

Numerically Solving Advection-Diffusion-Reaction Equations

Carl Denton

September 18, 2016

1 Introduction

Advection-Diffusion-Reaction (ADR) Equations are partial differential equations (PDEs) of the form

$$\frac{\partial}{\partial t}\varphi + U\nabla \cdot \varphi = D\nabla^2 \cdot \phi + R(\varphi)$$

That is, they are characterized by the presence of advection: $U\nabla \cdot \varphi$; diffusion: $D\nabla^2 \cdot \varphi$; and arbitrary reaction: $R(\varphi)$. They encompass a wide range of phenomena including diffusion (for example of heat or fluid), atmospheric transport, various chemical phenomena, bacterial growth, and more. Thus it is desirable to simulate their solutions and we do so here.

Code for the following can be found at <https://github.com/cfdenton/ac274>

2 Methods

2.1 Numerical Differentiation

We can compute numerical derivatives by the method of finite differences. By computing Taylor expansions we observe that

$$\varphi'(x) \approx \frac{\varphi(x+dx) - \varphi(x)}{dx} \approx \frac{\varphi(x+dx) - \varphi(x-dx)}{2dx}$$

for some small dx , where the first expression is accurate to first order in dx and the second is accurate to second order. Similarly, we get that

$$\varphi''(x) \approx \frac{\varphi(x+dx) - 2\varphi(x) + \varphi(x-dx)}{dx^2}.$$

2.2 Transport Matrix

Exploiting the fact that the numerical derivatives given above are linear operators in φ , and assuming that $R(\varphi)$ is also linear in φ , we can write an update of the ADR equation as

$$\varphi^{n+1} = L\varphi^n$$

where φ^t is the state φ at some time t and L is a linear operator encompassing the differentiation operations as well as the coefficients of the ADR equation. If φ^t is represented as a vector of N dimensions, then L is a $N \times N$ tridiagonal matrix (in the one-dimensional case). In the two-dimensional case, L has 5 diagonals. This property can be used to derive useful stability conditions for the simulations.

3 Experiments

We use python (with the NumPy package) for the numerics, and we use GnuPlot and the VisPy package for visualizing the results. By the above analysis, we can execute the simulation by constructing an initial state and a transport matrix, and obtain subsequent states by iteratively multiplying the transport matrix by the state at any given time.

3.1 Single Dimension

3.1.1 Transport Matrix

We have the following:

$$\frac{\varphi_j^{n+1} - \varphi_j^n}{dt} = -U \frac{\varphi_{j+1}^n - \varphi_{j-1}^n}{2dx} + D \frac{\varphi_{j+1}^n - 2\varphi_j^n + \varphi_{j-1}^n}{dx^2} + R(\varphi)$$

where we say that $\varphi_{j+1} = \varphi(x+dx)$ and we choose dx and dt to be some sufficiently small numbers. Under this representation, assuming that $R(\varphi_j^n) =$

$R\varphi_j^n$ we get that L is a tridiagonal matrix $\{a, c, b\}$ where

$$a = dt \left(\frac{D}{dx^2} - \frac{U}{2dx} \right)$$

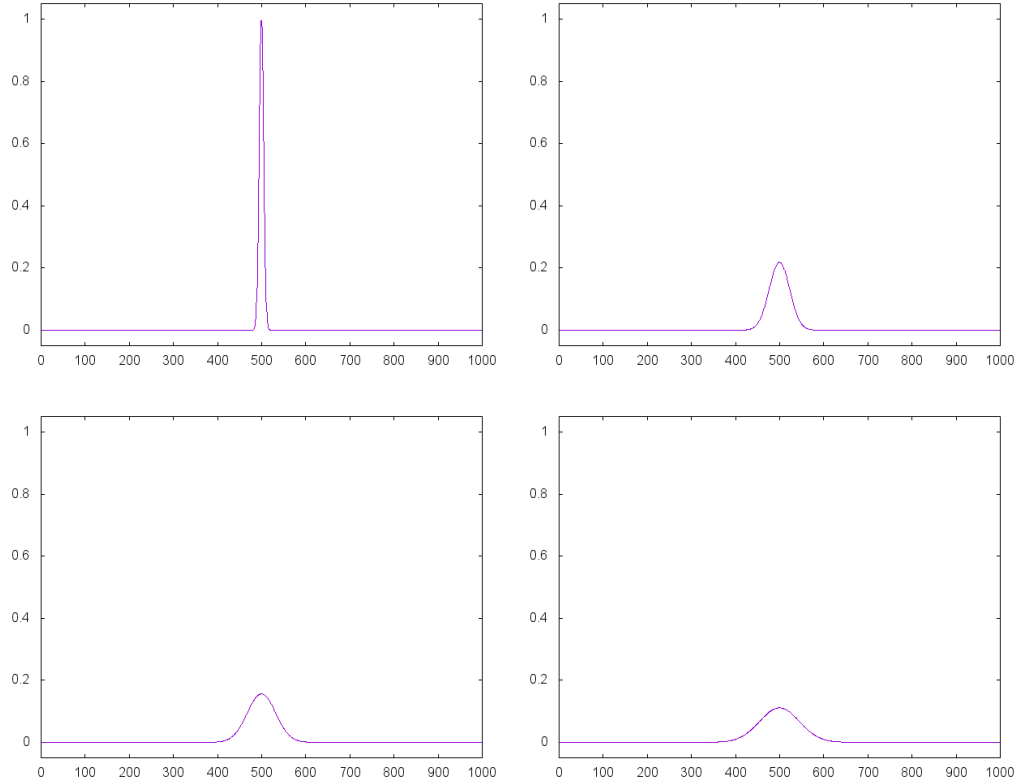
$$b = dt \left(\frac{D}{dx^2} + \frac{U}{2dx} \right)$$

and

$$c = 1 - 2dt \frac{D}{dx^2} + R$$

3.1.2 Diffusion

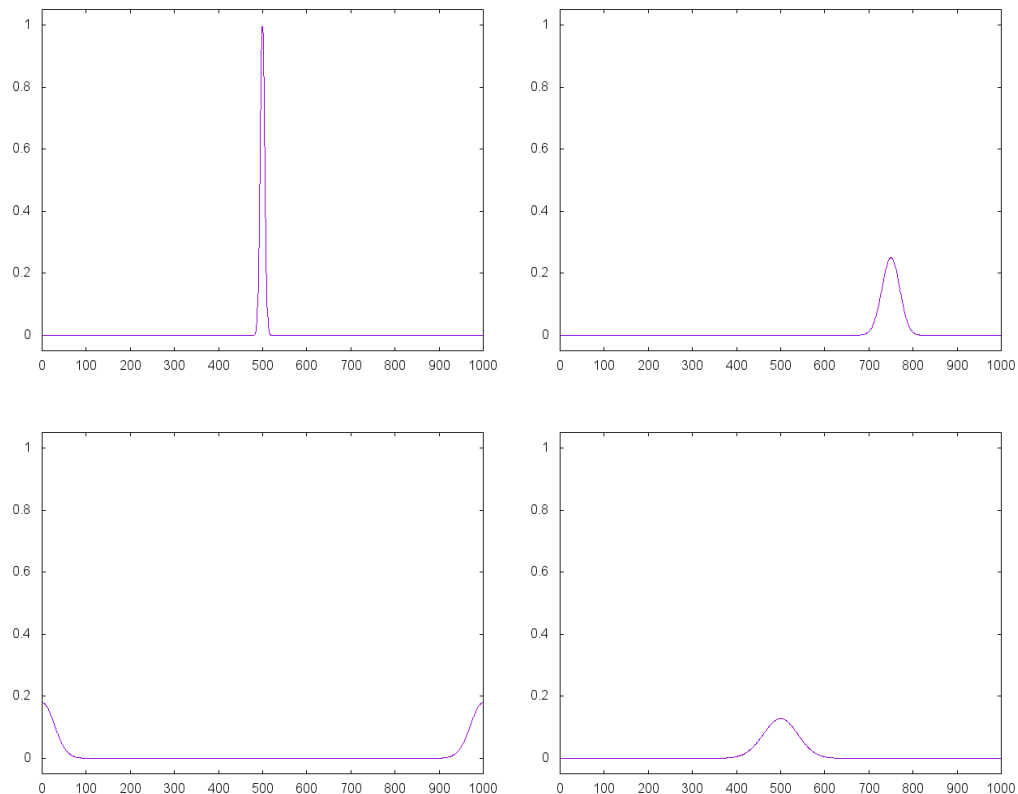
We set our initial condition to be a tall gaussian in the middle of the mesh. The results are plotted below at time steps 0, 500, 1000, and 2000 with a grid size of 1000:



These correspond well with our intuition for a diffusive process— we will do a more formal validation of the results in section 4.

3.1.3 Diffusion+Advection

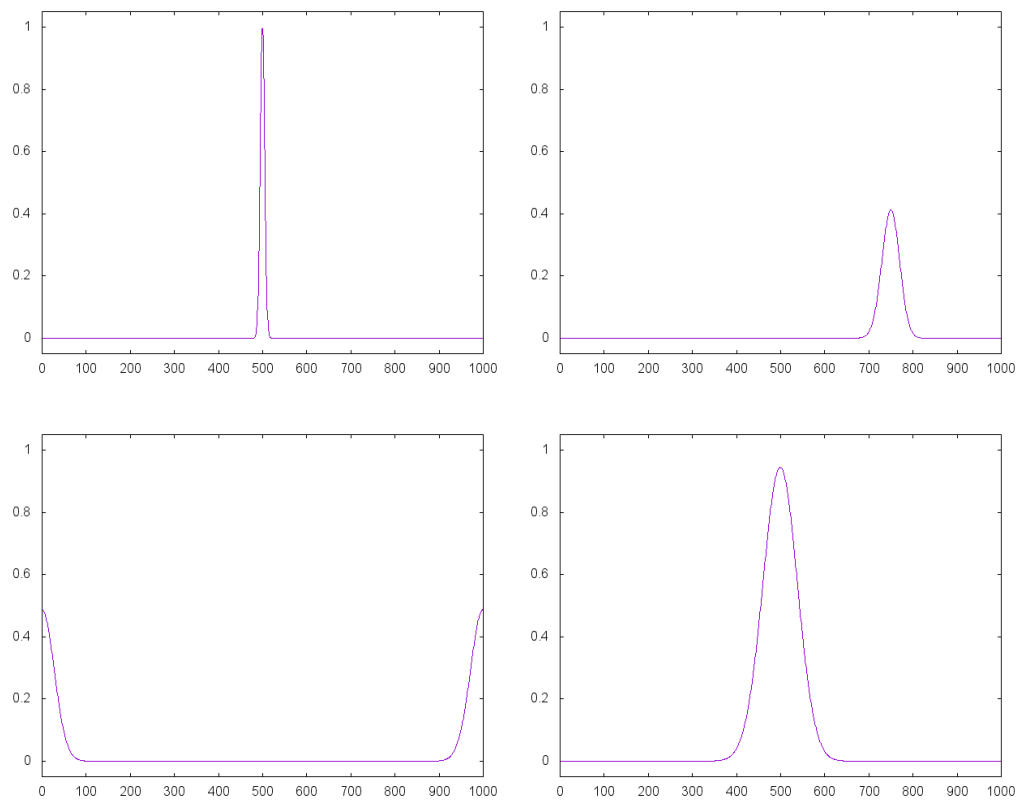
We can repeat the above experiment while adding advection by adding a nonzero “ U ” term. The results are as follows, again at time steps 0, 500, 1000, and 2000:



Note that since we have set periodic boundary conditions, the wave reappears on the left side of the screen after leaving the right side.

3.1.4 Diffusion+Advection+Reaction

Finally we add a small amount of positive reaction and obtain the following (at $t = 0, 500, 1000, 2000$):

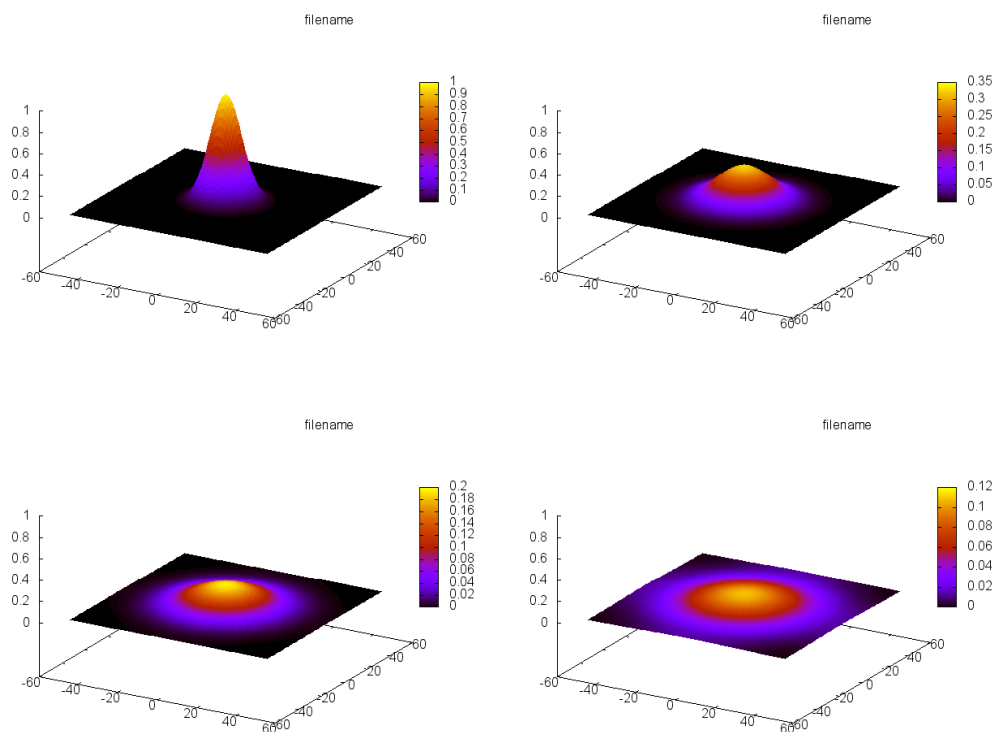


3.2 Two Dimensions

We repeat the experiments in two dimensions:

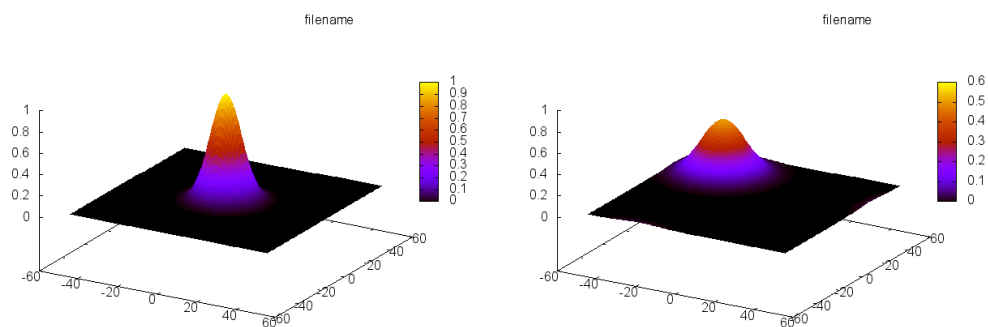
3.2.1 Diffusion

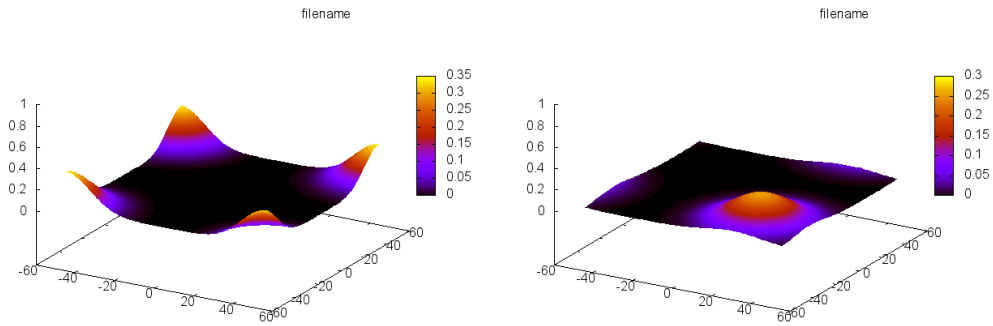
With $t = 0, 500, 1000, 2000$:



3.2.2 Diffusion+Advection

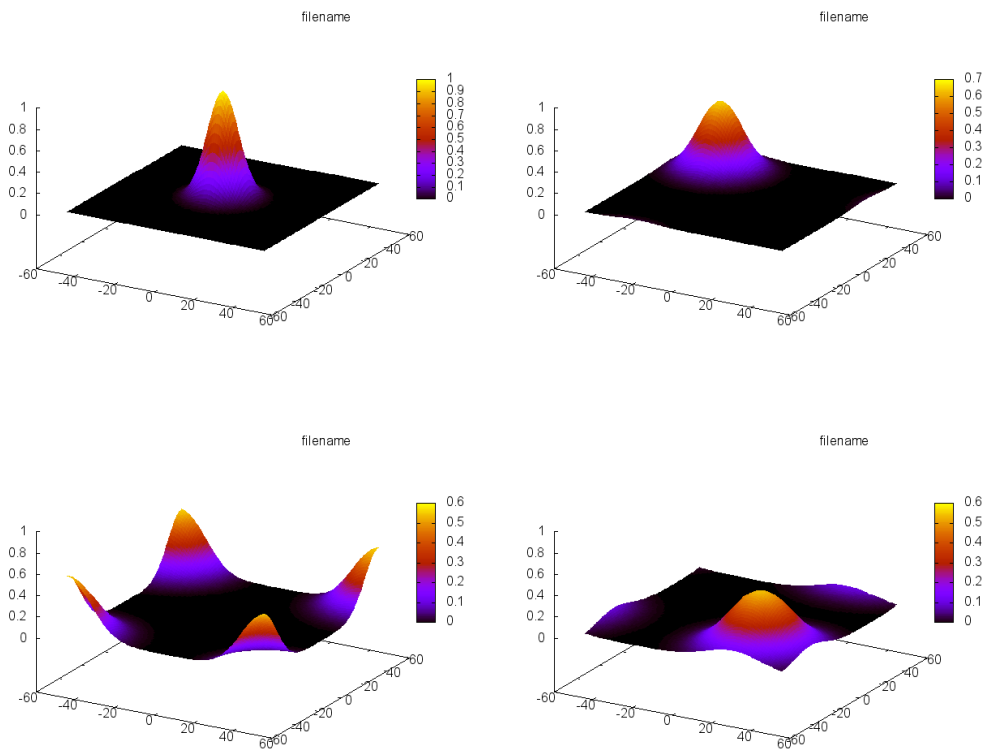
With $t = 0, 250, 500, 750$:





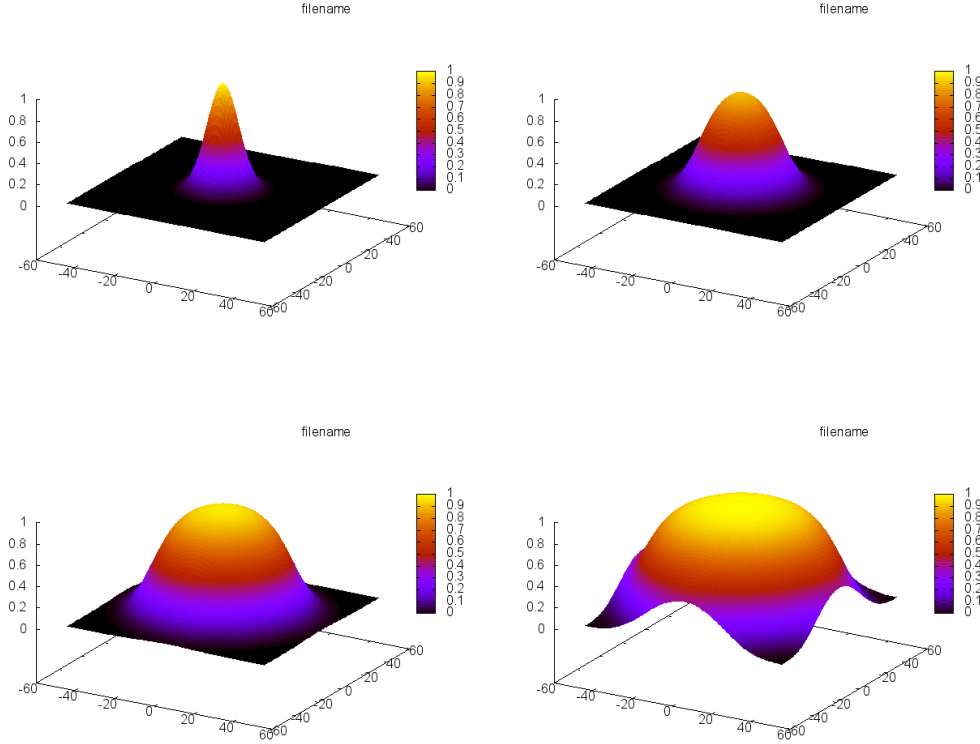
3.2.3 Diffusion+Advection+Reaction

With $t = 0, 250, 500, 750$:



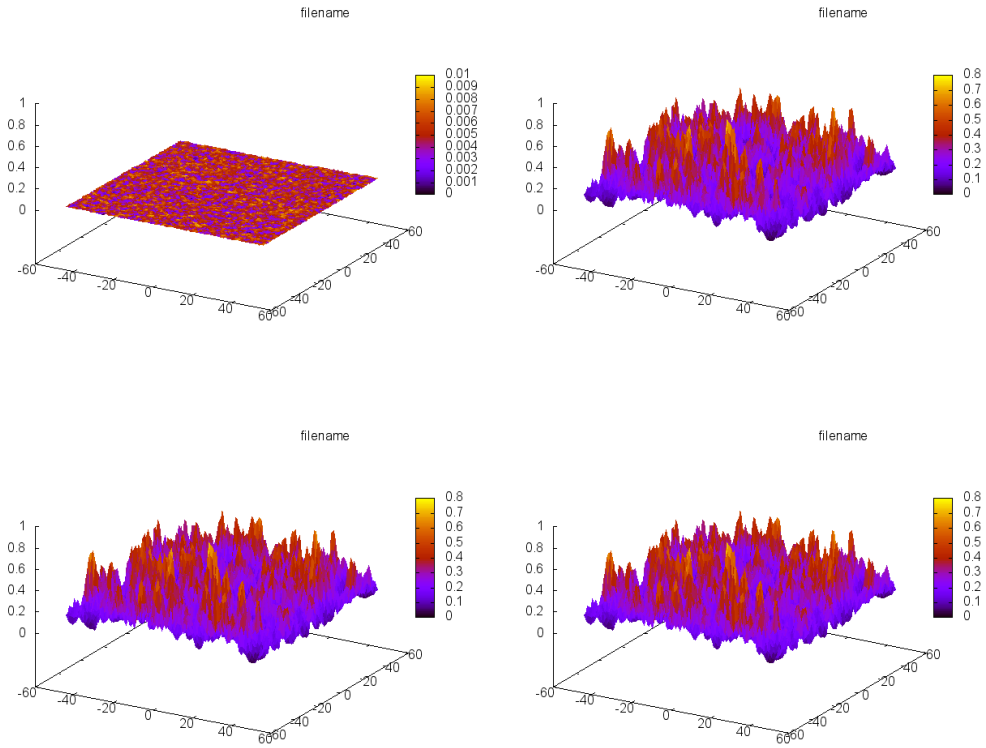
3.2.4 Nonlinear Chemistry

We also implement a few types of non-linear chemistry. This can be accomplished by adding the necessary term to the state after the transport matrix has been applied. First we model logistic growth as $R(\varphi) = R\varphi(1 - \varphi)$:

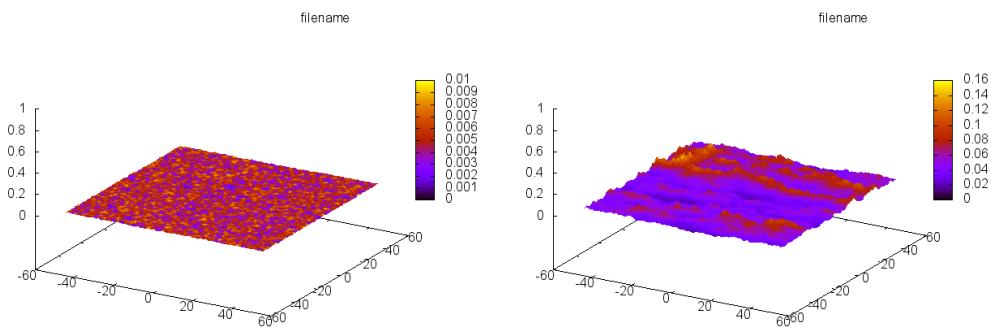


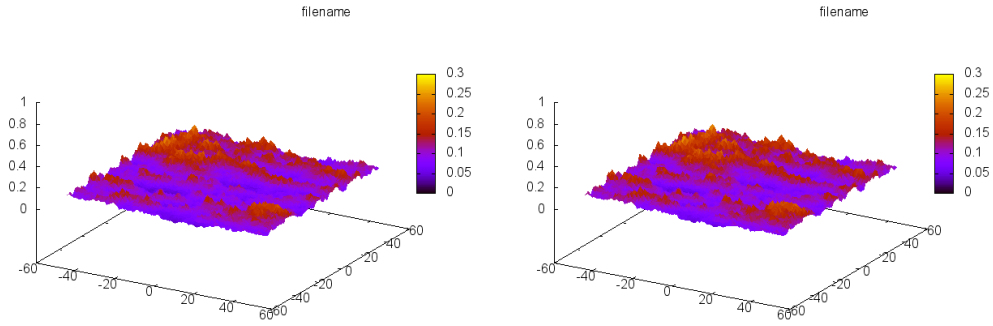
As expected, the concentration plateaus at $\varphi = 1$. Following the demonstration in class, we also model chemistry of the form $R(\varphi) = a(x, y)\varphi - b\varphi^2$, where $a(x, y)$ is a randomly varying function of the position. The initial configuration is also a small random variation, and as a result we obtain a

highly random final distribution:



We also manage to replicate the types of striations found in the work from class:





4 Analysis

4.1 Comparison with Analytic Solutions

We plot the mass, average position, variance, and entropy of the system over time to validate our simulation. In the one dimensional case, we get:

Figure 1: Mass and Average Position

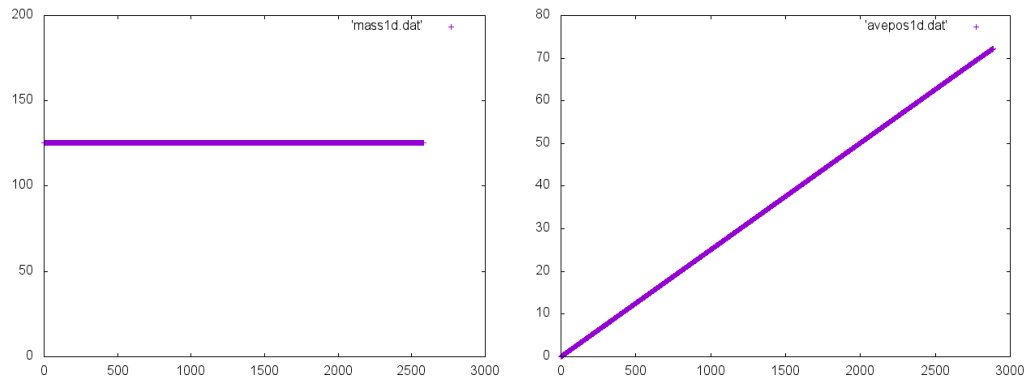
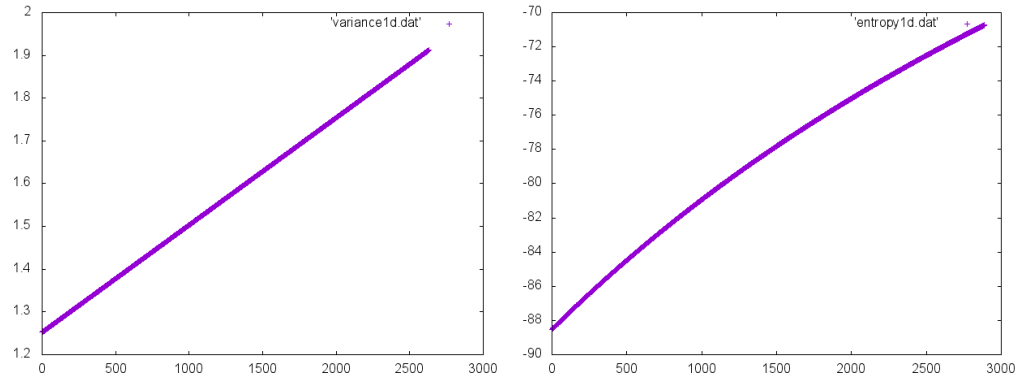


Figure 2: Variance and Entropy



These correspond well with the predictions of analytic solutions. Analogues hold in the two dimensional case.