

Studying Discrete Time Dynamics of 2 x2 Normal – Form Games Using Mathematica

In this notebook, I give some examples of how to use the functions defined in my ``Discrete2x2Dynamics`` package contained in the file ``Discr2x2.m``. I have used the 'Matching Pennies' game throughout this notebook, and I have evaluated the sample cells. I have included the outputs I got when I ran these examples. If you need help with the syntax of my functions, you can get online help for them by evaluating the expression (e.g.) `?DarwinOrbit` once you have loaded the package (i.e., once you have evaluated the first cell in this notebook -- see below).

The two dynamical models (i.e., darwin and nash) seen here are taken from Brian Skyrms's book *The Dynamics of Rational Deliberation*. It should be easy enough to add new dynamical maps of your own to my package.

First, load in the ``Discrete2x2Dynamics`` package, by evaluating the following cell (here, the path should be set appropriately to find the package file `Discr2x2.m`).

```
<< "discr2x2.m";
Off[Solve::"svars"];

In[3]:= G = {R11, C11, R12, C12, R21, C21, R22, C22} = {1, 0, 0, 1, 0, 1, 1, 0};
Print["The game in question is, G = ", TableForm[Partition[G, 4]]];

The game in question is, G =


|   |   |   |   |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |


```

You can use the `?` function to obtain information about the various functions defined in the package.

```
In[5]:= ?DarwinOrbit

DarwinOrbit[{G_, K_, SAT_, IMAX_, PrR20_, PrC20_}]. Takes the five-tuple <G, K,
SAT, IMAX_, PrR20, PrC20> into a graphics object of the orbit starting from the
initial point <PrR20, PrC20> under Darwin dynamics. Here, SAT is the satisficing
index (IMAX is the maximum number of iterations allowed), and K is the index of
caution. G is the 2x2 Normal form game--it is an eight-tuple of the form: G =
{R11, C11, R12, C12, R21, C21, R22, C22} See Skyrms (1990) Appendices for details.
```

```
In[6]:= ?NashOrbit

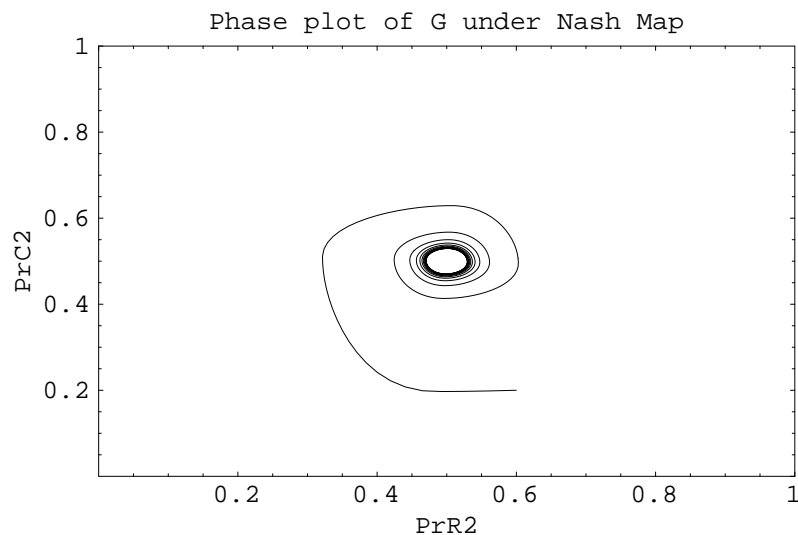
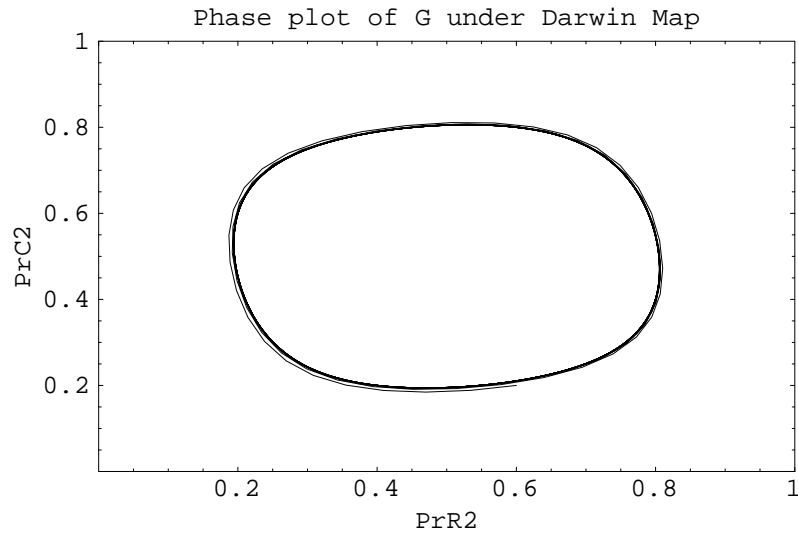
NashOrbit[{G_, K_, SAT_, IMAX_, PrR20_, PrC20_}]. Takes the five-tuple <G, K,
SAT, IMAX, PrR20, PrC20> into a graphics object of the orbit starting from the
initial point <PrR20, PrC20> under Nash dynamics. Here, SAT is the satisficing
index (IMAX is the maximum number of iterations allowed), and K is the index of
caution. G is the 2x2 Normal form game--it is an eight-tuple of the form: G =
{R11, C11, R12, C12, R21, C21, R22, C22} See Skyrms (1990) Appendices for details.
```

Here, we look at an orbit of the game **G** above (viz., "matching pennies") under both the darwin map and the nash map.

```

In[7]:= Sims = {{G, 5, .01, 1000, .6, .2}};
DarwinPhasePlot = DarwinOrbit /@ Sims;
NashPhasePlot = NashOrbit /@ Sims;
Show[{DarwinPhasePlot}, Prolog -> Thickness[.0005], PlotRange -> {{0, 1}, {0, 1}},
  Frame -> True, FrameLabel -> {"PrR2", "PrC2", "Phase plot of G under Darwin Map", ""}];
Show[{NashPhasePlot}, Prolog -> Thickness[.0005], PlotRange -> {{0, 1}, {0, 1}},
  Frame -> True, FrameLabel -> {"PrR2", "PrC2", "Phase plot of G under Nash Map", ""}];

```



There are also functions for theoretically analysing the stability properties of equilibria in the games under the darwin map (the nash map is discontinuous, and so the standard stability analysis techniques do not apply to it).

```
In[12]:= ? DarwinStability
```

DarwinStability[G_], (1) finds all of the Nash equilibria of the 2x2 Normal-Form game G, and then (2) performs a linear stability analysis for the discrete time Darwin dynamical map at each of these equilibria. Specifically, this function outputs the absolute values of the eigenvalues of the Jacobian Matrix of the Darwin map of G at each of its Nash points.

```
In[13]:= DarwinStability[G]
```

```
The Nash equilibria of G are: {{0.5, 0.5}}
```

```
The Jacobian of the Discrete Darwin Map at the point {0.5, 0.5} is  $\begin{matrix} 1. & 1. \\ -1. & 1. \end{matrix}$ 
```

```
The Absolute values of the Eigenvalues of the  
Jacobian of the Discrete Darwin Map at the point {0.5, 0.5} are: {1.41421, 1.41421}
```

So, under the darwin map, the equilibrium $\langle 1/2, 1/2 \rangle$ of the matching pennies game is *not* a stable attractor (under the nash map, the numerical simulations seem to indicate that it *is* a stable attractor).

There are a couple of other useful functions in this package as well:

```
In[14]:= G1 = {{R11, C11}, {R12, C12}},  
            {{R21, C21}, {R22, C22}}
```

```
Out[14]= {{1, 0}, {0, 1}}, {{0, 1}, {1, 0}}
```

```
In[15]:= ? IsNash
```

```
IsNash[game_, strategies_] returns True if strategies is a Nash Equilibrium  
of game and False otherwise. Example input: IsNash[game, {{2/3, 1/3}, {1/3, 2/3}}].
```

```
In[16]:= IsNash[G1, {{1/2, 1/2}, {1/2, 1/2}}]
```

```
Out[16]= True
```

```
In[17]:= ? Nash
```

```
Nash[game_] finds the Nash Equilibria of game, a game  
in normal form. Example input: Nash[{{2, 1}, {0, 0}}, {{0, 0}, {1, 2}}].  
Nash returns the probability weights on the different pure strategies.
```

```
In[18]:= Nash[G1]
```

```
Out[18]= {{{1/2, 1/2}, {1/2, 1/2}}
```

```
In[19]:= Nash[{{2, 1}, {0, 0}}, {{0, 0}, {1, 2}}]
```

```
Out[19]= {{0, 1}, {0, 1}}, {{2/3, 1/3}, {1/3, 2/3}}, {{1, 0}, {1, 0}}
```