Supply and Demand Dynamics

We start with as simple model of supply and demand. The demand curve is a standard downward sloping linear function of price, while the supply curve is linear and upward sloping. This means that quantity supplied and demanded can be represented by a system of two equations:

$$q_d(t) = a_0 - a_1 p(t) \quad (1)$$

$$q_s(t) = b_0 + b_1 p(t) \quad (2)$$

With a_0, a_1, b_0, b_1 being positive constants and $a_0 > b_0$. Note that price here is assumed to be time dependent, implying that quantity supplied and demand are also functions of time.

To model the time dynamics of supply and demand, we are going to introduce a variable for inventory $S(t)$. Changes in inventory over time will then be the difference between quantity supplied and quantity demanded. Mathematically that can be represented as the differential equation:

$$\frac{dS}{dt} = q_s - q_d \quad (3)$$

If we insert (1) and (2) into (3) we get the differential equation:

$$\frac{dS}{dt} = (a_1 + b_1)p + (b_0 - a_0) \quad (4)$$

The only thing missing now is a description of how the price evolves over time. It is reasonable to assume that changes in prices is negatively related to the size of the inventory. As inventories increases, businesses will want to reduce price to reduce inventories. Let us then define the following differential equation for the price:

$$\frac{dp}{dt} = -rS \quad (5)$$

Where r is a positive constant.

We now have a supply and demand model that consists of a system of two linear inhomogenous differential equations. In matrix form this system will look like:

$$\begin{bmatrix} \dot{S} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & a_1 + b_1 \\ -r & 0 \end{bmatrix} \begin{bmatrix} S \\ p \end{bmatrix} + \begin{bmatrix} b_0 - a_0 \\ 0 \end{bmatrix} \quad (6)$$

Or more compactly:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} \quad (7)$$

Let us also assume the following initial conditions $S(0) = S_0, p(0) = p_0$

Analytical solution

To solve this system we start by finding a solution to the homogenous system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$. This is done by solving the characteristic equation $|\mathbf{A}\mathbf{v} - \lambda\mathbf{I}| = 0$. This yields the following two eigenvalues:

$$\lambda_{1,2} = \pm\sqrt{-r(a_1 + b_1)} \quad (8)$$

We earlier assumed that a_1, b_1, r were all positive constants. This means that both eigenvalues of $\mathbf{A}$ are complex numbers with real parts equalling zero. With a bit of algebra we find that the eigenvector corresponding to the eigenvalue $-\sqrt{-r(a_1+b_1)}$ is $\mathbf{v_1} = \begin{bmatrix} \frac{\sqrt{-r(a_1+b_1)}}{a_1+b_1} \\ 1 \end{bmatrix}$, and for the eigenvalue $\sqrt{-r(a_1+b_1)}$ we get the eigenvector $\mathbf{v_2} = \begin{bmatrix} -\frac{\sqrt{-r(a_1+b_1)}}{a_1+b_1} \\ 1 \end{bmatrix}$.

The solution to the homogenous system is then:

$$\mathbf{x_h} = C_1\,\mathbf{v_1}\begin{bmatrix} \cos\left(-\sqrt{r(a_1+b_1)t}\right) \\ -\sin\left(-\sqrt{r(a_1+b_1)t}\right) \end{bmatrix} + C_2\,\mathbf{v_2}\begin{bmatrix} sin\left(-\sqrt{r(a_1+b_1)t}\right) \\ cos\left(-\sqrt{r(a_1+b_1)t}\right) \end{bmatrix} \qquad (9)$$

The next step is to find a particular solution to the inhomogenous system. Note that the inhomogenous part in this case is the vector $\begin{bmatrix} b_0 - a_0 \\ 0 \end{bmatrix}$. Let us then calculate the steady state solution of the system, implying that $\dot{\mathbf{x}} = \mathbf{0}$, or:

$$\begin{bmatrix} 0 & a_1+b_1 \\ -r & 0 \end{bmatrix}\begin{bmatrix} S \\ p \end{bmatrix} + \begin{bmatrix} b_0 - a_0 \\ 0 \end{bmatrix} = 0 \qquad (10)$$

Equation 10 yields the steady state solution $(S, p) = \left(0, \frac{a_0-b_0}{a_1+b_1}\right)$.

Combining the solution to the homogenous system with our particular solution, we get the general solution to the system:

$$\mathbf{x} = C_1\,\mathbf{v_1}\begin{bmatrix} \cos\left(-t\sqrt{r(a_1+b_1)}\right) \\ -\sin\left(-t\sqrt{r(a_1+b_1)}\right) \end{bmatrix} + C_2\,\mathbf{v_2}\begin{bmatrix} sin\left(-t\sqrt{r(a_1+b_1)}\right) \\ cos\left(-t\sqrt{r(a_1+b_1)}\right) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{a_0-b_0}{a_1+b_1} \end{bmatrix} \qquad (11)$$

The final step is to solve for the arbitrary constants C_1, C_2. Inserting $t = 0$ into 11 and setting it equal to the initial values for S, p gives us:

$$C_1\,\mathbf{v_1}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + C_2\,\mathbf{v_2}\begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{a_0-b_0}{a_1+b_1} \end{bmatrix} = \begin{bmatrix} S_0 \\ p_0 \end{bmatrix} \qquad (12)$$

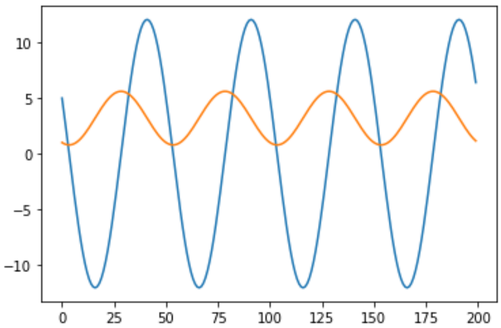
Numerical solution

Because the system of ODEs we have looked at is a linear system with constant coefficients, it was possible for us to find an analytical solution to the problem. It is usually not possible to find general solutions to such systems if they are nonlinear. In that case numerical techniques are need.

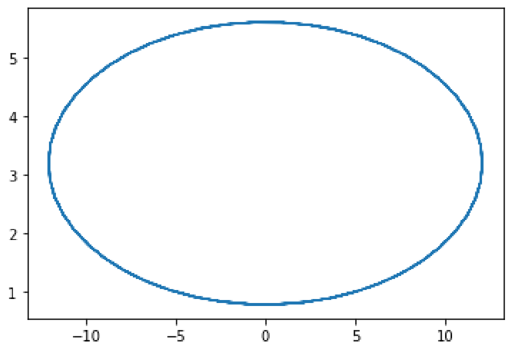
Even for systems where analytical solutions are known, it can be useful to simulate the system using numerical methods. There are a number of ways to go about this. For this example, I will use a simple Python program using the scipy package to find a numerical solution to the system. The program uses initial conditions $\left(S(0), p(0)\right) = (5,1)$ $(a_0, a_1, b_0, b_1, r) = (10,1,2,1.5,0.1)$The code is presented below.

```
import numpy as np
from matplotlib import pyplot as plt
from scipy.integrate import odeint

def dx_dt(x,t):
    S,p = x
    a0 = 10
    a1 = 1
    b0 = 2
    b1 = 1.5
    r = 0.1
    q_d = a0 - a1 * p
    q_s = b0 + b1 * p
    dS_dt = q_s - q_d
    dP_dt = -r * S

    return dS_dt, dP_dt
t = np.linspace(0,50,200)
x0 = (5,1)

solution = odeint(dx_dt, x0, t)
plt.plot(solution[:,0],solution[:,1])
```

The plot of the solution over time looks like:



While the plot of the S values against the p values is an ellipse:



A property of this model is that, unless we start at the steady state point, the system will never reach a steady state. Instead the values of S and p fluctuates permanently around the steady state values. This is a consequence of the system only having imaginary eigenvalues for its homogenous solution.

Note that there are no restrictions on the value of S. It can be both positive and negative. A positive value of S has a clear interpretation as a physical build up of inventory. Although it is impossible for inventories to be physically negative, a negative value for S can still be interpreted as a backlog of orders.

In this model, suppliers only adjust the quantity supplied when prices change. One possible adjustment to the model is to assume that the suppliers respond directly to a build up of inventory by cutting production. Similarly, it would be realistic to assume that a large backlog of orders leads to an increase in supply.

To model this we can adjust equation 2 to:

$$q_s(t) = b_0 + b_1 p(t) - b_2 S(t) \quad (13)$$

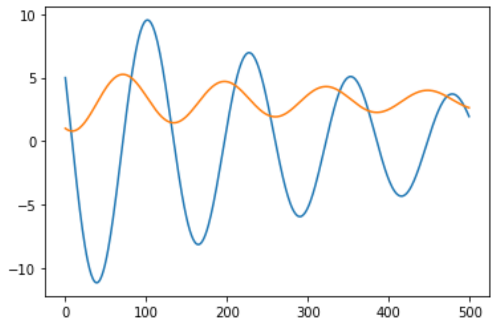
Which gives the following differential equation for S:

$$\frac{dS}{dt} = -b_2 S + (a_1 + b_1)p + (b_0 - a_0) \qquad (14)$$

The easiest way to illustrate the effect of this modification is to add it to our Python simulation, and then run the model with some value for b_2.

```
import numpy as np
from matplotlib import pyplot as plt
from scipy.integrate import odeint

def dx_dt(x,t):
    S,p = x
    a0 = 10
    a1 = 1
    b0 = 2
    b1 = 1.5
    b2 = 0.5
    r = 0.1
    q_d = a0 - a1 * p
    q_s = b0 + b1 * p – b2 * S
    dS_dt = q_s - q_d
    dp_dt = -r * S

    return dS_dt, dp_dt
t = np.linspace(0,50,200)
x0 = (5,1)

solution = odeint(dx_dt, x0, t)
plt.plot(solution[:,0],solution[:,1])
```

If we run the simulation now we get a somewhat different behaviour of our variables:



The values for both S and p are still cyclical, but now they converge to their steady state values. If producers adjust quantity supplied directly as a function of inventory, the cycles are dampened.