

Experiment with different values of the model parameters and initial conditions. Try several cases, and verify that there is an epidemic outbreak when $R_0 = S_0\beta/\alpha > 1$ and no outbreak when $R_0 < 1$.

8.4.2 A Predator-Prey Model

We denote the size of the prey population at time t by $x(t)$ and the size of the predator population by $y(t)$. We assume that in the absence of a predator the prey population approaches its carrying capacity as modelled by the logistic law (Verhulst's growth model),

$$\dot{x} = ax(1 - x/K),$$

where $a > 0$ is the per capita growth rate and $K > 0$ is the carrying capacity. We also assume that the predator population cannot survive without prey, and model this with an exponential decay equation,

$$\dot{y} = -dy.$$

For each encounter of a predator with a prey there is a certain probability that the prey will be eaten. We apply the law of mass action model and represent removal of prey by predators as $-bxy$, where $b > 0$ is a rate constant. Several successful hunts by the predator will result in the production of offspring. This is modelled with a term cxy , where $c > 0$ is a reproduction rate. In general, $b \neq c$ (why?). We obtain the following predator-prey model:

$$\begin{aligned}\dot{x} &= ax\left(1 - \frac{x}{K}\right) - bxy \\ \dot{y} &= cxy - dy.\end{aligned}$$

We nondimensionalize this model by letting $\tau = at$, $\kappa = \frac{c}{d}K$, $g = \frac{d}{a}$, $u(\tau) = \frac{c}{d}x\left(\frac{\tau}{a}\right)$, and $v(\tau) = \frac{b}{a}y\left(\frac{\tau}{a}\right)$, to get

$$\begin{aligned}\dot{u} &= u\left(1 - \frac{u}{\kappa}\right) - uv, \\ \dot{v} &= g(u - 1)v.\end{aligned}$$

The steady states of the system satisfy the following algebraic system:

$$\begin{aligned}u\left(1 - \frac{u}{\kappa}\right) - uv &= 0, \\ g(u - 1)v &= 0.\end{aligned}$$

We solve for the steady states with the “solve” command, for the specific case $\kappa = 2$ and $g = 1$:

```
> kappa := 2: g := 1:
> solve( {u*(1-u/kappa)-u*v=0, g*(u-1)*v=0}, {u,v});
```

$$\{v = 0, u = 0\}, \{v = 0, u = 2\}, \{u = 1, v = \frac{1}{2}\}$$

Maple finds three steady states. We wish to determine the stability of each of the steady states. For this, we need the Jacobian matrix of the right-hand side of the system of equations, and it is

$$Df(u, v) := \begin{pmatrix} 1 - 2\frac{u}{\kappa} - v & -u \\ gv & g(u-1) \end{pmatrix}.$$

We need to evaluate the Jacobian matrix at each of the three steady states and then determine its eigenvalues to deduce their stability. We will use Maple to do this as well. We need to use the package “linalg”:

```
> with(linalg):
```

The “jacobian” command gives us exactly what we are looking for (only now the actual values of κ and g are included):

```
> desys := vector( [u*(1-u/kappa)-u*v, g*(u-1)*v] ):
> jacobian( desys, [u,v] );
```

$$\begin{bmatrix} 1 - u - v & -u \\ v & u - 1 \end{bmatrix}$$

Unfortunately, we have to jump through some hoops to get Maple to understand the above as a function of u and v , as follows:

```
> temp := (u,v) -> jacobian( desys, [u,v] ):
> Df := (s,t) -> subs( u=s, v=t, temp(u,v) ):
> Df(u,v);
```

$$\begin{bmatrix} 1 - u - v & -u \\ v & u - 1 \end{bmatrix}$$

We define the three matrices **m1**, **m2**, and **m3** to be the Jacobian matrix evaluated at the three steady states, respectively:

```
> m1 := Df(0, 0);
```

$$m1 := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

```
> m2 := Df(2, 0);
```

$$m2 := \begin{bmatrix} -1 & -2 \\ 0 & 1 \end{bmatrix}$$

```
> m3 := Df(1, 0.5);
```

$$m3 := \begin{bmatrix} -0.5 & -1 \\ 0.5 & 0 \end{bmatrix}$$

We proceed to find the eigenvalues with the “eigenvals” command:

```
> ev1 := eigenvals(m1);  
> ev2 := eigenvals(m2);  
> ev3 := eigenvals(m3);
```

Exercise 8.4.3:

Verify that the steady states (0,0) and (2,0) are saddles, and that (1,0.5) is a stable spiral.

Note that if you also are interested in eigenvectors for each eigenvalue, then the “eigenvects” command will be useful.

We can go further and visualize the complete phase portrait. For this, we need the “DEtools” package for differential equations:

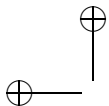
```
> with(DEtools);
```

First, we define the differential equations:

```
> de1 := diff(u(t),t) = u(t)*(1-u(t)/kappa)-u(t)*v(t);  
> de2 := diff(v(t),t) = g*(u(t)-1)*v(t);
```

We now can graph the vector field defined by the above differential equations using the “dfieldplot” command:

```
> dfieldplot( [de1,de2], [u(t),v(t)],  
t=0..1, u=0..2, v=0..0.8 );
```



Finally, we specify a list of initial conditions with the “seq” command. Each initial condition is in the form $[t_0, u(t_0), v(t_0)]$. The corresponding trajectories then can be drawn in the phase plane with the “phaseportrait” command:

```
> initcond := seq( [0,2,i*0.1], i=1..5);  
> phaseportrait( [de1,de2], [u(t),v(t)], t=0..10,  
  {initcond}, stepsize=0.1, linecolor=black );
```

