

7

Bifurcations and Hysteresis

Recall that any ordinary differential equation can be written as a first order system of ODEs,

$$x' = F(x), \quad x' := \frac{d}{dt}x(t). \quad (7.1)$$

Many interesting applications and physical phenomena can be modeled using ODEs. Given a mathematical model of the form (7.1), it is important to understand geometrically how its solutions behave. This information can then be conveyed in a phase portrait, a graph describing solutions of (7.1) with differential initial conditions. The first step in constructing a phase portrait is to find the equilibrium solutions of the equation, i.e., the zeros of $F(x)$, and to determine their stability.

It is often the case that the mathematical model we study depends on some parameter or set of parameters λ . Thus the ODE becomes

$$x' = F(x, \lambda). \quad (7.2)$$

The parameter λ can then be tuned to better fit the physical application. As λ varies, the equilibrium solutions and other geometric features of (7.2) may suddenly change. A value of λ where the phase portrait changes is called a *bifurcation point*; the study of how these changes occur is called *bifurcation theory*. The parameter values and corresponding equilibrium solutions are often graphed together in a bifurcation diagram.

As an example, consider the scalar differential equation

$$x' = x^2 + \lambda. \quad (7.3)$$

For $\lambda > 0$ equation (7.3) has no equilibrium solutions. At $\lambda = 0$ the equilibrium point $x = 0$ appears, and for $\lambda < 0$ it splits into two equilibrium points. For this system, a bifurcation occurs at $\lambda = 0$. This is an example of a saddle-node bifurcation. The bifurcation diagram is shown in Figure 7.1

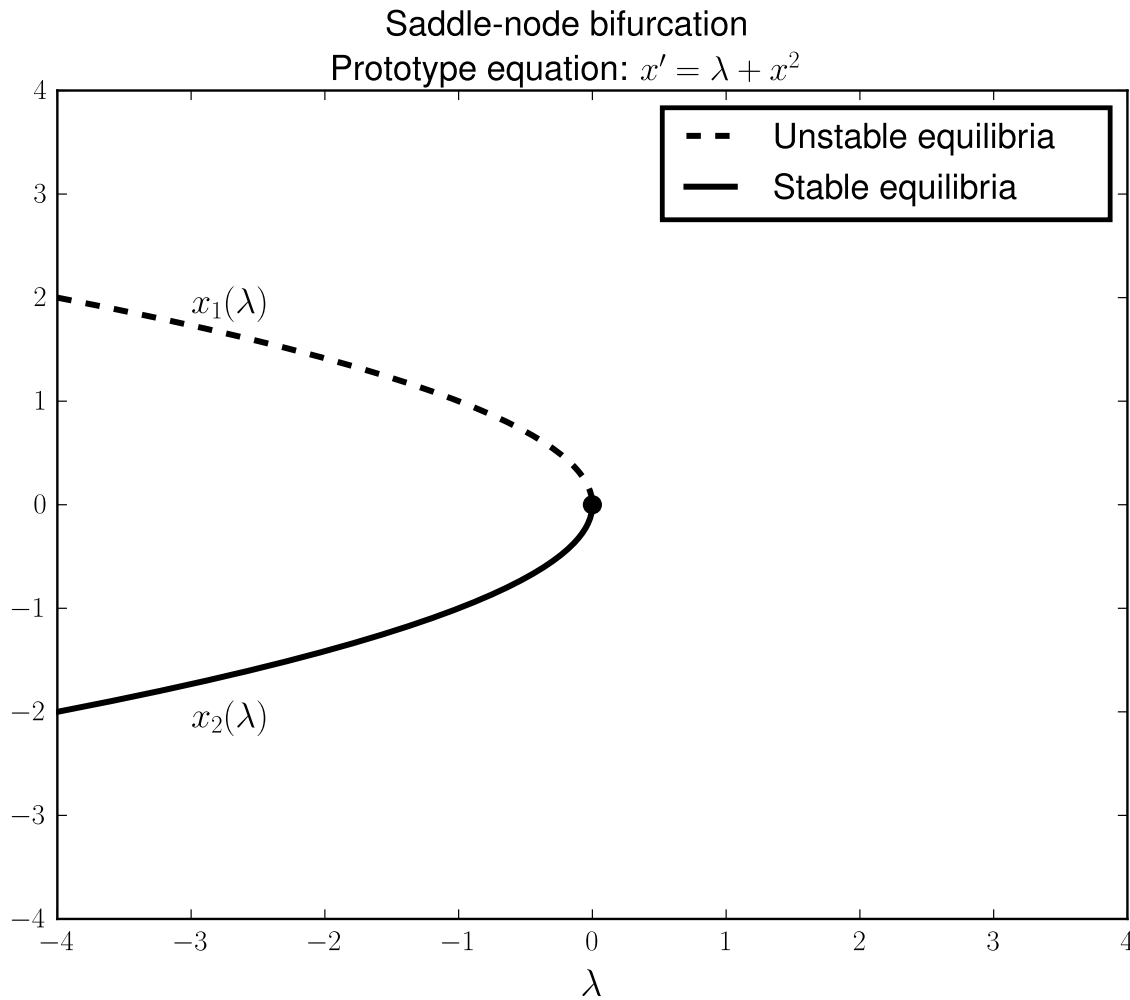


Figure 7.1: Bifurcation diagram for the equation $x' = \lambda + x^2$.

Suppose that $F(x_0, \lambda_0) = 0$. We use a method called natural embedding to find zeros (x, λ) of F for nearby values of λ . Specifically, we step forward in λ by letting $\lambda_1 = \lambda_0 + \Delta\lambda$, and use Newton's method to find the value x_1 that satisfies $F(x_1, \lambda_1) = 0$. This method works well except when λ is near a bifurcation point λ^* .

The following code implements the natural embedding algorithm, and then uses that algorithm to find the curves in the bifurcation diagram for (7.3). Notice that this algorithm needs a good initial guess for x_0 to get started.

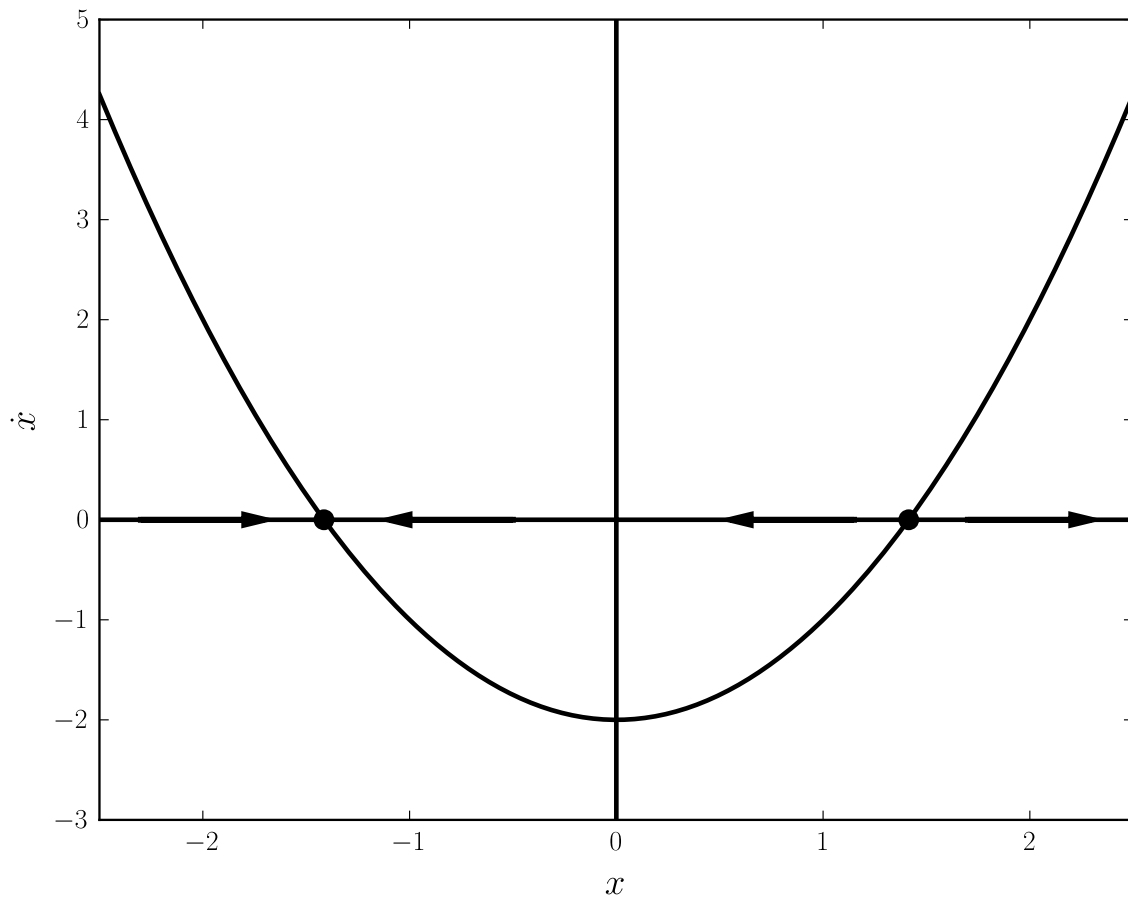


Figure 7.2: Phase Portrait for the equation $x' = -2 + x^2$.

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import newton

def EmbeddingAlg(param_list, guess, F):
    X = []
    for param in param_list:
        try:
            # Solve for x_value making F(x_value, param) = 0.
            x_value = newton(F, guess, fprime=None, args=(param,), tol=1E-7, ←
                             maxiter=50)
            # Record the solution and update guess for the next iteration.
            X.append(x_value)
            guess = x_value
        except RuntimeError:
            # If Newton's method fails, return a truncated list of parameters
            # with the corresponding x values.

```

```

        return param_list[:len(X)], X
    # Return the list of parameters and the corresponding x values.
    return param_list, X

def F(x, lambda):
    return x**2 + lambda

# Top curve shown in the bifurcation diagram
C1, X1 = EmbeddingAlg(np.linspace(-5, 0, 200), np.sqrt(5), F)
# The bottom curve
C2, X2 = EmbeddingAlg(np.linspace(-5, 0, 200), -np.sqrt(5), F)

```

Problem 1. Use the natural embedding algorithm to create a bifurcation diagram for the differential equation

$$x' = \lambda x - x^3.$$

This type of bifurcation is called a pitchfork bifurcation (you should see a pitchfork in your diagram).

Hints: Essentially this amounts to running the same code as the example, but with different parameters and function calls so that you are tracing through the right curves for this problem. To make this first problem work, you will want to have your 'linspace' run from high to low instead of from low to high. There will be three different lines in this image all of which must be produced using the EmbeddingAlg function. Any hard coding will result in an automatic 0. See Figure 7.3.

Problem 2. Create bifurcation diagrams for the differential equation

$$x' = \eta + \lambda x - x^3,$$

where $\eta = -1, -.2, .2$ and 1. Notice that when $\eta = 0$ you can see the pitchfork bifurcation of the previous problem. There should be four different images, one for each value of η . Each image will be built of 3 pieces. See Figure 7.4.

Hysteresis

The following ODE exhibits an interesting bifurcation phenomenon called hysteresis:

$$x' = \lambda + x - x^3.$$

This system has a bifurcation diagram containing what is known as a hysteresis loop, shown in Figure 7.5. In the hysteresis loop, when the parameter λ moves beyond the bifurcation point the equilibrium solution makes a sudden jump to the other stable branch. When this occurs the system cannot reach its previous equilibrium by simply rewinding the parameter slightly. The next section discusses a model with a hysteresis loop.

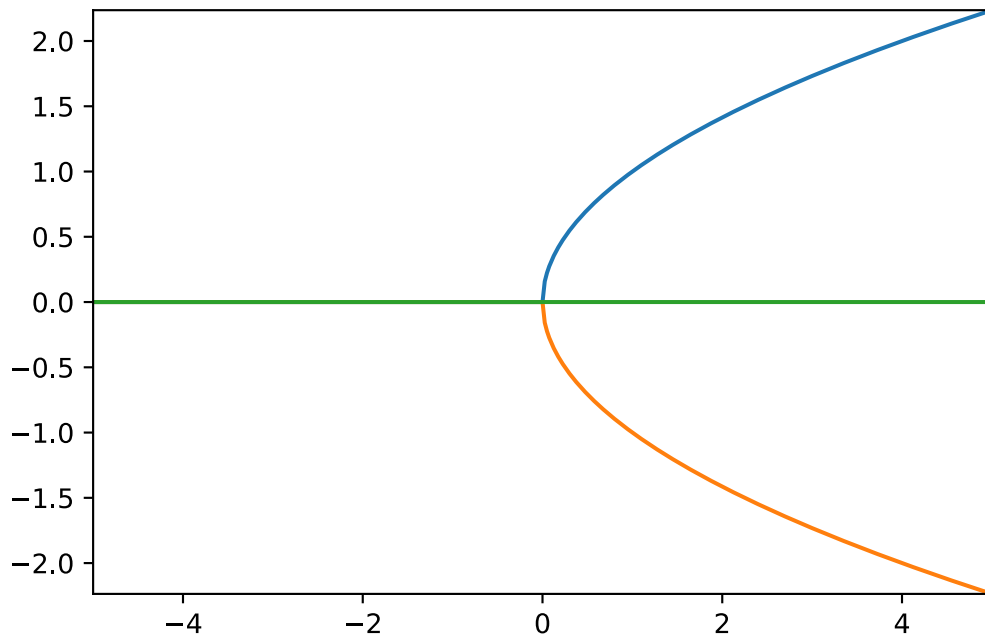


Figure 7.3: Bifurcation diagram for the equation $x' = \lambda x - x^3$.

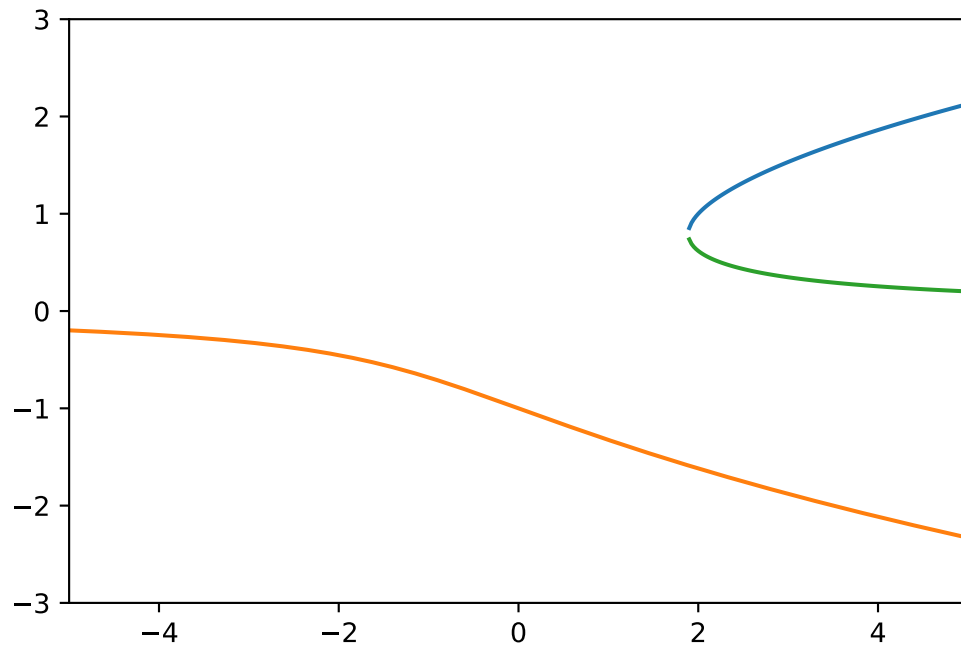


Figure 7.4: Bifurcation diagram for the equation $x' = \eta + \lambda x - x^3$ with $\eta = -1$.

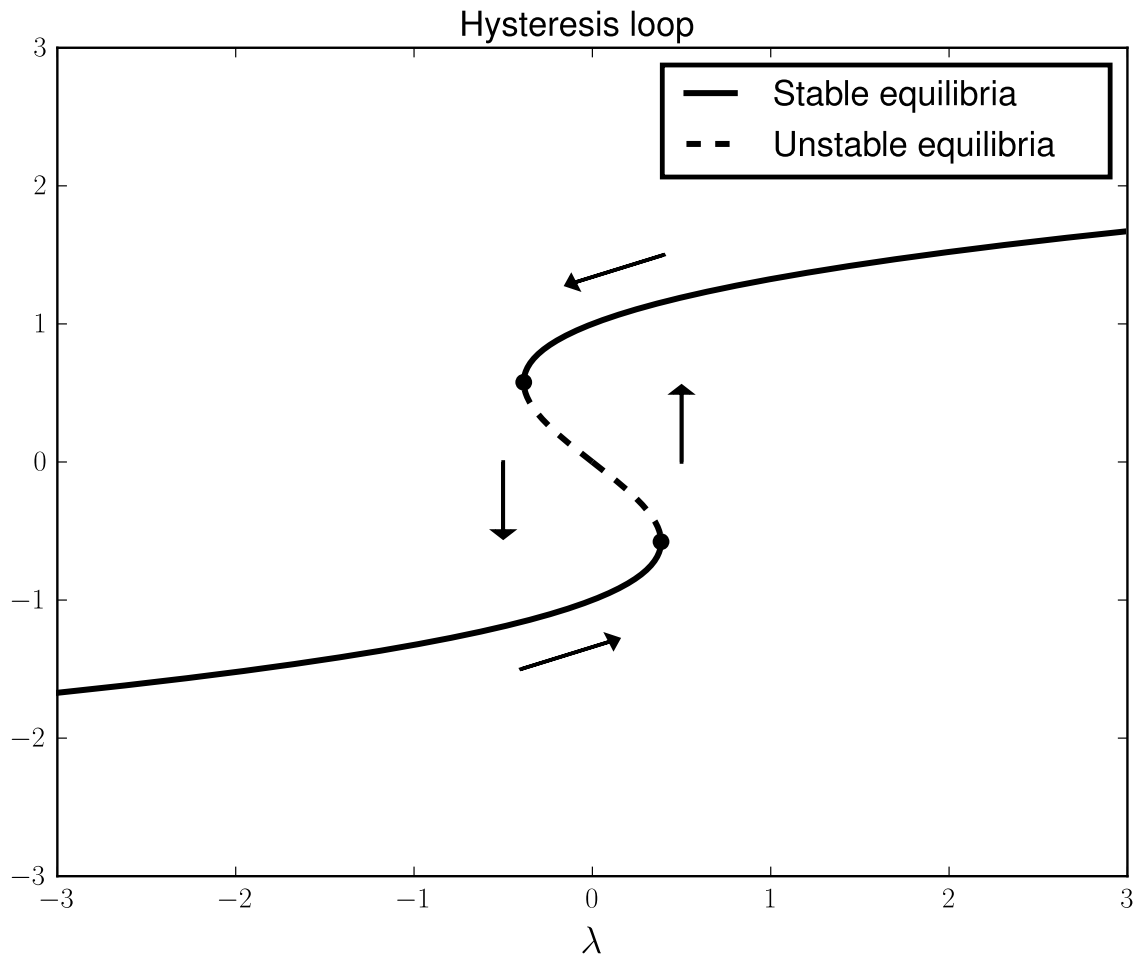


Figure 7.5: Bifurcation diagram for the ODE $x' = \lambda + x - x^3$.

Budworm Population Dynamics

Here we study a mathematical model describing the population dynamics of an insect called the spruce budworm. In eastern Canada, an outbreak in the budworm population can destroy most of the trees in a forest of balsam fir trees in about 4 years. The mathematical model is given by

$$N' = RN \left(1 - \frac{N}{K} \right) - p(N). \quad (7.4)$$

This model was studied by Ludwig et al (1978), and is described well in Strogatz's text *Nonlinear Dynamics and Chaos*. Here $N(t)$ represents the budworm population at time t , R is the growth rate of the budworm population and K represents the carrying capacity of the environment. We could interpret K to represent the amount of food available to the budworms. $p(N)$ represents the death rate of budworms due to predators (birds); we assume specifically that $p(N)$ has the form $P(N) = \frac{BN^2}{A^2 + N^2}$.

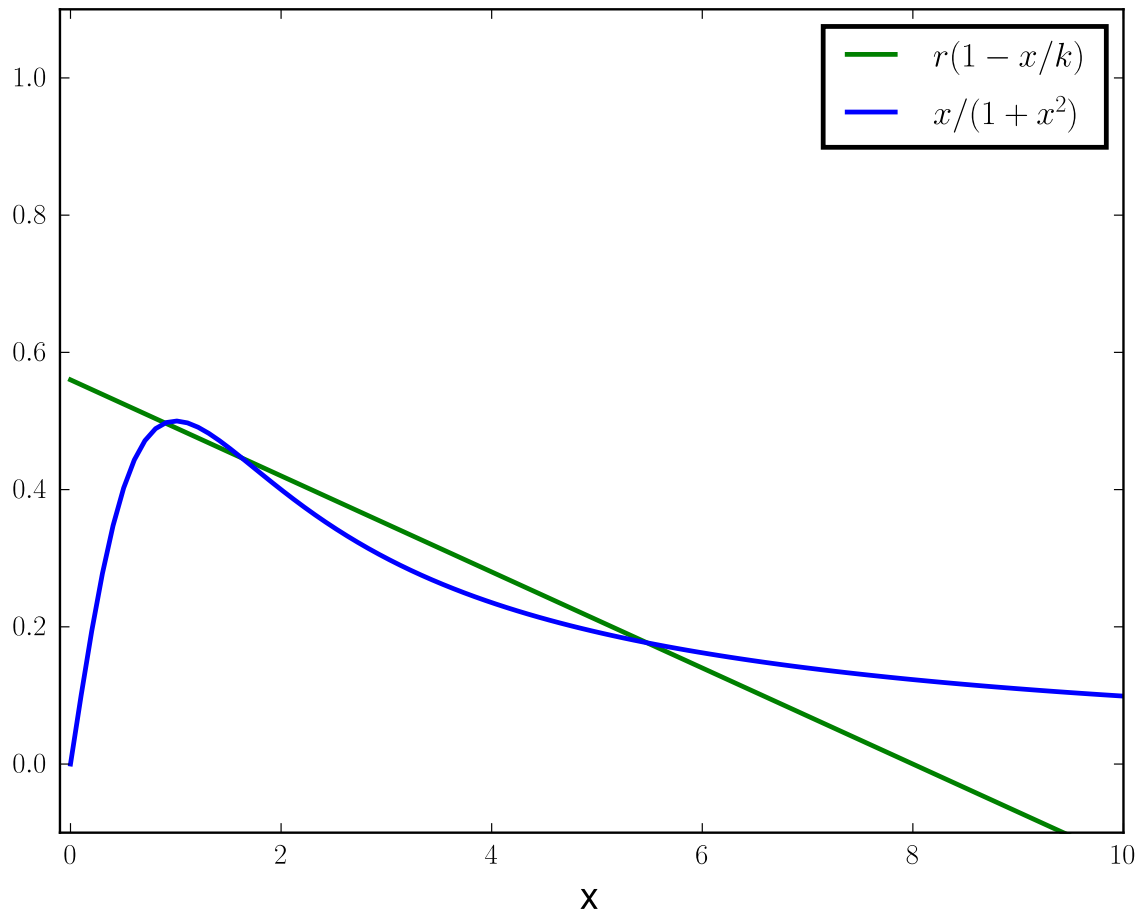


Figure 7.6: Graphical demonstration of nonzero equilibrium solutions for the budworm population (here $r = .56$, $k = 8$); equilibrium solutions occur where the curves cross. As k increases, the line $y = r(1 - x/k)$ gets more shallow and the number of solutions goes from one to three and then back to one.

Before studying the equilibrium points of (7.4) it is important to reduce the number of parameters in the system by nondimensionalizing. Thus, we make the coordinate change $x = N/A$, $\tau = Bt/A$, $r = RA/B$, and $k = K/A$, obtaining finally the system

$$\frac{dx}{d\tau} = rx(1 - x/k) - \frac{x^2}{1 + x^2}. \quad (7.5)$$

Note that $x = 0$ is always an equilibrium solution. To find other equilibrium solutions we study the equation $r(1 - x/k) - x/(1 + x^2) = 0$. Fix $r = .56$, and consider Figure (7.6) ($k = 8$ in the figure).

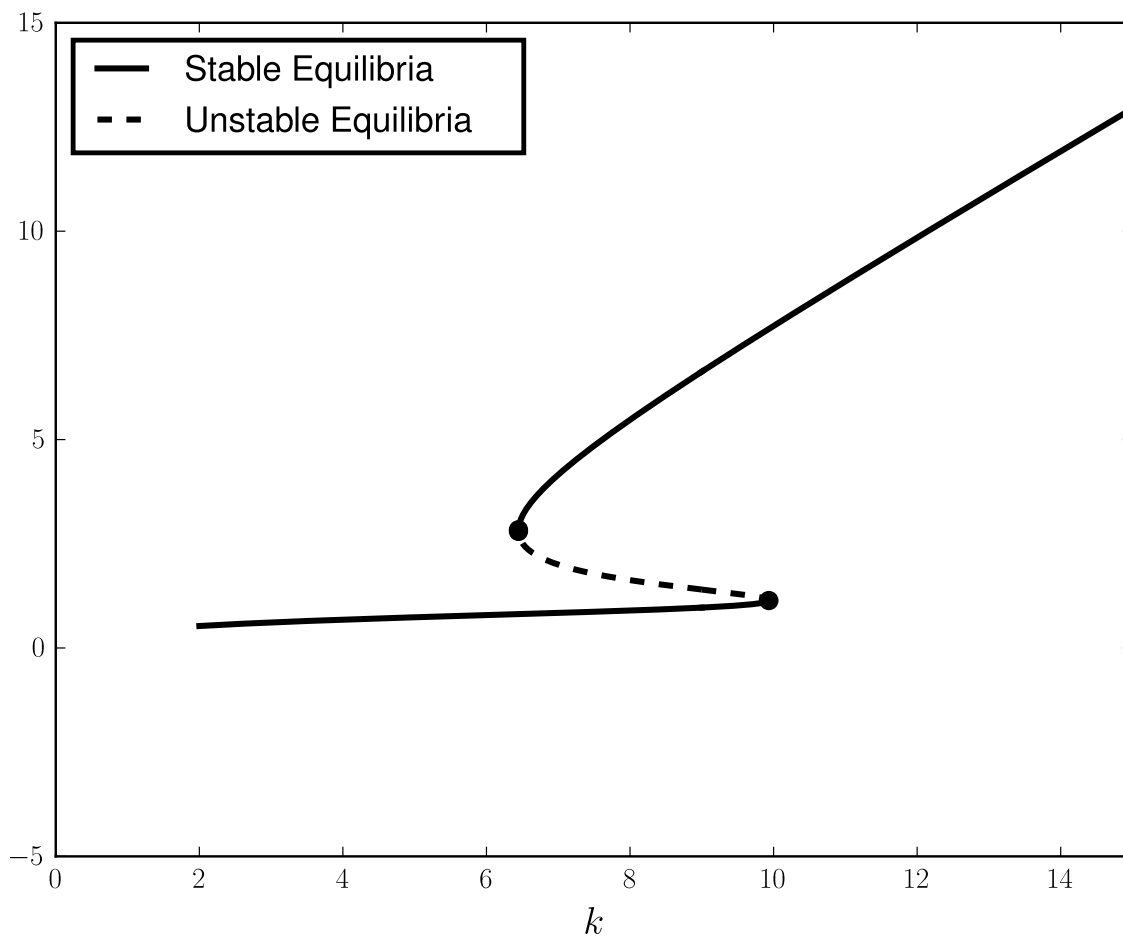


Figure 7.7: Bifurcation diagram for the budworm population model. The parameter r is fixed at 0.56. The lower stable branch is known as the refuge level of the budworm population, while the upper stable branch is known as the outbreak level. Once the budworm population reaches an outbreak level, the available food (foliage of the balsam fir trees) in the system must be reduced drastically to jump back down to refuge level. Thus many of the balsam fir trees die before the budworm population returns to refuge level.

Problem 3 (Budworm Population). Reproduce the bifurcation diagram (7.5) for the differential equation

$$\frac{dx}{d\tau} = rx(1 - x/k) - \frac{x^2}{1 + x^2},$$

where $r = 0.56$.

Hint: Find a value for k that you know is in the middle of the plot (i.e. where there are three possible solutions), then use the code from the previous problems to expand along each contour till you obtain the desired curve. Now find the proper initial guesses that give you the right bifurcation curve. The final plot will look like the one in Figure 7.7, but you will have to run the embedding algorithm 4-6 times to get every part of the plot. In order to make a black dashed line, add `'-k'` as the third argument in `plt.plot()` and use `'-k'` as the third argument in `plt.plot()` to get the solid black line.