Lecture 28: Coda — Analysis of a System

MATH 303 ODE and Dynamical Systems

Konstantinos Efstathiou

Brusselator

We consider the following planar system

$$
\begin{aligned}
x' &= 1 - (b+1)x + x^2 y \\
y' &= bx - x^2 y
\end{aligned}
$$

We want to analyze the dynamics of the system as the parameter b changes.

Equilibria

If we have a mechanical system then we start by looking at the energy (the conserved quantity). This is not the case here. For general planar systems we start the analysis by looking at equilibria.

The equilibria here satisfy the equations

$$\begin{aligned} 1 - (b+1)x + x^2y &= 0 \\ bx - x^2y &= 0 \end{aligned}$$

We start with the second equation and we note that it factors as $x(b - xy) = 0$.

One solution is $x = 0$ and substituting into the first equation we find $1 = 0$, so there is no equilibrium with $x = 0$.

The second possibility is $xy = b$. Using this to substitute xy in the first equation

$$1 - (b+1)x + x^2y = 0$$

we get

$$1 - (b+1)x + bx = 0$$

that is, $x = 1$. From $xy = b$ we then get $y = b$.

Therefore, the only equilibrium is $(x_0, y_0) = (1, b)$.

Linearization and stability

We now look at the linearization of the system at the equilibrium. The Jacobian matrix is

$$D\mathbf{F}(x,y) = \begin{bmatrix} -b-1+2xy & x^2 \\ b-2xy & -x^2 \end{bmatrix}.$$

Therefore, at the equilibrium we have

$$A := D\mathbf{F}(1,b) = \begin{bmatrix} b-1 & 1 \\ -b & -1 \end{bmatrix}.$$

We have $\det A = 1 > 0$ and $\operatorname{tr} A = b - 2$.

Placing these values on the trace-determinant plane we find a straight horizontal line at $\det A = 1$.

The equilibrium is stable when $\operatorname{tr} A < 0$, that is, when $b < 2$.

It is unstable when $\operatorname{tr} A > 0$, that is, when $b > 2$.

At $b = 2$ we have $\operatorname{tr} A = 0$.

Moreover, the eigenvalues are complex when $\det A > \frac{1}{4}(\operatorname{tr} A)^2$, that is, when $1 > \frac{1}{4}(b-2)^2$. It is not difficult to see that this corresponds to $0 < b < 4$.

Summarizing:

- for $b < 0$ the equilibrium is stable node;

- for $0 < b < 2$ the equilibrium is stable spiral;

- for $2 < b < 4$ the equilibrium is unstable spiral — note here that the transition at $b = 2$ is a strong indication of a Hopf bifurcation;

- for $4 < b$ the equilibrium is unstable node.

Hopf bifurcation

The fact that the stability changes from stable spiral to unstable spiral at $b = 2$ is a strong indication that a Hopf bifurcation takes place.

To locate the limit cycle and the type of the bifurcation (supercritical or subcritical) we numerically compute some solutions curves.

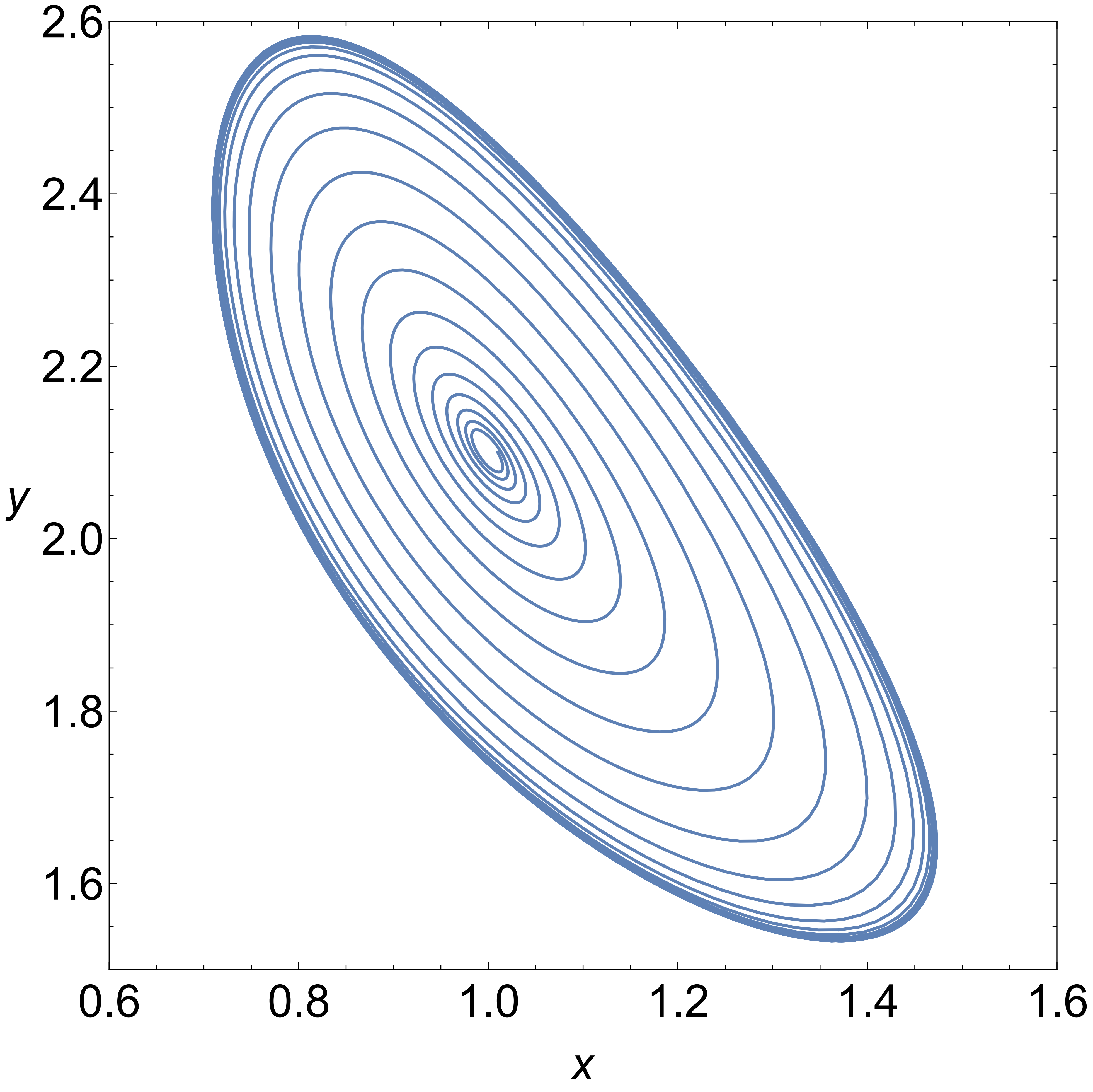
First we compute a solution curve starting near the equilibrium for $b > 2$ and close to $b_0 = 2$. For example, we can take $b = 2.1$. Note here that typically the size of the limit cycle grows as $\sqrt{|b - b_0|}$. Therefore, if we take $b = 2.1$ we expect to find a limit cycle of "radius" $\approxeq 0.3$. This estimate can guide us in choosing the initial condition for the solution curve. Here we can take $x = 1.01,\ y = b$.

Moreover, notice that since for $b > 2$ the equilibrium is an unstable spiral, then the limit cycle (if it exists) will be stable. Therefore, if the cycle exists, integrating forward in time the solution will eventually reach it.

This can be done with the following code.

```
ParametricPlot[
 Evaluate[
  NDSolveValue[
   With[{b = 2.1},
    x'[t] == 1 - (b + 1) x[t] + x[t]^2 × y[t]
     && y'[t] == b x[t] - x[t]^2 × y[t]
     && x[0] == 1.01
     && y[0] == b],
   {x[t], y[t]},
   {t, 0, 200}]],
 {t, 0, 200},
 Frame → True, PlotRange → {{0.6, 1.6}, {1.5, 2.6}}, FrameStyle → Directive[Black, 24],
 ImageSize → Large, AspectRatio → 1, FrameLabel → {x, y}, RotateLabel → False]
```

The result is shown in the following picture where we observe that indeed the solution curve reaches a limit cycle. The limit cycle is stable and therefore we have a supercritical Hopf bifurcation.

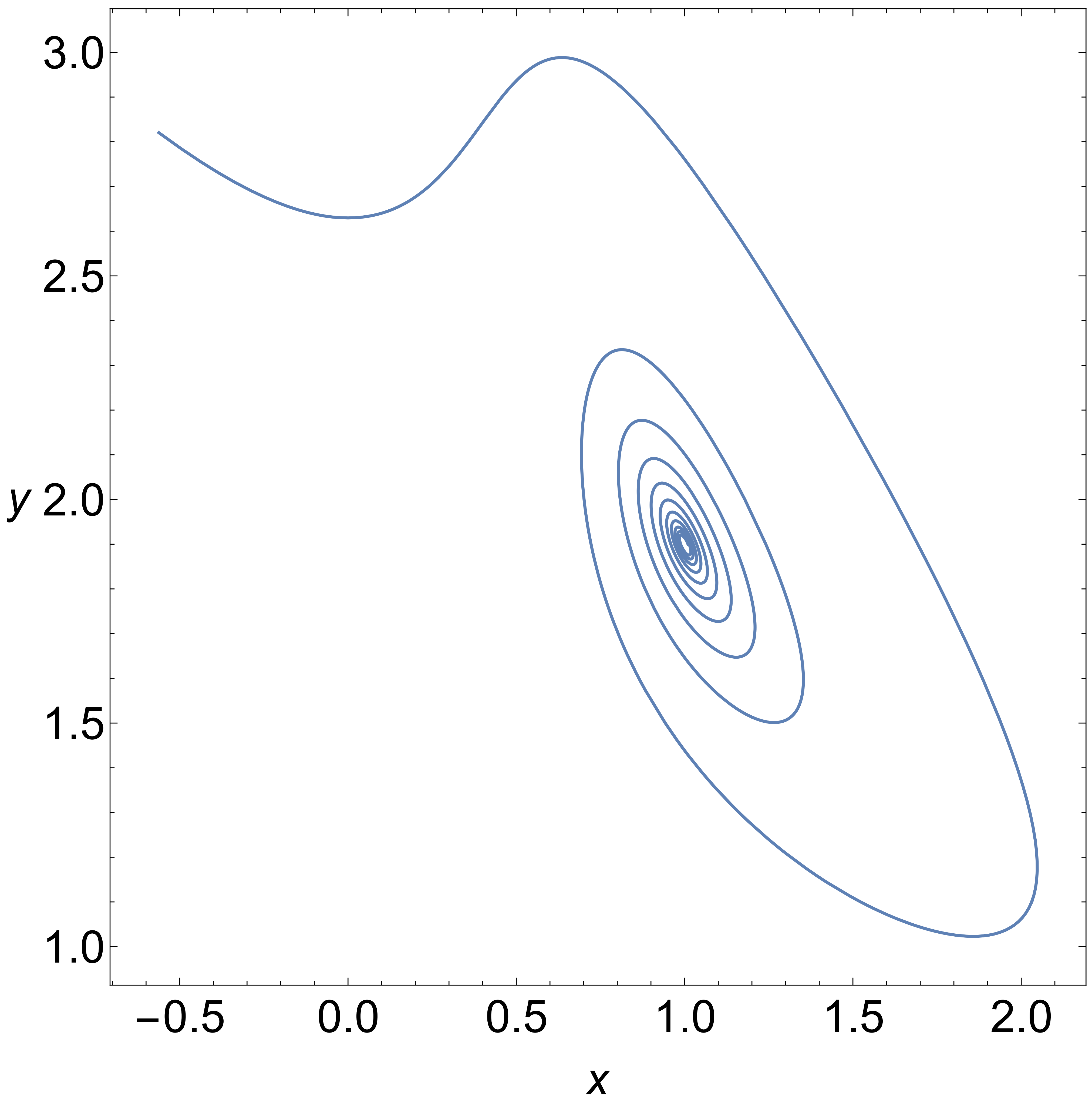


For completeness we also check the case $b < 2$. Here, take $b = 1.9$. If a limit cycle exists, it must be unstable (since the equilibrium is a stable spiral). Therefore, it can be reached by a solution curve going backward in time.

If we consider the initial condition $x = 1.01, y = b$ and we integrate backward in time then we get

```
NDSolveValue: At t == -66.2362, step size is effectively zero; singularity or stiff system suspected.
```

If we integrate until $t = -66$ we get the solution curve in the following slide which shows that there is no limit cycle for $b = 1.9$. The code for this is also shown in the following slide.

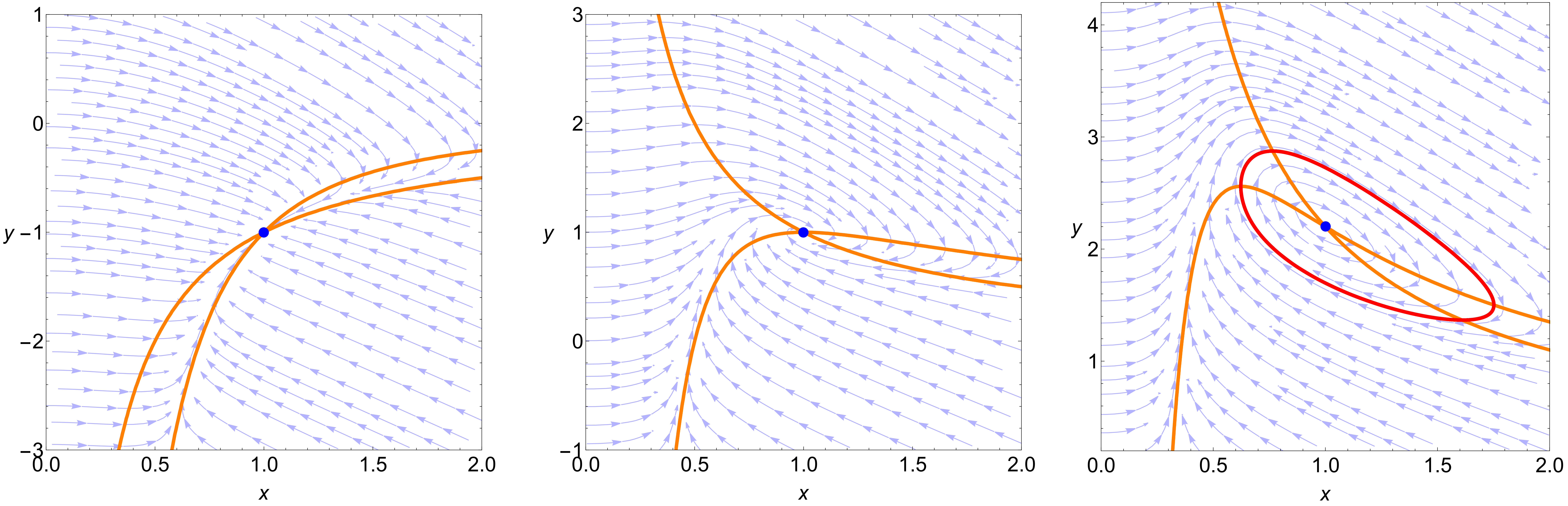



```
ParametricPlot[
 Evaluate[
  NDSolveValue[
   With[{b = 1.9},
    x'[t] == 1 - (b + 1) x[t] + x[t]^2 × y[t]
     && y'[t] == b x[t] - x[t]^2 × y[t]
     && x[0] == 1.01
     && y[0] == b],
   {x[t], y[t]},
   {t, 0, -66}]],
 {t, 0, -66},
 Frame → True, PlotRange → All, ImageSize → Large,
 FrameStyle → Directive[Black, 24], AspectRatio → 1,
 FrameLabel → {x, y}, RotateLabel → False]
```

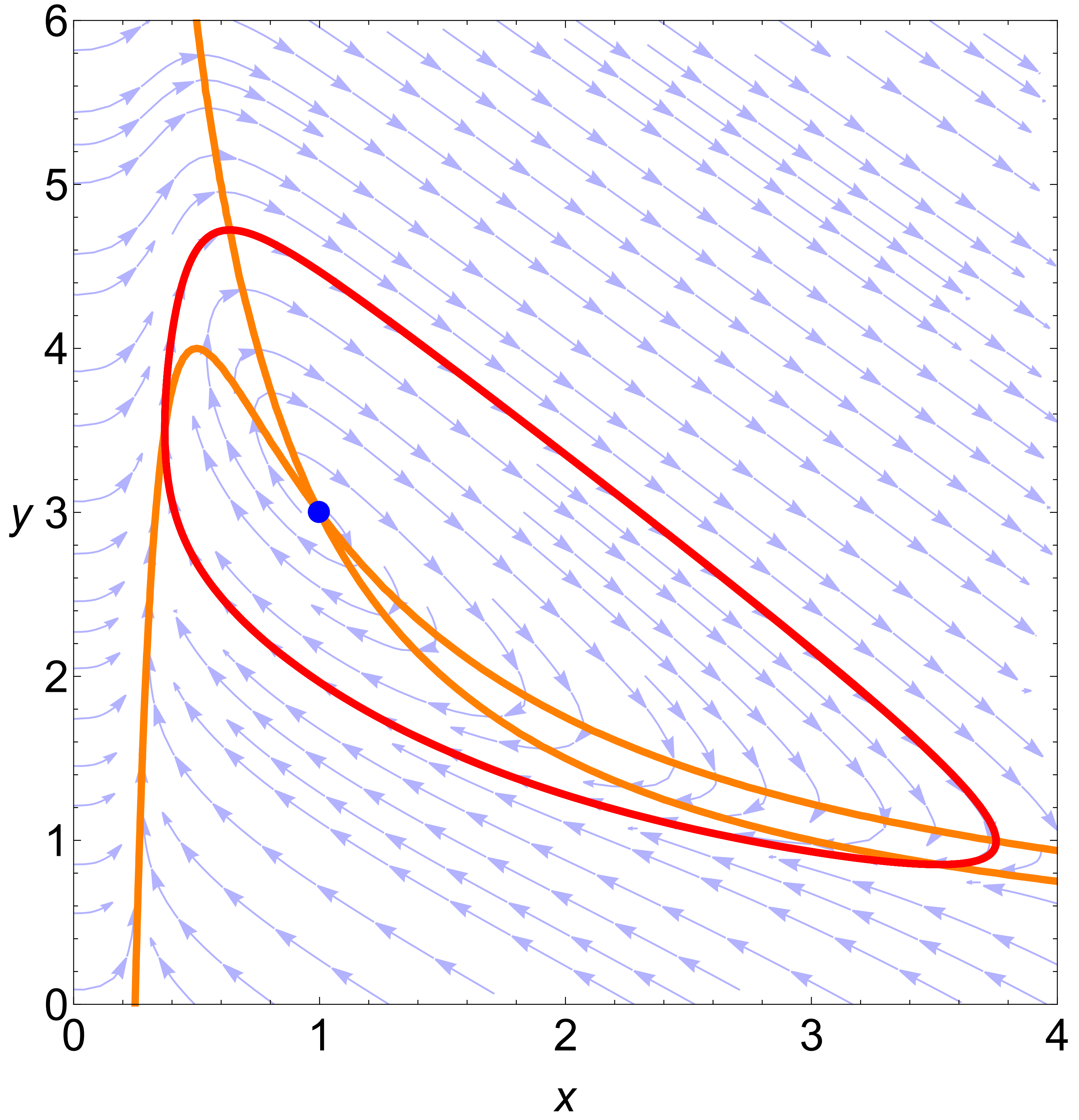
Phase portraits

Having done the analysis of the system we can now draw phase portraits for different values of b corresponding to different qualitative regions.

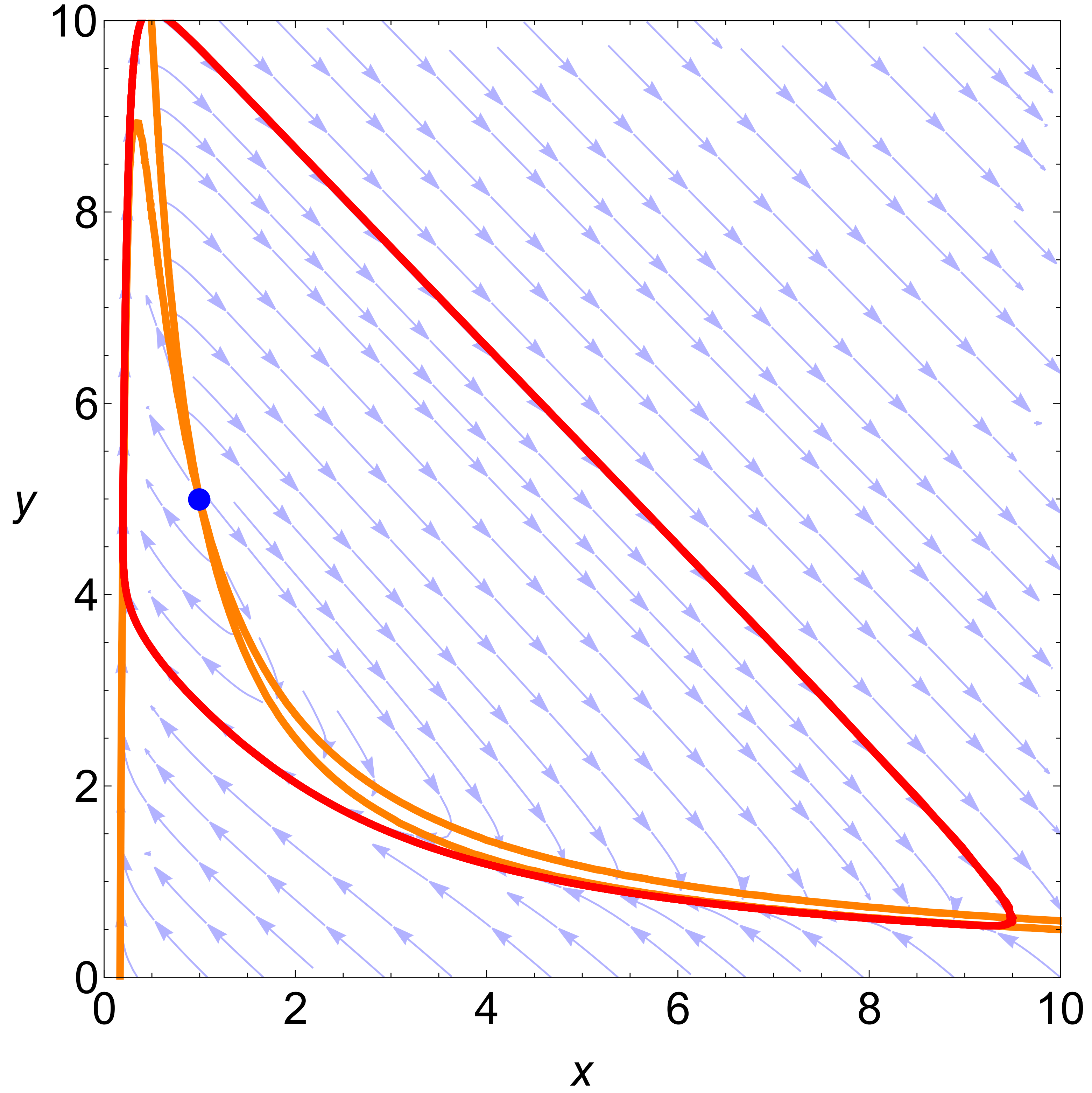
The phase portraits show the "stream plot", the nullclines (that is, the curves defined by $x' = 0$ or by $y' = 0$), the equilibrium (which is the intersection point of the two isoclines), and the limit cycle (if it exists).



$b = -1$ $\qquad$ $b = 1$ $\qquad$ $b = 2.2$



$b = 3$



$b = 5$

Tracking the limit cycle

We have seen that the limit cycle is created at $b = 2$. Often we need to be able to be able to find a point on the limit cycle that we can use as an initial condition to numerically compute the whole cycle.

For this type of computation the Poincaré map is very useful. This is an autonomous system. Therefore, in this case we consider a surface of section. There are many choices that can work. Let's take $y = b$ with $x > 1$. Looking at the phase portraits in the previous two slides shows that the limit cycle intersects this half line exactly once.

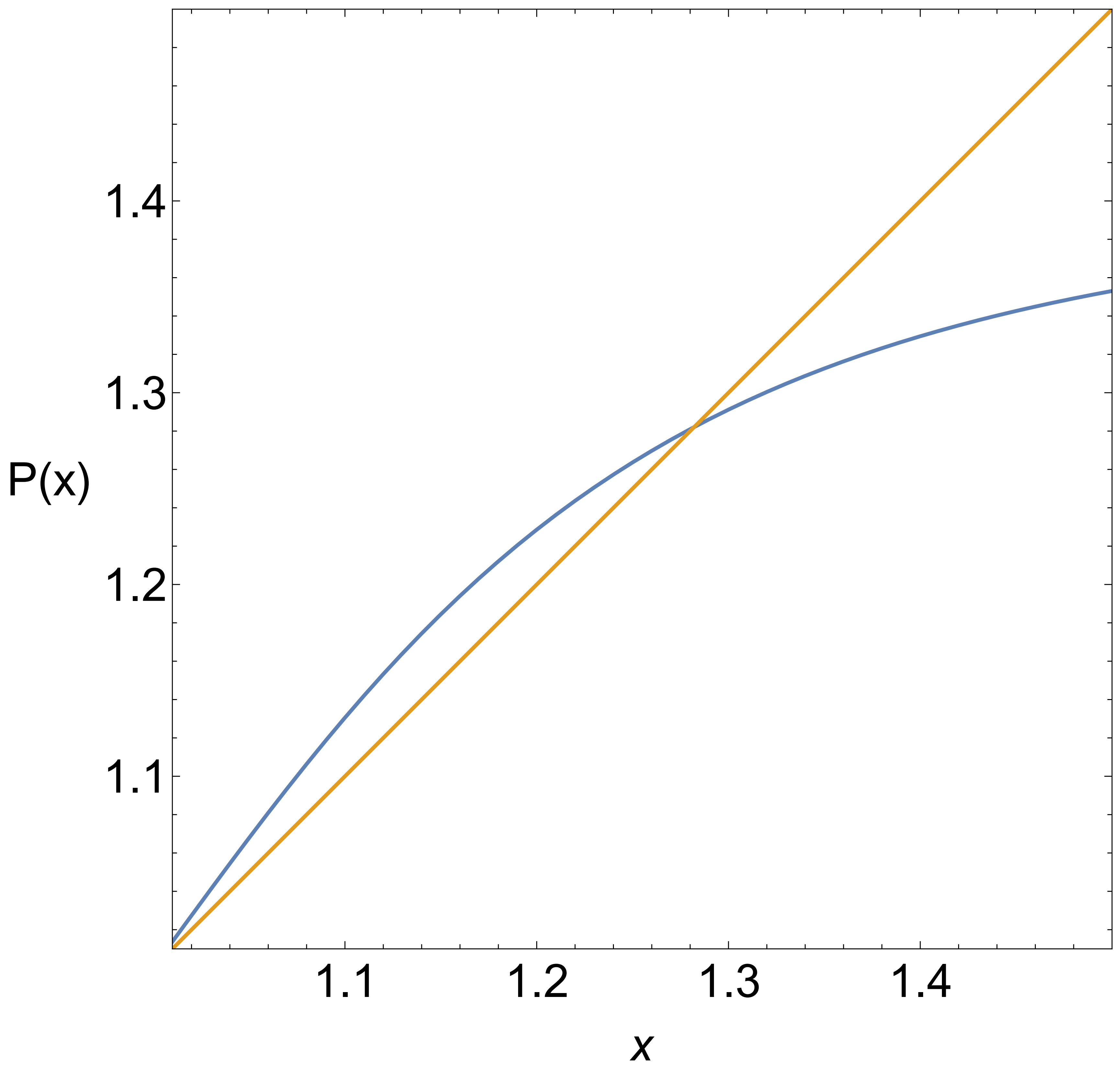
The problem is to find the point where the limit cycle intersects the half line $y = b$ with $x > 1$.

If we consider an initial condition (x, b) on the chosen half line then we can integrate the system until the solution hits again the half line at a point $(P(x), b)$. This defines the Poincaré map $x \mapsto P(x)$.

In this system we cannot compute P analytically and we study it numerically. In Mathematica this can be implemented using the following function (compare this with the function for the Hénon-Heiles system).

```
pmap[b_?NumberQ][x0_?NumberQ] := Module[{stopTime, x, y},
  NDSolveValue[
   x'[t] == 1 - (b + 1) x[t] + x[t]^2 × y[t]
    && y'[t] == b x[t] - x[t]^2 × y[t]
    && x[0] == x0
    && y[0] == b
    && WhenEvent[y[t] == b && x[t] > 1, stopTime = t; "StopIntegration"],
   x[stopTime], {t, 0, Infinity}]]
```

The graph of the Poincaré map $P(x)$ together with the graph of the identity function $y = x$ is shown below.



Then the limit cycle corresponds to fixed points of the Poincaré map, that is, points x_0 such that $P(x_0) = x_0$. These can be found numerically using the function FindRoot.

For example, the command

```
FindRoot[pmap[2.1][x0] == x0, {x0, 1.3}]

{x0 → 1.28209}
```

starts with the guess $x_0 = 1.3$ for the fixed point and finds the more accurate value $x_0 = 1.28209$. To find the dependence of x_0 on b we need to repeat this process for several values of b. For example we can start at $b = 2.01$ and continue until $b = 4$ with a step 0.01. The guess for each value of b can be taken as the x_0 we found for the previous value of b.

This can be done with the following code.

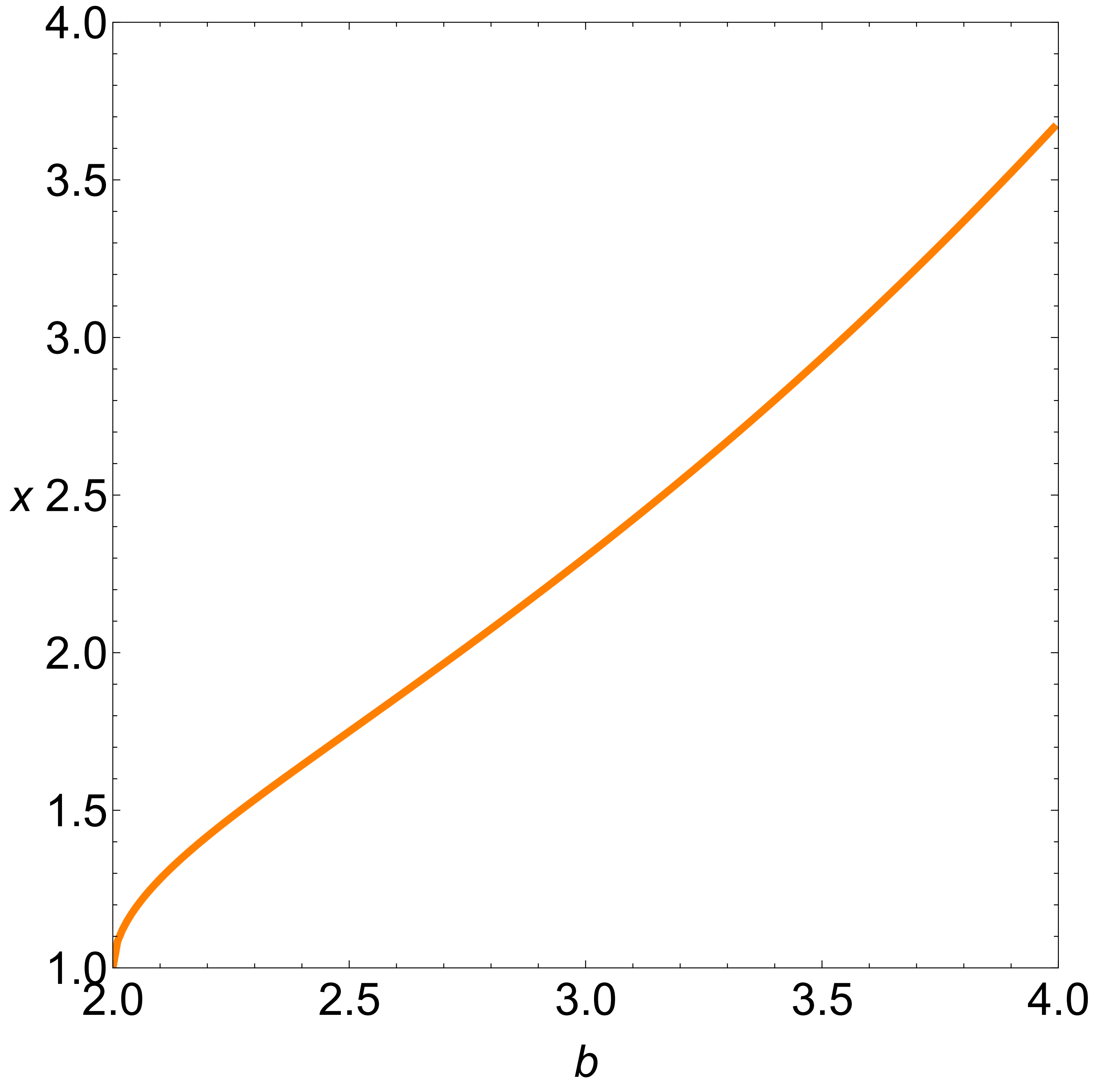
```
fp = Reap[Module[{bv, xv},
      Sow[{2, 1}];
      bv = 2.01;
      xv = 1.1;
      While[bv < 4.0,
       xv = x0 /. FindRoot[pmap[bv][x0] == x0, {x0, xv}, AccuracyGoal → 6, PrecisionGoal → 6];
       Sow[{bv, xv}];
       bv += 0.01;]]][[2, 1]];
```

This returns a list of (b, x_0) pairs whose first 10 elements are

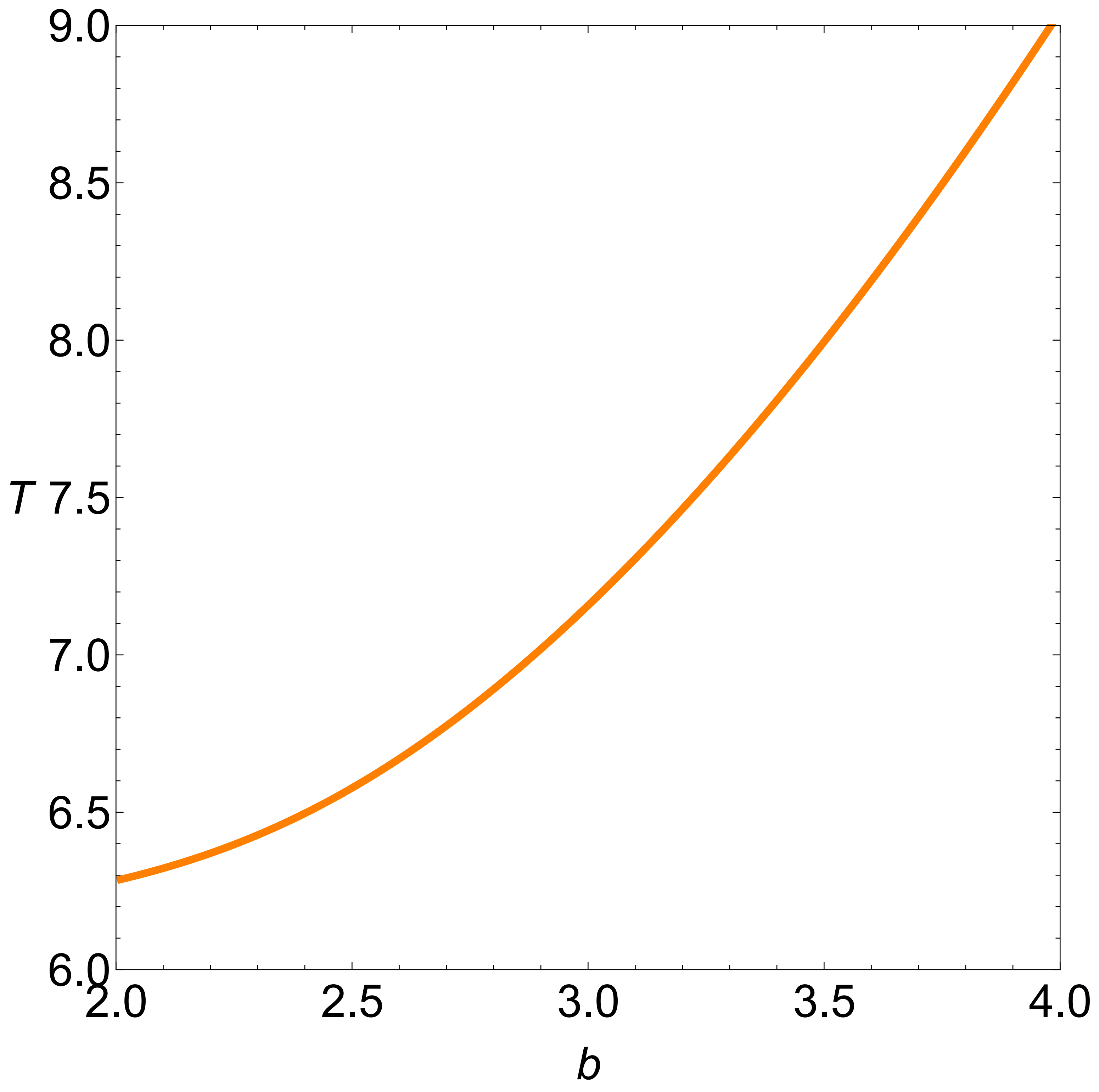
```
fp[[ ;; 10]]

{{2, 1}, {2.01, 1.08388}, {2.02, 1.11995}, {2.03, 1.14819}, {2.04, 1.17239},
 {2.05, 1.19402}, {2.06, 1.21385}, {2.07, 1.23232}, {2.08, 1.24972}, {2.09, 1.26626}}
```

Plotting the whole list we get the following picture. Note that for values of b close to $b = 2$, x_0 increases as $\sqrt{b - b_0}$. However, this is no longer true for larger values of b.



Now that we have the initial conditions of the limit cycles for different values of b we can track other quantities such as the period of the limit cycle. This is shown in the plot below. We notice that for b close to 2 the period is close to 2π. The period for a given value of b can be computed using the code below, assuming that x_0 is the point where the limit cycle intersects the half line $y = b$, $x > 1$.




```
period[b_?NumberQ][x0_?NumberQ] := Module[{stopTime, x, y, T},
  NDSolveValue[
   x'[t] == 1 - (b + 1) x[t] + x[t]^2 × y[t]
    && y'[t] == b x[t] - x[t]^2 × y[t]
    && T'[t] == 1
    && x[0] == x0
    && y[0] == b
    && T[0] == 0
    && WhenEvent[y[t] == b && x[t] > 1, stopTime = t; "StopIntegration"],
   T[stopTime], {t, 0, Infinity}]]
```