Example 4.2. The blue whale and fin whale are two similar species that inhabit the same areas. Hence, they are thought to compete. The intrinsic growth rate of each species is estimated at 5% per year for the blue whale and 8% per year for the fin whale. The environmental carrying capacity (the maximum number of whales that the environment can support) is estimated at 150,000 blues and 400,000 fins. The extent to which the whales compete is unknown. In the last 100 years intense harvesting has reduced the whale population to around 5,000 blues and 70,000 fins. Will the blue whale become extinct?

$B =$ number of blue whales
$F =$ number of fin whales
$g_B =$ growth rate of blue whale population (per year)
$g_F =$ growth rate of fin whale population (per year)
$c_B =$ effect of competition on blue whales (whales per year)
$c_F =$ effect of competition on fin whales (whales per year)

In [1]:
```
using Symbolics
```

In [2]:
```
gB(B,F)=0.05*B*(1-B/150000)
gF(B,F)=0.08*F*(1-F/400000)
cB(B,F)=alpha*B*F
cF(B,F)=alpha*B*F
```

Out[2]:
```
cF (generic function with 1 method)
```

In [3]:
```
@variables B,F,alpha
```

Out[3]:
```
3-element Vector{Num}:
     B
     F
 alpha
```

Model the growth and competition between whales using a differential equation

$$\frac{dX}{dt} = G(X)$$

where

$$X = (B, F) \quad \text{and} \quad G(B, F) = \begin{bmatrix} g_B - c_B \\ g_F - c_F \end{bmatrix}$$

In [4]:
```
G(B,F)=[gB(B,F)-cB(B,F),gF(B,F)-cF(B,F)]
```

Out[4]:
```
G (generic function with 1 method)
```

In [5]:
```
G(B,F)
```

Out[5]:
```
2-element Vector{Num}:
 0.05B*(1 - (1//150000)*B) - B*F*alpha
 0.08F*(1 - (1//400000)*F) - B*F*alpha
```

Trying to solve for $G(B,F)=0$ for $B>0$ and $F>0$. Note that we can divide the first component by B and the second by F to obtain a linear system.

In [6]:
```
Q=simplify.(G(B,F)./[B,F])
```

Out[6]:
```
2-element Vector{Num}:
 0.05 - 3.3333333333333335e-7B - F*alpha
 0.08 - 2.0000000000000002e-7F - B*alpha
```

In [7]:
```
Qs="fQ(B,F)="*string(Symbolics.toexpr(Q))
eval(Meta.parse(Qs))
```

Out[7]:
```
fQ (generic function with 1 method)
```

In [8]:
```
fQ(B,F)
```

Out[8]:
```
2-element Vector{Num}:
 0.05 - 3.3333333333333335e-7B - F*alpha
 0.08 - 2.0000000000000002e-7F - B*alpha
```

In [9]:
```
b=fQ(0,0)
```

Out[9]:
```
2-element Vector{Num}:
 0.05
 0.08
```

In [10]:
```
A=Symbolics.jacobian(fQ(B,F),[B,F])
```

Out[10]:
```
2×2 Matrix{Num}:
 -3.33333e-7   -alpha
      -alpha  -2.0e-7
```

Note that $Q(X)=Q(0)+(DQ(0))(X-0)$ since Q is linear.

In [11]:
```
X0=-A\b
```

Out[11]:
```
2-element Vector{Num}:
 (0.05 + (-(0.08 - 150000.0alpha)*alpha) / (2.0000000000000002e-7 - 3.0e6(a
lpha^2))) / 3.3333333333333335e-7
                                               (0.08 - 150000.0alpha) / (2.0000
000000000002e-7 - 3.0e6(alpha^2))
```

In [12]:
```
X0s="fX0(alpha)="*string(Symbolics.toexpr(X0))
eval(Meta.parse(X0s))
```

Out[12]:
```
fX0 (generic function with 1 method)
```

In [13]:
```
fX0(alpha)
```

Out[13]: 2-element Vector{Num}:

```
 (0.05 + ((-0.08 + 150000.0alpha)*alpha) / (2.0000000000000002e-7 - 3.0e6(alpha^2))) / 3.3333333333333335e-7
                                                (0.08 - 150000.0alpha) / (2.0000000000000002e-7 - 3.0e6(alpha^2))
```

In [14]:

```
fX0(1e-7)
```

Out[14]:

```
2-element Vector{Float64}:
  35294.11764705884
 382352.94117647054
```

In [15]:

```
fX0(0.0)
```

Out[15]:

```
2-element Vector{Float64}:
 150000.0
 400000.0
```

In [16]:

```
fX0(1e-8)
```

Out[16]:

```
2-element Vector{Float64}:
 138207.31096644967
 393089.6344516775
```

In [17]:

```
fX0(1e-6)
```

Out[17]:

```
2-element Vector{Float64}:
 75000.00000000001
 25000.0
```

In [18]:

```
fX0(1e-5)
```

Out[18]:

```
2-element Vector{Float64}:
 7905.270180120063
 4736.490993995997
```

In [19]:

```
fX0(1e-4)
```

Out[19]:

```
2-element Vector{Float64}:
 799.0053267022054
 497.33664891099266
```

In [20]:

```
0.05/400000
```

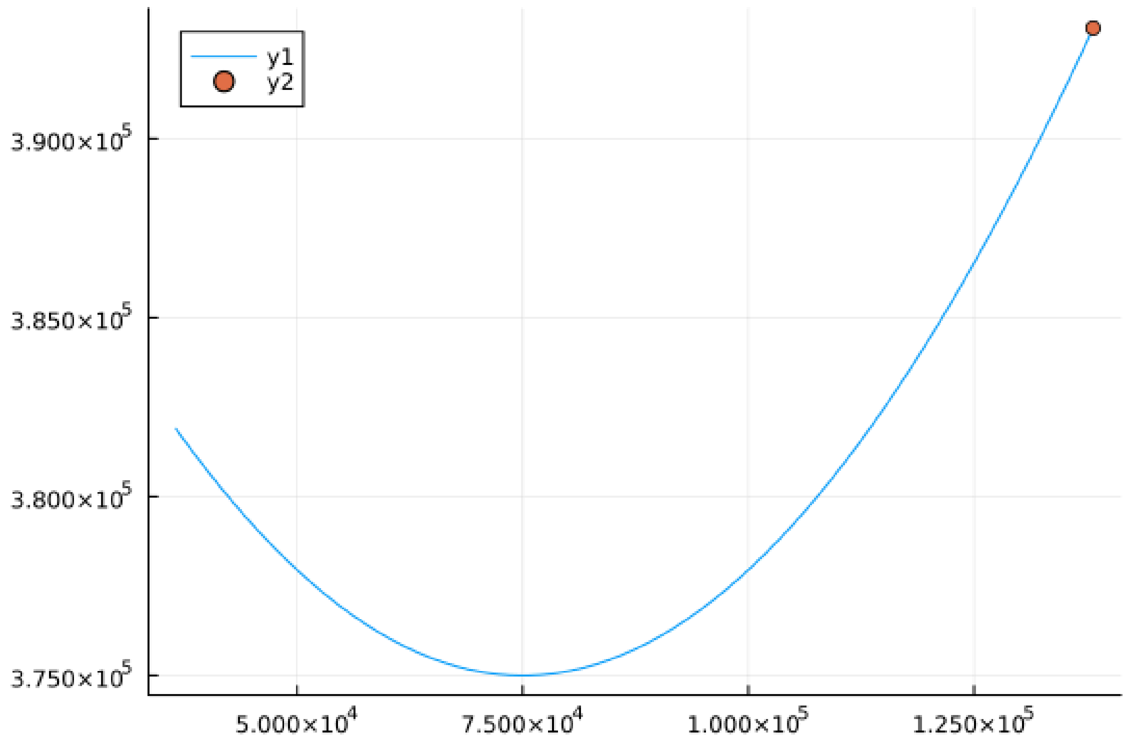
Out[20]:

```
1.25e-7
```

In [21]:

```
using Plots
plot(alpha->fX0(alpha)[1],alpha->fX0(alpha)[2],1e-8:0.1e-08:1e-7)
scatter!((fX0(1e-8)[1],fX0(1e-8)[2]))
```

Out[21]:



Now lets set $\alpha = 10^{-7}$ and linearize G about the corresponding equilibrium to see stability.

In [22]:
```
alpha=1e-7
G(B,F)
```

Out[22]:
```
2-element Vector{Num}:
  0.05B*(1 - (1//150000)*B) - 1.0e-7B*F
 -1.0e-7B*F + 0.08F*(1 - (1//400000)*F)
```

Now linearlize G as

$$G(X) \approx G(X_0) + (DG(X_0))(X - X_0)$$

In [23]:
```
myX0=fX0(alpha)
```

Out[23]:
```
2-element Vector{Float64}:
  35294.11764705884
 382352.94117647054
```

In [24]:
```
DG=Symbolics.jacobian(G(B,F),[B,F])
```

Out[24]:
```
2×2 Matrix{Num}:
 -3.33333e-7B + 0.05(1 - (1//150000)*B) - 1.0e-7F  …
-1.0e-7B
                                         -1.0e-7F     -1.0e-7B - 2.0e-7F +
0.08(1 - (1//400000)*F)
```

In [25]:
```
DG0=substitute(DG,[B=>myX0[1],F=>myX0[2]])
myDG0=eval(Symbolics.toexpr(DG0))
```

Out[25]:
```
2×2 Matrix{Float64}:
 -0.0117647  -0.00352941
 -0.0382353  -0.0764706
```

In [26]:
```
using LinearAlgebra
```

In [27]:
```
eigvals(myDG0)
```

Out[27]:
```
2-element Vector{Float64}:
 -0.07849294196161641
 -0.009742352156030629
```

This means along the eigendirections the populations decrease towards the fixed point X_0. In order words, the fixed point is stable in a neighborhood.

In [ ]:
```

```