## Prey-predator coevolution

**Strategy**

x, y are, respectively, prey and predator log-body mass.

**Full population dynamics**

$$\frac{d}{dt} m_i = r[x_i]\, m_i \left(1 - \frac{\sum_{j=1}^{k} a[x_i, x_j]\, m_j}{K[x_i]}\right) - m_i \sum_{j=1}^{l} b[x_i, y_j]\, n_j \quad (i = 1, \ldots, k) \quad \text{(prey)}$$

$$\frac{d}{dt} n_i = n_i \sum_{j=1}^{k} c[x_j, y_i]\, b[x_j, y_i]\, m_j - d[y_i]\, n_i \quad (i = 1, \ldots, l) \quad \text{(predator)}$$

**Idem, but somewhat rewritten**

$$\frac{d}{dt} \mathrm{Log}[m_i] = r[x_i] \left(1 - \frac{\sum_{j=1}^{k} a[x_i, x_j]\, m_j}{K[x_i]}\right) - \sum_{j=1}^{l} b[x_i, y_j]\, n_j \quad (i = 1, \ldots, k) \quad \text{(prey)}$$

$$\frac{d}{dt} \mathrm{Log}[n_i] = \sum_{j=1}^{k} c[x_j, y_i]\, b[x_j, y_i]\, m_j - d[y_i] \quad (i = 1, \ldots, l) \quad \text{(predator)}$$

**Invader dynamics**

$$\frac{d}{dt} \mathrm{Log}[m_i^*] = r[x_i^*] \left(1 - \frac{\sum_{j=1}^{k} a[x_i^*, x_j]\, m_j}{K[x_i^*]}\right) - \sum_{j=1}^{l} b[x_i^*, y_j]\, n_j \quad (i = 1, \ldots, k) \quad \text{(prey mutant)}$$

$$\frac{d}{dt} \mathrm{Log}[n_i^*] = \sum_{j=1}^{k} c[x_j, y_i^*]\, b[x_j, y_i^*]\, m_j - d[y_i^*] \quad (i = 1, \ldots, l) \quad \text{(predator mutant)}$$

**Invasion fitness**

$$\left\langle \frac{d}{dt} \mathrm{Log}[m_i^*] \right\rangle = r[x_i^*] \left(1 - \frac{\sum_{j=1}^{k} a[x_i^*, x_j]\, \langle m_j \rangle}{K[x_i^*]}\right) - \sum_{j=1}^{l} b[x_i^*, y_j]\, \langle n_j \rangle \quad (i = 1, \ldots, k) \quad \text{(prey mutant)}$$

$$\left\langle \frac{d}{dt} \mathrm{Log}[n_i^*] \right\rangle = \sum_{j=1}^{k} c[x_j, y_i^*]\, b[x_j, y_i^*]\, \langle m_j \rangle - d[y_i^*] \quad (i = 1, \ldots, l) \quad \text{(predator mutant)}$$

### MONOMORPHIC PREY POPULATION and MONOMORPHIC PREDATOR POPULATION

**Population equations**

m = prey pop dens; n = pred pop dens; x = prey log body size; y = pred log body size

```
dLogm[m_, n_] := r[x] (1 - a[x, x] m / k[x]) - b[x, y] n;

dLogn[m_, n_] := c[x, y] b[x, y] m - d[y];
```

**Population equilibrium**

```
Solve[{dLogm[m, n] == 0, dLogn[m, n] == 0}, {m, n}]
```

$$\left\{\left\{n \to \frac{r[x]}{b[x, y]} - \frac{a[x, x]\, d[y]\, r[x]}{b[x, y]^2\, c[x, y]\, k[x]},\ m \to \frac{d[y]}{b[x, y]\, c[x, y]}\right\}\right\}$$

```
m[{x_, y_}] := d[y] / (b[x, y] c[x, y]);

n[{x_, y_}] := r[x] / b[x, y] - a[x, x] d[y] r[x] / (b[x, y]^2 c[x, y] k[x]);
```

## Parameter values and functions

```
r[x_] := 1;

k[x_] := e^(-x^2);

a[x1_, x2_] := e^(-α (x1-x2)^2); α = 0.5;

b[x_, y_] := β1 e^(-β2 (x-y-p)^2); β1 = 100; β2 = 0.2; p = Log[0.5];

c[x_, y_] := γ e^(x-y); γ = 0.2;

d[y_] := δ1 e^(-δ2 y); δ1 = 1; δ2 = 1;
```

## Coexistence

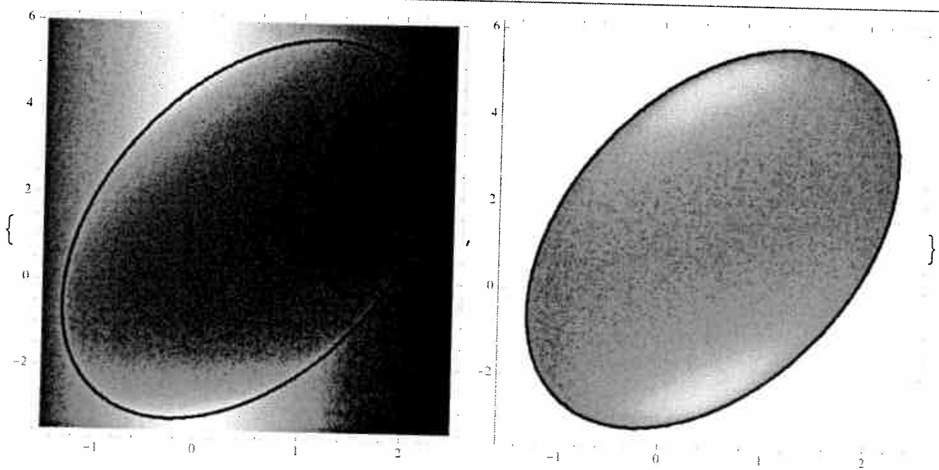
```
xmin = -1.5; xmax = 2.5; ymin = -3.5; ymax = 6;
```

```
coexBnd = ContourPlot[If[m[{x, y}] > 0 && n[{x, y}] > 0, 1, -1], {x, xmin, xmax},
    {y, ymin, ymax}, Contours → {0}, ContourStyle → {Black, Thick}, ContourShading → False,
    PlotPoints → 60];
```

```
preyInt = DensityPlot[If[m[{x, y}] > 0 && n[{x, y}] > 0, m[{x, y}], k[x]],
    {x, xmin, xmax}, {y, ymin, ymax}, ColorFunction → "AvocadoColors", PlotPoints → 60];
```

```
predInt = DensityPlot[If[m[{x, y}] > 0 && n[{x, y}] > 0, n[{x, y}]], {x, xmin, xmax},
    {y, ymin, ymax}, ColorFunction → "SolarColors", PlotPoints → 60];
```

```
{Show[preyInt, coexBnd], Show[predInt, coexBnd]}
```




## Invasion fitness

### Prey invasion fitness and its derivatives

```
sPrey_mo[{x_, y_}, X_] := r[X] (1 - a[X, x] m[{x, y}] / k[X]) - b[X, y] n[{x, y}];

dsPrey_mo[{x_, y_}] := ∂_X sPrey_mo[{x, y}, X] /. {X → x};

ddsPrey_mo[{x_, y_}] := ∂_{X,X} sPrey_mo[{x, y}, X] /. {X → x};
```

**Predator invasion fitness and its derivatives:**

```
sPredmo[{x_, y_}, Y_] := c[x, Y] b[x, Y] m[{x, y}] - d[Y];

dsPredmo[{x_, y_}] := ∂Y sPredmo[{x, y}, Y] /. {Y → y};

ddsPredmo[{x_, y_}] := ∂Y,Y sPredmo[{x, y}, Y] /. {Y → y};
```

## Isocline plot

**Prey isoclines**

*solid = evolutionarily stable; dashed = not evolutionarily stable*

```
preyES = ContourPlot[If[n[{x, y}] > 0 && ddsPreymo[{x, y}] ≤ 0, dsPreymo[{x, y}]],
    {x, xmin, xmax}, {y, ymin, ymax}, Contours → {0}, ContourShading → False,
    ContourStyle → {Black, Thick}, PlotPoints → 30];

preyNES = ContourPlot[If[n[{x, y}] > 0 && ddsPreymo[{x, y}] > 0, dsPreymo[{x, y}]],
    {x, xmin, xmax}, {y, ymin, ymax}, Contours → {0}, ContourShading → False,
    ContourStyle → {Black, Thick, Dashed}, PlotPoints → 30];
```

**Predator isoclines**

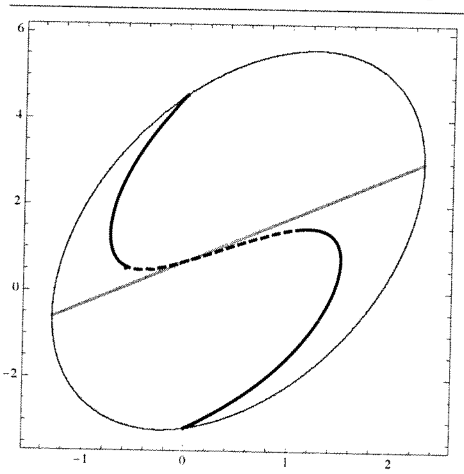
*solid = evolutionarily stable; dashed = not evolutionarily stable*

```
predES = ContourPlot[If[n[{x, y}] > 0 && ddsPredmo[{x, y}] ≤ 0, dsPredmo[{x, y}]],
    {x, xmin, xmax}, {y, ymin, ymax}, Contours → {0}, ContourShading → False,
    ContourStyle → {Red, Thick}, PlotPoints → 30];

predNES = ContourPlot[If[n[{x, y}] > 0 && ddsPredmo[{x, y}] > 0, dsPredmo[{x, y}]],
    {x, xmin, xmax}, {y, ymin, ymax}, Contours → {0}, ContourShading → False,
    ContourStyle → {Red, Thick, Dashed}, PlotPoints → 30];
```

**Isocline plot**

*(black = prey isocline; red = predator isocline; solid = ES; dashed = NES)*

```
Show[predES, predNES, preyES, preyNES, coexBnd]
```




## Canonical equation

Mutation rate and standard deviation of the mutant step distribution; only the relative prey/pred sizes matter

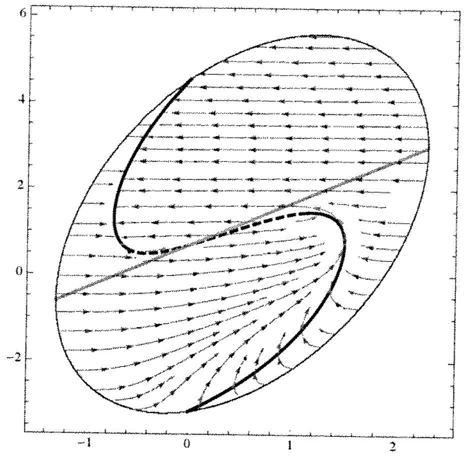
```
μPrey = 1;
σPrey = 1;
μPred = 1;
σPred = 1;
```

**Deterministic drift**

```
driftmo[{x_, y_}] :=
  {1/2 μPrey σPrey^2 m[{x, y}] dsPreymo[{x, y}], 1/2 μPred σPred^2 n[{x, y}] dsPredmo[{x, y}]};
```

Stream plot (CE)

```
CEstream = StreamPlot[If[n[{x, y}] > 0, drift_mo[{x, y}], {0, 0}], {x, xmin, xmax},
    {y, ymin, ymax}];

Show[preyES, preyNES, predES, predNES, coexBnd, CEstream]
```

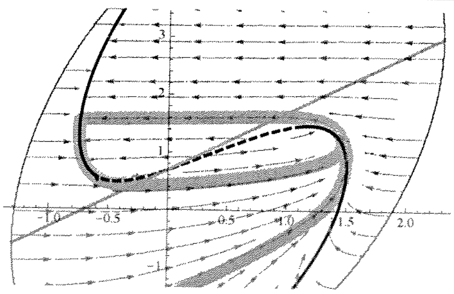



Deterministic orbit (Euler metod)

```
v0 = {-1, -2};(* starting point of orbit *)
t0 = 0;(* start time *)
t∞ = 20000;(* stop time *)
Δt = 1;(* integration time step *)

data = {};
v = v0;
t = t0;
While[t ≤ t∞ && n[v] > 0,
  data = Join[data, {Append[v, t]}];
  v = v + Δt drift_mo[v];
  t = t + Δt;
 ];
```

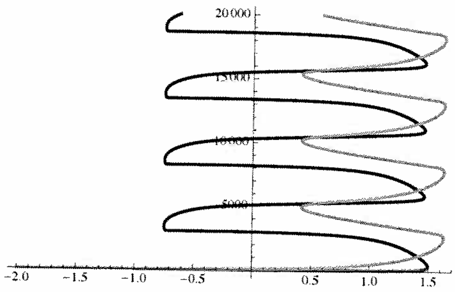
Projected orbit on the (x,y)-plane

```
CEorbit = ListPlot[data[[All, {1, 2}]], PlotStyle → {Gray, PointSize[0.03]},
    Joined → False];

Show[CEorbit, preyES, preyNES, predES, predNES, coexBnd, CEstream]
```



### Orbit as evolutionary tree

```
xtree = ListPlot[data[[All, {1, 3}]], PlotStyle → {Black, Thick}, Joined → True];
ytree = ListPlot[data[[All, {2, 3}]], PlotStyle → {Red, Thick}, Joined → True];
Show[xtree, ytree]
```




## Stability

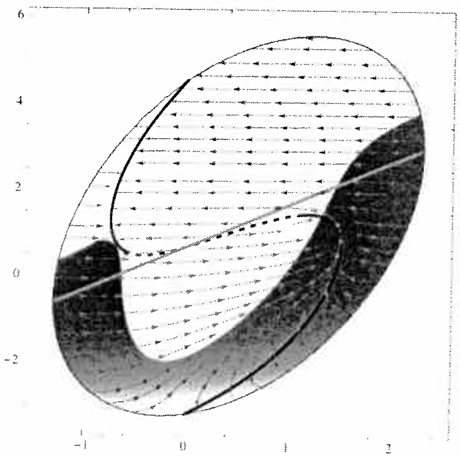
### Jacobi matrix for stability calculations

```
Jxx[{x_, y_}] := (∂X1,X2 sPreymo[{X1, y}, X2] + ∂X2,X2 sPreymo[{x, y}, X2]) /. {X1 → x, X2 → x};

Jxy[{x_, y_}] := ∂X,Y sPreymo[{x, Y}, X] /. {X → x, Y → y};

Jyx[{x_, y_}] := ∂X,Y sPredmo[{X, y}, Y] /. {X → x, Y → y};

Jyy[{x_, y_}] := (∂Y1,Y2 sPredmo[{x, Y1}, Y2] + ∂Y2,Y2 sPredmo[{x, y}, Y2]) /. {Y1 → y, Y2 → y};
```

### Total stability

```
totStab =
  DensityPlot[
   If[n[{x, y}] > 0 && Jxx[{x, y}] < 0 && Jyy[{x, y}] < 0 &&
     Jxx[{x, y}] Jyy[{x, y}] > Abs[Jxy[{x, y}] Jyx[{x, y}]],
    Abs[dsPreymo[{x, y}]] + Abs[dsPredmo[{x, y}]]], {x, xmin, xmax}, {y, ymin, ymax},
   ColorFunction → "FuchsiaTones", PlotPoints → 60];
```

```
Show[totStab, preyES, preyNES, predES, predNES, CEstream, coexBnd]
```

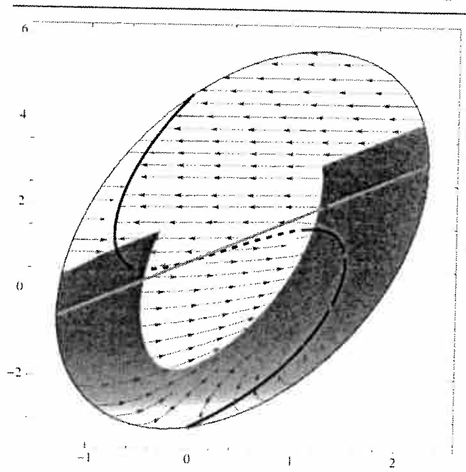



*Conclusion: the intersection of the isoclines is NOT ABSOLUTELY STABLE.*

**Strong stability**

```
strStab =
  DensityPlot[
   If[n[{x, y}] > 0 && Jxx[{x, y}] < 0 && Jyy[{x, y}] < 0 &&
     Jxx[{x, y}] Jyy[{x, y}] > Jxy[{x, y}] Jyx[{x, y}],
    Abs[dsPrey_mo[{x, y}]] + Abs[dsPred_mo[{x, y}]]], {x, xmin, xmax}, {y, ymin, ymax},
   ColorFunction → "CherryTones", PlotPoints → 60];
```

```
Show[strStab, preyES, preyNES, predES, predNES, CEstream, coexBnd]
```

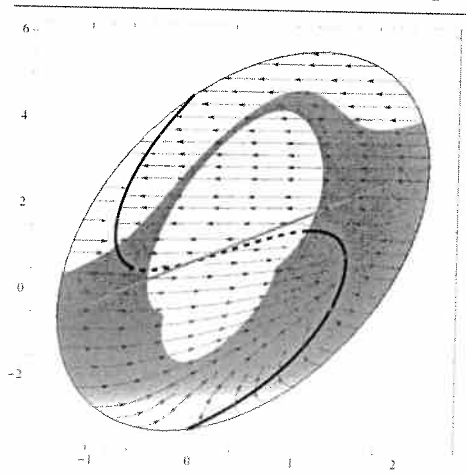



*Conclusion: the intersection of the isoclines is NOT STRONGLY STABLE.*

**Weak stability**

```
wkStab =
  DensityPlot[
   If[n[{x, y}] > 0 && μPrey σPrey^2 m[{x, y}] Jxx[{x, y}] + μPred σPred^2 n[{x, y}] Jyy[{x, y}] <
      0 && Jxx[{x, y}] Jyy[{x, y}] > Jxy[{x, y}] Jyx[{x, y}],
    Abs[dsPrey_mo[{x, y}]] + Abs[dsPred_mo[{x, y}]]], {x, xmin, xmax}, {y, ymin, ymax},
   ColorFunction → "CoffeeTones", PlotPoints → 60];
```

```
Show[wkStab, preyES, preyNES, predES, predNES, CEstream, coexBnd]
```




*Conclusion: the intersection of the isoclines is for the present mutation rates and mutation variances NOT WEAKLY STABLE*

The relative mutation rates and/or mutation standard deviations can affect weak stability, but not absolute and strong stability.

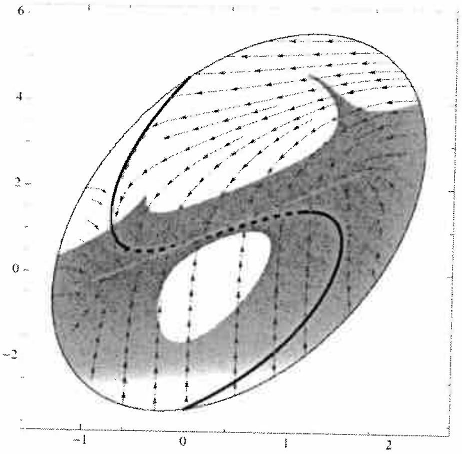
```
μPrey = 1;
σPrey = 0.1;
μPred = 1;
σPred = 1;
```

```
wkStab =
  DensityPlot[
   If[n[{x, y}] > 0 && μPrey σPrey² m[{x, y}] Jxx[{x, y}] + μPred σPred² n[{x, y}] Jyy[{x, y}] <
      0 && Jxx[{x, y}] Jyy[{x, y}] > Jxy[{x, y}] Jyx[{x, y}],
    Abs[dsPreymo[{x, y}]] + Abs[dsPredmo[{x, y}]]], {x, xmin, xmax}, {y, ymin, ymax},
   ColorFunction → "CoffeeTones", PlotPoints → 60];
```

```
CEstream = StreamPlot[If[n[{x, y}] > 0, driftmo[{x, y}], {0, 0}], {x, xmin, xmax},
   {y, ymin, ymax}];
```

Weak stability for the new mutation parameter values

```
Show[wkStab, preyES, preyNES, predES, predNES, CEstream, coexBnd]
```




*Conclusion: for the new mutation rates and mutation variances, the intersection of the isoclines is WEAKLY STABLE.*

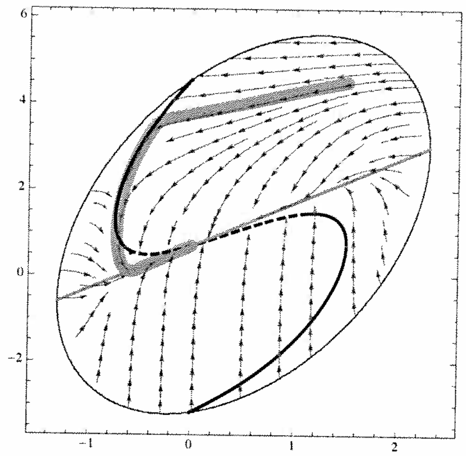
Deterministic orbit (CE)

```
v0 = {1.5, 4.5}; (* starting point of orbit *)
t0 = 0; (* start time *)
t∞ = 200000; (* stop time *)
Δt = 50; (* integration time step *)

data = {};
v = v0;
t = t0;
While[t ≤ t∞ && n[v] > 0,
  data = Join[data, {Append[v, t]}];
  v = v + Δt driftmo[v];
  t = t + Δt;
 ];
```

Projected orbit on the (x,y)-plane

```
CEorbit = ListPlot[data[[All, {1, 2}]], PlotStyle → {Gray, PointSize[0.03]},
   Joined → False];
```

```
Show[coexBnd, CEorbit, preyES, preyNES, predES, predNES, CEstream]
```



# Evolutionary branching in the prey?

Notice that the prey-isocline near the intersection is not evolutionarily stable (dashed isocline). Does that mean that the prey strategy will branch?

## Pairwise invadability plot (PIP) for the prey in the neighborhood of the intersection of the prey-predator isoclines

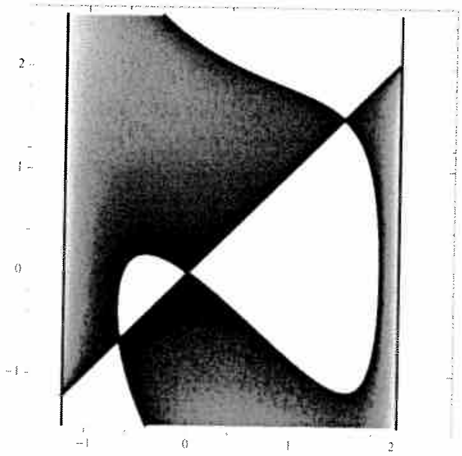
```
PIPbnd = ContourPlot[If[n[{x, 0.69}] > 0, sPreymo[{x, 0.69}, X]], {x, xmin, xmax},
   {X, xmin, xmax}, Contours → {0}, ContourStyle → {Black, Thick}, ContourShading → False,
   PlotPoints → 60];
```

```
nPos = ContourPlot[n[{x, 0.69}], {x, xmin, xmax}, {X, xmin, xmax}, Contours → {0},
   ContourStyle → {Black, Thick}, ContourShading → False, PlotPoints → 60];
```

```
PIPint = DensityPlot[If[n[{x, 0.69}] > 0 && sPreymo[{x, 0.69}, X] > 0, sPreymo[{x, 0.69}, X]],
   {x, xmin, xmax}, {X, xmin, xmax}, ColorFunction → "AvocadoColors", PlotPoints → 60];
```

Pairwise invadability plot (PIP) for fixed predator trait

```
Show[PIPint, PIPbnd, nPos]
```



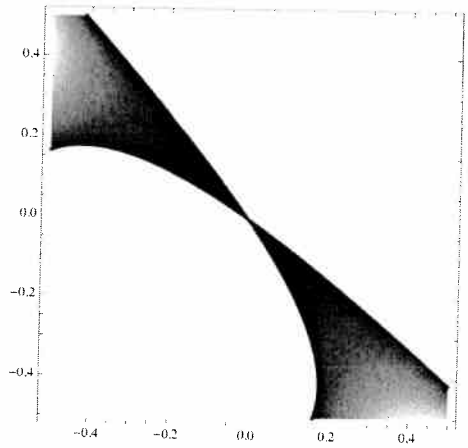
### Mutual invadability plot (MIP) for the prey in the neighborhood of the intersection of the prey-predator isoclines (detail)

```
MIPbnd = ContourPlot[If[Abs[x - X] > 0.001, sPreymo[{x, 0.69}, X] sPreymo[{X, 0.69}, x]],
    {x, -0.5, 0.5}, {X, -0.5, 0.5}, Contours → {0}, ContourStyle → {Black, Thick},
    ContourShading → False, PlotPoints → 60];
```

```
MIPint = DensityPlot[If[sPreymo[{x, 0.69}, X] > 0 && sPreymo[{X, 0.69}, x] > 0,
     sPreymo[{x, 0.69}, X] sPreymo[{X, 0.69}, x]], {x, -0.5, 0.5}, {X, -0.5, 0.5},
    ColorFunction → "AvocadoColors", PlotPoints → 60];
```

Mutual invadability plot (MIP) for fixed predator trait

```
Show[MIPint, MIPbnd]
```




Conclusion: evolutionary branching cannot yet be excluded. Since the cone of mutual invadability is so very narrow, the dimorphic prey orbit might not stay inside it. This we have to investigate further . . .

## DIMORPHIC PREY POPULATION and MONOMORPHIC PREDATOR POPULATION

### Reset

```
Clear[r, k, a, b, c, d];
```

### Population equations

m1, m2 = prey pop dens; nn = pred pop dens; x1, x2 = prey log body size; y = pred log body size

```
dLogm1[m1_, m2_, nn_] := r[x1] (1 - (a[x1, x1] m1 + a[x1, x2] m2)/k[x1]) - b[x1, y] nn;

dLogm2[m1_, m2_, nn_] := r[x2] (1 - (a[x2, x1] m1 + a[x2, x2] m2)/k[x2]) - b[x2, y] nn;

dLognn[m1_, m2_, nn_] := c[x1, y] b[x1, y] m1 + c[x2, y] b[x2, y] m2 - d[y];
```

## Population equilibrium

```
Solve[{dLogm1[m1, m2, nn] == 0, dLogm2[m1, m2, nn] == 0, dLognn[m1, m2, nn] == 0},
  {m1, m2, nn}]
```

```
{{nn → -(-a[x1, x2] a[x2, x1] d[y] r[x1] r[x2] + a[x1, x1] a[x2, x2] d[y] r[x1] r[x2] -
      a[x2, x2] b[x1, y] c[x1, y] k[x1] r[x1] r[x2] + a[x2, x1] b[x2, y] c[x2, y] k[x1] r[x1] r[x2] +
      a[x1, x2] b[x1, y] c[x1, y] k[x2] r[x1] r[x2] - a[x1, x1] b[x2, y] c[x2, y] k[x2] r[x1] r[x2]) /
    (-a[x1, x2] b[x1, y] b[x2, y] c[x1, y] k[x2] r[x1] + a[x1, x1] b[x2, y]^2 c[x2, y] k[x2] r[x1] +
      a[x2, x2] b[x1, y]^2 c[x1, y] k[x1] r[x2] - a[x2, x1] b[x1, y] b[x2, y] c[x2, y] k[x1] r[x2]),
  m1 → -(a[x1, x2] b[x2, y] d[y] k[x2] r[x1] - b[x2, y]^2 c[x2, y] k[x1] k[x2] r[x1] -
      a[x2, x2] b[x1, y] d[y] k[x1] r[x2] + b[x1, y] b[x2, y] c[x2, y] k[x1] k[x2] r[x2]) /
    (-a[x1, x2] b[x1, y] b[x2, y] c[x1, y] k[x2] r[x1] + a[x1, x1] b[x2, y]^2 c[x2, y] k[x2] r[x1] +
      a[x2, x2] b[x1, y]^2 c[x1, y] k[x1] r[x2] - a[x2, x1] b[x1, y] b[x2, y] c[x2, y] k[x1] r[x2]),
  m2 → -(b[x1, y] c[x1, y] (b[x2, y] r[x1] - b[x1, y] r[x2]) + d[y] (-a[x1,x1] b[x2,y] r[x1]/k[x1] + a[x2,x1] b[x1,y] r[x2]/k[x2])) /
    (-b[x2, y] c[x2, y] (-a[x1,x1] b[x2,y] r[x1]/k[x1] + a[x2,x1] b[x1,y] r[x2]/k[x2]) + b[x1, y] c[x1, y] (-a[x1,x2] b[x2,y] r[x1]/k[x1] + a[x2,x2] b[x1,y] r[x2]/k[x2]))}}
```

```
m1[{x1_, x2_, y_}] :=
  -(a[x1, x2] b[x2, y] d[y] k[x2] r[x1] - b[x2, y]^2 c[x2, y] k[x1] k[x2] r[x1] -
      a[x2, x2] b[x1, y] d[y] k[x1] r[x2] + b[x1, y] b[x2, y] c[x2, y] k[x1] k[x2] r[x2]) /
   (-a[x1, x2] b[x1, y] b[x2, y] c[x1, y] k[x2] r[x1] +
      a[x1, x1] b[x2, y]^2 c[x2, y] k[x2] r[x1] + a[x2, x2] b[x1, y]^2 c[x1, y] k[x1] r[x2] -
      a[x2, x1] b[x1, y] b[x2, y] c[x2, y] k[x1] r[x2]);
m2[{x1_, x2_, y_}] := m1[{x2, x1, y}];
nn[{x1_, x2_, y_}] :=
  -(-a[x1, x2] a[x2, x1] d[y] r[x1] r[x2] + a[x1, x1] a[x2, x2] d[y] r[x1] r[x2] -
      a[x2, x2] b[x1, y] c[x1, y] k[x1] r[x1] r[x2] +
      a[x2, x1] b[x2, y] c[x2, y] k[x1] r[x1] r[x2] +
      a[x1, x2] b[x1, y] c[x1, y] k[x2] r[x1] r[x2] -
      a[x1, x1] b[x2, y] c[x2, y] k[x2] r[x1] r[x2]) /
   (-a[x1, x2] b[x1, y] b[x2, y] c[x1, y] k[x2] r[x1] +
      a[x1, x1] b[x2, y]^2 c[x2, y] k[x2] r[x1] + a[x2, x2] b[x1, y]^2 c[x1, y] k[x1] r[x2] -
      a[x2, x1] b[x1, y] b[x2, y] c[x2, y] k[x1] r[x2]);
```

## Parameter values and functions

```
r[x_] := 1;
```

$k[x\_] := e^{-x^2};$

$a[x1\_, x2\_] := e^{-\alpha (x1-x2)^2};\ \alpha = 0.5;$

$b[x\_, y\_] := \beta1\, e^{-\beta2\, (x-y-p)^2};\ \beta1 = 100;\ \beta2 = 0.2;\ p = \mathrm{Log}[0.5];$

$c[x\_, y\_] := \gamma\, e^{x-y};\ \gamma = 0.2;$

$d[y\_] := \delta1\, e^{-\delta2\, y};\ \delta1 = 1;\ \delta2 = 1;$

## Invasion fitness

### Prey invasion fitness and its derivatives

```
sPreydi[{x1_, x2_, y_}, X_] :=
  r[X] (1 - (a[X, x1] m1[{x1, x2, y}] + a[X, x2] m2[{x1, x2, y}]) / k[X]) - b[X, y] nn[{x1, x2, y}];

d1sPreydi[{x1_, x2_, y_}] := ∂X sPreydi[{x1, x2, y}, X] /. {X → x1};
d2sPreydi[{x1_, x2_, y_}] := ∂X sPreydi[{x1, x2, y}, X] /. {X → x2};
dd1sPreydi[{x1_, x2_, y_}] := ∂X,X sPreydi[{x1, x2, y}, X] /. {X → x1};
dd2sPreydi[{x1_, x2_, y_}] := ∂X,X sPreydi[{x1, x2, y}, X] /. {X → x2};
```

### Predator invasion fitness and its derivatives:

```
sPreddi[{x1_, x2_, y_}, Y_] :=
  c[x1, Y] b[x1, Y] m1[{x1, x2, y}] + c[x2, Y] b[x2, Y] m2[{x1, x2, y}] - d[Y];

dsPreddi[{x1_, x2_, y_}] := ∂Y sPreddi[{x1, x2, y}, Y] /. {Y → y};
ddsPreddi[{x1_, x2_, y_}] := ∂Y,Y sPreddi[{x1, x2, y}, Y] /. {Y → y};
```

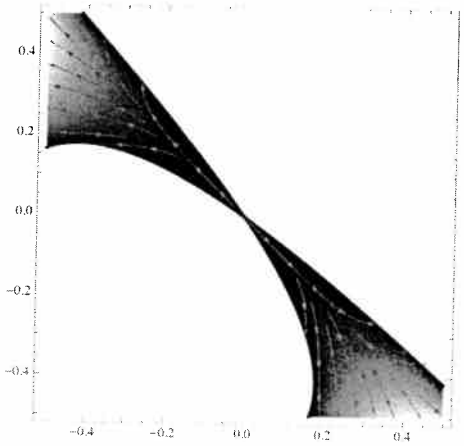
## Canonical equation

### Deterministic drift

```
driftdi[{x1_, x2_, y_}] :=
  {1/2 μPrey σPrey^2 m1[{x1, x2, y}] d1sPreydi[{x1, x2, y}],
   1/2 μPrey σPrey^2 m2[{x1, x2, y}] d2sPreydi[{x1, x2, y}],
   1/2 μPred σPred^2 nn[{x1, x2, y}] dsPreddi[{x1, x2, y}]};
```

Stream plot (note below that the boundaries of the mutual invadability cone are attracting, and so the cone is not forward invarant under the CE)

```
CEstream =
  StreamPlot[If[m1[{x1, x2, 0.69}] > 0 && m2[{x1, x2, 0.69}] > 0 && Abs[x1 - x2] > 0.01,
    driftdi[{x1, x2, 0.69}][[1 ;; 2]], {0, 0}], {x1, -0.5, 0.5}, {x2, -0.5, 0.5}];

Show[MIPint, MIPbnd, CEstream]
```

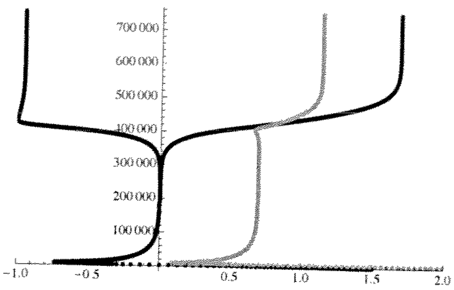


## Deterministic evolutionary tree

```
v0 = {1.5, 4.5};(* starting point of orbit *)
t0 = 0;(* start time *)
t∞ = 750000;(* stop time *)
Δt = 50;(* integration time step *)

(* positional date *)
xdata = {};
ydata = {};

(* stability data ES/NES *)
xstab = {};
ystab = {};

(* monomorphic part of the tree; note the extra stopping condition *)
v = v0;
t = t0;
While[t ≤ t∞ && n[v] > 0 &&
   dsPrey_mo[v - σPrey {0.001, 0}] dsPrey_mo[v + σPrey {0.001, 0}] > 0,
  xdata = Join[xdata, {{v[[1]], t}}];
  ydata = Join[ydata, {{v[[2]], t}}];
  xstab = Join[xstab, {{ddsPrey_mo[v], t}}];
  ystab = Join[ystab, {{ddsPred_mo[v], t}}];
  v = v + Δt drift_mo[v];
  t = t + Δt;
 ];



(* dimorphic part of the tree *)
v = {v[[1]] - 0.01 σPrey, v[[1]] + 0.01 σPrey, v[[2]]};
While[t ≤ t∞ && m1[v] > 0 && m2[v] > 0 && nn[v] > 0,
  xdata = Join[xdata, {{v[[1]], t}, {v[[2]], t}}];
  ydata = Join[ydata, {{v[[3]], t}}];
  xstab = Join[xstab, {{dd1sPrey_di[v], t}, {dd2sPrey_di[v], t}}];
  ystab = Join[ystab, {{ddsPred_di[v], t}}];
  v = v + Δt drift_di[v];
  t = t + Δt;
 ];
```

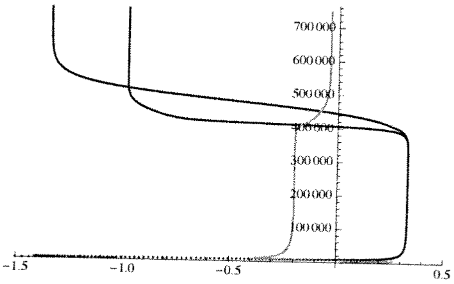
Trait values

```
xtree = ListPlot[xdata, PlotStyle → {Black}, Joined → False];

ytree = ListPlot[ydata, PlotStyle → {Red}, Joined → False];

Show[xtree, ytree, PlotRange → {{-1, 2}, All}]
```



**Stability data**

```
xstree = ListPlot[xstab, PlotStyle → {Black, PointSize[0.002]}, Joined → False];

ystree = ListPlot[ystab, PlotStyle → {Red, PointSize[0.002]}, Joined → False];

Show[xstree, ystree, PlotRange → {{-1.5, 0.5}, All}]
```




*Conclusion: after branching in the pray species, the population evoves to an evolutionarily stable point with two pray types and one predator type. However, as themutual invadability cone is not forward invariant (see above) stochastic orbits may readily leave the cone so that branching may fail. For sufficiently small mutation step sizes, however, there is a* **positive probability of not leaving the cone,** *an so there is a positive probability of branching. Since after failed branching the population will in the neighborhood of the singular point, branching will occur sooner or later with probability one. The time till successful branching should be exponentially distributed and may be very long. This phenomenon has been dubbed "evolutionary loitering" (pc. Eva Kisdi).*