Lotka-Volterra asymmetric competition

- **Issues**

  - multiple singular points
  - bifurcation plots
  - canonical equation (CE)
  - when stochasticity matters (SDE)

- **The model**

$$\frac{d}{dt} n_i = r[x_i]\, n_i \left(1 - \frac{\sum_{\forall j} a[x_i, x_j]\, n_j}{k[x_i]}\right) \qquad (i = 1, \ldots, k)$$

Monomorphic resident population

- **Invasion fitness**

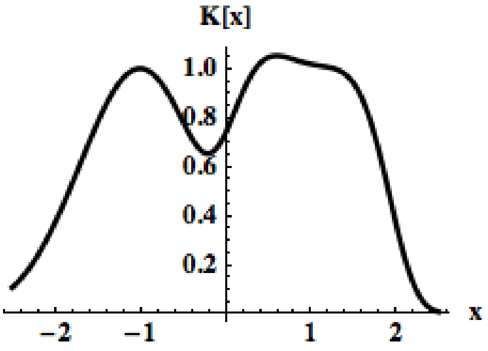
$$s_{x\_}[y\_] = r[y] \left(1 - \frac{a[y, x]\, K[x]}{K[y]}\right);$$

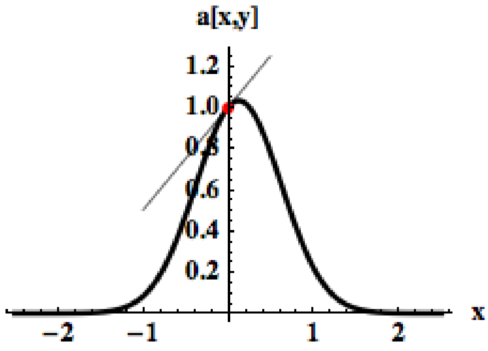
- **Default parameter values & functions**

```
r[x_] = 1;

K[x_] = E^(-(x-γ)^4) + E^(-(x+γ)^2);

a[x_, y_] = E^(-α (x-y)^2 - β (x-y));

α = 2;
γ = 1;

βVals = {-2, -1.065, -.5, .5, .9, 2};

xMin = -2.5;
xMax = 2.5;
```

■ Carrying capacity & competition kernel

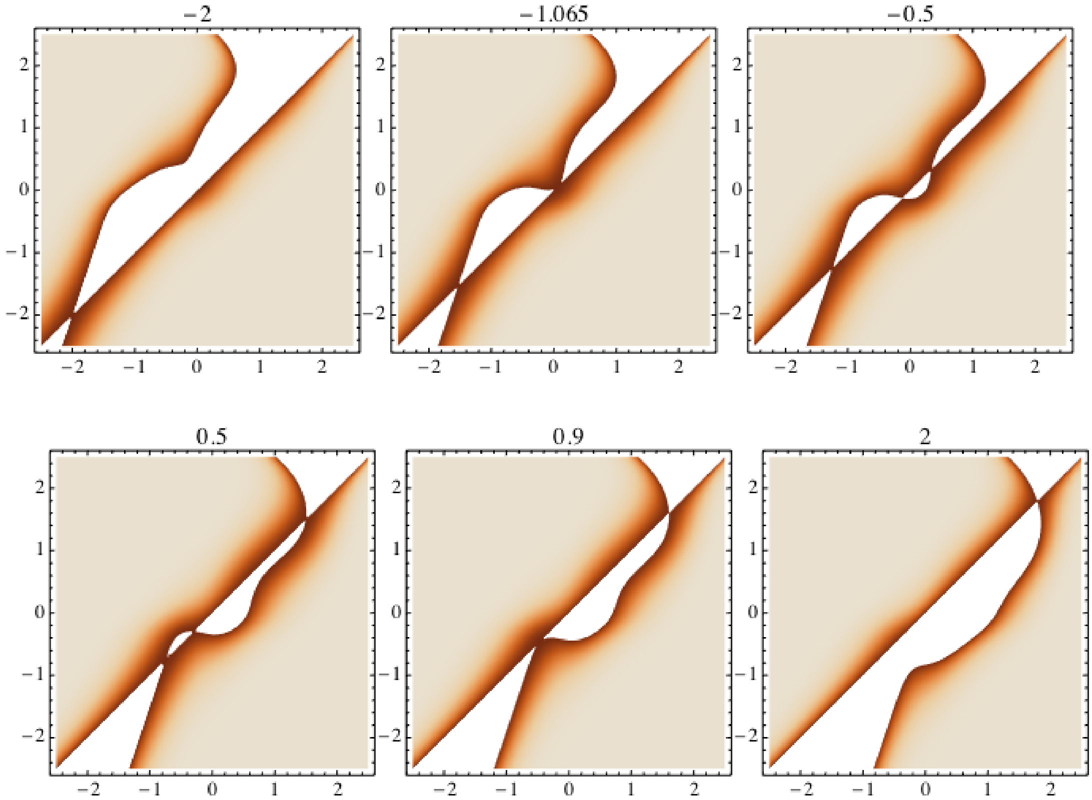
```
β = -.5;

Row[{

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

  Show[
   Plot[
    a[x, 0],
    {x, xMin, xMax},
    AxesOrigin → {0, 0},
    PlotRange → All,
    PlotStyle → {Thick, Black},
    AxesLabel → {"x", "a[x,y]"},
    LabelStyle → {Bold},
    ImageSize → {Small}
   ],

   Plot[
    1 + x (∂ξ a[ξ, 0] /. {ξ → 0}),
    {x, -1, .5},
    PlotStyle → {Black}
   ],
   Graphics[
    {Red, PointSize[Medium], Point[{0, 1}]}
   ]

  ]

 }]

Clear[β];
```



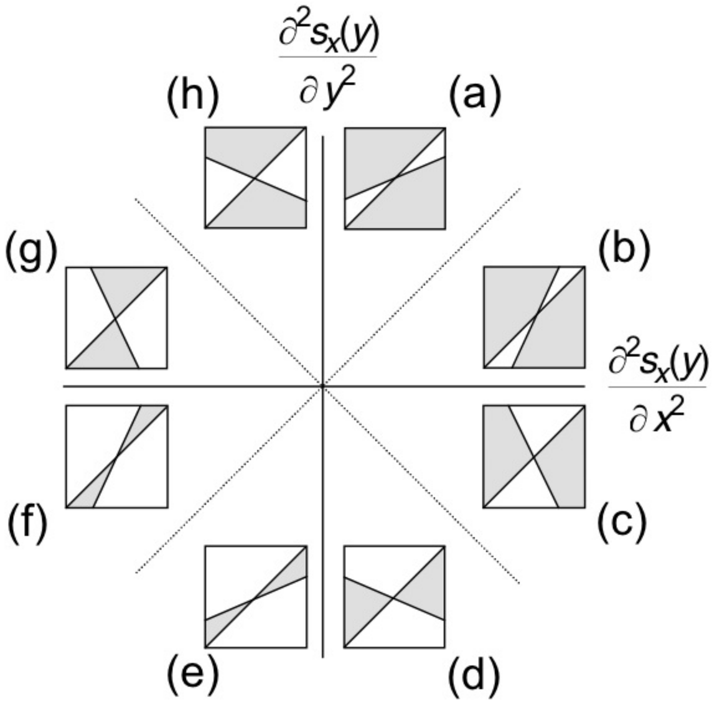


■ Pairwise invadability plot (PIP)

```
Do[
  PIP[β] =
   DensityPlot[
    Block[{inv},
     inv = sx[y];
     If[inv > 0, inv]],
    {x, xMin, xMax},
    {y, xMin, xMax},
    PlotPoints → 50,
    ColorFunction → "SiennaTones"
   ],
  {β, βVals}
 ];

Row[
 Table[
  Show[
   PIP[β],
   PlotLabel → β,
   ImageSize → Small
  ],
  {β, βVals}
 ]
]
```



- **The "8 cases"**

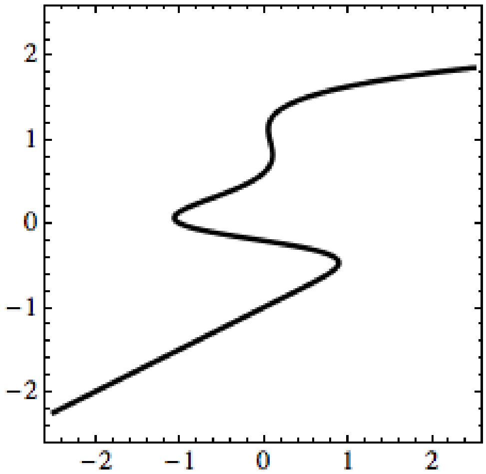


- **Fitness gradient & curvatures**

```
Clear[β];
grad[x_] = ∂η sξ[η] /. {ξ → x, η → x};
xCurv[x_] = ∂ξ,ξ sξ[η] /. {ξ → x, η → x};
yCurv[x_] = ∂η,η sξ[η] /. {ξ → x, η → x};
```

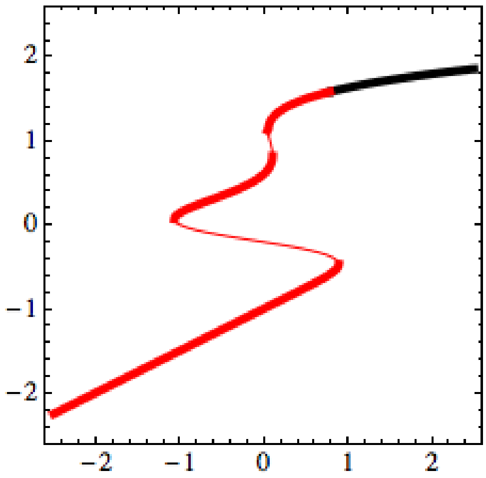
■ Bifurcation Plot (x vs β)

```
ContourPlot[
 grad[x] == 0,
 {β, -2.5, 2.5},
 {x, xMin, xMax},
 ContourStyle → {Black, Thick},
 ContourShading → False,
 PlotPoints → 30,
 ImageSize → Small
]
```



- **Slow procedure : do NOT run in classroom**

```
Show[

 (* ES attractor *)
 ContourPlot[
  If[yCurv[x] < Min[xCurv[x], 0], grad[x]],
  {β, -2.5, 2.5},
  {x, xMin, xMax},
  Contours → {0},
  ContourStyle → {Black, Thickness[0.02]},
  ContourShading → False,
  PlotPoints → 30
 ],

 (* ES repeller *)
 ContourPlot[
  If[xCurv[x] < yCurv[x] < 0, grad[x]],
  {β, -2.5, 2.5},
  {x, xMin, xMax},
  Contours → {0},
  ContourStyle → {Black, Thickness[.005]},
  ContourShading → False,
  PlotPoints → 30
 ],

 (* non-ES repeller *)
 ContourPlot[
  If[Max[xCurv[x], 0] < yCurv[x], grad[x]],
  {β, -2.5, 2.5},
  {x, xMin, xMax},
  Contours → {0},
  ContourStyle → {Red, Thickness[.005]},
  ContourShading → False,
  PlotPoints → 30
 ],

 (* BP *)
 ContourPlot[
  If[0 < yCurv[x] < xCurv[x], grad[x]],
  {β, -2.5, 2.5},
  {x, xMin, xMax},
  Contours → {0},
  ContourStyle → {Red, Thickness[0.02]},
  ContourShading → False,
  PlotPoints → 30
 ],

 ImageSize → Small

]
```



Dimorphic resident population

Invasion fitness

```
Clear[β];
```

$$s_{x1\_,x2\_}[y\_] = r[y]\left(1 - \frac{a[y,\ x1]\ n1[x1,\ x2] + a[y,\ x2]\ n2[x1,\ x2]}{K[y]}\right);$$

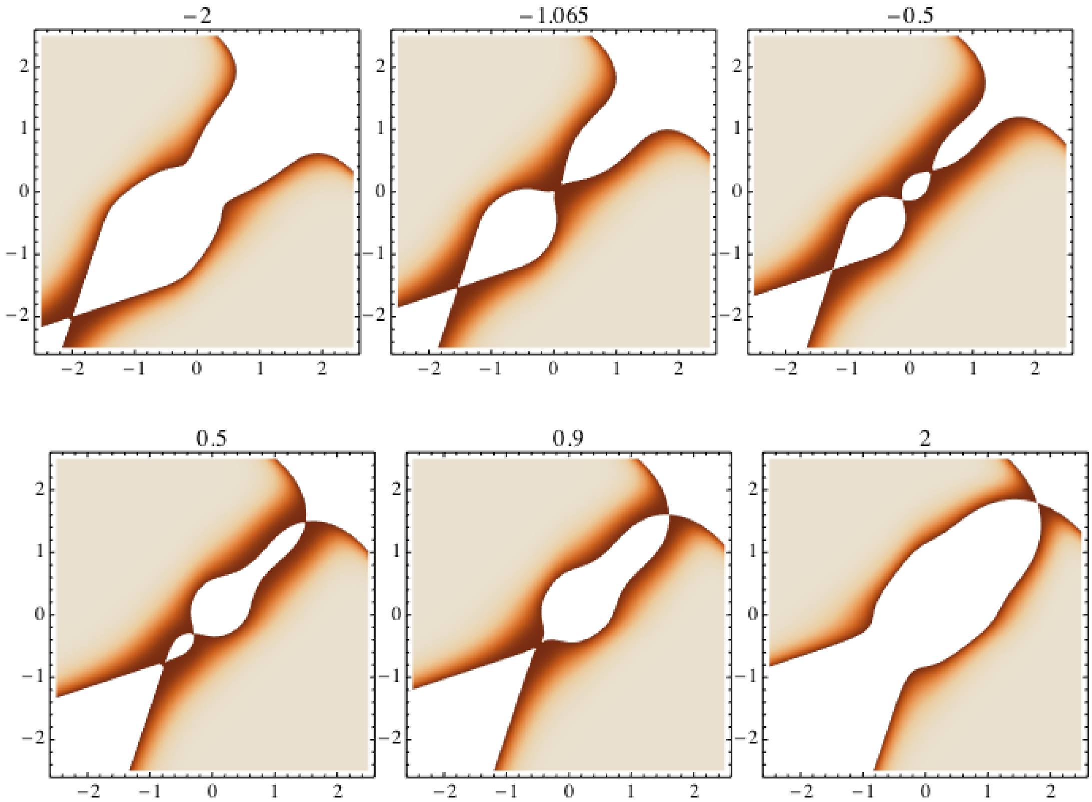
$$n1[x1\_,\ x2\_] = \frac{K[x1] - a[x1,\ x2]\ K[x2]}{1 - a[x1,\ x2]\ a[x2,\ x1]};$$

$$n2[x1\_,\ x2\_] = \frac{K[x2] - a[x2,\ x1]\ K[x1]}{1 - a[x1,\ x2]\ a[x2,\ x1]};$$

Mutual invadability plot (MIP)

```
Do[
 MIP[β] =
  DensityPlot[
   Block[{inv},
    inv = {sx[y], sy[x]};
    If[inv[[1]] > 0 ∧ inv[[2]] > 0, inv[[1]] inv[[2]]]
   ],
   {x, xMin, xMax},
   {y, xMin, xMax},
   PlotPoints → 50,
   ColorFunction → "SiennaTones"
  ],
 {β, βVals}
]
```

```
Row[
 Table[
  Show[MIP[β], PlotLabel → β, ImageSize → Small],
  {β, βVals}
 ]
]
```

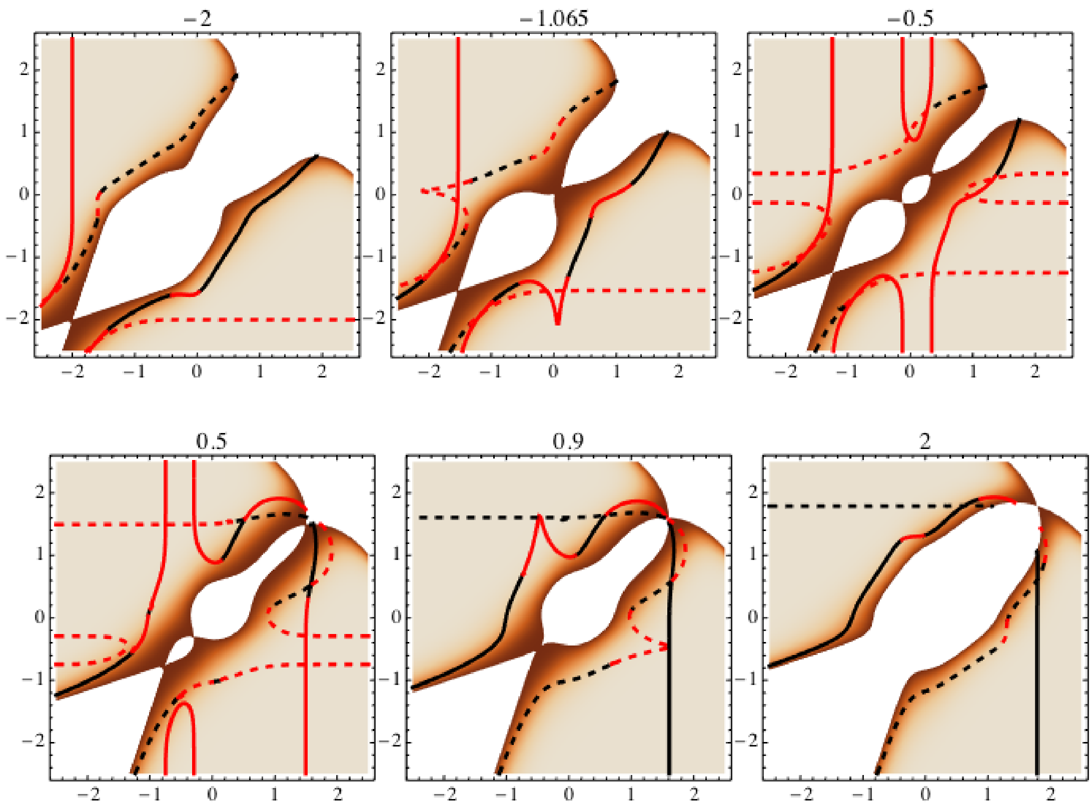


▪ Evolutionary isoclines

```
Clear[β];

grad1[x1_, x2_] = ∂η sξ1,ξ2[η] /. {ξ1 → x1, ξ2 → x2, η → x1};
grad2[x1_, x2_] = ∂η sξ1,ξ2[η] /. {ξ1 → x1, ξ2 → x2, η → x2};

curv1[x1_, x2_] = ∂η,η sξ1,ξ2[η] /. {ξ1 → x1, ξ2 → x2, η → x1};
curv2[x1_, x2_] = ∂η,η sξ1,ξ2[η] /. {ξ1 → x1, ξ2 → x2, η → x2};

coex[x1_, x2_] = 0 < n1[x1, x2] ⋀ 0 < n2[x1, x2] ⋀ .01 (xMax - xMin) < Abs[x1 - x2];
```

■ **Slow procedure : do NOT run in classroom**

```
Do[

  IPes1[β] =
   ContourPlot[If[0 > curv1[x1, x2] ∧ coex[x1, x2], grad1[x1, x2]],
    {x1, xMin, xMax}, {x2, xMin, xMax}, Contours → {0},
    ContourStyle → {Thick, Black}, ContourShading → False, PlotPoints → 30];

  IPbp1[β] =
   ContourPlot[If[0 < curv1[x1, x2] ∧ coex[x1, x2], grad1[x1, x2]],
    {x1, xMin, xMax}, {x2, xMin, xMax}, Contours → {0},
    ContourStyle → {Thick, Red}, ContourShading → False, PlotPoints → 30];

  IPes2[β] =
   ContourPlot[If[0 > curv2[x1, x2] ∧ coex[x1, x2], grad2[x1, x2]],
    {x1, xMin, xMax}, {x2, xMin, xMax}, Contours → {0},
    ContourStyle → {Thick, Black, Dashed}, ContourShading → False, PlotPoints → 30];

  IPbp2[β] =
   ContourPlot[If[0 < curv2[x1, x2] ∧ coex[x1, x2], grad2[x1, x2]],
    {x1, xMin, xMax}, {x2, xMin, xMax}, Contours → {0},
    ContourStyle → {Thick, Red, Dashed}, ContourShading → False, PlotPoints → 30],

  {β, βVals}
 ];

Row[
 Table[
  Show[
   MIP[β], IPes1[β], IPbp1[β], IPes2[β], IPbp2[β],
   PlotLabel → β, ImageSize → Small
  ],
  {β, βVals}
 ]
]
```



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

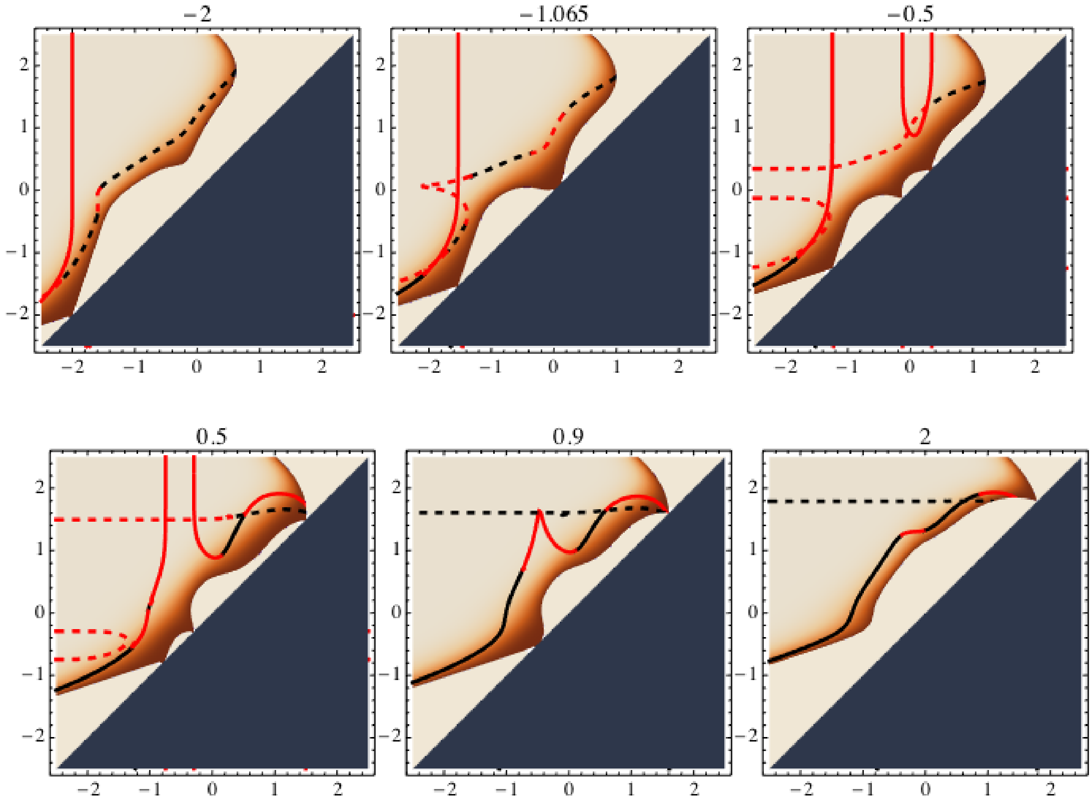
```
Do[
 antiMIP[β] =
  DensityPlot[
   Block[{inv}, inv = {sx1[x2], sx2[x1]}; If[inv[[1]] < 0 ∨ inv[[2]] < 0, 1]],
   {x1, xMin, xMax}, {x2, xMin, xMax},
   PlotPoints → 50, ColorFunction → "LakeColors"
  ],
 {β, βVals}
]
```

- **"HLT" (another mask to cover the lower triangle of the MIP)**

```
Do[
 HLT[β] =
  DensityPlot[
   If[x2 < x1, 1],
   {x1, xMin, xMax}, {x2, xMin, xMax},
   PlotPoints → 50, ColorFunction → "GrayYellowTones"
  ],
 {β, βVals}
]
```

- **Show isocline plots with masks**

```
Row[
 Table[
  Show[
   MIP[β], IPes1[β], IPbp1[β], IPes2[β], IPbp2[β], antiMIP[β], HLT[β],
   PlotLabel → β, ImageSize → Small
  ],
  {β, βVals}
 ]
]
```



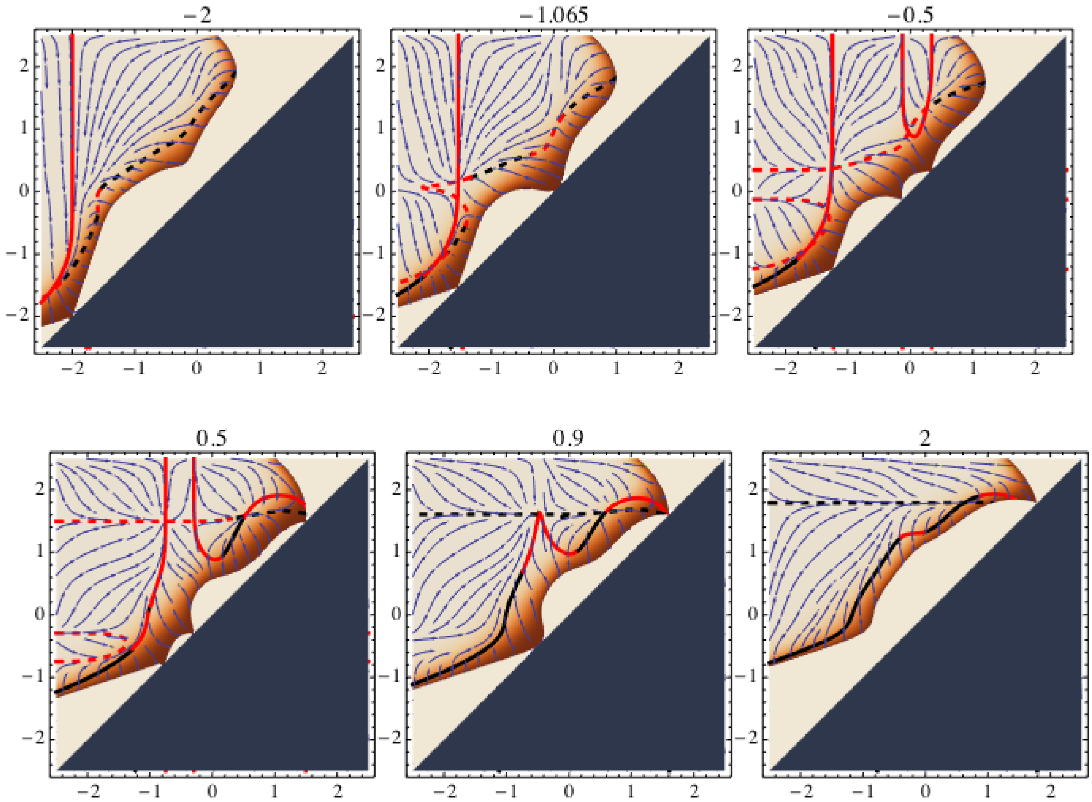
Canonical equation (CE)

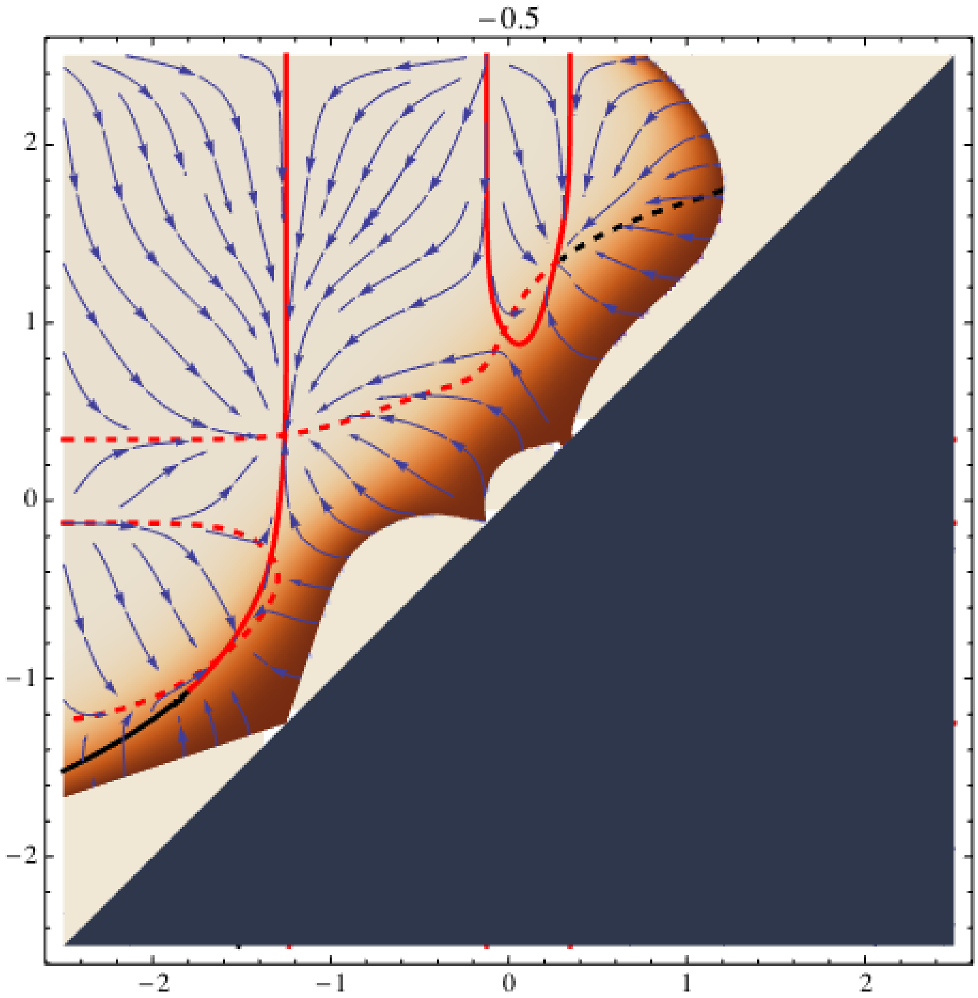
```
μ = 1; (* Mutation probability per birth event *)
σ = .01; (* Standard deviation of mutation step size *)

drift[{x1_, x2_}] = {.5 μ σ^2 n1[x1, x2] grad1[x1, x2], .5 μ σ^2 n2[x1, x2] grad2[x1, x2]};
```

- **Slow procedure : do NOT run in classroom**

```
Do[
  stream[β] =
   StreamPlot[
    drift[{x1, x2}],
    {x1, xMin, xMax}, {x2, xMin, xMax}
   ],
  {β, βVals}
 ];
```

```
Row[
 Table[
  Show[
   MIP[β], IPes1[β], IPbp1[β], IPes2[β], IPbp2[β], stream[β], antiMIP[β], HLT[β],
   PlotLabel → β, ImageSize → Small],
  {β, βVals}
 ]
]
```

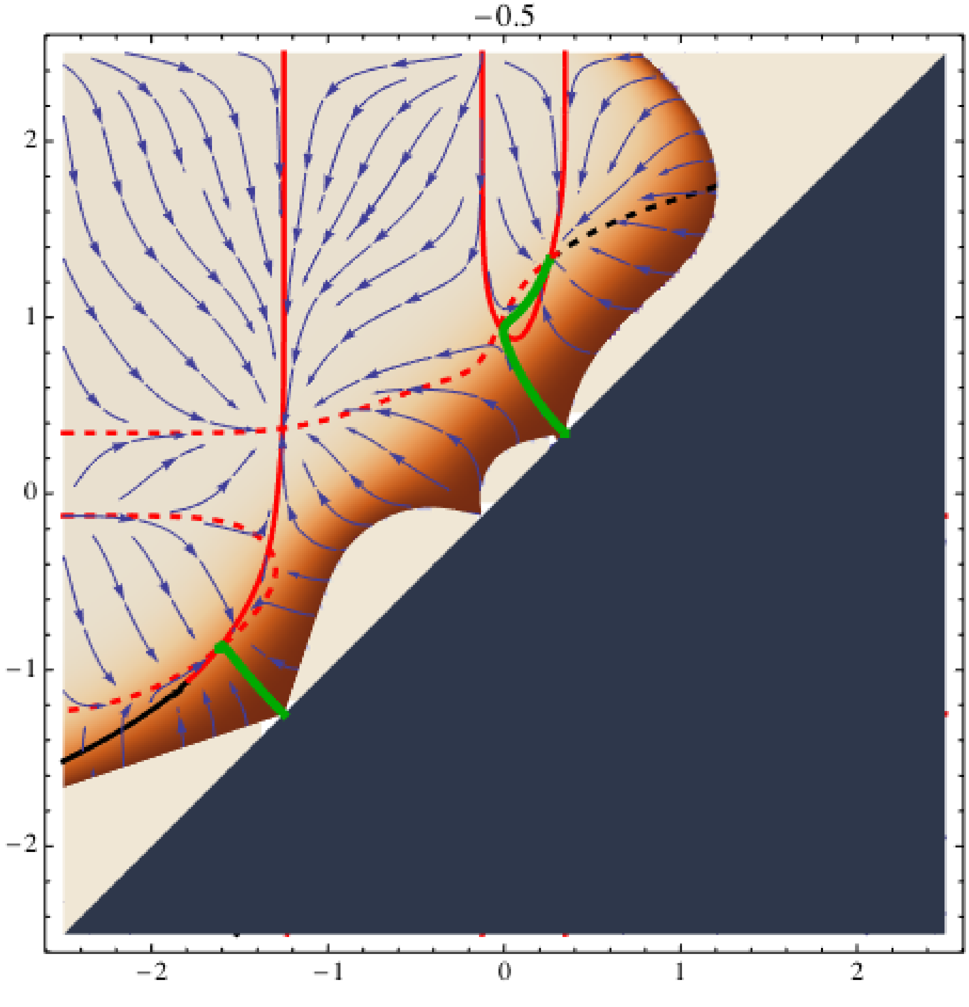




- **Particular orbits of the canonical equation**

```
β = -.5;

x0 = {.343690, .343691}; (* starting point *)
t0 = 0;(* start time *)
t∞ = 15000000;(* stop time *)
Δt = 5000;(* integration time step *)

data = {};
x = x0;
t = t0;
While[t ≤ t∞,
  data = Join[data, {Append[x, t]}];
  x = x + Δt drift[x];
  t = t + Δt;
 ];

CEorbit1 = ListPlot[data[[All, {1, 2}]],
   PlotStyle → {Darker[Green], Thickness[0.01]}, Joined → True];

Clear[β];
```

```
β = -.5;

x0 = {-1.25001, -1.24999}; (* starting point *)
t0 = 0; (* start time *)
t∞ = 15 000 000; (* stop time *)
Δt = 5000; (* integration time step *)

data = {};
x = x0;
t = t0;
While[t ≤ t∞,
  data = Join[data, {Append[x, t]}];
  x = x + Δt drift[x];
  t = t + Δt;
 ];

CEorbit2 = ListPlot[data[[All, {1, 2}]],
   PlotStyle → {Darker[Green], Thickness[0.01]}, Joined → True];

Clear[β];

β = -.5;

Show[
 MIP[β], IPes1[β], IPbp1[β], IPes2[β], IPbp2[β],
 stream[β], antiMIP[β], HLT[β],
 CEorbit1, CEorbit2,
 PlotLabel → β
]

Clear[β];
```



Stochastic orbits (SDE - ITO)

- **$\theta_3[x]$ = third absolute moment of the mutation step distribution (assumed to be Gaussian with mean zero and standard variation $\sigma[x]$)**

```
θ3 = 2 σ^3 Sqrt[2 / Pi];
```

- **Diffusion coefficient**

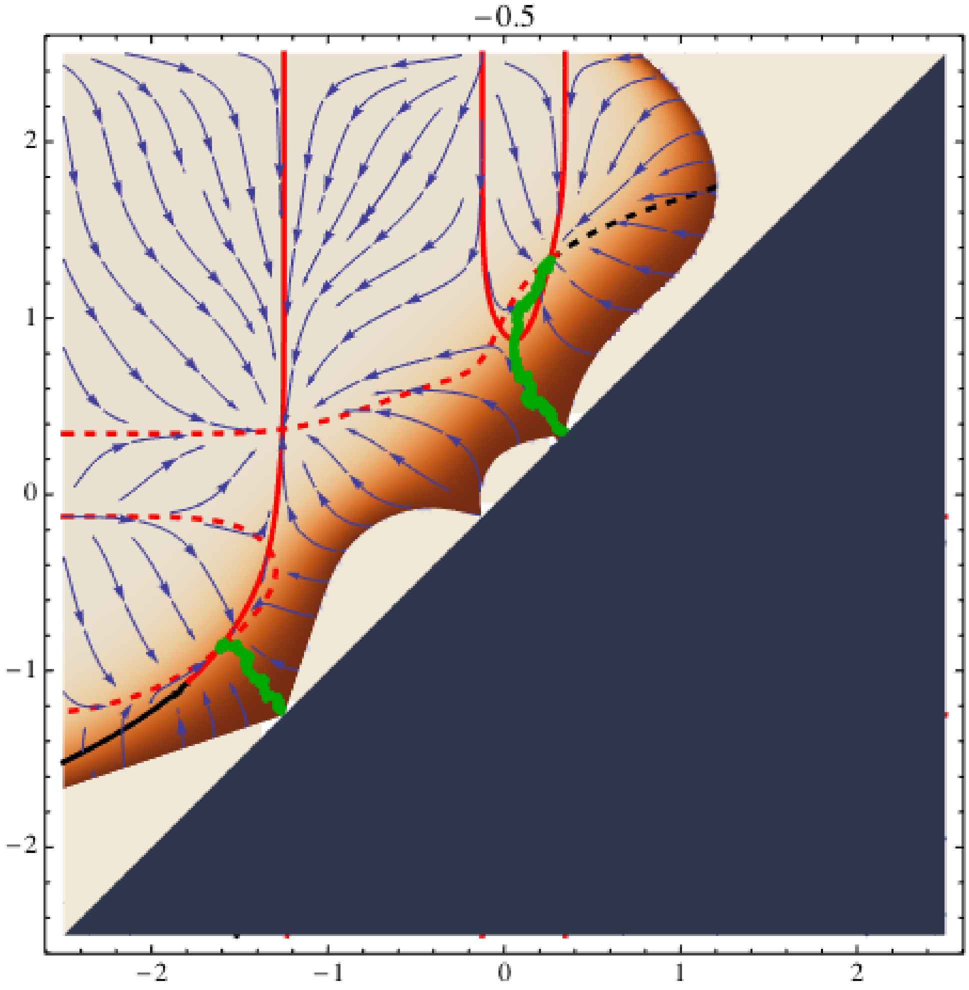
```
diff[{x1_, x2_}] :=
  {1/2 μ θ3 n1[x1, x2] Abs[grad1[x1, x2]], 1/2 μ θ3 n2[x1, x2] Abs[grad2[x1, x2]]};
```

- **Stochastic orbits (Euler method)**

```
β = -.5;

x0 = {.343690 - .02, .343691 + .02};(* starting point orbit *)
t0 = 0;(* start time *)
t∞ = 300 000;(* stop time *)
Δt = 10;(* integration time step *)
no = 1;(* number of orbits from the same starting point *)

data = {};
For[i = 1, i ≤ no, i++,
  x = x0;
  t = t0;
  While[t ≤ t∞,
   data = Join[data, {x}];
   z = RandomReal[NormalDistribution[0, 1], 2];
   x = x + Δt drift[x] + z Sqrt[Δt diff[x]];
   t = t + Δt;
  ];
 ];

SDEorbit1 =
  ListPlot[data, Joined → False, PlotStyle → {Darker[Green], Thickness[0.001]}];

Clear[β];
```

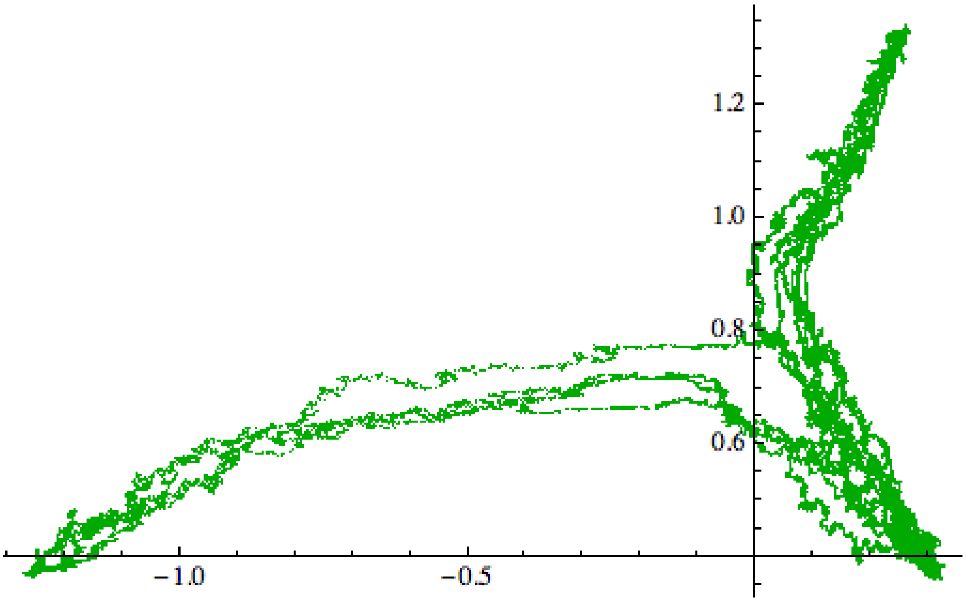
```
β = -.5;

x0 = {-1.25001 - .02, -1.24999 + .02};(* starting point orbit *)
t0 = 0;(* start time *)
t∞ = 300 000;(* stop time *)
Δt = 10;(* integration time step *)
no = 1;(* number of orbits from the same starting point *)

data = {};
For[i = 1, i ≤ no, i++,
  x = x0;
  t = t0;
  While[t ≤ t∞,
   data = Join[data, {x}];
   z = RandomReal[NormalDistribution[0, 1], 2];
   x = x + Δt drift[x] + z Sqrt[Δt diff[x]];
   t = t + Δt;
  ];
 ];

SDEorbit2 =
  ListPlot[data, Joined → False, PlotStyle → {Darker[Green], Thickness[0.001]}];

Clear[β];
```

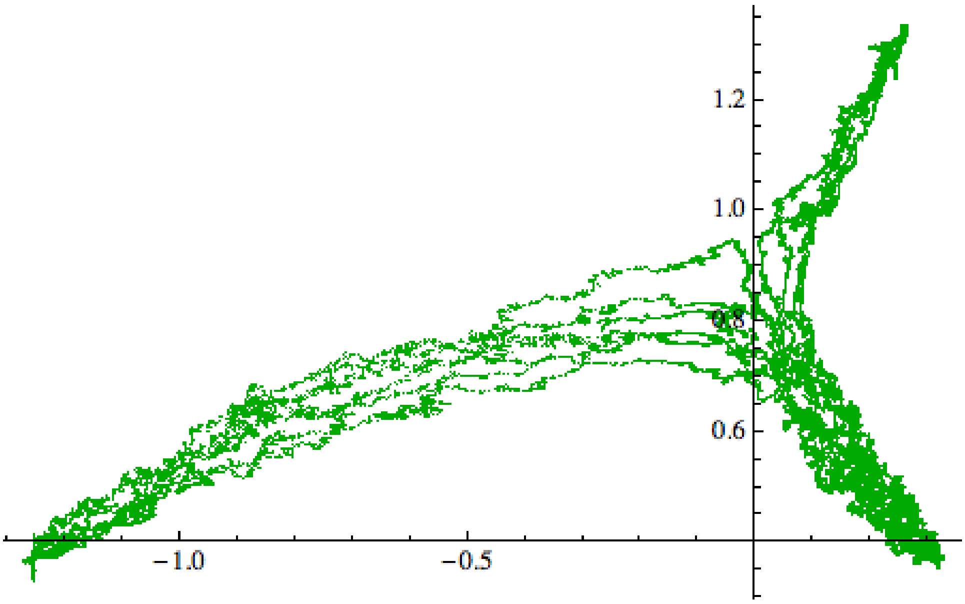
```
β = -.5;

Show[
 MIP[β], IPes1[β], IPbp1[β], IPes2[β], IPbp2[β],
 stream[β], antiMIP[β], HLT[β],
 SDEorbit1, SDEorbit2,
 PlotLabel → β
]

Clear[β];
```



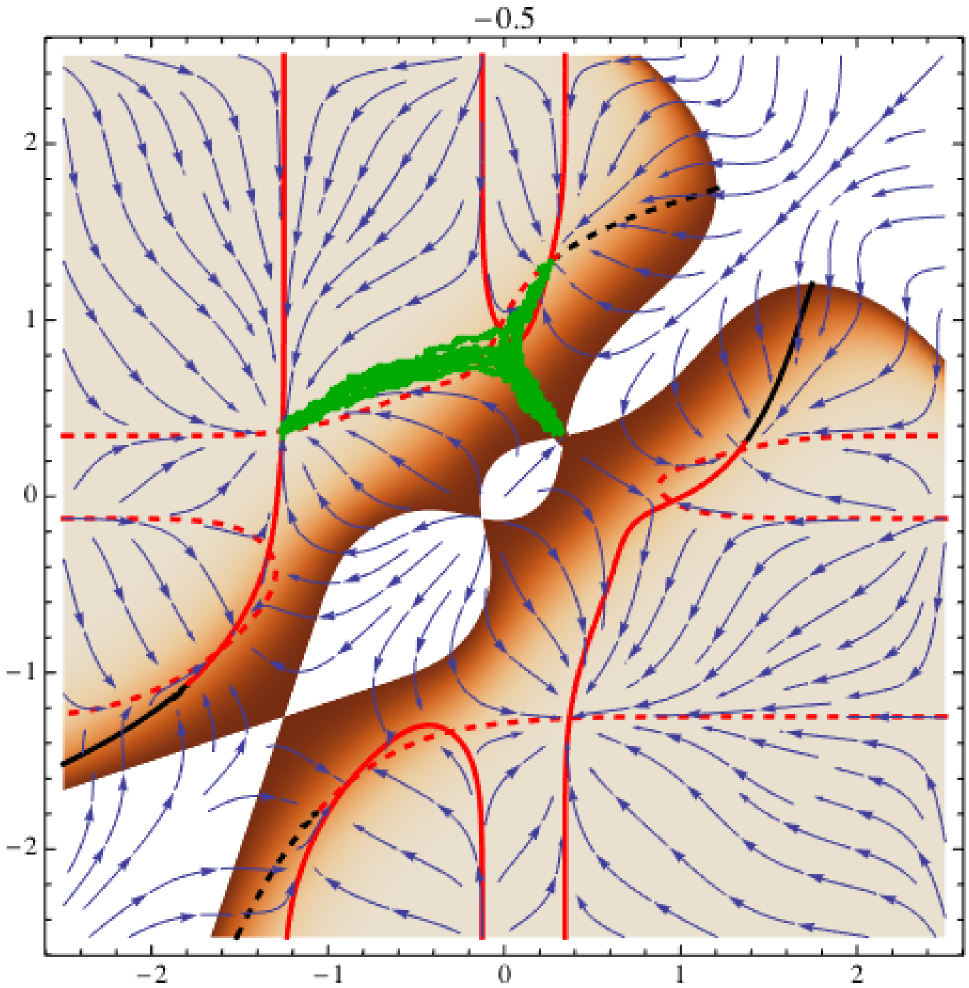
▪ When stochasticity matters . . .

```
β = -.5;

x0 = {.343690 - .03, .343691 + .03};(* starting point orbit *)
t0 = 0;(* start time *)
t∞ = 200000;(* stop time *)
Δt = 20;(* integration time step *)
no = 10;(* number of orbits from the same starting point *)

data = Table[{}, {i, 1, no}];
For[i = 1, i ≤ no, i++,
  x = x0;
  t = t0;
  While[t ≤ t∞ ∧ 0 < n1[x[[1]], x[[2]]] ∧ 0 < n2[x[[1]], x[[2]]],
   data[[i]] = Join[data[[i]], {x}];
   z = RandomReal[NormalDistribution[0, 1], 2];
   x = x + Δt drift[x] + z √(Δt diff[x]);
   t = t + Δt;
  ];
 ];

Clear[β];

BinCounts[Sort[data[[All, -1]]]]

{{4, 0}, {0, 0}, {0, 6}}

SDEorbit3 =
 ListPlot[Flatten[data, 1], Joined → False, PlotStyle → {Darker[Green], PointSize[.001]}]
```






```
β = -.5;

SDEorbit3 = ListPlot[Flatten[data, 1],
   Joined → False, PlotStyle → {Darker[Green], PointSize[.001]}];

Show[
 MIP[β], IPes1[β], IPbp1[β], IPes2[β], IPbp2[β],
 stream[β],
  ListPlot[Flatten[data, 1], Joined → False, PlotStyle → {Darker[Green], PointSize[.001]}],
 PlotLabel → β
 ]
```


−0.5
2
1
0
−1
−2
−2
−1
0
1
2