Example: The Lotka - Volterra cannibalism time budget model

Resident dynamics:

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

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

$(i = 1, \; ..., \; l)$

MONOMORPHIC RESIDENT POPULATION

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 == r - r R / k - α (1 - x) n,
   0 == ϵ α (1 - x) R - δ + γ β[x] x (1 - x) n - (1 - x) β[x] x n},
  {R, n}]
```

```
{{R → -(k α δ + k r x β[x] - k r x γ β[x]) / (-k α^2 ϵ + k x α^2 ϵ - r x β[x] + r x γ β[x]),
  n → -(r δ - k r α ϵ + k r x α ϵ) / ((-1 + x) (-k α^2 ϵ + k x α^2 ϵ - r x β[x] + r x γ β[x]))}}
```

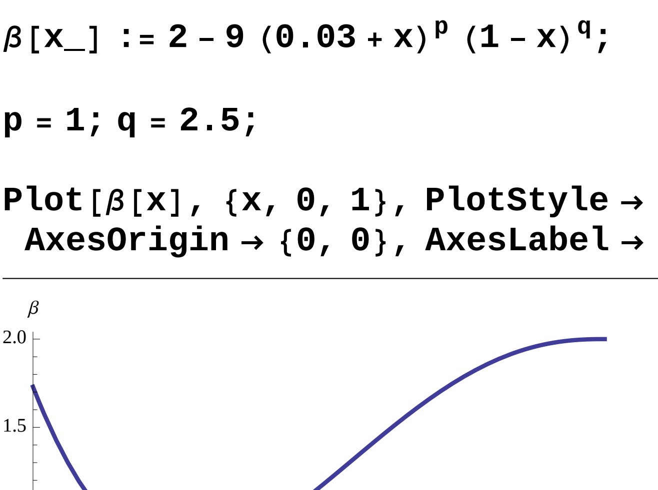
```
R[x_] := -(k (α δ - r x (-1 + γ) β[x])) / (k (-1 + x) α^2 ϵ + r x (-1 + γ) β[x]);

n[x_] := -(r (δ + k (-1 + x) α ϵ)) / ((-1 + x) (k (-1 + x) α^2 ϵ + r x (-1 + γ) β[x]));
```

Invasion fitness and its derivatives:

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

dsmo[x_] := (∂y smo[x, y]) /. {y → x};
ddsmo[x_] := (∂y ∂y smo[x, y]) /. {y → x};
```

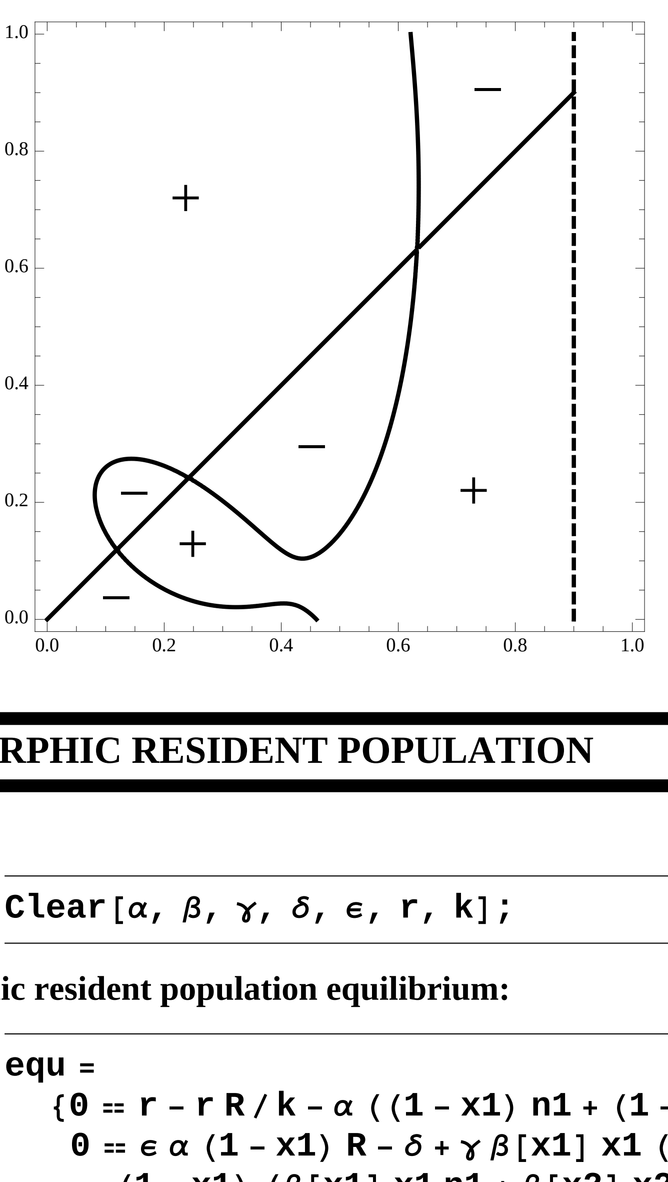
Parameter values and functions:

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

β[x_] := 2 - 9 (0.03 + x)^p (1 - x)^q;

p = 1; q = 2.5;

Plot[β[x], {x, 0, 1}, PlotStyle → {Thick},
 AxesOrigin → {0, 0}, AxesLabel → {x, β}]
```



Pairwise invadability plot (PIP):

```
ContourPlot[{n[x], If[n[x] > 0, smo[x, y]]}, {x, 0, 1},
 {y, 0, 1}, Contours → {0},
 ContourStyle → {{Black, Thick, Dashed}, {Black, Thick}},
 PlotPoints → 100]
```



DIMORPHIC RESIDENT POPULATION

Reset:

```
Clear[α, β, γ, δ, ϵ, r, k];
```

Dimorphic resident population equilibrium:

```
equ =
  {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] 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[equ, 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) α γ ϵ)) β[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) β[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) 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 (-1 + x2) α γ ϵ)) β[x2]) /
   (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 + x1) α ϵ) β[x2])) /
   (r γ (x1 (-1 + x2) β[x1] - (-1 + x1) x2 β[x2])^2);
```

Dimorphic invasion fitness and derivatives:

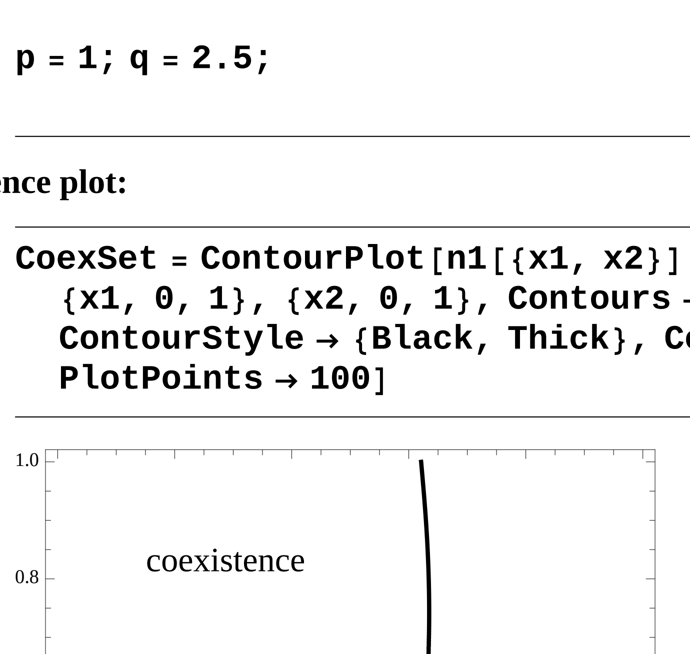
```
sdi[{x1_, x2_}, y_] :=
  ϵ α (1 - y) R[{x1,x2}] - δ +
   γ β[y] y ((1 - x1) n1[{x1, x2}] + (1 - x2) n2[{x1, x2}]) -
   (1 - y) (β[x1] x1 n1[{x1, x2}] + β[x2] x2 n2[{x1, x2}]);

x1dsdi[{x1_, x2_}] := (∂y sdi[{x1, x2}, y]) /. {y → x1};
x2dsdi[{x1_, x2_}] := (∂y sdi[{x1, x2}, y]) /. {y → x2};
x1ddsdi[{x1_, x2_}] := (∂y ∂y sdi[{x1, x2}, y]) /. {y → x1};
x2ddsdi[{x1_, x2_}] := (∂y ∂y sdi[{x1, x2}, y]) /. {y → x2};
```

Default parameter values and functions :

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

β[x_] := 2 - 9 (0.03 + x)^p (1 - x)^q;

p = 1; q = 2.5;
```

Coexistence plot:

```
CoexSet = ContourPlot[n1[{x1, x2}] n2[{x1, x2}],
  {x1, 0, 1}, {x2, 0, 1}, Contours → {0},
  ContourStyle → {Black, Thick}, ContourShading → False,
  PlotPoints → 100]
```



Isocline plot:

(solid = x1-isocline; dashed = x2-isocline; black = evolutionarily stable; red = not evolutionarily stable)

```
iso1ES = ContourPlot[
   If[n1[{x1, x2}] > 0 && n2[{x1, x2}] > 0 &&
     x1ddsdi[{x1, x2}] ≤ 0, x1dsdi[{x1, x2}]],
   {x1, 0, 1}, {x2, 0, 1}, Contours → {0},
   ContourShading → False, ContourStyle → {Black, Thick},
   PlotPoints → 30];

iso1NES =
  ContourPlot[
   If[n1[{x1, x2}] > 0 && n2[{x1, x2}] > 0 &&
     x1ddsdi[{x1, x2}] > 0, x1dsdi[{x1, x2}]],
   {x1, 0, 1}, {x2, 0, 1}, Contours → {0},
   ContourShading → False, ContourStyle → {Red, Thick},
   PlotPoints → 30];

iso2ES = ContourPlot[
   If[n1[{x1, x2}] > 0 && n2[{x1, x2}] > 0 &&
     x2ddsdi[{x1, x2}] ≤ 0, x2dsdi[{x1, x2}]],
   {x1, 0, 1}, {x2, 0, 1}, Contours → {0},
   ContourShading → False,
   ContourStyle → {Black, Thick, Dashed},
   PlotPoints → 30];

iso2NES =
  ContourPlot[
   If[n1[{x1, x2}] > 0 && n2[{x1, x2}] > 0 &&
     x2ddsdi[{x1, x2}] > 0, x2dsdi[{x1, x2}]],
   {x1, 0, 1}, {x2, 0, 1}, Contours → {0},
   ContourShading → False,
   ContourStyle → {Red, Thick, Dashed}, PlotPoints → 30];

Show[CoexSet, iso1ES, iso1NES, iso2ES, iso2NES]
```