Exercise 3.3

task a)

In[351]:=

```
ClearAll["Global`*"]
xdot[x_, y_, μ_] := μ * x + y - x^2;
ydot[x_, y_, μ_] := -x + μ * y + 2 * x^2;

(*Use this to find the fixed points so i
 can determine initial condition for trajectory.*)
fixedPoints = Solve[{xdot[x, y, μ] == 0, ydot[x, y, μ] == 0}, {x, y}]
```

Out[354]=

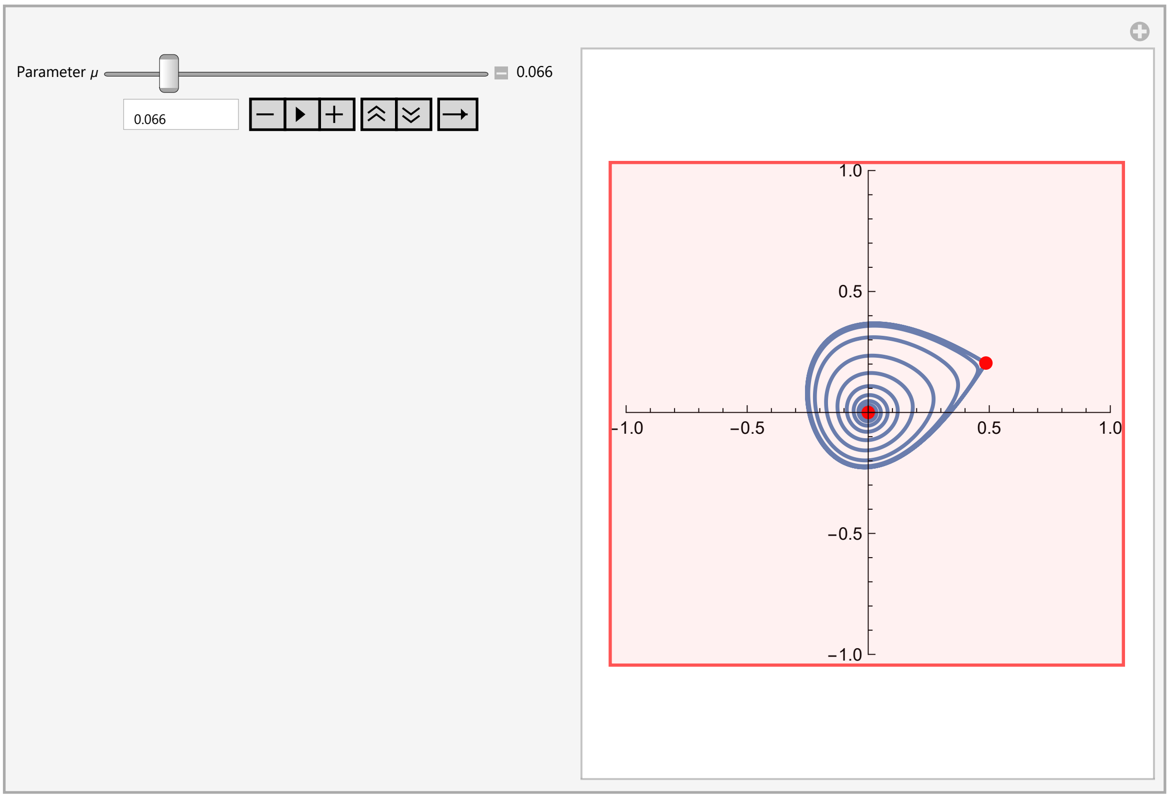
$$\left\{\{x \to 0, y \to 0\}, \left\{x \to \frac{1+\mu^2}{2+\mu}, y \to \frac{1-2\mu+\mu^2-2\mu^3}{(2+\mu)^2}\right\}\right\}$$

Found fixed points.
Plot trajectory for both and determine which to use.
Homoclinic bifurcation occur when the limit cycle collide with the saddle point.
So, refine μ until we see that the flow escapes.

In[355]:=

```
ClearAll["Global`*"]
(*Trajectory dont show if i start at (0,0), choose IC close*)
Manipulate[Module[{xdot, ydot, fixedPoints, traj, plotRange, sol},
  xdot[x_, y_] := μ * x + y - x^2;
  ydot[x_, y_] := -x + μ * y + 2 * x^2;
  fixedPoints = NSolve[{xdot[x, y] == 0, ydot[x, y] == 0}, {x, y}, Reals];
  sol = NDSolve[{
      x'[t] == xdot[x[t], y[t]],
      y'[t] == ydot[x[t], y[t]],
      x[0] == 0.01,
      y[0] == 0.01},
     {x, y}, {t, 0, 100}];

  plotRange = 1;

  Show[ParametricPlot[{x[t], y[t]} /. sol, {t, 0, 100},
    PlotStyle → Thick,
    PlotRange → {{-plotRange, plotRange}, {-plotRange, plotRange}}] ,
   Graphics[{Red, PointSize[Large], Point[{x, y} /. fixedPoints]}]]
 ],
 {{μ, 0.5, "Parameter μ"}, 0, 0.5, 0.01,
  Appearance → "Labeled"}]
```

Out[356]=



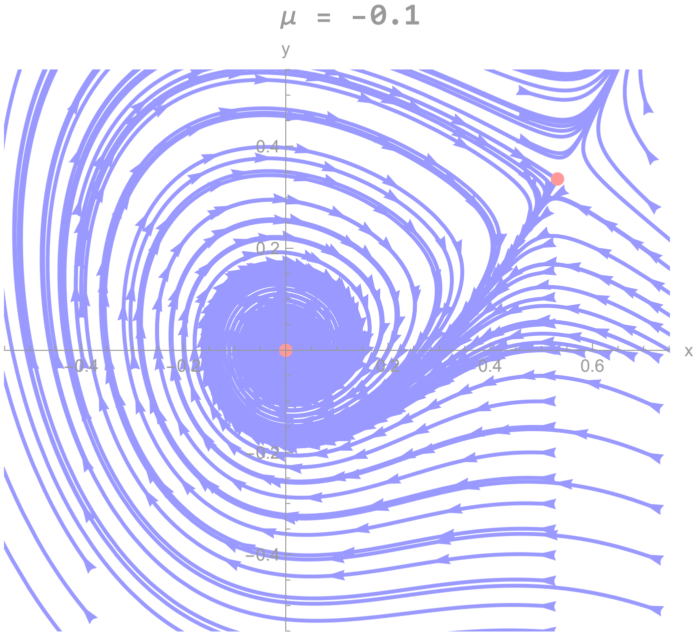
NDSolve: Error test failure at t == 59.55615430053791`; unable to continue.

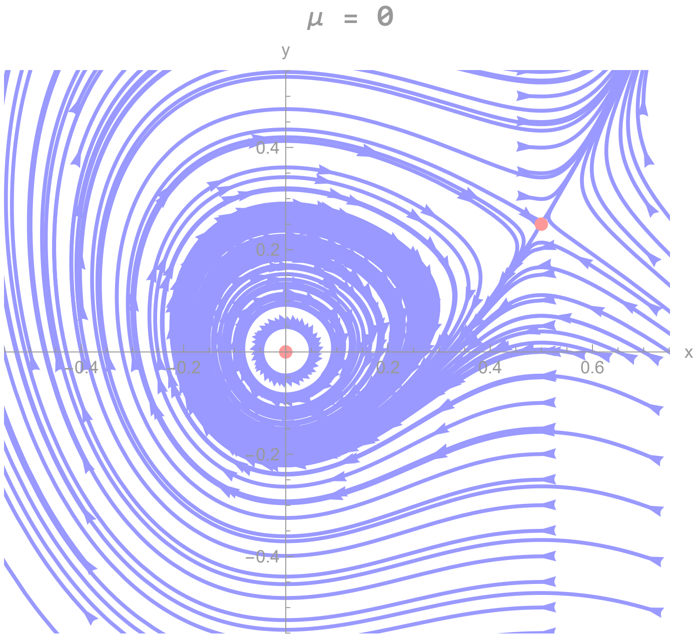
NDSolve: Error test failure at t == 59.55615430053791`; unable to continue.

task b)

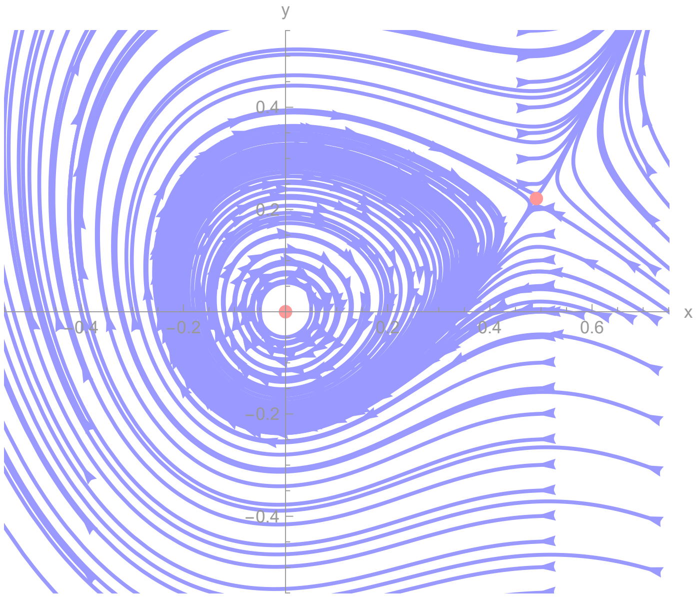
```
ClearAll["Global`*"]
muValues = {-0.1, 0, 0.04, 0.066, 0.1};

xdot[x_, y_, μ_] := μ * x + y - x^2;
ydot[x_, y_, μ_] := -x + μ * y + 2 * x^2;

initConditions = Join[
   Table[{x, 0}, {x, -0.5, 1, 0.05}],
   Table[{0.7, y}, {y, -0.6, 0.6, 0.1}],
   Table[{0.49, y}, {y, -0.6, 0.6, 0.05}],
   Table[{-0.4, y}, {y, -0.6, 0.6, 0.1}]];

(*Solve and plot for each Mu value*)
arrows = Table[{0.03, t}, {t, 0, 1, 0.05}];
plots = Table[Module[{sol, fixedPoints, arrows},
     sol = Table[NDSolve[{
          x'[t] == xdot[x[t], y[t], μ],
          y'[t] == ydot[x[t], y[t], μ],
          x[0] == ic[[1]],
          y[0] == ic[[2]]},
         {x, y}, {t, 0, 20}],
        {ic, initConditions}];

     fixedPoints = NSolve[{xdot[x, y, μ] == 0, ydot[x, y, μ] == 0}, {x, y}];


     Labeled[
      Show[
       ParametricPlot[Evaluate[{x[t], y[t]} /. sol], {t, 0, 20},
        PlotRange → {{-0.55, 0.75}, {-0.55, 0.55}},
        PlotStyle → Blue,
        AxesLabel → {"x", "y"},
        ImageSize → Medium],
       Graphics[{Red, PointSize[Large], Point[{x, y} /. fixedPoints]}]
      ],
      Style[Row[{"\[\Mu] = ", μ}], Bold, 16], Top]],
    {μ, muValues}] /. Line[x_] :→ {Arrowheads[arrows], Arrow[x]};

(*Display all plots in a column*)
Column[plots, Spacings → 2]
```

Out[429]=

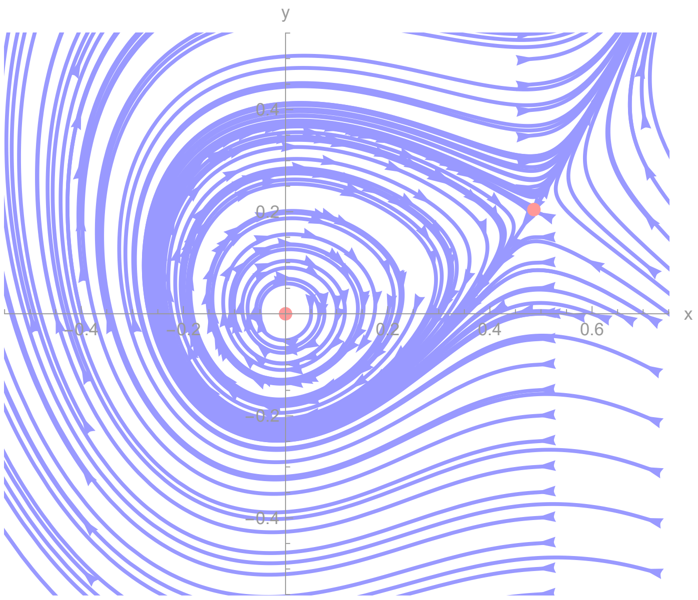


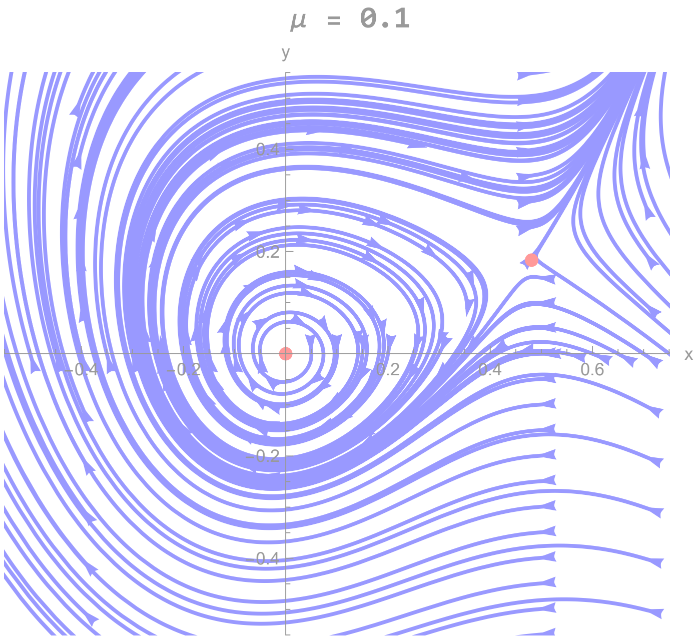


μ = 0.04



μ = 0.066





Determined visually by placing a trajectory very close to the FP in (0,0), the stability is a stable spiral for $\mu < 0$, center for $\mu = 0$, unstable spiral for $\mu > 0$.

The other FP is a saddle point for any μ.

Homoclinic orbit occur in the case when the limit cycle collide with the saddle point, at $\mu = 0.066$. For $\mu = 0$ and $0<\mu<0.66$ a limit cycle can be seen where all the trajectories approach, i.e, the thick blue orbit. Also for $\mu = 0$ we have closed orbits inside the limit cycle due to the stability of the FP being a center.

For $\mu > 0.066$ we have no closed orbits or limit cycles since they vanish during the homoclinic bifurcation.

task c)

```
xdot[x_, y_, μ_] := u*x
ydot[x_, y_, μ_] := s*y
```

Using the hint from the problem session:

$$\frac{dx}{dt} = u\,x \Rightarrow \frac{dx}{x} = u\,d\,t$$

Now we integrate the x direction from γ to 1, since x(t1) = 1, and t from 0 to t1. Thus we have:

$$\int_{\gamma}^{1} (1/x)\, dx = \int_{0}^{t1} u\, d\mathrm{dt} \Rightarrow \ln(1) - \ln(\gamma) = u^{*}t1 \Rightarrow t1 = \frac{-ln(\gamma)}{u}$$

task d)

In[367]:=

```
ClearAll["Global`*"]
xdot[x_, y_, μ_] := μ * x + y - x^2
ydot[x_, y_, μ_] := -x + μ * y + 2 * x^2

jacobi[x_, y_, μ_] = D[{xdot[x, y, μ], ydot[x, y, μ]}, {{x, y}}];

fp = NSolve[{xdot[x, y, μ] == 0, ydot[x, y, μ] == 0}, {x, y}];

stabMatrix = jacobi[x /. fp[[2, 1]], y /. fp[[2, 2]], μ];

u = Eigenvalues[stabMatrix][[2]] (*Unstable is > 0*)
s = Eigenvalues[stabMatrix][[1]]
```

Out[373]=

$$\frac{-2. + 4.\ \mu + \sqrt{20. + 36.\ \mu^2 + 16.\ \mu^3 + 4.\ \mu^4}}{2\ (2. + \mu)}$$

Out[374]=

$$\frac{-2. + 4.\ \mu - \sqrt{20. + 36.\ \mu^2 + 16.\ \mu^3 + 4.\ \mu^4}}{2\ (2. + \mu)}$$

task e)

In[375]:=

```
ClearAll["Global`*"]
μc = 0.066;
xdot[x_, y_, μ_] := μ * x + y - x^2
ydot[x_, y_, μ_] := -x + μ * y + 2 * x^2

(*Saddle fp is on the second position.*)
fixedPoint[μ_] = NSolve[{xdot[x, y, μ] == 0, ydot[x, y, μ] == 0}, {x, y}][[2]];

μs = Table[Abs[μc - ϵ], {ϵ, 0.01, μc, 0.005}];

(*Choose IC close to (0,0) to make sure we are within the orbit for all mu.*)
initialConditions = Join[
   Table[{y, 0}, {y, -0.1, 0.1, 0.01}],
   Table[{x, -0.1}, {x, -0.1, 0.1, 0.01}],
   Table[{x, 0}, {x, -0.1, 0.1, 0.01}],
   Table[{x, 0.1}, {x, -0.1, 0.1, 0.01}]];

trajectories = Table[
   Module[{fp, sol},
    fp = fixedPoint[μ];
    fpx = x /. fp[[1]];
    fpy = y /. fp[[2]];
    sol = Table[NDSolve[{
        x'[t] == xdot[x[t], y[t], μ],
        y'[t] == ydot[x[t], y[t], μ],
```

```
          x[0] == ic[[1]],
          y[0] == ic[[2]]},
         {x, y}, {t, 0, 100}],
        {ic, initialConditions}];
      Min[
       Table[Sqrt[(fpx - x[t])^2 + (fpy - y[t])^2] /. sol, {t, 0, 100, 0.1}]
      ]
     ], {μ, μs}];

lnm = Log[Abs[μs - μc]];
lng = Log[trajectories];

data = Transpose[{lnm, lng}];
lm = LinearModelFit[data, x, x]

(*Extract the slope (scaling exponent a)*)
a = lm["BestFitParameters"][[2]]

Show[
 ListPlot[
  data,
  PlotStyle → Red,
  PlotRange → Full,
  AxesLabel → {"ln|μ - μc|", "lnγ"},
  Frame → True],
 Plot[lm[x], {x, Min[lnm], Max[lnm]},
  PlotRange → Full,
  PlotStyle → Blue,
  Frame → True]]
```

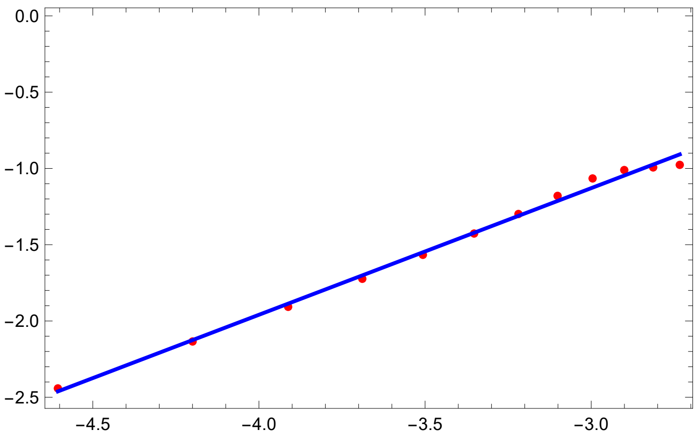
Out[386]=

FittedModel[1.36 + 0.83 x]

Out[387]=

0.830114

Out[388]=



In[389]:=