**Lab 10: The Volterra Predator-Prey Model**

In this lab, you will plot the predator-prey model in both discrete and continuous time.

Let \( H(t) \) represent the numbers of prey and \( P(t) \) represent the numbers of predators. The prey are assumed to grow exponentially, but they are captured by predators at a rate that depends on both availability of prey and predators.

The discrete time recursions are thus:

- **Prey:** \( H(t+1) = H(t) + r H(t) - b H(t) P(t) \)
- **Predator:** \( P(t+1) = P(t) + c H(t) P(t) - d P(t) \)

which uses the following parameters:
- the growth rate of the prey in the absence of the predator \( r \),
- the capture rate at which predators contact and kill prey \( b \),
- the rate at which eaten prey are turned into predator babies \( c \),
- and the death rate of predators \( d \).

The analog in continuous time is:

- \( \frac{dH}{dt} = r H(t) - b H(t) P(t) \)
- \( \frac{dP}{dt} = c H(t) P(t) - d P(t) \)

There are two main ways of graphing these equations. You can look at how each species (the predator or the prey) changes over time or you can look at the relationship of predators to prey. This second type of graph is called a phase-plane analysis.

**Graphs of discrete time system**

Here is a Mathematica simulation giving the number of prey \( (\text{popH}) \) and the number of predators \( (\text{popP}) \) over time:

```mathematica
Clear[popH, popP]
popH[x_, b_, c_, d_, t_] := popH[x, b, c, d, t - 1] + popH[x, b, c, d, t - 1] * b * popP[x, b, c, d, t - 1] + popP[x, b, c, d, t - 1] * c * popH[x, b, c, d, t - 1] - popP[x, b, c, d, t - 1] * d
popP[x_, b_, c_, d_, 0] := 500
popP[x_, b_, c_, d_, t_] := 0
(ListPlot[Table[{t, popH[0.9, 0.01, 0.0002, 0.1, t]}, {t, 0, 50}], PlotRange -> {0, 1000}])
```

Now try plotting the number of predators over time using a species-time graph:

Now try plotting the number of predators as a function of the number of prey using a phase-plane graph (each dot will represent a different time point):

```mathematica
Clear[system, mH, mP]

system[x_, b_, c_, d_, mH_, mP_, tmin_, tmax_] :=
  system[x, b, c, d, mH, mP, tmin, tmax] =
  NDsolve[
    \{mH'[t] = mH[t] * (r - b * mP[t])\},
    \{mP'[t] = mP[t] * (c * mH[t] - d)\},
    \{mH[0] == mH0, mP[0] == mP0\},
    \{mH[t], mP[t]\}, \{t, tmin, tmax\}\]

ListPlot[Table[{t, popP[0.5, 0.01, 0.0002, 0.1, t]}, {t, 0, 50}],
    PlotRange -> \{0, 100\}, \{0, 100\}]
```

Which is the x-axis and which is the y-axis? [Hint: Look at the initial numbers]

**Questions for the lab**

Sketch one of the above phase-plane diagrams, making sure to specify which axis corresponds to prey and which to predator.

In which direction do the spirals go? [Try plotting the graphs for one, two, five, and ten generations.]

Can you make sense of this biologically?

In the discrete model, what is the effect of halving both the intrinsic growth rate of prey \( r \) and the capture rate of predators \( b \)?
rate (b), but keeping \( r/b \) constant?

In the continuous model, what is the effect of changing the initial numbers of prey and predators?

Can you make sense of the fact that if the spirals form closed loops in the continuous-time model that they must spiral out in the discrete-time model? [Think about the difference in time step.]

**Using Block to simplify code [Extra]**

Writing simulations for discrete-time models as we did above can become tedious and hard to read when there are lots of variables, each of which is passed all of the parameters. The coding is more elegant using the "Block" command in Mathematica. Basically, within a Block, you can define terms that are used locally (only within the block), sending out only the last piece of information.

Here, we define a new function, simpop, that keeps track of the variables in a list [number of hosts, number of parasites].

```mathematica
Clear[simpop]
simpop[r_, b_, c_, d_, t_] := Block[{h = simpop[r, b, c, d, t - 1][[1]], p = simpop[r, b, c, d, t - 1][[2]],
    newh = h*(1 - c*h*b); p = p*(1 + c*h + d); newh, newp]
simpop[r_, b_, c_, d_, 0] := {500, 45}"
```

The block is initialized by defining the number of prey and predators at time \( t - 1 \) (the first part in []). Here, we’ve asked Mathematica to grab the first element of \( \text{simpop}[r,b,c,d,t-1] \) and call this the number of prey using \( h = \text{simpop}[r,b,c,d,t-1][[1]] \). Similarly, \( \{2\} \) grabs the second element out of the list for the number of predators.

The main advantage is that the recursions embedded within the Block are now easier to read, involving the variables in the previous time step, written simply as \( h \) and \( p \). The output is the list [number of hosts, number of parasites] at time \( t \), e.g., at time \( t = 50 \):

```mathematica
simpop[0.5, 0.01, 0.0002, 0.1, 50]
```

[158.086, 48.4537]

Thus, the following code plots the prey as the \( x \)-values and the predators as the \( y \)-values, showing the phase-plane diagram:

```mathematica
ListPlot[Table[simpop[0.5, 0.01, 0.0002, 0.1, t], {t, 0, 50}], PlotRange -> {0, 70}]
```

How would you change the above ListPlot so that the \( x \)-axis is time and the \( y \)-axis is just the number of hosts?

```mathematica
ListPlot[Table[CHANGExY, {t, 0, 50}], PlotRange -> {0, 70}]
```

*Hint, you may need to change the scale of your graph to see the number of hosts over time.