Figures Bistab cup / sigma

```
params[ss_, mm_] =
  {c → 0.05, mu → 1, β0 → 10, k → 0.0, sig → ss, eps → 1, rm → 2, cup → mm};

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

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

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

getres[aa_, gg_] = Solve[{0 == rS[gg] x + rI[gg] y - (mu + (β[aa] y)) x + gg y,
     0 == β[aa] x y - (mu + aa + gg) y}, {x, y}][[2]] //
  Simplify; (*solve 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] + (rI'[ga] β[a] Ieq) / (mu + a + ga) + (rS[ga] - mu) / (mu + a + ga)] //
  Simplify; (*host selection gradient*)

getcoess[sig_, cup_] := Solve[
  {0 == para[a, ga], 0 == hote[a, ga]} /. params[sig, cup], {a, ga}] (*find coess*)

list = Table[{sig, cup, getcoess[sig, cup]},
   {sig, 0.0, 0.4, 0.001}, {cup, 0.0, 0.08, 0.0005}];
```

```
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["datacupsigma.csv", dataAll, "CSV"];
Export["datacupsigmadat.csv", dataAll2, "CSV"];
Export["listbrutcup.csv", list, "CSV"];
```

This nb to get the low and high coess values (not elegant but “does the job”)

```
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 < 0.7, AppendTo[dataEqIlow,
     {list[[j, i, 2]], list[[j, i, 1]], findEqI}]];
   If[findEqI > 0 && findEqI > 0.7, AppendTo[dataEqIhi,
     {list[[j, i, 2]], list[[j, i, 1]], findEqI}]];
   If[findEqS < 100 && findEqS < 0.7, AppendTo[dataEqSlow,
     {list[[j, i, 2]], list[[j, i, 1]], findEqS}]];
   If[findEqS < 100 && findEqS > 0.7, AppendTo[dataEqShi,
     {list[[j, i, 2]], list[[j, i, 1]], findEqS}]];
  ];
 ]
]

Export["datacupsigmaEqSlow.csv", dataEqSlow, "CSV"];
Export["datacupsigmaEqShi.csv", dataEqShi, "CSV"];
Export["datacupsigmaEqIlow.csv", dataEqIlow, "CSV"];
Export["datacupsigmaEqIhi.csv", dataEqIhi, "CSV"];
```

This nb is to get the brut data

```
dataEqI = {};
dataEqS = {};
dataAllEqI = {};
dataAllEqS = {};
dataEqIcup = {};
dataEqScup = {};
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[[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]];
   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[dataEqScup, {list[[j, i, 2]], list[[j, i, 1]], Length[EqS]}];
  AppendTo[dataEqIcup, {list[[j, i, 2]], list[[j, i, 1]], Length[EqI]}];

  dataEqI = {}; dataEqS = {};
 ]
]

Export["datacupsigmaEqS.csv", dataEqScup, "CSV"];
Export["datacupsigmaEqI.csv", dataEqIcup, "CSV"];
Export["datacupsigmaEqS-brut.csv", dataAllEqS, "CSV"];
Export["datacupsigmaEqI-brut.csv", dataAllEqI, "CSV"];
```