Replicator Dynamics - three player game

Simplex Plot

Plotting

```
originalgamematrix = {{{17.6, -19, 0.85}, {-7, 12.7, 10}, {-16.25, -11.2, 2}},
   {{0.8, -19, 0.85}, {12.65, -10, 20}, {8.25, 2.8, -9}},
   {{-2, 2, -1}, {1, 2, 1}, {-1, 1, 0}}};
(*transformedmatrix={{{19.6,-21,1.85},{-8,10.7,9},{-15.25,-12.2,2}},
  {{2.8,-21,1.85},{11.65,-12,19},{9.25,1.8,-9}},{{0,0,0},{0,0,0},{0,0,0}}};
transformedmatrix2={{{19.6,-21,0},{-8,10.7,0},{-13.4,-3.2,2}},
  {{2.8,-21,1.85},{11.65,-12,19},{9.25,1.8,-9}},{{0,0,0},{0,0,0},{0,0,0}}};*)
```

In[185]:=

```
payoffs = transformedmatrix2;
fits[x_, y_] :=
```

$$\text{Table}\left[\sum_{j=1}^{3}\left(\sum_{k=1}^{3}(\text{payoffs}[\![i,j,k]\!]\,x_j\,x_k)\right),\{i,1,3\}\right]\ /.\ \{x_1\to x,\ x_2\to y,\ x_3\to 1-x-y\};$$

In[187]:=

```
π1[x_, y_, z_] := fits[x, y][[1]];

π2[x_, y_, z_] := fits[x, y][[2]];

π3[x_, y_, z_] := fits[x, y][[3]];
πbar[x_, y_, z_] := x π1[x, y, z] + y π2[x, y, z] + z π3[x, y, z];
dx[x_,  y_,  z_] := x (π1[x, y, z] - πbar[x, y, z]);
dy[x_,  y_,  z_] := y (π2[x, y, z] - πbar[x, y, z]);
dz[x_,  y_,  z_] := z (π3[x, y, z] - πbar[x, y, z]);

Fermi[x_] := {
   dx[x[[1]], x[[2]],  x[[3]]],
   dy[x[[1]], x[[2]],  x[[3]]],
   dz[x[[1]], x[[2]],  x[[3]]]
  }
x =.; y =.; z = 1 - x - y;
fa = fits[x, y][[1]];
fb = fits[x, y][[2]];
fc = fits[x, y][[3]];
```

In[199]:=

```
x =.
y =.
sol = NSolve[{dx[x, y, 1 - x - y] == 0, dy[x, y, 1 - x - y] == 0}, {x, y}, Reals];
```

```
data = {x, y} /. sol;
relData = Select[data, Total[#] ≤ 1 && Total[#] ≥ 0 &];
p1 = ListPlot[relData, PlotStyle → {Black, PointSize[0.03]}];
p2 = ListPlot[relData, PlotStyle → {White, PointSize[0.015]}];
p3 = ListPlot[{{1, 0}, {0, 1}, {0, 0}}, PlotStyle → {Red, PointSize[0.03]}];
p4 = ListPlot[{{1, 0}, {0, 1}, {0, 0}}, PlotStyle → {White, PointSize[0.015]}];
points = Show[p1, p2, p3, p4];
```

Out[202]=

```
{{1., 0.}, {0.49745, 0.314111}, {0., 1.}, {0.118732, 0.456688},
 {0.371639, 0.0378092}, {0.316725, 0.}, {0., 0.473684},
 {0.0807156, 0.345594}, {0., 0.454545}, {0.180418, 0.}, {0., 0.}}
```

Plotting isoclines

In[209]:=

```
eq1 = fa - fc;
eq2 = fb - fc;
```

In[211]:=

```
cplt = ContourPlot[eq1 == 0, {x, 0, 1},
   {y, 0, 1}, RegionFunction → Function[{x, y}, x + y ≤ 1],
   PlotRange → All, ContourStyle → Darker[Red]];
cplt2 = ContourPlot[eq2 == 0, {x, 0, 1},
   {y, 0, 1}, RegionFunction → Function[{x, y}, x + y ≤ 1],
   PlotRange → All, ContourStyle → Darker[Blue]];

sp = StreamDensityPlot[{dx[x, y, 1 - x - y], dy[x, y, 1 - x - y]}, {x, 0, 1}, {y, 0, 1},
   Frame → True, ColorFunction → "Pastel", ColorFunctionScaling → False,
   StreamStyle → Darker[Gray], StreamColorFunction → None,
   StreamPoints → Fine, StreamMarkers →  {"PinDart"}, StreamScale → Large,
   RegionFunction → Function[{x, y, vx, vy, n}, x + y ≤ 1 && x ≥ 0 && y ≥ 0]];
```

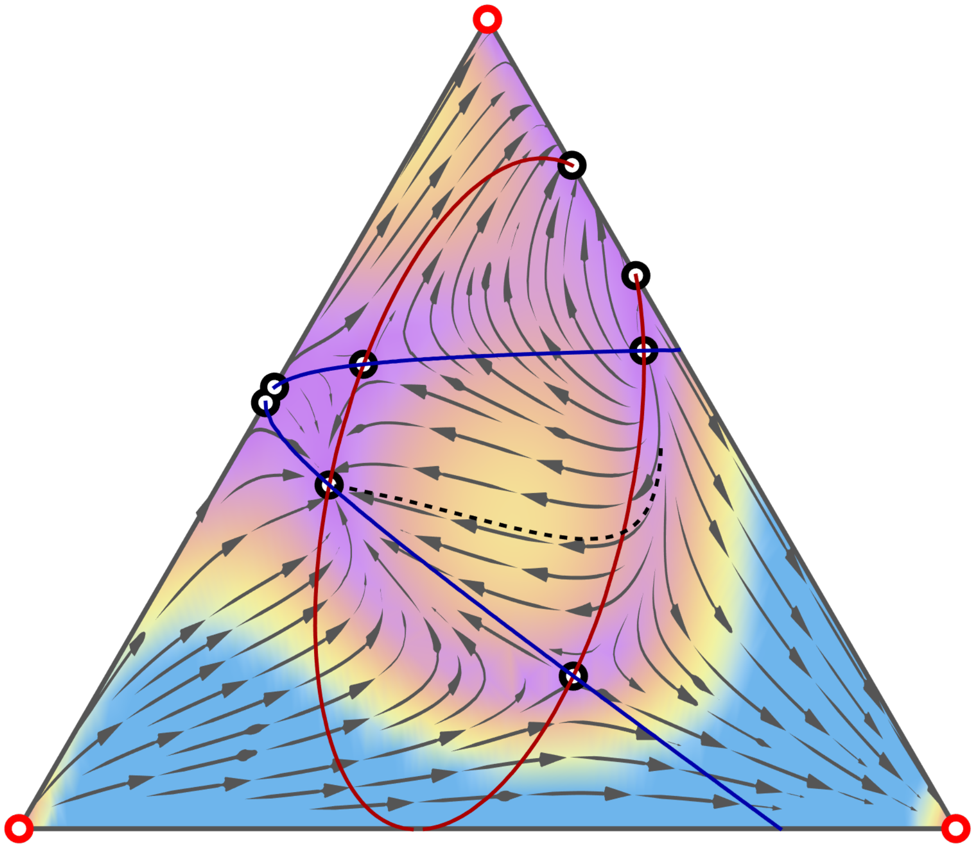
In[257]:=

```
trajdata = NDSolve[
   {x'[t] == dx[x[t], y[t], 1 - x[t] - y[t]], y'[t] == dy[x[t], y[t], 1 - x[t] - y[t]],
    x[0] == 0.45, y[0] == 0.08}, {x, y}, {t, 0, 50}];
parpl = ParametricPlot[Evaluate[{x[t], y[t]} /. trajdata],
   {t, 0, 50}, PlotRange → All, PlotStyle → Directive[Black, Dashed]];
```

In[259]:=

```
psim = Show[
  Graphics[GeometricTransformation[First[Show[sp]], trans]], triangle,
  Graphics[GeometricTransformation[First[Show[points]], trans]],
  Graphics[GeometricTransformation[First[Show[cplt, cplt2]], trans]],
  Graphics[GeometricTransformation[First[parpl], trans]](*,
  Graphics[GeometricTransformation[Text["label",{0,1}],trans]]*)
 ]
```

Out[259]=



Equivalent Lotka Volterra system

Use the game from above but call it A for ease

In[233]:=

```
A = originalgamematrix;
```

In[260]:=

```
A // MatrixForm
```

Out[260]//MatrixForm=

$$\begin{pmatrix} \begin{pmatrix} 17.6 \\ -19 \\ 0.85 \end{pmatrix} & \begin{pmatrix} -7 \\ 12.7 \\ 10 \end{pmatrix} & \begin{pmatrix} -16.25 \\ -11.2 \\ 2 \end{pmatrix} \\ \begin{pmatrix} 0.8 \\ -19 \\ 0.85 \end{pmatrix} & \begin{pmatrix} 12.65 \\ -10 \\ 20 \end{pmatrix} & \begin{pmatrix} 8.25 \\ 2.8 \\ -9 \end{pmatrix} \\ \begin{pmatrix} -2 \\ 2 \\ -1 \end{pmatrix} & \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} & \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \end{pmatrix}$$

The number of species is one less than the number of strategies

In[234]:=

```
n = 3; m = n - 1;
```

Transformation

First create a nxnxn matrix of zeros

In[235]:=

```
Table[b_{i,j,k} = 0, {i, 1, n}, {j, 1, n}, {k, 1, n}];
```

Now we fill in the growth rates

In[236]:=

```
Table[b_{i,n,n} = A⟦i, n, n⟧ - A⟦n, n, n⟧, {i, 1, 2, 1}]
```

Out[236]=

```
{2, -9}
```

Now we include the payoffs for the linear interactions for strategy 1

In[237]:=

```
Table[b_{1,j,n} = A⟦1, j, n⟧ - A⟦n, j, n⟧, {j, 1, m, 1}]
Table[b_{1,n,k} = A⟦1, n, k⟧ - A⟦n, n, k⟧, {k, 1, m, 1}]
```

Out[237]=

```
{1.85, 9}
```

Out[238]=

```
{-15.25, -12.2}
```

Now we include the payoffs for the nonlinear interactions for strategy 1

In[239]:=

```
Table[b_{1,j,k} = A⟦1, j, k⟧ - A⟦n, j, k⟧, {k, 1, m, 1}, {j, 1, m, 1}]
```

Out[239]=

```
{{19.6, -8}, {-21, 10.7}}
```

Now we include the payoffs for the linear interactions for strategy 2

In[240]:=

```
Table[b2,j,n = A⟦2, j, n⟧ - A⟦n, j, n⟧, {j, 1, m, 1}]
Table[b2,n,k = A⟦2, n, k⟧ - A⟦n, n, k⟧, {k, 1, m, 1}]
```

Out[240]=

```
{1.85, 19}
```

Out[241]=

```
{9.25, 1.8}
```

Now we include the payoffs for the nonlinear interactions for strategy 2

In[242]:=

```
Table[b2,j,k = A⟦2, j, k⟧ - A⟦n, j, k⟧, {k, 1, m, 1}, {j, 1, m, 1}]
```

Out[242]=

```
{{2.8, 11.65}, {-21, -12}}
```

Collect all in matrix B

In[243]:=

```
B = Table[bi,j,k, {i, 1, n}, {j, 1, n}, {k, 1, n}]
```

Out[243]=

```
{{{19.6, -21, 1.85}, {-8, 10.7, 9}, {-15.25, -12.2, 2}},
 {{2.8, -21, 1.85}, {11.65, -12, 19}, {9.25, 1.8, -9}},
 {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}}
```

In[244]:=

```
B // MatrixForm
```

Out[244]//MatrixForm=

$$\begin{pmatrix} \begin{pmatrix} 19.6 \\ -21 \\ 1.85 \end{pmatrix} & \begin{pmatrix} -8 \\ 10.7 \\ 9 \end{pmatrix} & \begin{pmatrix} -15.25 \\ -12.2 \\ 2 \end{pmatrix} \\ \begin{pmatrix} 2.8 \\ -21 \\ 1.85 \end{pmatrix} & \begin{pmatrix} 11.65 \\ -12 \\ 19 \end{pmatrix} & \begin{pmatrix} 9.25 \\ 1.8 \\ -9 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \end{pmatrix}$$

In[245]:=

$$\text{lveqs} = \text{Table}\Big[y_i'[t] == y_i[t] \Big((b_{i,n,n} - b_{n,n,n}) + \sum_{j=1}^{m} (b_{i,j,n} - b_{n,j,n})\, y_j[t] + \sum_{k=1}^{m} (b_{i,n,k} - b_{n,n,k})\, y_k[t] + \Big(\sum_{j=1}^{m} \sum_{k=1}^{m} (b_{i,j,k} - b_{n,j,k})\, y_j[t]\, y_k[t] \Big) \Big), \{i, 1, 2, 1\} \Big]$$

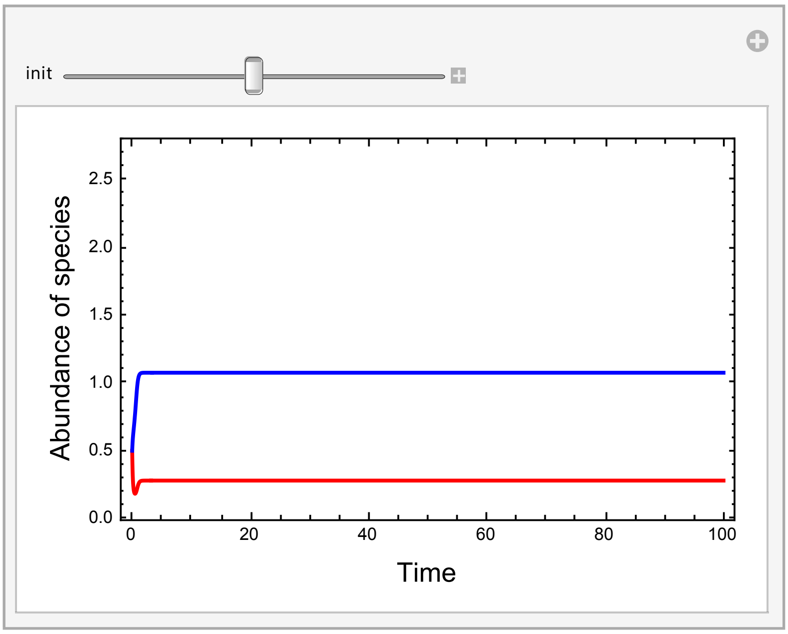
Out[245]=

$$\{y_1'[t] == y_1[t]\,(2 - 13.4\, y_1[t] + 19.6\, y_1[t]^2 - 3.2\, y_2[t] - 29\, y_1[t]\, y_2[t] + 10.7\, y_2[t]^2),$$
$$y_2'[t] == y_2[t]\,(-9 + 11.1\, y_1[t] + 2.8\, y_1[t]^2 + 20.8\, y_2[t] - 9.35\, y_1[t]\, y_2[t] - 12\, y_2[t]^2)\}$$

In[251]:=

```
tmax = 100;
Manipulate[lv = NDSolve[{lveqs,
    y1[0] == init, y2[0] == 1 - init}, {y1, y2}, {t, 0, tmax}];
 Plot[Evaluate[{y1[t], y2[t]} /. lv], {t, 0, tmax}, PlotStyle → {Red, Blue},
  PlotStyle → Automatic, Frame → True, FrameStyle → Directive[Black,
    Thickness[0.003]], PlotRange → {Automatic, {-0.01, 2.8}}, FrameLabel →
   {Style["Time", 14], Style["Abundance of species", 14]}], {{init, 0.5}, 0, 1}]
```

Out[252]=



In[253]:=

```
Evaluate[{y1[tmax], y2[tmax]} /. lv]
```

Out[253]=

```
{{0.279646, 1.07562}}
```

In[254]:=

```
ParametricPlot[Evaluate[{y1[t], y2[t]} /. lv], {t, 0, tmax}, Axes → None,
 PlotStyle → Black, PlotStyle → Black, PlotStyle → Automatic, Frame → True,
 FrameStyle → Directive[Black, Thickness[0.004]], PlotRange → Full,
 FrameLabel → {Style["Species 1", 16, RGBColor["#807FFF"]],
   Style["Species 2", 16, RGBColor["#DEAD26"]]}, AspectRatio → 1]
```

Out[254]=