Modeling Tutorial: From One-Dimensional Growth to Predator–Prey Systems

Prof. Robert J. Flassig

October 20, 2025

1 Learning Goals

By the end of this tutorial you will be able to

- translate verbal assumptions into quantitative models,
- simulate and visualize the dynamics of nonlinear systems,
- interpret the meaning of model parameters in a physical or ecological context,
- explore how parameter changes affect qualitative system behavior,
- optionally: perform stability and phase-space analysis.

2 The Logistic Growth Model (1D)

We start with a single variable $x(t)$ representing, for example, a population size or concentration.

2.1 Model construction

(Verbal) Assumptions:

A1 The population grows at a rate proportional to its current size.

A2 Growth slows down as the population approaches a maximum capacity K.

Modeling Verbal Assumptions:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = r\,x\Big(1 - \frac{x}{K}\Big)\,,$$

where

- r: intrinsic growth rate [1/time],
- K: carrying capacity,
- x: population size.

Interpretation: At small x, the term $(1 - x/K) \approx 1$, and we recover exponential growth. As $x \to K$, the growth term approaches zero—resources limit further increase.

2.2 Numerical simulation in Python/Colab

Below is an example of a minimal simulation using the `Euler` method:

Listing 1: Simple Euler integration of the logistic model

```python
import numpy as np
import matplotlib.pyplot as plt

# Parameters
r = 1.0      # growth rate
K = 10.0     # carrying capacity
x0 = 0.5     # initial population
dt = 0.01    # time step
T  = 10.0    # total time

# Time vector and storage
t = np.arange(0, T, dt)
x = np.zeros_like(t)
x[0] = x0

# Euler integration
for i in range(1, len(t)):
    dxdt = r * x[i-1] * (1 - x[i-1]/K)
    x[i] = x[i-1] + dt * dxdt

# Plot
plt.plot(t, x)
plt.xlabel("time t")
plt.ylabel("population x")
plt.title("Logistic growth")
plt.show()
```

2.3 Tasks

T1 Interpret the parameters r and K: what does changing each do biologically or physically?

T2 Re-run the simulation for several r values. How does the system's approach to equilibrium change?

T3 Add an initial condition $x_0 > K$ and interpret the dynamics.

T4 Create a small plot grid comparing different r and K combinations.

T5 **Optional:** Determine analytically the equilibrium points and their stability.

3 The Predator–Prey System (2D)

Now consider a coupled system of two interacting populations:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = r\,x\left(1 - \frac{x}{K}\right) - \alpha xy, \tag{1}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \beta xy - \delta y. \tag{2}$$

3.1 Meaning of parameters

- r: prey reproduction rate,
- K: prey carrying capacity,
- α: predation rate coefficient,
- β: efficiency with which prey consumption translates to predator growth,
- δ: predator death rate.

3.2 Initial code example

The same Euler idea extends easily:

Listing 2: Predator-prey simulation (Euler scheme)

```
# Parameters
r, K = 1.0, 5.0
alpha, beta, delta = 0.2, 0.1, 0.4
x0, y0 = 1.0, 0.5
dt, T = 0.01, 100.0

t = np.arange(0, T, dt)
x, y = np.zeros_like(t), np.zeros_like(t)
x[0], y[0] = x0, y0

for i in range(1, len(t)):
    dx = r*x[i-1]*(1 - x[i-1]/K) - alpha*x[i-1]*y[i-1]
    dy = beta*x[i-1]*y[i-1] - delta*y[i-1]
    x[i] = x[i-1] + dt*dx
    y[i] = y[i-1] + dt*dy

plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
plt.plot(t, x, label="prey")
plt.plot(t, y, label="predator")
plt.xlabel("time"); plt.legend()

plt.subplot(1,2,2)
plt.plot(x, y)
plt.xlabel("prey x"); plt.ylabel("predator y")
plt.title("Phase space")
plt.tight_layout(); plt.show()
```

3.3 Tasks

P1 Interpret each parameter in words. Which ones affect prey equilibrium? Which influence oscillation amplitude?

P2 Run the model for baseline parameters. Describe the qualitative time evolution.

P3 Produce a phase-space plot (x, y). What kind of trajectory emerges?

P4 Vary one parameter systematically (e.g. α or δ) and discuss changes in the qualitative behavior.

P5 **Optional:** Add nullcline plots and use arrows to indicate direction fields.

4 Mini Projects

4.1 Project 1: Logistic growth with harvesting

Add a constant harvest term h:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = r\,x\Big(1 - \frac{x}{K}\Big) - h.$$

Questions:

a. For what h does the population go extinct?

b. Simulate several h values and visualize $x(t)$.

c. Estimate the critical h_c separating survival and extinction.

d. Discuss management implications (sustainable yield).

4.2 Project 2: Predator–prey with saturation

Use the Holling's type II response:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = r\,x\Big(1 - \frac{x}{K}\Big) - \frac{\alpha x}{1 + hx}y, \quad \frac{\mathrm{d}y}{\mathrm{d}t} = \beta xy - \delta y.$$

Tasks:

a. Implement the model and vary h.

b. Compare time series and phase-space trajectories.

c. Interpret biologically how the handling time h changes stability.

d. **Optional:** derive the nullclines and discuss stability.

5 Supporting Hints for Simulation

- Reuse and modify the **Google Colab templates** provided in class. They contain ready-made plotting cells and Euler integration loops.
- When in doubt, test your code with small time steps (`dt = 0.001`) and compare to larger ones.
- Experiment with `scipy.integrate.solve_ivp` for higher accuracy.
- Use modern AI assistants (**ChatGPT**, **Claude.ai**, etc.) to:
  - debug code and syntax errors,
  - generate plotting routines or parameter sweep loops,
  - check your understanding of equations and their derivatives.
- Always document AI-assisted contributions and verify numerical results independently.
- Comment each code cell clearly: what parameters did you change and why?

6 Deliverables

Your short report (2–4 pages) should include:

- Equations and parameter definitions.
- Plots of time series and phase-space trajectories.
- Interpretation of how parameters affect dynamics.
- **Optional:** analytical or stability results.
- Reflection on modeling choices and learning outcomes.

7 General Hints

- Use dimensionless time $\tau = rt$ to reduce parameter count.
- Normalize populations by K to simplify visualization.
- If results diverge or oscillate unnaturally, reduce Δt.
- For visual clarity, annotate phase-space arrows with `quiver()` or `streamplot()`.