The Lotka - Volterra cannibalism time buc

$$\frac{d}{dt}R = r\,R\left(1-\frac{R}{K}\right) - \alpha\,R\sum_{j=1}^{k}\left(1-x_j\right)n_j$$

$$\frac{d}{dt}n_i = \boxed{\epsilon\,\alpha\,R\,(\mathbf{1}-\boldsymbol{x_i})\,\boldsymbol{n_i} - \delta\,(\mathbf{1}-\boldsymbol{x_i})\,\boldsymbol{n_i} - \left(\sum_{j=1}^{k}\beta\left[x_j\right]x_j\,n_j\right)(\mathbf{1}-\boldsymbol{x_i})\,\boldsymbol{n_i}} +$$

$$\boxed{\gamma\,\beta[x_i]\,\boldsymbol{x_i}\,\boldsymbol{n_i}\sum_{j=1}^{k}\left(1-x_j\right)n_j - \delta\,\boldsymbol{x_i}\,\boldsymbol{n_i}} \quad \text{for } i = 1, \ldots, k$$

MONOMORPHIC RESIDENT POPULATION

```
Clear[α, β, β0, β1, γ, δ, ϵ, r, K, n, R]
```

Monomorphic resident population dynamics:

(R = resource density; n = resident population density)

```
dLogR = r - r R / K - α (1 - x) n;
dLogn = ϵ α (1 - x) R - δ + γ β[x] x (1 - x) n - (1 - x) β[x] x n;
```

Monomorphic resident population equilibrium:

```
Solve[{0 == dLogR, 0 == dLogn}, {R, n}]
```

$$\left\{\left\{R \to -\frac{K\,\alpha\,\delta + K\,r\,x\,\beta[x] - K\,r\,x\,\gamma\,\beta[x]}{-K\,\alpha^2\,\epsilon + K\,x\,\alpha^2\,\epsilon - r\,x\,\beta[x] + r\,x\,\gamma\,\beta[x]},\ n \to -\frac{r\,\delta - K\,r\,\alpha\,\epsilon + K\,r\,x\,\alpha\,\epsilon}{(-1 + x)\left(-K\,\alpha^2\,\epsilon + K\,x\,\alpha^2\,\epsilon - r\,x\,\beta[x] + r\,x\,\gamma\,\beta[x]\right)}\right\}\right\}$$

```
(*copy and paste from above*)
```

$$R[x\_] := -\frac{K\,(\alpha\,\delta - r\,x\,(-1 + \gamma)\,\beta[x])}{K\,(-1 + x)\,\alpha^2\,\epsilon + r\,x\,(-1 + \gamma)\,\beta[x]};$$

$$n[x\_] := -\frac{r\,(\delta + K\,(-1 + x)\,\alpha\,\epsilon)}{(-1 + x)\left(K\,(-1 + x)\,\alpha^2\,\epsilon + r\,x\,(-1 + \gamma)\,\beta[x]\right)};$$

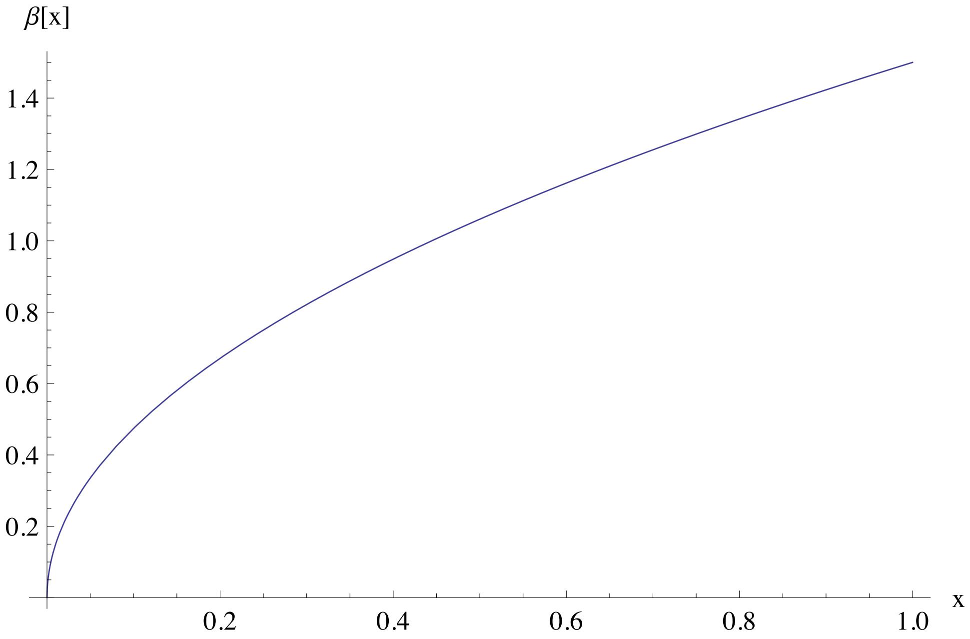
Invasion fitness and its derivatives:

```
s_x_[y_] := ϵ α (1 - y) R[x] - δ + γ β[y] y (1 - x) n[x] - (1 - y) β[x] x n[x];

ds[x_] := (∂_y s_x[y]) /. {y → x};

dds[x_] := (∂_y ∂_y s_x[y]) /. {y → x};
```

Default parameter values and functions:

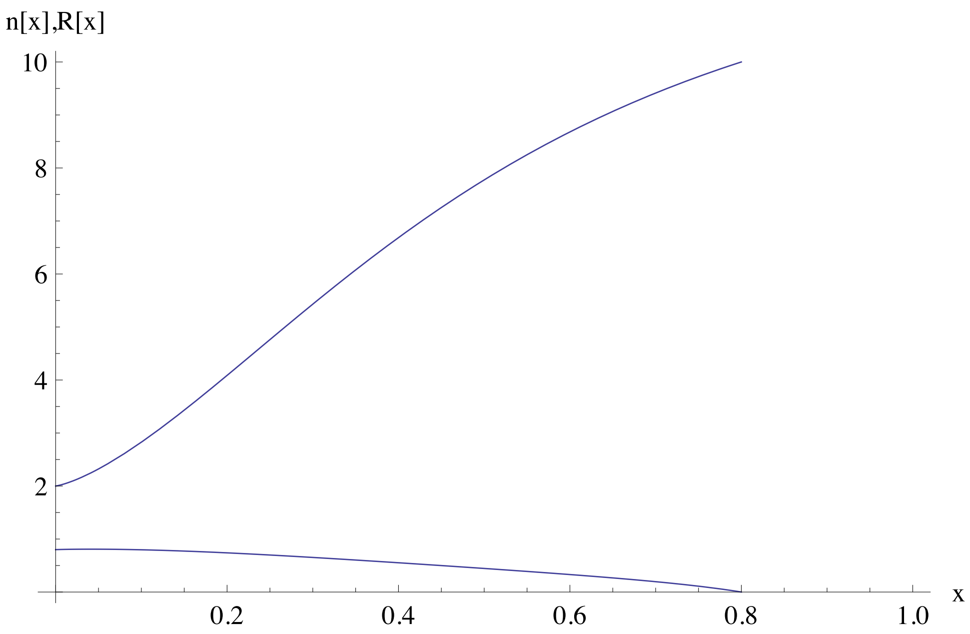
```
α = 1; γ = 0.2; δ = 0.1; ϵ = 0.05; r = 1; K = 10;

β0 = 0.; β1 = 1.5 ; p = .5;

β[x_] := β0 + β1 x^p;
```

```
Plot[β[x], {x, 0, 1}, AxesLabel → {"x", "β[x]"}, ImageSize → Medium]
```



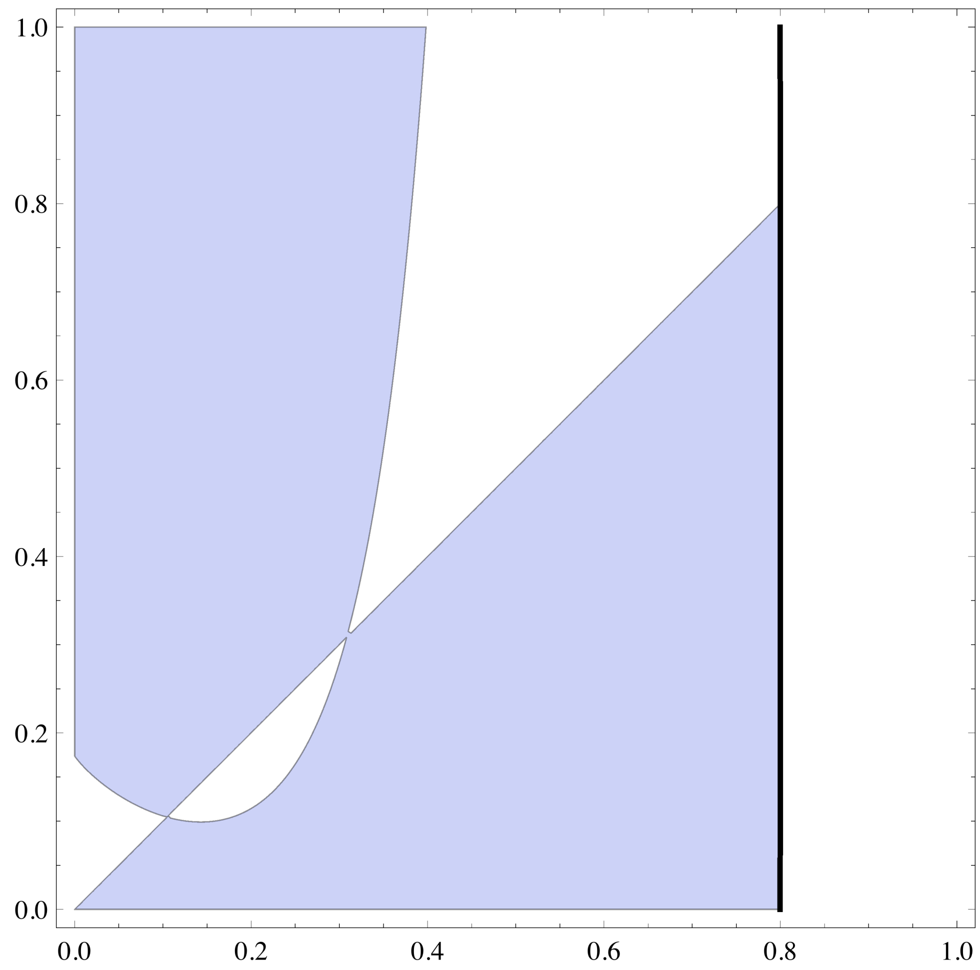
■ Plots of R[x] and n[x]:

```
Plot[If[n[x] ≥ 0, {R[x], n[x]}], {x, 0, 1}, AxesLabel → {"x", "n[x],R[x]"}, ImageSize → Medium]
```



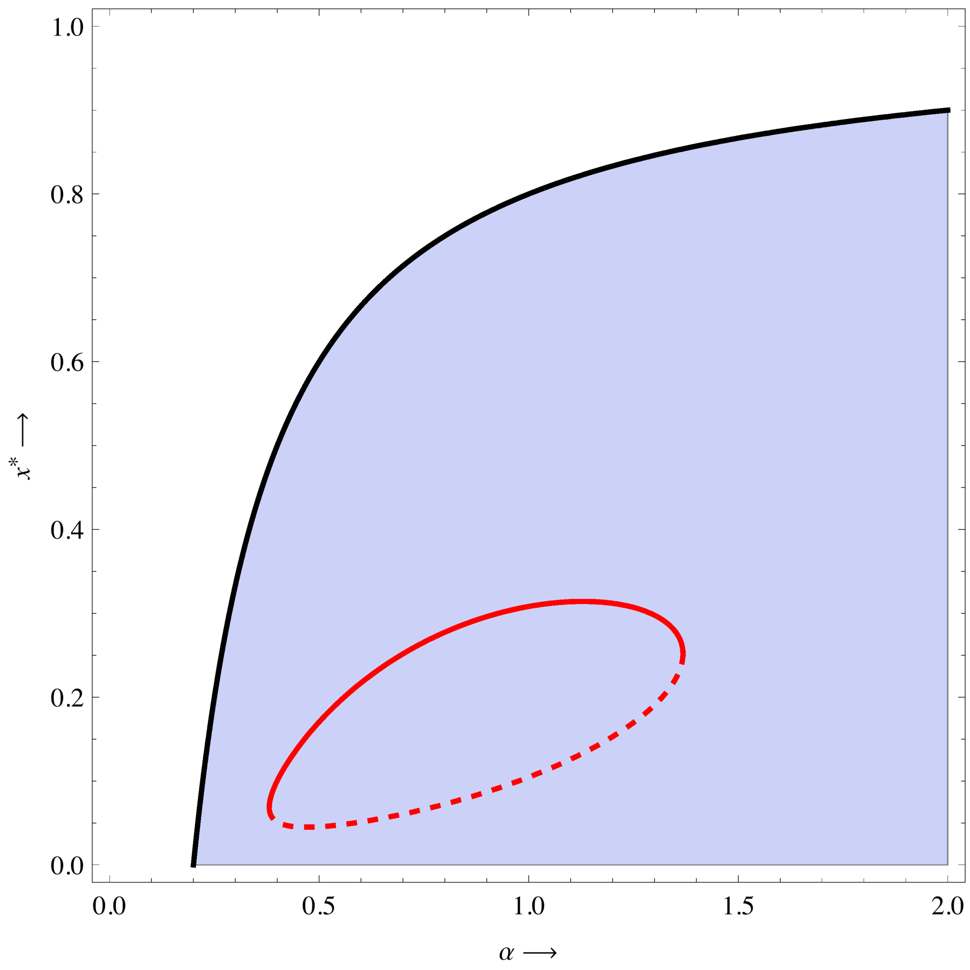
Pairwise invadability plot (PIP):

```
PIPint = RegionPlot[n[x] ≥ 0 && sx[y] > 0, {x, 0, 1}, {y, 0, 1}, PlotPoints → 100];

nPos = ContourPlot[n[x], {x, 0, 1}, {y, 0, 1}, Contours → {0}, ContourStyle → {Black, Thick}, Conto

Show[PIPint, nPos, ImageSize → Medium]
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

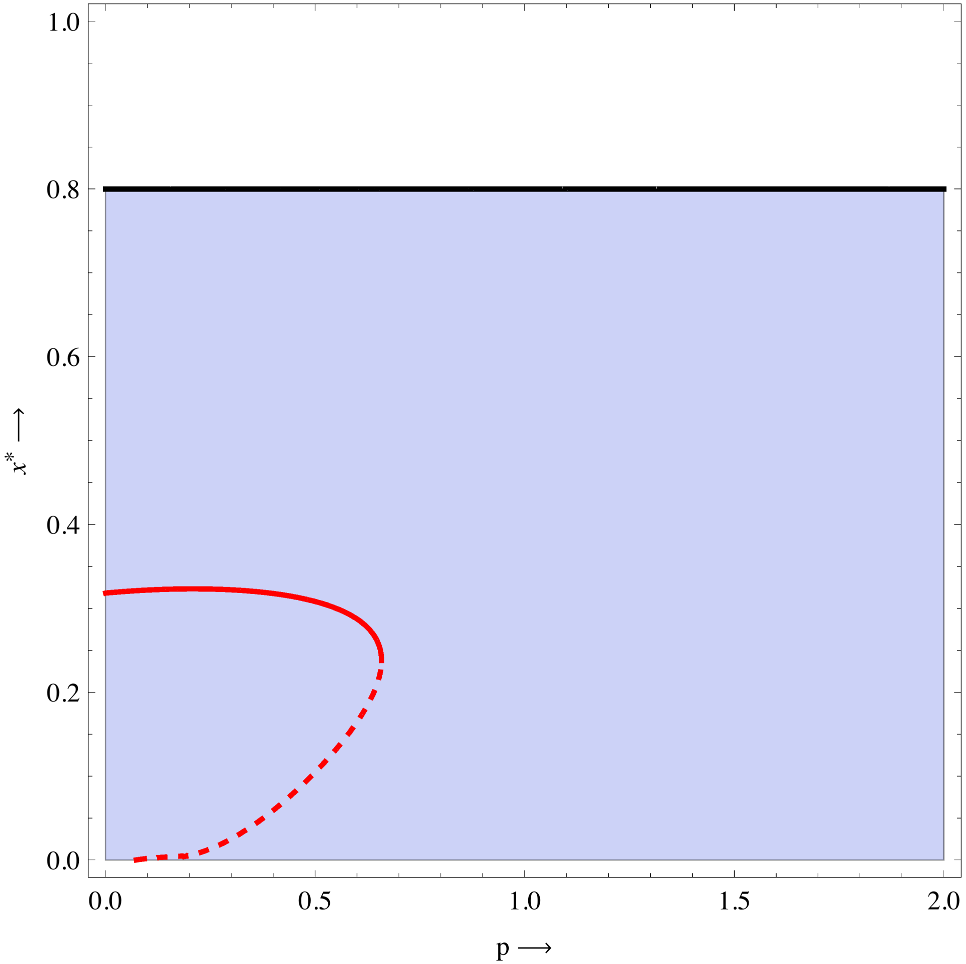
Bifurcation plot singular strategy x versus parameter α:

```
posEQ = RegionPlot[n[x] ≥ 0, {α, 0, 2}, {x, 0, 1}];

attESS = ContourPlot[If[n[x] > 0 && ds′[x] < 0 && dds[x] < 0, ds[x]], {α, 0, 2}, {x, 0, 1}, Contours →
    PlotPoints → 50];

repESS = ContourPlot[If[n[x] > 0 && ds′[x] > 0 && dds[x] < 0, ds[x]], {α, 0, 2}, {x, 0, 1}, Contours →
    ContourShading → False, PlotPoints → 50];

attNESS = ContourPlot[If[n[x] > 0 && ds′[x] < 0 && dds[x] > 0, ds[x]], {α, 0, 2}, {x, 0, 1}, Contours
    PlotPoints → 50];

repNESS = ContourPlot[If[n[x] > 0 && ds′[x] > 0 && dds[x] > 0, ds[x]], {α, 0, 2}, {x, 0, 1}, Contours
    ContourShading → False, PlotPoints → 50];

posEQbnd = ContourPlot[n[x], {α, 0, 2}, {x, 0, 1}, Contours → {0}, ContourStyle → {Black, Thick}, (

Show[posEQ, posEQbnd, attESS, repESS, attNESS, repNESS, FrameLabel → {"α ⟶", "x* ⟶"}, ImageSiz
```


1.0
0.8
0.6
0.4
0.2
0.0
$x^* \longrightarrow$
0.0
0.5
1.0
1.5
2.0
$\alpha \longrightarrow$

▪ Bifurcation plot singular strategy x versus parameter p:

```
posEQ = RegionPlot[n[x] ≥ 0, {p, 0, 2}, {x, 0, 1}];

attESS = ContourPlot[If[n[x] > 0 && ds′[x] < 0 && dds[x] < 0, ds[x]], {p, 0, 2}, {x, 0, 1}, Contours →
    PlotPoints → 50];

repESS = ContourPlot[If[n[x] > 0 && ds′[x] > 0 && dds[x] < 0, ds[x]], {p, 0, 2}, {x, 0, 1}, Contours →
    ContourShading → False, PlotPoints → 50];

attNESS = ContourPlot[If[n[x] > 0 && ds′[x] < 0 && dds[x] > 0, ds[x]], {p, 0, 2}, {x, 0, 1}, Contours
    PlotPoints → 50];

repNESS = ContourPlot[If[n[x] > 0 && ds′[x] > 0 && dds[x] > 0, ds[x]], {p, 0, 2}, {x, 0, 1}, Contours
    ContourShading → False, PlotPoints → 50];

posEQbnd = ContourPlot[n[x], {p, 0, 2}, {x, 0, 1}, Contours → {0}, ContourStyle → {Black, Thick}, (

Show[posEQ, posEQbnd, attESS, repESS, attNESS, repNESS, FrameLabel → {"p ⟶", "x* ⟶"}, ImageSiz
```


1.0
0.8
0.6
0.4
0.2
0.0
$x^* \longrightarrow$
0.0
0.5
1.0
1.5
2.0
p ⟶

DIMORPHIC RESIDENT POPULATION

Reset:

```
Clear[α, β, γ, δ, ϵ, r, K];
```

Dimorphic resident population equilibrium:

```
eqn =
  {0 == r - r R / K - α ((1 - x1) n1 + (1 - x2) n2), 0 == ϵ α (1 - x1) R - δ + γ β[x1] x1 ((1 - x1) n1 + (1 - x2) n2
   0 == ϵ α (1 - x2) R - δ + γ β[x2] x2 ((1 - x1) n1 + (1 - x2) n2) - (1 - x2) (β[x1] x1 n1 + β[x2] x2 n2)};
var = {R, n1, n2};
Solve[eqn, var];
Simplify[%]
```

```
{{R → K ((x1 - x2) α δ + r x1 (-1 + x2) γ β[x1] - r (-1 + x1) x2 γ β[x2]) / (r γ (x1 (-1 + x2) β[x1] - (-1 + x1) x2 β[x2])),
  n1 → (K (x1 - x2) (-1 + x2) α^2 δ ϵ + r x1 (-1 + x2) γ (δ + K (-1 + x2) α ϵ) β[x1] + r x2 (γ (δ - K α ϵ) + x2 (δ
    (r γ (x1 (-1 + x2) β[x1] - (-1 + x1) x2 β[x2])^2),
  n2 → (r x1 (-x2 δ + γ δ - K α γ ϵ + K x2 α γ ϵ + x1 (δ - γ δ - K (-1 + x2) α γ ϵ)) β[x1] - (-1 + x1) (K (x1 - x2
    (r γ (x1 (-1 + x2) β[x1] - (-1 + x1) x2 β[x2])^2)}}
```

```
R[x1_, x2_] :=
   (K ((x1 - x2) α δ + r x1 (-1 + x2) γ β[x1] - r (-1 + x1) x2 γ β[x2])) / (r γ (x1 (-1 + x2) β[x1] - (-1 + x1)

n1[x1_, x2_] := (K (x1 - x2) (-1 + x2) α^2 δ ϵ + r x1 (-1 + x2) γ (δ + K (-1 + x2) α ϵ) β[x1] + r x2 (γ (δ - K α
    (r γ (x1 (-1 + x2) β[x1] - (-1 + x1) x2 β[x2])^2);

n2[x1_, x2_] := (r x1 (-x2 δ + γ δ - K α γ ϵ + K x2 α γ ϵ + x1 (δ - γ δ - K (-1 + x2) α γ ϵ)) β[x1] - (-1 + x1) (K
    (r γ (x1 (-1 + x2) β[x1] - (-1 + x1) x2 β[x2])^2);
```

- **Dimorphic invasion fitness and derivatives:**

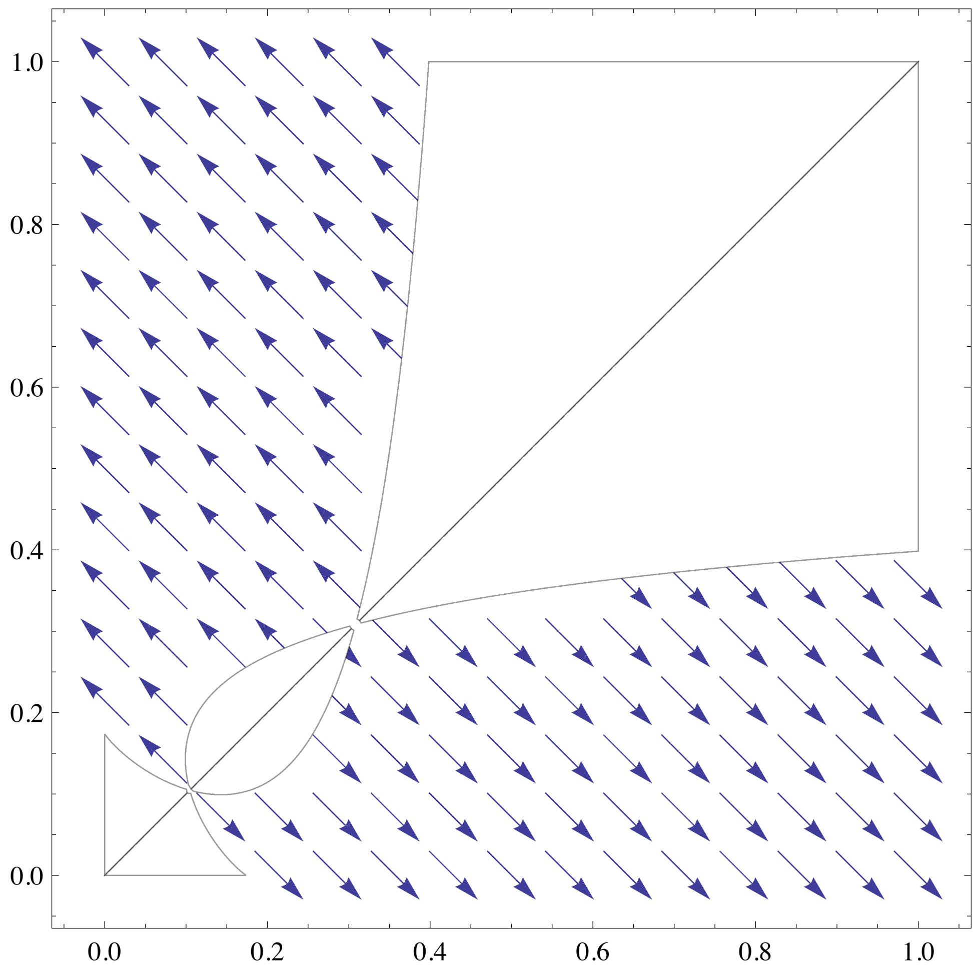
```
s_{x1_,x2_}[y_] := ϵ α (1 - y) R[x1, x2] - δ + γ β[y] y ((1 - x1) n1[x1, x2] + (1 - x2) n2[x1, x2]) - (1 - y) (β

x1ds[x1_, x2_] := ∂_y s_{x1,x2}[y] /. {y → x1};
x2ds[x1_, x2_] := ∂_y s_{x1,x2}[y] /. {y → x2};
x1dds[x1_, x2_] := ∂_y ∂_y s_{x1,x2}[y] /. {y → x1};
x2dds[x1_, x2_] := ∂_y ∂_y s_{x1,x2}[y] /. {y → x2};
```

- **Default parameter values and functions:**

```
α = 1; γ = 0.2; δ = 0.1; ϵ = 0.05; r = 1; K = 10;

β0 = 0.; β1 = 1.5 ; p = .5;

β[x_] := β0 + β1 x^p;
```

Coexistence plot (MIP):

```
MIP = RegionPlot[n1[x1, x2] > 0 ⋀ n2[x1, x2] > 0, {x1, 0, 1}, {x2, 0, 1}, PlotPoints → 100];

antiMIP = RegionPlot[n1[x1, x2] < 0 ⋁ n2[x1, x2] < 0, {x1, 0, 1}, {x2, 0, 1}, PlotPoints → 100, Plot

grad = VectorPlot[If[n1[x1, x2] > 0 && n2[x1, x2] > 0, {Sign[x1ds[x1, x2]], Sign[x2ds[x1, x2]]}, {0

Show[MIP, grad, antiMIP]
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0