Exercises ToolMeeting

Pau Rué & A. J. Pons

I. 1D DIFFUSION EQUATION

As a first example of the things we have learned in the previous part of the ToolMeeting we are going to solve the Diffusion Equation in one dimension using periodic boundary conditions:

$$\frac{d\psi(\mathbf{x},t)}{dt} = D\nabla^2\psi(\mathbf{x},t) \tag{1}$$

To solve this equation we are going to use Euler's evolution scheme and one of the properties of the Fourier Transform.

A. Euler scheme

First order ordinary differential equations

$$\frac{d\psi(\mathbf{x},t)}{dt} = N(\mathbf{x},n) \tag{2}$$

may be solved using Euler's evolution scheme:

$$\psi_i^{n+1} = \psi_i^n + \delta t N(x_q, n) \tag{3}$$

where $\psi_i^n = \psi(x = i\delta x, t = n\delta t)$.

B. Fourier transforms

Fourier Transform is defined as follows

$$\hat{\psi}(\mathbf{k}) = (\mathcal{F}\psi)(\mathbf{k}) = \int_{-\infty}^{\infty} \psi(\mathbf{x}) \exp^{-i\mathbf{x}\cdot\mathbf{k}} d\mathbf{k}, \tag{4}$$

and the Inverse Fourier Transform as

$$\psi(\mathbf{x}) = (\mathcal{F}^{-1}\hat{\psi})(\mathbf{x}) = \int_{-\infty}^{\infty} \hat{\psi}(\mathbf{k}) \exp^{i\mathbf{x}\cdot\mathbf{k}} d\mathbf{x}. \tag{5}$$

Therefore, derivatives in real space may be transformed to products in Fourier space:

$$\mathcal{F}(\nabla\psi)(\mathbf{k}) = -i\mathbf{k}\mathcal{F}(\psi)(\mathbf{k}). \tag{6}$$

As a consequence, the derivative term in Eq. (1), may be written as

$$\nabla^2\psi(\mathbf{x},t) = \mathcal{F}^{-1}\left\{(-i\mathbf{k})^2(\mathcal{F}\psi)\right\}(\mathbf{x},t). \tag{7}$$

So, to solve Eq. (1) numerically, we use the following iteration scheme:

$$\psi_i^{n+1} = \psi_i^n + \delta t D\mathcal{F}^{-1}\left\{(-i\mathbf{k})^2(\mathcal{F}\psi^i)\right\}_i^n \tag{8}$$

C. Python code

To solve the equation with Python we will use the following class and functions defined in the scripts **Parameters.py** and **Functions.py**, respectively:

- Parameters class. It will contain all the parameters to be given to functions.
- fft_1D(Data), ifft_1D(Data), Freq_1D(n,dx) for the Fourier commands
- Euler_Step(psi,p,Change_t=True) for the Euler method
- Plot_1D(psi,p,Add_Gaussian=True) to plot the results

II. EXERCISE:

Using previous conceps, solve the following equation:

$$\frac{d\psi(\mathbf{x},t)}{dt} = -v\nabla\psi(\mathbf{x},t) \tag{9}$$

III. 2D SWIFT-HOHENBERG

Swift-Hohenberg equation [1] [2]

$$\frac{d\psi(\mathbf{x},t)}{dt} = \epsilon\psi(\mathbf{x},t) - (\nabla^2+1)^2\psi(\mathbf{x},t) + g\psi^2(\mathbf{x},t) - \psi^3(\mathbf{x},t) = N(\mathbf{x},t) \tag{10}$$

may produce stripes, hexagons, quasi-crystal patterns, superlatices, etc. This equation has been used extensively to study instabilities (e.g. Eckhaus or Zigzag instabilities) produced in pattern forming systems.

A. Semi-implicit method

To solve this equation we are going to use a semi-implicit method. We may split the right-hand side of Eq. (10) in its linear and nonlinear part:

$$\frac{d\psi(\mathbf{x},t)}{dt} = L[\psi(\mathbf{x},t)] + N[\psi(\mathbf{x},t)] \tag{11}$$

First, we consider the backwards derivative approximation of the linear part:

$$\bar{\psi}_{i,j}^{n+1} = \psi_{i,j}^{n} + \delta t L(x_{p,q}, n+1) \tag{12}$$

and from this we get $\bar{\psi}_{i,j}^{n+1}$ in terms of $\psi_{i,j}^{n}$ and some combinations of Fourier opeartions. We substitute the $\bar{\psi}_{i,j}^{n+1}$ in the nonlinear part of the right-hand side of Eq. (10) with an error of $o(\delta t^2)$. So, we have

$$\psi_{i,j}^{n+1} = \bar{\psi}_{i,j}^{n+1} + N[\bar{\psi}_{i,j}^{n+1}]. \tag{13}$$

B. Python code

Using the class **Parameters** and the following functions (defined in **Functions.py**)

- IC_2D(N,kx,xs) to get the initial conditions
- fft_2D(psi), ifft_2D(Data), Freq_2D(nx,ny,dx) for the Fourier tranasform operations in 2D
- Euler_Step(psi,p,Change_t=True)

- Plot_2D(psi,p) to plot results

solve the equation. Once you have implemented the solution check that you get the behaviour shown in Fig. 1.

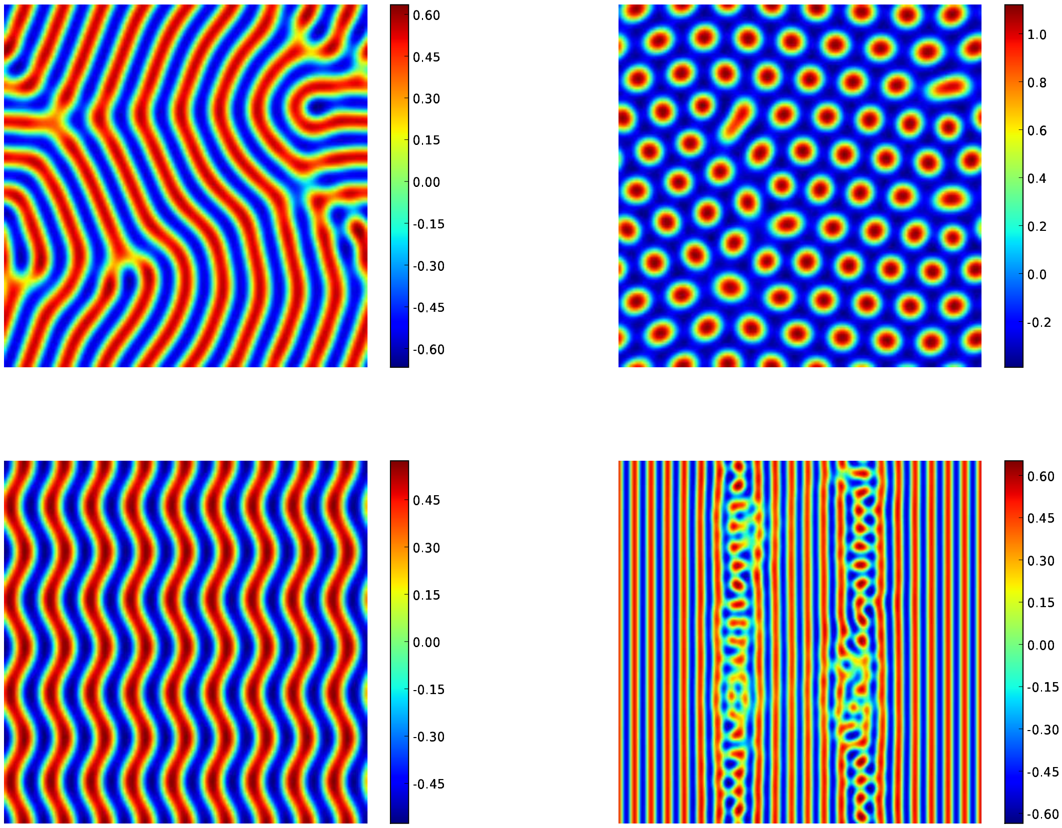

FIG. 1: Simulation results a) Stripes ($\epsilon = 0.3, g = 0.0$, random initial conditions). b) Hexagons ($\epsilon = 0.1, g = 1.0$, random initial conditions). c) Zigzag instability ($\epsilon = 0.3, g = 0.0$, initial conditions=sine, kx=0.88357 + random perturbation). d) Eckhaus instability ($\epsilon = 0.3, g = 0.0$), initial conditions=sine, kx=1.178 + random perturbation).

[1] J. Swift and P. C. Hohenberg. Hydrodynamic fluctuations at the convective instability. *Phys. Rev. A*, 15(1):319–328, Jan 1977.

[2] Michael Cross. Pattern formation in the swift-hohenberg equation. Website, 2008. `http://www.cmp.caltech.edu/~mcc/Patterns/SwiftHohenberg.html`.