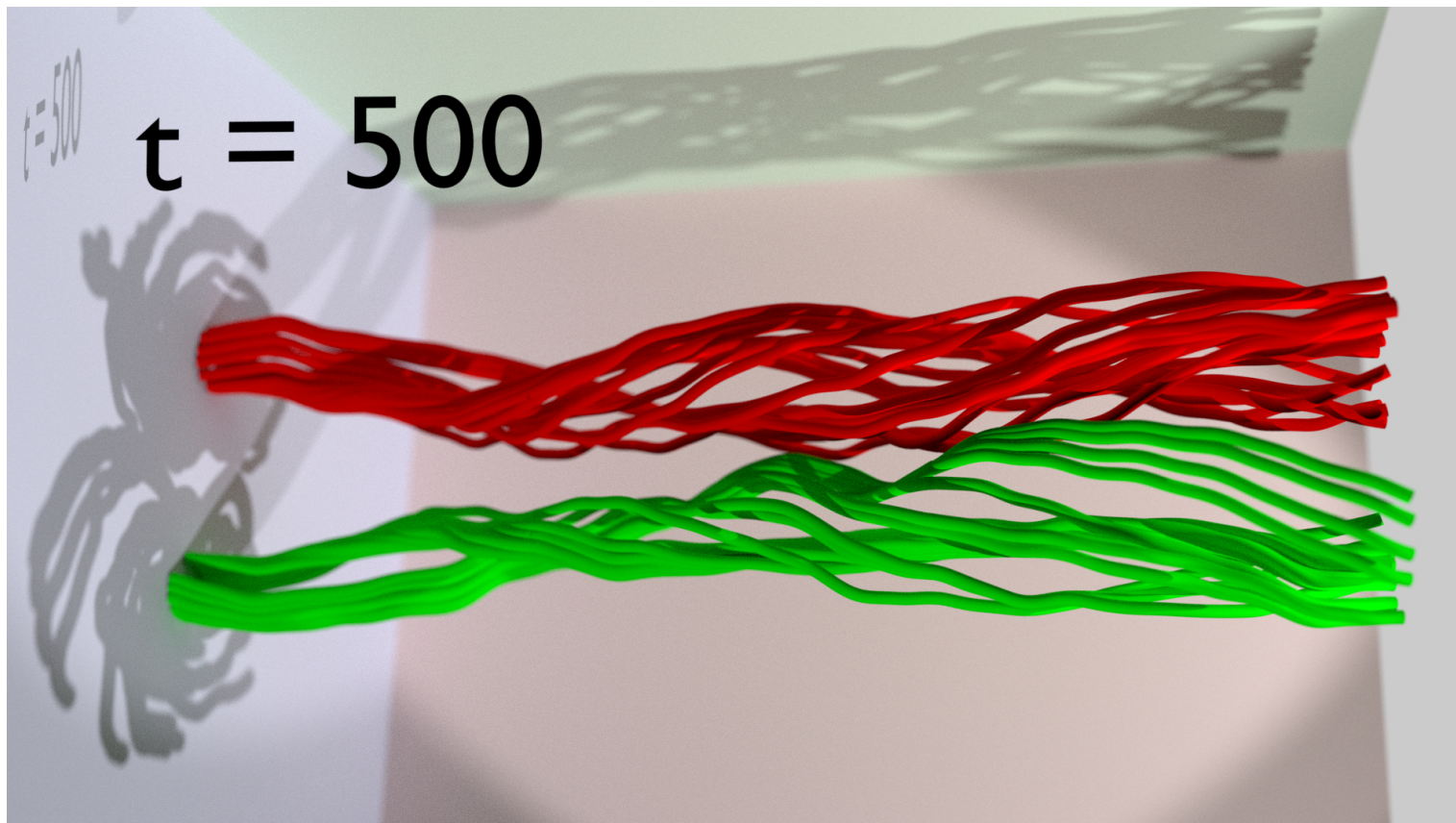




# Vortex Reconnection and the Role of Topology

Simon Candelaresi, Gunnar Hornig,  
Benjamin Podger, David I. Pontin



# Magnetic Case

Conservation of magnetic helicity:

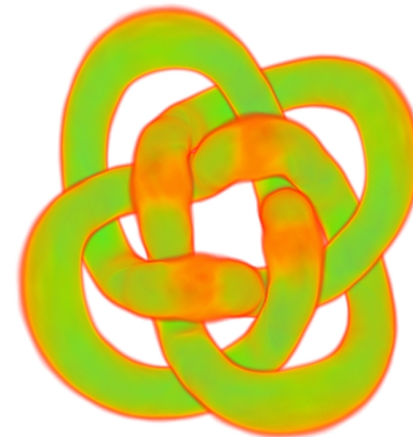
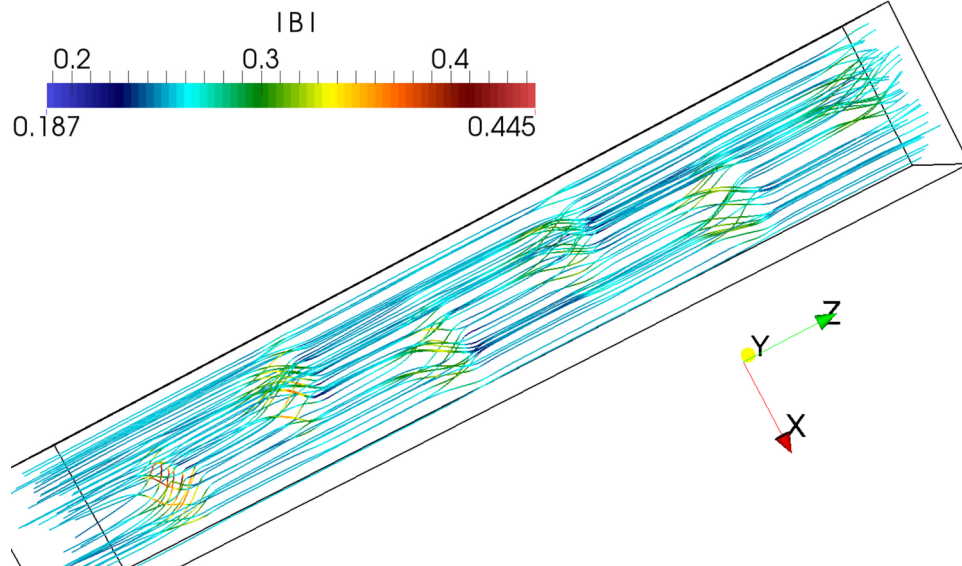
$$\lim_{\eta \rightarrow 0} \frac{\partial}{\partial t} \langle \mathbf{A} \cdot \mathbf{B} \rangle = 0 \quad \eta = \text{magnetic resistivity}$$

Realizability condition:

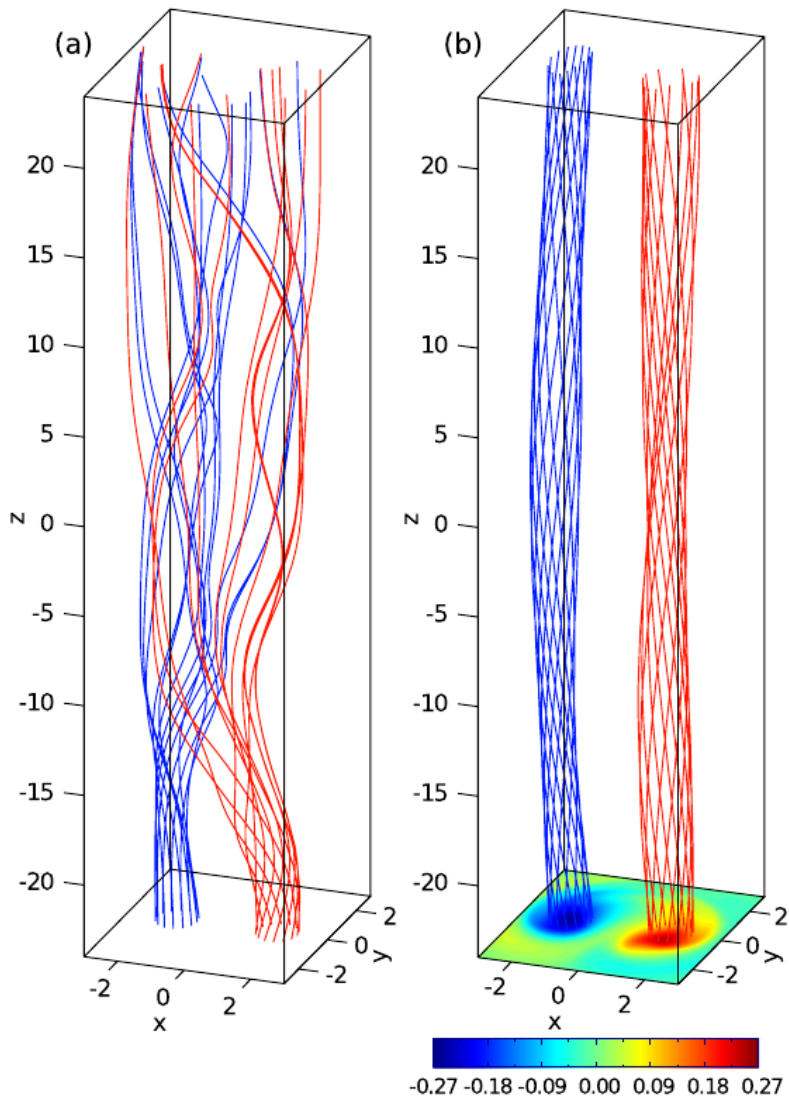
$$E_m(k) \geq k |H(k)| / 2\mu_0$$



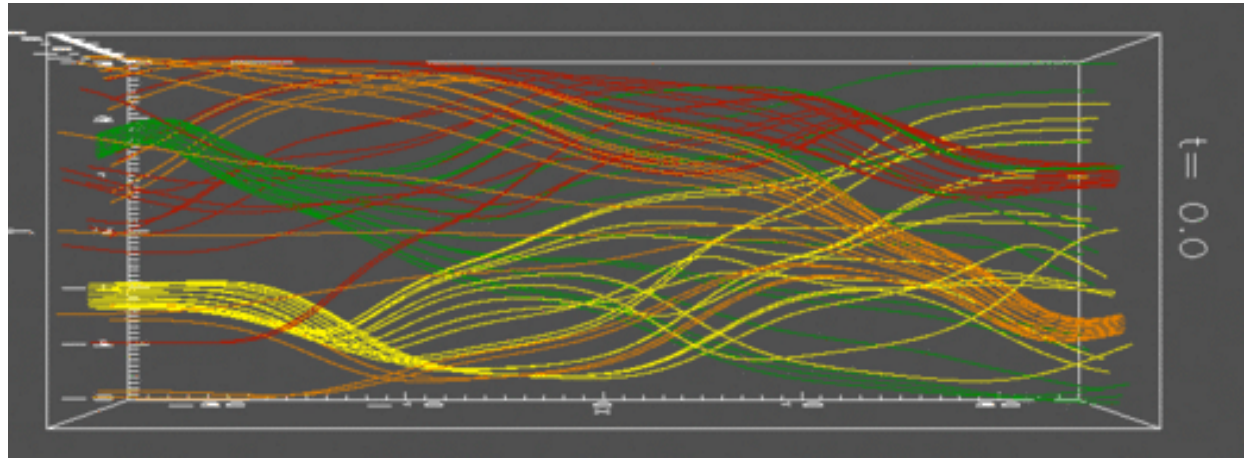
Magnetic energy is bound from below by magnetic helicity.



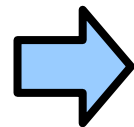
# Magnetic Braid



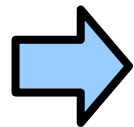
(Yeates 2011)



(Wilmot-Smith 2010)



Separation into two twisted field regions.



Conserved invariants like fixed point index and field line helicity.

# Navier-Stokes Case

$$\nabla \times \mathbf{A} = \mathbf{B} \quad \mathbf{B} \rightarrow \boldsymbol{\omega} \quad H_m = \int \mathbf{A} \cdot \mathbf{B} \, dV$$

$$\nabla \times \mathbf{u} = \boldsymbol{\omega} \quad \mathbf{A} \rightarrow \mathbf{u} \quad H_k = \int \mathbf{u} \cdot \boldsymbol{\omega} \, dV$$



No realizability condition for the hydro case.



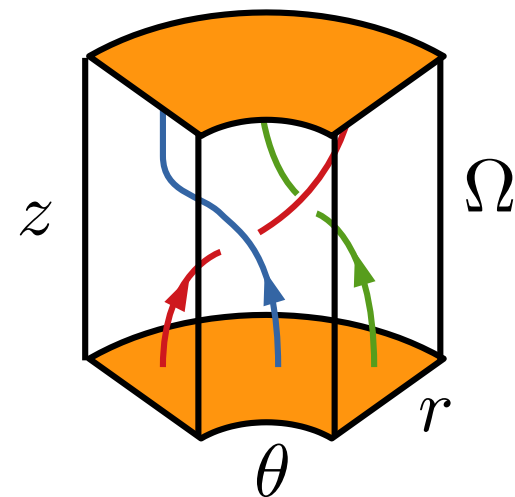
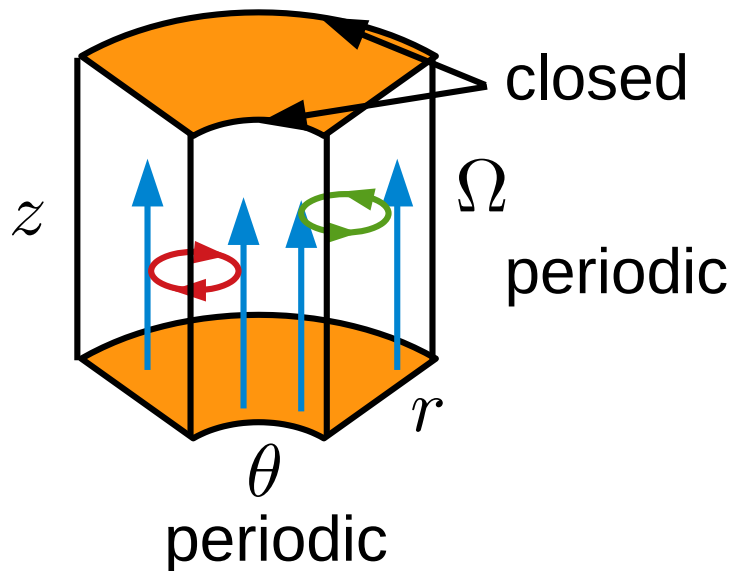
How does the field line topology affect the dynamics?

# Vortex Braid Experiments

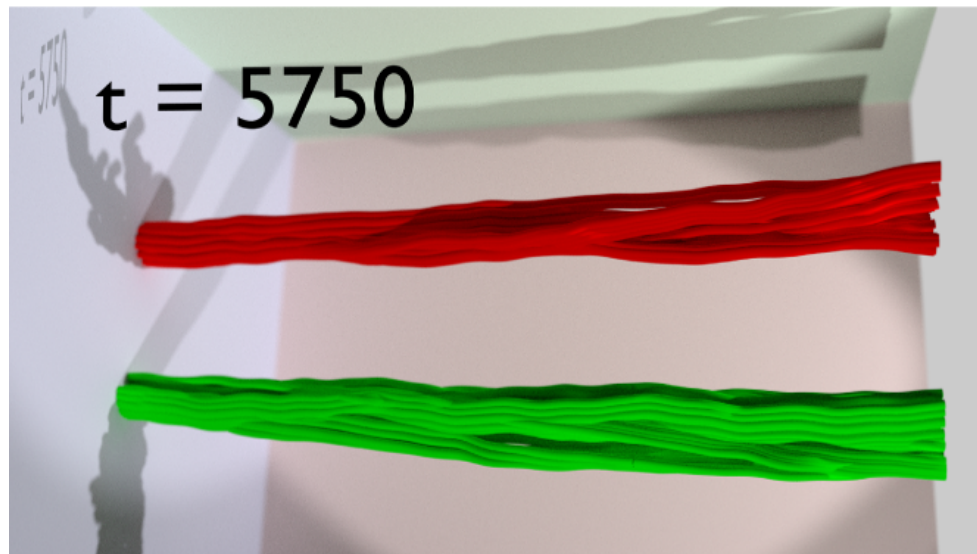
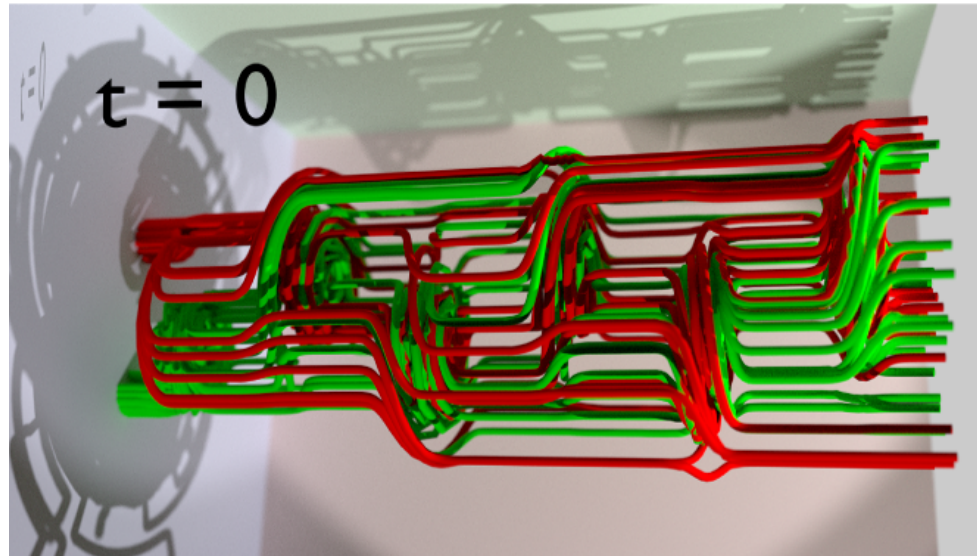
Full viscous simulations with the PencilCode.

$$\frac{D\mathbf{u}}{Dt} = -c_s^2 \nabla \ln(\rho) + 2\mathbf{u} \times \tilde{\Omega} + \mathbf{F}_{\text{visc}}$$

$$\frac{D \ln(\rho)}{Dt} = -\nabla \cdot \mathbf{u}$$



# Vortex Reconnection



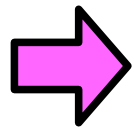
Field lines untangle into two twisted vortex tubes.

# Kinetic Energy Bound

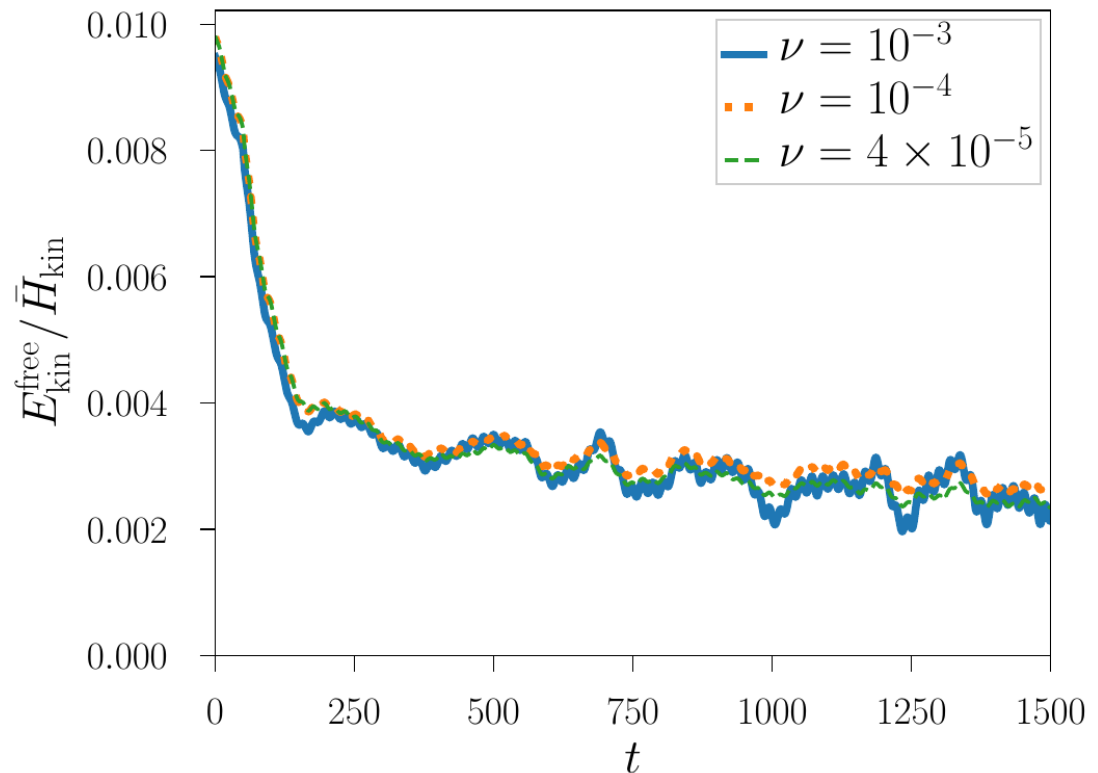
free kinetic energy:  $E_{\text{kin}}^{\text{free}} = \int \rho \left( \frac{1}{2} \mathbf{u}^2 + \mathbf{u} \cdot \mathbf{U} \right) dV$

$$\nabla \times \mathbf{U} = \boldsymbol{\Omega}$$

unsigned kinetic helicity:  $\bar{H}_{\text{kin}} = \int |(\boldsymbol{\omega} + \boldsymbol{\Omega}) \cdot (\mathbf{u} + \mathbf{U})| dV$



Field topology  
affects relaxation.

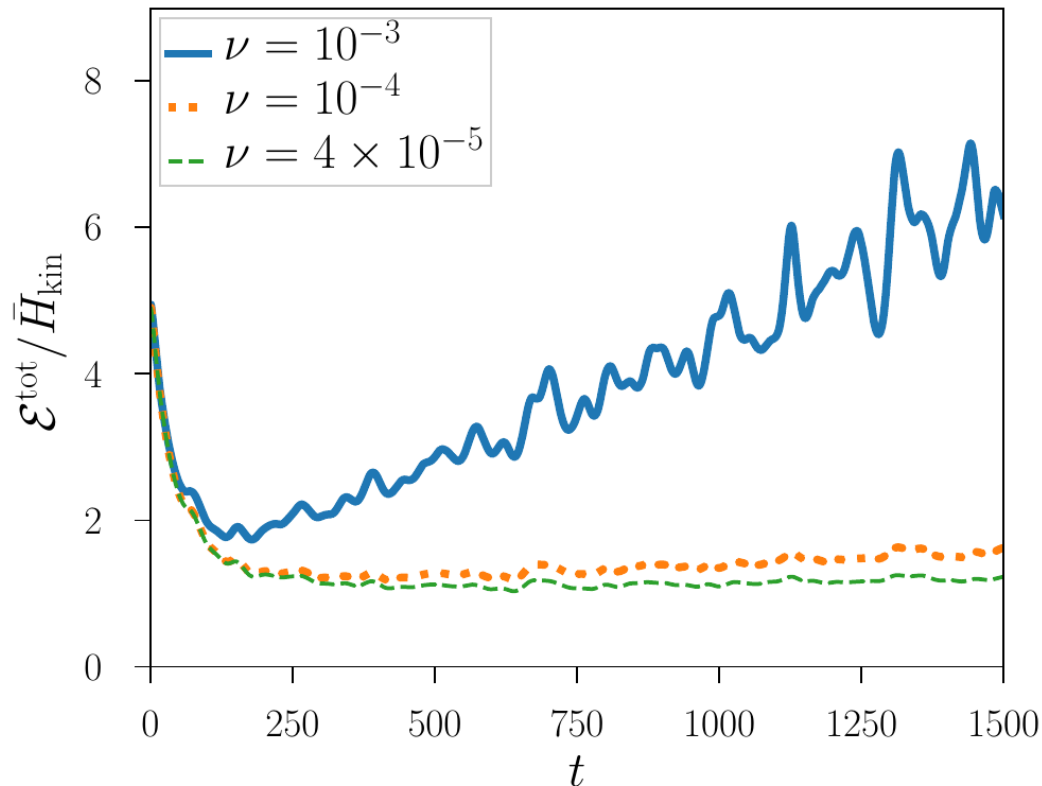


# Vortex Braid Enstrophy

enstrophy:  $\mathcal{E}^{\text{tot}} = \int (\boldsymbol{\omega} + \boldsymbol{\Omega})^2 dV$

$$\bar{H} \leq \frac{1}{\lambda} \mathcal{E}$$

$\lambda$  depends on the geometry of the domain.



Hydrodynamical realizability condition.



# Conclusions

- Topology preserving relaxation of vortex fields.
- Unbraiding into two twisted vortex flux tubes.
- Unsigned helicity limits energy decay.
- Realizability condition with enstrophy.

Physics of Fluids 33, 056101 (2021); [doi.org/10.1063/5.0047033](https://doi.org/10.1063/5.0047033)



[simon.candelaresi@gmail.com](mailto:simon.candelaresi@gmail.com)

# BlenDaViz

The screenshot displays the Blender 2.81.1 interface. The top menu bar includes File, Edit, Render, Window, Help, Layout, Modeling, Sculpting, UV Editing, Texture Paint, Shading, Animation, Rendering, Compositing, and Scripting. The main 3D viewport is split into two views: 'User Perspective' (top) and 'Camera Perspective' (bottom). Both views show a green helix object. The 'User Perspective' view shows the object from a top-down angle, while the 'Camera Perspective' view shows it from a side angle. The right sidebar contains the 'Scene' and 'World' properties panels. The 'World' panel shows the 'Surface' property set to 'Background' and 'Strength' set to 10.000. The 'Volume' property is set to 'None'. The 'Ray Visibility' panel shows 'Camera' checked. The 'Settings' panel shows 'Color' set to white. The 'View Console' window in the center-right displays the following Python code and output:

```
PYTHON INTERACTIVE CONSOLE 3.7.4 (default, Oct 8 2019, 15:23:02) [GCC 6.3.1 20170216 (Red Hat 6.3.1-3)]

Command History:      Up/Down Arrow
Cursor:                Left/Right Home/End
Remove:                Backspace/Delete
Execute:               Enter
Autocomplete:         Ctrl-Space
Zoom:                  Ctrl +/-, Ctrl-Wheel
Builtin Modules:      bpy, bpy.data, bpy.ops, bpy.props,
bpy.types, bpy.context, bpy.utils, bgl, blf, mathutils
Convenience Imports:  from mathutils import *; from math
import *
Convenience Variables: C = bpy.context, D = bpy.data

>>> import blendaviz as blt
>>> import numpy as np
>>> y = np.linspace(0, 6*np.pi, 400)
>>> x = 2*np.cos(y)
>>> z = 2*np.sin(y)
>>> pl = blt.plot(x, y, z, radius=0.5)
>>>
```

The bottom status bar shows the following information: Collection | Verts:8,000 | Faces:7,980 | Tris:15,960 | Objects:0/4 | Mem: 34.5 MiB | v2.81.1

[github.com/SimonCan/BlenDaViz](https://github.com/SimonCan/BlenDaViz)