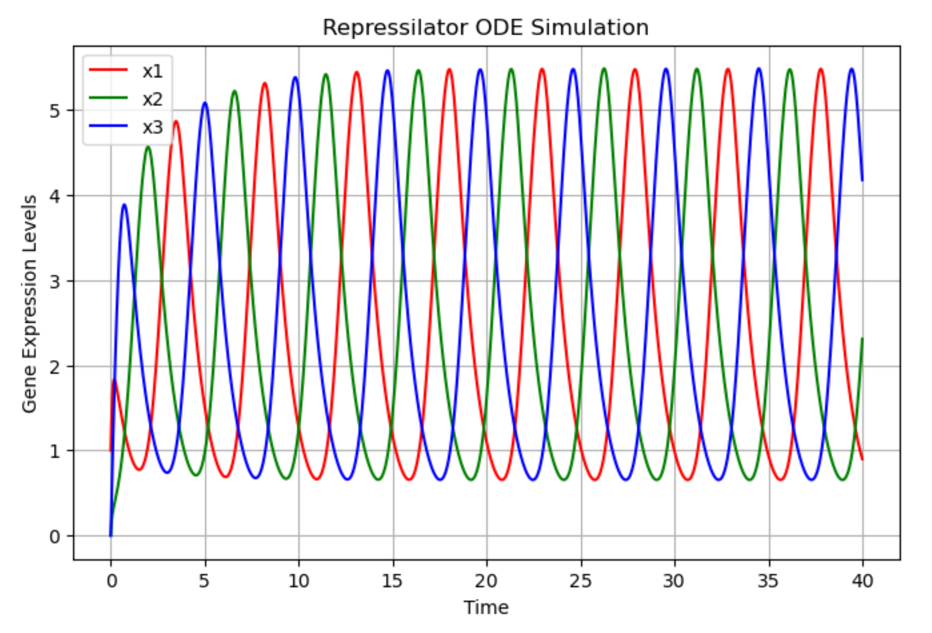
```python
# Kiley Huffman
# QBIO 482
# Spring 2025
# Homework 3
```

Question 3:

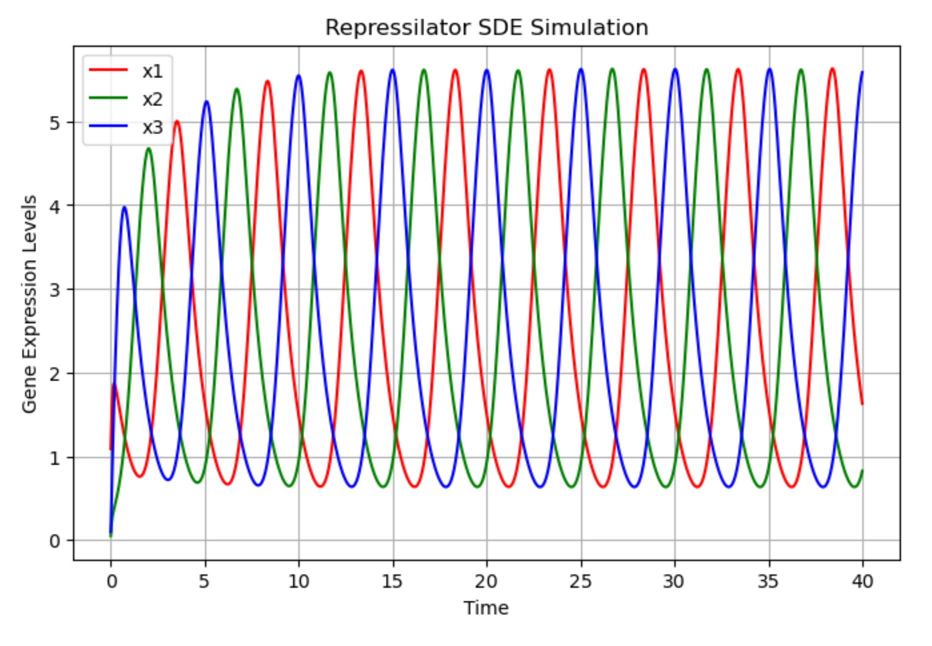
3d. Simulate the ODE version of this model in Python. Use β=10, γ=1, n=3 and simulate the model between a time range of [0,40].

```python
# Get packages
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

# Define ODE system:
def repressilator(t, x, beta, gamma, n):
    x1, x2, x3 = x
    dx1 = beta / (1 + x3**n) - gamma * x1
    dx2 = beta / (1 + x1**n) - gamma * x2
    dx3 = beta / (1 + x2**n) - gamma * x3
    return [dx1, dx2, dx3]

# Parameters (given in question)
beta = 10
gamma = 1
n = 3
time_range = (0, 40)
t_eval = np.linspace(0, 40, 1000)  # Time points for evaluation

# Initial conditions
x0 = [1, 0, 0]

# Solve the ODE
solution = solve_ivp(repressilator, time_range, x0, args=(beta, gamma, n), t_eval=t_eval)

# Plot results
plt.figure(figsize=(8, 5))
plt.plot(solution.t, solution.y[0], label='x1', color='r')
plt.plot(solution.t, solution.y[1], label='x2', color='g')
plt.plot(solution.t, solution.y[2], label='x3', color='b')
plt.xlabel("Time")
plt.ylabel("Gene Expression Levels")
plt.title("Repressilator ODE Simulation")
plt.legend()
plt.grid()
plt.show()
```



3e. Now simulate the SDE model in Python. Describe the results of simulation with δ = λ= 0.01. Describe the results of simulation if δ = 0.01 and λ = 10? In this latter case, do you think the core behaviors of the model are preserved? Explain your reasoning.

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

# Define repressilator SDE system
def repressilator_sde(x, beta, gamma, n, delta, lambda_, dt):
    x1, x2, x3 = x
    dx1 = beta / (1 + x3**n) - gamma * x1 + delta * x1 * np.random.normal(0, np.sqrt(dt))
    dx2 = beta / (1 + x1**n) - gamma * x2 + delta * x2 * np.random.normal(0, np.sqrt(dt))
    dx3 = beta / (1 + x2**n) - gamma * x3 + lambda_ * x3 * np.random.normal(0, np.sqrt(dt))
    return [dx1, dx2, dx3]

# Parameters
beta = 10
gamma = 1
n = 3
delta = 0.1  # Noise parameter for x1 and x2
lambda_ = 0.1  # Noise parameter for x3
time_range = (0, 40)
dt = 0.01
t_eval = np.arange(0, 40, dt)  # Time points for evaluation

# Initial conditions
x = np.array([1, 0, 0])
x_hist = []

# Simulate SDE using Euler-Maruyama method
for t in t_eval:
    x = x + np.array(repressilator_sde(x, beta, gamma, n, delta, lambda_, dt)) * dt
    x_hist.append(x.copy())

x_hist = np.array(x_hist).T

# Plot results
plt.figure(figsize=(8, 5))
plt.plot(t_eval, x_hist[0], label='x1', color='r')
plt.plot(t_eval, x_hist[1], label='x2', color='g')
plt.plot(t_eval, x_hist[2], label='x3', color='b')
plt.xlabel("Time")
plt.ylabel("Gene Expression Levels")
plt.title("Repressilator SDE Simulation")
plt.legend()
plt.grid()
plt.show()
```



Describe the results of simulation with δ = λ= 0.01:

The noise level of 0.01 is small, so the stochastic fluctuations are minor. The oscilliations here remain regular and periodic, closely resembling the ODE model. This system still exhibits limit-cycle behavior, but there are small variations in amplitude and phase shifts due to stochastic perturbations.

Describe the results of simulation if δ = 0.01 and λ = 10:

When the noise in x3 is significantly larger than the noise in x1 and x2, the oscillations in x3 become highly erratic and unpredictable. This causes irregularity and noisy dynamics. In turn, this disrupts the regular oscillitory behavior in x1 and x2. Thus, the sustained oscillations in the system are no longer preserved.

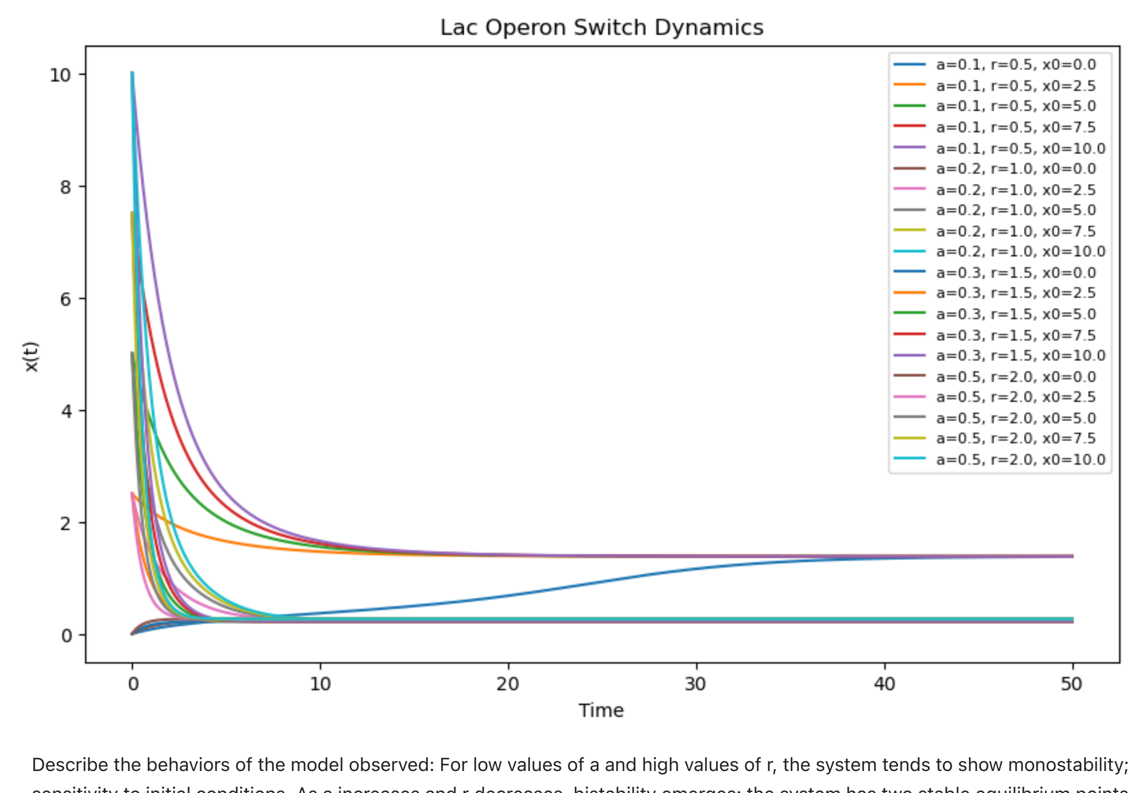
In this latter case, do you think the core behaviors of the model are preserved? Explain your reasoning:

No, the core behaviors of the model are not preserved in the later case. This occurs because the excessive noise in x3 disrupts the cyclic interaction, making oscillations less coherent.

Question 4:

4a. Simulate the Lac operon switch model for different values of the parameters using appropriate numerical ODE solvers in Python. Describe the behaviors of the model observed. Consider initial conditions $x(\in [0,10]$).

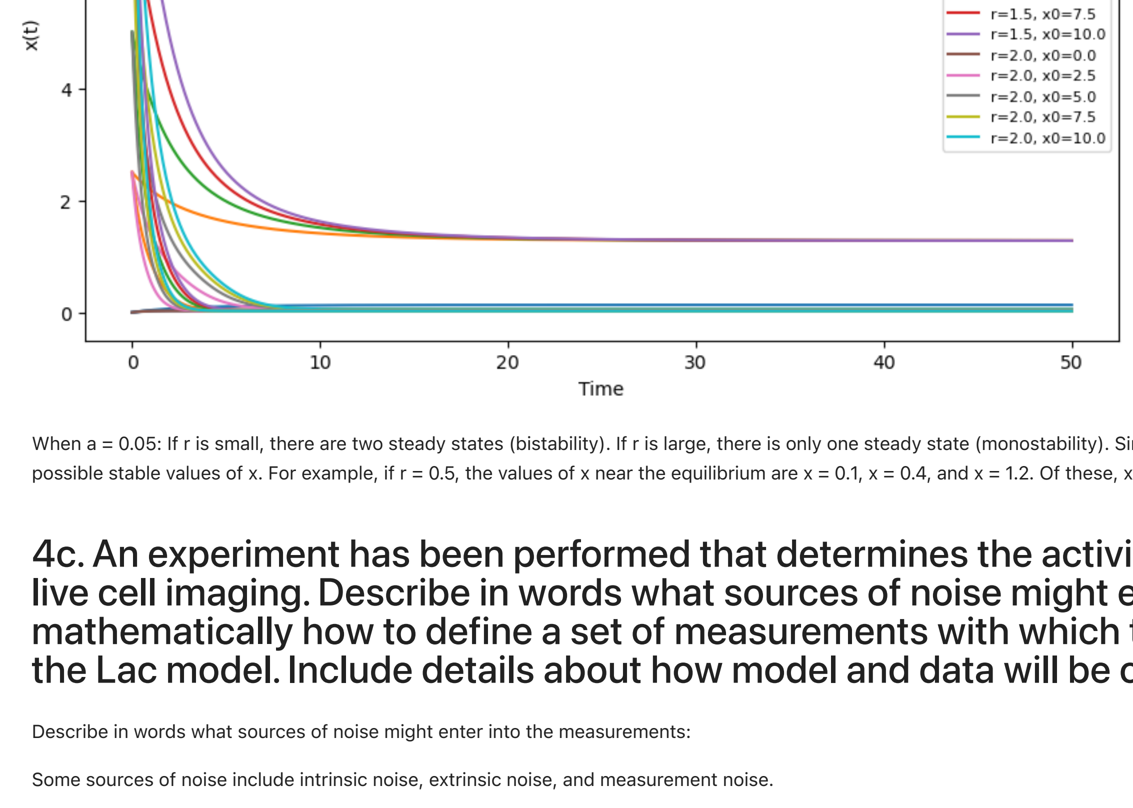
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# Define the Lac operon switch ODE
def lac_operon(x, t, a, r):
    return (a + x**2) / (1 + x**2) - r*x

# Time range for simulation
t = np.linspace(0, 50, 1000)

# Parameter sets to explore
parameter_sets = [(0.1, 0.5), (0.2, 1.0), (0.3, 1.5), (0.5, 2.0)]

# Initial conditions
initial_conditions = np.linspace(0, 10, 5)

plt.figure(figsize=(10, 6))

# Simulate and plot for each parameter set
for a, r in parameter_sets:
    for x0 in initial_conditions:
        sol = odeint(lac_operon, x0, t, args=(a, r))
        plt.plot(t, sol, label=f'a={a}, r={r}, x0={x0}')

plt.xlabel("Time")
plt.ylabel("x(t)")
plt.title("Lac Operon Switch Dynamics")
plt.legend(loc="best", fontsize=8)
plt.show()
```



Describe the behaviors of the model observed: For low values of a and high values of r, the system tends to show monostability; the dynamics converge to a single stable steady state, with little sensitivity to initial conditions. As a increases and r decreases, bistability emerges; the system has two stable equilibrium points. Thus, the long-term behavior depends on the initial condition. Lower initial conditions lead to one steady state, and higher initial conditions lead to another. This suggests the model acts like a biological switch. The system is highly sensitive to the values of a and r. While larger values of a and smaller values of r promote bistability, while larger r values tend to suppress bistability.

4b. For a = 0.05, explore the behavior of the model. Simulate for a range of initial conditions and discuss whether you think is one steady state or more than one. Why? What is/are the values of x near the equilibrium?

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# Define the Lac operon switch ODE with a = 0.05
def lac_operon(x, t, a, r):
    return (a + x**2) / (1 + x**2) - r*x

# Time range for simulation
t = np.linspace(0, 50, 1000)

# Parameter value
a = 0.05

# Explore a range of r values and initial conditions
r_values = [0.5, 1.0, 1.5, 2.0]
initial_conditions = np.linspace(0, 10, 5)

plt.figure(figsize=(10, 6))

# Simulate and plot for each r value and initial condition
for r in r_values:
    for x0 in initial_conditions:
        sol = odeint(lac_operon, x0, t, args=(a, r))
        plt.plot(t, sol, label=f'r={r}, x0={x0}')

plt.xlabel("Time")
plt.ylabel("x(t)")
plt.title("Lac Operon Switch Dynamics for a = 0.05")
plt.legend(loc="best", fontsize=8)
plt.show()
```



When a = 0.05: If r is small, there are two steady states (bistability). If r is large, there is only one steady state (monostability). Since the equilibrium values depend on r, lower r values allow for two possible stable values of x. For example, if r = 0.5, the values of x near the equilibrium are x = 0.1, x = 0.4, and x = 1.2. Of these, x = 0.1 and x = 1.2 are stable equilibrium.

4c. An experiment has been performed that determines the activity of the Lac operon over time via live cell imaging. Describe in words what sources of noise might enter into the measurements. Define mathematically how to define a set of measurements with which to perform parameter inference for the Lac model. Include details about how model and data will be compared.

Describe in words what sources of noise might enter into the measurements:

Some sources of noise include intrinsic noise, extrinsic noise, and measurement noise.

```
-Intrisic noise occurs to the random binding/unbinding of transcription factors, RNA polymerase, and ribosomes.
 The production and degradation of mRNA and proteins occur in small, discrete bursts, which lead to fluctuations in expression levels.

-Extrinsic noise occurs due to differences in cell size, metabolic state, or growth phase (cell to cell variability).

-Measurement noise occurs due to variability in fluorophore maturation rates, photobleaching, and differences in expression efficiency.
```

Define mathematically how to define a set of measurements with which to perform parameter inference for the Lac model. Include details about how model and data will be compared:

First, we will need to define the experimental dataset D = {(t, y)} as noisy measurements of Lac operon activity. Second, we will need to simulate the model x(t; theta) by solving the ODE with parameters theta = (a, r). Then, we will need to compare the model and data using statistical method, such as likelihood-based inference (MLE) to find the best parameters, and bayesian inference (MCMC) to estimate parameter distribution. Lastly, we will access model fit using MSE, R**2, and cross validation.

4d. Describe two possible alternative ways to estimate the parameters of the model given these measurements. Choose one that you would implement, giving reasons for your choice. Discuss any challenges you can foresee with your chosen approach.

Describe two possible alternative ways to estimate the parameters of the model given these measurements:

1. Least Squares Estimation (LSE): This method minimizes the sum of squared differences between the observed data and the model predictions. It is simple and computationally efficient. It works well when noise is Gaussian and errors are independently distributed. This method can be implemented using numerical optimization techniques like gradient descent or the Levenberg-Marquardt algorithm.
2. Bayesian Inference via Markov Chain Monte Carlo (MCMC): This method estimates a probability distribution over the parameters instead of a single best-fit value. It applies Bayes' Theoreom and samples from the posterior distribution, providing uncertainty quantification. This method is robust to noise and small datasets, but is computationally expensive.

Choose one that you would implement, giving reasons for your choice. Discuss any challenges you can foresee with your chosen approach:

I would implement the Bayesian Inference using Markov Chain Monte Carlo (MCMC) method because it provides uncertainty estimates for parameters rather than just point estimates, can incorporate prior knowledge about biologically realistic values of a and r, and is more robust to noise and data sparsity. Some challenges to this approach are that it is computationally expensive and can have issues with convergence. zA

4e. Without performing parameter inference for the model, sketch how the posterior probability parameter distribution might look. Do you think the parameters are likely to be correlated or uncorrelated? Explain your reasoning.

Sketch how the posterior probabillity parameter distribution might look: A rough sketch of the posterior probability might show a banana-shaped or elongated contour if the parameters are correlated.

The parameters are likely correlated because they jointly determine the balance of activation and repression in the system. As a result, the posterior probability distribution would likely be elongated along a diagonal ridge.