In this project, worth 3 points, you will simulate the dynamics of the Gray-Scott reaction diffusion system. (see the following: http://www-swiss.ai.mit.edu/projects/amorphous/GrayScott/, http://www.sciencemag.org/cgi/content/abstract/261/5118/189)

1 Background: The Gray-Scott Reaction-Diffusion System

Consider an autocatalytic reacting system containing species U, V, and P in which the following irreversible reactions occur to form the inert product P:

\[
\begin{align*}
U + 2V &\rightarrow 3V \quad \text{rate} = k_1[U][V]^2 \\
V &\rightarrow P \quad \text{rate} = k_2[V],
\end{align*}
\]

where \(k_1\) and \(k_2\) are reaction rate constants and \([U]\) and \([V]\) represent the concentrations of species U and V. If U is fed into the reacting system with a concentration of \([U]_0\) and a rate of \(k_f\) and U, V, and P are the removed from the system at their respective concentrations and the same rate \(k_f\), then the reaction kinetics for the system can be written as

\[
\begin{align*}
\frac{\partial[U]}{\partial t} &= -k_1[U][V]^2 + k_f ([U]_0 - [U]), \\
\frac{\partial[V]}{\partial t} &= +k_1[U][V]^2 - k_2[V] - k_f[V].
\end{align*}
\]

If we nondimensionalize the system by choosing \([U]_0\) as a characteristic concentration and \(k_1\) as a characteristic rate constant, then \((k_1[U]_0^2)^{-1}\) is a characteristic time and the above system can be written in dimensionless variables as

\[
\begin{align*}
\frac{\partial U}{\partial \tilde{t}} &= -UV^2 + F (1 - U), \\
\frac{\partial V}{\partial \tilde{t}} &= +UV^2 - (k + F) V,
\end{align*}
\]

where \(U \equiv [U]/[U]_0\) and \(V \equiv [V]/[U]_0\) are the dimensionless concentrations of U and V, \(k \equiv k_2/(k_1[U]_0^2)\) is the dimensionless rate constant for the second reaction, \(F \equiv k_f/(k_1[U]_0^2)\) is the dimensionless feed rate, and \(\tilde{t} \equiv k_1[U]_0^2 t\) is the dimensionless time. If we further allow for diffusion of both U and V, then we have the Gray-Scott reaction-diffusion system:

\[
\begin{align*}
\frac{\partial U}{\partial \tilde{t}} &= \tilde{D}_U \nabla^2 U - UV^2 + F (1 - U), \\
\frac{\partial V}{\partial \tilde{t}} &= \tilde{D}_V \nabla^2 V + UV^2 - (k + F) V,
\end{align*}
\]
where \( \tilde{D}_U \equiv D_U/(k_1[U]^2_0) \) and \( \tilde{D}_V \equiv D_V/(k_1[U]^2_0) \) are rescaled diffusion coefficients of \( U \) and \( V \), respectively.

The Gray-Scott system exhibits a wide array of pattern-forming behavior [GS84], [Pea93], so it is a particularly interesting system to simulate numerically. As can be seen from (4) and (5), the dynamics of the system are governed by four parameters: \( \tilde{D}_U, \tilde{D}_V, k, \) and \( F \). In this project, following Pearson [Pea93], the diffusion coefficients will be fixed and the dynamics of the system will be explored for different combinations of \( k \) and \( F \).

2 Problem Statement

For this project, you will write a program to simulate the 2-D Gray-Scott reaction diffusion system using the finite-difference method. This will be accomplished by tracking the time evolution of the dimensionless concentrations \( U \) and \( V \) over a square domain with explicit forward-Euler time stepping. The code will start with initial conditions for the concentrations of \( U \) and \( V \) as shown in Fig. 1 and will track the time evolution of the concentrations of \( U \) and \( V \) as they react and diffuse in the plane. Periodic boundary conditions will be utilized, so that an outward flux of species along one side of the domain will result in an inward flux of species along the opposite side. Proceed as follows in writing your code:

1. Write a function that will compute and return the two-dimensional finite-difference approximation to the Laplacian of a field \( f \) given an input 2-D array of \( f \) values and a scalar node spacing \( h \) as inputs. Use periodic boundary conditions, so that when the finite-difference formula for the Laplacian refers to a node outside of the domain, a value from the other side of the domain is used.

   Hints:
   (a) In 2-D Cartesian coordinates, the Laplacian of a scalar field \( f \) is given as
   \[
   \nabla^2 f = \left( \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right). 
   \]
   (b) Write your function using array operations so that the entire Laplacian is computed in a single line.
   (c) Use the \texttt{scipy.roll()} function to simplify using periodic boundary conditions.

2. Write a function that will compute and return the right-hand sides of (4) and (5) given input scalar values for \( h, \tilde{D}_U, \tilde{D}_V, F, \) and \( k \) and input 2-D arrays for the current concentration fields for \( U \) and \( V \). Use the Laplacian function you created in the previous step to compute the \( \nabla^2 \) terms, and use simple array operations to compute the other terms.

3. Discretize the left-hand sides of (4) and (5) in time using forward-difference approximations to the time derivatives. Solve the resulting expressions for the future concentrations of \( U \) and \( V \) as a functions of the current concentrations by taking the right-hand
Figure 1: Initial conditions for $U$ and $V$. The ±0.01 notation indicates the addition of uniform random noise in the range $[-0.01, 0.01)$ to break the symmetry of the concentrations.
sides of (4) and (5) to refer to the values of the $U$ and $V$ fields at the current time step.

4. Create a pair of 2-D arrays $U$ and $V$, each with a size of $n_x \times n_y$ nodes. Set the initial concentrations of $U$ and $V$ as shown in Fig. 1, where the $\pm 0.01$ notation here refers to the addition of uniformly-distributed random noise in the range $[-0.01, 0.01]$. Hint: Use `scipy.stats.uniform.rvs` function with suitable arguments for the `loc`, `scale`, and `size` arguments to generate an $n_x \times n_y$ array of uniform random numbers in the desired range.

5. Given the initial concentration fields $U$ and $V$ as computed in the previous step and values for the parameters as given in Table 1, simulate the system from dimensionless time $\tilde{t} = 0$ to time $\tilde{t} = 10000$. Table 1 gives several interesting values for $k$ and $F$ to explore. See [Pea93] for further details.

6. Every 10 time steps during the simulation, create and save side-by-side plots of the concentration fields for $U$ and $V$. Your plots should be labeled and have color keys next to each plot. Save the plots with numbered file names so that they can be easily combined into a movie.

Hints:

(a) Use `pylab.imshow()` with values supplied for the `vmin` and `vmax` arguments to render the image using the same color scale for every frame.

(b) Use `pylab.colorbar()` to draw the color legend.

Submit your well-commented code along with end-of-simulation sample plots for each of the $(k, F)$ pairs listed in Table 1.

References


Table 1: Parameters for use in simulating the 2-D Gray-Scott system

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Numerical Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of nodes in $x$ direction</td>
<td>$n_x$</td>
<td>64</td>
</tr>
<tr>
<td>number of nodes in $y$ direction</td>
<td>$n_y$</td>
<td>64</td>
</tr>
<tr>
<td>spacing between nodes (nondim)</td>
<td>$h$</td>
<td>0.01</td>
</tr>
<tr>
<td>diffusion coefficient of $U$</td>
<td>$\dot{D}_u$</td>
<td>$2 \times 10^{-5}$</td>
</tr>
<tr>
<td>diffusion coefficient of $V$</td>
<td>$\dot{D}_v$</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>time step</td>
<td>$\Delta t$</td>
<td>$0.9h^2/(4D_u)$</td>
</tr>
<tr>
<td>initial composition of $U$</td>
<td>$U_0$</td>
<td>see Fig. 1</td>
</tr>
<tr>
<td>initial composition of $V$</td>
<td>$V_0$</td>
<td>see Fig. 1</td>
</tr>
<tr>
<td>dimensionless reaction constant and feed rate</td>
<td>$(k, F)$</td>
<td>(0.060, 0.040)</td>
</tr>
<tr>
<td>(interesting values taken from [Pea93])</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.056, 0.024)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.059, 0.040)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.056, 0.020)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.050, 0.016)</td>
</tr>
</tbody>
</table>