

Dedalus

Kelvin-Helmholtz with Dedalus (a 2-D PDE)

It's time to start playing with PDEs and [Dedalus](#). We'll jump straight in to 2-D PDEs of the Kelvin-Helmholtz instability. This one can be solved on your laptops.

1. Use Mercurial and clone the Kelvin-Helmholtz repository located at https://bitbucket.org/bpbrown/kelvin_helmholtz. If you have Mercurial on your system, this can be done with

```
$ hg clone https://bitbucket.org/bpbrown/kelvin_helmholtz
```

The Kelvin-Helmholtz problem will be downloaded and stored in a subdirectory named `kelvin_helmholtz`. If the `hg` command isn't recognized, please install Mercurial (<http://mercurial.selenic.com>).

Run the basic problem by executing:

```
python3 KH_incompressible.py
```

and plot the results with

```
python3 plot_results_parallel.py KH_incompressible slices 1 1 10
```

2. The equations for this Kelvin-Helmholtz problem are:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{Re} \nabla^2 \mathbf{u} = -\mathbf{u} \cdot \nabla \mathbf{u} \quad (1)$$

$$\frac{\partial T}{\partial t} - \frac{1}{Pe} \nabla^2 T = -\mathbf{u} \cdot \nabla T \quad (2)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (3)$$

with vector velocity $\mathbf{u} = u\hat{x} + w\hat{z}$ and passive scalar tracer (here temperature) T . Here we have non-dimensionalized the equations based on the large-scale flow crossing time $\tau = L/U_0$, where L is the domain size and U_0 is the jump in amplitude of the horizontal shear flow across the tanh profile. The control parameters are the thermal Reynolds number Re and the Prandtl number Pr , with the Peclet number $Pe = RePr$. The initial horizontal flow and temperature field are given by a tanh. Open `equations.py` to find the equations as implemented. Note that here the boundary conditions are set up to continue enforcing the shear flow, and are set to be consistent with our initial u and T .

3. The initial conditions, parameters, and simulation resolution are all set in `KH_incompressible.py`. This is also where the output and analysis is done, though the actual image generation is done in `plot_results_parallel.py`. The `KH.py` file includes a somewhat similar case, but with a different non-dimensionalization, while `KH_periodic.py` applies doubly-periodic boundary conditions. Best of luck!