Implementation of Euler's Method for systems

Rabbits and Wolves

ODE:

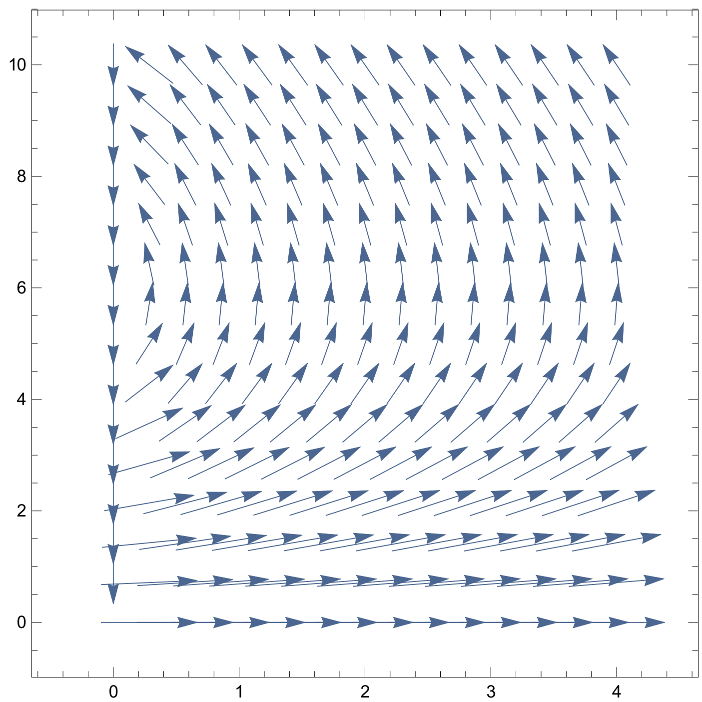
r’ = 3r - wr/2
w’ = -w/10 + 7rw/10

In[1]:=
```
vf[r_, w_] = {3 r - w r / 2, -w / 10 + 7 r w / 10}
```

Out[1]= $\left\{3\,r - \frac{r\,w}{2},\ -\frac{w}{10} + \frac{7\,r\,w}{10}\right\}$

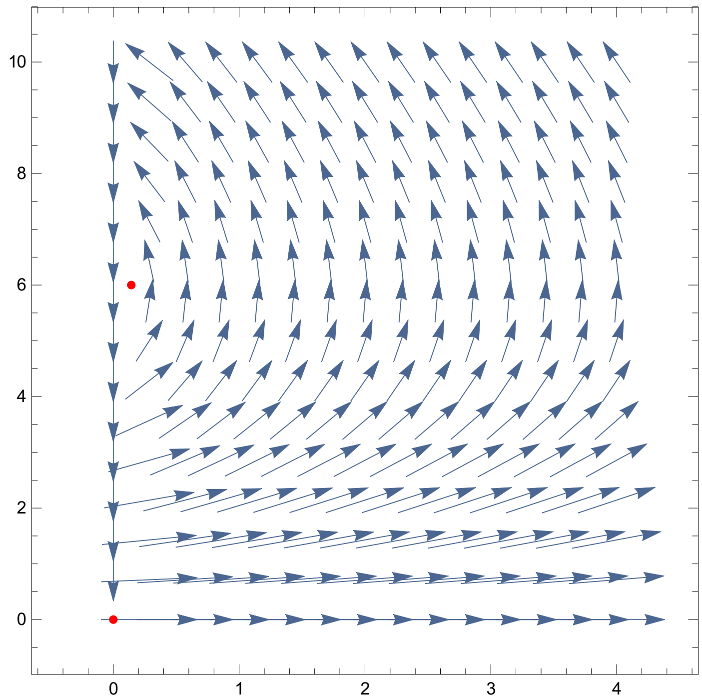
In[2]:=
```
df = VectorPlot[vf[r, w] / Norm[vf[r, w]],
  {r, 0, 4}, {w, 0, 10}, VectorStyle → Arrowheads[.03]]
```

Out[2]=

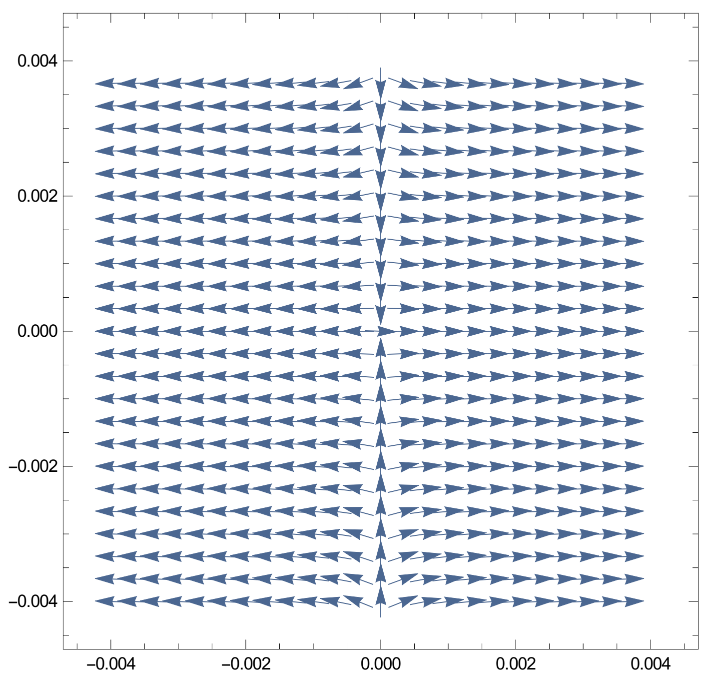


Equilibria: (0,0), (1/7,6)

In[3]:=
```
Show[df, ListPlot[{{0, 0}, {1 / 7, 6}}, PlotStyle → Red]]
```

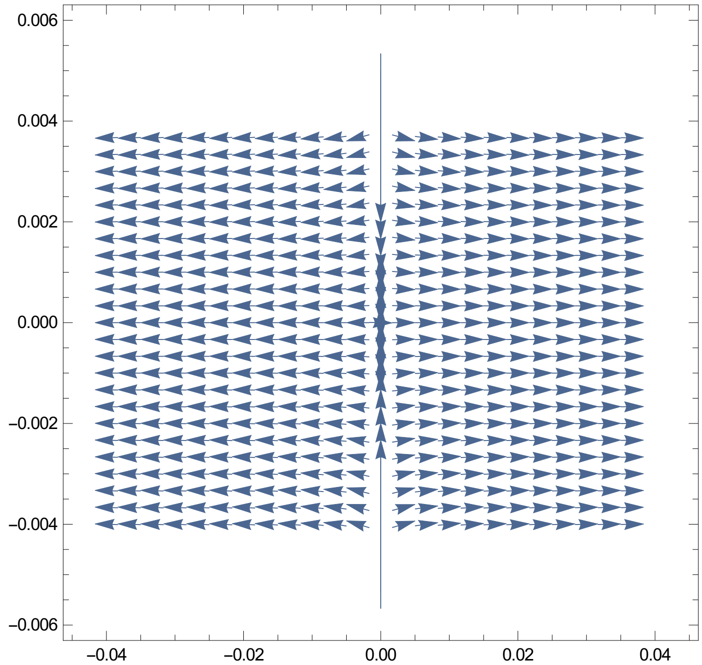
Out[3]= 

In[6]:=
```
dfcloseup1 = VectorPlot[vf[r, w] / Norm[vf[r, w]], {r, -0.004, 0.004},
  {w, -0.004, 0.004}, VectorStyle → Arrowheads[.03], VectorPoints → Fine]
```

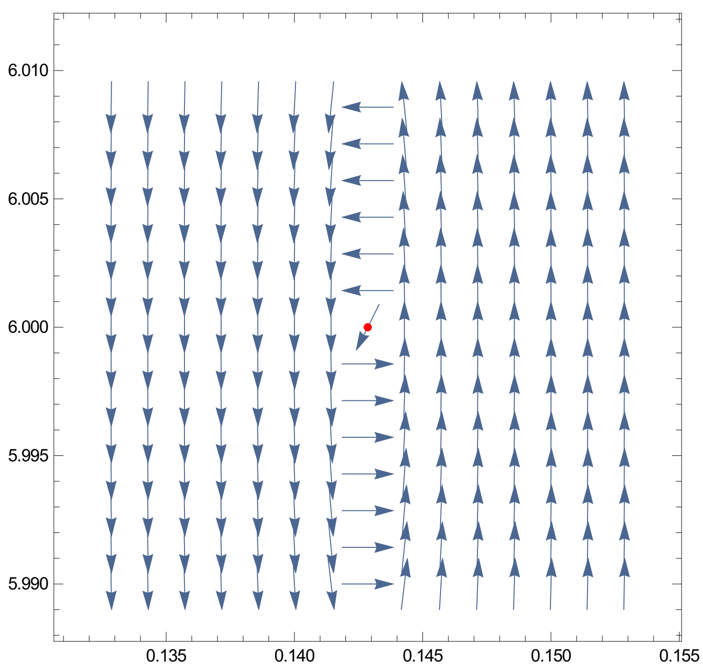
Out[6]= 

In[9]:=
```
VectorPlot[vf[r, w] / Norm[vf[r, w]], {r, -0.04, 0.04},
 {w, -0.004, 0.004}, VectorStyle → Arrowheads[.03], VectorPoints → Fine]
```

Out[9]=




```
Show[dfcloseup2 = VectorPlot[vf[r, w] / Norm[vf[r, w]],
   {r, 1 / 7 - 0.01, 1 / 7 + 0.01}, {w, 6 - 0.01, 6 + 0.01}, VectorStyle → Arrowheads[.03]],
 ListPlot[{{1 / 7, 6}}, PlotStyle → Red]]
```

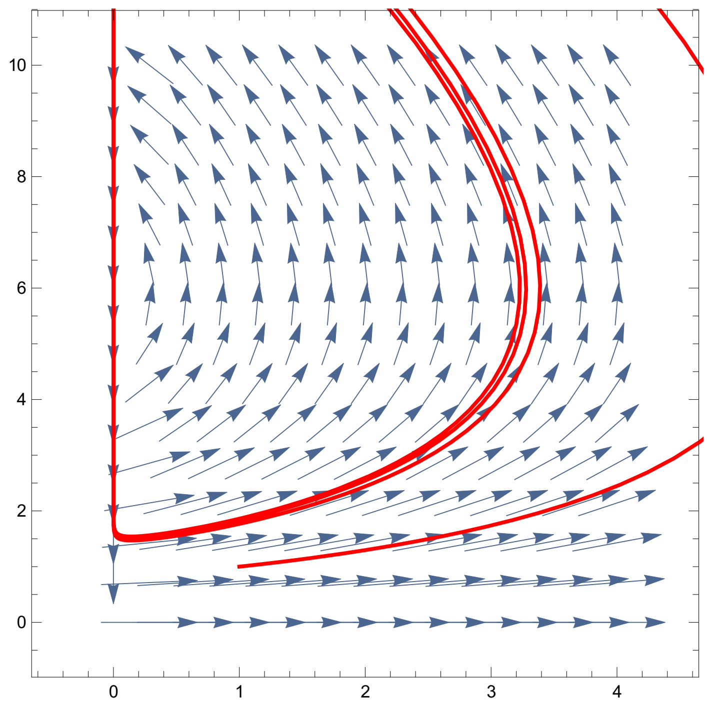


Well, both of these are quite unsatisfying pictures. I'll guess the nontrivial equilibrium point (1/7,6) is a spiral sink. Let's try to see if we can confirm that with a solution.

```
soln[rzero_?NumericQ, wzero_?NumericQ] :=
 NDSolve[{dr'[t] == 3 dr[t] - 0.5 dr[t] dw[t], dw'[t] == -0.1 dw[t] + 0.7 dw[t] dr[t],
   dr[0] == rzero, dw[0] == wzero}, {dr, dw}, {t, 0, 100}]
rr[t_?NumericQ, rzero_?NumericQ, wzero_?NumericQ] :=
 dr[t] /. soln[rzero, wzero][[1]]
ww[t_?NumericQ, rzero_?NumericQ, wzero_?NumericQ] := dw[t] /. soln[rzero, wzero][[1]]
```

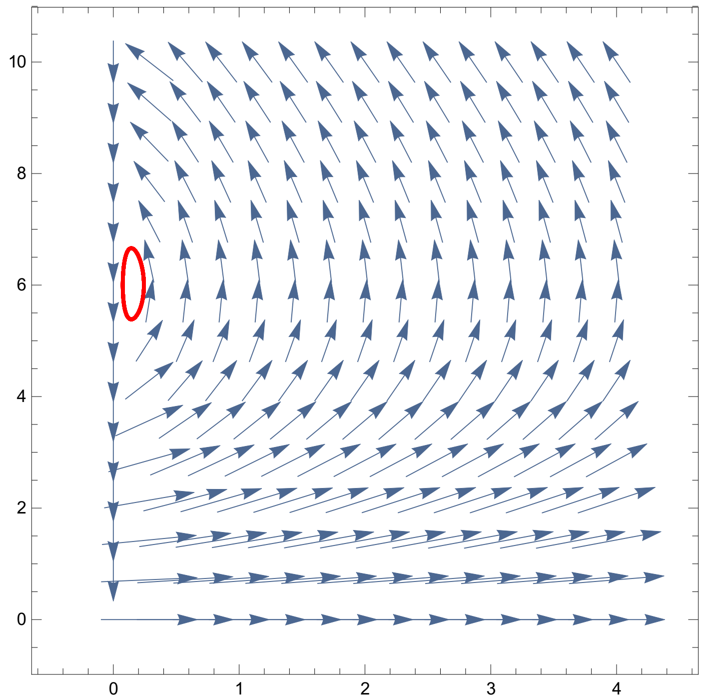
```
orbit1 =
  ParametricPlot[{rr[s, 1, 1], ww[s, 1, 1]}, {s, 0, 100}, PlotStyle → {Thick, Red}];
```

```
Show[df, orbit1]
```




```
orbit2 = ParametricPlot[{rr[s, 1 / 7 + 0.1, 6], ww[s, 1 / 7 + 0.1, 6]},
   {s, 0, 100}, PlotStyle → {Thick, Red}];
```

```
Show[df, orbit2]
```



My plots are inconclusive. One can see what turns up via linearization. If there really are periodic orbits, then I'll guess we have a Hamiltonian. In any case, let's go ahead and try to implement Euler's method. Since the ODE is autonomous, I'll take tzero = 0 and start with an arbitrary initial population distribution. (Why does this make sense?) I'll also restrict to the time interval [0,10]. Finally, I'll suppress the time, and compute the image in phase space.

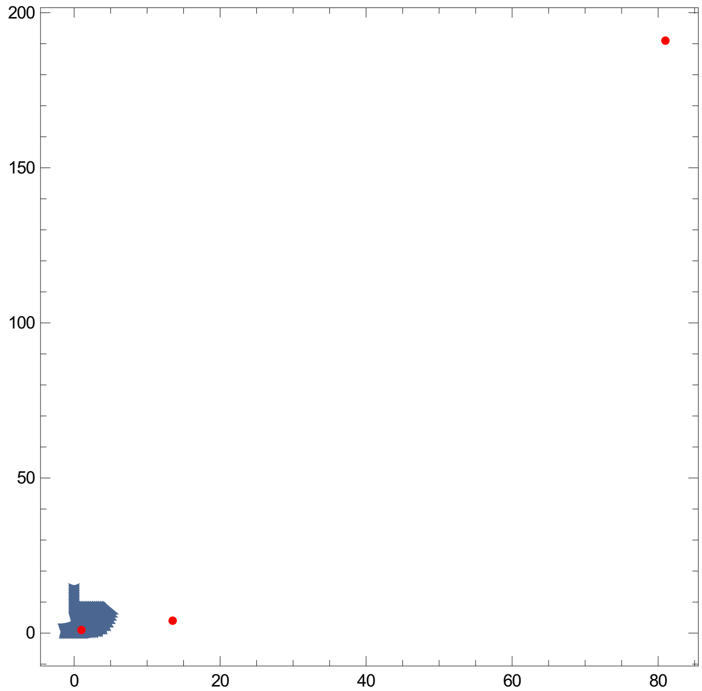
```
In[10]:= euler[xzero_?NumericQ, yzero_?NumericQ, N_?NumericQ] :=
  For[j = 1; image = {{xzero, yzero}}, j < N + 1, j ++,
   image = Append[image, image[[j]] + vf[image[[j]][[1]], image[[j]][[2]]] (10 / N)]]
```

```
euler[1, 1, 2]
```

```
image
```

$\left\{\{1, 1\}, \left\{\frac{27}{2}, 4\right\}, \{81, 191\}\right\}$

```
show2 = Show[df, ListPlot[image, PlotStyle → Red], PlotRange → All]
```

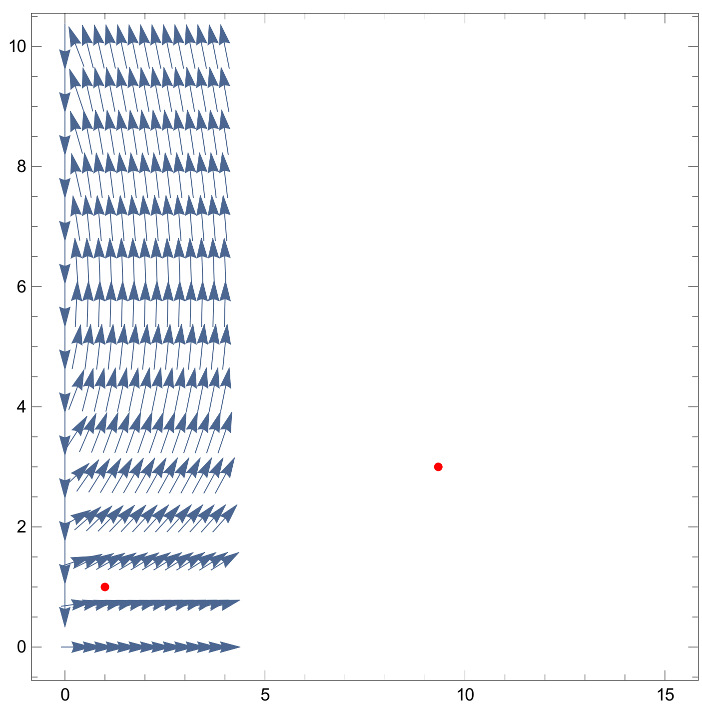



```
euler[1, 1, 3]
```

```
image
```

$$\left\{\{1, 1\}, \left\{\frac{28}{3}, 3\right\}, \left\{56, \frac{202}{3}\right\}, \left\{-\frac{51016}{9}, \frac{79588}{9}\right\}\right\}$$

```
show3 = Show[df, ListPlot[image, PlotStyle → Red], PlotRange → {{0, 15}, {0, 10}}]
```

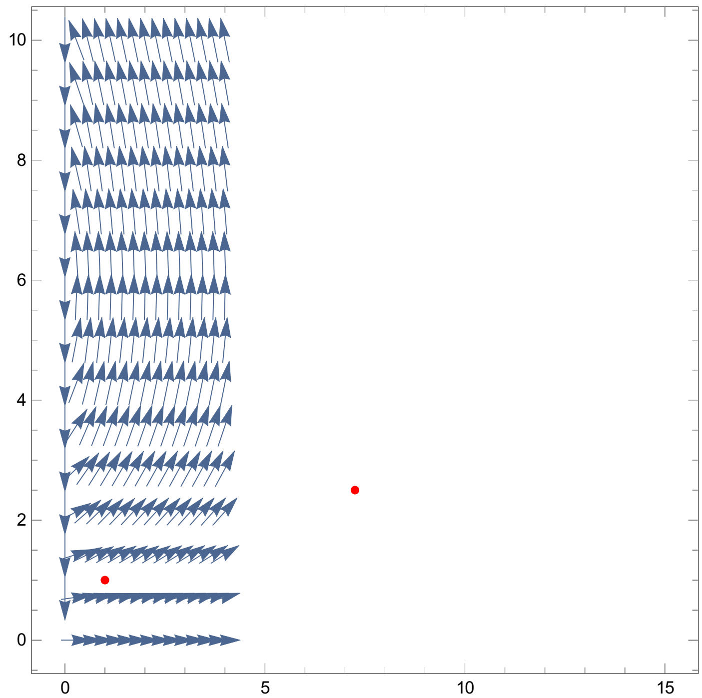


```
euler[1, 1, 4]

image
```

$$\left\{\{1, 1\}, \left\{\frac{29}{4}, \frac{5}{2}\right\}, \left\{\frac{1247}{32}, \frac{1075}{32}\right\}, \left\{-\frac{5\,345\,889}{4096}, \frac{9\,486\,875}{4096}\right\}, \left\{\frac{252\,834\,413\,648\,679}{67\,108\,864}, -\frac{354\,893\,890\,228\,125}{67\,108\,864}\right\}\right\}$$

```
show4 = Show[df, ListPlot[image, PlotStyle → Red], PlotRange → {{0, 15}, {0, 10}}]
```

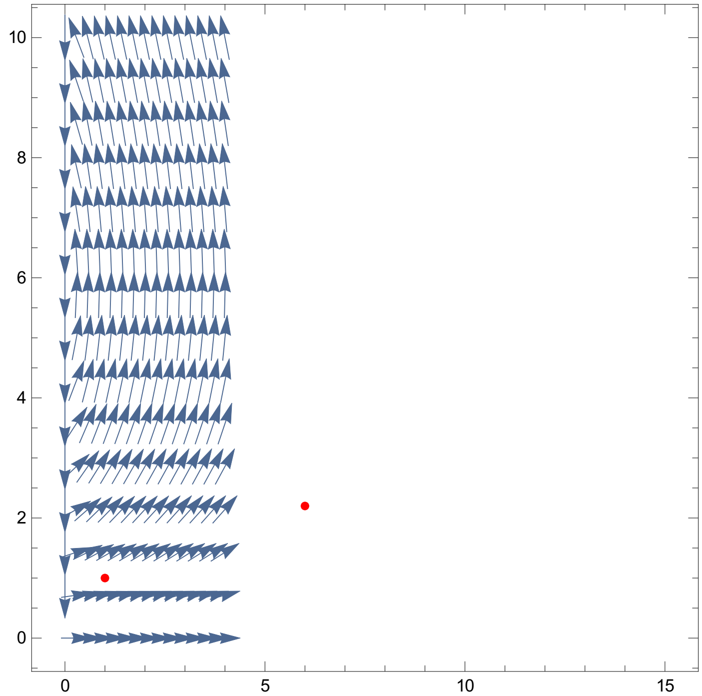



```
euler[1, 1, 5]

image
```

$$\left\{\{1, 1\}, \left\{6, \frac{11}{5}\right\}, \left\{\frac{144}{5}, \frac{506}{25}\right\}, \left\{-\frac{47\,664}{125}, \frac{520\,168}{625}\right\}, \left\{\frac{24\,584\,757\,552}{78\,125}, -\frac{173\,292\,928\,864}{390\,625}\right\}, \left\{\frac{4\,260\,431\,865\,543\,854\,030\,928}{30\,517\,578\,125}, -\frac{29\,822\,606\,645\,222\,229\,466\,496}{152\,587\,890\,625}\right\}\right\}$$

```
show5 = Show[df, ListPlot[image, PlotStyle → Red], PlotRange → {{0, 15}, {0, 10}}]
```



In[12]:=
```
euler[1, 1, 20]
```

In[17]:=
```
showend = Show[df, ListPlot[image, PlotStyle → Red], PlotRange → {{0, 15}, {0, 10}}]
```

Out[17]=