models

April 20, 2025

Figure I: Asexual-Only Model (3 Regimes)

[1]:
```
import numpy as np
import matplotlib.pyplot as plt

t = np.linspace(0, 10, 500)
params = {
    "Supercritical (  >  )": (0.6, 0.2),
    "Critical (  =  )": (0.4, 0.4),
    "Subcritical (  <  )": (0.2, 0.6),
}
colors = ['tab:blue', 'tab:orange', 'tab:red']

plt.figure(figsize=(8, 5))
for (label, (lam, mu)), color in zip(params.items(), colors):
    N_t = np.exp((lam - mu) * t)
    plt.plot(t, N_t, label=label, color=color)

plt.title("Figure 1: Asexual-Only Diversification")
plt.xlabel("Time")
plt.ylabel("Expected Number of Species")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("figure1_asexual_regimes.png")
plt.show()
```

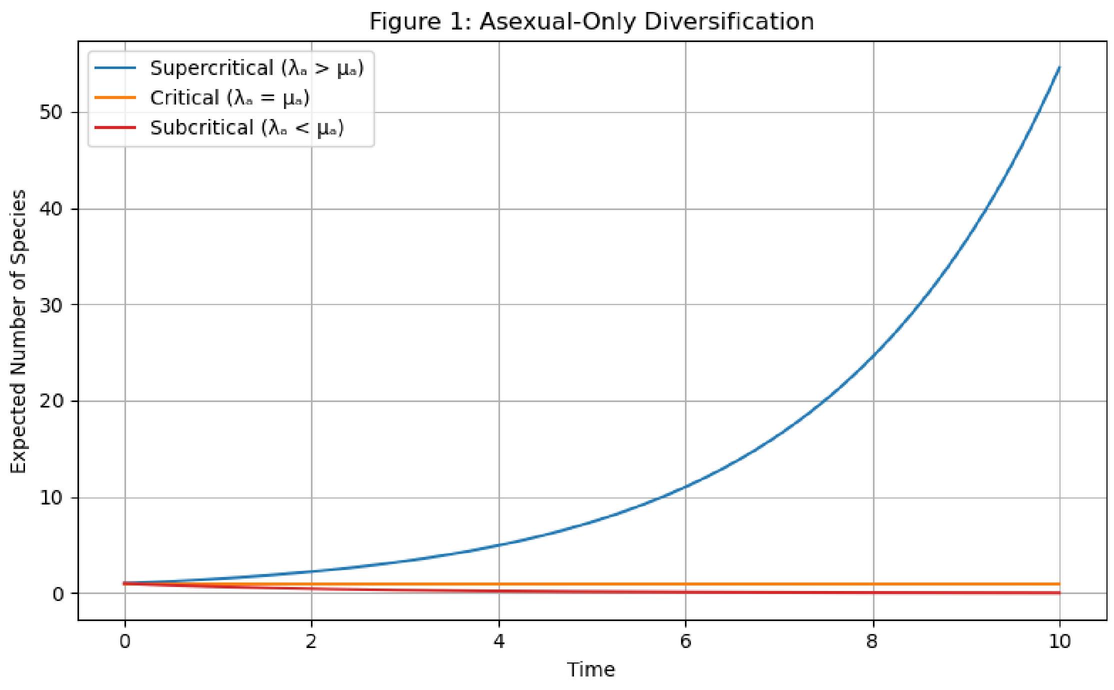


Figure 2: Subplots for Asexual and Sexual Growth (Side-by-Side)

[2]:
```python
import numpy as np
import matplotlib.pyplot as plt

# Time range
t = np.linspace(0, 20, 500)

# Parameters for asexual and sexual species
lambda_a, mu_a = 0.7, 0.5
lambda_s, mu_s = 0.4, 0.2

# Expected number of species over time
N_a = np.exp((lambda_a - mu_a) * t)
N_s = np.exp((lambda_s - mu_s) * t)

# Create side-by-side subplots
fig, axs = plt.subplots(1, 2, figsize=(14, 5))

# --- Asexual subplot ---
axs[0].plot(t, N_a, label="Asexual Species", color='tab:red', linewidth=2)
axs[0].set_title("Figure 2a: Asexual Lineage Growth")
axs[0].set_xlabel("Time")
axs[0].set_ylabel("Expected Number of Species")
axs[0].grid(True)
```

```
axs[0].legend()

# --- Sexual subplot ---
axs[1].plot(t, N_s, label="Sexual Species", color='tab:blue', linewidth=2)
axs[1].set_title("Figure 2b: Sexual Lineage Growth")
axs[1].set_xlabel("Time")
axs[1].set_ylabel("Expected Number of Species")
axs[1].grid(True)
axs[1].legend()

# Layout and save
plt.tight_layout()
plt.savefig("figure2_dual_model_side_by_side.png")
plt.show()
```

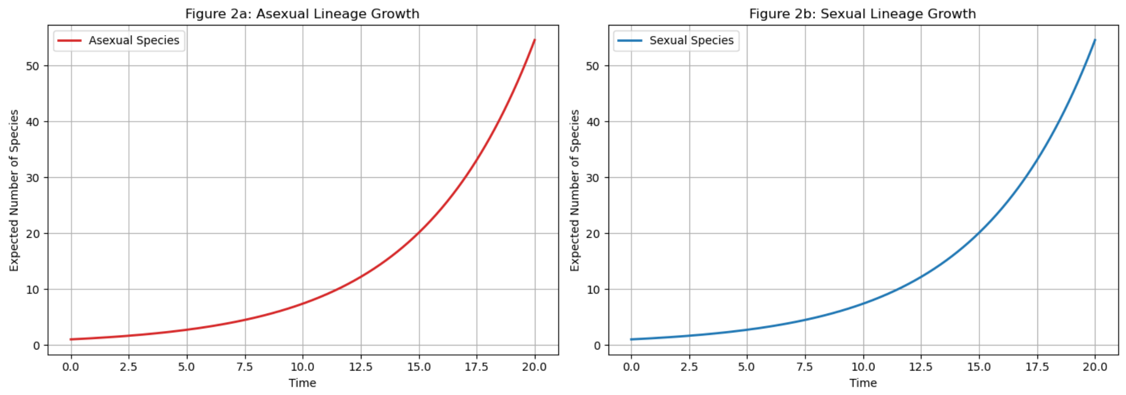


Figure III: ODE Model with Switching Between Reproductive Modes

[3]:
```
from scipy.integrate import solve_ivp

def switching_model(t, y):
    Na, Ns = y
    dNa = (0.7 - 0.5)*Na - 0.1*Na + 0.05*Ns
    dNs = (0.4 - 0.2)*Ns + 0.1*Na - 0.05*Ns
    return [dNa, dNs]

sol = solve_ivp(switching_model, (0, 20), [1, 0], t_eval=np.linspace(0, 20,␣
 ↪500))

plt.figure(figsize=(8, 5))
plt.plot(sol.t, sol.y[0], label="Asexual", linestyle='--', color='tab:orange',␣
 ↪linewidth=2)
plt.plot(sol.t, sol.y[1], label="Sexual", linestyle='-', color='tab:blue',␣
 ↪linewidth=2)
```

```
plt.title("Figure 3: ODE Solution with Switching")
plt.xlabel("Time")
plt.ylabel("Expected Number of Species")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("figure3_switching_odes.png")
plt.show()
```

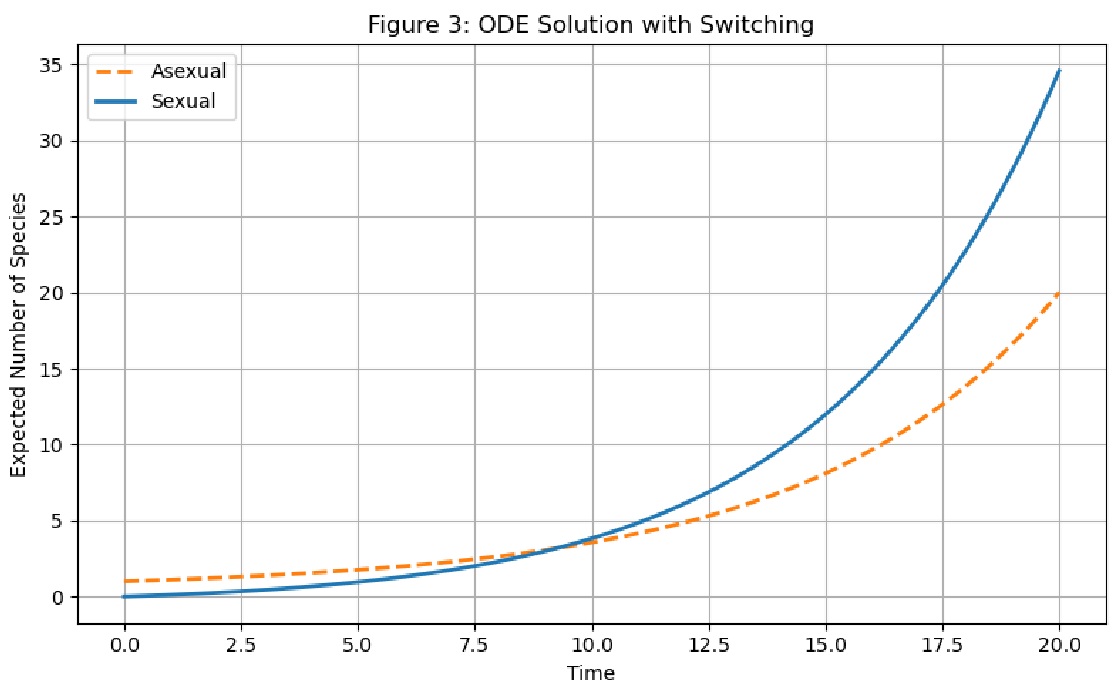


Figure IV: Gillespie Stochastic Simulation for Switching Model

[5]:
```
import numpy as np
import matplotlib.pyplot as plt

def gillespie_sim(t_max, lambda_a, mu_a, lambda_s, mu_s, sigma_a_to_s,␣
 ↪sigma_s_to_a, N_a0, N_s0):
    t, na, ns = 0, N_a0, N_s0
    times, Na_list, Ns_list = [0], [na], [ns]

    while t < t_max and (na > 0 or ns > 0):
        rates = [
            lambda_a * na,
            mu_a * na,
            sigma_a_to_s * na,
            lambda_s * ns,
            mu_s * ns,
            sigma_s_to_a * ns
        ]
        total = sum(rates)
        if total == 0:
            break
        dt = np.random.exponential(1 / total)
        event = np.random.choice(6, p=np.array(rates) / total)
        t += dt

        if event == 0: na += 1
        elif event == 1 and na > 0: na -= 1
        elif event == 2 and na > 0: na -= 1; ns += 1
        elif event == 3: ns += 1
        elif event == 4 and ns > 0: ns -= 1
        elif event == 5 and ns > 0: ns -= 1; na += 1

        times.append(t)
        Na_list.append(na)
        Ns_list.append(ns)

    return np.array(times), np.array(Na_list), np.array(Ns_list)

# Plot corrected version
plt.figure(figsize=(8, 5))

for i in range(3):
    t_vals, Na_vals, Ns_vals = gillespie_sim(20, 0.7, 0.5, 0.4, 0.2, 0.1, 0.05,␣
 ↪1, 0)
    # Asexual: dashed green, only label once
    plt.plot(t_vals, Na_vals, linestyle='--', color='tab:green',␣
 ↪label='Asexual' if i == 0 else "")
    # Sexual: solid blue, only label once
    plt.plot(t_vals, Ns_vals, linestyle='-', color='tab:blue', label='Sexual'␣
 ↪if i == 0 else "")

plt.title("Figure 4: Gillespie Simulation of Switching Model")
plt.xlabel("Time")
plt.ylabel("Number of Species")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("figure4_gillespie_corrected.png")
plt.show()
```

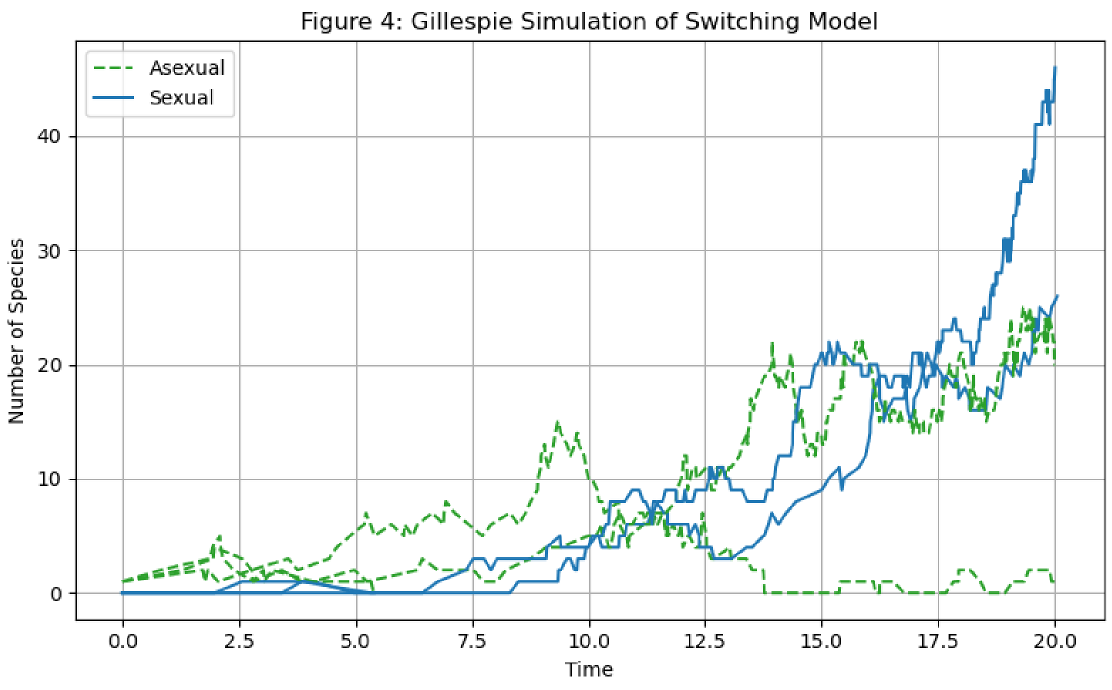


Figure V: Comparison of all models

[6]:
```
t = np.linspace(0, 20, 500)
N_model1 = np.exp((0.6 - 0.2) * t)
N_model2 = np.exp((0.7 - 0.5) * t) + np.exp((0.4 - 0.2) * t)
N_model3 = sol.y[0] + sol.y[1]

plt.figure(figsize=(8, 5))
plt.plot(t, N_model1, label="Model I: Asexual Only", linestyle=':', color='tab:
  ↪red')
plt.plot(t, N_model2, label="Model II: No Switching", linestyle='--',␣
  ↪color='tab:purple')
plt.plot(sol.t, N_model3, label="Model III: Switching", linestyle='-',␣
  ↪color='tab:green')

plt.title("Figure 5: Comparison of Diversification Models")
plt.xlabel("Time")
plt.ylabel("Total Expected Number of Species")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("figure5_model_comparison.png")
plt.show()
```

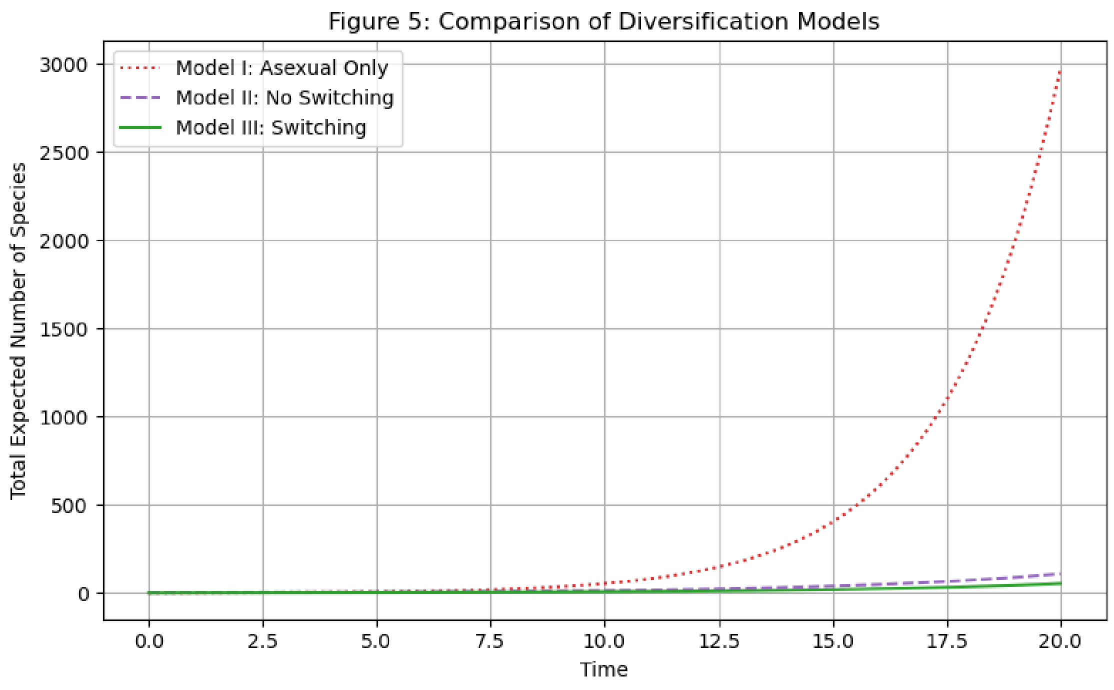


Figure VI: Gillespie Simulations for Asexual vs Sexual Lineages

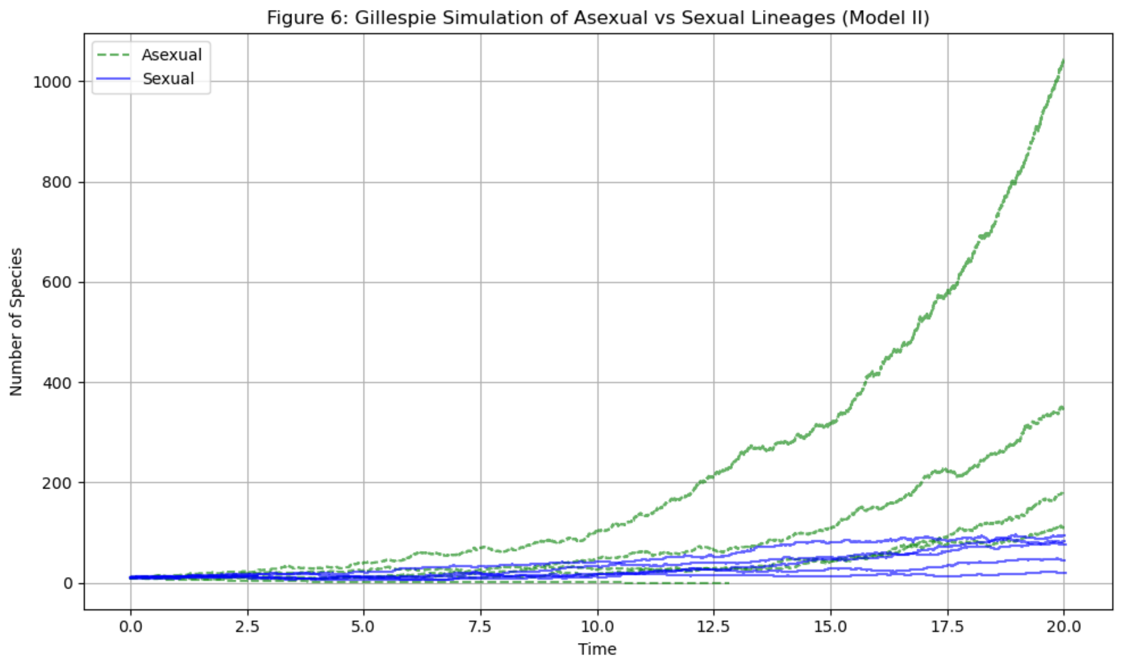
[1]:
```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
lambda_a, mu_a = 0.6, 0.4
lambda_s, mu_s = 0.3, 0.2
N0_a, N0_s = 10, 10
T_max = 20
num_simulations = 5

# Gillespie function
def gillespie(lambda_, mu_, N0, T_max):
    t, N = 0, N0
    times, species = [0], [N]
    while t < T_max and N > 0:
        rate = lambda_ * N + mu_ * N
        if rate == 0: break
        t += np.random.exponential(1 / rate)
        N += 1 if np.random.rand() < lambda_ * N / rate else -1
        times.append(t)
        species.append(N)
    return np.array(times), np.array(species)
```

```
# Simulations
asexual_runs = [gillespie(lambda_a, mu_a, N0_a, T_max) for _ in␣
  ↪range(num_simulations)]
sexual_runs = [gillespie(lambda_s, mu_s, N0_s, T_max) for _ in␣
  ↪range(num_simulations)]

# Plot
plt.figure(figsize=(10, 6))
for t, N in asexual_runs:
    plt.step(t, N, color='green', linestyle='--', alpha=0.6, label='Asexual' if␣
  ↪t is asexual_runs[0][0] else "")
for t, N in sexual_runs:
    plt.step(t, N, color='blue', linestyle='-', alpha=0.6, label='Sexual' if t␣
  ↪is sexual_runs[0][0] else "")
plt.title("Figure 6: Gillespie Simulation of Asexual vs Sexual Lineages (Model␣
  ↪II)")
plt.xlabel("Time")
plt.ylabel("Number of Species")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("figure6_gillespie_model2.png", dpi=300)
plt.show()
```



Model I Gillespie Simulation

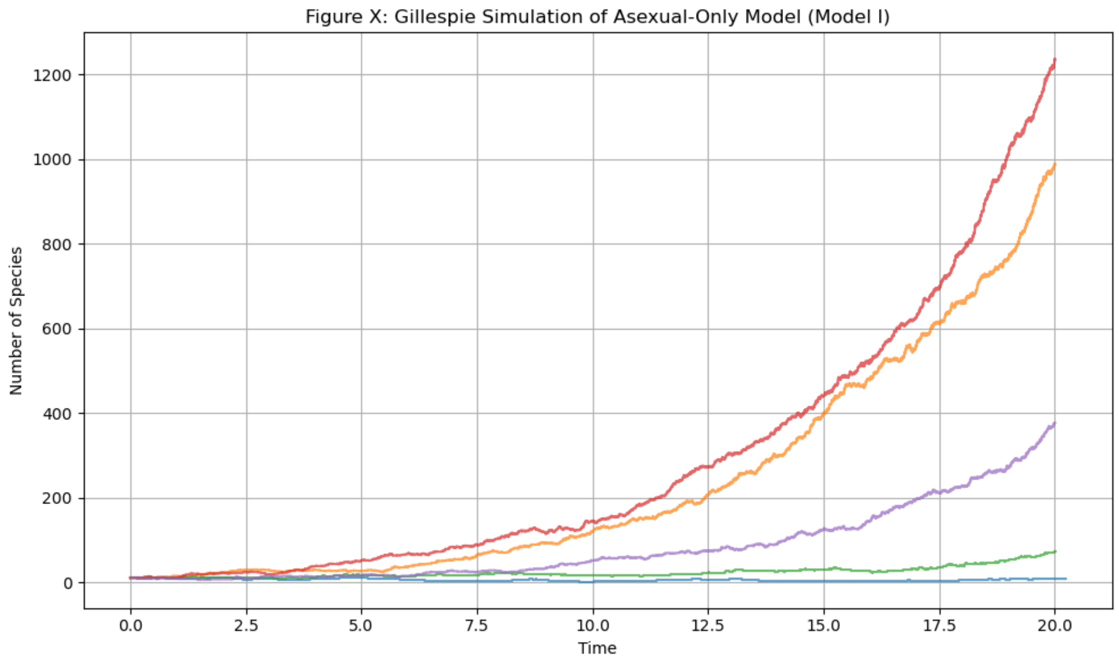
[2]:
```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
lambda_a = 0.6
mu_a = 0.4
N0 = 10
T_max = 20
num_simulations = 5

def gillespie_asexual(lambda_a, mu_a, N0, T_max):
    t = 0
    N = N0
    times = [0]
    species = [N]
    while t < T_max and N > 0:
        rate = lambda_a * N + mu_a * N
        if rate == 0:
            break
        t += np.random.exponential(1 / rate)
        if np.random.rand() < lambda_a * N / rate:
            N += 1
        else:
            N -= 1
        times.append(t)
        species.append(N)
    return np.array(times), np.array(species)

# Run simulations
trajectories = [gillespie_asexual(lambda_a, mu_a, N0, T_max) for _ in␣
 ↪range(num_simulations)]

# Plot
plt.figure(figsize=(10, 6))
for t, N in trajectories:
    plt.step(t, N, where='post', alpha=0.7)
plt.title("Figure X: Gillespie Simulation of Asexual-Only Model (Model I)")
plt.xlabel("Time")
plt.ylabel("Number of Species")
plt.grid(True)
plt.tight_layout()
plt.savefig("figureX_gillespie_model1_asexual.png", dpi=300)
plt.show()
```



Each line represents one simulation trajectory of an asexual species population.This highlights the stochastic variability — even with a positive net growth rate, some populations can collapse due to bad luck early on.

```
[ ]:
```