```
Clear["Global`*"];
```

```
Quit[]
```

```
LaunchKernels[]
```

Figures Bistab eps / sigma

```
params[ss_, mm_] = {c → 0.05, mu → 1, β0 → 10, k → 0.01, sig → ss, eps → mm, rm → 2};

β[a_] = β0 a / (1 + a); (* parasite trade-off*)

rS[ga_] = rm / (1 + c ga) ;(* susceptible hosts trade-off*)

rI[ga_] = eps rS[ga];(*infected hosts trade-off*)

getres[aa_, gg_] =
  Solve[{0 == (rS[gg] x + rI[gg] y) (1 - k (x + y)) - (mu + (β[aa] y)) x + gg y,
      0 == β[aa] x y - (mu + aa + gg) y}, {x, y}][[4]] //
   Simplify;(* epidemiological system*)

para[a_, ga_] = Block[{}, res = getres[a, ga];
   Seq = x /. res;
   Ieq = y /. res;
   β'[a] - β[a] / (mu + a + ga + sig β[a] Ieq)];(* parasite selection gradient*)

hote[a_, ga_] = Block[{}, res = getres[a, ga];
    Seq = x /. res;
    Ieq = y /. res;
    rS'[ga] (1 - k (Seq + Ieq)) + (rI'[ga] (1 - k (Seq + Ieq)) β[a] Ieq) / (mu + a + ga) +
     (rS[ga] (1 - k (Seq + Ieq)) - mu) / (mu + a + ga)] //
   Simplify;(*host selection gradient*)
getcoess[sig_, eps_] := Solve[{0 == para[a, ga], 0 == hote[a, ga]} /.
   params[sig, eps], {a, ga}]
```

```
(*list=Table[{"sig = "<>ToString[sig],"eps = "<>ToString[eps],
    getcoess[sig,eps]},{sig,0,0.5,0.1},{eps,0.8,1.3,0.1}];*)
```

```
(*list=Table[{sig,eps,getcoess[sig,eps]},{sig,0,0.5,0.1},{eps,0.8,0.9,0.1}];*)
```

```
list = Table[{sig, eps, getcoess[sig, eps]},
   {sig, 0.0, 0.5, 0.005}, {eps, 0.8, 1.3, 0.005}];
```

```
list = ParallelTable[{sig, eps, getcoess[sig, eps]},
   {sig, 0.0, 0.5, 0.001}, {eps, 0.8, 1.3, 0.001}];
```

```
dataAll = {};
dataAll2 = {};
For[i = 1, i ≤ Dimensions[list][[2]], i++,
 For[j = 1, j ≤ Dimensions[list][[1]], j++,
  mysol = Select[{a, ga} /. list[[j, i, 3]], Element[#[[1]], Reals] &&
      #[[1]] > 0 && Element[#[[2]], Reals] && #[[2]] > 0 &];

  AppendTo[dataAll, {list[[j, i, 2]], list[[j, i, 1]], Length[mysol]}];
  AppendTo[dataAll2, {list[[j, i, 2]], list[[j, i, 1]], mysol}];

 ]
]
Export["dataepssigma.csv", dataAll, "CSV"];
Export["dataepssigmadat.csv", dataAll2, "CSV"];
Export["listbruteps.csv", list, "CSV"];
```

Ici cette boucle avec un pas plus grand, juste pour vérifier les équilibres.

```
list = Table[{sig, eps, getcoess[sig, eps]},
   {sig, 0.0, 0.5, 0.05}, {eps, 0.8, 1.3, 0.05}];
```

```
dataEqI = {};
dataEqS = {};
dataAllEqI = {};
dataAllEqS = {};
dataEqIeps = {};
dataEqSeps = {};
dataAll = {};
dataAll2 = {};
For[i = 1, i ≤ Dimensions[list][[2]], i++,
 For[j = 1, j ≤ Dimensions[list][[1]], j++,
  mysol = Select[{a, ga} /. list[[j, i, 3]], Element[#[[1]], Reals] &&
       #[[1]] > 0 && Element[#[[2]], Reals] && #[[2]] > 0 &];

  For[l = 1, l ≤ Length[mysol], l++,
   findEqI = Ieq /. res /. params[list[[i, j, 1]], list[[i, j, 2]]] /.
      a → mysol[[l, 1]] /. ga → mysol[[l, 2]];
   findEqS = Seq /. res /. params[list[[i, j, 1]], list[[i, j, 2]]] /.
      a → mysol[[l, 1]] /. ga → mysol[[l, 2]];
   AppendTo[dataEqI, {list[[j, i, 2]], list[[j, i, 1]], findEqI}];
   AppendTo[dataEqS, {list[[j, i, 2]], list[[j, i, 1]], findEqS}];
  ];

  EqI = Select[dataEqI[[All, 3]], # > 0 &] ;
  EqS = Select[dataEqS[[All, 3]], # < 100 &] ;
  AppendTo[dataAllEqS, {list[[j, i, 2]], list[[j, i, 1]], EqS}];
  AppendTo[dataAllEqI, {list[[j, i, 2]], list[[j, i, 1]], EqI}];

  AppendTo[dataAll, {list[[j, i, 2]], list[[j, i, 1]], Length[mysol]}];
  AppendTo[dataAll2, {list[[j, i, 2]], list[[j, i, 1]], mysol}];

  AppendTo[dataEqSeps, {list[[j, i, 2]], list[[j, i, 1]], Length[EqS]}];
  AppendTo[dataEqIeps, {list[[j, i, 2]], list[[j, i, 1]], Length[EqI]}];
  dataEqI = {}; dataEqS = {};
 ]
]
Export["dataepssigmaEqS.csv", dataEqSeps, "CSV"];
Export["dataepssigmaEqI.csv", dataEqIeps, "CSV"];
Export["dataepssigmaEqI-brut.csv", dataAllEqI, "CSV"];
Export["dataepssigmaEqS-brut.csv", dataAllEqS, "CSV"];
```

Récupération des densités d' hôtes et de parasites

```
dataEqIhi = {}; dataEqShi = {}; dataEqIlow = {}; dataEqSlow = {};
For[i = 1, i ≤ Dimensions[list][[2]], i++,
 For[j = 1, j ≤ Dimensions[list][[1]], j++,
  mysol = Select[{a, ga} /. list[[j, i, 3]], Element[#[[1]], Reals] &&
      #[[1]] > 0 && Element[#[[2]], Reals] && #[[2]] > 0 &];

  For[l = 1, l ≤ Length[mysol], l++,
   findEqI = Ieq /. res /. params[list[[j, i, 1]], list[[j, i, 2]]] /.
      a → mysol[[l, 1]] /. ga → mysol[[l, 2]];
   findEqS = Seq /. res /. params[list[[j, i, 1]], list[[j, i, 2]]] /.
      a → mysol[[l, 1]] /. ga → mysol[[l, 2]];
   If[findEqI > 0 && findEqI < 0.53, AppendTo[dataEqIlow,
     {list[[j, i, 2]], list[[j, i, 1]], findEqI}]];
   If[findEqI > 0 && findEqI > 0.53, AppendTo[dataEqIhi,
     {list[[j, i, 2]], list[[j, i, 1]], findEqI}]];
   If[findEqS < 100 && findEqS < 0.63, AppendTo[dataEqSlow,
     {list[[j, i, 2]], list[[j, i, 1]], findEqS}]];
   If[findEqS < 100 && findEqS > 0.63, AppendTo[dataEqShi,
     {list[[j, i, 2]], list[[j, i, 1]], findEqS}]];
  ];
 ]
]

Export["dataepssigmaEqSlow.csv", dataEqSlow, "CSV"];
Export["dataepssigmaEqShi.csv", dataEqShi, "CSV"];
Export["dataepssigmaEqIlow.csv", dataEqIlow, "CSV"];
Export["dataepssigmaEqIhi.csv", dataEqIhi, "CSV"];
```