A Population Growth Equation with Diffusion

The logistic population equation studied in ordinary differential equations classes has the form:

$$p'(t) = rp(t)\left(1 - \frac{p(t)}{K}\right),$$

where $p(t)$ is the population of some species at time t, r is its intrinsic growth rate, and K is the carrying capacity of the ecosystem in which it lives. If the ecosystem contains two interacting populations, a predator population $p(t)$ and its prey $q(t)$, the following system of equations is sometimes used by biologists to model their growth:

$$\begin{aligned}\frac{dp}{dt} &= \frac{a\epsilon qp}{1+aTq} - \mu p^c \\ \frac{dq}{dt} &= rq(1-q/K) - \frac{aqp}{1+aTq},\end{aligned} \qquad (1)$$

where a is the predator attack rate, T is the predator handling time, ϵ is the predator conversion efficiency, μ is the predator death rate, and the exponent c defines the predator density dependence.

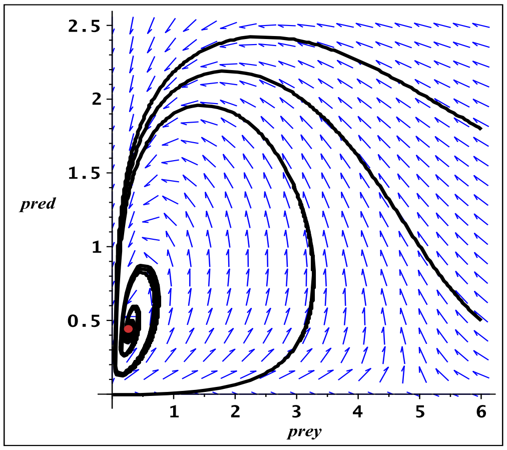
It is also a standard technique to assume that the populations are diffusing in space; that is, to add diffusion terms to each equation to obtain the following system of partial differential equations in the population variables $p(x,y,t)$ and $q(x,y,t)$:

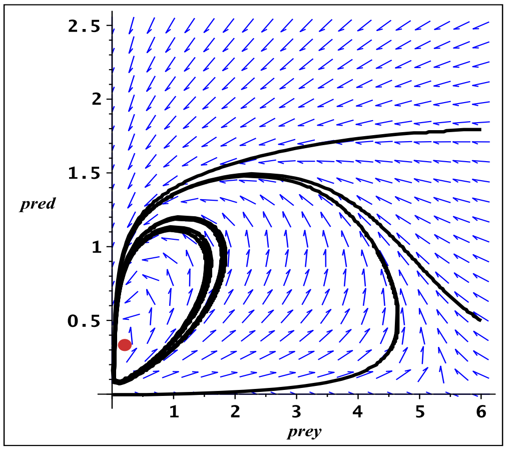
$$\begin{aligned}\frac{\partial p}{\partial t} &= D_p\left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}\right) + \frac{a\epsilon qp}{1+aTq} - \mu p^c \\ \frac{\partial q}{\partial t} &= D_q\left(\frac{\partial^2 q}{\partial x^2} + \frac{\partial^2 q}{\partial y^2}\right) + rq(1-q/K) - \frac{aqp}{1+aTq}.\end{aligned} \qquad (2)$$

Here, the constants D_p and D_q represent the diffusion rates for the predator and prey, respectively.

The following theoretical problem was brought to the writer's attention by a biologist: if the exponent c is greater than one, is there a possibility of diffusive instability in the case where the non-spatial model (1) is stable, and D_p is sufficiently greater than D_q? The following parameter values were suggested by the biologist: $r = 0.3$, $K = 5$, $a = 1$, $T = 2$, $\epsilon = 1.5$, $\mu = 0.4$, $c = 1.5$, $D_p = 0.1$, $D_q = 0.03$. These values were chosen to model a problem involving a lake containing populations of zooplankton (predators) and algae (prey). The populations are assumed to inhabit a rectangular region $0 \leq x \leq 20$ and $0 \leq y \leq 15$.

The non-spatial model (1) can be solved numerically using the `DEplot` command in Maple. With the above parameter values, `DEplot` was used to graph the phase plane of (1), and the result is shown on the left below. Note that there appears to be a stable spiral sink at $(\bar{p}, \bar{q}) \approx (0.4385, 0.2730)$. This can be easily checked by finding the equilibrium points and then using the trace-determinant plane to determine their type.





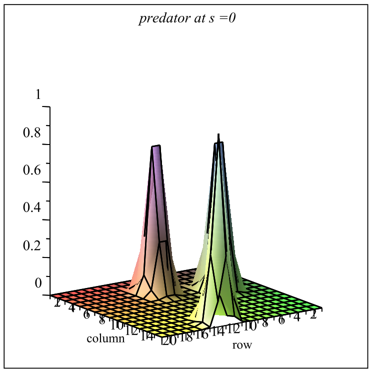
Exercise 1. Compute the Jacobian for the system (1), using the given parameters, and show that the single equilibrium at $(0.4385, 0.2730)$ is a spiral sink.

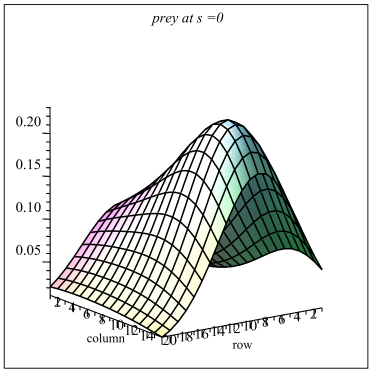
Exercise 2. Show that if the following parameter changes are made: $a = 1.5, K = 5.5, T = 2.2, \epsilon = 1.0$, and $\mu = 0.33$, the equilibrium moves to $(p, q) = (0.3294, 0.2164)$ and becomes unstable, and a limit cycle appears. A phase plane for this system is shown above on the right.

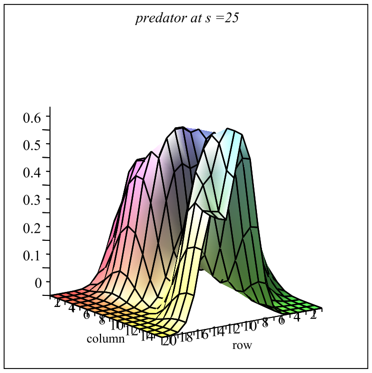
To solve the diffusive system (2), we note that each equation is a parabolic partial differential equation, but each contains nonlinear terms in p and q; however, the numerical method we have used for parabolic p.d.e.s still works. Approximating $\frac{\partial p}{\partial t}$ and $\frac{\partial q}{\partial t}$ by forward differences, and the second-order partials by central differences the reader is invited (indeed encouraged) to check that the result is the following system of difference equations for $P_{i,k,j} \equiv p(i \cdot dx, k \cdot dy, j \cdot dt)$ and $Q_{i,k,j} \equiv q(i \cdot dx, k \cdot dy, j \cdot dt)$, $i = 0, 1, \cdots, N$, $k = 0, 1, \cdots, M$. The new constants are $C_{px} = \frac{\Delta t D_p}{(\Delta x)^2}$, $C_{py} = \frac{\Delta t D_p}{(\Delta y)^2}$, $C_{qx} = \frac{\Delta t D_q}{(\Delta x)^2}$, $C_{qy} = \frac{\Delta t D_q}{(\Delta y)^2}$.

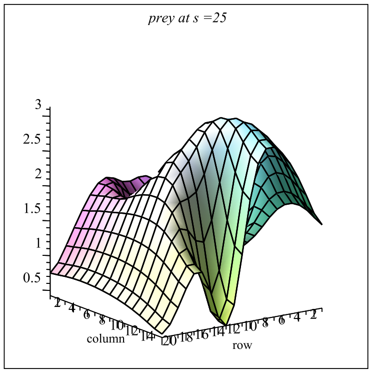
$$\begin{aligned}
P_{i,k,j+1} &= P_{i,k,j} + C_{px}(P_{i+1,k,j} - 2P_{i,k,j} + P_{i-1,k,j}) \\
&+ C_{py}(P_{i,k+1,j} - 2P_{i,k,j} + P_{i,k-1,j}) + \Delta t \left(\frac{a\epsilon P_{i,k,j} Q_{i,k,j}}{1 + aTQ_{i,k,j}} - \mu (P_{i,k,j})^c \right) \\
Q_{i,k,j+1} &= Q_{i,k,j} + C_{qx}(Q_{i+1,k,j} - 2Q_{i,k,j} + Q_{i-1,k,j}) \\
&+ C_{qy}(Q_{i,k+1,j} - 2Q_{i,k,j} + Q_{i,k-1,j}) + \Delta t \left(rQ_{i,k,j} \left(1 - \frac{Q_{i,k,j}}{K} \right) - \frac{aQ_{i,k,j} P_{i,k,j}}{1 + aTQ_{i,k,j}} \right)
\end{aligned}$$

If neither population flows across any of the edges of the lake, we can assume $\frac{\partial p}{\partial x}$ and $\frac{\partial q}{\partial x}$ are zero on the boundaries where $x = 0$ and $x = 20$, and similarly, $\frac{\partial p}{\partial y}$ and $\frac{\partial q}{\partial y}$ are zero on the boundaries where $y = 0$ and $y = 15$. In the Maple program, making $\frac{\partial p}{\partial x} \equiv 0$, for example, is done by assuming the central difference $\frac{p(\Delta x,y,t) - p(-\Delta x,y,t)}{2\Delta x} = 0$ for all y and t. This is simulated by making $P_{-1,k,j} = P_{1,k,j}$ for $k = 0, 1, \cdots, M$ at the beginning of each time step. The other three conditions are handled similarly.





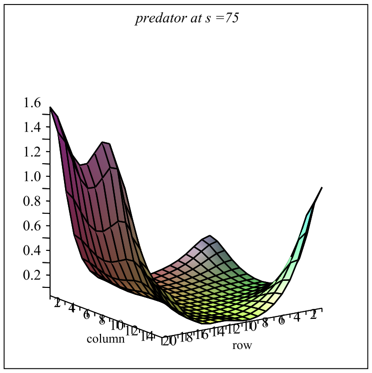


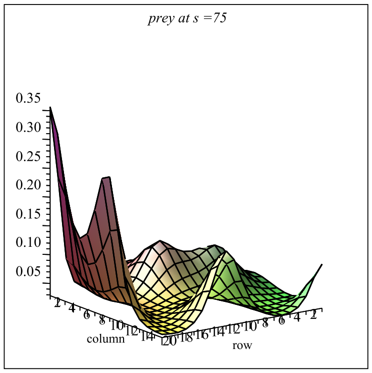


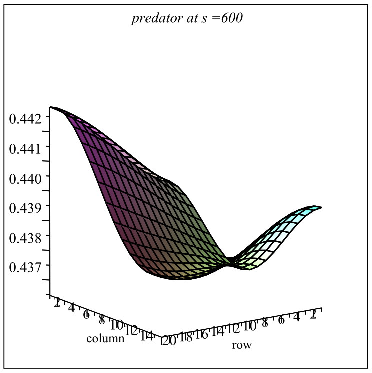
The initial condition requires specifying an initial predator population

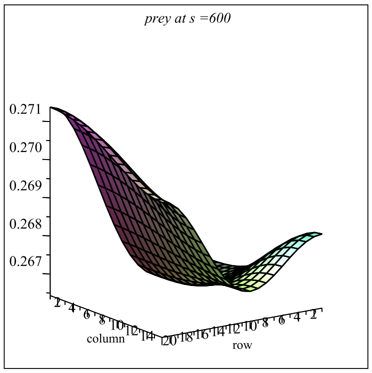
$p(x,y,0) = f(x,y)$ and initial prey population $q(x,y,0) = g(x,y)$ where f and g are functions defined on the rectangle $0 \leq x \leq 20,\ 0 \leq y \leq 15$.

The Maple program on page 6 was used to compute 600 time steps with $\Delta t = 0.5$, and with the x and y intervals partitioned so that $\Delta x = \Delta y = 1.0$. For each $j = 0, 1, \cdots, 600$ the results were stored in two 20×15 matrices, one for the predator distribution and the other for the prey. The surface defined by each of these matrices was plotted using the command `matrixplot` and the resulting plots were stored in two lists. Each of these lists can be animated to show how the corresponding population varies over time. The eight figures on pages 3 and 4 show the two populations at four different times. Be sure to take notice of the vertical scale on the graphs. When $s = 600$, both populations are nearly constant at their limiting values $p \approx 0.44$ and $q \approx 0.27$.



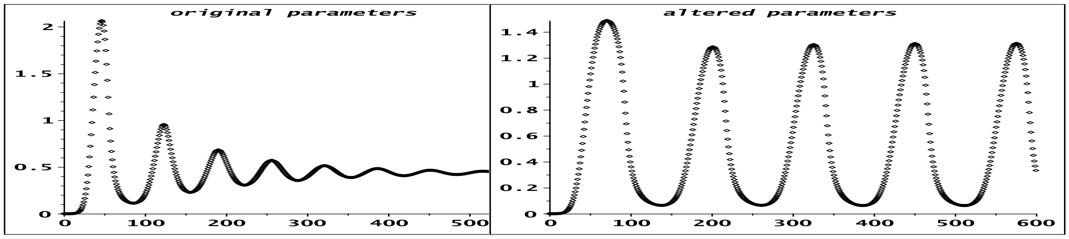






The initial functions were chosen arbitrarily. The initial predator population is modelled by a sum of three exponential functions of the form $e^{-((x-\bar{x})^2+(y-\bar{y})^2)}$ where the three points $(\bar{x}, \bar{y})$ were chosen to be $(0.3H, 0.3L)$, $(0.7H, 0.2L)$, and $(0.8H, 0.7L)$, with $H = 20$ and $L = 15$ being the dimensions of the rectangular lake. The initial prey population is assumed to be of a similar form. It was found that, while there was a rather interesting transient behavior initially, by the 600th time step ($t = 300.0$), both populations had nearly converged to their equilibrium values over the entire rectangle. Notice that the Maple program maintains a list LP, containing the predator population size in the middle of the rectangle over the time interval $s = 0..S$. When plotted, this makes it possible to see if the populations are converging to a fixed value, or oscillating in a near periodic fashion.

Exercise 3. Run the pde program with the initial parameter values (with Dp=0.1 and Dq=0.03) and notice that the populations converge to the values determined in Exercise 1. Then make the alterations given in Exercise 2 and show that the populations oscillate. The two graphs below are plots of the list LP generated by the Maple program. If you estimate the period of the oscillation, does it match your estimate for the period of the limit cycle in Exercise 2? Remember that the time t is equal to $s/2$.



Predator population in center of lake for $0 \leq s \leq S$

Exercise 4. Use the original parameters and run the pde program. See if you can make it oscillate by changing only the values of c, Dp, and Dq.

Maple program for the predator-prey system with diffusion

```
> Len:=20: N:=20: #length of pond
> Hgt:=15: M:=15: #width of pond
> r:=0.3: #prey intrinsic growth rate
> K:=5: #prey carrying capacity
> a:=1.0: #predator attack rate
> T:=2.0: #predator handling time
> eps:=1.5: #predator conversion efficiency
> mu:=0.4: #predator death rate
> c:=1.5: #predator density dependence
> Dp:=0.1: Dq:=0.03: #predator and prey diffusion rates
> g:=(x,y)->0.2*exp(-0.02*(x-0.5*Hgt)^2-0.02*(y-0.6*Len)^2)
    +0.1*exp(-0.02*(x-0.75*Hgt)^2-0.01*(y-0.1*Len)^2):
> f:=(x,y)->exp(-(x-0.3*Hgt)^2-(y-0.3*Len)^2)+exp(-(x-0.7*Hgt)^2
     -(y-0.2*Len)^2)+exp(-(x-0.8*Hgt)^2-(y-0.7*Len)^2):
> dx:=Len/N: dy:=Hgt/M: dt:=0.5: Cpx:=dt*Dp/(dx*dx):
> Cpy:=dt*Dp/(dy*dy): Cqx:=dt*Dq/(dx*dx): Cqy:=dt*Dq/(dy*dy):
> for i from 0 to N do for k from 0 to M do ## store the initial populations
    P[i,k,0]:=f(i*dx,k*dy); Q[i,k,0]:=g(i*dx,k*dy); od; od: LP:=[[0,P[10,7,0]]]:
> S:=600: for j from 0 to S-1 do ## start a time step
    for i from 0 to N do
      P[i,-1,j]:=P[i,1,j]; Q[i,-1,j]:=Q[i,1,j];
      P[i,M+1,j]:=P[i,M-1,j]; Q[i,M+1,j]:=Q[i,M-1,j]; od;
    for k from 0 to M do
      P[-1,k,j]:=P[1,k,j]; Q[-1,k,j]:=Q[1,k,j];
      P[N+1,k,j]:=P[N-1,k,j]; Q[N+1,k,j]:=Q[N-1,k,j]; od;
    for i from 0 to N do for k from 0 to M do
      P[i,k,j+1]:=P[i,k,j]+Cpx*(P[i+1,k,j]-2*P[i,k,j]+P[i-1,k,j])
        +Cpy*(P[i,k+1,j]-2*P[i,k,j]+P[i,k-1,j])
        +dt*a*eps*P[i,k,j]*Q[i,k,j]/(1+a*T*Q[i,k,j])-dt*mu*P[i,k,j]^c;
      Q[i,k,j+1]:=Q[i,k,j]+Cqx*(Q[i+1,k,j]-2*Q[i,k,j]+Q[i-1,k,j])
        +Cqy*(Q[i,k+1,j]-2*Q[i,k,j]+Q[i,k-1,j])
        +dt*r*Q[i,k,j]*(1.0-Q[i,k,j]/K)-dt*a*Q[i,k,j]*P[i,k,j]/(1+a*T*Q[i,k,j]);
    od; od; LP:=[op(LP),[j+1,P[7,10,j+1]]];
 od:  with(plots): pointplot(LP);
```

To set up the animation:

```
> with(plots): with(linalg):
> for s from 0 to S do Pred[s]:=linalg[matrix](N+1,M+1);
    Prey[s]:=linalg[matrix](N+1,M+1);
    for i from 1 to N+1 do for k from 1 to M+1 do Pred[s][i,k]:=P[i-1,k-1,s];
      Prey[s][i,k]:=Q[i-1,k-1,s]; od; od;
    GPrey[s]:=matrixplot(Prey[s],axes=frame,gap=0.02,style=patch);
    GPred[s]:=matrixplot(Pred[s],axes=frame,gap=0.02,style=patch);od:
    Lpred:=[]: Lprey:=[]: for u from 0 to S do
    Lpred:=[op(Lpred),[GPred[u]]]; Lprey:=[op(Lprey),[GPrey[u]]]; od:
> display3d(op(Lpred),insequence=true);  display3d(op(Lprey),insequence=true);
```