The Lotke-Volterra system

Introduction. Setting up.

Here is the Lotke-Volterra system, suitably scaled: (Putting the factor r in one equation and $\frac{1}{r}$ in the other will normalize the period at the fixed point, as will be seen in the linearized analysis below.)

```
> restart;
  `Lotke-Volterra`:=D(x)=(1/r)*x*(1-y),D(y)=r*y*(x-1):
  `Lotke-Volterra`[1]; `Lotke-Volterra`[2];
```

$$\mathrm{D}(x) = \frac{x\,(1-y)}{r}$$

$$\mathrm{D}(y) = r\,y\,(x-1)$$

DEplot wants the equations with the dependence of t made explicit:

```
> dynamics:=subs({x=x(t),D(x)=D(x)(t),y=y(t),D(y)=D(y)(t)},
            [`Lotke-Volterra`]);
```

$$dynamics := \left[\mathrm{D}(x)(t) = \frac{\mathrm{x}(t)\,(1-\mathrm{y}(t))}{r}, \mathrm{D}(y)(t) = r\,\mathrm{y}(t)\,(\mathrm{x}(t)-1)\right]$$

Set up some useful colours

```
> macro(darkred=COLOR(RGB,0.5,0,0),
        darkgreen=COLOR(RGB,0,0.5,0),
        darkblue=COLOR(RGB,0,0,0.5)):
```

Linearized analysis

Compute the linearization of the system at the equilibrium point $(x, y) = (1, 1)$:

```
> linalg[matrix](2,2,eval(subs(`Lotke-Volterra`,

  '[diff(D(x),x),diff(D(x),y),diff(D(y),x),diff(D(y),y)]')));
```

$$\begin{bmatrix} \frac{1-y}{r} & -\frac{x}{r} \\ r\,y & r\,(x-1) \end{bmatrix}$$

```
> subs(x=1,y=1,%);
```

$$\begin{bmatrix} 0 & -\frac{1}{r} \\ r & 0 \end{bmatrix}$$

```
> linalg[eigenvals](%);
```

$$I, -I$$

(We should have been able to do the last step in our head.) Thus the linearized system has a center with period $2\,\pi$.

The solutions, plotted against time

We create a numerical solution:

```
> numsolutions:=dsolve({op(subs(r=0.8,dynamics)),x(0)=0.4,y(0)=
  0.4},
     {x(t),y(t)}, type=numeric, output=listprocedure);
```

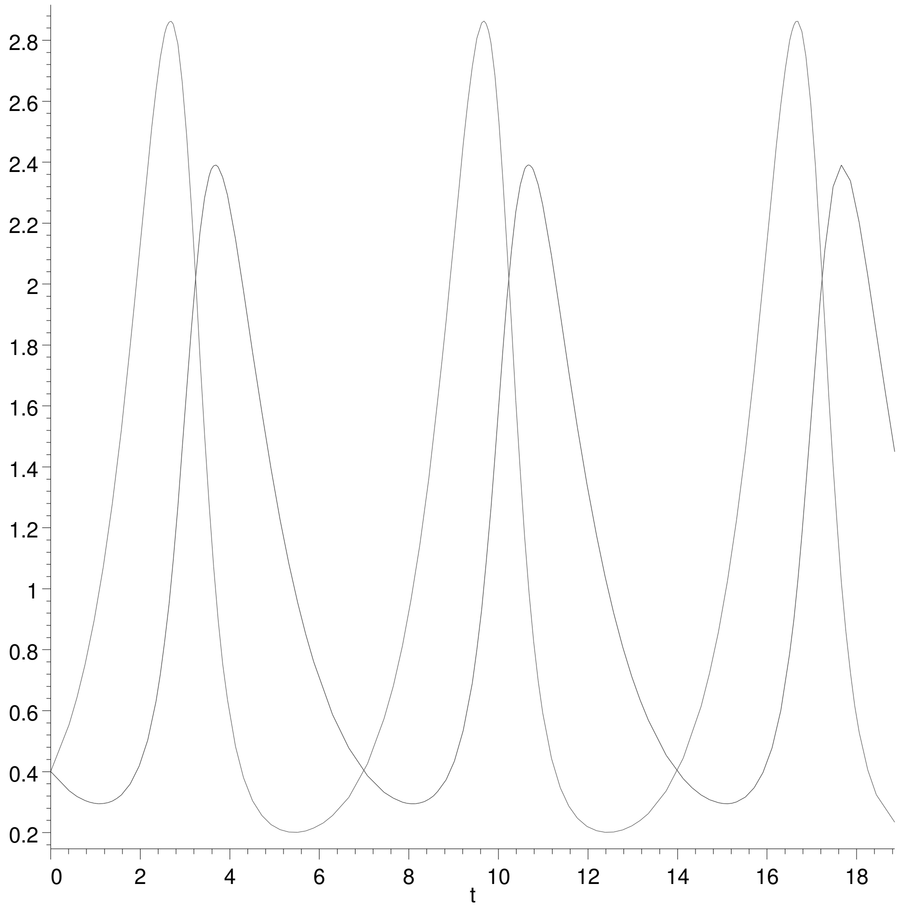
numsolutions :=

$$[t = (\textbf{proc}(t)\ \ldots\ \textbf{end}), \mathrm{x}(t) = (\textbf{proc}(t)\ \ldots\ \textbf{end}), \mathrm{y}(t) = (\textbf{proc}(t)\ \ldots\ \textbf{end})]$$

The output is merely a list of procedures, that, when called, will compute the numerical solutions. We assign names to these and use them to plot the solution. In order to control colour (prey=green, predator=red) we create plots of the two solutions separately and then display them together using display:

```
> nx:=subs(numsolutions,x(t)):
  ny:=subs(numsolutions,y(t)):
```

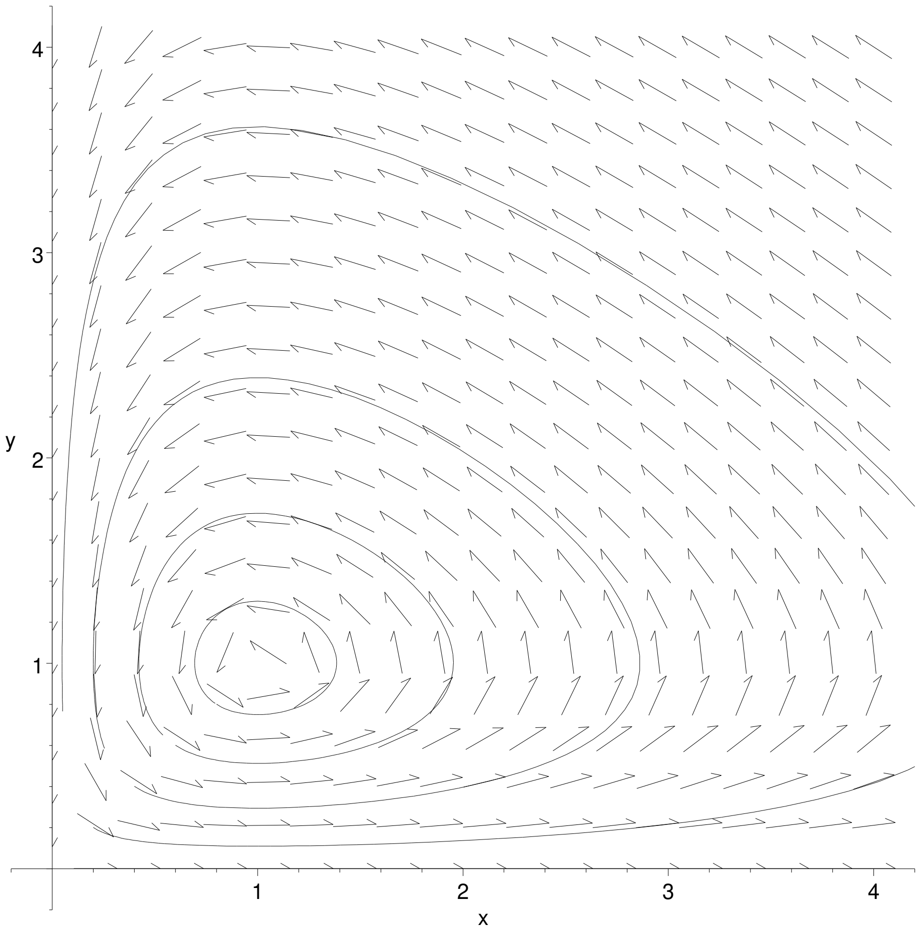
```
> preyplot:=plot('nx(t)',t=0..6*Pi,colour=darkgreen):
  predatorplot:=plot('ny(t)',t=0..6*Pi,colour=darkred):
  plots[display](preyplot,predatorplot);
```



A look in the phase plane

```
> initials:=map(p->[x(0)=p[1],y(0)=p[2]],
                [[0.2,0.2],[0.4,0.4],[0.6,0.6],[0.8,0.8]]):
  [[s(0)=1,c(0)=0],[s(0)=0.5,c(0)=0],[s(0)=0.2,c(0)=1],
             [s(0)=0.5,c(0)=0.5]]:
  deplot:=DEtools[DEplot](subs(r=0.8,dynamics),

  [x,y],t=0..2*Pi,initials,x=0..4,y=0..4,stepsize=0.05,
             colour=black,linecolour=darkblue):
  plots[display](deplot,scaling=CONSTRAINED);
```



An invariant

One way to find an invariant for the system: Divide one differential equation by the other and solve.

```
> D(y)(x)=subs({`Lotke-Volterra`},y=y(x),D(y)/D(x));
```

$$\mathrm{D}(y)(x) = \frac{r^2\,\mathrm{y}(x)\,(x-1)}{x\,(1-\mathrm{y}(x))}$$

```
> dsolve(%,y(x),implicit);
```

$$x - \ln(x) + \frac{\mathrm{y}(x)}{r^2} - \frac{\ln(\mathrm{y}(x))}{r^2} + \_C1 = 0$$

```
> solve(subs(y(x)=y,%),_C1);
```

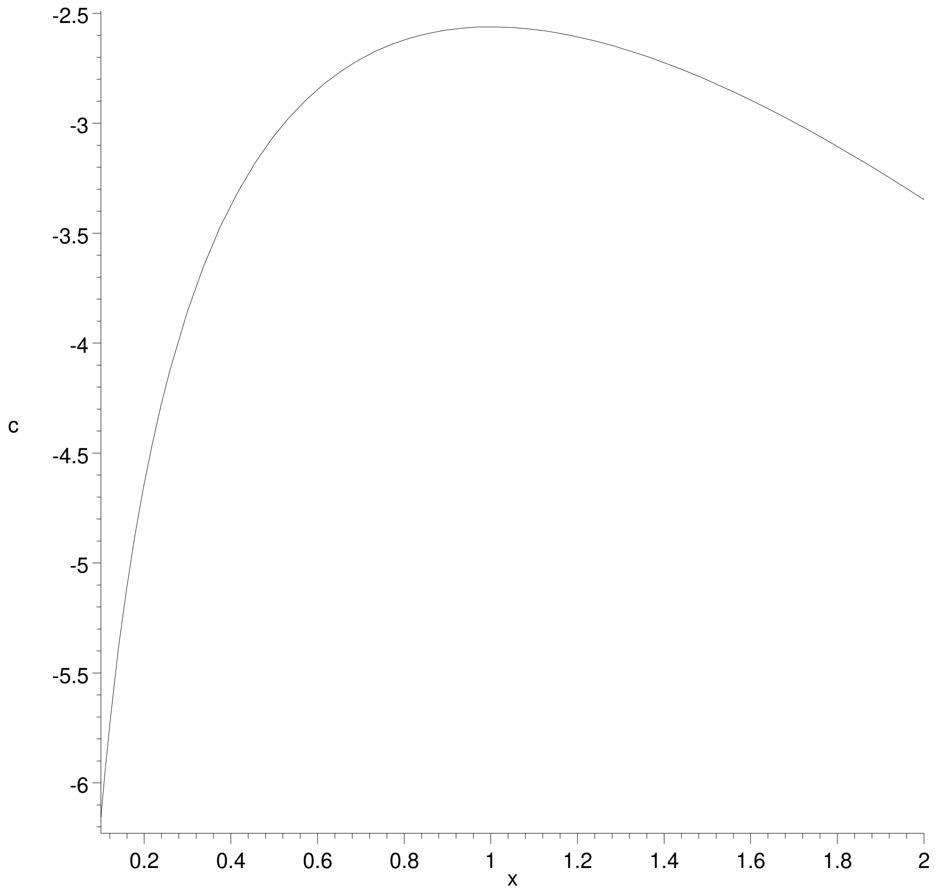
$$-\frac{x\,r^2 - \ln(x)\,r^2 + y - \ln(y)}{r^2}$$

```
> H:=unapply(%,x,y):
> eval(subs(r=0.8,H(1,1)));
```

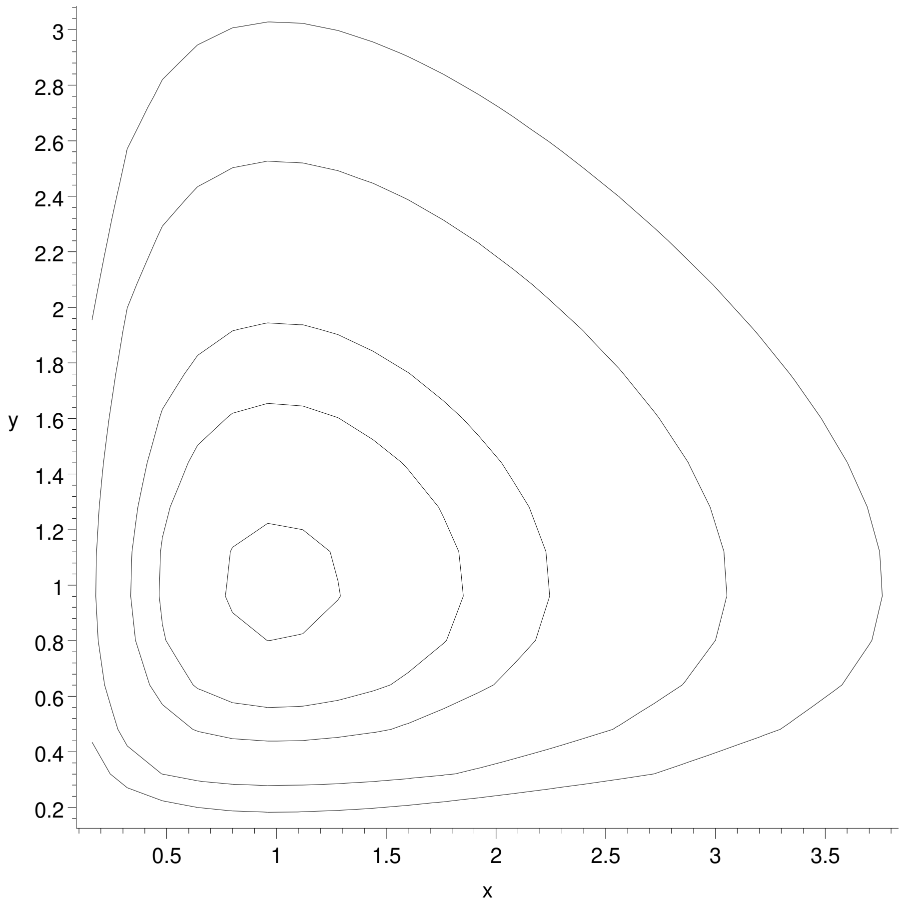
$$-2.562500000$$

The contourplot routine did not by itself find good contour values to use for the plot. So I plot the values of H along the diagonal, to find suitable values to use:

```
> plot(subs(r=0.8,H(x,x)),x=0.1..2,c,color=darkblue);
```

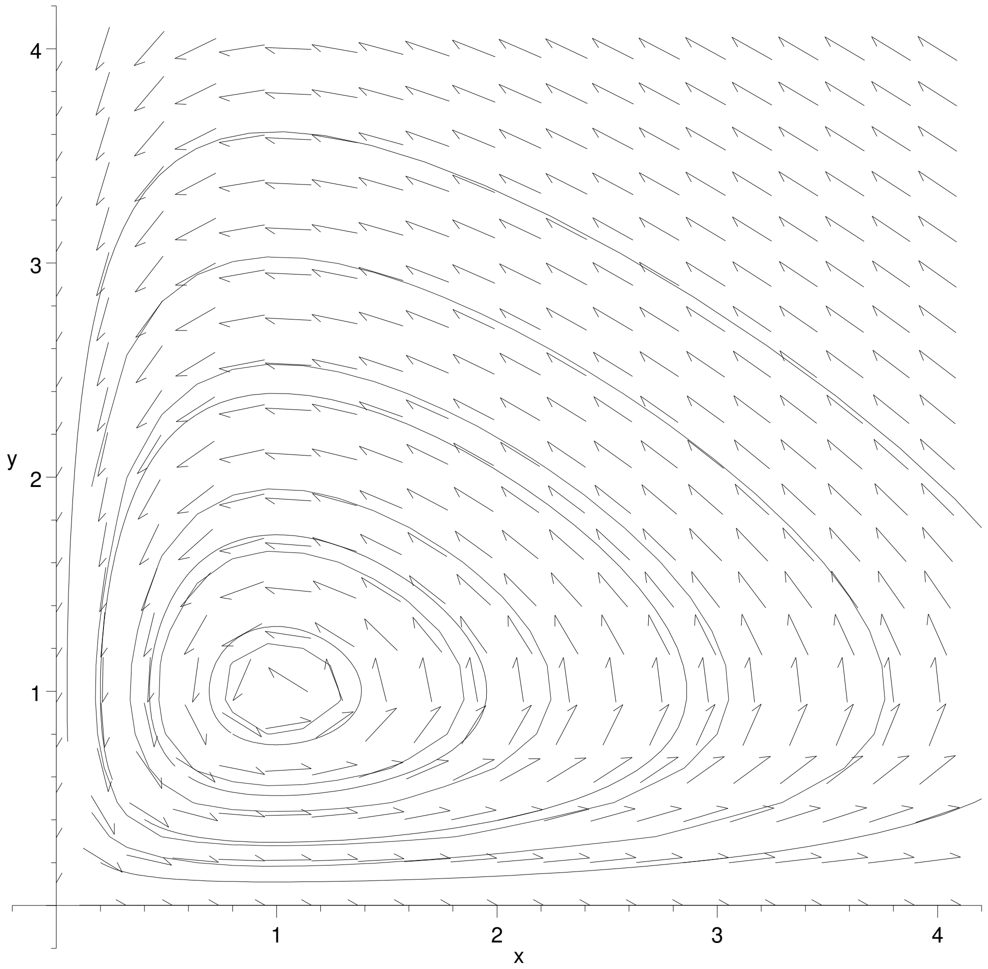



```
> coplot:=plots[contourplot](subs(r=0.8,H(x,y)),x=0..4,y=0..4,
        contours=[-4,-3.5,-3,-2.8,-2.6],colour=darkblue):
  coplot;
```



Let us plot this contourplot together with the earlier phase plane plot together:

```
> plots[display](deplot,coplot);
```



The trajectories follow these contourlines:
Put G(t)=H(x(t),y(t)), differentiate it, and substitute the dynamics...

```
> G:=unapply(H(x(t),y(t)),t);
```

$$G := t \to -\frac{\mathrm{x}(t)\,r^2 - \ln(\mathrm{x}(t))\,r^2 + \mathrm{y}(t) - \ln(\mathrm{y}(t))}{r^2}$$

```
> D(G)(t);
```

$$-\frac{\mathrm{D}(x)(t)\,r^2 - \dfrac{\mathrm{D}(x)(t)\,r^2}{\mathrm{x}(t)} + \mathrm{D}(y)(t) - \dfrac{\mathrm{D}(y)(t)}{\mathrm{y}(t)}}{r^2}$$

```
> subs(dynamics,%);
```

$$-\frac{r\,\mathrm{x}(t)\,(1-\mathrm{y}(t)) - r\,(1-\mathrm{y}(t)) + r\,\mathrm{y}(t)\,(\mathrm{x}(t)-1) - r\,(\mathrm{x}(t)-1)}{r^2}$$

```
> simplify(%(t));
```

$$0$$

Voila!