On a three-dimensional tumor-virus compartmental model

This Mathematica Notebook is a supplementary material to the paper“On a three-dimensional and two four-dimensional
oncolytic viro-therapy models”. It contains some of the calculations and illustrations appearing in the paper.

1) Section 2 (in paper): Deterministic model with Logistic growth [Tian2011]

0) Definition of the model [Tian11]:

In[1]:=
```
SetDirectory[NotebookDirectory[]];
AppendTo[$Path,Directory];
Clear["Global`*"];
(*Some aliases*)
Format[βv]:=Subscript[β,v];Format[βy]:=Subscript[β,y];
(****** Four dim Deterministic  epidemic model with Logistic growth ****)
x1=λ  x(1-(x+y)/K)-   β x v ;
y1=β x v -βy y z- γ y;
v1=-β x v - βv  v z+ b  γ y - δ v;

par={λ,δ,β,b};X={x,y,v};
val={0.36,0.44,0.11,28};
parN=Thread[par→val];
par2=Thread[{λ,δ}→{0.36,0.44}];
cp={λ>0,δ>0,β>0,b>1};
c3dim={βv→0,βy→0,γ→1, K→1};
cnTian={λ→0.36,β→0.11,δ→0.44,K→1,γ→1,βv→0, βy→0}(*Numerical values of Tian*);

dyn={x1,y1,v1}/.βy→0/.βv→0(*Tian 11 case with K>0*);
dynKeq1=dyn/.c3dim;
           x'
Print[" (y')=",dyn//FullSimplify//MatrixForm,
           v'
                                                   x'
"and the reparametrized dynamics [Tian 2011] are  (y')=",
                                                   v'
dynKeq1//FullSimplify//MatrixForm]

R0=b β K/(β K+δ);b0=1+δ/(β K);
Print["R0=", R0, " , b0=", b0]
```

$$\begin{pmatrix} x' \\ y' \\ v' \end{pmatrix} = \begin{pmatrix} -v\,x\,\beta + x\left(1 - \frac{x+y}{K}\right)\lambda \\ v\,x\,\beta - y\,\gamma \\ b\,y\,\gamma - v\,(x\,\beta + \delta) \end{pmatrix}$$

and the reparametrized dynamics [Tian 2011] are $\begin{pmatrix} x' \\ y' \\ v' \end{pmatrix} = \begin{pmatrix} -x\,(v\,\beta + (-1 + x + y)\,\lambda) \\ -y + v\,x\,\beta \\ b\,y - v\,(x\,\beta + \delta) \end{pmatrix}$

$R0 = \frac{b\,K\,\beta}{K\,\beta + \delta}$, $b0 = 1 + \frac{\delta}{K\,\beta}$

Fixed points and analysis of the Stability via Routh Hurwitz:

In[20]:=
```
cfp=Solve[Thread[dyn=={0,0,0}],{x,y,v}]//FullSimplify;
fp=X/.cfp;

Print["Number of fixed points is ",Length[fp]," The third FP is E*="]
fp[[3]]//FullSimplify
fpN=fp/.c3dim/.parN;
Print["in first fig. E*=",fpN[[3]]]
(*"Jacobian is"*)
Jac=Grad[dyn,{x,y,v}]//FullSimplify;
J0=Jac/.cfp[[1]];J0//MatrixForm;
Print["J(E_K) and its det are"]
J1=Jac/.cfp[[2]];J1//MatrixForm
Det[J1]//Factor
Print["J(E_*) "]
Jst=Jac/.cfp[[3]]//FullSimplify;Jst//MatrixForm
(*Jstcr=Jst/.b→b0//FullSimplify;*)

(*Routh Hurwitz conditions for the stability of E_**)
pc=Collect[Det[ψ IdentityMatrix[3]-Jst],ψ];
coT=CoefficientList[pc,ψ]//FullSimplify;
Print["a1=",a1=Apart[coT[[3]]]//.c3dim, ", a2=",a2=coT[[2]]//.c3dim, ", a3=",a3=coT[[1]]//.c3dim]
H2=a1*a2-a3;
Print["Hurw(b) when K=γ=1 is H2=", H2//.c3dim//FullSimplify]
Print["H2(b0) when K=γ=1 is",H2/.b→b0//.c3dim//FullSimplify]
Print["Denominator of H2 is posi",Denominator[Together[H2]]/.K→1//FullSimplify]
Together[H2//FullSimplify];
ϕb=Collect[Numerator[Together[H2]]/(δ λ),b]/.K→1//FullSimplify;
(*Print["ϕ(b)=", Collect[ϕb,b]]*)
cofi=CoefficientList[ϕb,b]//FullSimplify;
Print["Coefficients of numerator ϕ(b)are {B0,B1,B2,B3,B4}=",cofi]
(*ϕb/.b→1//FullSimplify*)

(*FindInstance[Join[{cofi[[4]]>0&&cofi[[3]]<0},cp],par];
Reduce[Join[{cofi[[4]]>0&&cofi[[3]]<0},cp],par]//FullSimplify;*)
Print["There is no solution of B2<0 and B3>0 : "]
Solve[Join[{cofi[[4]]>0&&cofi[[3]]<0},cp],λ]
Print["value of ϕ(b) at crit b is posi "]
ϕb/.b→b0/.K→1//FullSimplify
```

Number of fixed points is 3 The third FP is E*=

Out[23]= $\left\{\frac{\delta}{(-1+b)\,\beta}, \frac{((-1+b)\,K\,\beta - \delta)\,\delta\,\lambda}{(-1+b)\,\beta\,((-1+b)\,K\,\beta\,\gamma + \delta\,\lambda)}, \frac{\gamma\,((-1+b)\,K\,\beta - \delta)\,\lambda}{\beta\,((-1+b)\,K\,\beta\,\gamma + \delta\,\lambda)}\right\}$

in first fig. E*={0.148148, 0.0431317, 2.64672}

J(E_K) and its det are

Out[29]//MatrixForm=

$$\begin{pmatrix} -\lambda & -\lambda & -K\beta \\ 0 & -\gamma & K\beta \\ 0 & b\gamma & -K\beta-\delta \end{pmatrix}$$

Out[30]= $\gamma\,(-K\beta + bK\beta - \delta)\,\lambda$

J(E_*)

Out[32]//MatrixForm=

$$\begin{pmatrix} \frac{\delta\lambda}{K\beta-bK\beta} & \frac{\delta\lambda}{K\beta-bK\beta} & -\frac{\delta}{-1+b} \\ \frac{\gamma\,((-1+b)K\beta-\delta)\,\lambda}{(-1+b)K\beta\gamma+\delta\lambda} & -\gamma & \frac{\delta}{-1+b} \\ \frac{\gamma\,(K(\beta-b\beta)+\delta)\,\lambda}{(-1+b)K\beta\gamma+\delta\lambda} & b\gamma & \frac{b\delta}{1-b} \end{pmatrix}$$

$a_1=\dfrac{-1+b+b\delta}{-1+b}+\dfrac{\delta\lambda}{(-1+b)\beta}$, $a_2=$

$$\frac{\delta\lambda\left((-1+b)\beta(-1+b+\beta-b\beta+\delta+b\delta)+\left((-1+b)^2\beta+b\delta^2\right)\lambda\right)}{(-1+b)^2\beta\,((-1+b)\beta+\delta\lambda)},\ a_3=\delta\left(1+\frac{\delta}{\beta-b\beta}\right)\lambda$$

Hurw(b) when K=γ=1 is H2=
$$\delta\lambda\left(-1+\frac{\delta}{(-1+b)\beta}-\frac{(\beta(-1+b+b\delta)+\delta\lambda)\left((-1+b)^2\beta^2-b\delta^2\lambda-(-1+b)\beta(-1+\delta-\lambda+b(1+\delta+\lambda))\right)}{(-1+b)^3\beta^2((-1+b)\beta+\delta\lambda)}\right)$$

H2(b0) when K=γ=1 is $(1+\beta+\delta)\,\lambda\,(1+\beta+\delta+\lambda)$

Denominator of H2 is posi $(-1+b)^3\beta^2((-1+b)\beta+\delta\lambda)$

Coefficients of numerator ϕ(b) are {B0,B1,B2,B3,B4}=
$$\left\{\beta(1+\lambda)(-\beta+\delta\lambda),\ -\beta^3(-1+\delta)+\delta^3\lambda^2-\beta\delta\lambda(2+3\delta+2\lambda)+\beta^2\left(3+3\delta-\delta^2+3\lambda\right),\ \beta\left(\beta^2(-3+2\delta)-3\beta(1+2\delta+\lambda)+\delta\lambda(1+\delta(3+\delta)+\lambda)\right),\ \beta^2(1-\beta(-3+\delta)+\delta(3+\delta)+\lambda),\ -\beta^3\right\}$$

There is no solution of B2<0 and B3>0 :

••• Solve: When parameter values satisfy the condition ≪1≫, the solution set contains a full-dimensional component; use Reduce for complete solution information.

Out[45]= { }

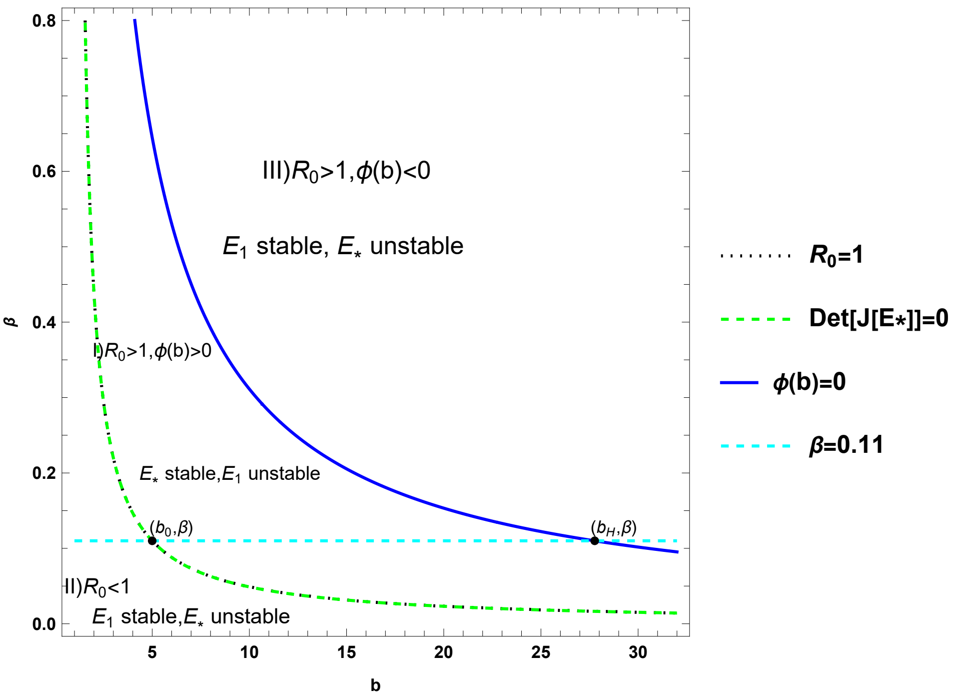
value of ϕ(b) at crit b is posi

Out[47]= $\dfrac{\delta^3(1+\beta+\delta)\times(1+\lambda)\times(1+\beta+\delta+\lambda)}{\beta}$

Bifurcation diagram of codimension 2:

In[48]:=
```
bm=1; bM=32; ym=0; yM=0.8;
R01=R0/.c3dim/.par2;
a3N=a3/.par2;
a1N=a1/.par2;
phiN=ϕb/.par2//FullSimplify;

pt1s=Text["(b0,β)",Offset[{10,6},{b0/.c3dim/.par2/.β→0.11,0.11}]];
pt1={PointSize[Medium],Style[Point[{b0/.c3dim/.par2/.β→0.11,0.11}],Black]};
ptHs=Text["(bH,β)",Offset[{10,6},{27.7664,0.11}]];
ptH1={PointSize[Medium],Style[Point[{27.7664,0.11}],Black]};
R1=Style[Text["I)R0>1,ϕ(b)>0",{5,0.36}],10];R1a=Style[Text["E* stable,E1 unstable",{9,0.2}],1
R2=Style[Text["II)R0<1",{2.2,0.05}],11];R2a=Style[Text["E1 stable,E* unstable",{7,0.01}],11];
R3=Style[Text["III)R0>1,ϕ(b)<0",{15,0.6}],13];R3a=Style[Text["E1 stable, E* unstable ",{15,0.
epi={pt1,pt1s,ptHs,ptH1,R1,R1a,R2,R2a,R3,R3a};

pR0=ContourPlot[R01==1,{b,bm,bM},{β,ym,yM},ContourStyle→{Black,Dotted},
 FrameLabel→{"b","β"},LabelStyle→{Black,Bold},Frame→True,PlotLegends→{"R0=1"}];
pphi1=ContourPlot[phiN==0,{b,bm,bM},{β,ym,yM},ContourStyle→{Blue,Bold},PlotPoints → 200,
 FrameLabel→{"b","β"},LabelStyle→{Black,Bold},Frame→True,PlotLegends→{"ϕ(b)=0"}];
pa1=ContourPlot[a1N==0,{b,bm,bM},{β,ym,yM},ContourStyle→{Red,Bold},
 FrameLabel→{"b","β"},LabelStyle→{Black,Bold},Frame→True,PlotLegends→{"Tr[J[E*]]=0"}];
pa3=ContourPlot[a3N==0,{b,bm,bM},{β,ym,yM},ContourStyle→{Green,Dashed},PlotPoints → 180,
 FrameLabel→{"b","β"},LabelStyle→{Black,Bold},Frame→True,PlotLegends→{"Det[J[E*]]=0"}];
pbeta=ContourPlot[β==0.11,{b,bm,bM},{β,ym,yM},ContourStyle→{Cyan,Dashed},PlotPoints → 180,
 FrameLabel→{"b","β"},LabelStyle→{Black,Bold},Frame→True,PlotLegends→{"β=0.11"}];

 peq={pR0,pa3,(*pa1,*)pphi1,pbeta};
pcut=Show[peq,Epilog→epi]
Export["map3.pdf",pcut]
```

Out[67]=



Out[68]= map3.pdf

3D parametric plot:

In[69]:=

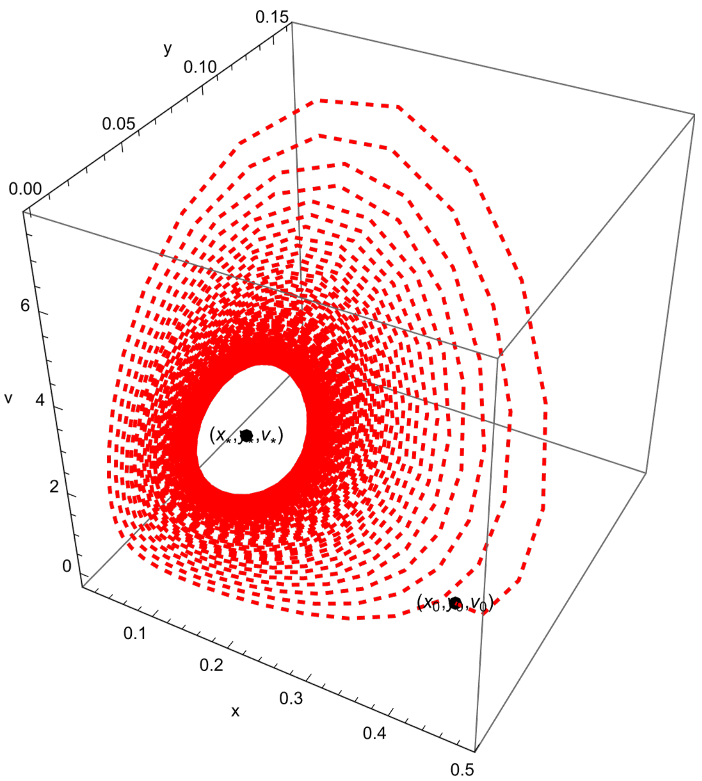
```
Xt=Map[#[t]&,X ];
ct=Thread[X→Xt];
dynt=dynKeq1/.ct;
x1=dynt[[1]] ;
y1=dynt[[2]];
v1=dynt[[3]];
X0={0.4,0.05,0.05};
ode={x'[t]==x1,y'[t]==y1,v'[t]==v1,x[0]==X0[[1]],y[0]==X0[[2]],v[0]==X0[[3]]};
T=2800;
odeN=ode/.parN
solr=NDSolve[odeN,{x,y,v},{t,0,T}];

cyR=ParametricPlot3D[{  x[t],y[t],v[t]}/.solr,{t,0,T}, BoxRatios→{1,1,1},AxesLabel→{"x","y",
PlotRange→Full,PlotStyle→{Red,Dashed}];
pt0={PointSize@.02, Style[Point[X0],Black]};pt0s=Text[Style["(x₀,y₀,v₀)", 10], X0];
pts={PointSize@.02, Style[Point[fpN[[3]]],Black]}; ptss=Text[Style["(x⋆,y⋆,v⋆)", 10] ,fpN[[3]]
];
pts={ptss,pt0,pt0s,pts};
```

Out[78]= $\{x'[t] == -0.11\, v[t] \times x[t] + 0.36\, x[t]\,(1 - x[t] - y[t]),\ y'[t] == 0.11\, v[t] \times x[t] - y[t],$
$v'[t] == -0.44\, v[t] - 0.11\, v[t] \times x[t] + 28\, y[t],\ x[0] == 0.4,\ y[0] == 0.05,\ v[0] == 0.05\}$

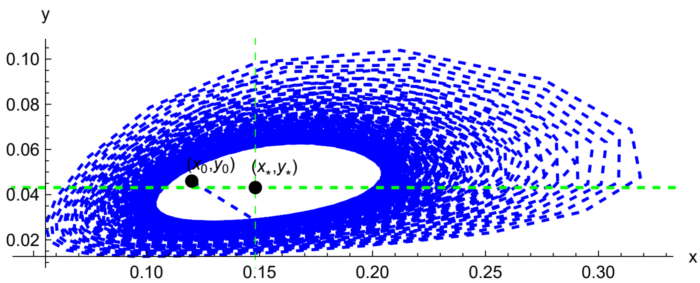
In[ ]:=

```
cyR3=Show[{cyR}, Graphics3D[pts], Axes → True,
 BoxRatios → 1]
```

Out[ ]=



```
(****************b=28; different initial values   *****************)
b=28;T1=4000;
x0n=0.12; y0n=0.046;v0n=0.01;
ode={x'[t]==x1,y'[t]==y1,v'[t]==v1,x[0]==x0n,y[0]==y0n,v[0]==v0n};
sol=NDSolve[ode/.cnTian,{x,y,v},{t,0,T1}];
(*****..Parametric plot conditions***)

cyB=ParametricPlot[{  x[t],(y[t])}/.sol,{t,0,T1}, AxesLabel→{"x","y"},
PlotRange→Full,PlotStyle→{Dashed,Blue}];
pyn=Plot[(fp[[3,2]]/.cnTian),{t,0,T1},PlotStyle→{Dashed,Green}];
cyBall=Show[{cyB,pyn, Graphics[{Green,Dashed,
Line[{{fp[[3,1]]/.cnTian,0},{fp[[3,1]]/.cnTian,1}}]}] },
Epilog→{{Text["(x*,y*)",Offset[{10,10},{(fp[[3,1]]/.cnTian),(fp[[3,2]]/.cnTian)}]],
{PointSize[Large],Style[Point[{(fp[[3,1]]/.cnTian),(fp[[3,2]]/.cnTian)}],Black]}},
{PointSize[Large],Point[{x0n,y0n}]},Text["(x0,y0)",Offset[{10,8},{x0n,y0n}]]}]


cyr=ParametricPlot[{  x[t],(y[t])}/.solr,{t,0,T}, AxesLabel→{"x","y"},PlotStyle→{Red}];
cyRall=Show[{cyr},PlotRange→{{0.05,0.35},{0,0.12}},Epilog→
{{PointSize[Large],Point[{X0[[1]],X0[[2]]}]},Text["(x0,y0)",Offset[{10,8},{X0[[1]],X0[[2]]}]]}];

Export["T11R.pdf",cyr]
Export["T11B.pdf",cyB]
```

Out[•]=


Out[•]= T11R.pdf

Out[•]= T11B.pdf

Computations of the Jacobians and Eigenvalues using EcoEvo package:

In[ ]:=
```
<<EcoEvo`
(*EcoEvoDocs;*)
(********Analysis of the Model, K=γ=1; must redefine fp, cp***)
ClearParameters;Clear["b"];
SetModel[{Pop[x]→{Equation:→dynKeq1[[1]],Color→Red},Pop[y]→
{Equation:→(dynKeq1[[2]]),Color→Green},
Pop[v]→{Equation:→(dynKeq1[[3]]),Color→Purple},
 Parameters:→(cp)}]

fpT=SolveEcoEq[]//FullSimplify

J0T=EcoJacobian[fpT[[1]]]//FullSimplify;
J1T=EcoJacobian[fpT[[2]]]//FullSimplify;
Jst=EcoJacobian[fpT[[3]]]//FullSimplify;
Print["Jac(E0)=",J0T//MatrixForm]
Print["Jac(E1)=",J1T//MatrixForm]
Print["Jac(E*)=",Jst//MatrixForm]
Print["Eigenvalues of E1 are:",eiT=EcoEigenvalues[fpT[[2]]]//FullSimplify]

Print["b0=bs1=",bs1=Apart[Last[Last[Reduce[Join[{eiT[[2]]>0},cp],b]]]]]
```

Out[ ]= EcoEvo Package Version 1.6.4 (November 5, 2021)
Christopher A. Klausmeier <christopher.klausmeier@gmail.com>

Out[ ]=
$$\left\{\{x\to 0,\ y\to 0,\ v\to 0\},\ \{x\to 1,\ y\to 0,\ v\to 0\},\ \left\{x\to \frac{\delta}{(-1+b)\,\beta},\ y\to \frac{((-1+b)\,\beta-\delta)\,\delta\,\lambda}{(-1+b)\,\beta\,((-1+b)\,\beta+\delta\,\lambda)},\ v\to \frac{((-1+b)\,\beta-\delta)\,\lambda}{\beta\,((-1+b)\,\beta+\delta\,\lambda)}\right\}\right\}$$

$$\mathrm{Jac}(E_0)=\begin{pmatrix}\lambda & 0 & 0\\ 0 & -1 & 0\\ 0 & b & -\delta\end{pmatrix}$$

$$\mathrm{Jac}(E_1)=\begin{pmatrix}-\lambda & -\lambda & -\beta\\ 0 & -1 & \beta\\ 0 & b & -\beta-\delta\end{pmatrix}$$

$$\mathrm{Jac}(E_*)=\begin{pmatrix}\frac{\delta\,\lambda}{\beta-b\,\beta} & \frac{\delta\,\lambda}{\beta-b\,\beta} & -\frac{\delta}{-1+b}\\ \frac{((-1+b)\,\beta-\delta)\,\lambda}{(-1+b)\,\beta+\delta\,\lambda} & -1 & \frac{\delta}{-1+b}\\ \frac{(\beta-b\,\beta+\delta)\,\lambda}{(-1+b)\,\beta+\delta\,\lambda} & b & \frac{b\,\delta}{1-b}\end{pmatrix}$$

Eigenvalues of E_1 are:

$$\left\{\frac{1}{2}\times\left(-1-\beta-\delta-\sqrt{(1+\beta+\delta)^2-4\,(\beta-b\,\beta+\delta)}\right),\ \frac{1}{2}\times\left(-1-\beta-\delta+\sqrt{(1+\beta+\delta)^2-4\,(\beta-b\,\beta+\delta)}\right),\ -\lambda\right\}$$

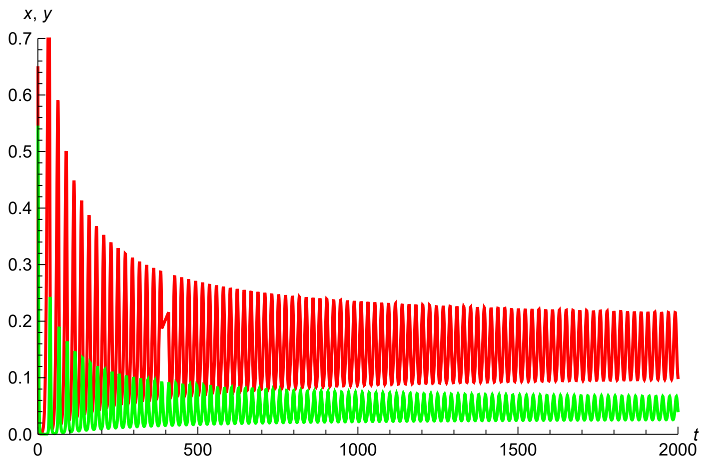
$$b_0=b_{s1}=1+\frac{\delta}{\beta}$$

In[ ]:=
```
λ=0.36;β=0.11;δ=0.44;b=28;K=1;γ=1;T1=4000;T2=2000;T=800;
X0={x→0.5,y→0.5,v→1.5};
fpN=fpT(*Numerical values of FP*)
Print["Eigenvalues corresponding to E* are:"]
EcoEigenvalues[fpN[[3]]](*Eigenvalues corresponding to E*  **)
solE3=EcoSim[RuleListAdd[(fpN[[3]]),X0],T2];

Fig5T=PlotDynamics[{solE3[[1]],solE3[[2]]},PlotRange→{{0,T2},{0,0.7}}]
Export["Fig5T.pdf",Fig5T]
```

Out[ ]= {{x → 0, y → 0, v → 0}, {x → 1, y → 0, v → 0}, {x → 0.148148, y → 0.0431317, v → 2.64672}}

Eigenvalues corresponding to E* are:

Out[ ]= {-1.51022, 0.000296187 + 0.298909 i, 0.000296187 - 0.298909 i}

Out[ ]=


Out[ ]= Fig5T.pdf

In[ ]:=

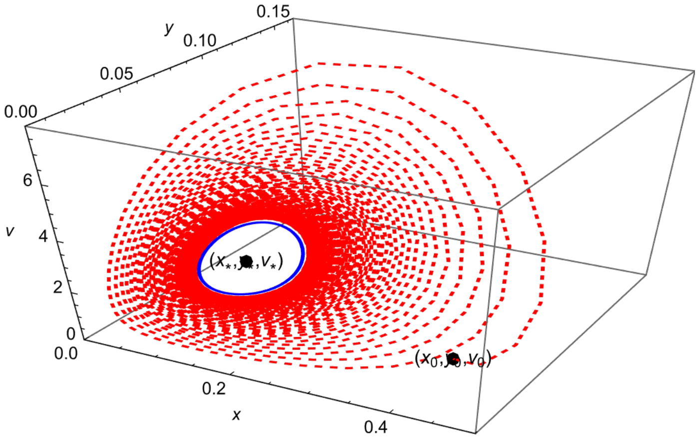
```
E2n = RuleListTweak[fpT[[3]], {x, y}, {0.1, 0.1}]
Print["Time plot around E2"]
sol = EcoSim[E2n, T2];
lc = FindEcoCycle[FinalSlice[sol]];
cy3D = RuleListPlot[lc, {x, y, v}, PlotStyle → Blue];
cy113D = Show[{cy3D, cyR3}]
Export["cy113D.pdf", cy113D]
```

Out[ ]= {v → 2.64672, x → 0.248148, y → 0.143132}

Time plot around E2

FindEcoCycle: Failed to converge to the requested accuracy within 100 iterations.

Out[ ]=



Out[ ]= cy113D.pdf

1) Numerical simulations:

Bifurcation Diagram:

In[ ]:=

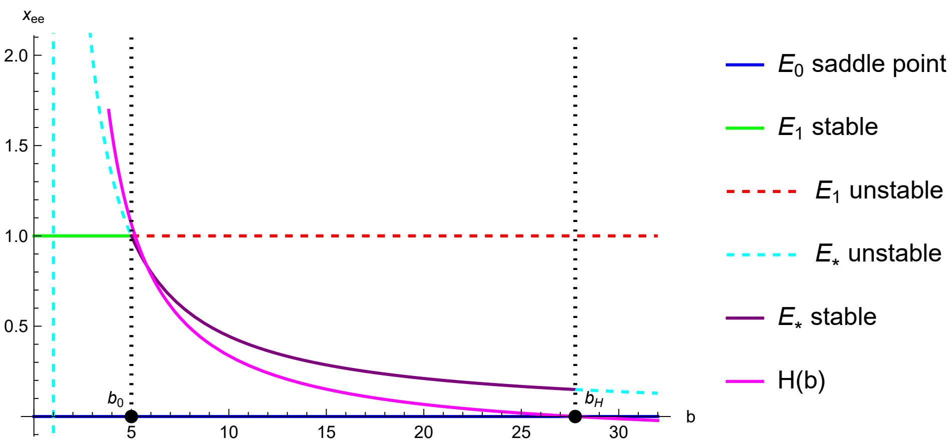
```
Clear["b"];
bL=32;
Print["b0=",bs1//N]
cb=NSolve[(ϕb)==0,b,WorkingPrecision→10]
bM=Max[Table[Re[b/.cb⟦i⟧],{i,Length[cb]}]];
Print["bH=",bH=N[bM,30]]
fpT//N;
linb0=Line[{{ bs1,0},{ bs1,10}}];
lib0=Graphics[{Thick,Black,Dotted,linb0}];
linbH=Line[{{ bH,0},{ bH,10}}];
libH=Graphics[{Thick,Black,Dotted,linbH}];
px0=Plot[0,{b,0,bL},PlotStyle→{Blue},PlotRange→All,PlotLegends→{"E0 saddle point"}];
px1a=Plot[{1},{b,0,bs1},PlotStyle→{Green},PlotRange→All,PlotLegends→{"E1 stable"}];
px1b=Plot[1,{b,bs1,bL},PlotStyle→{Red,Dashed},PlotRange→All,
PlotLegends→{"E1 unstable"}];
pxe1=Plot[{x/.fpT⟦3⟧},{b,0, bs1},PlotStyle→{Cyan,Dashed},PlotRange→All,
PlotLegends→{"E* unstable"}];
pxe2=Plot[{x/.fpT⟦3⟧},{b,bs1, bH},PlotStyle→{Purple},PlotRange→All,
PlotLegends→{"E* stable"}];
pxe2a=Plot[{x/.fpT⟦3⟧},{b,bH, bL},PlotStyle→{Cyan,Dashed},PlotRange→All];
epi={Text["b0",Offset[{-8,10},{ bs1,0}]],
{PointSize[Large],Style[Point[{ bs1,0}],Black]},Text["bH",Offset[{10,10},{ bH,0}]],
{PointSize[Large],Style[Point[{ bH,0}],Black]}};
pHb=Plot[{H2},{b,0,bL},PlotStyle→{Magenta},Epilog→epi,AxesLabel→{"b","H(b)"},PlotLegends→{"H
bif11T=Show[{px0,px1a,px1b,pxe1,pxe2,pxe2a,pHb,lib0,libH},Epilog→epi,
PlotRange→{{0,bL},{0,2}},AxesLabel→{"b","xee"}]
pHbS=Show[{pHb,lib0}];
Export["pHb.pdf",pHbS]
Export["bif11T.pdf",bif11T]
```

b0=5.

Out[ ]= {{b → −0.347032}, {b → 0.63211 − 0.406197 i}, {b → 0.63211 + 0.406197 i}, {b → 27.7664}}

bH=27.7664

Out[ ]=



Out[ ]= pHb.pdf

Out[ ]= bif11T.pdf

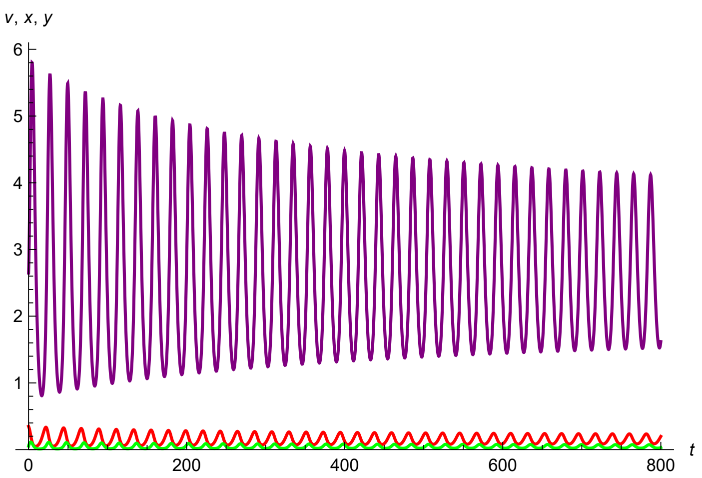
Time plot when b=28:

In[∘]:=
```
xm = 0; ym = 0; xM = 1; yM = 1; b = 28;
E2n = RuleListTweak[fpT[[3]], {x, y}, {0.2, 0.01}]
Print["Time plot around E2"]
sol = EcoSim[E2n, T]
psol = PlotDynamics[sol]
Print["Finding the cycle using time slice"]
lc = FindEcoCycle[FinalSlice[sol]]
Print["Final time slice is"]
FinalTime[lc]

eq = {Drop[fpT[[1]], {3}], Drop[fpT[[2]], {3}], Drop[fpT[[3]], {3}]}
Print["FP="]
fpT
Print["Eigenvalues of FP:", EcoEigenvalues[fpT]]
pp = Show[PlotEcoPhasePlane[{x, xm, xM}, {y, ym, yM}, PlotPoints → 180],
   RuleListPlot[eq, PlotMarkers → EcoStableQ[fpT]]];
(*sp11=Show[pp,RuleListPlot[Drop[sol,{3}],PlotStyle→Blue]]
  Export["sp11.pdf",sp11]*)
(*RuleListPlot[Append[eq,lc]]*)
cy1B = RuleListPlot[Drop[sol, {3}], PlotStyle → Blue];
```

Out[∘]= {v → 2.64672, x → 0.348148, y → 0.0531317}

Time plot around E2

Out[∘]= {x → InterpolatingFunction[Domain: {{0., 800.}} Output: scalar],
y → InterpolatingFunction[Domain: {{0., 800.}} Output: scalar],
v → InterpolatingFunction[Domain: {{0., 800.}} Output: scalar]}

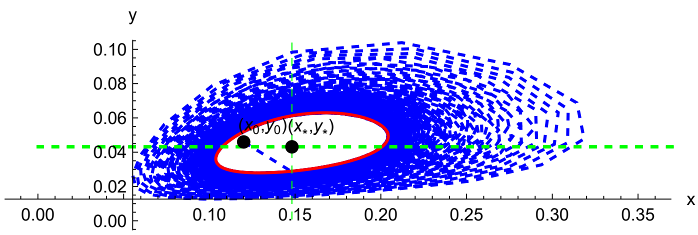
Out[∘]=


Finding the cycle using time slice

FindEcoCycle: Failed to converge to the requested accuracy within 100 iterations.

```
Out[∘]= {v → InterpolatingFunction[ Domain: {{0., 21.2}} Output: scalar ],
         x → InterpolatingFunction[ Domain: {{0., 21.2}} Output: scalar ],
         y → InterpolatingFunction[ Domain: {{0., 21.2}} Output: scalar ]}

Final time slice is

Out[∘]= 21.1984

Out[∘]= {{x → 0, y → 0}, {x → 1, y → 0}, {x → 0.148148, y → 0.0431317}}

FP=

Out[∘]= {{x → 0, y → 0, v → 0}, {x → 1, y → 0, v → 0}, {x → 0.148148, y → 0.0431317, v → 2.64672}}

Eigenvalues of FP:{{-1., -0.44, 0.36}, {-2.54436, 0.994357, -0.36},
  {-1.51022, 0.000296187 + 0.298909 i, 0.000296187 - 0.298909 i}}

In[∘]:= T = 3000;
E2nb = RuleListTweak[fpT[[3]], {x, y}, {0.1, 0.01}]
solb = EcoSim[E2nb, T];
lcb = FindEcoCycle[FinalSlice[solb]];
lcbp = RuleListPlot[lcb, {x, y}, PlotStyle → {Black}];
cy1B = RuleListPlot[Drop[solb, {3}], PlotStyle → {Dashed, Blue}];
cy11E = Show[
  {cyBall, RuleListPlot[lc, {x, y}, PlotStyle → {Red}], pyn, Graphics[{Green, Dashed,
Line[{{x /. x → fp[[3, 1]], 0}, {x /. x → fp[[3, 1]], 1}}]}]},
  PlotRange → {{0, 0.35}, {0, 0.1}}, Epilog → {Text["(x*,y*)",
Offset[{10, 10}, {(x /. x → fp[[3, 1]]), (y /. y → fp[[3, 2]])}]],
{PointSize[Large], Style[Point[{(x /. x → fp[[3, 1]]), (y /. y → fp[[3, 2]])}], Black]},
    {PointSize[Large], Point[{x0n, y0n}]},
    Text["(x0,y0)", Offset[{10, 8}, {x0n, y0n}]]}]
(*cy11F=Show[{cy11,RuleListPlot[lc,{x,y},PlotStyle→{Red}]}]*)

Export["cy11E.pdf", cy11E]
(*Export["cy11F.pdf",cy11F]*)

Out[∘]= {v → 2.64672, x → 0.248148, y → 0.0531317}

FindEcoCycle: Failed to converge to the requested accuracy within 100 iterations.
```

Out[∘]=




```
Out[∘]= cy11E.pdf
```