Cannibalism time budget model

Resident population dynamics:

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

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

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

Mutant population dynamics:

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

$$+ y\, m \left(\gamma\, \beta[y] \sum_{j=1}^{k} \left(1 - x_j\right) n_j - \delta \right)$$

Invasion fitness:

$$s_{x_1,\ldots,x_k}(y) = (1 - y) \left(\epsilon\, \alpha \langle R \rangle - \sum_{j=1}^{k} \beta\left[x_j\right] x_j \left\langle n_j \right\rangle - \delta \right) +$$

$$+ y \left(\gamma\, \beta[y] \sum_{j=1}^{k} \left(1 - x_j\right) \left\langle n_j \right\rangle - \delta \right)$$

Three-dimensional environment:

$$E = \begin{pmatrix} \langle R \rangle \\ \sum_{j=1}^{k} \beta\left[x_j\right] x_j \left\langle n_j \right\rangle \\ \sum_{j=1}^{k} \left(1 - x_j\right) \left\langle n_j \right\rangle \end{pmatrix}$$

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 *)
```

$$dLogR = r - \frac{r\,R}{K} - \alpha\,(1 - x)\,n;$$

$$dLogn = \epsilon\,\alpha\,(1 - x)\,R - \delta + \gamma\,\beta[x]\,x\,(1 - x)\,n - (1 - x)\,\beta[x]\,x\,n;$$

- **Monomorphic resident population equilibrium:**

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

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

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

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

- **Invasion fitness and its derivatives:**

$$s_{x\_}[y\_] = \epsilon\,\alpha\,(1 - y)\,R[x] - \delta + \gamma\,\beta[y]\,y\,(1 - x)\,n[x] - (1 - y)\,\beta[x]\,x\,n[x]\ //\ Simplify;$$

$$ds[x\_] = \partial_y s_x[y]\ /.\ \{y \to x\}\ //\ Simplify;$$

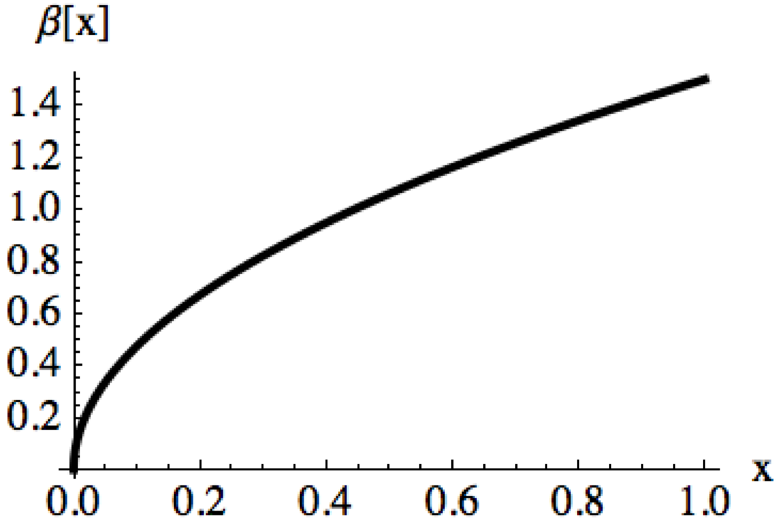
$$dds[x\_] = \partial_y \partial_y s_x[y]\ /.\ \{y \to x\}\ //\ Simplify;$$

- **Default parameter values and functions (set 1):**

```
α = 1; γ = 0.2; δ = 0.1; ϵ = 0.05; r = 1; K = 10;
β0 = 0.; β1 = 1.5; p = .5;
```

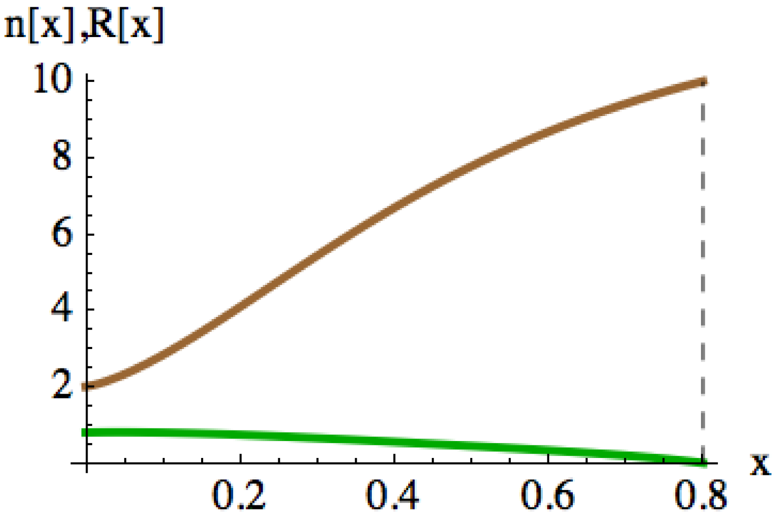
$$\beta[x\_] := \beta 0 + \beta 1\,x^p;$$

```
Plot[β[x], {x, 0, 1}, AxesLabel → {"x", "β[x]"}, PlotStyle → {Black, Thick}, ImageSize → Small]
```



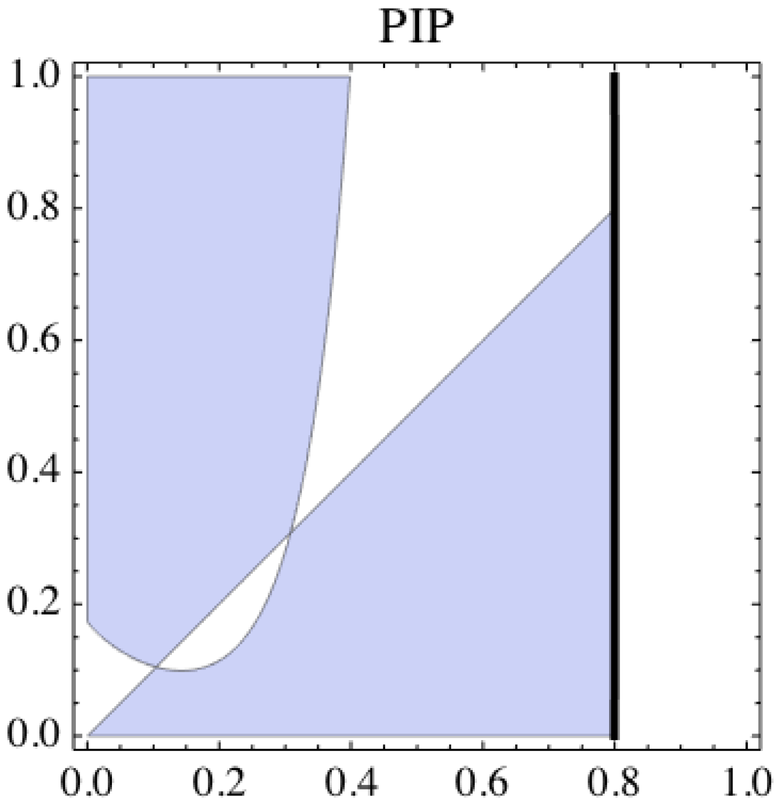
- **Plots of R[x] and n[x]:**

```
Show[
 Plot[If[n[x] ≥ 0, R[x]], {x, 0, 1}, PlotStyle → {Brown, Thick}],
 Plot[If[n[x] ≥ 0, n[x]], {x, 0, 1}, PlotStyle → {Darker[Green], Thick}],
 Graphics[{Dashed, Line[{{.8, 0}, {.8, 10}}]}],
 PlotRange → All, AxesOrigin → {0, 0}, AxesLabel → {"x", "n[x],R[x]"}, ImageSize → Small]
```



- **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}, ContourShading → False, PlotPoints →
Show[PIPint, nPos, ImageSize → Small, PlotLabel → "PIP"]
```



- **Singular strategies:**

```
xSing1 = x /. FindRoot[ds[x] == 0, {x, .1}]
xSing2 = x /. FindRoot[ds[x] == 0, {x, .3}]
```

```
0.104529
```

```
0.308179
```

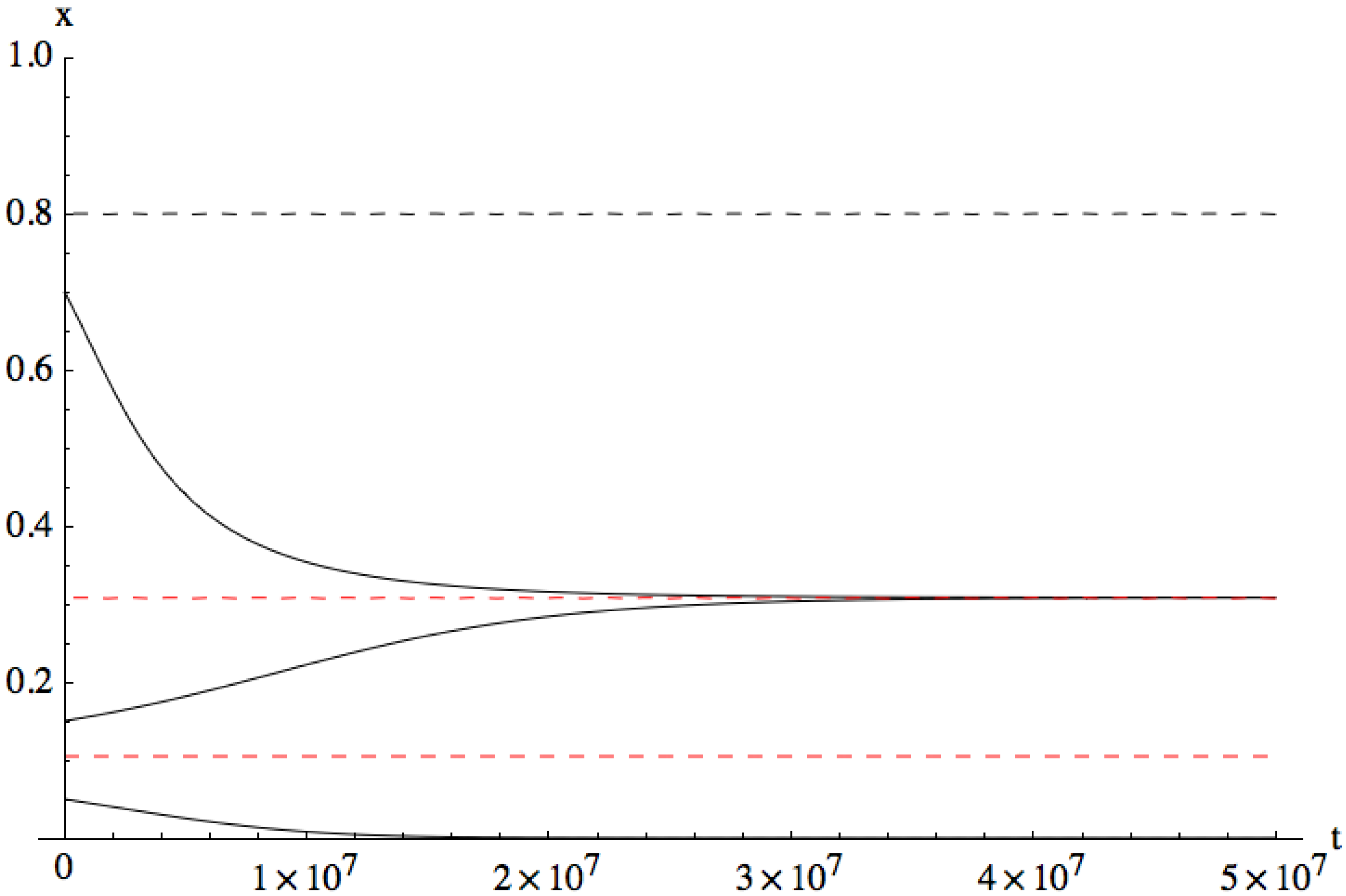
- **Canonical equation:**

- **Mutation probability, mutation variance and deterministic drift:**

```
μ[x_] := .001;
σ2[x_] := .01 x (1 - x);
drift[x_] := 1/2 μ[x] σ2[x] n[x] ds[x];
```

- **Deterministic orbit (Euler method):**

```
x0 = {0.05, 0.15, 0.7};(* different starting points *)
t0 = 0;
t∞ = 5 × 10^7;
Δt = 10^4;
noofStarts = Length[x0];
For[i = 1, i ≤ noofStarts, i++,
  data[i] = {};
  x = x0[[i]];
  t = t0;
  While[t ≤ t∞ && n[x] > 0,
   data[i] = Join[data[i], {{t, x}}];
   x = x + Δt drift[x];
   t = t + Δt];
 ];

CEorbit =
 Show[
  Table[ListPlot[data[i], PlotStyle → {Black}, Joined → True], {i, 1, noofStarts}],
  ContourPlot[ds[x], {t, 0, t∞}, {x, 0, 1}, Contours → {0}, ContourShading → False, ContourStyle → {Red, Dashed}],
  ContourPlot[n[x], {t, 0, t∞}, {x, 0, 1}, Contours → {0}, ContourShading → False, ContourStyle → {Black, Dashed}],
  AxesOrigin → {0, 0}, PlotRange → {0, 1}, AxesLabel → {"t", "x"}
 ]
```

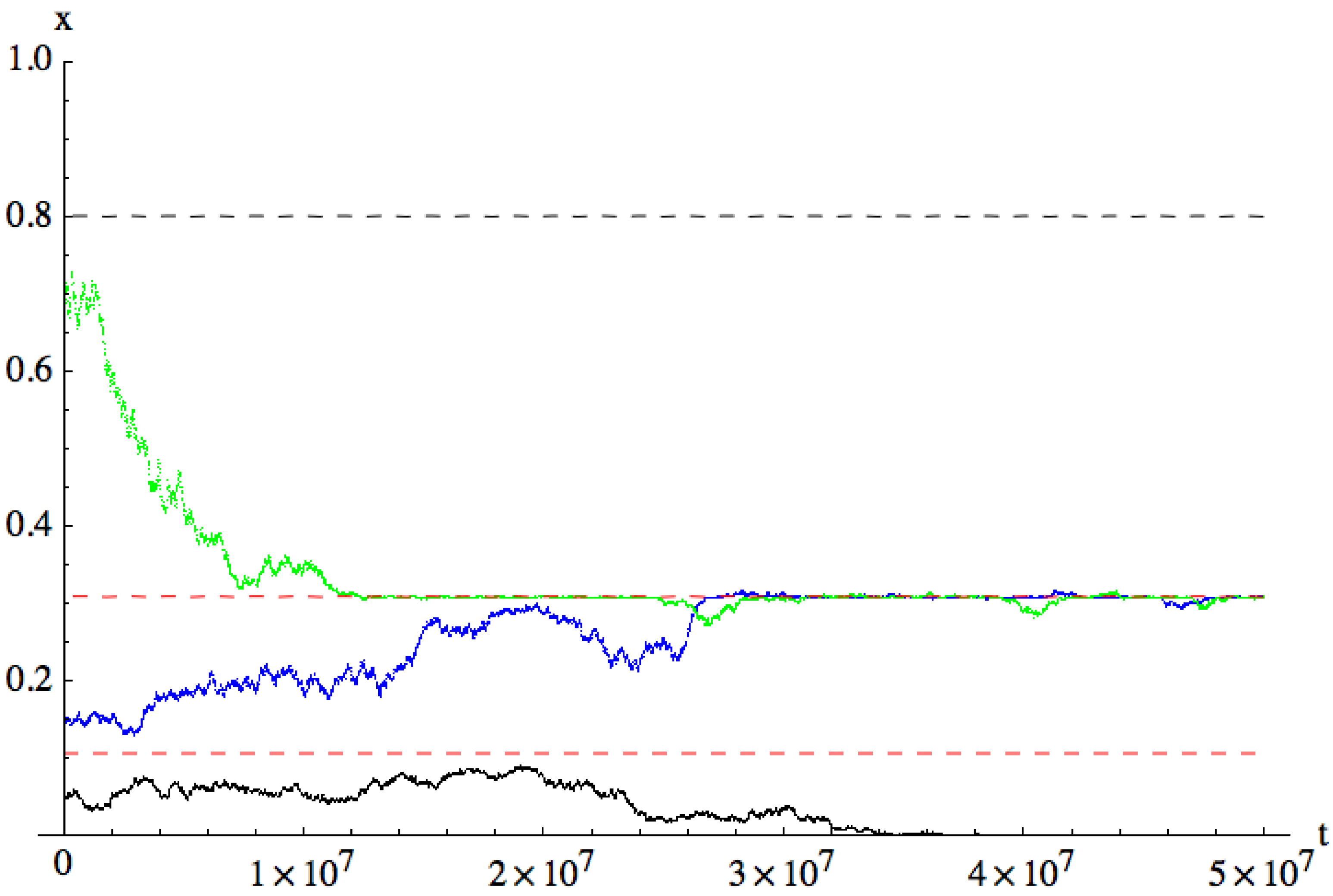

x
1.0
0.8
0.6
0.4
0.2
0
1×10^7
2×10^7
3×10^7
4×10^7
5×10^7
t

- **Stochastic differential equation (Ito):**
- **Third absolute moment of the mutation step distribution and diffusion coefficient:**

```
θ[x_] := 2 σ2[x]^(3/2) Sqrt[2/π]; (* assuming Gausian *)

diff[x_] := 1/2 μ[x] θ[x] n[x] Abs[ds[x]];
```

- **Single stochastic orbit per starting point (Euler metod):**

```
x0 = {0.05, 0.15, 0.7}; (* different starting points *)
t0 = 0;
t∞ = 5 × 10^7;
Δt = 10^4;
noofStarts = Length[x0];
For[i = 1, i ≤ noofStarts, i++,
  data[i] = {};
  x = x0[[i]];
  t = t0;
  While[t ≤ t∞ && n[x] > 0,
   data[i] = Join[data[i], {{t, x}}];
   z = RandomReal[NormalDistribution[0, 1]]; x = x + Δt drift[x] + z Sqrt[Δt diff[x]];
   t = t + Δt];
 ];

Colour = {Black, Blue, Green};
SDEorbit =
 Show[
  Table[ListPlot[data[i], PlotStyle → {Colour[[i]], PointSize[.001]}, Joined → False], {i, 1, noofStarts}],
  ContourPlot[ds[x], {t, 0, t∞}, {x, 0, 1}, Contours → {0}, ContourShading → False, ContourStyle → {Red, Dashed}],
  ContourPlot[n[x], {t, 0, t∞}, {x, 0, 1}, Contours → {0}, ContourShading → False, ContourStyle → {Black, Dashed}],
  AxesOrigin → {0, 0}, PlotRange → {0, 1}, AxesLabel → {"t", "x"}
 ]
```



- **Larger samples:**

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) - (1 - x1) (β[x1] x1 n1 + β[x2] x:
   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 - γ δ + K α γ ϵ) - x1 (δ + K (-1 + x2) (
     (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) α^2 δ ϵ - r x2 γ (δ + K (-1 + x1) α ϵ) /
     (r γ (x1 (-1 + x2) β[x1] - (-1 + x1) x2 β[x2])^2)}}
```

```
(* Note: stability of equilibrium does not matter! Why? *)
```

```
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) x2 β[x2]));
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 ϵ) + x2 (δ - γ δ + K α γ ϵ) - x1 (δ + K
    (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 (x1 - x2) α^2 δ ϵ - r x2 γ (δ + K (-1 +
    (r γ (x1 (-1 + x2) β[x1] - (-1 + x1) x2 β[x2])^2);
```

- **Dimorphic invasion fitness and derivatives:**

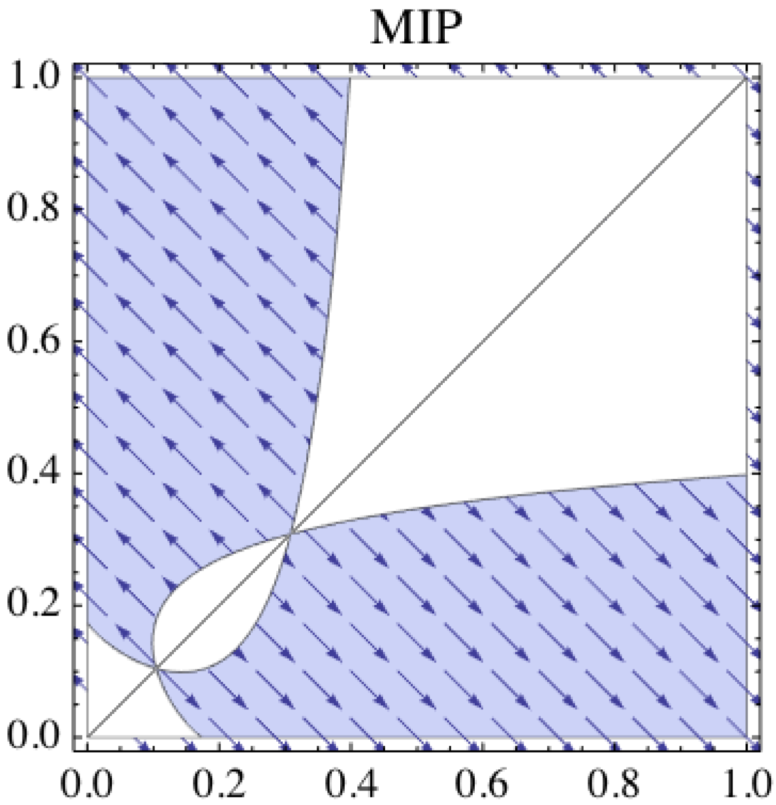
```
s_{x1_,x2_}[y_] := ϵ α (1 - y) R[x1, x2] - δ + γ β[y] y ((1 - x1) n1[x1, x2] + (1 - x2) n2[x1, x2]) - (1 - y) (β[x1] x1 n1[x1, x2] + β[x2] x2 n2[x1

x1ds[x1_, x2_] := ∂_y s_{x1,x2}[y] /. {y → x1};
x2ds[x1_, x2_] := ∂_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] // Quiet;
antiMIP = RegionPlot[n1[x1, x2] ≤ 0 ∨ n2[x1, x2] ≤ 0, {x1, 0, 1}, {x2, 0, 1}, PlotStyle → White, PlotPoints → 100] // Quiet;
gradVec = VectorPlot[{Sign[x1ds[x1, x2]], Sign[x2ds[x1, x2]]}, {x1, 0, 1}, {x2, 0, 1}, VectorScale → Small] // Quiet;

Show[MIP, gradVec, antiMIP, ImageSize → Small, PlotLabel → "MIP"]
```



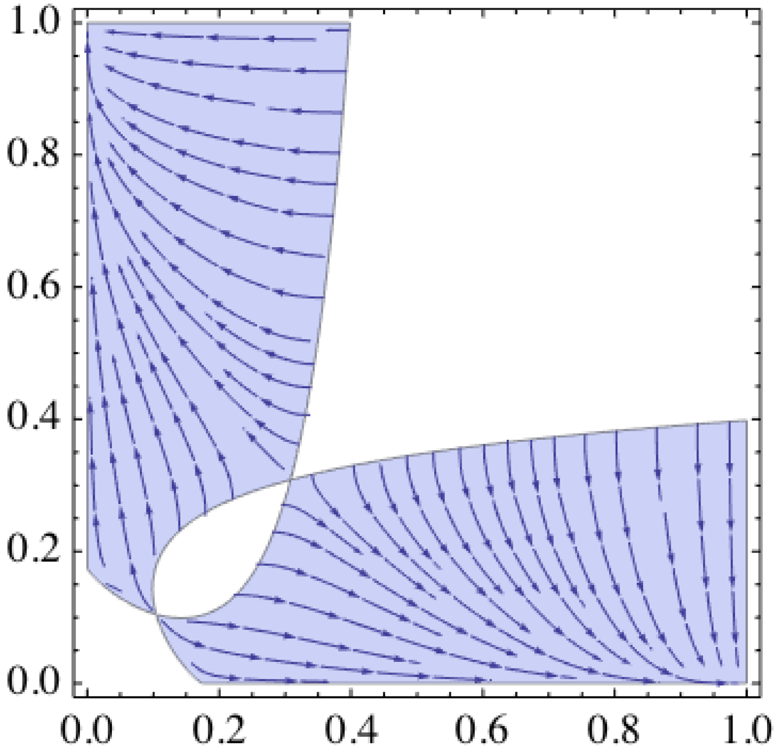
- **Canonical equation:**

- **Deterministic drift:**

```
μ[x_] := .001;
σ2[x_] := .001 x (1 - x);
drift[x1_, x2_] := {1/2 μ[x1] σ2[x1] n1[x1, x2] x1ds[x1, x2], 1/2 μ[x2] σ2[x2] n2[x1, x2] x2ds[x1, x2]};
```

- **Stream plot:**

```
CEstream = StreamPlot[If[n1[x1, x2] > 0 ∧ n2[x1, x2] > 0, drift[x1, x2], {0, 0}], {x1, 0, 1}, {x2, 0, 1}] // Quiet;

Show[MIP, CEstream, ImageSize → Small]

(* Note: CE-orbit either horizontal or vertical at invasion boundary!
   Why? *)
```



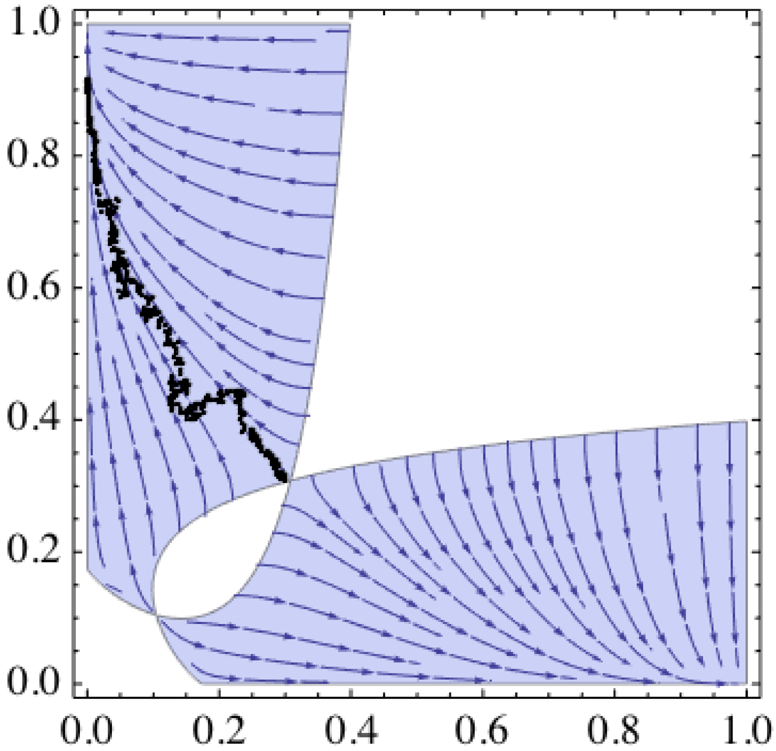
- **Stochastic orbits:**

- **Diffusion coefficient:**

```
θ[x_] := 2 σ2[x]^(3/2) Sqrt[2/π];(* assuming Gausian *)

diff[x1_, x2_] := {1/2 μ[x1] θ[x1] n1[x1, x2] Abs[x1ds[x1, x2]], 1/2 μ[x2] θ[x2] n2[x1, x2] Abs[x2ds[x1, x2]]};
```

- **Single stochastic orbit (Euler metod):**

```
t∞ = 10^10;
Δt = 10^6;
x1 = xSing2 - .01;
x2 = xSing2 + .01;
data = {{x1, x2}};
t = 0;
While[t ≤ t∞ ∧ n1[x1, x2] > 0 ∧ n2[x1, x2] > 0,
  z = RandomReal[NormalDistribution[0, 1], 2]; {x1, x2} = {x1, x2} + Δt drift[x1, x2] + z Sqrt[Δt diff[x1, x2]];
  data = Join[data, {{x1, x2}}];
  t = t + Δt;
 ];

SDEorbit =
  ListPlot[data, PlotStyle → {Black, PointSize[.0075]}, Joined → False];

Show[MIP, CEstream, SDEorbit, ImageSize → Small]
```


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