Competition colonization trade – off

General model

Population equation

$$\frac{dp_i}{dt} = \beta_i\, p_i \left(1 - \sum_{\forall j} (1 - \gamma_{j,i})\, p_j\right) - p_i \left(\sum_{\forall j} \gamma_{i,j}\, \beta_j\, p_j\right) - \delta_i\, p_i;$$

⟵−−−−−−−−−−−⟶ per offspring probability of colonization

⟵−−−−−−⟶ rate of displacement from a site

with

p_i = proportion of sites occupied by ' species ' i

and

β_i = per capita birth rate for type i

δ_i = per capita death rate for type i

$\gamma_{i,j}$ = probability of site owner of type i
being displaced by intruder of type j

Explanation

The probability that an offspring of type i gets lost because of loosing competition with a site owner of type j is $(1 - \gamma_{j,i})\, p_j$, i.e., the probability p_j of landing in a site occupied by type j multiplied by the probability $1 - \gamma_{j,i}$ of not being able to displace the site owner;

Hence, the probability of being lost because of failing to displace any site owner is

$$\sum_{\forall j} (1 - \gamma_{j,i})\, p_j,$$

and hence the probability of finding a site, either because it does not have an owner, or because the owner is displaced is

$1 - \sum_{\forall j} (1 - \gamma_{j,i})\, p_j$;

The probability of a type i owner to be displaced by any type j intruder is

$\sum_{\forall j} \gamma_{i,j}\, \beta_j\, p_j$;

The term $\delta_i\, p_i$ describes additional death unrelated to site competition;

Rewrite the population equation as

$$\frac{dp_i}{dt} = r_i\, p_i \left(1 - \frac{\sum_{\forall j} a_{i,j}\, p_j}{K_i}\right)$$

with

$r_i = \beta_i - \delta_i$

$$K_i = 1 - \frac{\delta_i}{\beta_i}$$

$$a_{i,j} = 1 - \gamma_{j,i} + \frac{\beta_j}{\beta_i}\, \gamma_{i,j}$$

Strategy

x_i = body size

Hence we write

$\beta_i = \beta[x_i]$

$\delta_i = \delta[x_i]$

$\gamma_{i,j} = \gamma[x_i, x_j]$

Invasion fitness (monomorphic)

$$s_x[y] = r[y] \left(1 - \frac{a[y, x]\, K[x]}{K[y]}\right)$$

Invasion fitness (dimorphic)

$$s_{x_1, x_2}[y] = r[y] \left(1 - \frac{a[y, x_1]\, n_1[x_1, x_2] + a[y, x_2]\, n_2[x_1, x_2]}{K[y]}\right)$$

with

$$n_1[x_1, x_2] = \frac{K[x_1] - a[x_1, x_2]\, K[x_2]}{1 - a[x_1, x_2]\, a[x_2, x_1]}$$

$$n_2[x_1, x_2] = \frac{K[x_2] - a[x_2, x_1]\, K[x_1]}{1 - a[x_1, x_2]\, a[x_2, x_1]}$$

etc.

Implementation

Ecology

```
r[x_] = β[x] - δ[x];
K[x_] = 1 - δ[x]/β[x];
a[x_, y_] = 1 - γ[y, x] + γ[x, y] β[y]/β[x];
```

Invasion fitness, gradient, curvature, drift (*monomorphic*)

```
s1[x_, y_] = r[y] (1 - a[y, x] n11[x]/K[y]);
n11[x_] = K[x];
grad11[x_] = s1^(0,1)[x, x];
xCurv11[x_] = s1^(2,0)[x, x];
yCurv11[x_] = s1^(0,2)[x, x];
driftMonomorph[x_] = .5 μ σ2[x] n11[x] grad11[x];
diffMonomorph[x_] = .5 μ θ3[x] n11[x] Abs[grad11[x]];
```

Invasion fitness, gradient, curvature, drift (*dimorphic*)

```
s2[x1_, x2_, y_] = r[y] (1 - (a[y, x1] n21[x1, x2] + a[y, x2] n22[x1, x2])/K[y]);
n21[x1_, x2_] = (K[x1] - a[x1, x2] K[x2])/(1 - a[x1, x2] a[x2, x1]);
n22[x1_, x2_] = (K[x2] - a[x2, x1] K[x1])/(1 - a[x1, x2] a[x2, x1]);
grad21[x1_, x2_] = s2^(0,0,1)[x1, x2, x1];
grad22[x1_, x2_] = s2^(0,0,1)[x1, x2, x2];
curv21[x1_, x2_] = s2^(0,0,2)[x1, x2, x1];
curv22[x1_, x2_] = s2^(0,0,2)[x1, x2, x2];
driftDimorph[{x1_, x2_}] =
  {.5 μ σ2[x1] n21[x1, x2] grad21[x1, x2], .5 μ σ2[x2] n22[x1, x2] grad22[x1, x2]};
diffDimorph[{x1_, x2_}] =
  {.5 μ θ3[x1] n21[x1, x2] Abs[grad21[x1, x2]], .5 μ θ3[x2] n22[x1, x2] Abs[grad22[x1, x2]]};
```

Invasion fitness, gradient, curvature, drift (*trimorphic*)

```
s3[x1_, x2_, x3_, y_] =
  r[y] (1 - (a[y, x1] n31[x1, x2, x3] + a[y, x2] n32[x1, x2, x3] + a[y, x3] n33[x1, x2, x3]) / K[y]);
n31[x1_, x2_, x3_] = ((a[x1, x3] a[x3, x2] - a[x1, x2] a[x3, x3]) K[x2] + a[x2, x3]
       (-a[x3, x2] K[x1] + a[x1, x2] K[x3]) + a[x2, x2] (a[x3, x3] K[x1] - a[x1, x3] K[x3])) /
    (a[x1, x3] (-a[x2, x2] a[x3, x1] + a[x2, x1] a[x3, x2]) +
      a[x1, x2] (a[x2, x3] a[x3, x1] - a[x2, x1] a[x3, x3]) +
      a[x1, x1] (-a[x2, x3] a[x3, x2] + a[x2, x2] a[x3, x3]));
n32[x1_, x2_, x3_] = ((-a[x1, x3] a[x3, x1] + a[x1, x1] a[x3, x3]) K[x2] + a[x2, x3]
       (a[x3, x1] K[x1] - a[x1, x1] K[x3]) + a[x2, x1] (-a[x3, x3] K[x1] + a[x1, x3] K[x3])) /
    (a[x1, x3] (-a[x2, x2] a[x3, x1] + a[x2, x1] a[x3, x2]) +
      a[x1, x2] (a[x2, x3] a[x3, x1] - a[x2, x1] a[x3, x3]) +
      a[x1, x1] (-a[x2, x3] a[x3, x2] + a[x2, x2] a[x3, x3]));
n33[x1_, x2_, x3_] = ((a[x1, x2] a[x3, x1] - a[x1, x1] a[x3, x2]) K[x2] + a[x2, x2]
       (-a[x3, x1] K[x1] + a[x1, x1] K[x3]) + a[x2, x1] (a[x3, x2] K[x1] - a[x1, x2] K[x3])) /
    (a[x1, x3] (-a[x2, x2] a[x3, x1] + a[x2, x1] a[x3, x2]) +
      a[x1, x2] (a[x2, x3] a[x3, x1] - a[x2, x1] a[x3, x3]) +
      a[x1, x1] (-a[x2, x3] a[x3, x2] + a[x2, x2] a[x3, x3]));
grad31[x1_, x2_, x3_] = s3^(0,0,0,1)[x1, x2, x3, x1];
grad32[x1_, x2_, x3_] = s3^(0,0,0,1)[x1, x2, x3, x2];
grad33[x1_, x2_, x3_] = s3^(0,0,0,1)[x1, x2, x3, x3];
driftTrimorph[{x1_, x2_, x3_}] =
  {.5 μ σ2[x1] n31[x1, x2, x3] grad31[x1, x2, x3], .5 μ σ2[x2] n32[x1, x2, x3]
    grad32[x1, x2, x3], .5 μ σ2[x3] n33[x1, x2, x3] grad33[x1, x2, x3]};
diffTrimorph[{x1_, x2_, x3_}] =
  {.5 μ θ3[x1] n31[x1, x2, x3] Abs[grad31[x1, x2, x3]], .5 μ θ3[x2] n32[x1, x2, x3]
    Abs[grad32[x1, x2, x3]], .5 μ θ3[x3] n33[x1, x2, x3] Abs[grad33[x1, x2, x3]]};
```

Case: $\gamma[x_i, x_j] = 1 - \gamma[x_j, x_i]$ (i.e., no effect of ownership)

Functions and parameters

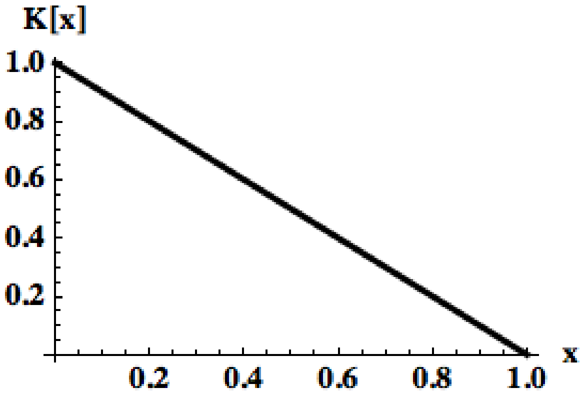
```
β[x_] = R/x;
R = 1;
δ[x_] = 1;
γ[x_, y_] = c[y] / (c[x] + c[y]);
c[x_] = e^(α x);

μ = 1;

σ2[x_] := .001 x (R - x);

θ3[x_] := 2 σ2[x]^(3/2) Sqrt[2/π]; (* assuming Gausian *)
```

Strategy range

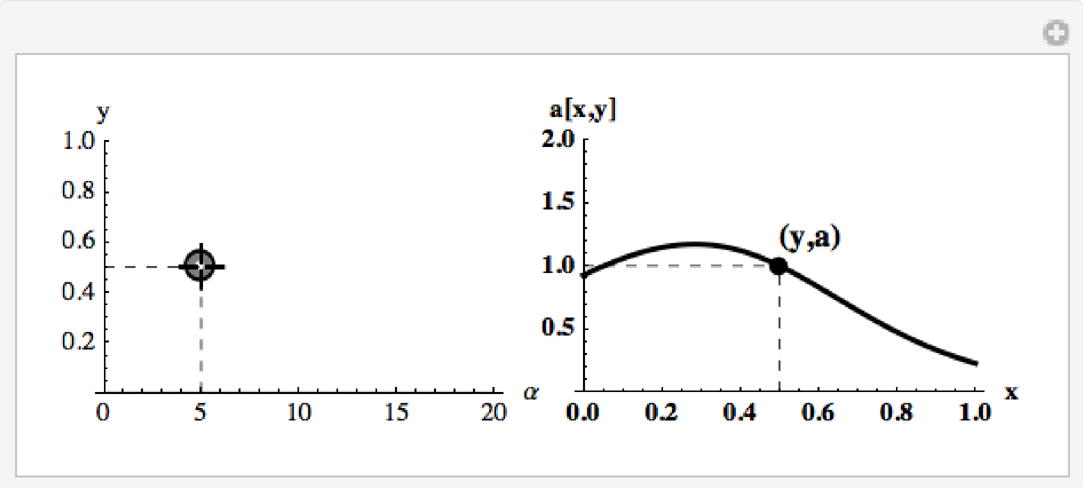
```
xMin = 0;
xMax = R;
```

- **Carrying capacity & competition kernel**

```
Plot[K[x], {x, xMin, xMax}, AxesOrigin → {0, 0}, PlotStyle → {Thick, Black},
 AxesLabel → {"x", "K[x]"}, LabelStyle → {Bold}, ImageSize → {Small}]
```



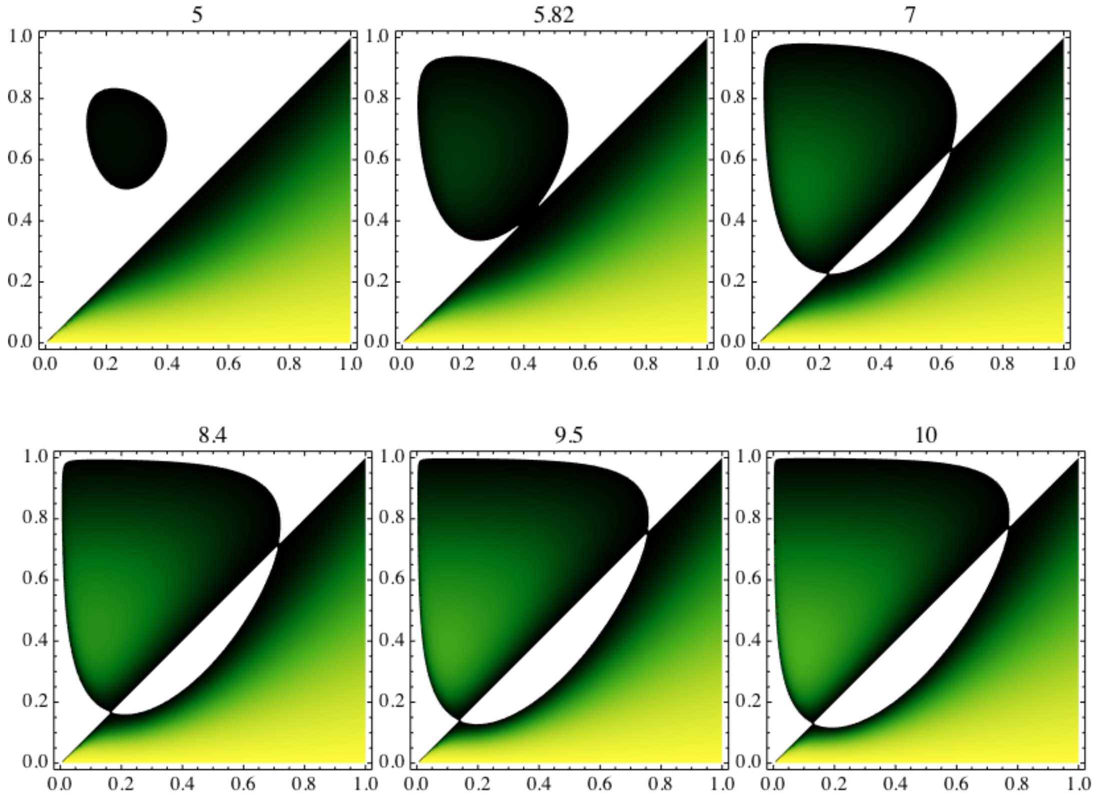

```
Manipulate[α = αy[[1]]; y = αy[[2]];
 Row[{Show[Plot[0, {α, 0, 20}, AxesOrigin → {0, xMin}, PlotRange → {xMin, xMax},
     AxesLabel → {"α", "y"}, Frame → False, ImageSize → Small],
    Graphics[Locator[αy], PlotRange → {{0, 20}, {xMin, xMax}}],
    Graphics[{Dashed, Line[{{{α, xMin}, {α, y}}, {{0, y}, {α, y}}}]}]],
   Show[Plot[a[x, y], {x, xMin, xMax}, AxesOrigin → {0, 0}, AxesLabel → {"x", "a[x,y]"},
     LabelStyle → {Bold}, PlotStyle → {Black, Thick}, PlotRange → {0, 2},
     ImageSize → Small], Graphics[{PointSize[Large], Point[{y, a[y, y]}],
      Text[Style["(y,a)", Medium, Bold], {y, 1.1}, {-1, -1}]}],
    Graphics[{Dashed, Line[{{{y, 0}, {y, 1}}, {{xMin, 1}, {y, 1}}}]}]]}],
 {{αy, {5, .5 (xMax - xMin)}}, Locator}]
```



Monomorphic evolution

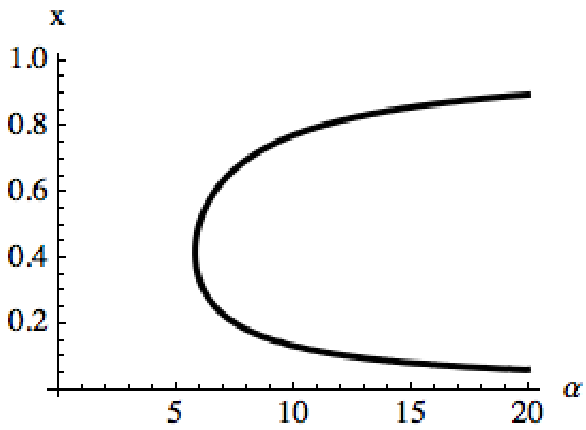
▪ Pairwise invadability plots PIP

```
Do[PIP[α] = DensityPlot[Block[{inv}, inv = s1[x, y]; If[0 < inv, ArcTan[inv]]],
    {x, xMin, xMax}, {y, xMin, xMax}, PlotPoints → 50,
    ColorFunction → "AvocadoColors"], {α, {5, 5.82, 7, 8.4, 9.5, 10}}];
Row[Table[Show[PIP[α], PlotLabel → α, ImageSize → Small], {α, {5, 5.82, 7, 8.4, 9.5, 10}}]]
```

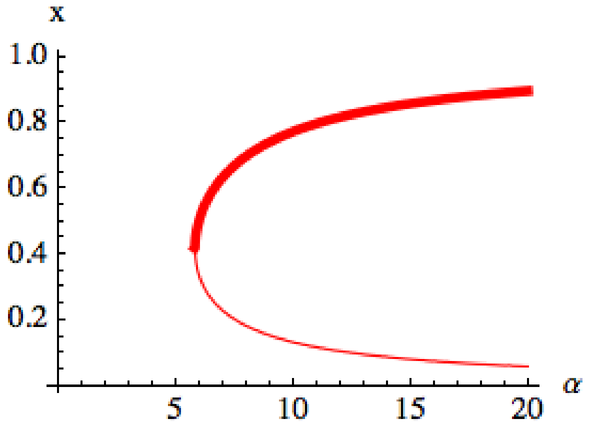


▪ Bifurcation Plot (x vs α)

```
ContourPlot[grad11[x], {α, 0, 20}, {x, xMin, xMax}, Contours → {0},
 ContourStyle → {Black, Thick}, ContourShading → False, PlotPoints → 20, Frame → False,
 Axes → True, AspectRatio → .7, AxesLabel → {"α", "x"}, ImageSize → Small]
```

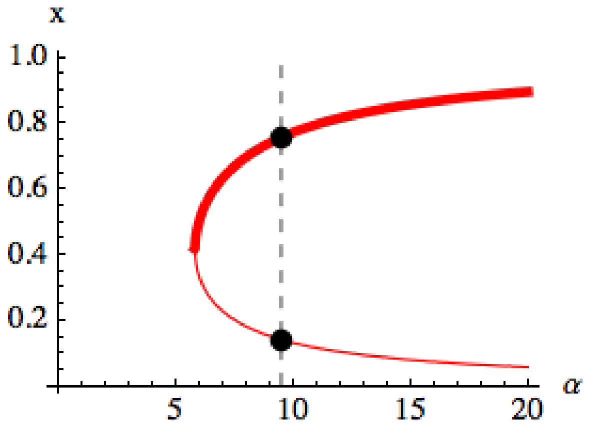


```
monoBifPlot =
 Quiet[Show[ContourPlot[If[yCurv11[x] < Min[xCurv11[x], 0], grad11[x]], {α, 0, 20},
    {x, xMin, xMax}, Contours → {0}, ContourStyle → {Black, Thickness[0.02]},
    ContourShading → False, PlotPoints → 20], ContourPlot[
    If[xCurv11[x] < yCurv11[x] < 0, grad11[x]], {α, 0, 20}, {x, xMin, xMax}, Contours → {0},
    ContourStyle → {Black, Thickness[.005]}, ContourShading → False, PlotPoints → 20],
   ContourPlot[If[Max[xCurv11[x], 0] < yCurv11[x], grad11[x]], {α, 0, 20}, {x, xMin, xMax},
    Contours → {0}, ContourStyle → {Red, Thickness[.005]}, ContourShading → False,
    PlotPoints → 20], ContourPlot[If[0 < yCurv11[x] < xCurv11[x], grad11[x]],
    {α, 0, 20}, {x, xMin, xMax}, Contours → {0}, ContourStyle → {Red, Thickness[0.02]},
    ContourShading → False, PlotPoints → 20], Frame → False,
   Axes → True, AspectRatio → .7, AxesLabel → {"α", "x"}, ImageSize → Small]]
```



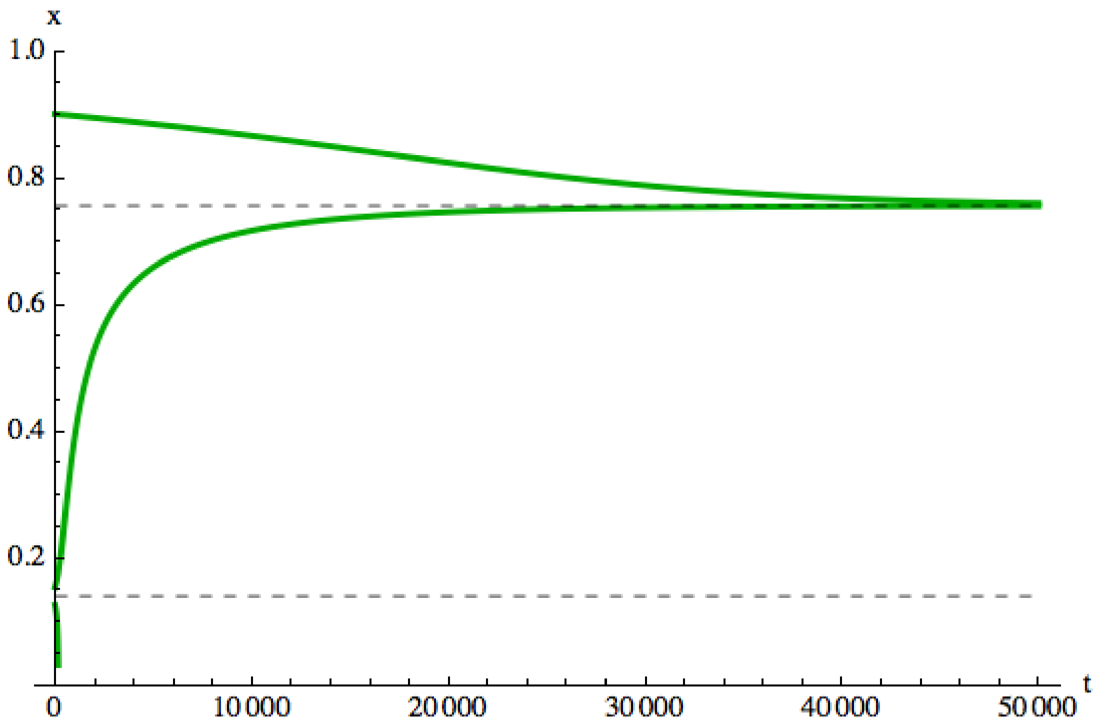
▪ Singular strategies

```
α = 9.5;

sing11 = ξ /. FindRoot[grad11[ξ], {ξ, .1}];
sing12 = ξ /. FindRoot[grad11[ξ], {ξ, .9}];
Show[monoBifPlot, Graphics[{Dashed, Line[{{α, xMin}, {α, xMax}}]}],
 Graphics[{PointSize[Large], Point[{{α, sing11}, {α, sing12}}]}], Frame → False,
 Axes → True, AspectRatio → .7, AxesLabel → {"α", "x"}, ImageSize → Small]
```



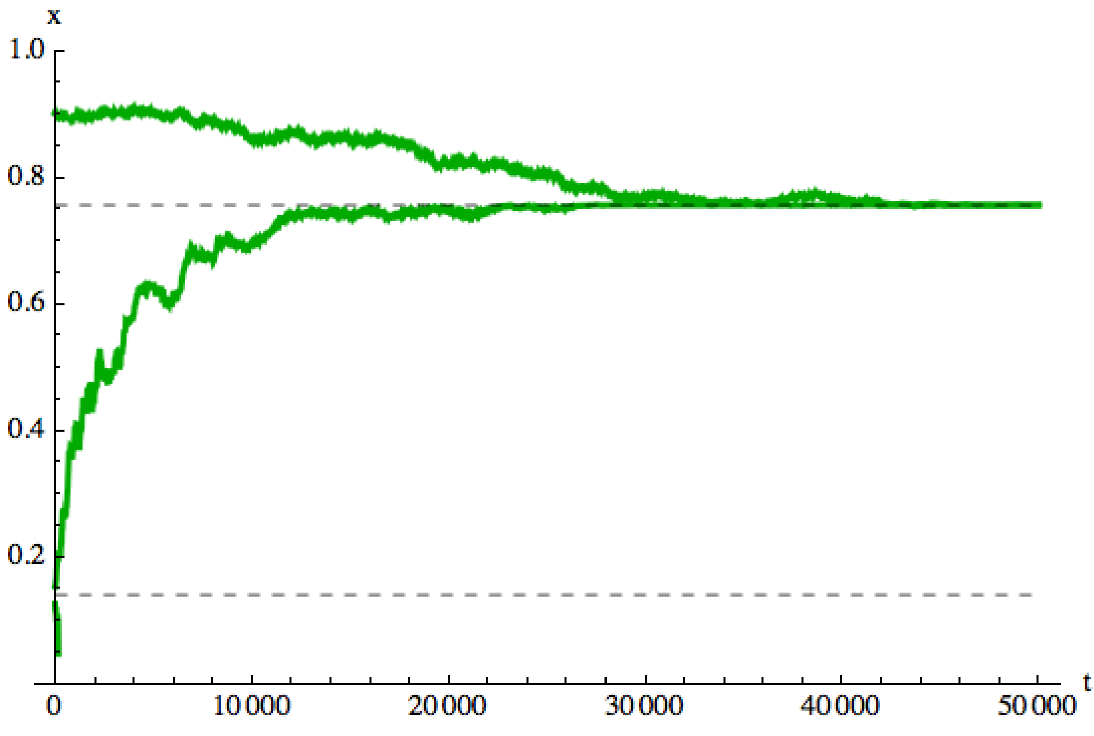
Canonical equation

```
α = 9.5;

x0 = {.9 sing11, 1.1 sing11, .9 xMax};(* different starting points *)
t0 = 0;
t∞ = 50 000;
Δt = 10;
noofStarts = Length[x0];
For[i = 1, i ≤ noofStarts, i++,
  data[i] = {};
  x = x0[[i]];
  t = t0;
  While[t ≤ t∞ ⋀ n11[x] > 0 ⋀ x > xMin,
   data[i] = Join[data[i], {{t, x}}];
   x = x + Δt driftMonomorph[x];
   t = t + Δt];
 ];

Show[
 Table[ListPlot[data[i], PlotStyle → {Thick, Darker[Green]}, Joined → True],
  {i, 1, noofStarts}],
 Graphics[{Dashed, Line[{{t0, sing11}, {t∞, sing11}}]}],
 Graphics[{Dashed, Line[{{t0, sing12}, {t∞, sing12}}]}],
 AxesOrigin → {0, 0}, PlotRange → {0, 1}, AxesLabel → {"t", "x"}
]
```



- **Stochastic differential equation (Ito)**

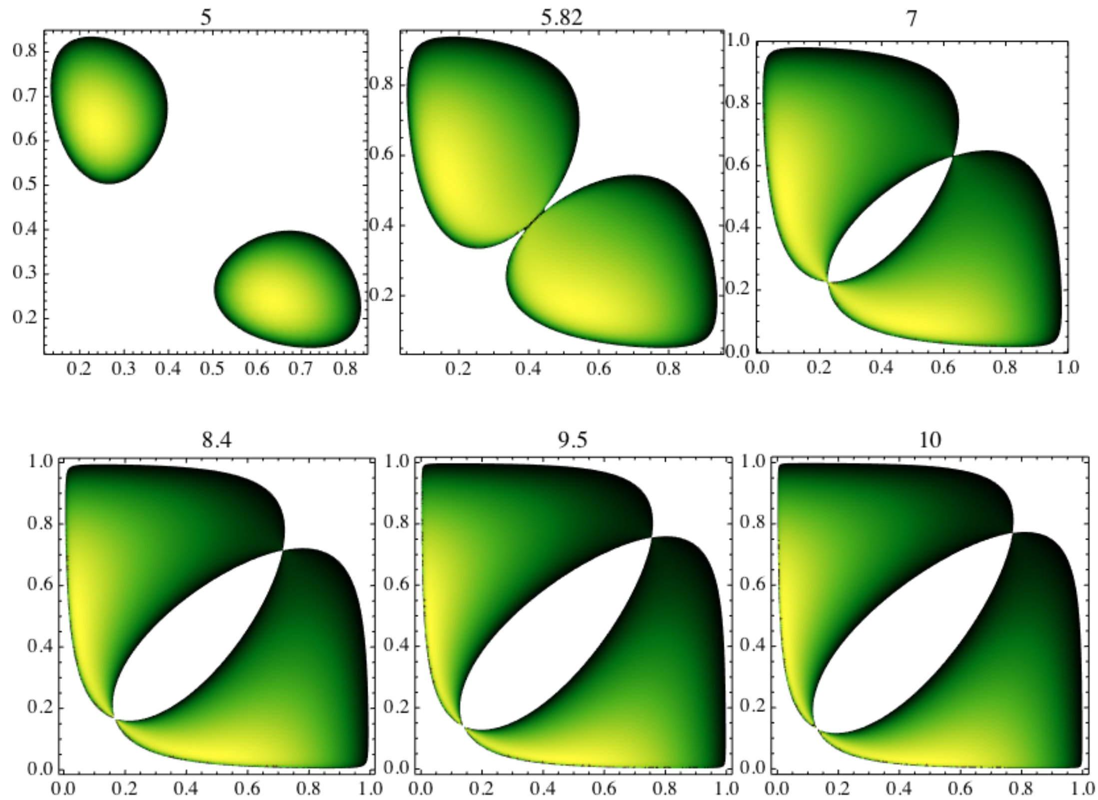
```
α = 9.5;

x0 = {.9 sing11, 1.1 sing11, .9 xMax};(* different starting points *)
t0 = 0;
t∞ = 50 000;
Δt = 10;
noofStarts = Length[x0];
For[i = 1, i ≤ noofStarts, i++,
  data[i] = {};
  x = x0[[i]];
  t = t0;
  While[t ≤ t∞ ⋀ n11[x] > 0 ⋀ x > xMin,
   data[i] = Join[data[i], {{t, x}}];
   z = RandomReal[NormalDistribution[0, 1]];
   x = x + Δt driftMonomorph[x] + z Sqrt[Δt diffMonomorph[x]] ;
   t = t + Δt];
 ];

Show[
 Table[ListPlot[data[i], PlotStyle → {Thick, Darker[Green]}, Joined → True],
  {i, 1, noofStarts}],
 Graphics[{Dashed, Line[{{t0, sing11}, {t∞, sing11}}]}],
 Graphics[{Dashed, Line[{{t0, sing12}, {t∞, sing12}}]}],
 AxesOrigin → {0, 0}, PlotRange → {0, 1}, AxesLabel → {"t", "x"}
]
```



Dimorphic evolution

- **Mutual invadability plots MIP**

```
Do[MIP[α] = Quiet[DensityPlot[Block[{inv}, inv = {s1[x1, x2], s1[x2, x1]};
     If[0 < inv⟦1⟧ && 0 < inv⟦2⟧, n21[x1, x2] n22[x1, x2]]],
    {x1, xMin, xMax}, {x2, xMin, xMax}, PlotPoints → 50, PlotRange → All,
    ColorFunction → "AvocadoColors"]], {α, {5, 5.82, 7, 8.4, 9.5, 10}}];
Row[Table[Show[MIP[α], PlotLabel → α, ImageSize → Small], {α, {5, 5.82, 7, 8.4, 9.5, 10}}]]
```

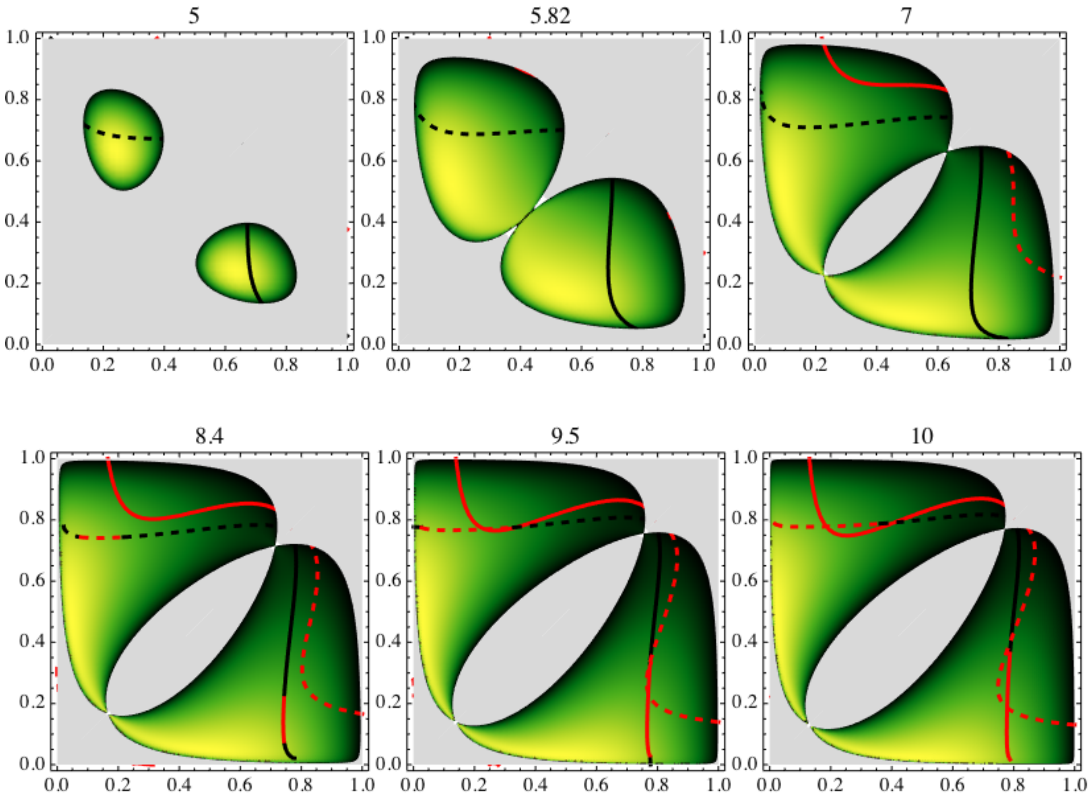


- **"Anti-MIP"**
  (**to be used later as a mask to cover what happens outside the oexistence region**)

```
Do[antiMIP[α] = Quiet[
   RegionPlot[s1[x, y] < 0 ⋁ s1[y, x] < 0, {x, xMin, xMax}, {y, xMin, xMax}, PlotPoints → 50,
    PlotStyle → LightGray, BoundaryStyle → None]], {α, {5, 5.82, 7, 8.4, 9.5, 10}}]
```

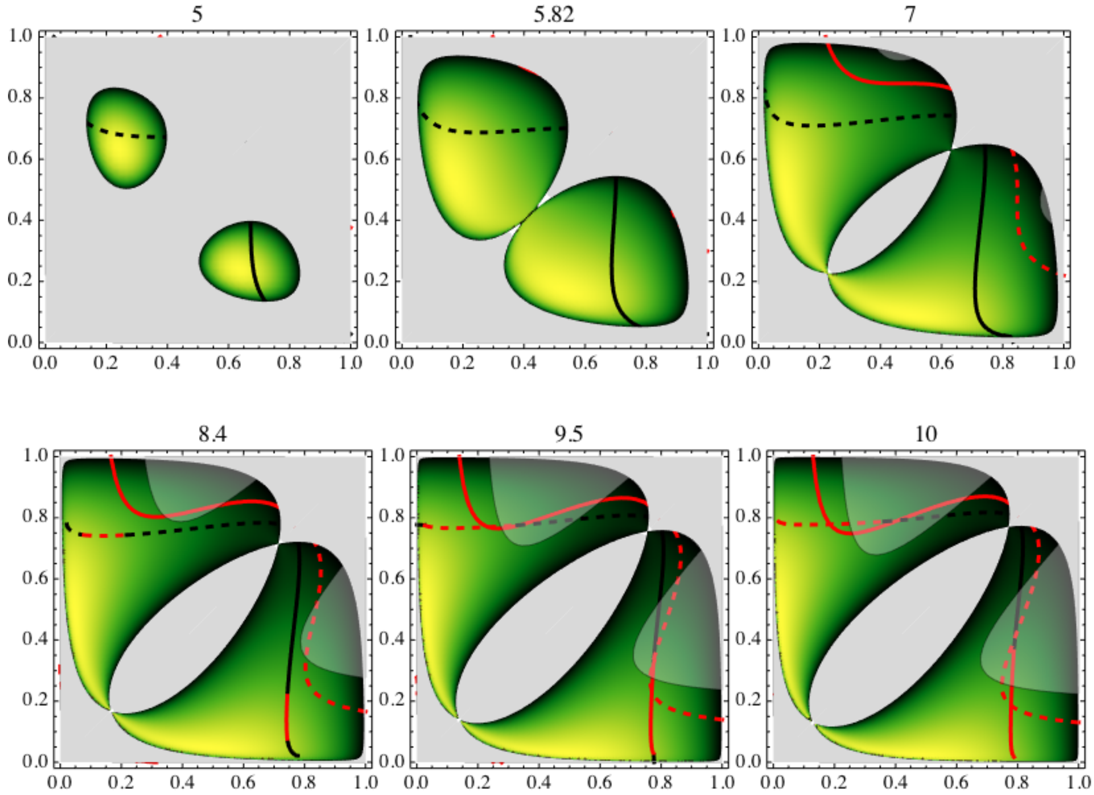
- **Evolutionary isoclines**

```
Do[IPes1[α] = Quiet[ContourPlot[If[0 > curv21[x1, x2] && Abs[x1 - x2] > .005 (xMax - xMin),
      grad21[x1, x2]], {x1, xMin, xMax}, {x2, xMin, xMax}, Contours → {0},
    ContourStyle → {Thick, Black}, ContourShading → False, PlotPoints → 10]];
  IPbp1[α] = Quiet[ContourPlot[If[0 < curv21[x1, x2] && Abs[x1 - x2] > .005 (xMax - xMin),
      grad21[x1, x2]], {x1, xMin, xMax}, {x2, xMin, xMax}, Contours → {0},
    ContourStyle → {Thick, Red}, ContourShading → False, PlotPoints → 10]];
  IPes2[α] = Quiet[ContourPlot[If[0 > curv22[x1, x2] && Abs[x1 - x2] > .005 (xMax - xMin),
      grad22[x1, x2]], {x1, xMin, xMax}, {x2, xMin, xMax}, Contours → {0},
    ContourStyle → {Thick, Black, Dashed}, ContourShading → False, PlotPoints → 10]];
  IPbp2[α] = Quiet[ContourPlot[If[0 < curv22[x1, x2] && Abs[x1 - x2] > .005 (xMax - xMin),
      grad22[x1, x2]], {x1, xMin, xMax}, {x2, xMin, xMax}, Contours → {0},
    ContourStyle → {Thick, Red, Dashed}, ContourShading → False,
    PlotPoints → 10]], {α, {5, 5.82, 7, 8.4, 9.5, 10}}];

Row[Table[Show[MIP[α], IPes1[α], IPbp1[α], IPes2[α], IPbp2[α],
   antiMIP[α], PlotLabel → α, ImageSize → Small], {α, {5, 5.82, 7, 8.4, 9.5, 10}}]]
```



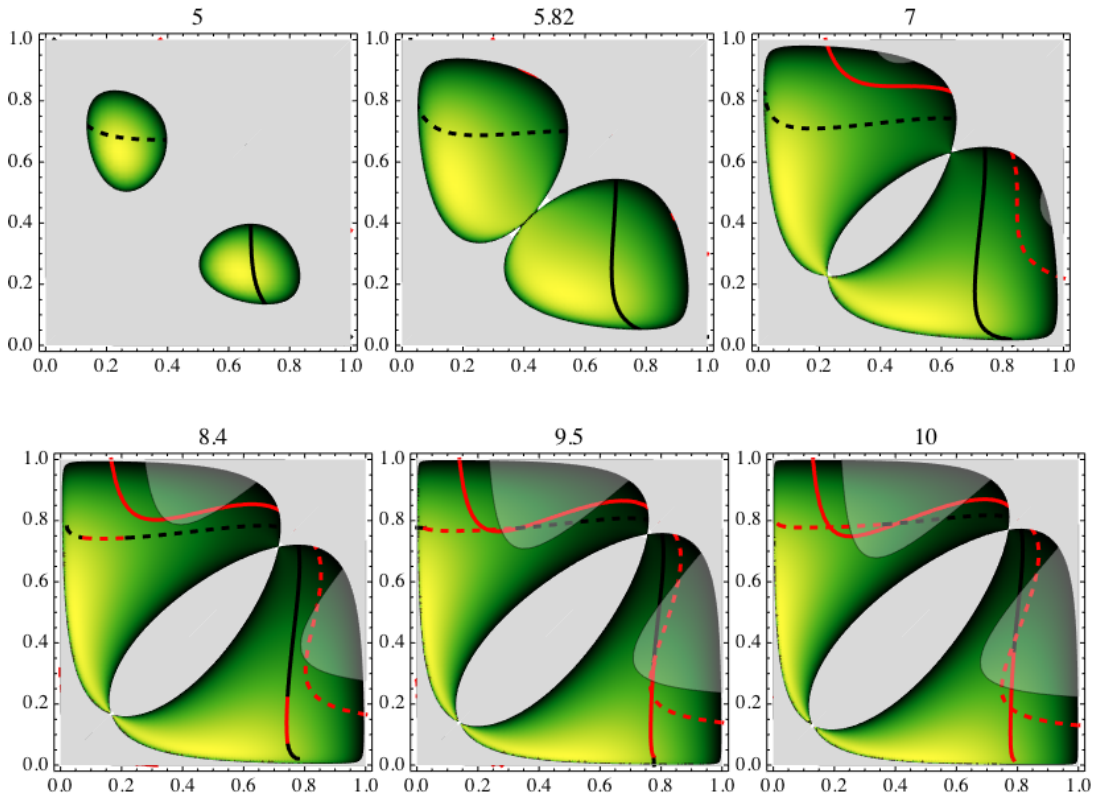
- **Total stability**

```
Do[totStab[α] = Block[{A, a11, a12, a21, a22}, A = ∂{{x1,x2}} {grad21[x1, x2], grad22[x1, x2]};
    a11 = A⟦1, 1⟧; a12 = A⟦1, 2⟧; a21 = A⟦2, 1⟧; a22 = A⟦2, 2⟧;
    Quiet[RegionPlot[a11 < 0 && a22 < 0 && Abs[a12 a21] < a11 a22, {x1, xMin, xMax}, {x2, xMin,
        xMax}, PlotStyle → {LightGray, Opacity[0.4]}]]], {α, {5, 5.82, 7, 8.4, 9.5, 10}}];
Row[Table[Show[MIP[α], IPes1[α], IPbp1[α], IPes2[α], IPbp2[α], totStab[α],
   antiMIP[α], PlotLabel → α, ImageSize → Small], {α, {5, 5.82, 7, 8.4, 9.5, 10}}]]
```



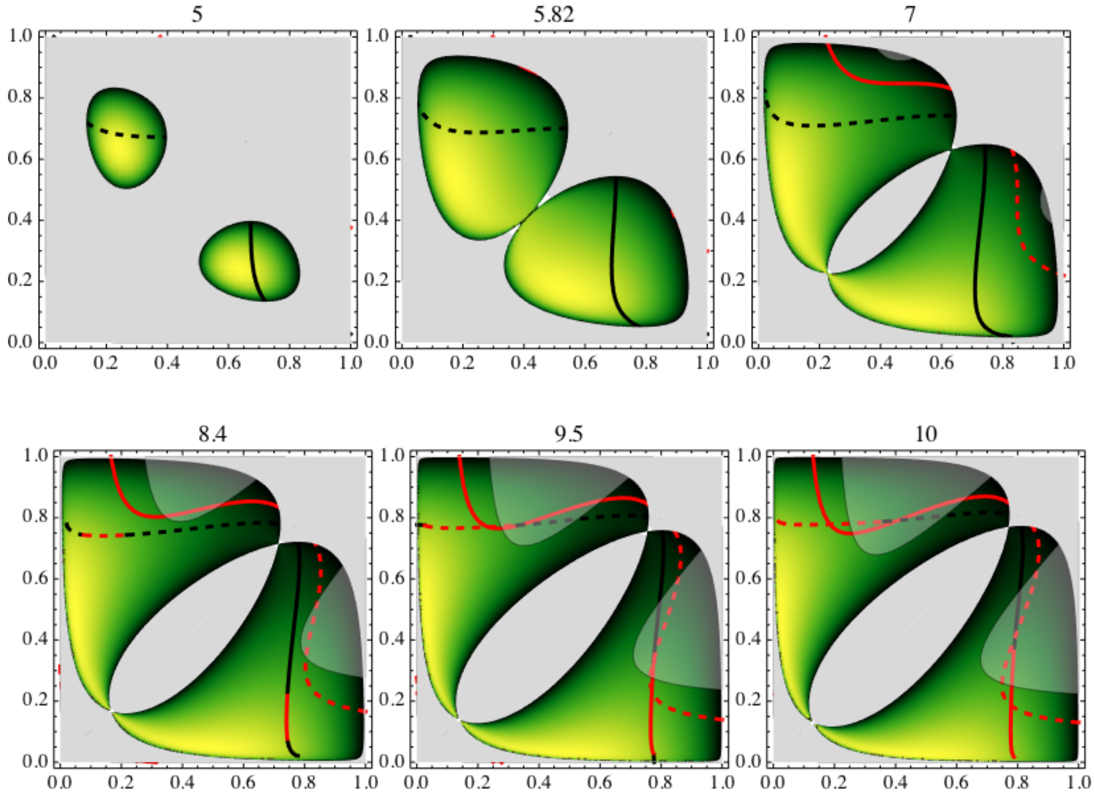
- **Strong stability**

```
Do[strongStab[α] =
   Block[{A, a11, a12, a21, a22}, A = ∂{{x1,x2}} {grad21[x1, x2], grad22[x1, x2]};
    a11 = A⟦1, 1⟧; a12 = A⟦1, 2⟧; a21 = A⟦2, 1⟧; a22 = A⟦2, 2⟧; Quiet[
     RegionPlot[a11 < 0 && a22 < 0 && a12 a21 < a11 a22, {x1, xMin, xMax}, {x2, xMin, xMax},
      PlotStyle → {LightGray, Opacity[0.4]}]]], {α, {5, 5.82, 7, 8.4, 9.5, 10}}];
Row[Table[Show[MIP[α], IPes1[α], IPbp1[α], IPes2[α], IPbp2[α], strongStab[α],
   antiMIP[α], PlotLabel → α, ImageSize → Small], {α, {5, 5.82, 7, 8.4, 9.5, 10}}]]
```


5
5.82
7
8.4
9.5
10
0.0
0.2
0.4
0.6
0.8
1.0

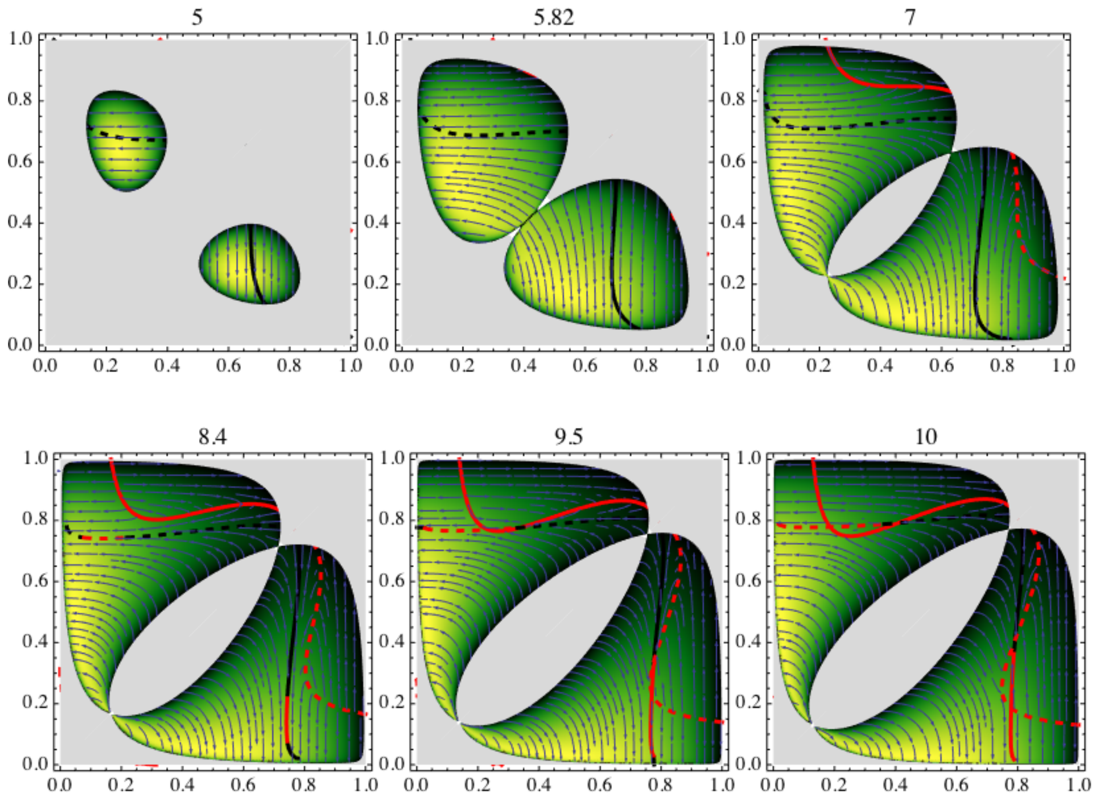
- **Weak stability**

```
Do[weakStab[α] = Block[{A, a11, a12, a21, a22}, A = ∂{{x1,x2}} {grad21[x1, x2], grad22[x1, x2]};
    a11 = A[[1, 1]]; a12 = A[[1, 2]]; a21 = A[[2, 1]]; a22 = A[[2, 2]]; Quiet[
     RegionPlot[(a11 < 0 || a22 < 0) && a12 a21 < a11 a22, {x1, xMin, xMax}, {x2, xMin, xMax},
      PlotStyle → {LightGray, Opacity[0.4]}]]], {α, {5, 5.82, 7, 8.4, 9.5, 10}}];
Row[Table[Show[MIP[α], IPes1[α], IPbp1[α], IPes2[α], IPbp2[α], weakStab[α],
   antiMIP[α], PlotLabel → α, ImageSize → Small], {α, {5, 5.82, 7, 8.4, 9.5, 10}}]]
```



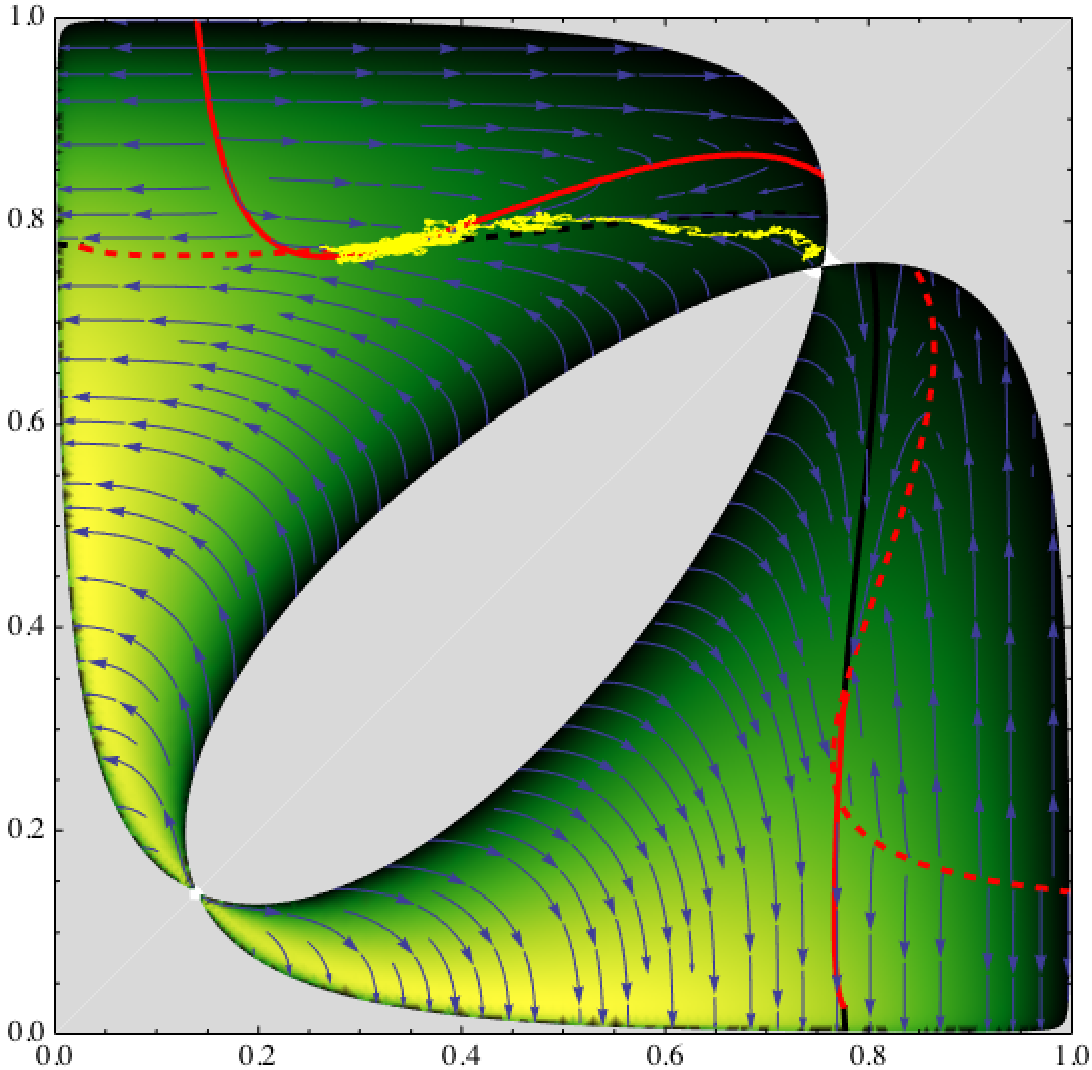
Canonical equation

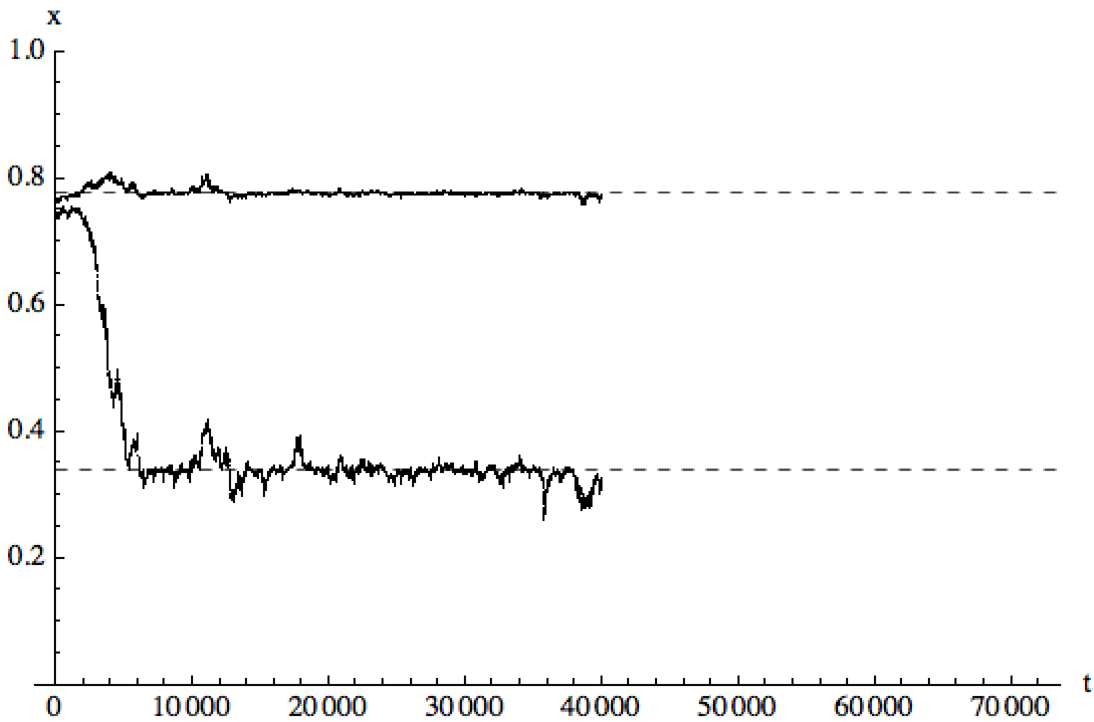
```
Do[stream[α] = StreamPlot[If[n21[x1, x2] > 0 ∧ n22[x1, x2] > 0 ∧ Abs[x1 - x2] > .01,
      driftDimorph[{x1, x2}], {0, 0}], {x1, xMin, xMax}, {x2, xMin, xMax},
     StreamPoints → Fine], {α, {5, 5.82, 7, 8.4, 9.5, 10}}] // Quiet;

Row[Table[Show[MIP[α], IPes1[α], IPbp1[α], IPes2[α], IPbp2[α], stream[α],
   antiMIP[α], PlotLabel → α, ImageSize → Small], {α, {5, 5.82, 7, 8.4, 9.5, 10}}]]
```



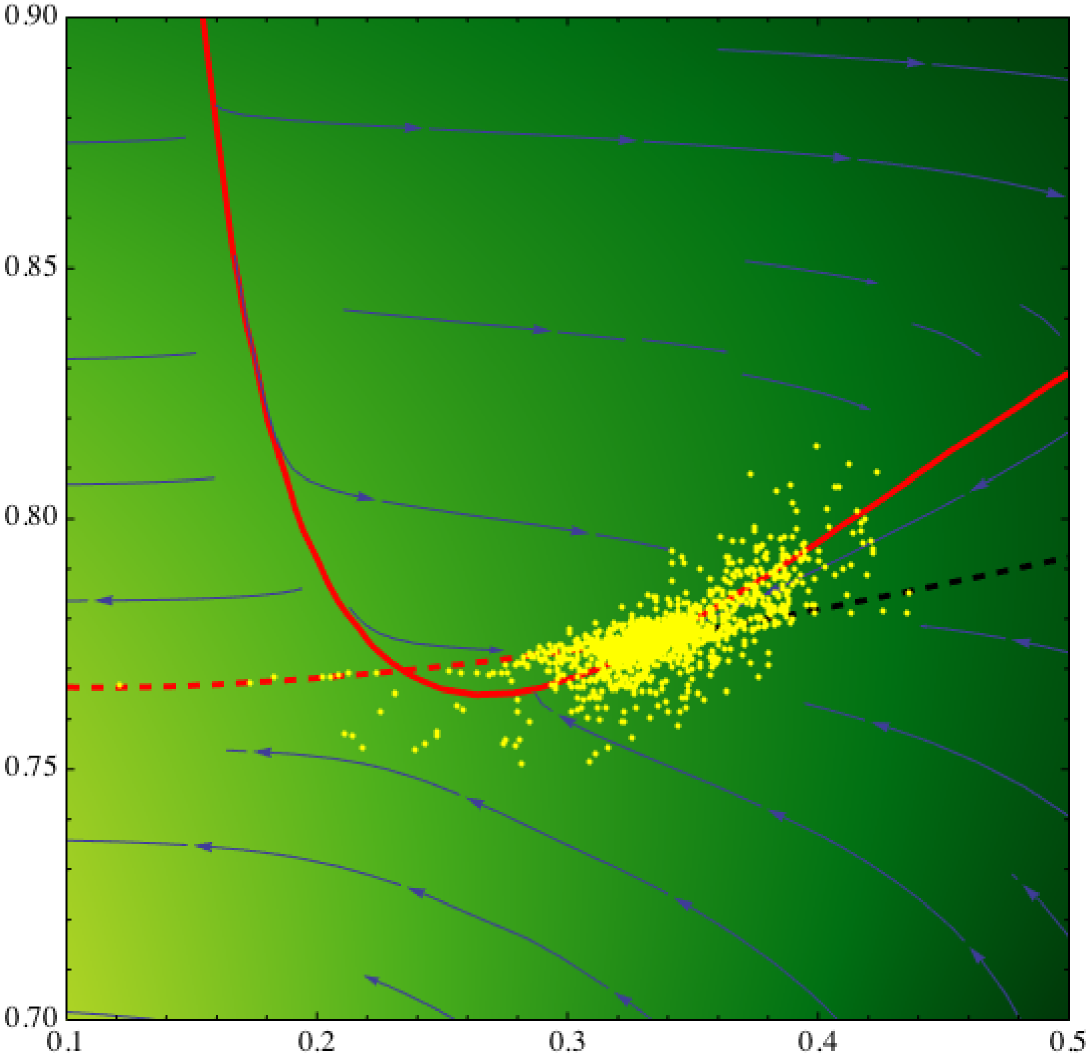
- **Stochastic differential equation (Ito)**

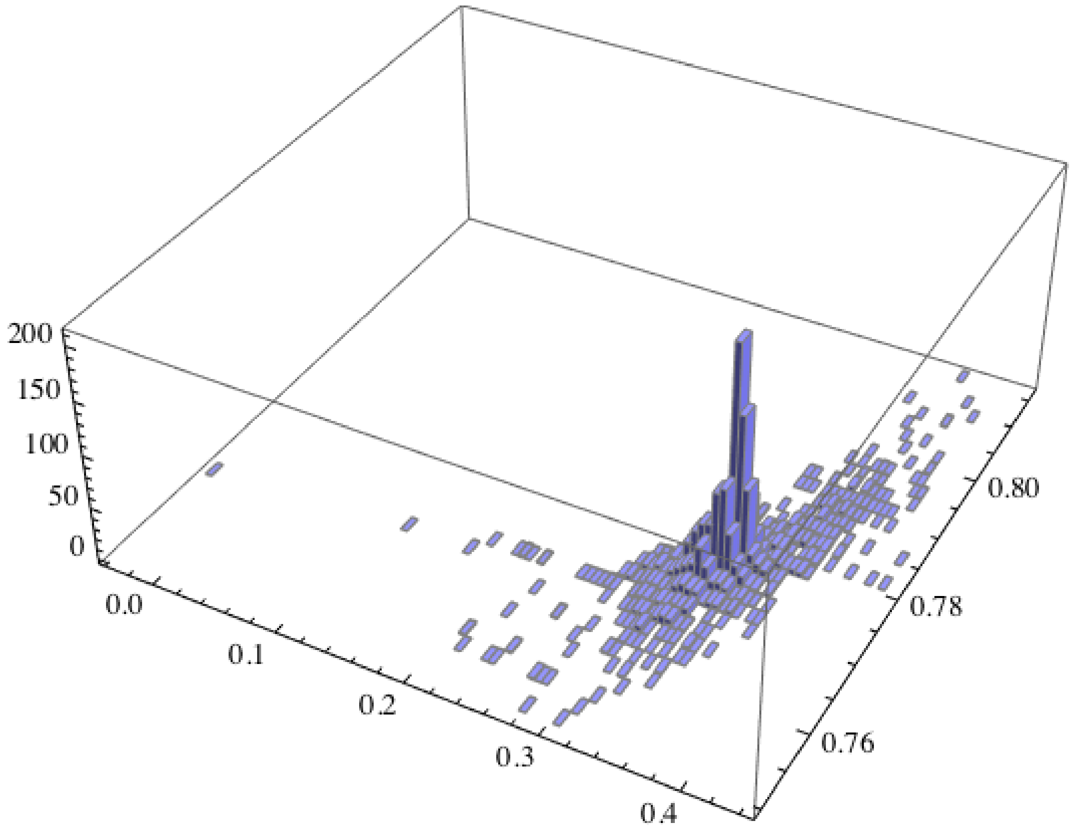
```
α = 9.5;

t∞ = 1000000;
Δt = 50;
x1 = sing12 - .01;
x2 = sing12 + .01;
data = {{x1, x2}};
t = 0;
While[t ≤ t∞ ⋀ n21[x1, x2] > 0 ⋀ n22[x1, x2] > 0 ⋀ xMin < x1 < xMax ⋀ xMin < x2 < xMax,
  z = RandomReal[NormalDistribution[0, 1], 2];
  {x1, x2} = {x1, x2} + Δt driftDimorph[{x1, x2}] + z Sqrt[Δt diffDimorph[{x1, x2}]] ;
  data = Join[data, {{x1, x2}}];
  t = t + Δt;
 ];

SDEorbit =
  ListPlot[data, PlotStyle → Yellow, Joined → True];

Show[
 MIP[α], IPes1[α], IPbp1[α], IPes2[α], IPbp2[α], stream[α], SDEorbit,
 antiMIP[α], PlotRange → {{xMin, xMax}, {xMin, xMax}}, ImageSize → Medium]

{sing21, sing22} =
  {ξ1, ξ2} /. FindRoot[{grad21[ξ1, ξ2] == 0, grad22[ξ1, ξ2] == 0}, {{ξ1, .4}, {ξ2, .8}}];

Show[
 ListPlot[Flatten[data], PlotStyle → {Black, PointSize[.001]}],
 Graphics[{Dashed, Line[{{t0, sing21}, {t, sing21}}]}],
 Graphics[{Dashed, Line[{{t0, sing22}, {t, sing22}}]}],
 PlotRange → {xMin, xMax}, AxesLabel → {"t", "x"}]
```


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


x
1.0
0.8
0.6
0.4
0.2
0
10000
20000
30000
40000
50000
60000
70000
t

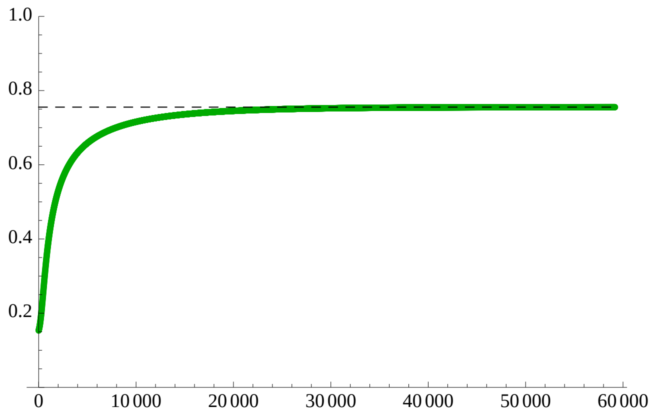
```
α = 9.5;

{sing21, sing22} =
  {ξ1, ξ2} /. FindRoot[{grad21[ξ1, ξ2] == 0, grad22[ξ1, ξ2] == 0}, {{ξ1, .4}, {ξ2, .8}}];

t∞ = 5 000 000;
Δt = 500;
x1 = sing21 + .01 (Random[] - .5);
x2 = sing22 + .01 (Random[] - .5);
data = {{x1, x2}};
t = 0;
While[t ≤ t∞ ∧ n21[x1, x2] > 0 ∧ n22[x1, x2] > 0 ∧ xMin < x1 < xMax ∧ xMin < x2 < xMax,
  z = RandomReal[NormalDistribution[0, 1], 2];
  {x1, x2} = {x1, x2} + Δt driftDimorph[{x1, x2}] + z √(Δt diffDimorph[{x1, x2}]) ;
  data = Join[data, {{x1, x2}}];
  t = t + Δt;
 ];

SDEorbit =
  ListPlot[data, PlotStyle → {Yellow, PointSize[Small]}, Joined → False];

Show[
 MIP[α], IPes1[α], IPbp1[α], IPes2[α], IPbp2[α], stream[α], SDEorbit,
 antiMIP[α], PlotRange → {{.1, .5}, {.7, .9}}, ImageSize → Medium]

Histogram3D[data]
```



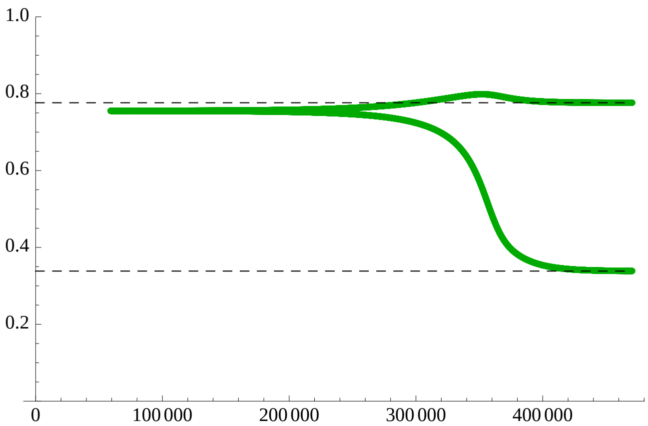

200
150
100
50
0
0.0
0.1
0.2
0.3
0.4
0.76
0.78
0.80

Evolutionary tree (CE)

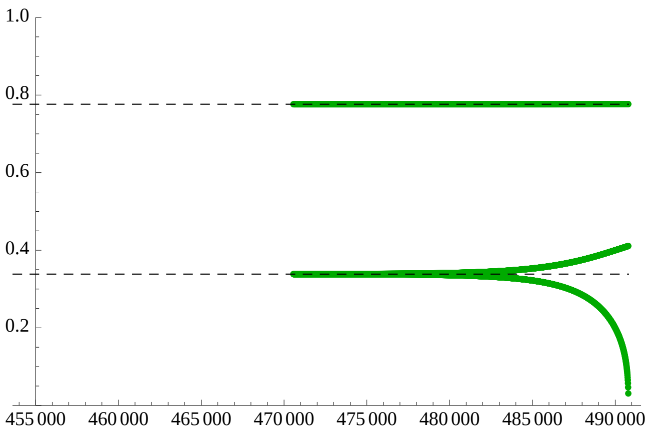
```
α = 9.5;

sing11 = ξ /. FindRoot[grad11[ξ], {ξ, .1}];
x0 = {1.1 sing11};
t0 = 0;
t∞ = 5 000 000;
Δt = 10;
ϵ = .0001;
dataMonomorph = {};
x = x0; t = t0;
While[t < t∞ && 0 < grad11[x⟦1⟧ - ϵ] grad11[x⟦1⟧ + ϵ],
  dataMonomorph = Join[dataMonomorph, {{t, x⟦1⟧}}];
  x = x + Δt driftMonomorph[x];
  t = t + Δt;
 ];

CEorbitMonomorph =
 Show[ListPlot[dataMonomorph, PlotStyle → {Darker[Green], Thick}, Joined → False,
   PlotRange → {0, 1}], Graphics[{Dashed, Line[{{t0, sing12}, {t, sing12}}]}]]
```



```
{sing21, sing22} =
  {ξ1, ξ2} /. FindRoot[{grad21[ξ1, ξ2] == 0, grad22[ξ1, ξ2] == 0}, {{ξ1, .4}, {ξ2, .8}}];
dataDimorph = {};
t = dataMonomorph[[-1]][[1]];
x = {dataMonomorph[[-1]][[2]] - ϵ, dataMonomorph[[-1]][[2]] + ϵ};
Δt = 100;
While[t ≤ t∞ && 0 < grad21[x[[1]] - ϵ, x[[2]]] grad21[x[[1]] + ϵ, x[[2]]],
  dataDimorph = Join[dataDimorph, {{t, x[[1]]}, {t, x[[2]]}}];
  x = x + Δt driftDimorph[x];
  t = t + Δt;
 ];

CEorbitDimorph =
 Show[ListPlot[dataDimorph, PlotStyle → {Darker[Green], Thick}, Joined → False,
   PlotRange → {0, 1}], Graphics[{Dashed, Line[{{t0, sing21}, {t, sing21}}]}],
  Graphics[{Dashed, Line[{{t0, sing22}, {t, sing22}}]}]]
```




```
dataTrimorph = {};
x = {sing21 - ϵ, sing21 + ϵ, sing22};
t = dataDimorph[[-1]][[1]];
Δt = 10;
While[t ≤ t∞ && xMin < x[[1]] < xMax && xMin < x[[2]] < xMax && xMin < x[[3]] < xMax,
  dataTrimorph = Join[dataTrimorph, {{t, x[[1]]}, {t, x[[2]]}, {t, x[[3]]}}];
  x = x + Δt driftTrimorph[x];
  t = t + Δt;
 ];

CEorbitTrimorph =
 Show[ListPlot[dataTrimorph, PlotStyle → {Darker[Green], Thick}, Joined → False,
   PlotRange → {0, 1}], Graphics[{Dashed, Line[{{t0, sing21}, {t, sing21}}]}],
  Graphics[{Dashed, Line[{{t0, sing22}, {t, sing22}}]}]]
```



```
Show[ListPlot[Join[dataMonomorph, dataDimorph, dataTrimorph],
   PlotStyle → {Darker[Green], Thick}, Joined → False, PlotRange → {0, 1}],
  Graphics[{Dashed, Line[{{t0, sing21}, {t, sing21}}]}],
  Graphics[{Dashed, Line[{{t0, sing22}, {t, sing22}}]}],
  AxesLabel → {"time", "strategy"}, PlotLabel → α]
```