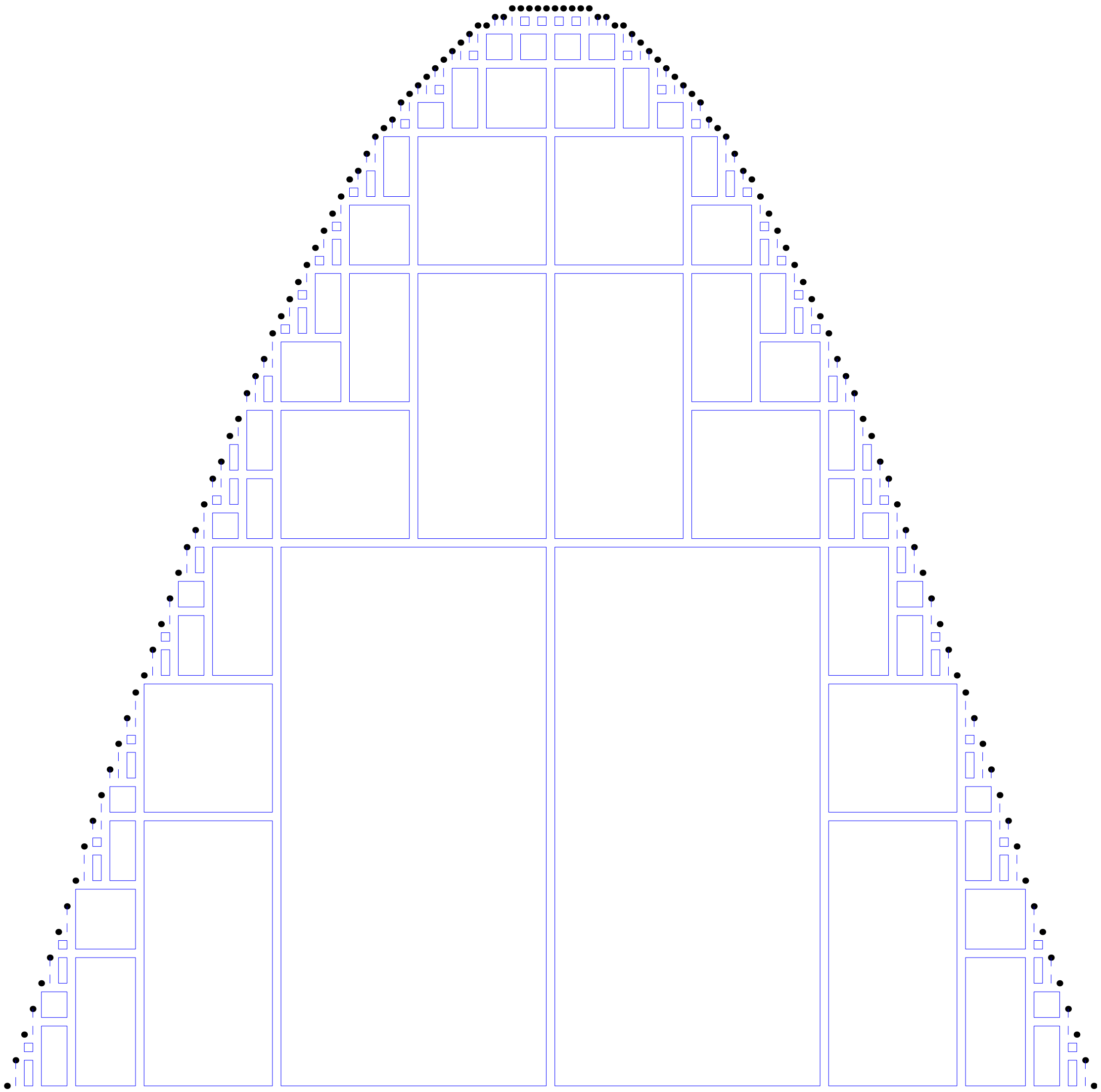
Rectangular subdivision for $c(j) = j$



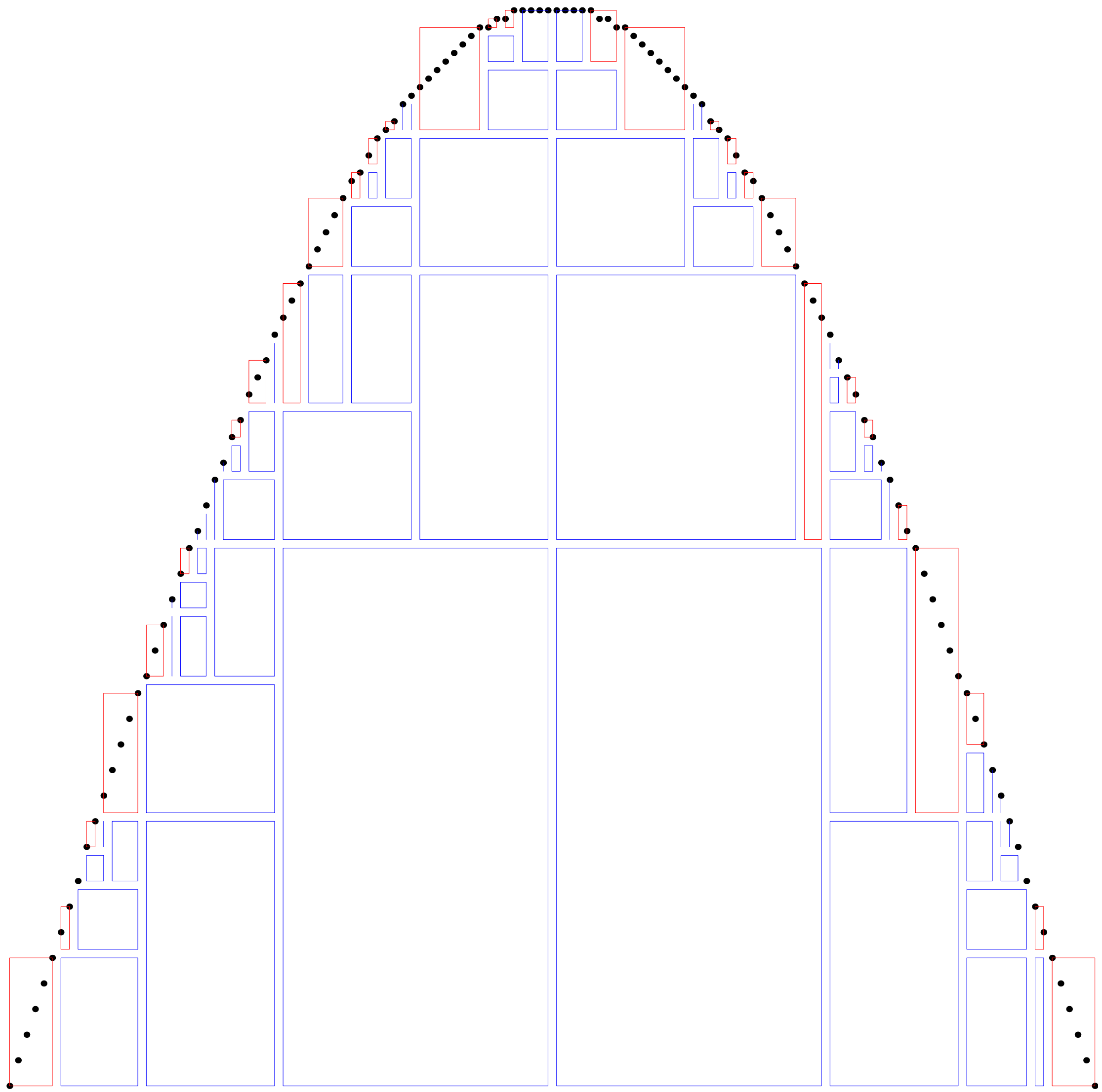
Triangular subdivision for $c(j) = j$



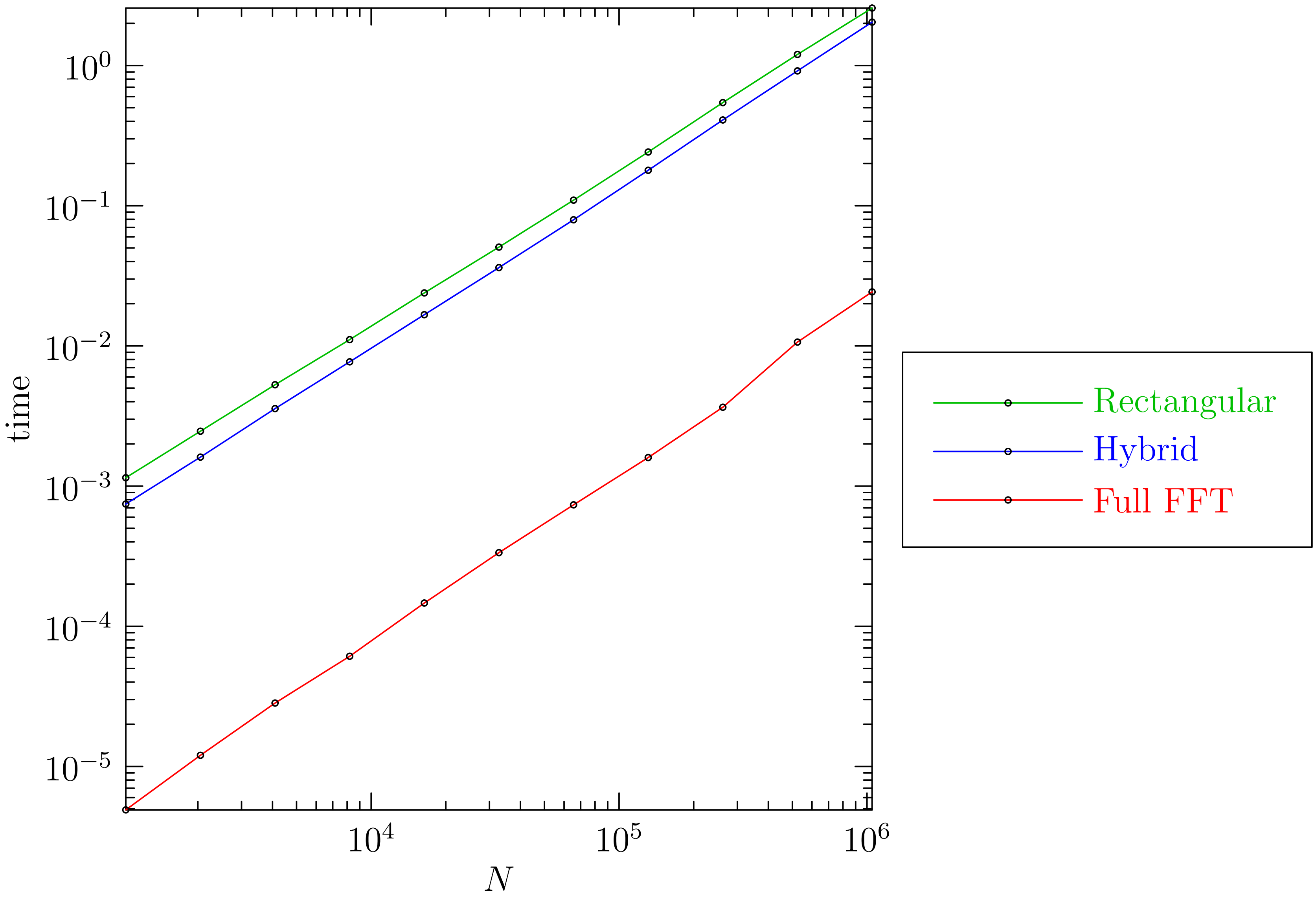
Rectangular subdivision for $c(j) = (N-1)\sin \pi j/(N-1)$



Hybrid subdivision for
$c(j) = (N-1)\sin \pi j/(N-1)$



Computation time



Casimir Invariants

- Inviscid unforced two dimensional turbulence has uncountably many other Casimir invariants.

Casimir Invariants

- Inviscid unforced two dimensional turbulence has uncountably many other Casimir invariants.

- Any continuously differentiable function of the (scalar) vorticity is conserved by the nonlinearity:

$$\frac{d}{dt}\int f(\omega)\,d\boldsymbol{x} = \int f'(\omega)\frac{\partial\omega}{\partial t}\,d\boldsymbol{x} = -\int f'(\omega)\boldsymbol{u}\boldsymbol{\cdot}\boldsymbol{\nabla}\omega\,d\boldsymbol{x}$$
$$= -\int \boldsymbol{u}\boldsymbol{\cdot}\boldsymbol{\nabla} f(\omega)\,d\boldsymbol{x} = \int f(\omega)\boldsymbol{\nabla}\boldsymbol{\cdot}\boldsymbol{u}\,d\boldsymbol{x} = 0.$$

Casimir Invariants

- Inviscid unforced two dimensional turbulence has uncountably many other Casimir invariants.
- Any continuously differentiable function of the (scalar) vorticity is conserved by the nonlinearity:

$$
\begin{aligned}
\frac{d}{dt}\int f(\omega)\, d\boldsymbol{x} = \int f'(\omega)\frac{\partial\omega}{\partial t}\, d\boldsymbol{x} &= -\int f'(\omega)\boldsymbol{u}\cdot\boldsymbol{\nabla}\omega\, d\boldsymbol{x} \\
= -\int \boldsymbol{u}\cdot\boldsymbol{\nabla} f(\omega)\, d\boldsymbol{x} &= \int f(\omega)\boldsymbol{\nabla}\cdot\boldsymbol{u}\, d\boldsymbol{x} = 0.
\end{aligned}
$$

- Do these invariants also play a fundamental role in the turbulent dynamics, in addition to the quadratic (energy and enstrophy) invariants? Do they exhibit cascades?

Conclusions

- One should distinguish between nonlocal transfer and flux.

Conclusions

- One should distinguish between nonlocal transfer and flux.
- To compute this decomposition efficiently, one needs a partial convolution.

Conclusions

- One should distinguish between nonlocal transfer and flux.
- To compute this decomposition efficiently, one needs a partial convolution.
- Partial dealiased convolutions can be used to compute detailed inertial-range flux profiles and for the first time verify a key underpinning assumption of Kolmogorov's famous power-law conjecture for turbulence.

Conclusions

- One should distinguish between nonlocal transfer and flux.
- To compute this decomposition efficiently, one needs a partial convolution.
- Partial dealiased convolutions can be used to compute detailed inertial-range flux profiles and for the first time verify a key underpinning assumption of Kolmogorov's famous power-law conjecture for turbulence.
- This will allow us to verify and exploit inertial-range self-similarity in 2D turbulence and study the *flux locality profile.*

Conclusions

- One should distinguish between nonlocal transfer and flux.
- To compute this decomposition efficiently, one needs a partial convolution.
- Partial dealiased convolutions can be used to compute detailed inertial-range flux profiles and for the first time verify a key underpinning assumption of Kolmogorov's famous power-law conjecture for turbulence.
- This will allow us to verify and exploit inertial-range self-similarity in 2D turbulence and study the *flux locality profile.*
- The locality profile can be used to infer the effective eddy damping contribution from each of truncated (subgrid) modes, allowing us to build a phenomenological dynamic subgrid model that on average removes the right amount of energy from each of the scales near the subgrid wavenumber cutoff.

References

[Bluestein 1970] L. I. Bluestein, IEEE Trans. Audio and Electroacoustics, **18**:451, 1970.

[Bowman & Roberts 2011] J. C. Bowman & M. Roberts, SIAM J. Sci. Comput., **33**:386, 2011.

[Roberts & Bowman 2016] M. Roberts & J. C. Bowman, submitted to SIAM J. Sci. Comput., 2016.

[Ying & Fomel 2009] L. Ying & S. Fomel, Multiscale Modeling and Simulation, **8**:110, 2009.