

Project

Hunter Schwartz

May 7, 2021

Project

The system was implemented in the manner described by Pearson as below. The Laplacian is formed from a finite difference approximation, timesteps are forward Euler steps, and the boundary conditions are periodic.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.animation as ani

def lap(A,p,q):
    # Approximation of Laplacian of A with periodic boundary conditions
    lapA = np.zeros_like(A)
    lapA = (np.roll(A, -1, axis=0) - 2*A + np.roll(A, 1, axis=0) ) / (q**2)
    lapA += (np.roll(A, -1, axis=1) - 2*A + np.roll(A, 1, axis=1) ) / (p**2)
    return lapA

# Dimensions of simulation
width = 2.5
height = 2.5

# Number of grid points in x, then y
M = 256
N = 256

# Timestep size and number of steps
t = 1.
```

```

T = 80000

Du = 2e-5
Dv = 1e-5

# Other paramters

F = .02
k = .045

# Initial conditions of U and V
U = np.ones((N,M))
V = np.zeros((N,M))

U[118:138, 118:138] = .5 * (1 + .02*(np.random.random_sample((20,20)) - .5))
V[118:138, 118:138] = .25 * (1 + .02*(np.random.random_sample((20,20)) - .5))

# END of parameters

# Grid spacing in x, y
p = width / M
q = height / N

Animate = False

if not Animate:
    for i in range(T):
        U = U + t*( Du*lap(U,p,q) - U*V**2 + F*(1-U) )
        V = V + t*( Dv*lap(V,p,q) + U*V**2 - (F+k)*V )

        fig = plt.figure()
        ax = plt.axes()

        ax.imshow(U, cmap='coolwarm', interpolation='spline16')
        ax.set_xticks(())
        ax.set_yticks(())

if Animate:

```

```

def take_timestep(n):
    global U,V,t,Du,Dv,f,k
    for i in range(n):
        U = U + t*( Du*lap(U,p,q) - U*V**2 + F*(1-U) )
        V = V + t*( Dv*lap(V,p,q) + U*V**2 - (F+k)*V )

for i in range(500):
    plt.imsave(f'img{i}.png', U, cmap = 'coolwarm')
    take_timestep(200)

plt.imshow(U, cmap='coolwarm')

```

In Figure 1 you can see many of the patterns demonstrated in the paper by Pearson, along with the parameter values that generated them. Of potential particular interest is pattern (c), which does not appear in the Pearson paper (it is well-known that Pearson's list is not exhaustive). Pattern (c), despite its apparent chaotic nature, seems to be a steady state of the problem at least empirically. Running the simulation multiple times with parameters $F = 0.02, k = 0.045$ yields qualitatively similar solutions (like a cross-section of cabbage) by 20,000 timesteps, although the specific pattern varies depending on the perturbations of the initial condition. After pulsating rapidly for many of the early timesteps, the pattern crystallizes without warning and then no longer evolves in time as far as the eye can see.

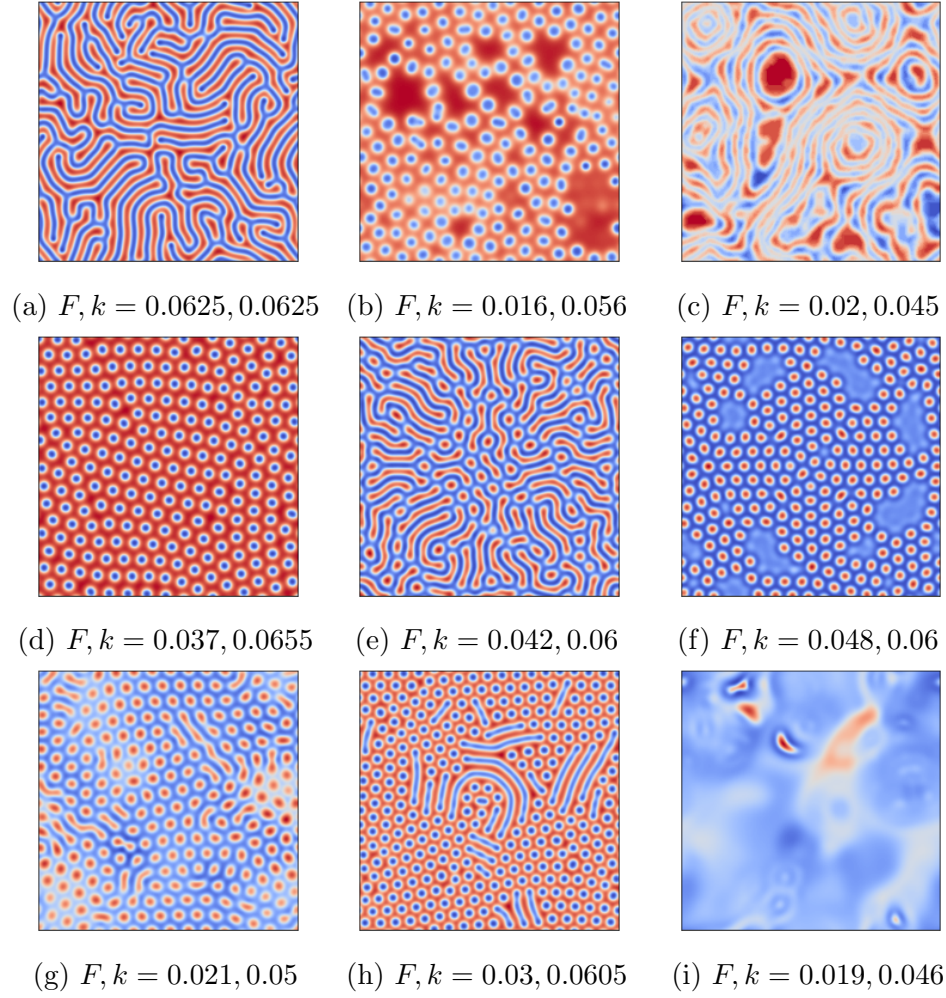


Figure 1: A variety of patterns formed from several permutations of parameters.