Numerical solutions of the Swift-Hohenberg equation

Gonzalo Ferrandez Quinto

March 2024

1 Introduction

In this manuscript I will be solving the Swift-Hohenberg equation with the form:

$$\frac{\partial \Psi}{\partial t} = \epsilon - (1+\Delta)^2\Psi + g\Psi^2 - \Psi^3 \tag{1}$$

This equation was first considered for instabilities in the Raleigh convection giving rise to pattern formation in fluids, explored in [Swift and Hohenberg, 1977]. As discussed by [Sánchez Pérez-Moreno et al., 2013], this equation gives rise to complex behaviour that could be found in the more complex Navier-Stokes equation. The physical interpretation of the equation for Ψ, is the vertical fluid flow (out of the page) at the center of two heated plates. As described by [Sánchez Pérez-Moreno et al., 2013], the parameter ϵ, represents the reduced Rayleigh number and g the strength of the symmetry break in the equation.

2 Numerical Method

In this equation, the domain will be expanded for x and y, with periodic boundary conditions on x and no-slipping Dirichlet boundary condition on y. With:

$$0 \leq x \leq L \quad 0 \leq y \leq L \quad \Psi(x, y=0, L) = 0 \tag{2}$$

with:

$$\begin{cases} \partial_y\Psi - \Psi_y + \tau_1 U_{M-1}(y) = 0 \\ \partial_y\Psi_y - \Psi_{yy} + \tau_2 U_{M-1}(y) = 0 \\ \partial_y\Psi_{yy} - \Psi_{yyy} + \tau_3 U_{M-1}(y) = 0 \\ \frac{\partial\Psi}{\partial t} + (1-\epsilon)\Psi + 2\Psi_{yy} + 2\partial_x^2\Psi + 2\partial_x^2\Psi_{yy} + \partial_x^4\Psi + \partial_y\Psi_{yyy} + \tau_4 U_{M-1}(y) = g\Psi^2 - \Psi^3 \end{cases}$$

Which corresponds to generalized τ method of the first order formulation to satisfy the boundary conditions in y, as described by [Burns et al., 2020]. This

system will be solved using a pseudo-spectral method using Fourier series and Chebyshev polynomials. There is no need to split the partial derivatives on x as this will be expanded using a Fourier series, therefore derivatives are simply calculated.

Then the Galerkin method is used to discretize each one of the variables, in this case each one of the Ψ and thee three first partials of y and the four τ's.

This have the form of:

$$\Psi = \sum_{n,m}^{N} a_{n,m}(t)cos(\frac{2\pi}{L}nx)T_m(my) - \sum_{n,m}^{N} b_{n,m}(t)sin(\frac{2\pi}{L}x)T_m(my) \tag{3}$$

$$\tau_1 = \sum_{n}^{N} \tau_{a_1,n}(t)T_m(my) - \sum_{n}^{N} \tau_{b_1,n}(t)(T_m(my)) \tag{4}$$

Therefore we will need to keep track of the a's and b's for the Ψ vectors and its derivatives. Also one τ_a and τ_b for each of the 4 τ's.

Using this projections, it gives rise to the system:

$$\mathrm{L} = \begin{pmatrix} D & Z & -C & Z & Z & Z & Z & Z & 0...0 \\ Z & D & Z & -C & Z & Z & Z & Z & 0...0 \\ Z & Z & D & Z & -C & Z & Z & Z & 0...0 \\ Z & Z & Z & D & Z & -C & Z & Z & 0...0 \\ Z & Z & Z & Z & D & Z & -C & Z & 0...0 \\ Z & Z & Z & Z & Z & D & Z & -C & 0...0 \\ W & Z & 2D & Z & -2Cn^2 & Z & D & Z & 0...0 \\ Z & W & Z & 2D & Z & -2Cn^2 & Z & D & 0...0 \\ 0 & . & . & . & . & . & . & . & 0...0 \\ . & . & . & . & . & . & . & . & 0...0 \\ 0 & . & . & . & . & . & . & . & 0...0 \end{pmatrix} \tag{5}$$

$\mathrm{W} = -2Cn^2 + Cn^4 + (1-\epsilon)C$

Where T_m are the translated Chebyshev polynomials of the first kind with $\frac{L}{2}(T_m(my)) + \frac{L}{2}$. Here D is a derivative matrix for Chebyshev polynomials of the second kind, C is a conversion matrix from Chebyshev polynomials of the first kind to second kind an n is a vector with the Fourier wave numbers $\frac{2\pi n_j}{L}$. Z is a matrix of zeros. Each one of the matrices has size $N \times N$ and the vector $N \times 1$. In addition, at the right of the matrix there are 8 columns of zeros, with a 1 in the $(i^{th}(N) - 1, j^{th},)$ entry for each one of the τ's coefficients. On the bottom, to force Dirichlet (No slipping) boundary conditions, there is going to be a row of ones, in the form $(i^{th}, 0 : N) = (-1)^{j^{th}}$ and $(i^{th} + 1, 0 : N) = 1$ for each one of the 4 τ's coefficients corresponding columns.

Then we have:

$$M = \left(\begin{array}{cc|c} \multicolumn{2}{c|}{\mathbf{0}} & \mathbf{0} \\ \hline C & Z & \multirow{2}{*}{$\mathbf{0}$} \\ Z & C & \end{array}\right)$$

(6)

$$X = \big(a, b, a', b', a'', b'', a''', b''', \tau_{(a_1,n)}, \tau_{(b_1,n)}, ..., \tau_{(a_4,n)}, \tau_{(b_4,n)}\big)$$

(7)

With size $(N+8)\times(N+8)$ for M, $N\times 1$ for the a's and b's and one single entry for the τ's

Then: $M\partial_t X + LX = F$. With F , vector of size $(8N+8)\times 1$, being all zeros except for the the rows corresponding to a''' and b'''. Which will be equal to the projection of the right hand side of the 4^{th} equation of 2. This will be done in grid space to ensure a faster convergence of the solution. A scale of order 2 will be used to prevent aliasing on the non-linearity.

3 Resolution study

In this section, there will be a resolution study to ensure the exponential convergence of the algorithm. Instead of using an analytic solution, I will use a high resolution solution ($N = 512$) as the true solution and compare it to lower order values of N ranging from $(16, 256)$ in multiples of 2. For this case, I will be using uniform initial conditions as in later simulations to show the behaviour of equation I will be uniformly randomly distributed initial conditions. In this case $L = 50$ with a time step of 10^{-3}. In Fig. 1, we can see the resolution study and how the system converges exponentially until $N = 128$ where the error stagnates. In addition for the Δt value, $\Delta X = L/N$ with minimum value of 0.1 when $N = 512$. Then $\Delta t \leq \frac{\Delta x}{32} \approx 0.003$ as seen in [wen Xi et al., 1991] for stability.

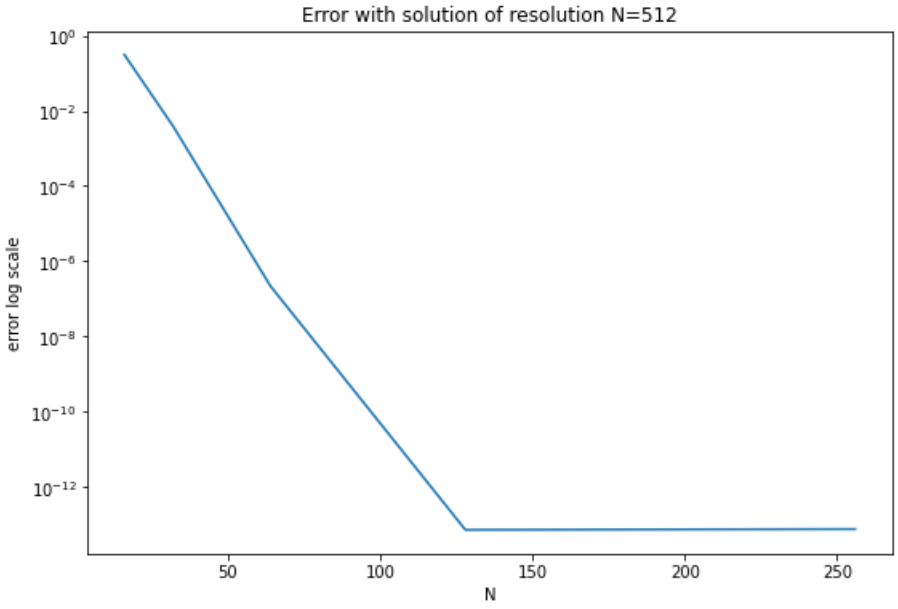

Figure 1: This figure represents the resolution study for the system with uniform initial conditions, $\Delta t = 10^{-3}$, $L = 5$. The tested N's ranged from $(16, 256)$ in multiples of 2, The y-axis is in log scale, we can see how is linear from (16,128) showing that the error is exponentially decaying. There is no much benefit in increasing N to reduce the error after 128. Error corresponds for 1000 iterations.

4 Simulations

In this section, I will be showing example simulations for the equation with $L = 50$, $N = 256$ and $\Delta t = 5 \times 10^{-3}$. I will be varying the values of ϵ and g. For this simulation the code in **SwiftPDE.ipynb** is used. This code saves the time-step graphs in the folder where the code is run as images. The code also allows to save the data as a GIF video.

4.1 Case with constant ϵ

The initial conditions will be uniform randomly distributed with a constant seed in the range $(0, 0.3)$ for this subsection. This would mean an initial general random upward flow.

In Fig. 2, we can see the case when $\epsilon = 1$ and $g = 0.1$. In this case linear formations take longer to progress and there is a larger ensemble of spots in intermediate steps.

In contrast, in Fig. 3, we can see the case when $\epsilon = 1$ and $g = 1$. Here linear formations in the grid are favoured after 10 seconds of simulation. This

exemplifies how the parameter g determines the braking of symmetry.

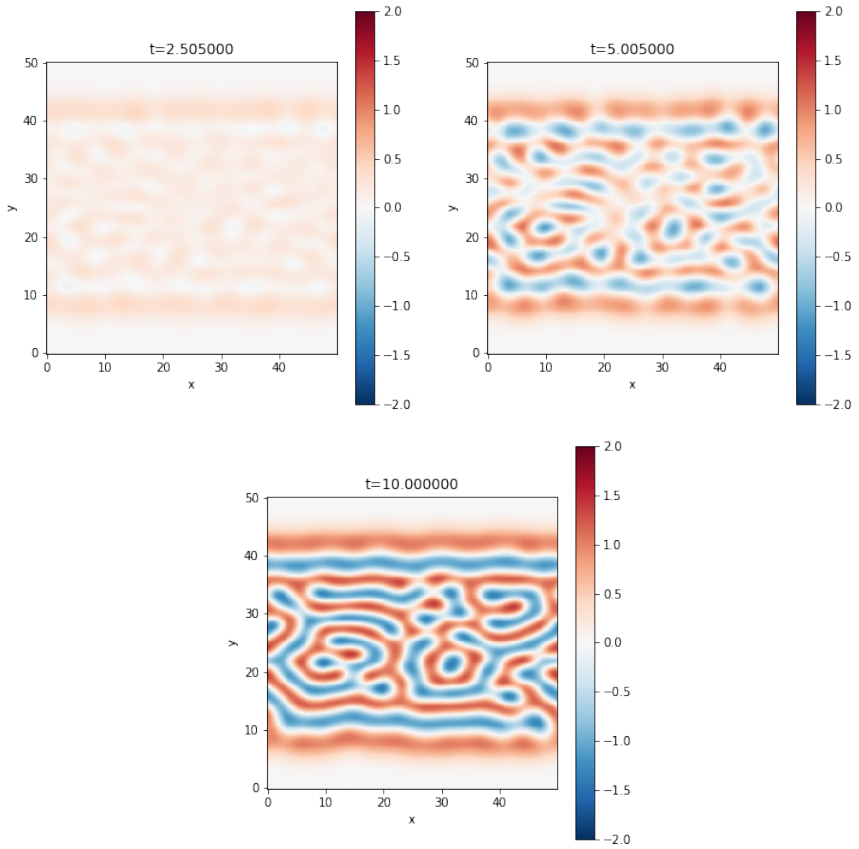

Figure 2: In this test, $g = 0.1$ and $\epsilon = 1$ with random initial conditions. It is noticeable that with $g = 0.1$, the formation of linear patterns and spots are mixed. With a slower progression of the line formation. The simulation is run for 2000 iterations.

4.2 Increasing the value of ϵ

The initial conditions will be uniform randomly distributed with a constant seed in the range $(-2.5, 2.5)$ for this subsection. This would mean an initial general non-dominant flow in any direction.

In Fig. 4 we can see the plots of this scenario with $g = 2$ and $\epsilon = 2$. It can be seen how the upward convection starts to dominate until eventually downward flow will disappear.

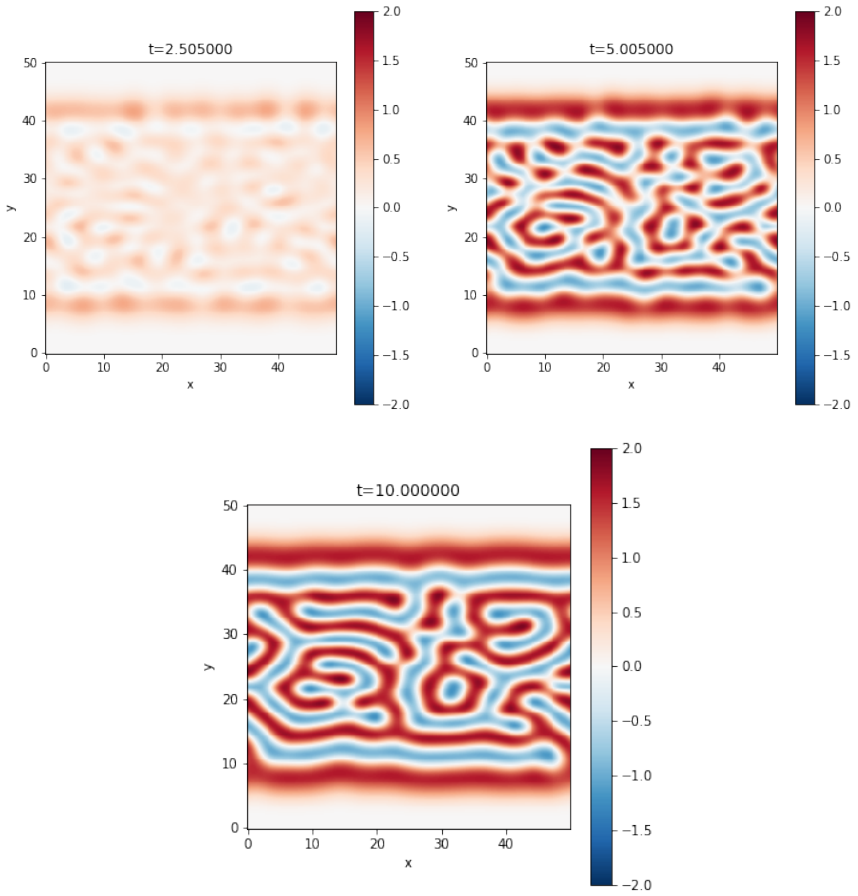

Figure 3: In this test, $g = 1$ and $\epsilon = 1$ with random initial conditions. It is noticeable that with $g = 1$, the formation of linear patterns is reinforced. The simulation is run for 2000 iterations.

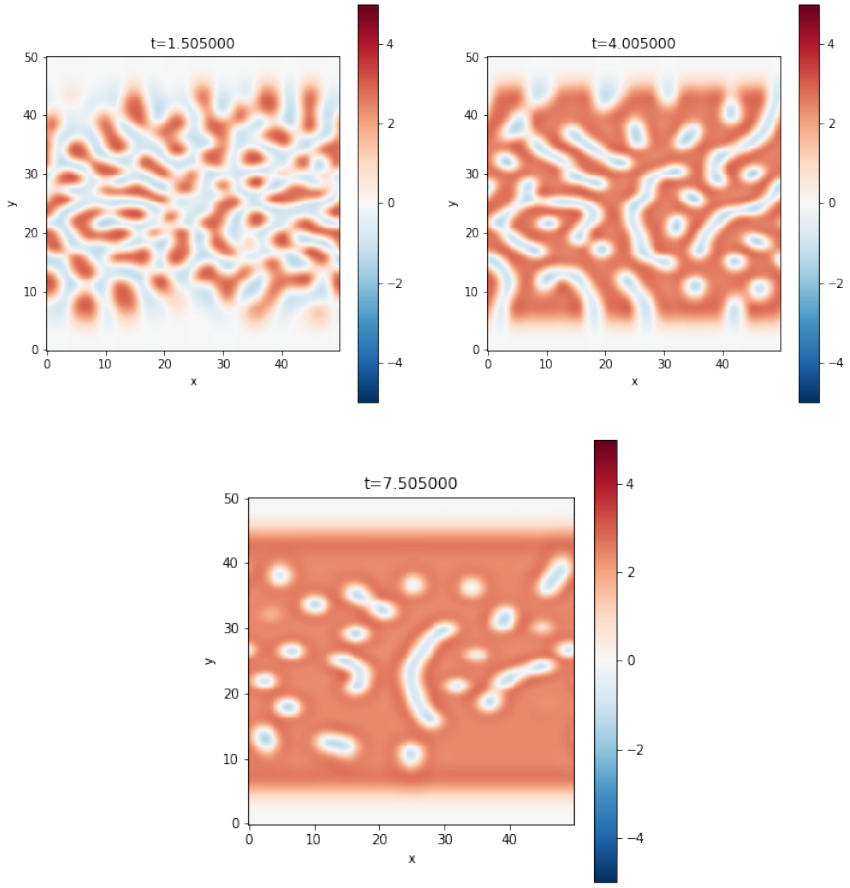

Figure 4: In this test, $g = 2$ and $\epsilon = 2$ with random initial conditions. It is noticeable that by increasing ϵ, the convection parameter starts to dominate over the pattern formation and stripes start to disappear. The simulation is run for 1500 iterations.

References

[Burns et al., 2020] Burns, K. J., Vasil, G. M., Oishi, J. S., Lecoanet, D., and Brown, B. P. (2020). Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, 2(2):023068.

[Sánchez Pérez-Moreno et al., 2013] Sánchez Pérez-Moreno, S., Ruiz Chavarría, S., and Ruiz Chavarría, G. (2013). Numerical solution of the swift–hohenberg equation. In *Experimental and computational fluid mechanics*, pages 409–416. Springer.

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

[wen Xi et al., 1991] wen Xi, H., Viñals, J., and Gunton, J. (1991). Numerical solution of the swift-hohenberg equation in two dimensions. *Physica A: Statistical Mechanics and its Applications*, 177(1):356–365.