MP307 Practical 4 Continuous Population Models

Dara Corr ID: 18483836

In [4]:
```
%matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

Q1. Verhulst Logistic Model.

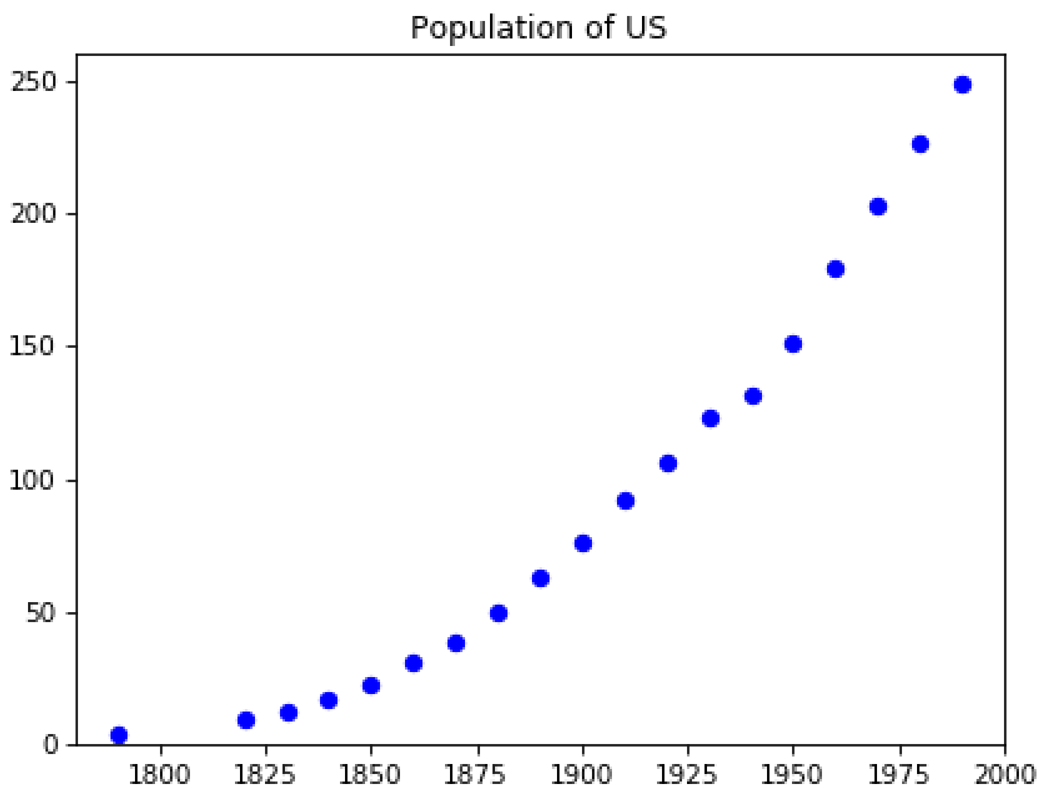
Consider the logistic population model

$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right)$$

for $r, K > 0$ as a model of the US population from 1790 to 1990. The data for this are as given in the worksheet Q2. of `lab3_discrete_population_models.ipynb` as follows:

In [3]:
```
Year = [1790, 1820, 1830, 1840, 1850, 1860, 1870, 1880, 1890, 1900, 1910, 1920
, 1930,
        1940, 1950, 1960, 1970, 1980, 1990]
Pop= [3.79, 9.6, 12.9, 17.1, 23.2, 31.4, 38.6, 50.2, 62.9, 76.0, 92.0, 106.5,
123.2,
      132.0, 151.3, 179.3, 203.3, 226.54, 248.7]
```

In [5]:
```
# plot of data
plt.figure(1)
plt.plot(Year,Pop,'bo')
plt.axis([1780,2000,0,260])
plt.title('Population of US')
plt.show()
```



The Logistic differential equation has solution

$$P(t) = \frac{K}{1 + \left(\frac{K}{P_0} - 1\right) e^{-rt}}$$

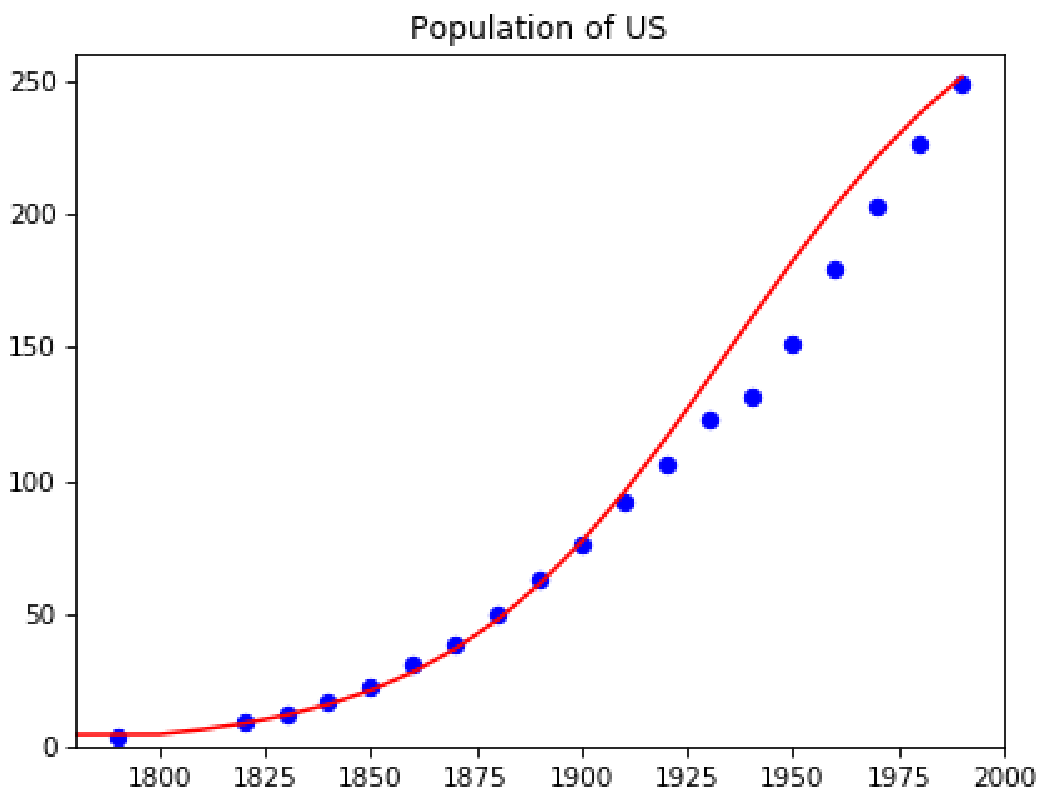
In [101]:
```
#Logistic Model

# Python function for Logistic solution where t=0 is 1790

def P(t):
    P_0=3.79
    r=0.3 #  in inverse decades
    K=300
    ans=K/(1.+(K/P_0-1.)*np.exp(-r*t))
    return ans
```

In [102]:
```
P(0)
```

Out[102]: 3.79

In [103]:
```
# plot Verhulst pop PV from 1790 to 1990 with 10 year intervals
PV=[0]*21 # list for Verhulst P initialized with 20 zeros
YV=[0]*21 # list for years initialized with 20 zeros

# choose 10 year intervals
for n in range(1,21):
    YV[n]=1790+10*n
    PV[n]=P(n)
```

In [104]:
```
plt.figure(2)
plt.plot(Year,Pop,'bo')
plt.plot(YV,PV,'r-')
plt.axis([1780,2000,0,260])
plt.title('Population of US')
plt.show()
```



In [ ]:

Q.2 Competitive Species Model

Consider two species with population sizes P_1, P_2 with growth rates r_1, r_2 and limiting population sizes of K_1, K_2 where

$$\frac{dP_1}{dt} = r_1 P_1 \left(1 - \frac{P_1 + P_2}{K_1}\right),$$
$$\frac{dP_2}{dt} = r_2 P_2 \left(1 - \frac{P_1 + P_2}{K_2}\right).$$

Analyse the behaviour of P_1, P_2 in the following cases, by plotting P_1, P_2 vs t and P_1 vs P_2:

(a). $r_1 = 1/10, r_2 = 1/10$, $K_1 = 100$, $K_2 = 50$ with $P_1(0) = 10$ and $P_2(0) = 15$.

(b). $r_1 = 1/10, r_2 = 1/10$, $K_1 = 100$, $K_2 = 50$ with $P_1(0) = 130$ and $P_2(0) = 200$.

(c). $r_1 = 1/10, r_2 = 1/100$, $K_1 = 40$, $K_2 = 50$ with $P_1(0) = 130$ and $P_2(0) = 20$.

(d). $r_1 = 1/10, r_2 = 1/100$, $K_1 = 50$, $K_2 = 60$ with $P_1(0) = 15$ and $P_2(0) = 10$.

In [5]:
```
# parameter values
r_1, r_2 = 1/10, 1/100

K_1, K_2 = 50, 60

P1_0, P2_0 = 15, 10 # initial populations
```

Define a function `dP_dt(P, t)` of a Python list `P=[P1,P2]` describing dP1/dt and dP2/dt.

Note: recall that indexing in Python goes 0,1,... so that Pop1 is stored in `P[0]` and Pop2 is stored in `P[1]`

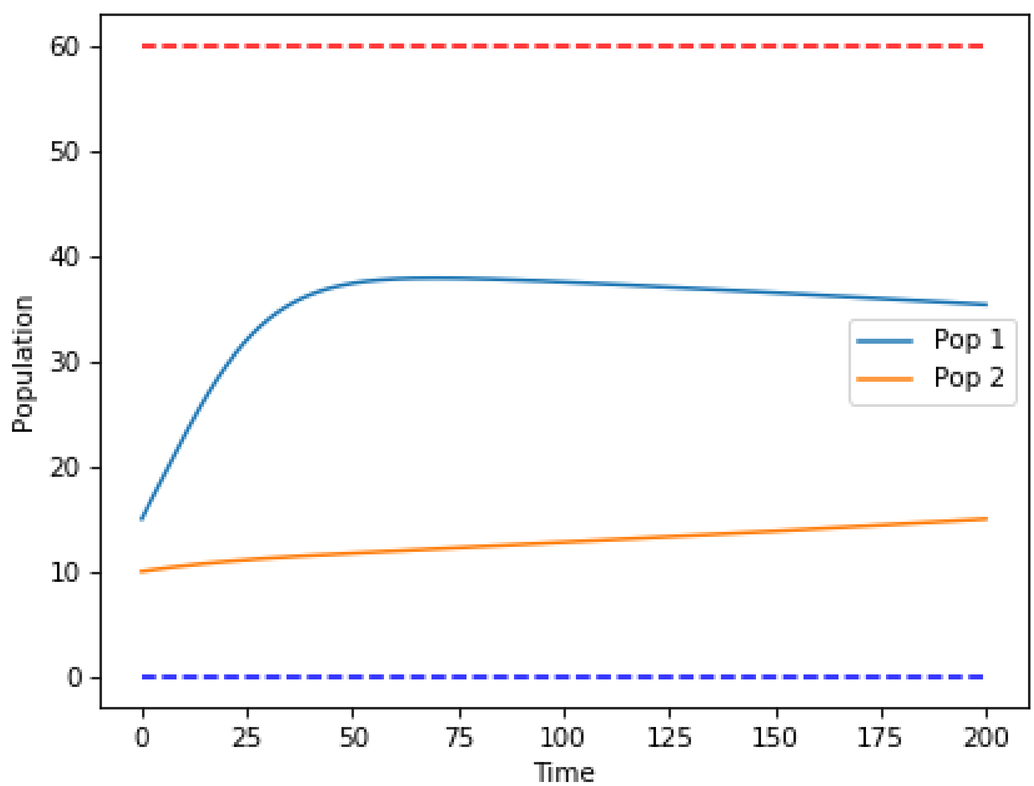
In [6]:
```
def dP_dt(P, t): # P=[P1,P2] list
    return [r_1*P[0]*(1 - (P[0]+P[1])/K_1), r_2*P[1]*(1 - (P[0]+P[1])/K_2)]
```

Use `odeint` Python function to integrate the pair of odes subject to the initial conditions for time from `tmin` to `tmax`.

In [7]:
```
tmin=0
tmax=200

#increased time to 200 steps for (d)

ts = np.linspace(tmin, tmax, 200) # 100 equally spaced t values from tmin to t
max
P0 = [P1_0, P2_0] #initial conditions
Ps = odeint(dP_dt, P0, ts)
P1 = Ps[:,0] # values of P1 on ts values
P2 = Ps[:,1] # values of P2 on ts values
```

Plot P_1 and P_2 vs t and asymptotic lines.

In [8]:
```
plt.figure(3)
plt.plot(ts, P1, "-", label="Pop 1")
plt.plot(ts, P2, "-", label="Pop 2")
plt.xlabel("Time")
plt.ylabel("Population")
plt.legend();
```



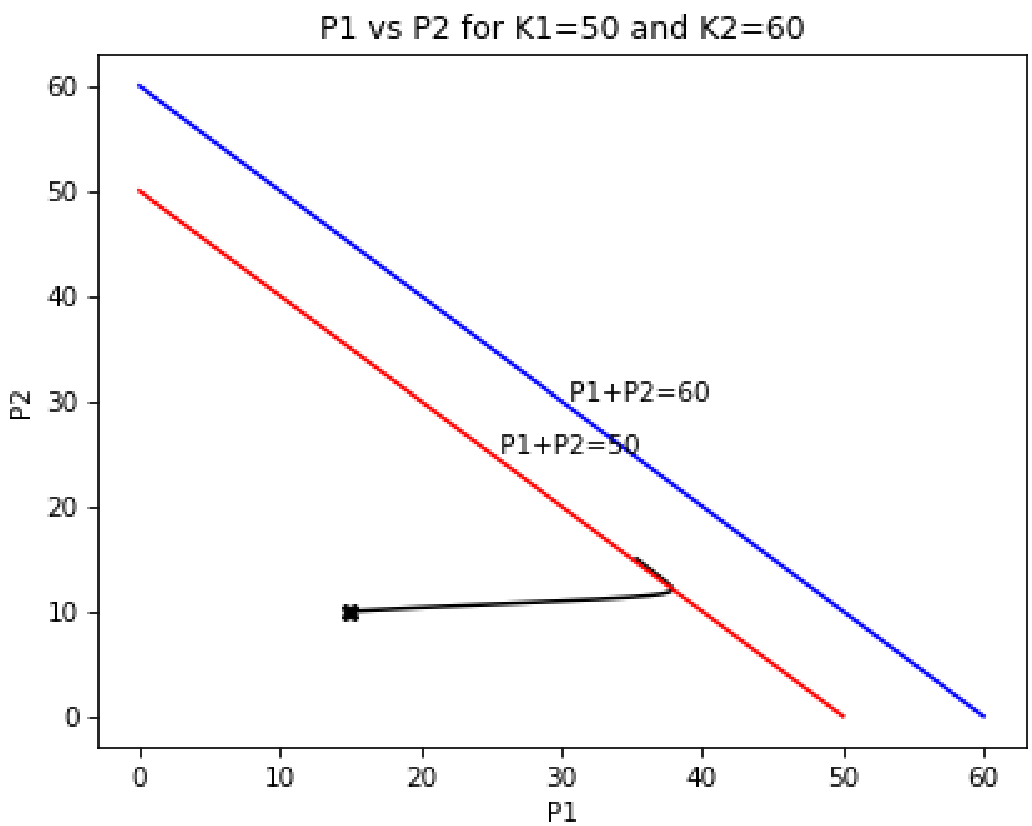
Notice that as $t \to \infty$ then $(P_1, P_2) \to (K_1, 0)$ for $K_1 > K_2$ and $(P_1, P_2) \to (0, K_2)$ for $K_1 < K_2$.

We can add the asymptotic lines to the above graph as follows

In [9]:
```
P1_infty, P2_infty = K_1, 0
if K_2>K_1:
    P1_infty, P2_infty = 0,K_2
plt.plot([tmin,tmax],[P1_infty,P1_infty],"b--")
plt.plot([tmin,tmax],[P2_infty,P2_infty],"r--");
```

Plot P_1 vs P_2 with initial point marked with an x. We also plot the lines $P_1 + P_2 = K_1$ and $P_1 + P_2 = K_2$.

In [10]:
```
plt.figure(4)

plt.plot(P1_0, P2_0,'kX') # mark initial point with x

plt.plot(P1, P2, 'k-')
plt.plot([0,K_1],[K_1,0],'r-')
plt.plot([0,K_2],[K_2,0],'b-')
plt.text(K_1/2, K_1/2,' P1+P2='+str(K_1))
plt.text(K_2/2, K_2/2,' P1+P2='+str(K_2))
plt.xlabel("P1")
plt.ylabel("P2")
plt.title("P1 vs P2 for K1="+str(K_1)+" and K2="+str(K_2));
```



(a) Population 1 grows in population size from a population of 10 and reaches final population of 100 after 100 time steps. Population 2 initially increases from initial population of 15 to 20 and then delines to population of 0. In the Plot of P2 vs P1, we see that (P1(t),P2(t)) start in region I and both P1 and P2 are increasing. Then in region II P2 starts to decline and P1 continues increasing until reaching (P1,P2) = (K1,0) = (100,0).

(b)Population 1 and 2 both see a rapid decrease initially until P1 = 76 and P2 = 24, then P1 starts to increase to equilibrium at 100 while P2 continues to decline until P2 = 0. In Figure 4, we observe that the plot begins in region III, P1 and P2 decrease until they enter region II when P1+P2 < 100, and then P2 continues to decrease to 0 and P1 increases until it reaches 100, resulting in the fixed point (P1,P2) = (K1,0) = (100,0).

(c)Here P1 decreases rapidly from its initial value of until it dies down at approximately 19. P2 starts with an initial population of 20, and slowly decreases initially until P1 + P2 = 50 = K2 is reached then P2 increases until it reaches a final value of about 21 and the growth stops at the fixed point P1+P2 = 40 = K1. In figure 4, the plot begins in region III but the plot reaches the K2 line before the K1 line which causes P2 to increase and P1 to decrease further until a fixed point on the K1 = 40 line is reached in region II.

(d)P1 increases from an initial value of 15 until it reaches a value of around 37 at the K1 line where P1 begins to decline at a slower rate until it reaches a value of around 35. P2 sees a small near-constant increase over time until it stops at a final value of about 15. Figure 4 shows the line (P1, P2) in region I moving towards region II on the P1 axis until the line K1 = P1 + P2 is reached. At the K1 line (P1,P2) move upwards on the line K1 with P1 decreasing and P2 decreasing until reaching its final fixed point on the line. In figure 3, P1 and P2 do not reach the asymptotes that predict the future values of (P_1, P_2) as $t \to \infty$. This is since the plot starts in Region I in figure 4 and K2 > K1.

Q.3 Lotka-Volterra Predator/Prey System

Consider a prey species with population size x and a predator species with population size y where

$$\frac{dx}{dt} = x(a_1 - b_1 y),$$
$$\frac{dy}{dt} = y(-a_2 + b_2 x),$$

with $a_1, a_2, b_1, b_2 > 0$. Analyse the behaviour of the system for $a_1 = 3, a_2 = 5/2$ and $b_1 = 2, b_2 = 1$ by plotting x, y vs t, and x vs y in the following cases:

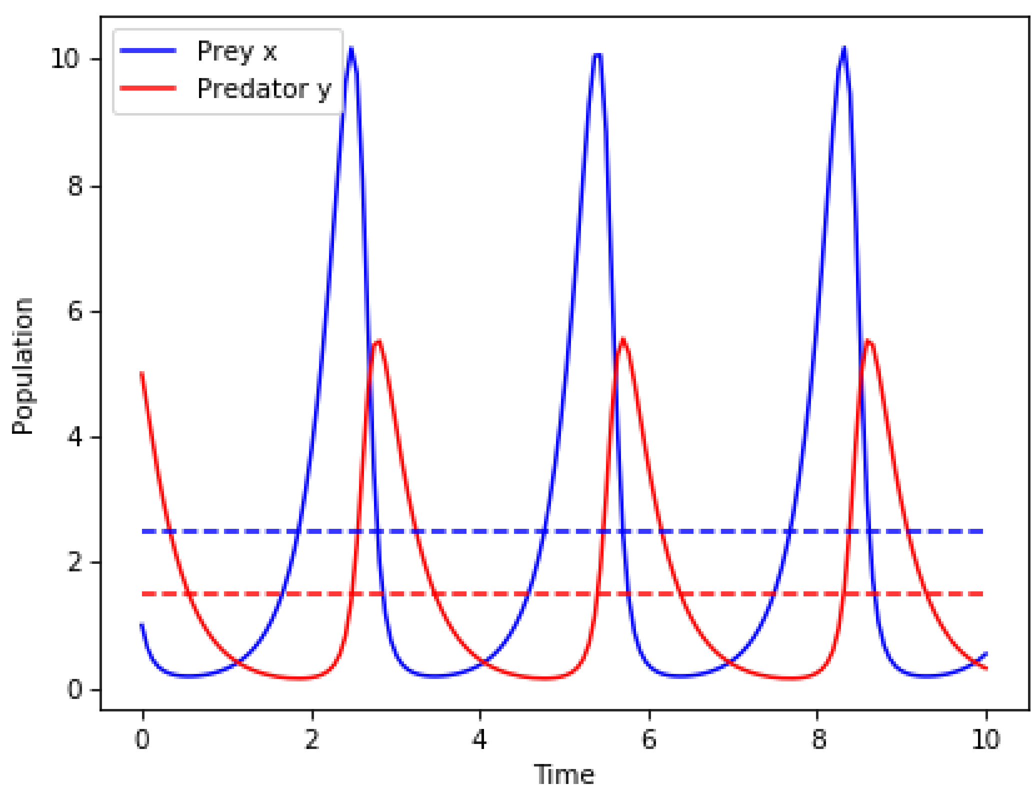
(a). $x(0) = 1$ and $y(0) = 1$.

(b). $x(0) = 0.1 + 5/2$ and $y(0) = 0.1 + 3/2$. What behaviour do you observe?

(c). $x(0) = 1$ and $y(0) = 5$.

Use similar coding to above

In [232]:
```
a_1, a_2, b_1, b_2 = 3, 5/2, 2, 1

x0, y0 =  1,5 # initial populations

timerange=10

def dP_dt(P, t): # P=[x,y] list
    return [P[0]*(a_1 - b_1*P[1]), P[1]*(-a_2 +b_2 *P[0])]

ts = np.linspace(0, timerange, 150)
P0 = [x0, y0] #initial conditions
Ps = odeint(dP_dt, P0, ts)
prey = Ps[:,0]
predators = Ps[:,1]

plt.figure(5)
plt.plot(ts, prey, "b-", label="Prey x")
plt.plot(ts, predators, "r-", label="Predator y")
plt.xlabel("Time")
plt.ylabel("Population")
plt.legend();
```



Equlibrium values $x_{eq} = a_2/b_2$ and $y_{eq} = a_1/b_1$. Notice that the maximum/minimum of x occurs when $y = y_{eq}$ and similarly, the maximum/minimum of y occurs when $x = x_{eq}$

In [233]:
```
xeq , yeq =a_2/b_2, a_1/b_1
print("(xeq,yeq)=",(xeq,yeq))
```

```
(xeq,yeq)= (2.5, 1.5)
```

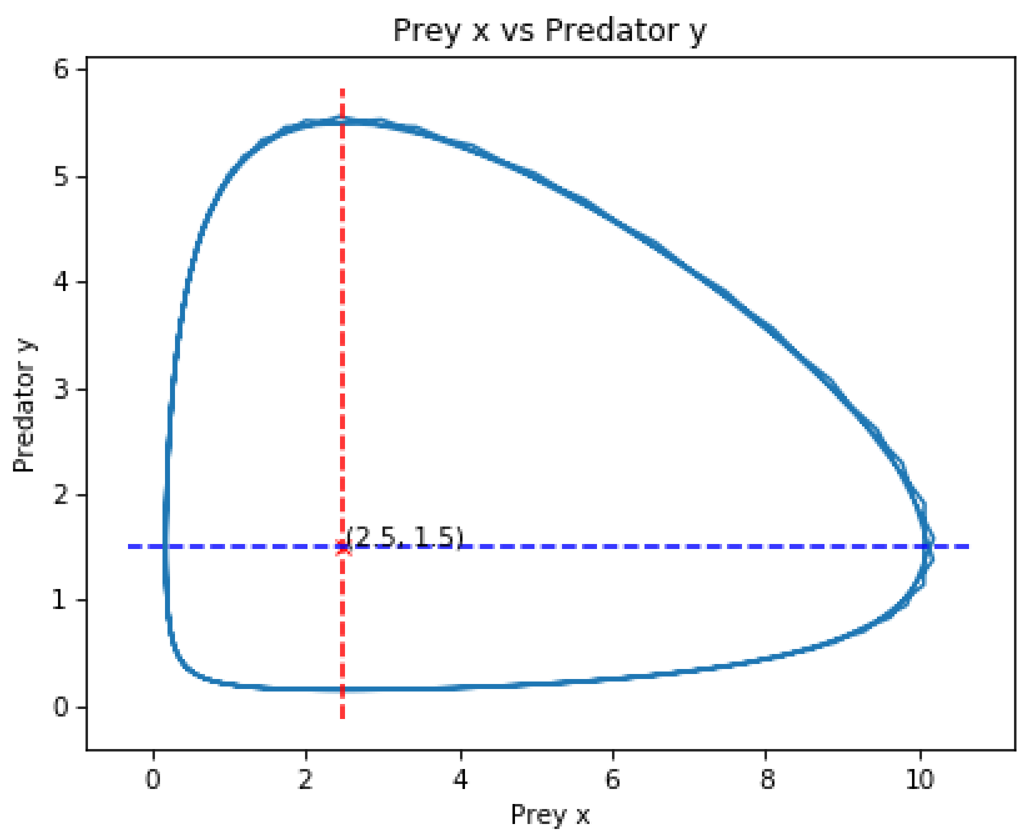
Look at values of x for $y = y_{eq}$ and of y for $x = x_{eq}$ in the last plot

In [234]:
```
plt.plot([0,timerange],[xeq,xeq],"b--")
plt.plot([0,timerange],[yeq,yeq],"r--");
```

In [ ]:

Plot x vs y and show the equilibrium point. This plot shows that the system is periodic.

In [235]:
```
plt.figure(6)
plt.plot(prey, predators, "-")
plt.plot([xeq],[yeq],'rx')
plt.text(xeq*(1.01),yeq*(1.01),str((xeq,yeq)))
plt.xlabel("Prey x")
plt.ylabel("Predator y")
plt.title("Prey x vs Predator y");
```



Add lines $x = x_{eq}$ and $y = y_{eq}$ to this plot

In [236]:
```
miny=min(predators)
maxy=max(predators)
dely=maxy-miny
plt.plot([xeq,xeq],[miny-dely*0.05,maxy+dely*0.05],"r--")

minx=min(prey)
maxx=max(prey)
delx=maxx-minx
plt.plot([minx-delx*0.05,maxx+delx*0.05],[yeq,yeq],"b--");
```

(a) There is a lag observed between the growth/decline of the prey population and the growth/decline of the predator population. Both populations show oscillatory behabiour over time. The dotted lines on the plot highlight the equilibrium values of the prey and predator populations. We notice when the prey population is below its equilibrium value, the predator population is decreasing and when the prey population is above the equilibrium value, the predator population is increasing. Similarly when the predator population is above its equilibrium value it causes the prey population to decrease and when it is below equilibrium value it causes the prey population to increase. The plot in figure 6 shows the cyclic behaviour between the Predator and Prey populations. The plots of the two populations overlap eachother and each population plotted over time looks like a sinusoidal plot shifted slightly towards one direction. The slight deformation from a perfect sine wave gives the unique egg shaped plot in figure 6 with centre (x_{eq}, y_{eq}). We notice also how the prey plot has much greater amplitude than the predator plot here.

(b)In this example we observe oscillatory behaviour with the Predator and Prey plots. The peaks of the predator populations line up perfectly with when the prey population is at its equilibrium value which was not the case for part (a). Both populations plotted against time resemble perfect sine waves shifted out of phase with eachother. Since both populations plotted against time result in sine waves out of phase with eachother, this results in a plot of a perfect ellipse showing the cyclic behaviour of the populations when Prey and Predator populations are plotted against eachother. This is because $x_0 = 0.1 + x_{eq}$ and $y_0 = 0.1 + y_{eq}$ here.

(c)This model begins with an initial predator population 5 times the size of the prey population. This leads to very rapid decline in the predator population as it runs out of food to eat. This is followed by a large increase in prey population which is followed again by very quick and large growth in the predator population until it exhausts the prey population and declines rapidly again. This is similar to the first case (a) but with larger and steeper peaks due to the large discepency in initial population sizes. The plot of prey x vs predator y populations results in a cyclic plot like the one in (a) except (x_{eq}, y_{eq}) is in the bottom left of the plot now instead of the centre because of the large starting population of the predator population.

In [ ]:
```

```