Advection-diffusion of a scalar

YBS

August 4, 2017

1 Governing equations

The time evolution of the concentration $c(\boldsymbol{r},t)$ of a chemical which undergoes simultaneous diffusion (with diffusion constant η) and advection by a velocity field $\boldsymbol{v}(\boldsymbol{r},t)$ is given by

$$\frac{\partial c}{\partial t}+\boldsymbol{v}\cdot\nabla c=\eta\nabla^2 c \tag{1}$$

Te left-hand-side is the material time derivative (aka advective derivative). The right-hand-side is diffusion. Physically, $\boldsymbol{v}$ should also satisfy an evolution equation of its own, but for our purposes we'll just assume it's constant in time. However, we will add the (physically valid) constraint that $\boldsymbol{v}$ is divergence-free, i.e. $\nabla\cdot v=\sum_i \partial_i v_i=0$. In 2D, a divergence-free vector field over a periodic domain $[0,2\pi]\times[0,2\pi]$ can be decomposed to Fourier components of the form

$$v_x=\sum_i A_i m_i\cos(m_i y+\beta_i)\cos(n_i x+\alpha_i)\ ,\qquad v_y=\sum_i A_i n_i\sin(m_i y+\beta_i)\sin(n_i x+\alpha_i)\ , \tag{2}$$

where A_i is the amplitude, m_i,n_i are integers ((m_i,n_i) is the wave-vector), and α_i,β_i are phases. For the simulation I simply draw all these parameters from a uniform distribution, using 5 terms for each simulation.

2 Numerical implementation

First, we write the time derivative explicitly as

$$\frac{\partial c}{\partial t}=\eta\nabla^2 c-\boldsymbol{v}\cdot\nabla c=\eta\nabla^2 c-\nabla\cdot(c\boldsymbol{v})\ . \tag{3}$$

The last transition is valid because $\nabla\cdot\boldsymbol{v}=0$.

The equations are solved on a square grid, using a staggered grid for v. That is, c is calculated on the points $\left(2\pi\frac{i}{N},2\pi\frac{j}{N}\right)$ where N is the number of grid points in each dimension (currently 200) and i,j are integers between 0 and $N-1$. v is (pre-)computed on the points $\left(2\pi\frac{i-\frac{1}{2}}{N},2\pi\frac{j-\frac{1}{2}}{N}\right)$. This allows implementing the right-hand-side of Eq. (3) in a conservative way, i.e. the spatial integral the right-hand-side vanishes (or alternatively, the spatial integral of c remains constant in time, as it should).

The generation of the velocity field (and the shifted spatial mesh) is done by `generate_v_field`. The script `run_collect_save` runs many simulations and collects them in a `hdf5` file. The file produced by `run_collect_save` have this format:

- The name of the file is the value of η.
- `t` is the list of times for which the solutions are calculated (uniformly spaced).
- `x` and `y` are matrices of shape $N\times N$ with the `x` and `y` coordinates of the finite-differences grid for `c`.

- Each file contains 50 realizations, which are at groups `/001`, `/002`, ..., `/050`:
- `/XXX/c` is an `ndarray` of shape `[n,n,len(t)]` that contains the concentration field. `c[i,j,k]` is the value of the concentration field at position `x[i,j]` and `y[i,j]` at time `t[k]`. In other words, `c[...,i]` is the snapshot of the concentration field at time `t[i]`.
- The group `/XXX` also contains `/XXX/u`,`/XXX/v`, which are the `x` and `y` components of the velocity field. Note that `u` and `v` are not evaluated at the same grid points as c since we use a staggered scheme to ensure conservation. The functional form of `u` and `v` is given in both MATLAB and Mathematica syntax in the attributes of `/XXX`.