Physics 141: Final Project: Written Report
Turing Patterns and Reaction Diffusion Equation
Katya Sumwalt

Description of Problem:

Alan Turing identified the reaction-diffusion system to help describe how the interplay between chemical interactions and diffusion can create instabilities that form spatial patterns. He found that a system containing "morphogens" that was stable without diffusion could be propelled into an unstable state by diffusion towards a new state where there exist stable, stationary or dynamic patterns. The system is subject to initially small perturbations that get amplified through diffusion through the medium and creates identifiable, stable patterns. Some motivating examples of this problem include the pattern formation in zebrafish, embryo development, and even patterns on shells (Turing, 1952).

In his paper, Turing characterizes the "breakdown of symmetry and homogeneity" (Turing, 1952) with an example of an embryo that starts with an approximate spherical symmetry. The case is made that certain small deviations from this spherical symmetry are propagated into instability until a new form of equilibrium is created without the prior symmetry. This is a proposed method for how asymmetries in development from embryos can come from specific deviations from an otherwise symmetric structure. This essentially all occurs through a basically homogenous, stable equilibrium becoming unstable due to the diffusion process and then in the final state a new inhomogeneous equilibrium is created (Turing, 1952).

He states that these systems can be incredibly complicated so it is important to focus on systems where the chemical and mechanical aspects can be almost entirely separated. Additionally, he proposes a set of chemical reactions that this mathematical system can be applied to. This project works to solve and understand one particular form of this problem that satisfies the conditions he layed out — the Gray Scott Model.

The particular model for this version of the the diffusion reaction was the gray scott model which defines the interaction between two species as:

$$U + 2V \rightarrow 3V$$

Source: Phys 141 Final Assignment

With the reaction kinetics:

$$f(u,v) = -uv^2 + F(1-u)$$

$$g(u,v) = uv^2 - (F+k)v$$

Source: Phys 141 Final Assignment

Where the first reaction kinetic equation corresponds to the adding of U and the second to the removal of the morphogen V. (from assignment description)

The reaction diffusion equation takes the form:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + f(u,v)$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + g(u,v)$$

Source: Phys 141 Final Assignment

Where the rate of change of each species is determined by a diffusion term and a non-linear term — the reaction kinetics.

Defining the problem within the context of code required guided decisions for the definition of the “kill rate” k and the feed rate “F”, defining the reaction kinetics in code, deciding on the the numerical methods to implement for both the time stepping and to define the laplacian- forward euler and finite differencing, defining the initial conditions, boundary conditions and finally visualizing the development of the pattern.

Description of Numerical Methods/Implementation:

I solved this problem first in one dimension then extended to in 2-Dimensions using the forward euler and finite differencing methods based on the implementation by Pierre Navaro, whose github is linked in the references. I present only the two dimensional results since they are more interesting to look at and represent essentially the same work. This section will be organized following order of the organization of the code for easier reference: Definition of reaction kinetics, central finite differencing, implementing boundary conditions, implementing initial conditions, and implementing forward euler in a for loop. I will then discuss stability and accuracy, as well as steps forward.

1. Definition of Reaction Kinetics

```
# Reaction kinetics as defined the in the assignment
def f(u, v, F):
    return -u * v**2 + F * (1 - u)
def g(u, v, F, k):
    return u * v**2 - (F + k) * v
```

2. Central Finite Differencing: I chose the central differencing to represent the laplacian in two dimensions since it is relatively easy to implement and accurate to a higher order than the other finite differencing methods. The implementation is as follows:

$$\Delta u(x,y) = \frac{u(x-h,y) + u(x,y-h) - 4u(x,y) + u(x+h,y) + u(x,y+h)}{h^2} + \mathcal{O}(h^2).$$

Image Source: janmr blog

```python
# Laplacian using finite differencing:
def laplacian_2D(x, dx=1):
    #x_i-1,j + x_i,j-1 - 4 x_i,j +x_i+1,j +x_i,j+1 /dx^2
    return (x[:-2, 1:-1] + x[1:-1, :-2] - 4 * x[1:-1, 1:-1] + x[2:, 1:-1] + x[1:-1, 2:]) / dx**2
```

3. Boundary Conditions: I decided to experiment with implementing no-flux and periodic boundary conditions.
4. Initializing the Perturbation: I initialized a circular perturbation in the center of the otherwise uniform field as suggested in the assignment.
5. Implementing forward euler: I then implemented forward euler in a for loop to contain the time stepping using the finite-difference version of the laplacian as well as the newly computed concentrations of the morphogens. The forward euler equation:

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Source: Brorson

Implemented with the 1D central finite differencing:

$$u_i^{n+1} = u_i^n + \Delta t \cdot D_u \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} + f(u, v)$$

Implemented in 2D:

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t \cdot D_u \left[\frac{u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n}{(\Delta x)^2} \right] + f(u_{i,j}^n, v_{i,j}^n)$$

Both the u and the v are calculated in the same way, just substitute the reaction kinetics equation.

Finally the way that I implemented it in my code was by running through a for loop of time steps and calculating the concentrations of u and v over the whole grid at each time step, plotting only one of the concentrations [V] at all of the grid points since the other concentration would be the inverse, displaying the same pattern. This was based on the method of implementation by Pierre Navarro. The code implementation is shown below:

```python
def gray_scott_finite_diff(U, V, D_u, D_v, F, k,dt,boundary): #used forward euler and made corrections f
    #forward Euler:
    #u_n+t = u_n + dt(D * laplacian + non-linear term)
    U[1:-1, 1:-1] = U[1:-1, 1:-1] + dt*(D_u * laplacian_2D(U) + f(U[1:-1, 1:-1] , V[1:-1, 1:-1], F))
    V[1:-1, 1:-1] = V[1:-1, 1:-1]+ dt*(D_v * laplacian_2D(V) + g(U[1:-1, 1:-1], V[1:-1, 1:-1], F, k))
    #apply the boundary conditions to relevant grid points
    boundary(U)
    boundary(V)
    return U, V
```

6. Discussing Stability: One condition that I needed to satisfy in order for the implementation of this code to be stable was a small enough step size, I just determined this step size through trial and error to get a step size of 1 being sufficiently small and also giving me a large enough time step that I did not have to create so many figures for the visualization.

7. Steps Forward: I started trying to employ the spectral method instead of the finite differencing but was having some issues with the stability of the solution. I would like to try to employ a semi-implicit method with the spectral method, and then hopefully extend into 3 dimensions.

Computational Results Obtained/ Visualizations:

Figure 1:Stationary Pattern
Du=0.2043, Dv= 0.0360,F= 0.0474, k = 0.0696

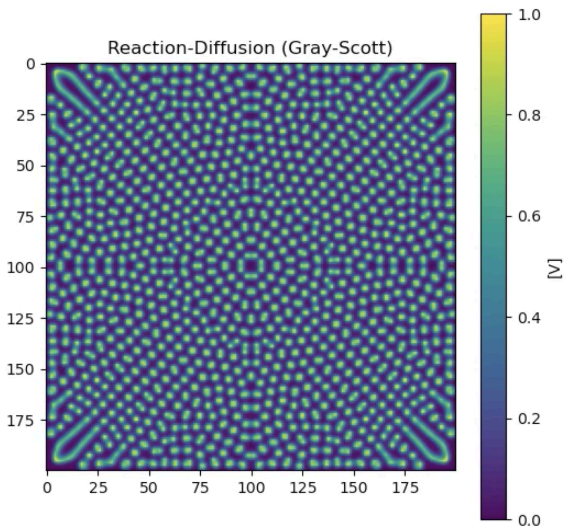
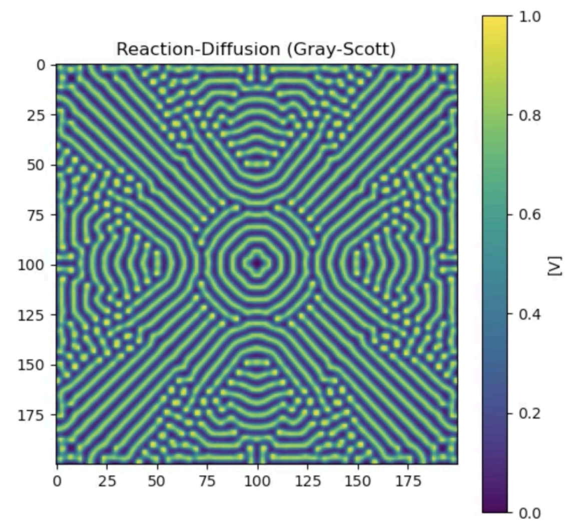
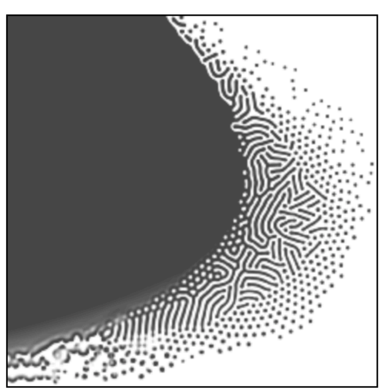


Figure 2: Stationary Pattern
Du=0.2043, Dv= 0.0360,F= 0.0540, k = 0.0616

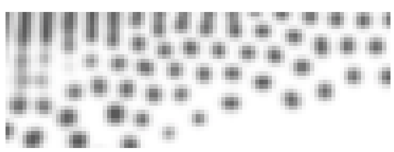


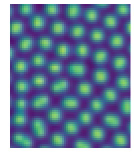
These figures illustrate the patterns selected from this grid. Where the x axis is F (0.045 → 0.07) and the y = k from 0.01 to 0.1. The image source and these values are by Karl Sims.

Reference 1: Diffusion-Reaction Map by Karl Sims

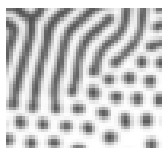


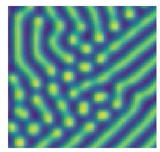
We can see that our values correspond well to the results established and illustrated by Reference 1. In the reference the values of F= 0.0474, k = 0.0696 roughly correspond to the results obtained illustrated in figure 1. This is illustrated below where the image on the left is the reference and on the right is the result:





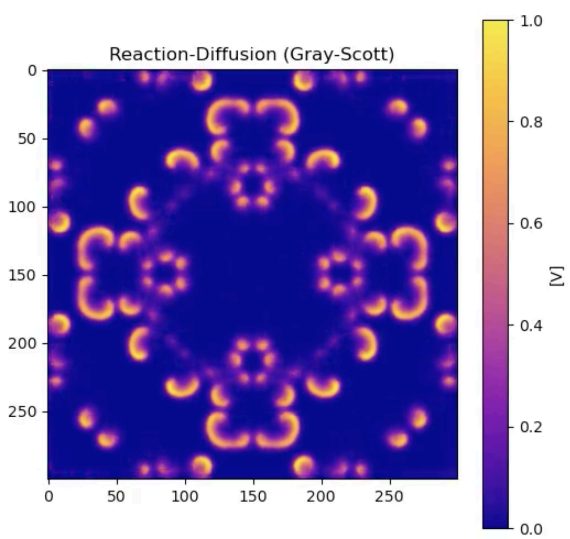
Similarly in Reference 1, the pattern that arises from the values of F= 0.0540, k = 0.0616 correspond to the image obtained in figure 2. This is illustrated below where the image on the left is the reference and on the right is the result:





The “labyrinth” pattern as it is called in the assignment begins to emerge in figure 2, while the figure 1 illustrates primarily spots. These patterns were both stationary and were subject to periodic boundary conditions.

Figure 3: Dynamic Pattern at Two Different Times
Du = 0.2097, Dv = 0.1050, F = 0.0082, k = 0.00451



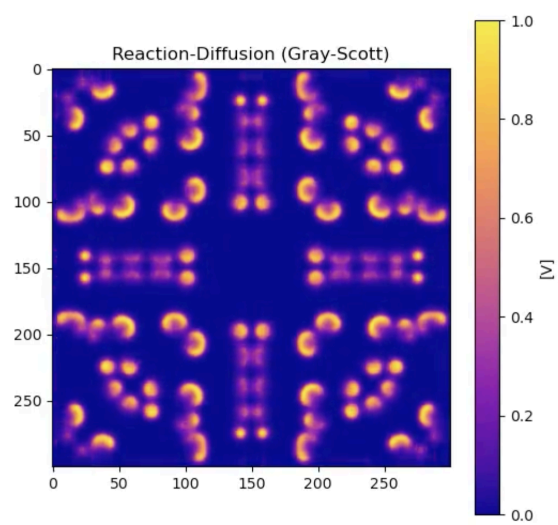
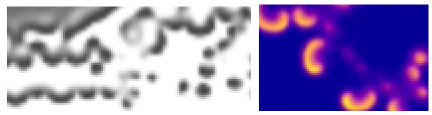


Figure 3 illustrates a pattern that was more dynamic than the patterns illustrated in Figure 2 and in Figure 1. I used a larger spatial domain and no flux boundary conditions. The no flux boundary conditions were really apparent in the way that the pattern would alternate from primarily going outwards to going inwards to outwards. These patterns correspond well to the lower left corner of patterns corresponding to values around F = 0.04 and k = 0.01, but the parameters that I implemented are quite different from the values illustrated in the reference. This is illustrated below where the image on the left is the reference and on the right is the result:



This may be due to the fact that my diffusion constants may be different as well as the fact that if we extended the reference to smaller values of F and k a similar pattern to the results shown in figure 3 may emerge.

Discussion of the results:

This project illustrates successful implementation of the central finite differencing method and euler methods to solve the diffusion reaction equation; the parameters and patterns used in my code match the reference patterns. However Figure 3: does illustrate some subtle differences from the reference patterns which may have been due to different diffusion coefficient values. One way of addressing this in the future is to identify a reference with displayed diffusion coefficients and use those as a reference for the diffusion coefficients that I would implment and cross examine the patterns resulting from solving the PDE.

Figure 1 and Figure 2 both illustrate stationary patterns that emerged and stabilized from a circular perturbation in the middle of the defined space. Figure 3 illustrates a dynamic pattern that emerges and changes throughout time with a circular perturbation in the middle of the defined space. These two types of categories of patterns are discussed in Turing's paper as a result of the different input coefficients.

The patterns in Figure 1 and Figure 2 were both calculated using periodic boundary conditions while the pattern in Figure 3 was calculated using a no-flux boundary condition. The no-flux boundary condition resulted in different behavior than the behavior illustrated by the periodic boundary conditions though they illustrated different stationary and dynamic pattern behaviors. In the simulation it is apparent that the morphogens are subject to a no flux boundary in the sense that the pattern diffuses outwards then inwards and back outwards as if it is being reflected.

The spotted and labyrinth patterns, as well as the dynamic and stationary behavior of these patterns corresponded with the references and the time evolved behavior of the concentrations of these morphogens correctly reflected the boundary and initial conditions. Future directions include implementing a spectral solver then expanding into 3D with semi-implicit methods.

AI Usage Statement:

I used ChatGPT to help generate corrections to my code for debugging, to help to format it to make it look nicer, to generate the latex equations for this paper, to make sure I understood the motivating paper by Alan Turing, and to better understand some of the decisions made in the implementations of the code in the sources mentioned in the next section. Indicated in the Jupyter notebook is where I used ChatGPT to make corrections and understand concepts.

References:

A. M. Turing, “The chemical basis of morphogenesis,” *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 237, no. 641, pp. 37–72, Aug. 1952.

Brorson, Stuart. “1.2: Forward Euler Method.” *Mathematics LibreTexts*, Libretexts, 26 July 2022, math.libretexts.org/Bookshelves/Differential_Equations/Numerically_Solving_Ordinary_Differential_Equations_(Brorson)/01%3A_Chapters/1.02%3A_Forward_Euler_method.

Navarro, Pierre. “Gray-Scott Model.” *Gray-Scott Model - Python-Fortran Notebooks*, pnavaro.github.io/python-fortran/06.gray-scott-model.html

Rasmussen, Jan. “Janmr Blog.” *Finite Difference Discretization of the 2D Laplace Operator*, janmr.com/blog/2024/05/finite-difference-discretization-of-2d-laplace/.

Sims, Karl. “Reaction-Diffusion Tutorial.” *Karl Sims Home Page*, www.karlsims.com/rd.html