On the Global Attractor of 2D Incompressible Turbulence with Random Forcing

John C. Bowman and Pedram Emami
Department of Mathematical and Statistical Sciences
University of Alberta

October 25, 2017

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

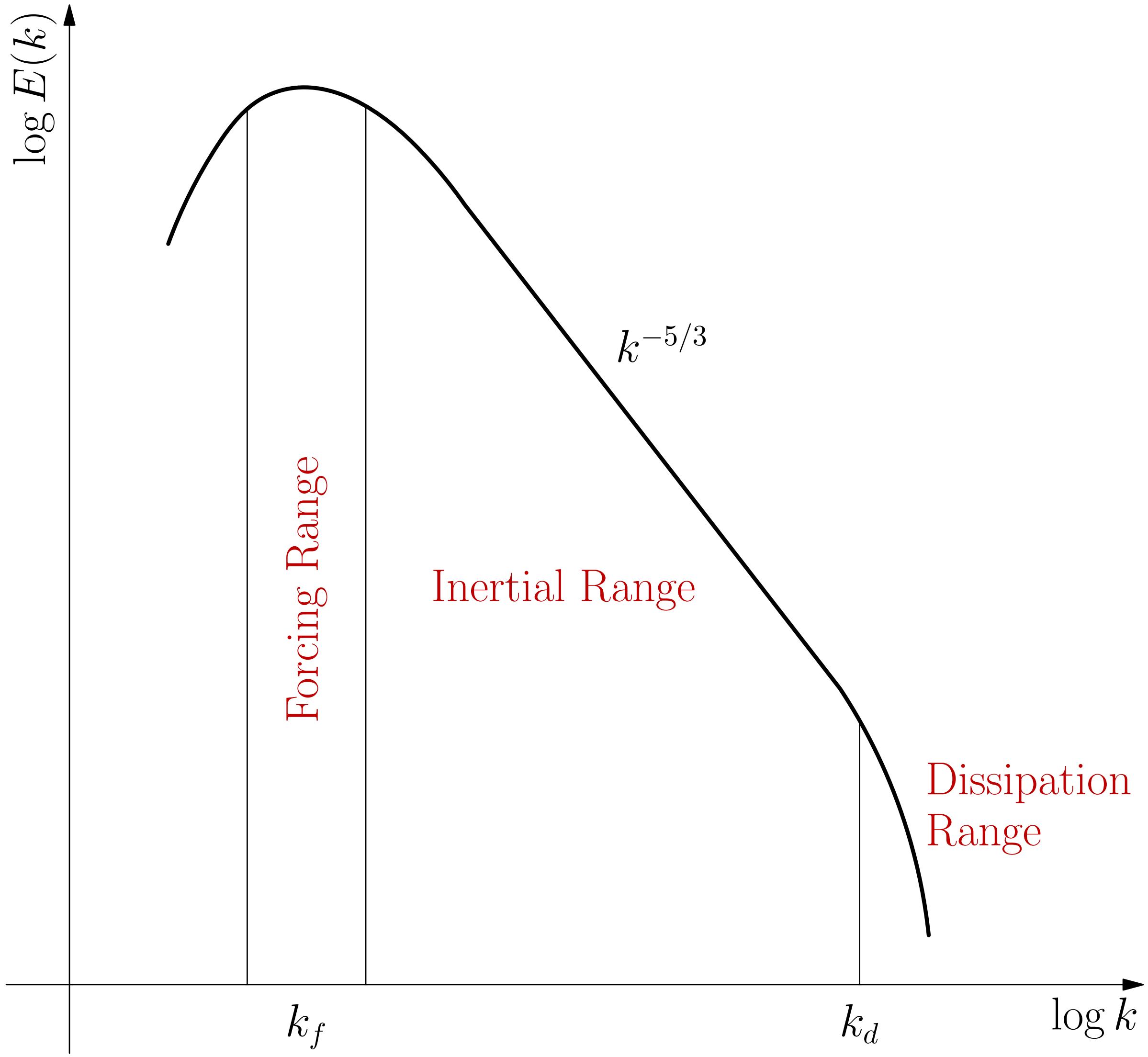
Big whirls have little whirls that feed on their velocity, and little whirls have littler whirls and so on to viscosity... [Richardson 1922]

- 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.

3D Energy Cascade



2D Incompressible Turbulence

- In 2D, where $\boldsymbol{u}$ maps a plane normal to $\hat{\boldsymbol{z}}$ to R^2, the vorticity vector $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$ is always perpendicular to $\boldsymbol{u}$.

2D Incompressible Turbulence

- In 2D, where $\boldsymbol{u}$ maps a plane normal to $\hat{\boldsymbol{z}}$ to R^2, the vorticity vector $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$ is always perpendicular to $\boldsymbol{u}$.
- Navier–Stokes equation for the scalar vorticity $\omega = \hat{\boldsymbol{z}} \boldsymbol{\cdot} \boldsymbol{\nabla} \times \boldsymbol{u}$:

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

2D Incompressible Turbulence

- In 2D, where $\boldsymbol{u}$ maps a plane normal to $\hat{\boldsymbol{z}}$ to R^2, the vorticity vector $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$ is always perpendicular to $\boldsymbol{u}$.

- Navier–Stokes equation for the scalar vorticity $\omega = \hat{\boldsymbol{z}} \boldsymbol{\cdot} \boldsymbol{\nabla} \times \boldsymbol{u}$:

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

- The incompressibility condition $\boldsymbol{\nabla} \boldsymbol{\cdot} \boldsymbol{u} = 0$ can be exploited to find $\boldsymbol{u}$ in terms of ω:

$$\boldsymbol{\nabla} \omega \times \hat{\boldsymbol{z}} = \boldsymbol{\nabla} \times \hat{\boldsymbol{z}} \omega = \boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{u}) = \boldsymbol{\nabla} (\boldsymbol{\nabla} \boldsymbol{\cdot} \boldsymbol{u}) - \nabla^2 \boldsymbol{u} = -\nabla^2 \boldsymbol{u}.$$

2D Incompressible Turbulence

- In 2D, where $\boldsymbol{u}$ maps a plane normal to $\hat{\boldsymbol{z}}$ to R^2, the vorticity vector $\boldsymbol{\omega} = \boldsymbol{\nabla}\times\boldsymbol{u}$ is always perpendicular to $\boldsymbol{u}$.
- Navier–Stokes equation for the scalar vorticity $\omega = \hat{\boldsymbol{z}}\cdot\boldsymbol{\nabla}\times\boldsymbol{u}$:

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

- The incompressibility condition $\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$ can be exploited to find $\boldsymbol{u}$ in terms of ω:

$$\boldsymbol{\nabla}\omega\times\hat{\boldsymbol{z}} = \boldsymbol{\nabla}\times\hat{\boldsymbol{z}}\omega = \boldsymbol{\nabla}\times(\boldsymbol{\nabla}\times\boldsymbol{u}) = \boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u}) - \nabla^2\boldsymbol{u} = -\nabla^2\boldsymbol{u}.$$

- Thus $\boldsymbol{u} = \hat{\boldsymbol{z}}\times\boldsymbol{\nabla}\nabla^{-2}\omega$. In Fourier space:

$$\frac{d\omega_{\boldsymbol{k}}}{dt} = S_{\boldsymbol{k}} - \nu k^2\omega_{\boldsymbol{k}} + f_{\boldsymbol{k}},$$

where $S_{\boldsymbol{k}} = \sum_{\boldsymbol{q}} \frac{\hat{\boldsymbol{z}}\times\boldsymbol{q}\cdot\boldsymbol{k}}{q^2}\overline{\omega_{\boldsymbol{q}}}\,\overline{\omega_{-\boldsymbol{k}-\boldsymbol{q}}} = \sum_{\boldsymbol{p},\boldsymbol{q}} \frac{\epsilon_{\boldsymbol{kpq}}}{q^2}\overline{\omega_{\boldsymbol{p}}}\,\overline{\omega_{\boldsymbol{q}}}.$

Here $\epsilon_{\boldsymbol{kpq}} \doteq \hat{\boldsymbol{z}} \cdot \boldsymbol{p} \times \boldsymbol{q}\, \delta_{\boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q}}$ is antisymmetric under permutation of any two indices.

$$\frac{d\omega_{\boldsymbol{k}}}{dt} + \nu k^2 \omega_{\boldsymbol{k}} = \sum_{\boldsymbol{p}} \sum_{\boldsymbol{q}} \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \overline{\omega_{\boldsymbol{p}}}\, \overline{\omega_{\boldsymbol{q}}} + f_{\boldsymbol{k}},$$

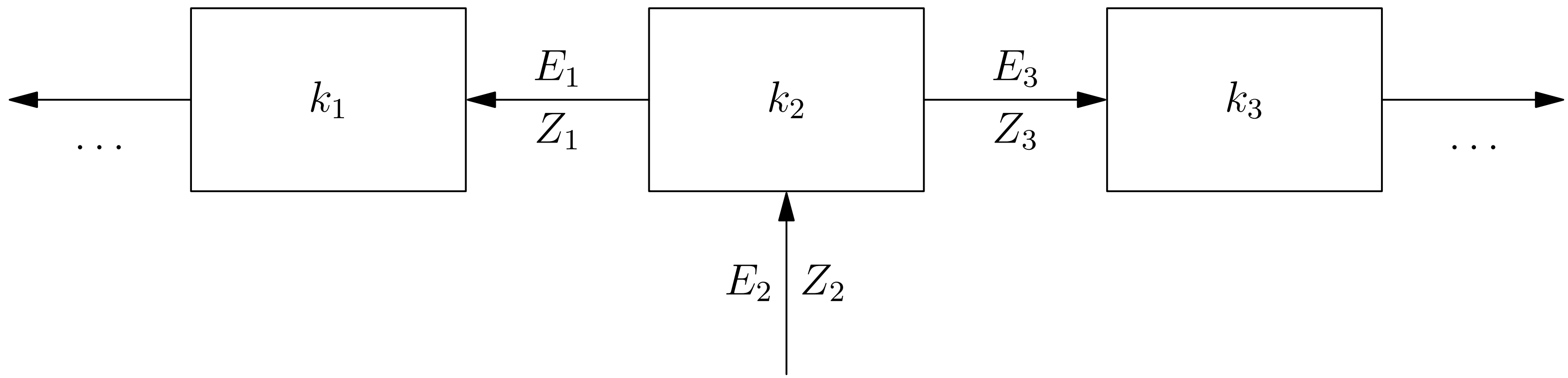
- When $\nu = f_{\boldsymbol{k}} = 0$:

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

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

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

Fjørtoft Dual Cascade Scenario

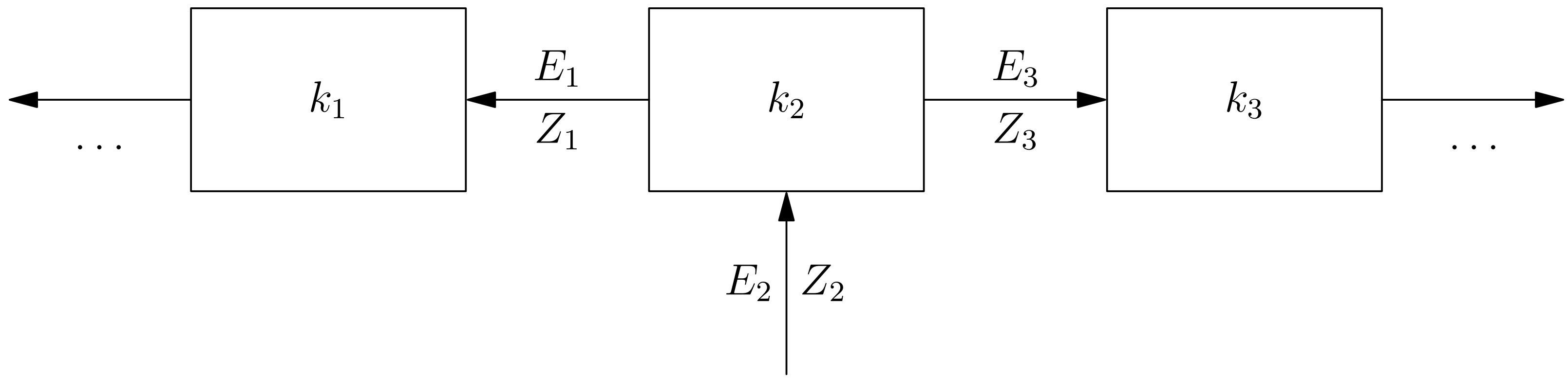


$$E_2 = E_1 + E_3, \qquad Z_2 = Z_1 + Z_3, \qquad Z_i \approx k_i^2 E_i.$$

- When $k_1 = k$, $k_2 = 2k$, and $k_3 = 4k$:

$$E_1 \approx \frac{4}{5} E_2, \quad Z_1 \approx \frac{1}{5} Z_2, \qquad E_3 \approx \frac{1}{5} E_2, \quad Z_3 \approx \frac{4}{5} Z_2.$$

Fjørtoft Dual Cascade Scenario



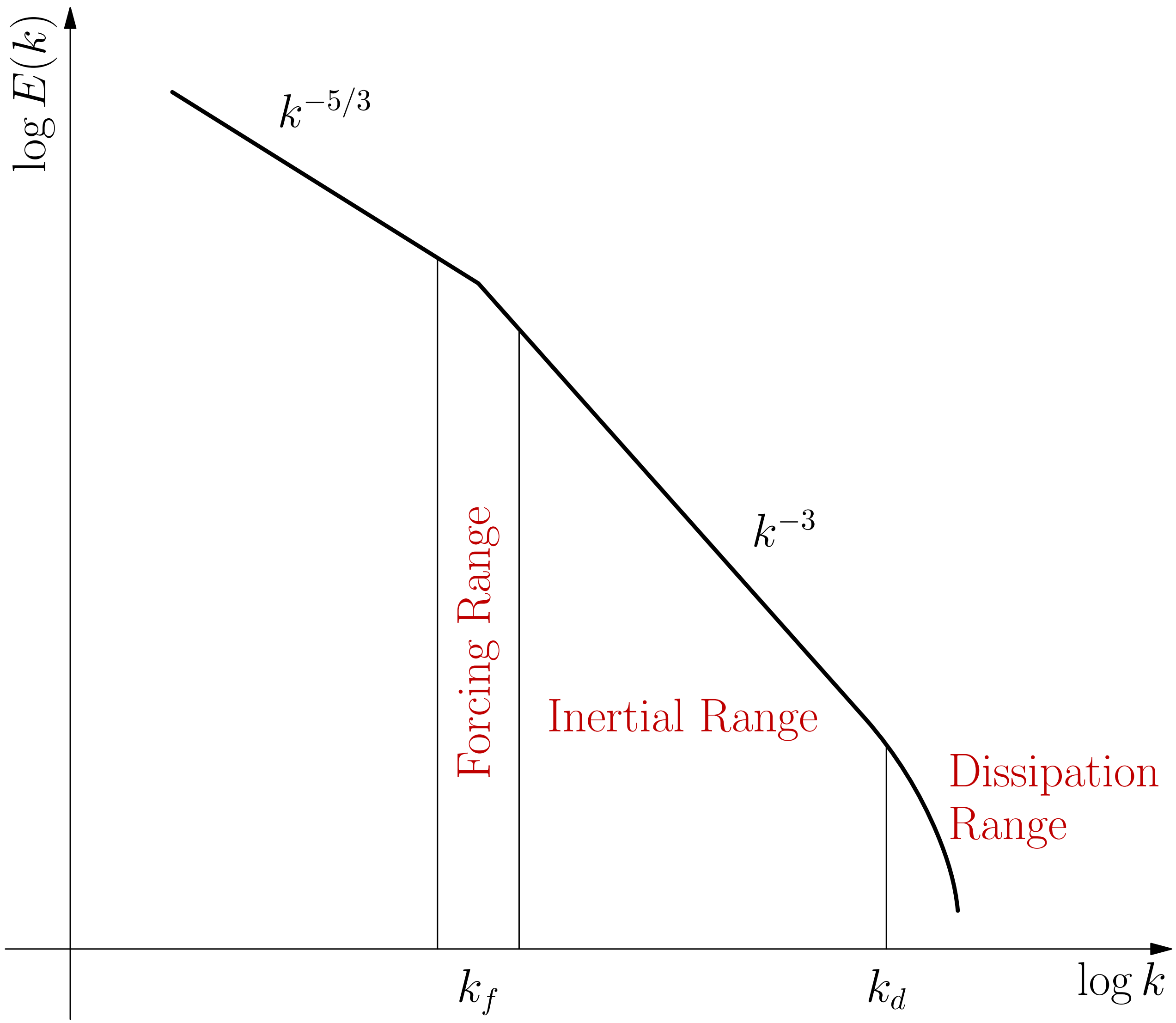
$$E_2 = E_1 + E_3, \qquad Z_2 = Z_1 + Z_3, \qquad Z_i \approx k_i^2 E_i.$$

- When $k_1 = k$, $k_2 = 2k$, and $k_3 = 4k$:

$$E_1 \approx \frac{4}{5} E_2, \quad Z_1 \approx \frac{1}{5} Z_2, \qquad E_3 \approx \frac{1}{5} E_2, \quad Z_3 \approx \frac{4}{5} Z_2.$$

- Fjørtoft [1953]: energy cascades to large scales and enstrophy cascades to small scales.

2D Energy Cascade



2D Turbulence: Mathematical Formulation

- Consider the Navier–Stokes equations for 2D incompressible homogeneous isotropic turbulence with density $\rho = 1$:

$$\frac{\partial \boldsymbol{u}}{\partial t} - \nu \nabla^2 \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} P = \boldsymbol{F},$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0,$$

$$\int_\Omega \boldsymbol{u} \, d\boldsymbol{x} = \boldsymbol{0}, \qquad \int_\Omega \boldsymbol{F} \, d\boldsymbol{x} = \boldsymbol{0},$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}),$$

with $\Omega = [0, 2\pi] \times [0, 2\pi]$ and periodic boundary conditions on $\partial\Omega$.

2D Turbulence: Mathematical Formulation

- Consider the Navier–Stokes equations for 2D incompressible homogeneous isotropic turbulence with density $\rho = 1$:

$$\frac{\partial \boldsymbol{u}}{\partial t} - \nu \nabla^2 \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} P = \boldsymbol{F},$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0,$$

$$\int_\Omega \boldsymbol{u}\, d\boldsymbol{x} = \boldsymbol{0}, \qquad \int_\Omega \boldsymbol{F}\, d\boldsymbol{x} = \boldsymbol{0},$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}),$$

with $\Omega = [0, 2\pi] \times [0, 2\pi]$ and periodic boundary conditions on $\partial\Omega$.

- Introduce the Hilbert space

$$H(\Omega) \doteq \mathrm{cl} \left\{ \boldsymbol{u} \in (C^2(\Omega) \cap L^2(\Omega))^2 \mid \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0,\ \int_\Omega \boldsymbol{u}\, d\boldsymbol{x} = \boldsymbol{0} \right\}.$$

with inner product $(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{u}(\boldsymbol{x}, t) \cdot \boldsymbol{v}(\boldsymbol{x}, t)\, d\boldsymbol{x}$ and L^2 norm $|\boldsymbol{u}| = (\boldsymbol{u}, \boldsymbol{u})^{1/2}$.

- For $\boldsymbol{u} \in H(\Omega)$, the Navier–Stokes equations can be expressed:

$$\frac{d\boldsymbol{u}}{dt} - \nu \nabla^2 \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} P = \boldsymbol{F}.$$

- For $\boldsymbol{u} \in H(\Omega)$, the Navier–Stokes equations can be expressed:

$$\frac{d\boldsymbol{u}}{dt} - \nu\nabla^2\boldsymbol{u} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}P = \boldsymbol{F}.$$

- Introduce $A \doteq -\mathcal{P}(\nabla^2)$, $\boldsymbol{f} \doteq \mathcal{P}(\boldsymbol{F})$, and the bilinear map

$$\mathcal{B}(\boldsymbol{u}, \boldsymbol{u}) \doteq \mathcal{P}\left(\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}P\right),$$

where $\mathcal{P}$ is the Helmholtz–Leray projection operator from $(L^2(\Omega))^2$ to $H(\Omega)$:

$$\mathcal{P}(\boldsymbol{v}) \doteq \boldsymbol{v} - \boldsymbol{\nabla}\nabla^{-2}\boldsymbol{\nabla}\cdot\boldsymbol{v}, \qquad \forall\boldsymbol{v} \in (L^2(\Omega))^2.$$

- For $\boldsymbol{u} \in H(\Omega)$, the Navier–Stokes equations can be expressed:

$$\frac{d\boldsymbol{u}}{dt} - \nu \nabla^2 \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} P = \boldsymbol{F}.$$

- Introduce $A \doteq -\mathcal{P}(\nabla^2)$, $\boldsymbol{f} \doteq \mathcal{P}(\boldsymbol{F})$, and the bilinear map

$$\mathcal{B}(\boldsymbol{u}, \boldsymbol{u}) \doteq \mathcal{P}\left(\boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} P\right),$$

where $\mathcal{P}$ is the Helmholtz–Leray projection operator from $(L^2(\Omega))^2$ to $H(\Omega)$:

$$\mathcal{P}(\boldsymbol{v}) \doteq \boldsymbol{v} - \boldsymbol{\nabla} \nabla^{-2} \boldsymbol{\nabla} \cdot \boldsymbol{v}, \qquad \forall \boldsymbol{v} \in (L^2(\Omega))^2.$$

- The dynamical system can then be compactly written:

$$\frac{d\boldsymbol{u}}{dt} + \nu A \boldsymbol{u} + \mathcal{B}(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}.$$

Stokes Operator A

- The operator $A = \mathcal{P}(-\nabla^2)$ is positive semi-definite and self-adjoint, with a compact inverse.

Stokes Operator A

- The operator $A = \mathcal{P}(-\nabla^2)$ is positive semi-definite and self-adjoint, with a compact inverse.

- On the periodic domain $\Omega = [0, 2\pi] \times [0, 2\pi]$, the eigenvalues of A are

$$\lambda = \boldsymbol{k}\cdot\boldsymbol{k}, \qquad \boldsymbol{k} \in \mathbb{Z} \times \mathbb{Z} \backslash \{\mathbf{0}\}.$$

Stokes Operator A

- The operator $A = \mathcal{P}(-\nabla^2)$ is positive semi-definite and self-adjoint, with a compact inverse.

- On the periodic domain $\Omega = [0, 2\pi] \times [0, 2\pi]$, the eigenvalues of A are

$$\lambda = \boldsymbol{k}\cdot\boldsymbol{k}, \qquad \boldsymbol{k} \in \mathbb{Z} \times \mathbb{Z} \backslash \{\mathbf{0}\}.$$

- The eigenvalues of A can be arranged as

$$0 < \lambda_0 < \lambda_1 < \lambda_2 < \cdots, \qquad \lambda_0 = 1$$

and its eigenvectors $\boldsymbol{w}_i,\ i \in \mathbb{N}_0$, form an orthonormal basis for the Hilbert space H, upon which we can define any quotient power of A:

$$A^\alpha \boldsymbol{w}_j = \lambda_j^\alpha \boldsymbol{w}_j, \qquad \alpha \in \mathbb{R}, \quad j \in \mathbb{N}_0.$$

Subspace of Finite Enstrophy

- We define the subspace of H consisting of solutions with finite enstrophy:

$$V \doteq \left\{ \boldsymbol{u} \in H \mid \sum_{j=0}^{\infty} \lambda_j (\boldsymbol{u}, \boldsymbol{w}_j)^2 < \infty \right\}.$$

Subspace of Finite Enstrophy

- We define the subspace of H consisting of solutions with finite enstrophy:

$$V \doteq \left\{ \boldsymbol{u} \in H \mid \sum_{j=0}^{\infty} \lambda_j (\boldsymbol{u}, \boldsymbol{w}_j)^2 < \infty \right\}.$$

- Another suitable norm for elements $\boldsymbol{u} \in V$ is

$$||\boldsymbol{u}|| = \left| A^{1/2} \boldsymbol{u} \right| = \left(\int_\Omega \sum_{i=1}^{2} \frac{\partial \boldsymbol{u}}{\partial x_i} \cdot \frac{\partial \boldsymbol{u}}{\partial x_i} \right)^{1/2} = \left(\sum_{j=0}^{\infty} \lambda_j (\boldsymbol{u}, \boldsymbol{w}_j)^2 \right)^{1/2}.$$

Properties of the Bilinear Map

- We will make use of the antisymmetry

$$(\mathcal{B}(\boldsymbol{u}, \boldsymbol{v}), \boldsymbol{w}) = -(\mathcal{B}(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{v}).$$

Properties of the Bilinear Map

- We will make use of the antisymmetry

$$(\mathcal{B}(\boldsymbol{u}, \boldsymbol{v}), \boldsymbol{w}) = -(\mathcal{B}(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{v}).$$

- In 2D, we also have orthogonality:

$$(\mathcal{B}(\boldsymbol{u}, \boldsymbol{u}), A\boldsymbol{u}) = 0$$

and the strong form of enstrophy invariance:

$$(\mathcal{B}(A\boldsymbol{v}, \boldsymbol{v}), \boldsymbol{u}) = (\mathcal{B}(\boldsymbol{u}, \boldsymbol{v}), A\boldsymbol{v}).$$

Properties of the Bilinear Map

- We will make use of the antisymmetry

$$(\mathcal{B}(\boldsymbol{u},\boldsymbol{v}),\boldsymbol{w}) = -(\mathcal{B}(\boldsymbol{u},\boldsymbol{w}),\boldsymbol{v}).$$

- In 2D, we also have orthogonality:

$$(\mathcal{B}(\boldsymbol{u},\boldsymbol{u}),A\boldsymbol{u}) = 0$$

and the strong form of enstrophy invariance:

$$(\mathcal{B}(A\boldsymbol{v},\boldsymbol{v}),\boldsymbol{u}) = (\mathcal{B}(\boldsymbol{u},\boldsymbol{v}),A\boldsymbol{v}).$$

- In 2D the above properties imply the symmetry

$$(\mathcal{B}(A\boldsymbol{u},\boldsymbol{u}),\boldsymbol{u}) + (\mathcal{B}(\boldsymbol{v},A\boldsymbol{v}),\boldsymbol{u}) + (\mathcal{B}(\boldsymbol{v},\boldsymbol{v}),A\boldsymbol{v}) = 0.$$

Dynamical Behaviour

- Our starting point is the incompressible 2D Navier–Stokes equation with periodic boundary conditions:

$$\frac{d\boldsymbol{u}}{dt} + \nu A\boldsymbol{u} + \mathcal{B}(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}, \qquad \boldsymbol{u} \in H.$$

Dynamical Behaviour

- Our starting point is the incompressible 2D Navier–Stokes equation with periodic boundary conditions:

$$\frac{d\boldsymbol{u}}{dt} + \nu A\boldsymbol{u} + \mathcal{B}(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}, \qquad \boldsymbol{u} \in H.$$

- Take the inner product with $\boldsymbol{u}$ (respectively $A\boldsymbol{u}$):

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}(t)|^2 + \nu||\boldsymbol{u}(t)||^2 = (\boldsymbol{f}, \boldsymbol{u}(t)),$$
$$\frac{1}{2}\frac{d}{dt}||\boldsymbol{u}(t)||^2 + \nu|A\boldsymbol{u}(t)|^2 = (\boldsymbol{f}, A\boldsymbol{u}(t)).$$

Dynamical Behaviour

- Our starting point is the incompressible 2D Navier–Stokes equation with periodic boundary conditions:

$$\frac{d\boldsymbol{u}}{dt} + \nu A\boldsymbol{u} + \mathcal{B}(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}, \qquad \boldsymbol{u} \in H.$$

- Take the inner product with $\boldsymbol{u}$ (respectively $A\boldsymbol{u}$):

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}(t)|^2 + \nu||\boldsymbol{u}(t)||^2 = (\boldsymbol{f}, \boldsymbol{u}(t)),$$
$$\frac{1}{2}\frac{d}{dt}||\boldsymbol{u}(t)||^2 + \nu|A\boldsymbol{u}(t)|^2 = (\boldsymbol{f}, A\boldsymbol{u}(t)).$$

- The Cauchy–Schwarz and Poincaré inequalities yield

$$(\boldsymbol{f}, \boldsymbol{u}(t)) \le |\boldsymbol{f}||\boldsymbol{u}(t)| \qquad \text{and} \qquad |\boldsymbol{u}(t)| \le ||\boldsymbol{u}(t)||.$$

Dynamical Behaviour

- Our starting point is the incompressible 2D Navier–Stokes equation with periodic boundary conditions:

$$\frac{d\boldsymbol{u}}{dt} + \nu A\boldsymbol{u} + \mathcal{B}(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}, \qquad \boldsymbol{u} \in H.$$

- Take the inner product with $\boldsymbol{u}$ (respectively $A\boldsymbol{u}$):

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}(t)|^2 + \nu||\boldsymbol{u}(t)||^2 = (\boldsymbol{f}, \boldsymbol{u}(t)),$$
$$\frac{1}{2}\frac{d}{dt}||\boldsymbol{u}(t)||^2 + \nu|A\boldsymbol{u}(t)|^2 = (\boldsymbol{f}, A\boldsymbol{u}(t)).$$

- The Cauchy–Schwarz and Poincaré inequalities yield

$$(\boldsymbol{f}, \boldsymbol{u}(t)) \leq |\boldsymbol{f}||\boldsymbol{u}(t)| \qquad \text{and} \qquad |\boldsymbol{u}(t)| \leq ||\boldsymbol{u}(t)||.$$

- Since the existence and uniqueness for solutions to the 2D Navier–Stokes equation has been proven, a global attractor can be defined [Ladyzhenskaya 1975], [Foias & Temam 1979].

Dynamical Behaviour: Constant Forcing

- If the force $\boldsymbol{f}$ is constant with respect to time, a Gronwall inequality can be exploited:

$$|\boldsymbol{u}(t)|^2 \leq e^{-\nu t}|\boldsymbol{u}(0)|^2 + (1 - e^{-\nu t})\left(\frac{|\boldsymbol{f}|}{\nu}\right)^2.$$

Dynamical Behaviour: Constant Forcing

- If the force $\boldsymbol{f}$ is constant with respect to time, a Gronwall inequality can be exploited:

$$|\boldsymbol{u}(t)|^2 \leq e^{-\nu t}|\boldsymbol{u}(0)|^2 + (1 - e^{-\nu t})\left(\frac{|\boldsymbol{f}|}{\nu}\right)^2.$$

- Defining a nondimensional Grashof number $G = \dfrac{|\boldsymbol{f}|}{\nu^2}$, the above inequality can be simplified to

$$|\boldsymbol{u}(t)|^2 \leq e^{-\nu t}|\boldsymbol{u}(0)|^2 + (1 - e^{-\nu t})\nu^2 G^2.$$

Dynamical Behaviour: Constant Forcing

- If the force $\boldsymbol{f}$ is constant with respect to time, a Gronwall inequality can be exploited:

$$|\boldsymbol{u}(t)|^2 \leq e^{-\nu t}|\boldsymbol{u}(0)|^2 + (1 - e^{-\nu t})\left(\frac{|\boldsymbol{f}|}{\nu}\right)^2.$$

- Defining a nondimensional Grashof number $G = \dfrac{|\boldsymbol{f}|}{\nu^2}$, the above inequality can be simplified to

$$|\boldsymbol{u}(t)|^2 \leq e^{-\nu t}|\boldsymbol{u}(0)|^2 + (1 - e^{-\nu t})\nu^2 G^2.$$

- Similarly,

$$||\boldsymbol{u}(t)||^2 \leq e^{-\nu t}||\boldsymbol{u}(0)||^2 + (1 - e^{-\nu t})\nu^2 G^2.$$

Dynamical Behaviour: Constant Forcing

- If the force $\boldsymbol{f}$ is constant with respect to time, a Gronwall inequality can be exploited:

$$|\boldsymbol{u}(t)|^2 \leq e^{-\nu t}|\boldsymbol{u}(0)|^2 + (1 - e^{-\nu t})\left(\frac{|\boldsymbol{f}|}{\nu}\right)^2.$$

- Defining a nondimensional Grashof number $G = \dfrac{|\boldsymbol{f}|}{\nu^2}$, the above inequality can be simplified to

$$|\boldsymbol{u}(t)|^2 \leq e^{-\nu t}|\boldsymbol{u}(0)|^2 + (1 - e^{-\nu t})\nu^2 G^2.$$

- Similarly,

$$||\boldsymbol{u}(t)||^2 \leq e^{-\nu t}||\boldsymbol{u}(0)||^2 + (1 - e^{-\nu t})\nu^2 G^2.$$

- Being on the attractor thus requires

$$|\boldsymbol{u}| \leq \nu G \qquad \text{and} \qquad ||\boldsymbol{u}|| \leq \nu G.$$

Attractor Set $\mathcal{A}$

- Let S be the solution operator:

$$S(t)\boldsymbol{u}_0 = \boldsymbol{u}(t), \qquad \boldsymbol{u}_0 = \boldsymbol{u}(0),$$

where $\boldsymbol{u}(t)$ is the unique solution of the Navier–Stokes equations.

Attractor Set $\mathcal{A}$

- Let S be the solution operator:

$$S(t)\boldsymbol{u}_0 = \boldsymbol{u}(t), \qquad \boldsymbol{u}_0 = \boldsymbol{u}(0),$$

  where $\boldsymbol{u}(t)$ is the unique solution of the Navier–Stokes equations.

- The closed ball $\mathfrak{B}$ of radius νG in the space V is a bounded absorbing set in H.

Attractor Set $\mathcal{A}$

- Let S be the solution operator:

$$S(t)\boldsymbol{u}_0 = \boldsymbol{u}(t), \qquad \boldsymbol{u}_0 = \boldsymbol{u}(0),$$

  where $\boldsymbol{u}(t)$ is the unique solution of the Navier–Stokes equations.

- The closed ball $\mathfrak{B}$ of radius νG in the space V is a bounded absorbing set in H.

- That is, for any bounded set $\mathfrak{B}'$ there exists a time t_0 such that

$$t_0 = t_0(\mathfrak{B}'), \quad \text{and} \quad S(t)\mathfrak{B}' \subset \mathfrak{B}, \quad \forall t \geq t_0.$$

Attractor Set $\mathcal{A}$

- Let S be the solution operator:

$$S(t)\boldsymbol{u}_0 = \boldsymbol{u}(t), \qquad \boldsymbol{u}_0 = \boldsymbol{u}(0),$$

where $\boldsymbol{u}(t)$ is the unique solution of the Navier–Stokes equations.

- The closed ball $\mathfrak{B}$ of radius νG in the space V is a bounded absorbing set in H.

- That is, for any bounded set $\mathfrak{B}'$ there exists a time t_0 such that

$$t_0 = t_0(\mathfrak{B}'), \quad \text{and} \quad S(t)\mathfrak{B}' \subset \mathfrak{B}, \quad \forall t \geq t_0.$$

- We can then construct the global attractor:

$$\mathcal{A} = \bigcap_{t \geq 0} S(t)\mathfrak{B},$$

so $\mathcal{A}$ is the largest bounded, invariant set such that $S(t)\mathcal{A} = \mathcal{A}$ for all $t \geq 0$.

Z–E Plane Bounds: Constant Forcing

- A trivial lower bound is provided by the Poincaré inequality:

$$|\boldsymbol{u}|^2 \leq ||\boldsymbol{u}||^2 \quad \Rightarrow \quad E \leq Z.$$

Z–E Plane Bounds: Constant Forcing

- A trivial lower bound is provided by the Poincaré inequality:

$$|\boldsymbol{u}|^2 \le ||\boldsymbol{u}||^2 \quad \Rightarrow \quad E \le Z.$$

- An upper bound is given by

Theorem 1 (Dascaliuc, Foias, and Jolly [2005])
For all $\boldsymbol{u} \in \mathcal{A}$,

$$||\boldsymbol{u}||^2 \le \frac{|\boldsymbol{f}|}{\nu}|\boldsymbol{u}|.$$

Z–E Plane Bounds: Constant Forcing

- A trivial lower bound is provided by the Poincaré inequality:

$$|\boldsymbol{u}|^2 \leq ||\boldsymbol{u}||^2 \quad \Rightarrow \quad E \leq Z.$$

- An upper bound is given by

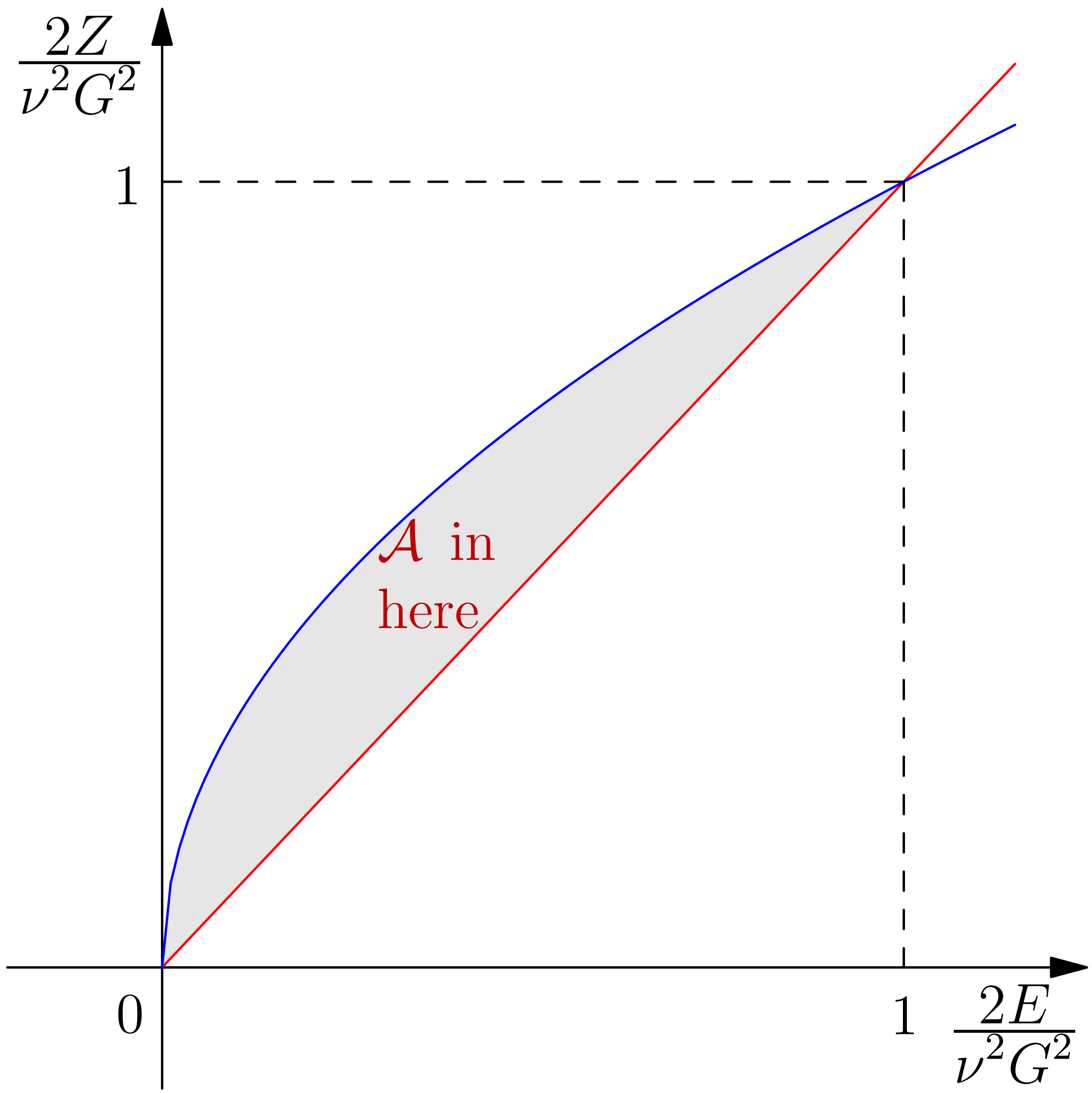
Theorem 2 (Dascaliuc, Foias, and Jolly [2005])
For all $\boldsymbol{u} \in \mathcal{A}$,

$$||\boldsymbol{u}||^2 \leq \frac{|\boldsymbol{f}|}{\nu}|\boldsymbol{u}|.$$

- That is,

$$Z \leq \nu G \sqrt{E}.$$

Z–E Plane Bounds: Constant Forcing



Extended Norm: Random Forcing

- For a random variable α, with probability density function P, define the ensemble average

$$\langle \alpha \rangle = \int_{-\infty}^{\infty} \alpha \left(\frac{dP}{d\zeta} \right) d\zeta.$$

Extended Norm: Random Forcing

- For a random variable α, with probability density function P, define the ensemble average

$$\langle \alpha \rangle = \int_{-\infty}^{\infty} \alpha \left(\frac{dP}{d\zeta} \right) d\zeta.$$

- The extended inner product is

$$(\boldsymbol{u}, \boldsymbol{v})_{\tilde{\omega}} \doteq \int_{\Omega} \langle \boldsymbol{u} \cdot \boldsymbol{v} \rangle \, d\boldsymbol{x} = \int_{\Omega} \left(\int_{-\infty}^{\infty} \boldsymbol{u} \cdot \boldsymbol{v} \, \frac{dP}{d\zeta} d\zeta \right) d\boldsymbol{x},$$

with norm

$$|\boldsymbol{f}|_{\tilde{\omega}} \doteq \left(\int_{\Omega} \langle |\boldsymbol{f}|^2 \rangle \, d\boldsymbol{x} \right)^{1/2}.$$

Dynamical Behaviour: Random Forcing

- Energy balance:

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 + \nu(A\boldsymbol{u}, \boldsymbol{u}) + (\mathcal{B}(\boldsymbol{u}, \boldsymbol{u}), \boldsymbol{u}) = (\boldsymbol{f}, \boldsymbol{u}) \doteq \epsilon,$$

where ϵ is the rate of energy injection.

Dynamical Behaviour: Random Forcing

- Energy balance:

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 + \nu(A\boldsymbol{u}, \boldsymbol{u}) + (\mathcal{B}(\boldsymbol{u}, \boldsymbol{u}), \boldsymbol{u}) = (\boldsymbol{f}, \boldsymbol{u}) \doteq \epsilon,$$

where ϵ is the rate of energy injection.

- From the energy conservation identity $(\mathcal{B}(\boldsymbol{u}, \boldsymbol{u}), \boldsymbol{u}) = 0$,

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 + \nu||\boldsymbol{u}||^2 = \epsilon.$$

Dynamical Behaviour: Random Forcing

- Energy balance:

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 + \nu(A\boldsymbol{u},\boldsymbol{u}) + (\mathcal{B}(\boldsymbol{u},\boldsymbol{u}),\boldsymbol{u}) = (\boldsymbol{f},\boldsymbol{u}) \doteq \epsilon,$$

where ϵ is the rate of energy injection.

- From the energy conservation identity $(\mathcal{B}(\boldsymbol{u},\boldsymbol{u}),\boldsymbol{u}) = 0$,

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 + \nu||\boldsymbol{u}||^2 = \epsilon.$$

- The Poincaré inequality $||\boldsymbol{u}|| \geq |\boldsymbol{u}|$ leads to

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 \leq \epsilon - \nu|\boldsymbol{u}|^2,$$

which implies that $|\boldsymbol{u}(t)|^2 \leq e^{-2\nu t}|\boldsymbol{u}(0)|^2 + \left(\frac{1-e^{-2\nu t}}{\nu}\right)\epsilon.$

Dynamical Behaviour: Random Forcing

- Energy balance:

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 + \nu(A\boldsymbol{u},\boldsymbol{u}) + (\mathcal{B}(\boldsymbol{u},\boldsymbol{u}),\boldsymbol{u}) = (\boldsymbol{f},\boldsymbol{u}) \doteq \epsilon,$$

where ϵ is the rate of energy injection.

- From the energy conservation identity $(\mathcal{B}(\boldsymbol{u},\boldsymbol{u}),\boldsymbol{u}) = 0$,

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 + \nu||\boldsymbol{u}||^2 = \epsilon.$$

- The Poincaré inequality $||\boldsymbol{u}|| \geq |\boldsymbol{u}|$ leads to

$$\frac{1}{2}\frac{d}{dt}|\boldsymbol{u}|^2 \leq \epsilon - \nu|\boldsymbol{u}|^2,$$

which implies that $|\boldsymbol{u}(t)|^2 \leq e^{-2\nu t}|\boldsymbol{u}(0)|^2 + \left(\frac{1-e^{-2\nu t}}{\nu}\right)\epsilon.$

- So for every $\boldsymbol{u} \in \mathcal{A}$, we expect $|\boldsymbol{u}(t)|^2 \leq \epsilon/\nu$.

- From $|\boldsymbol{u}(t)| \leq \sqrt{\epsilon/\nu}$ we then obtain a lower bound for $|\boldsymbol{f}|$:

$$\sqrt{\nu\epsilon} \leq \frac{\epsilon}{|\boldsymbol{u}|} = \frac{(\boldsymbol{f}, \boldsymbol{u})}{|\boldsymbol{u}|} \leq \frac{|\boldsymbol{f}||\boldsymbol{u}|}{|\boldsymbol{u}|} = |\boldsymbol{f}|.$$

- From $|\boldsymbol{u}(t)| \leq \sqrt{\epsilon/\nu}$ we then obtain a lower bound for $|\boldsymbol{f}|$:

$$\sqrt{\nu\epsilon} \leq \frac{\epsilon}{|\boldsymbol{u}|} = \frac{(\boldsymbol{f}, \boldsymbol{u})}{|\boldsymbol{u}|} \leq \frac{|\boldsymbol{f}||\boldsymbol{u}|}{|\boldsymbol{u}|} = |\boldsymbol{f}|.$$

- It is convenient to use this lower bound for $|\boldsymbol{f}|$ to define a lower bound for the Grashof number $G = |\boldsymbol{f}|/\nu^2$, which we use as the normalization $\tilde{G}$ for random forcing:

$$\tilde{G} = \sqrt{\frac{\epsilon}{\nu^3}}.$$

• From $|\boldsymbol{u}(t)| \leq \sqrt{\epsilon/\nu}$ we then obtain a lower bound for $|\boldsymbol{f}|$:

$$\sqrt{\nu\epsilon} \leq \frac{\epsilon}{|\boldsymbol{u}|} = \frac{(\boldsymbol{f}, \boldsymbol{u})}{|\boldsymbol{u}|} \leq \frac{|\boldsymbol{f}||\boldsymbol{u}|}{|\boldsymbol{u}|} = |\boldsymbol{f}|.$$

• It is convenient to use this lower bound for $|\boldsymbol{f}|$ to define a lower bound for the Grashof number $G = |\boldsymbol{f}|/\nu^2$, which we use as the normalization $\tilde{G}$ for random forcing:

$$\tilde{G} = \sqrt{\frac{\epsilon}{\nu^3}}.$$

• We recently proved the following theorem (submitted to JDE):

Theorem 3 (Emami & Bowman [2017]) *For all* $\boldsymbol{u} \in \mathcal{A}$ *with energy injection rate* ϵ,

$$||\boldsymbol{u}||^2 \leq \sqrt{\frac{\epsilon}{\nu}}\, |\boldsymbol{u}|.$$

- From $|\boldsymbol{u}(t)| \le \sqrt{\epsilon/\nu}$ we then obtain a lower bound for $|\boldsymbol{f}|$:

$$\sqrt{\nu\epsilon} \le \frac{\epsilon}{|\boldsymbol{u}|} = \frac{(\boldsymbol{f}, \boldsymbol{u})}{|\boldsymbol{u}|} \le \frac{|\boldsymbol{f}||\boldsymbol{u}|}{|\boldsymbol{u}|} = |\boldsymbol{f}|.$$

- It is convenient to use this lower bound for $|\boldsymbol{f}|$ to define a lower bound for the Grashof number $G = |\boldsymbol{f}|/\nu^2$, which we use as the normalization $\tilde{G}$ for random forcing:

$$\tilde{G} = \sqrt{\frac{\epsilon}{\nu^3}}.$$

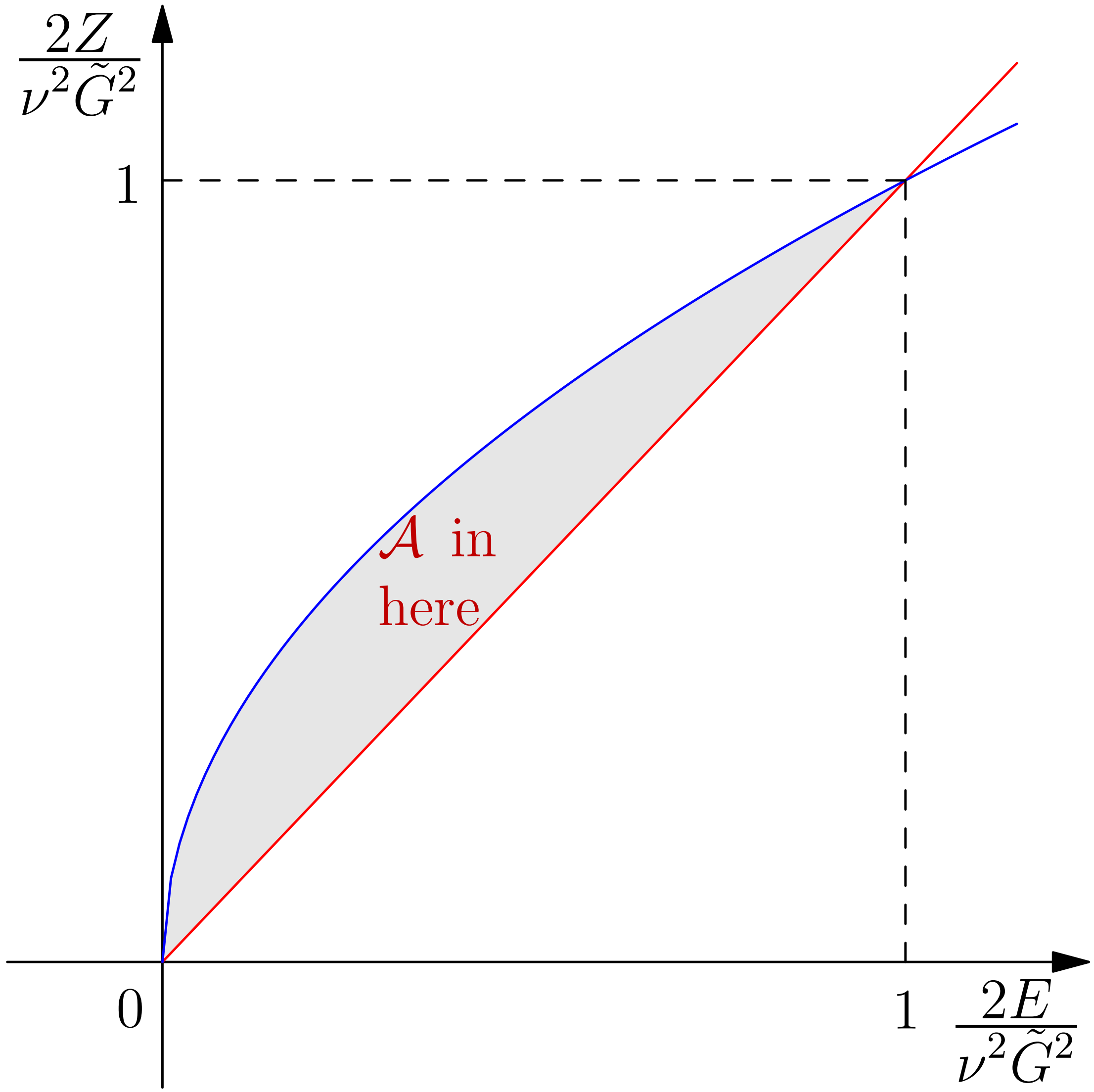
- We recently proved the following theorem (submitted to JDE):

Theorem 4 (Emami & Bowman [2017]) *For all* $\boldsymbol{u} \in \mathcal{A}$ *with energy injection rate* ϵ,

$$||\boldsymbol{u}||^2 \le \sqrt{\frac{\epsilon}{\nu}}\,|\boldsymbol{u}|.$$

- This leads to the same form as for a constant force: $Z \le \nu\tilde{G}\sqrt{E}$.

Z–E Plane Bounds: Random Forcing



DNS code

- We have released a highly optimized 2D pseudospectral code in C++: `https://github.com/dealias/dns`.

DNS code

- We have released a highly optimized 2D pseudospectral code in C++: `https://github.com/dealias/dns`.

- It uses our **FFTW++** library to implicitly dealias the advective convolution, while exploiting Hermitian symmetry [Bowman & Roberts 2011], [Roberts & Bowman 2017].

DNS code

- We have released a highly optimized 2D pseudospectral code in C++: `https://github.com/dealias/dns`.
- It uses our `FFTW++` library to implicitly dealias the advective convolution, while exploiting Hermitian symmetry [Bowman & Roberts 2011], [Roberts & Bowman 2017].
- Advanced computer memory management, such as implicit padding, memory alignment, and dynamic moment averaging allow `DNS` to attain its extreme performance.

DNS code

- We have released a highly optimized 2D pseudospectral code in C++: `https://github.com/dealias/dns`.
- It uses our **FFTW++** library to implicitly dealias the advective convolution, while exploiting Hermitian symmetry [Bowman & Roberts 2011], [Roberts & Bowman 2017].
- Advanced computer memory management, such as implicit padding, memory alignment, and dynamic moment averaging allow **DNS** to attain its extreme performance.
- It uses the formulation proposed by Basdevant [1983] to reduce the number of FFTs required for 2D (3D) incompressible turbulence to 4 (8).

DNS code

- We have released a highly optimized 2D pseudospectral code in C++: `https://github.com/dealias/dns`.
- It uses our **FFTW++** library to implicitly dealias the advective convolution, while exploiting Hermitian symmetry [Bowman & Roberts 2011], [Roberts & Bowman 2017].
- Advanced computer memory management, such as implicit padding, memory alignment, and dynamic moment averaging allow **DNS** to attain its extreme performance.
- It uses the formulation proposed by Basdevant [1983] to reduce the number of FFTs required for 2D (3D) incompressible turbulence to 4 (8).
- We also include simplified 2D (146 lines) and 3D (287 lines) versions called **ProtoDNS** for educational purposes: `https://github.com/dealias/dns/tree/master/protodns`.

Dynamic Moment Averaging

- Advantageous to precompute time-integrated moments like

$$M_n(t) = \int_0^t |\omega_{\boldsymbol{k}}(\tau)|^n \, d\tau.$$

Dynamic Moment Averaging

- Advantageous to precompute time-integrated moments like

$$M_n(t) = \int_0^t |\omega_{\boldsymbol{k}}(\tau)|^n \, d\tau.$$

- This can be accomplished done by evolving

$$\frac{dM_n}{dt} = |\omega_{\boldsymbol{k}}|^n,$$

  along with the vorticity $\omega_{\boldsymbol{k}}$ itself, using *the same* temporal discretization.

Dynamic Moment Averaging

- Advantageous to precompute time-integrated moments like

$$M_n(t) = \int_0^t \left|\omega_{\boldsymbol{k}}(\tau)\right|^n \, d\tau.$$

- This can be accomplished done by evolving

$$\frac{dM_n}{dt} = \left|\omega_{\boldsymbol{k}}\right|^n,$$

  along with the vorticity $\omega_{\boldsymbol{k}}$ itself, using *the same* temporal discretization.

- These evolved quantities M_n can be used to extract accurate statistical averages during the post-processing phase, once the saturation time t_1 has been determined by the user:

$$\int_{t_1}^{t_2} \left|\omega_{\boldsymbol{k}}\right|^n (\tau) \, d\tau = M_n(t_2) - M_n(t_1).$$

Enstrophy Balance

$$\frac{\partial \omega_{\boldsymbol{k}}}{\partial t} + \nu k^2 \omega_{\boldsymbol{k}} = S_{\boldsymbol{k}} + f_{\boldsymbol{k}},$$

- Multiply by $\omega_{\boldsymbol{k}}^*$ and integrate over wavenumber angle $\Rightarrow$ enstrophy spectrum $Z(k)$ evolves as:

$$\frac{\partial}{\partial t} Z(k) + 2\nu k^2 Z(k) = 2T(k) + G(k),$$

where $T(k)$ and $G(k)$ are the corresponding angular averages of $\operatorname{Re}\langle S_{\boldsymbol{k}} \omega_{\boldsymbol{k}}^* \rangle$ and $\operatorname{Re}\langle f_{\boldsymbol{k}} \omega_{\boldsymbol{k}}^* \rangle$.

Nonlinear Enstrophy Transfer Function

$$\frac{\partial}{\partial t} Z(k) + 2\nu k^2 Z(k) = 2T(k) + G(k).$$

- Let

$$\Pi(k) \doteq 2 \int_k^\infty T(p)\, dp$$

represent the nonlinear transfer of enstrophy into $[k, \infty)$.

Nonlinear Enstrophy Transfer Function

$$\frac{\partial}{\partial t} Z(k) + 2\nu k^2 Z(k) = 2T(k) + G(k).$$

- Let

$$\Pi(k) \doteq 2 \int_k^\infty T(p)\, dp$$

represent the nonlinear transfer of enstrophy into $[k, \infty)$.

- Integrate from k to ∞:

$$\frac{d}{dt} \int_k^\infty Z(p)\, dp = \Pi(k) - \epsilon_Z(k),$$

where $\epsilon_Z(k) \doteq 2\nu \int_k^\infty p^2 Z(p)\, dp - \int_k^\infty G(p)\, dp$ is the total enstrophy transfer, via dissipation and forcing, out of wavenumbers higher than k.

- A positive (negative) value for $\Pi(k)$ represents a flow of enstrophy to wavenumbers higher (lower) than k.

- A positive (negative) value for $\Pi(k)$ represents a flow of enstrophy to wavenumbers higher (lower) than k.

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

$$0 = \frac{d}{dt} \int_0^\infty Z(p)\, dp = 2 \int_0^\infty T(p)\, dp,$$

so that

$$\Pi(k) = 2 \int_k^\infty T(p)\, dp = -2 \int_0^k T(p)\, dp.$$

- A positive (negative) value for $\Pi(k)$ represents a flow of enstrophy to wavenumbers higher (lower) than k.

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

$$0 = \frac{d}{dt}\int_0^\infty Z(p)\,dp = 2\int_0^\infty T(p)\,dp,$$

so that

$$\Pi(k) = 2\int_k^\infty T(p)\,dp = -2\int_0^k T(p)\,dp.$$

- Note that $\Pi(0) = \Pi(\infty) = 0$.

- A positive (negative) value for $\Pi(k)$ represents a flow of enstrophy to wavenumbers higher (lower) than k.

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

$$0 = \frac{d}{dt}\int_0^\infty Z(p)\,dp = 2\int_0^\infty T(p)\,dp,$$

so that

$$\Pi(k) = 2\int_k^\infty T(p)\,dp = -2\int_0^k T(p)\,dp.$$

- Note that $\Pi(0) = \Pi(\infty) = 0$.

- In a steady state, $\Pi(k) = \epsilon_Z(k)$.

- A positive (negative) value for $\Pi(k)$ represents a flow of enstrophy to wavenumbers higher (lower) than k.

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

$$0 = \frac{d}{dt}\int_0^\infty Z(p)\,dp = 2\int_0^\infty T(p)\,dp,$$

  so that

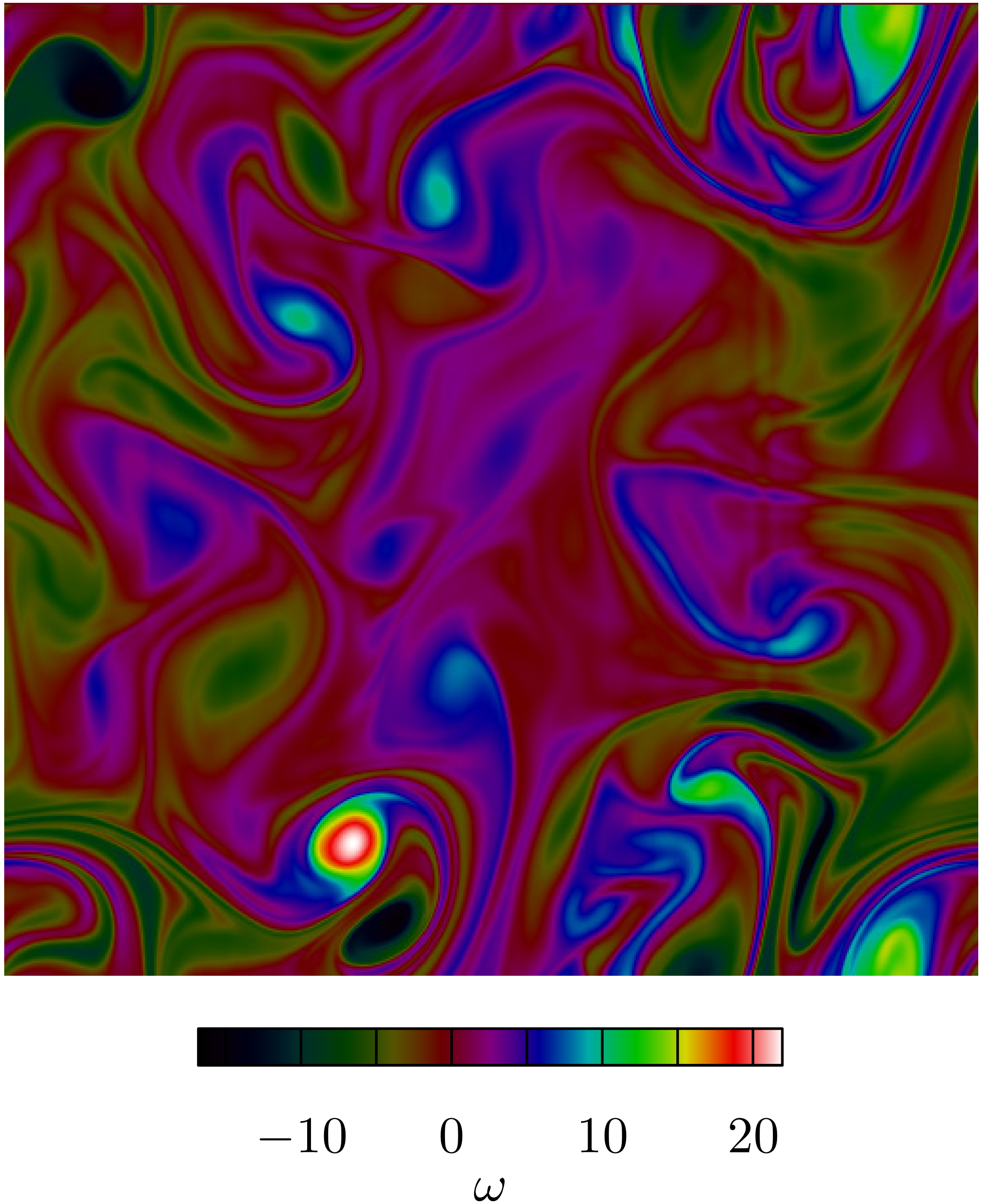
$$\Pi(k) = 2\int_k^\infty T(p)\,dp = -2\int_0^k T(p)\,dp.$$

- Note that $\Pi(0) = \Pi(\infty) = 0$.

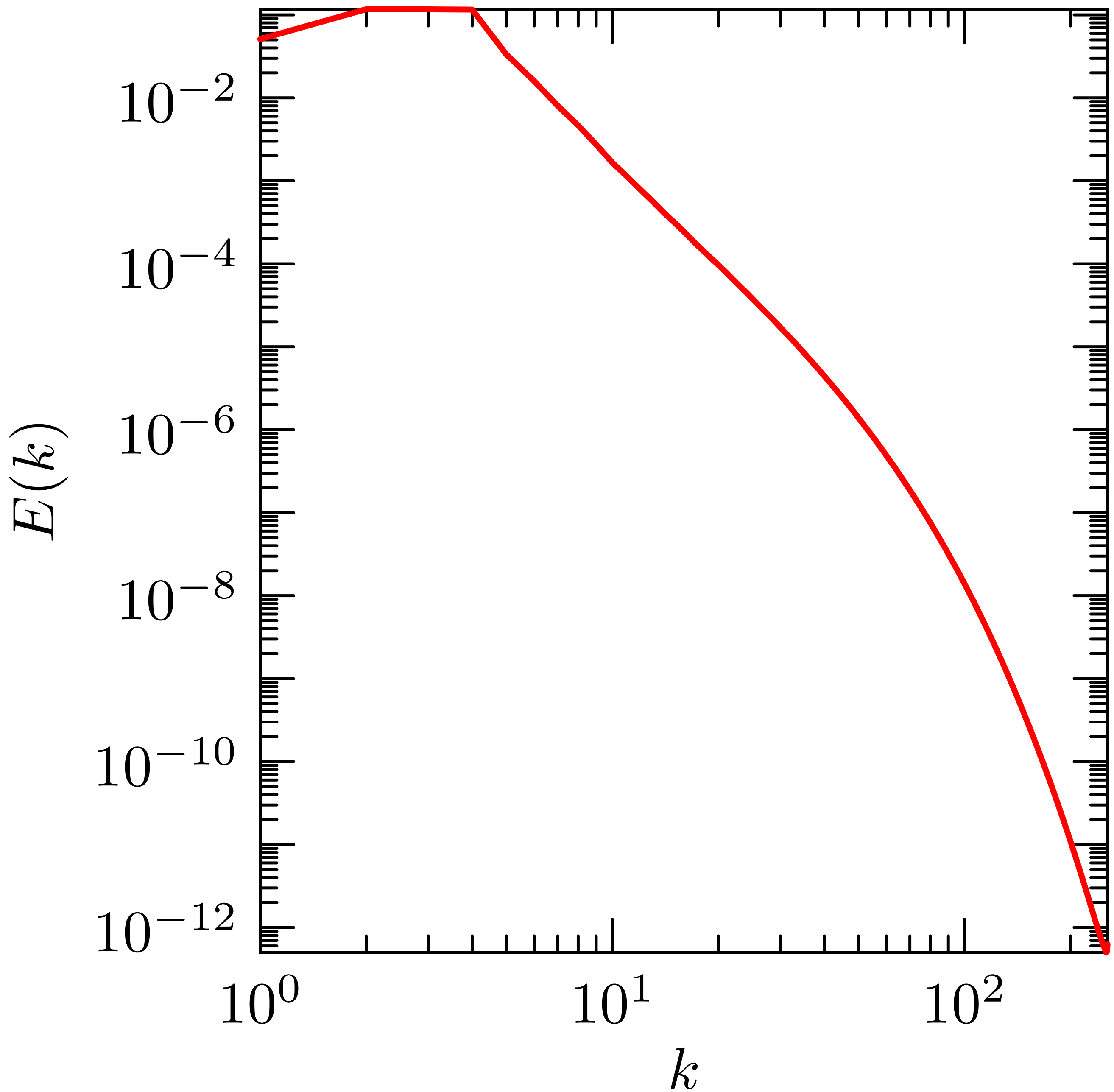
- In a steady state, $\Pi(k) = \epsilon_Z(k)$.

- This provides an excellent numerical diagnostic for determining the saturation time t_1.

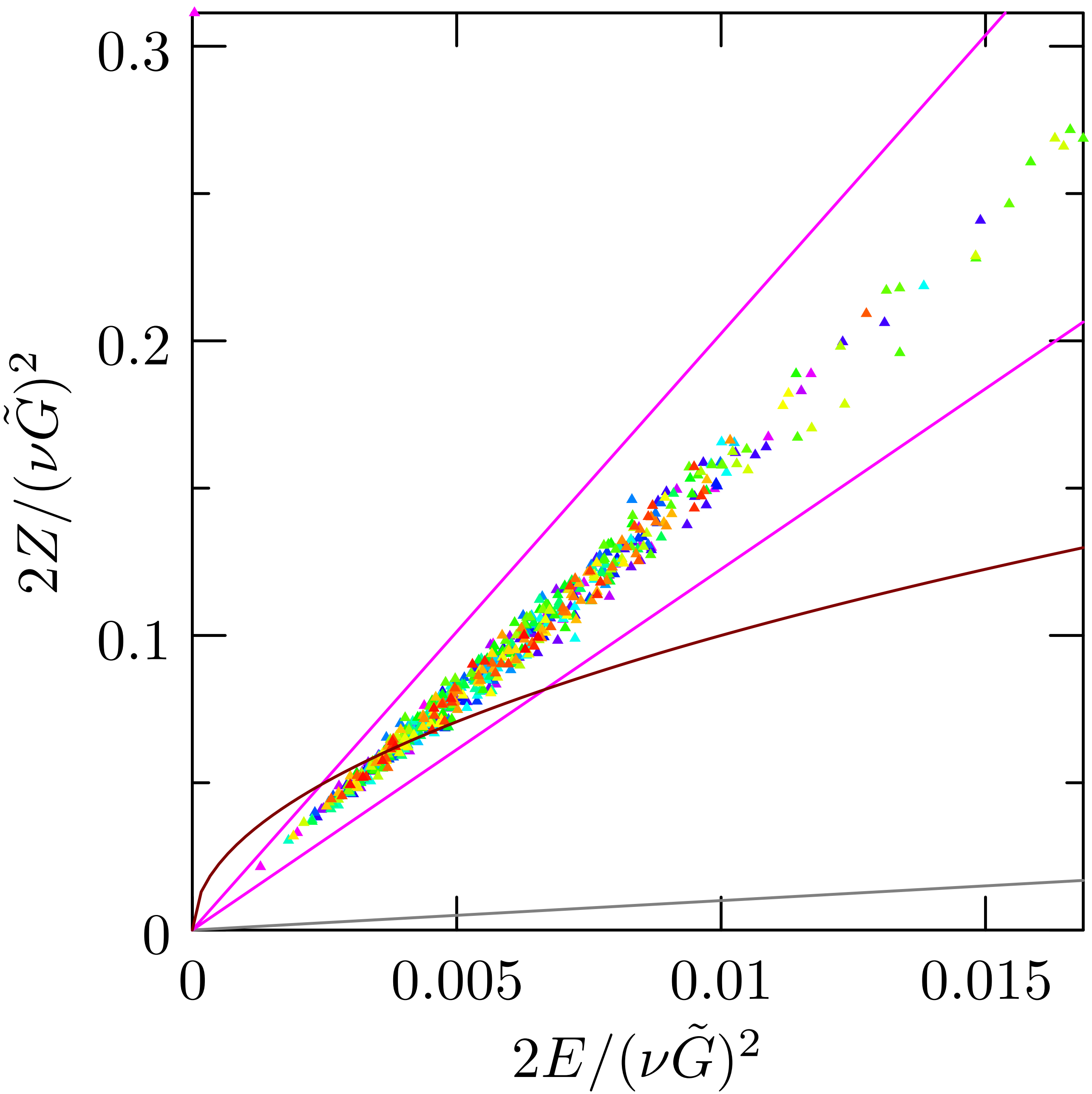
Vorticity Field with Hypoviscosity



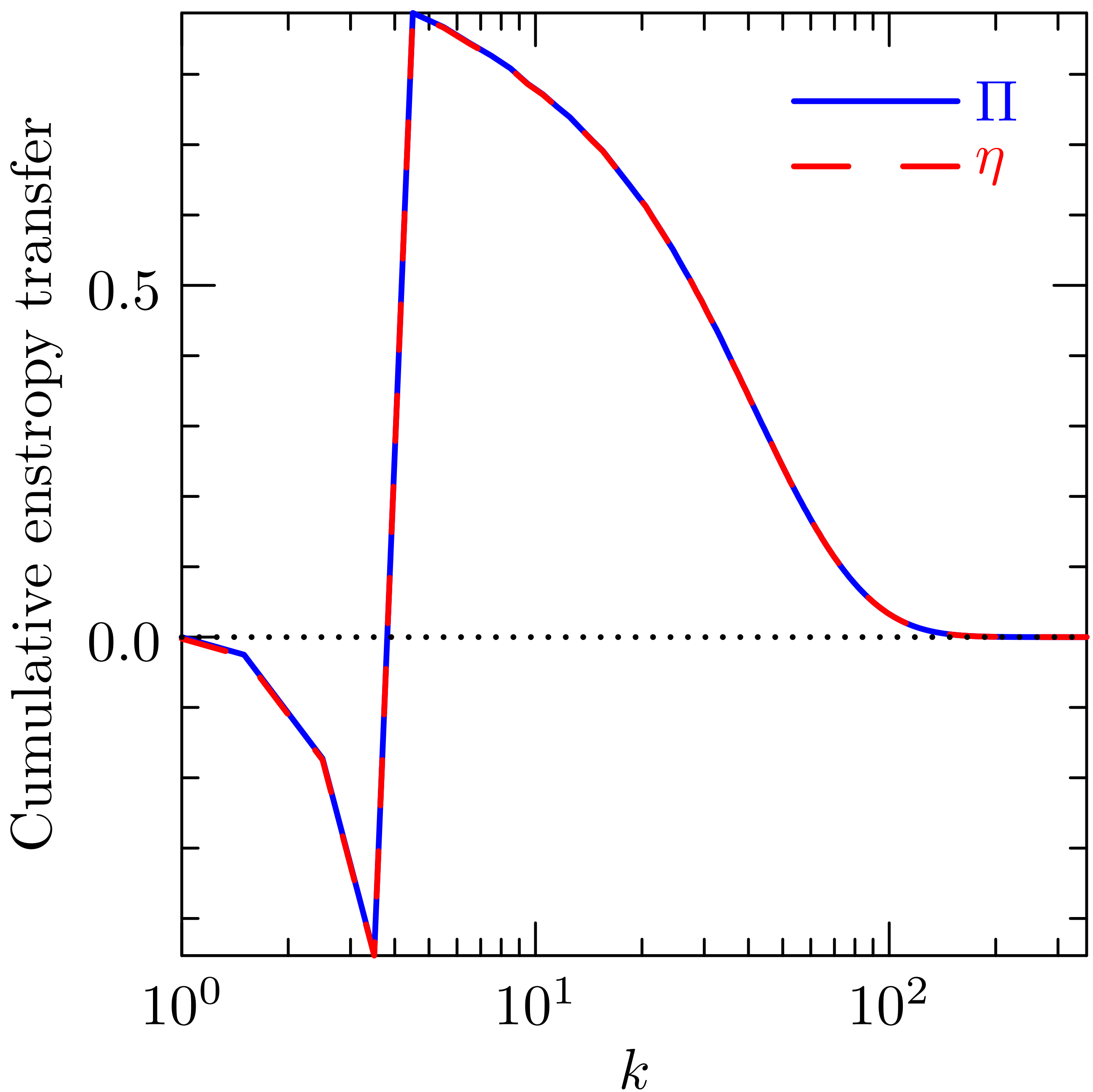
Energy Spectrum with Hypoviscosity



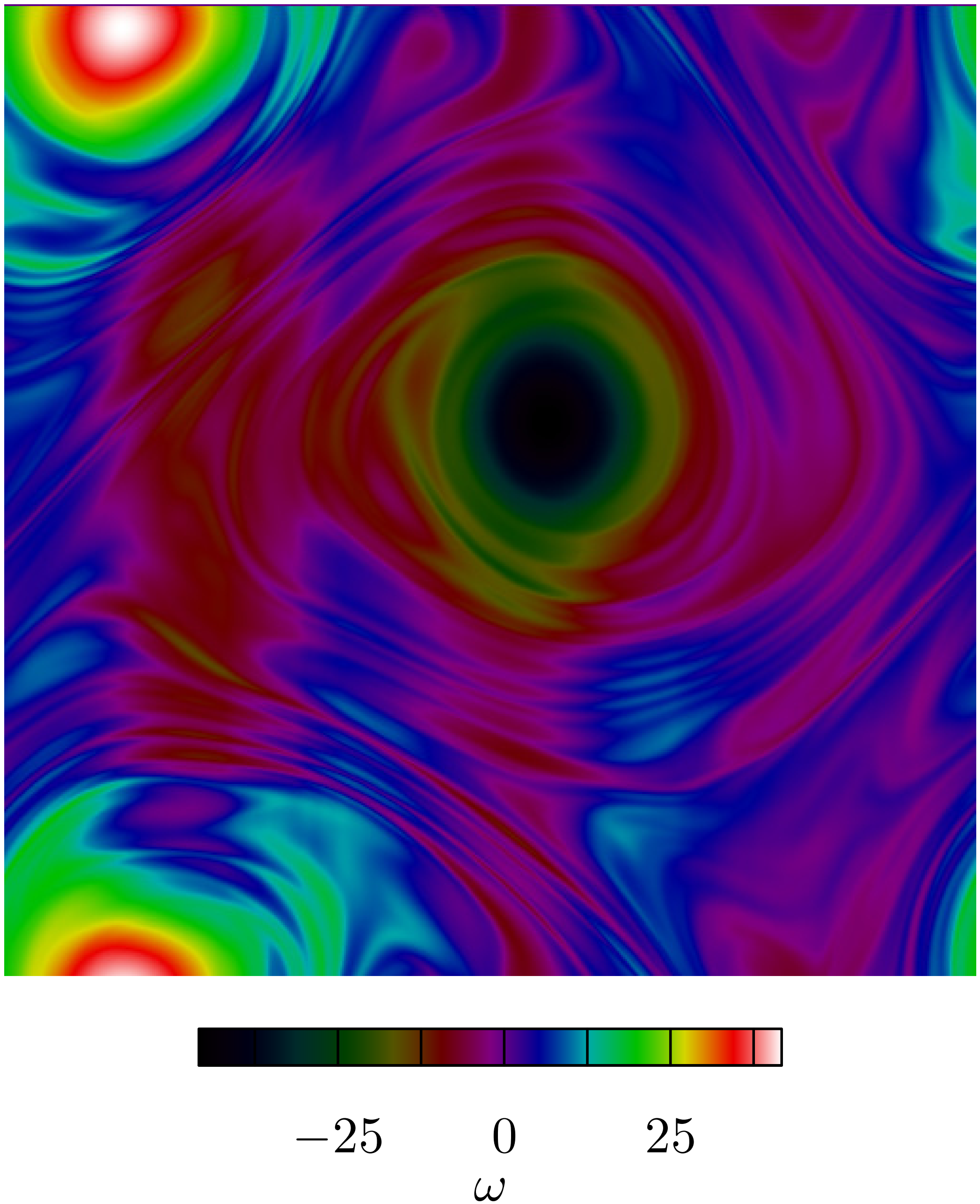
Bounds in the Z–E plane for random forcing.



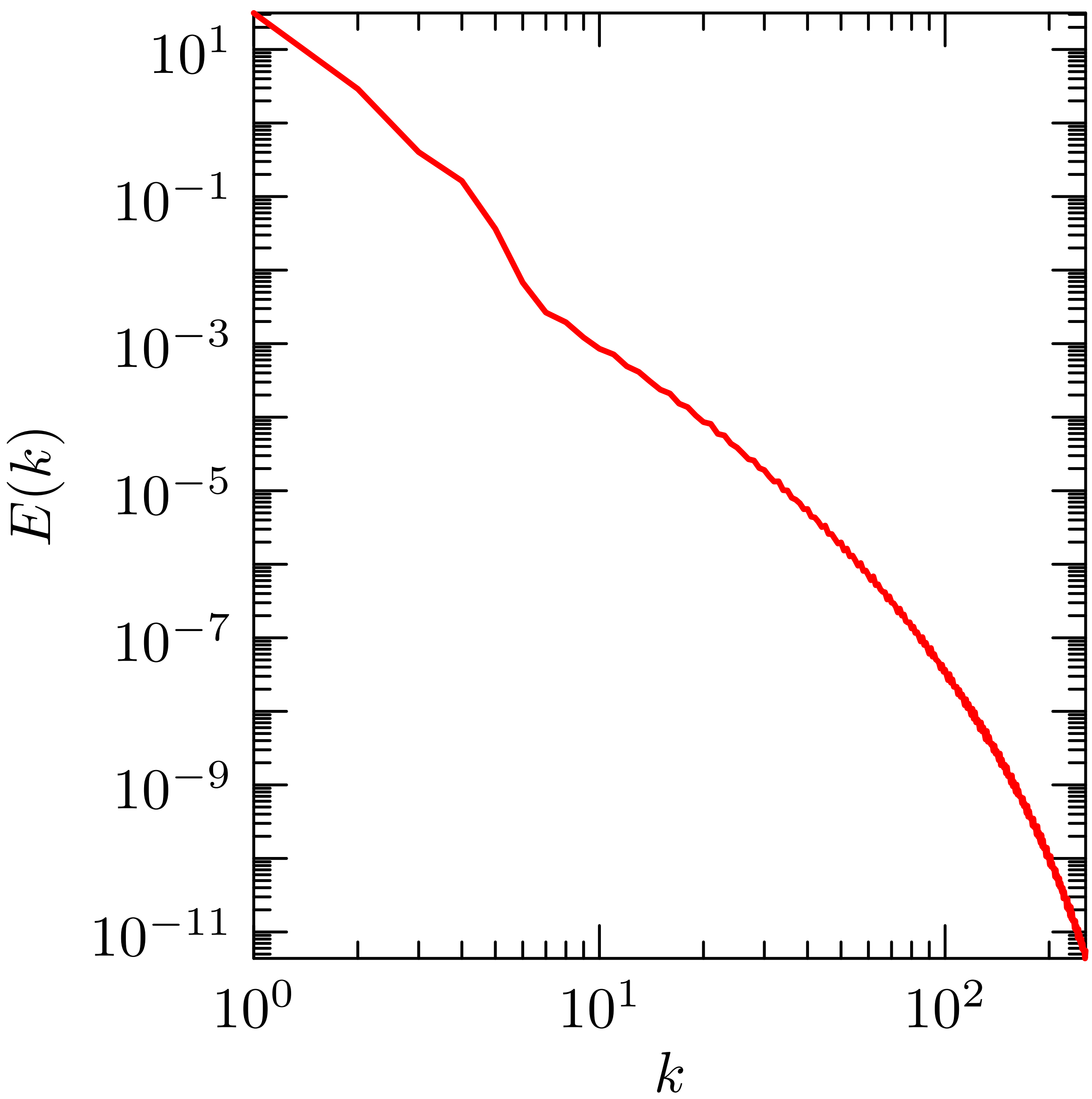
Energy Transfer with Hypoviscosity



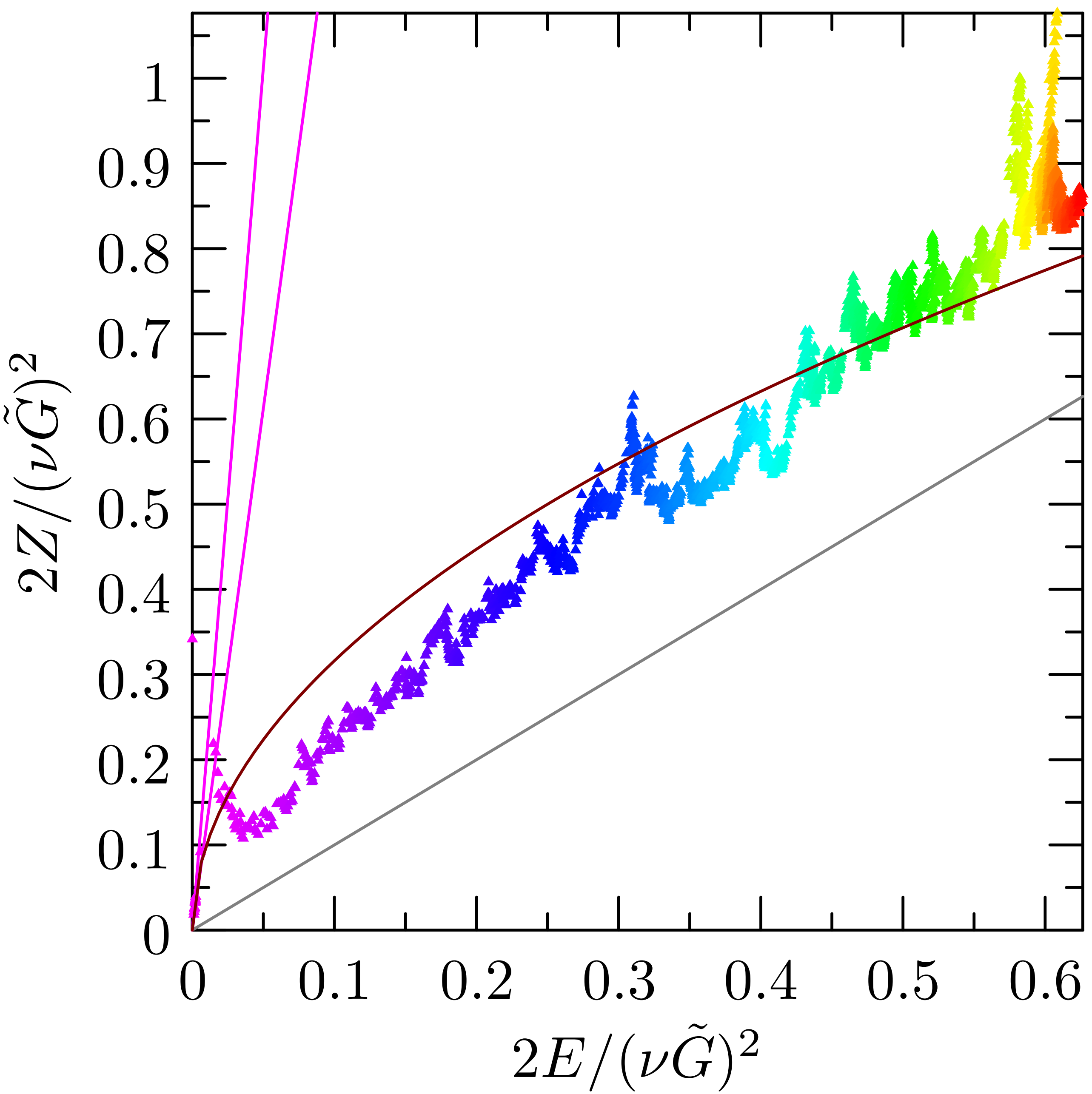
Vorticity Field without Hypoviscosity



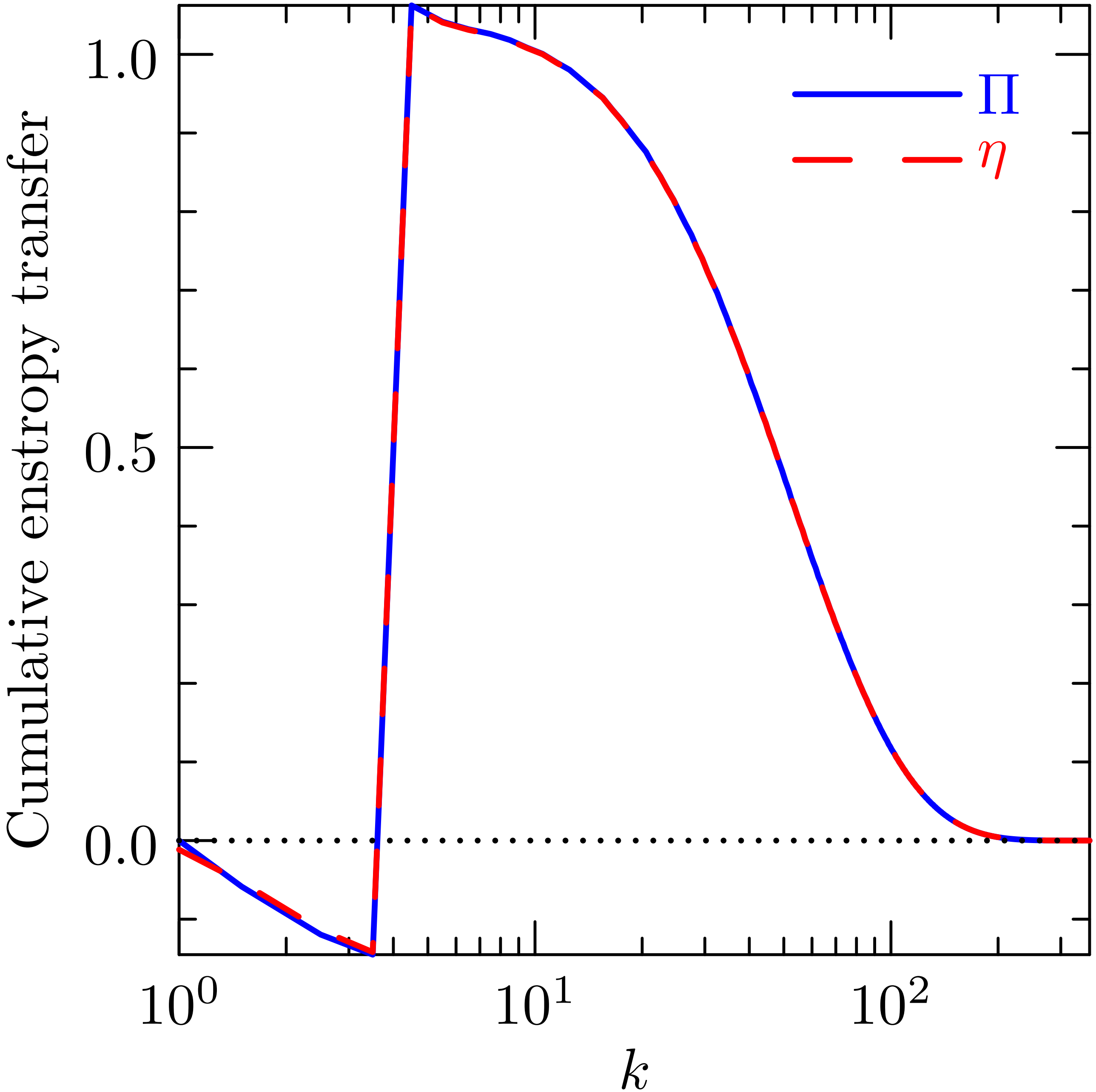
Energy Spectrum without Hypoviscosity



Bounds in the Z–E plane for random forcing.



Energy Transfer without Hypoviscosity



Special Case: White-Noise Forcing

- The Fourier transform of an isotropic Gaussian white-noise solenoidal force $\boldsymbol{f}$ has the form

$$\boldsymbol{f}_{\boldsymbol{k}}(t) = F_{\boldsymbol{k}}\left(\mathbf{1} - \frac{\boldsymbol{k}\boldsymbol{k}}{k^2}\right)\boldsymbol{\cdot}\boldsymbol{\xi}_{\boldsymbol{k}}(t), \quad \boldsymbol{k}\boldsymbol{\cdot}\boldsymbol{f}_{\boldsymbol{k}} = 0,$$

where $F_{\boldsymbol{k}}$ is a real number and $\boldsymbol{\xi}_{\boldsymbol{k}}(t)$ is a unit central real Gaussian random 2D vector that satisfies

$$\langle \boldsymbol{\xi}_{\boldsymbol{k}}(t)\boldsymbol{\xi}_{\boldsymbol{k}'}(t')\rangle = \delta_{\boldsymbol{k}\boldsymbol{k}'}\mathbf{1}\delta(t - t').$$

Special Case: White-Noise Forcing

- The Fourier transform of an isotropic Gaussian white-noise solenoidal force $\boldsymbol{f}$ has the form

$$\boldsymbol{f}_{\boldsymbol{k}}(t) = F_{\boldsymbol{k}} \left(\mathbf{1} - \frac{\boldsymbol{k}\boldsymbol{k}}{k^2} \right) \cdot \boldsymbol{\xi}_{\boldsymbol{k}}(t), \quad \boldsymbol{k} \cdot \boldsymbol{f}_{\boldsymbol{k}} = 0,$$

where $F_{\boldsymbol{k}}$ is a real number and $\boldsymbol{\xi}_{\boldsymbol{k}}(t)$ is a unit central real Gaussian random 2D vector that satisfies

$$\langle \boldsymbol{\xi}_{\boldsymbol{k}}(t) \boldsymbol{\xi}_{\boldsymbol{k}'}(t') \rangle = \delta_{\boldsymbol{k}\boldsymbol{k}'} \mathbf{1} \delta(t - t').$$

- This implies

$$\langle \boldsymbol{f}_{\boldsymbol{k}}(t) \cdot \boldsymbol{f}_{\boldsymbol{k}'}(t') \rangle = F_{\boldsymbol{k}}^2 \delta_{\boldsymbol{k},\boldsymbol{k}'} \, \delta(t - t').$$

Special Case: White-Noise Forcing

- To prescribe the forcing amplitude $F_{\boldsymbol{k}}$ in terms of ϵ:

Theorem 5 (Novikov [1964]) *If $f(\boldsymbol{x}, t)$ is a Gaussian process, and u is a functional of f, then*

$$\langle f(\boldsymbol{x}, t) u(f) \rangle = \int \int \langle f(\boldsymbol{x}, t) f(\boldsymbol{x}', t') \rangle \left\langle \frac{\delta u(\boldsymbol{x}, t)}{\delta f(\boldsymbol{x}', t')} \right\rangle d\boldsymbol{x}' \, dt'.$$

Special Case: White-Noise Forcing

- To prescribe the forcing amplitude $F_{\boldsymbol{k}}$ in terms of ϵ:

Theorem 6 (Novikov [1964]) *If* $f(\boldsymbol{x},t)$ *is a Gaussian process, and* u *is a functional of* f*, then*

$$\langle f(\boldsymbol{x},t)u(f)\rangle = \int\int \langle f(\boldsymbol{x},t)f(\boldsymbol{x}',t')\rangle \left\langle \frac{\delta u(\boldsymbol{x},t)}{\delta f(\boldsymbol{x}',t')} \right\rangle d\boldsymbol{x}'\, dt'.$$

- For white-noise forcing:

$$\epsilon = \mathrm{Re} \sum_{\boldsymbol{k}} \langle \boldsymbol{f}_{\boldsymbol{k}}(t)\cdot\overline{\boldsymbol{u}}_{\boldsymbol{k}}(t)\rangle = \mathrm{Re} \sum_{\boldsymbol{k},\boldsymbol{k}'} \int \left\langle \boldsymbol{f}_{\boldsymbol{k}}(t)\overline{\boldsymbol{f}}_{\boldsymbol{k}'}(t')\right\rangle \boldsymbol{:} \left\langle \frac{\delta \overline{\boldsymbol{u}}_{\boldsymbol{k}}(t)}{\delta \overline{\boldsymbol{f}}_{\boldsymbol{k}'}(t')} \right\rangle dt'$$

$$= \sum_{\boldsymbol{k}} F_{\boldsymbol{k}}^2 \left(\boldsymbol{1} - \frac{\boldsymbol{k}\boldsymbol{k}}{k^2}\right) \boldsymbol{:} \left(\boldsymbol{1} - \frac{\boldsymbol{k}\boldsymbol{k}}{k^2}\right) H(0)$$

$$= \frac{1}{2}\sum_{\boldsymbol{k}} F_{\boldsymbol{k}}^2,$$

on noting that $H(0) = 1/2$.

White-Noise Forcing: Implementation

- At the end of each time-step, we implement the contribution of white noise forcing with the discretization

$$\omega_{\boldsymbol{k},n+1} = \omega_{\boldsymbol{k},n} + \sqrt{2\tau\eta_{\boldsymbol{k}}}\,\xi,$$

where ξ is a unit complex Gaussian random number with $\langle\xi\rangle = 0$ and $\langle|\xi|\rangle^2 = 1$.

White-Noise Forcing: Implementation

- At the end of each time-step, we implement the contribution of white noise forcing with the discretization

$$\omega_{\boldsymbol{k},n+1} = \omega_{\boldsymbol{k},n} + \sqrt{2\tau\eta_{\boldsymbol{k}}}\,\xi,$$

  where ξ is a unit complex Gaussian random number with $\langle\xi\rangle = 0$ and $\langle|\xi|\rangle^2 = 1$.

- This yields the mean enstrophy injection

$$\frac{\left\langle|\omega_{\boldsymbol{k},n+1}|^2 - |\omega_{\boldsymbol{k},n}|^2\right\rangle}{2\tau} = \eta_{\boldsymbol{k}}.$$

3D Basdevant Formulation: 8 FFTs

- Using incompressibility, the 3D momentum equation can be written in terms of the symmetric tensor $D_{ij} = u_i u_j$:

$$\frac{\partial u_i}{\partial t} + \frac{\partial D_{ij}}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + F_i.$$

3D Basdevant Formulation: 8 FFTs

- Using incompressibility, the 3D momentum equation can be written in terms of the symmetric tensor $D_{ij} = u_i u_j$:

$$\frac{\partial u_i}{\partial t} + \frac{\partial D_{ij}}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + F_i.$$

- Naive implementation: 3 backward FFTs to compute the velocity components from their spectral representations, 6 forward FFTs of the independent components of D_{ij}.

3D Basdevant Formulation: 8 FFTs

- Using incompressibility, the 3D momentum equation can be written in terms of the symmetric tensor $D_{ij} = u_i u_j$:

$$\frac{\partial u_i}{\partial t} + \frac{\partial D_{ij}}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + F_i.$$

- Naive implementation: 3 backward FFTs to compute the velocity components from their spectral representations, 6 forward FFTs of the independent components of D_{ij}.

- Basdevant [1983]: avoid one FFT by subtracting the divergence of the symmetric matrix $S_{ij} = \delta_{ij} \operatorname{tr} D/3$ from both sides:

$$\frac{\partial u_i}{\partial t} + \frac{\partial (D_{ij} - S_{ij})}{\partial x_j} = -\frac{\partial (p\delta_{ij} + S_{ij})}{\partial x_j} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + F_i.$$

3D Basdevant Formulation: 8 FFTs

- Using incompressibility, the 3D momentum equation can be written in terms of the symmetric tensor $D_{ij} = u_i u_j$:

$$\frac{\partial u_i}{\partial t} + \frac{\partial D_{ij}}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + F_i.$$

- Naive implementation: 3 backward FFTs to compute the velocity components from their spectral representations, 6 forward FFTs of the independent components of D_{ij}.

- Basdevant [1983]: avoid one FFT by subtracting the divergence of the symmetric matrix $S_{ij} = \delta_{ij} \operatorname{tr} D/3$ from both sides:

$$\frac{\partial u_i}{\partial t} + \frac{\partial (D_{ij} - S_{ij})}{\partial x_j} = -\frac{\partial (p\delta_{ij} + S_{ij})}{\partial x_j} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + F_i.$$

- To compute the velocity components u_i, 3 backward FFTs are required. Since the symmetric matrix $D_{ij} - S_{ij}$ is traceless, it has just 5 independent components.

- Hence, a total of only 8 FFTs are required per integration stage.

- Hence, a total of only 8 FFTs are required per integration stage.
- The effective pressure $p\delta_{ij} + S_{ij}$ is solved as usual from the inverse Laplacian of the force minus the nonlinearity.

2D Basdevant Formulation: 4 FFTs

- The vorticity $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$ evolves according to

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\boldsymbol{u} + \nu \nabla^2 \boldsymbol{\omega} + \boldsymbol{\nabla} \times \boldsymbol{F},$$

where in 2D the vortex stretching term $(\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\boldsymbol{u}$ vanishes and $\boldsymbol{\omega}$ is normal to the plane of motion.

2D Basdevant Formulation: 4 FFTs

- The vorticity $\boldsymbol{\omega} = \boldsymbol{\nabla}\times\boldsymbol{u}$ evolves according to

$$\frac{\partial\boldsymbol{\omega}}{\partial t} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{\omega} = (\boldsymbol{\omega}\cdot\boldsymbol{\nabla})\boldsymbol{u} + \nu\nabla^2\boldsymbol{\omega} + \boldsymbol{\nabla}\times\boldsymbol{F},$$

  where in 2D the vortex stretching term $(\boldsymbol{\omega}\cdot\boldsymbol{\nabla})\boldsymbol{u}$ vanishes and $\boldsymbol{\omega}$ is normal to the plane of motion.

- For C^2 velocity fields, the curl of the nonlinearity can be written in terms of $\widetilde{D}_{ij} \doteq D_{ij} - S_{ij}$:

$$\frac{\partial}{\partial x_1}\frac{\partial}{\partial x_j}\widetilde{D}_{2j} - \frac{\partial}{\partial x_2}\frac{\partial}{\partial x_j}\widetilde{D}_{1j} = \left(\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2}\right)D_{12} + \frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2}(D_{22} - D_{11}),$$

  on recalling that S is diagonal and $S_{11} = S_{22}$.

2D Basdevant Formulation: 4 FFTs

- The vorticity $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$ evolves according to

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\boldsymbol{u} + \nu \nabla^2 \boldsymbol{\omega} + \boldsymbol{\nabla} \times \boldsymbol{F},$$

where in 2D the vortex stretching term $(\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\boldsymbol{u}$ vanishes and $\boldsymbol{\omega}$ is normal to the plane of motion.

- For C^2 velocity fields, the curl of the nonlinearity can be written in terms of $\widetilde{D}_{ij} \doteq D_{ij} - S_{ij}$:

$$\frac{\partial}{\partial x_1}\frac{\partial}{\partial x_j}\widetilde{D}_{2j} - \frac{\partial}{\partial x_2}\frac{\partial}{\partial x_j}\widetilde{D}_{1j} = \left(\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2}\right) D_{12} + \frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2}(D_{22} - D_{11}),$$

on recalling that S is diagonal and $S_{11} = S_{22}$.

- The scalar vorticity ω thus evolves as

$$\frac{\partial \omega}{\partial t} + \left(\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2}\right)(u_1 u_2) + \frac{\partial^2}{\partial x_1 \partial x_2}\left(u_2^2 - u_1^2\right) = \nu \nabla^2 \omega + \frac{\partial F_2}{\partial x_1} - \frac{\partial F_1}{\partial x_2}.$$

- To compute u_1 and u_2 in physical space, we need 2 backward FFTs.

- To compute u_1 and u_2 in physical space, we need 2 backward FFTs.
- The quantities $u_1 u_2$ and $u_2^2 - u_1^2$ can then be calculated and then transformed to Fourier space with 2 additional forward FFTs.

- To compute u_1 and u_2 in physical space, we need 2 backward FFTs.
- The quantities $u_1 u_2$ and $u_2^2 - u_1^2$ can then be calculated and then transformed to Fourier space with 2 additional forward FFTs.
- The advective term in 2D can thus be calculated with just 4 FFTs.

3D Incompressible MHD: 17 FFTs

$$\frac{\partial u_i}{\partial t} + \frac{\partial (D_{ij} - S_{ij})}{\partial x_j} = -\frac{\partial (p\delta_{ij} + S_{ij})}{\partial x_j} + \nu \frac{\partial^2 u_i}{\partial x_j^2},$$

$$\frac{\partial B_i}{\partial t} + \frac{\partial G_{ij}}{\partial x_j} = \eta \frac{\partial^2 B_i}{\partial x_j^2},$$

where $D_{ij} = u_i u_j - B_i B_j$, $S_{ij} = \delta_{ij} \operatorname{tr} D/3$, and

$$G_{ij} = B_i u_j - u_i B_j.$$

- The traceless matrix $D_{ij} - S_{ij}$ has 8 independent components.

3D Incompressible MHD: 17 FFTs

$$\frac{\partial u_i}{\partial t} + \frac{\partial (D_{ij} - S_{ij})}{\partial x_j} = -\frac{\partial (p\delta_{ij} + S_{ij})}{\partial x_j} + \nu \frac{\partial^2 u_i}{\partial x_j^2},$$

$$\frac{\partial B_i}{\partial t} + \frac{\partial G_{ij}}{\partial x_j} = \eta \frac{\partial^2 B_i}{\partial x_j^2},$$

where $D_{ij} = u_i u_j - B_i B_j$, $S_{ij} = \delta_{ij} \operatorname{tr} D/3$, and

$$G_{ij} = B_i u_j - u_i B_j.$$

- The traceless matrix $D_{ij} - S_{ij}$ has 8 independent components.
- The antisymmetric matrix G_{ij} has only 3.

3D Incompressible MHD: 17 FFTs

$$\frac{\partial u_i}{\partial t} + \frac{\partial (D_{ij} - S_{ij})}{\partial x_j} = -\frac{\partial (p\delta_{ij} + S_{ij})}{\partial x_j} + \nu \frac{\partial^2 u_i}{\partial x_j^2},$$

$$\frac{\partial B_i}{\partial t} + \frac{\partial G_{ij}}{\partial x_j} = \eta \frac{\partial^2 B_i}{\partial x_j^2},$$

where $D_{ij} = u_i u_j - B_i B_j$, $S_{ij} = \delta_{ij} \operatorname{tr} D/3$, and

$$G_{ij} = B_i u_j - u_i B_j.$$

- The traceless matrix $D_{ij} - S_{ij}$ has 8 independent components.
- The antisymmetric matrix G_{ij} has only 3.
- An additional 6 FFT calls are required to compute the components of $\boldsymbol{u}$ and $\boldsymbol{B}$ in x space.

3D Incompressible MHD: 17 FFTs

$$\frac{\partial u_i}{\partial t} + \frac{\partial(D_{ij} - S_{ij})}{\partial x_j} = -\frac{\partial(p\delta_{ij} + S_{ij})}{\partial x_j} + \nu\frac{\partial^2 u_i}{\partial x_j^2},$$

$$\frac{\partial B_i}{\partial t} + \frac{\partial G_{ij}}{\partial x_j} = \eta\frac{\partial^2 B_i}{\partial x_j^2},$$

where $D_{ij} = u_i u_j - B_i B_j$, $S_{ij} = \delta_{ij}\,\mathrm{tr}\,D/3$, and

$$G_{ij} = B_i u_j - u_i B_j.$$

- The traceless matrix $D_{ij} - S_{ij}$ has 8 independent components.
- The antisymmetric matrix G_{ij} has only 3.
- An additional 6 FFT calls are required to compute the components of $\boldsymbol{u}$ and $\boldsymbol{B}$ in x space.
- The MHD nonlinearity can thus be computed with 17 FFT calls.

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.

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

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

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

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.

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

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

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

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

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!

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 $2m \log_2 m = N \log_2 m$.

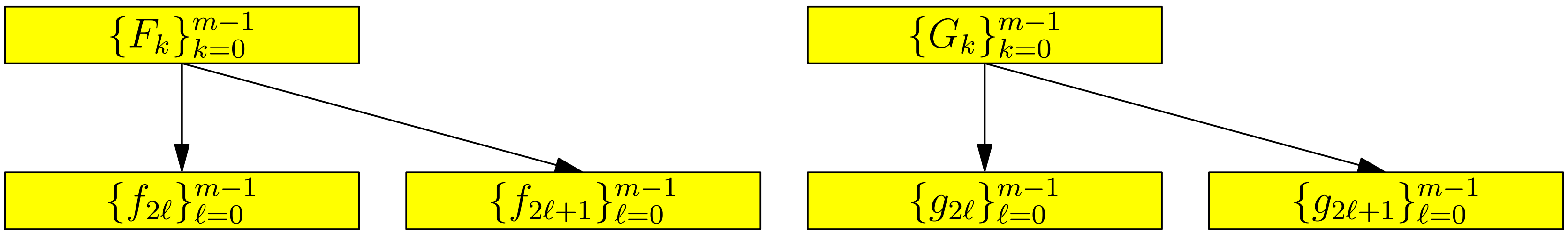
- Parallelized multidimensional implicit dealiasing routines have been implemented as a software layer **FFTW++** (v 2.05) on top of the **FFTW** library under the Lesser GNU Public License:

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

$\{F_k\}_{k=0}^{m-1}$ $\{G_k\}_{k=0}^{m-1}$

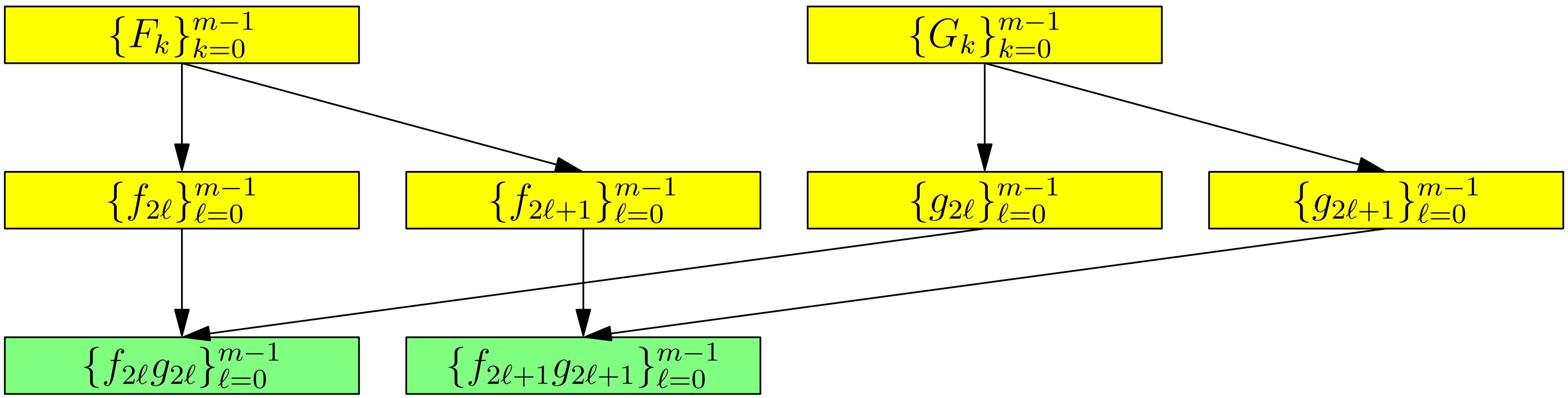
- Parallelized multidimensional implicit dealiasing routines have been implemented as a software layer **FFTW++** (v 2.05) on top of the **FFTW** library under the Lesser GNU Public License:

  http://fftwpp.sourceforge.net/



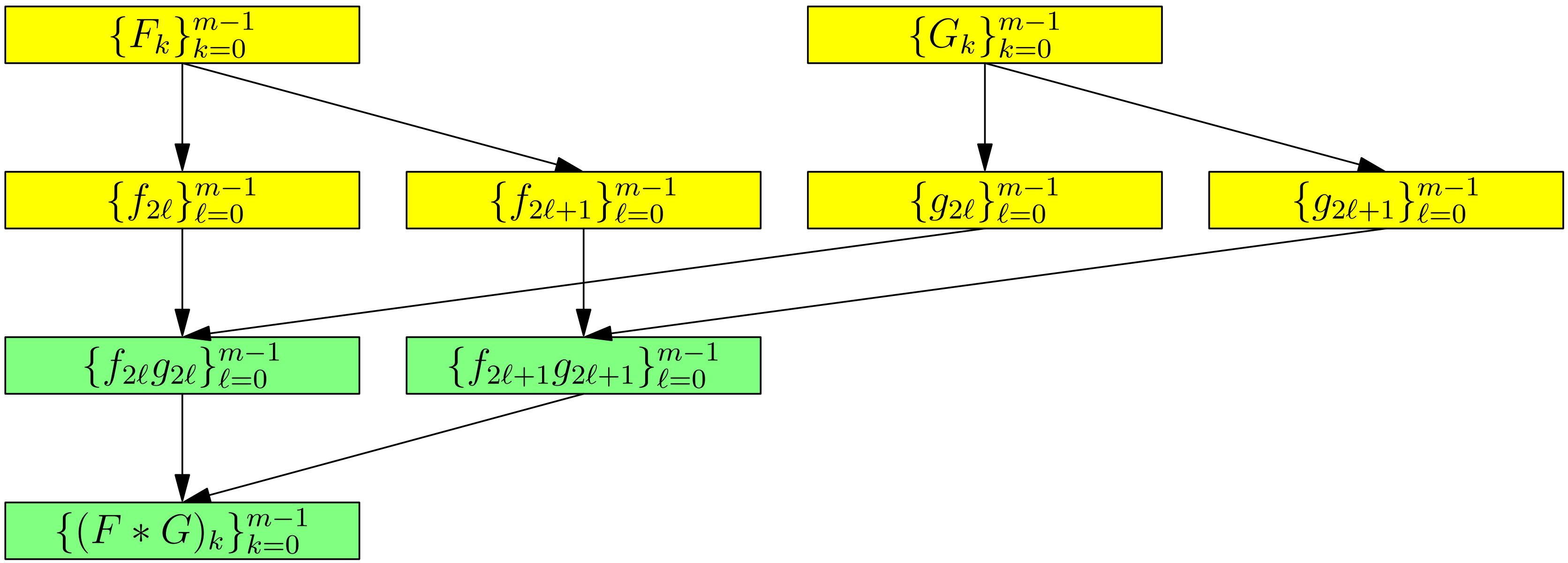
- Parallelized multidimensional implicit dealiasing routines have been implemented as a software layer **FFTW++** (v 2.05) on top of the **FFTW** library under the Lesser GNU Public License:

http://fftwpp.sourceforge.net/



- Parallelized multidimensional implicit dealiasing routines have been implemented as a software layer **FFTW++** (v 2.05) on top of the **FFTW** library under the Lesser GNU Public License:

  http://fftwpp.sourceforge.net/



Conclusions

- The upper bound in the Z–E plane obtained for constant forcing also works for the white-noise forcing.

Conclusions

- The upper bound in the Z–E plane obtained for constant forcing also works for the white-noise forcing.

- Adding hypoviscosity to the Navier–Stokes equation has a dramatic effect on the turbulent dynamics: it restricts the global attractor to the region characterized by the forcing annulus.

Conclusions

- The upper bound in the Z–E plane obtained for constant forcing also works for the white-noise forcing.
- Adding hypoviscosity to the Navier–Stokes equation has a dramatic effect on the turbulent dynamics: it restricts the global attractor to the region characterized by the forcing annulus.
- With these tools, it should now be possible to study the relation between white-noise and constant forcings by examining their effects on the global attractor.

Conclusions

- The upper bound in the Z–E plane obtained for constant forcing also works for the white-noise forcing.

- Adding hypoviscosity to the Navier–Stokes equation has a dramatic effect on the turbulent dynamics: it restricts the global attractor to the region characterized by the forcing annulus.

- With these tools, it should now be possible to study the relation between white-noise and constant forcings by examining their effects on the global attractor.

- This may lead to an explicit relation for the energy and enstrophy injection rates for constant forcing.

Conclusions

- The upper bound in the Z–E plane obtained for constant forcing also works for the white-noise forcing.
- Adding hypoviscosity to the Navier–Stokes equation has a dramatic effect on the turbulent dynamics: it restricts the global attractor to the region characterized by the forcing annulus.
- With these tools, it should now be possible to study the relation between white-noise and constant forcings by examining their effects on the global attractor.
- This may lead to an explicit relation for the energy and enstrophy injection rates for constant forcing.
- Analytical bounds for random forcing provide a means to evaluate various heuristic turbulent subgrid (and supergrid!) models by characterizing the behaviour of the global attractor under these models.

References

[Basdevant 1983] C. Basdevant, Journal of Computational Physics, **50**:209, 1983.

[Bowman & Roberts 2010] J. C. Bowman & M. Roberts, `FFTW++`: A fast Fourier transform C^{++} header class for the `FFTW3` library, http://fftwpp.sourceforge.net, May 6, 2010.

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

[Dascaliuc *et al.* 2005] R. Dascaliuc, M. Jolly, & C. Foias, Journal of Dynamics and Differential Equations, **17**:643, 2005.

[Dascaliuc *et al.* 2007] R. Dascaliuc, C. Foias, & M. Jolly, Journal of mathematical physics, **48**:065201, 2007.

[Dascaliuc *et al.* 2010] R. Dascaliuc, C. Foias, & M. Jolly, Journal of Differential Equations, **248**:792, 2010.

[Emami & Bowman 2017] P. Emami & J. C. Bowman, submitted to Journal of Differential Equations, 2017.

[Foias & Temam 1979] C. Foias & R. Temam, Journal de mathematiques pures et appliquees, **58**:339, 1979.

[Foias *et al.* 2013] C. Foias, M. S. Jolly, & M. Yang, Journal of Dynamics and Differential Equations, **25**:393, 2013.

[Ladyzhenskaya 1975] O. Ladyzhenskaya, Journal of Soviet Mathematics, **3**:458, 1975.

[Novikov 1964] E. A. Novikov, J. Exptl. Theoret. Phys. (U.S.S.R), **47**:1919, 1964.

[Roberts & Bowman 2017] M. Roberts & J. C. Bowman, Submitted to Journal of Computational Physics, 2017.