On the Dynamic Stability of Bayesian Nash Equilibrium

Jean Paul Rabanal and Daniel Friedman
Economics Department
University of California, Santa Cruz

December 19, 2009

Abstract

We illustrate techniques for assessing the dynamic stability of Bayesian Nash Equilibrium, using a combination of replicator and gradient dynamics. For a range of reasonable parameters in the Friedman and Singh (2009) noisy trust game, we obtain convergence in time average for initial conditions sufficiently near to the equilibrium values.

Keywords: Stability, Perfect Bayesian Equilibrium, evolutionary dynamics, vengeance.
JEL codes: C62, C73, Z13

1 Introduction

A major practical difficulty in applying the game theory is that games of incomplete information often have too many equilibria. Dynamic stability seems a natural equilibrium refinement in applied work, because unstable equilibria are unlikely to be observed. Indeed, dynamic stability is the intuitive basis for static refinements in other contexts, e.g., the neoclassical idea long-run competitive equilibrium, or the Maynard Smith and Price (1973) notion of evolutionarily stable states (ESS) for games of complete information.

For games of incomplete information, Friedman and Singh (2009, henceforth FS09) recently proposed a static equilibrium refinement called Evolutionary Perfect Bayesian Equilibrium (EPBE). At EPBE, each surviving type has the same expected payoff, and no potential entrant type has higher expected payoff. However, there are as yet no explicitly dynamic underpinnings for EPBE nor, as far as we know, for any other refinement for games of incomplete information.

The present paper constructs dynamics justifying EPBE. It shows how standard specifications—replicator dynamics and gradient dynamics—can be combined to assess computationally the dynamic stability of EPBE. It does so in the context of the noisy trust game for which EPBE was developed.

The next section describes the basic trust game, and its extension to a game of imperfect information. Section 3 sketches expected payoff calculations, while Section 4 writes out the adjustment dynamics as a system of ordinary differential equations. Section 5 summarizes their numerical solutions. For reasonable parameters, we achieve convergence in time average to the relevant EPBE from initial conditions near to the equilibrium values. That is, the EPBE of the noisy trust game is locally asymptotically stable in time average. Section 6 discusses the implications, and Appendix A shows the Matlab code for the numerical solutions.

2 The FS09 Game

Consider the simple two player game of complete information illustrated in Figure 1. The first mover, labelled Self (S), chooses whether to trust (T) or not trust (N). Choice N ends the game with zero payoffs to both players. Choice T gives the move to player Other (O), who can choose either to to cooperate (C) or defect (D). Choice C gives both players unit payoffs, while choice D yields payoffs 2 to Other and -1 to Self. However, following D, Self can take revenge at chosen cost $v \in [0, v_H]$ to himself and inflict harm v/c on Other, where the marginal cost parameter $c > 0$ is given.

From this simple game, FS09 constructs the noisy trust game illustrated in Figure 2. Nature

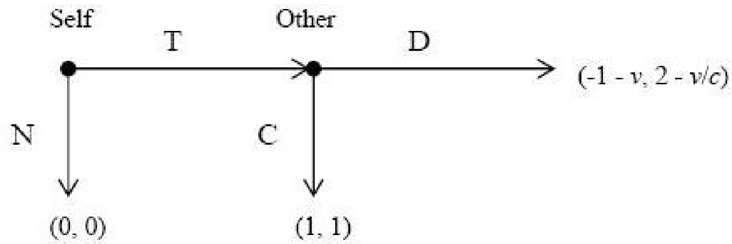

Figure 1: The basic trust game with vengeance.

chooses Self's non-vengeful type $v = 0$ with probability $1 - x$, or else chooses a given vengeful type $v = v_H > 0$ with probability x. Nature also independently chooses Other's perception as correct ($s = 0$ for $v = 0$, or $s = 1$ for $v = v_H$) with probability $1 - a$, or incorrect with probability a. Perception is more accurate for more extreme types; the misperception probability is

$$a = A(v_H) = 0.5 \exp(-k v_H^2) \tag{1}$$

where $k > 0$ represents a precision parameter.

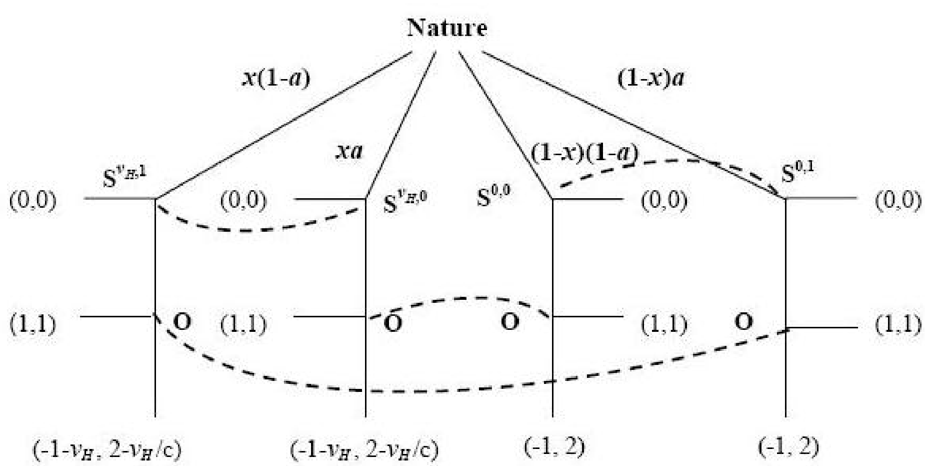

Figure 2: The noisy trust game. O denotes Other, S^{ij} denotes Self with vengeance level i and perception j, as determined by Nature's move. The four branch labels are Nature's move probabilities. Source: FS09

Let $p_0 = Pr[T|v = 0]$ denote the probability of trusting when S is non-vengeful, and $p_1 = Pr[T|v = v_H]$ the probability of trusting when S is vengeful. These probabilities are constrained by a tremble rate $e \in [0, 1/2)$, so that $e \leq p_0, p_1 \leq 1 - e$. Self's (mixed) strategy space thus is $[e, 1-e] \times [e, 1-e]$. Similarly, let $p_2 = Pr[C|s = 1]$ and $p_3 = Pr[C|s = 0]$ denote the probabilities of cooperating when Other observes a non-vengeful type and a vengeful type, respectively. Applying the same tremble rate to both populations, we see that Other's strategy space is also $[e, 1 - e] \times$

$[e, 1-e]$.

As explained in FS09, it makes sense to think of the noisy game as played between two large populations of randomly matched Selfs and Others. The state of the system is a vector $(p_0, p_1, p_2, p_3, x, v) \in [e, 1-e]^2 \times [e, 1-e]^2 \times [0, 1] \times [0, v_H]$ that specifies Self's actions (p_0 and p_1), Other's actions (p_2 and p_3), the fraction of the vengeful type (x) and its degree of vengefulness ($v = v_H$).

The equilibrium concept here is perfect Bayesian equilibrium (PBE), suitably phrased to deal with large populations and explicit trembles. PBE requires all players to optimize given beliefs, which are Bayesian posterior probabilities obtained from perceptions, observed actions, and prior information on the type proportions. Proposition 1 of FS09 identifies seven families of PBE that depend on game parameters x, a, v_H and e: three pure strategy PBE families and four mixed strategy PBE families.

Which of these many PBEs are sustainable in the long run? That is, which of them remains when the fraction x and the degree v_H of vengefulness (and the corresponding misperception probability a) can adjust? To answer, FS09 proposes a static refinement called evolutionary perfect Bayesian equilibrium (EPBE). In EPBE, all types in the support of the distribution in each population achieve equal and maximal expected fitness, and no potential entrant (a type outside the support) has higher expected payoff.

Proposition 2 of FS09 shows that only two states survive the EPBE refinement:

1. For each punishment cost $c \in (0, 1)$, behavioral error rate $e \in (0, \hat{e}(k))$[1] and precision parameter $k \in (0, 0.6)$, there is a unique "Good Hybrid" EPBE in which Self trusts regardless of her type and Other plays a specific mixed strategy when she perceives a non-vengeful type.

2. For all perception technologies, $c > 0$ and behavioral error rates $e \in (0, 1/2)$, there is a unique "Bad Pooling" EPBE in which (apart from trembles) Self never trusts and Other always defects.

For example, if $c = 0.80$, $e = 0.05$ and $k = 0.10$, then the unique Good Hybrid EPBE has $x = 0.75$, $v = 4.50$ and $a = 0.07$. It yields expected payoff 0.58 to both Self types, and 0.53 to Other in the case she observes a vengeful type. In this example, the unique Bad Pooling EPBE yields expected payoffs -0.05 to Self (who are all nonvengeful) and 0.05 to Other.

[1]See FS09 for the specification of the maximal admissible tremble rate $\hat{e}(k) \leq 0.5$.

3 Expected payoffs

The intuition behind EPBE is that evolutionary dynamics reduce the prevalence of types with lower expected payoff (or "fitness") and increase that of types with higher fitness until EPBE is achieved. In the next section we test that intuition by introducing plausible evolutionary dynamics. Here we set the stage by deriving the expected payoffs.

The expected payoffs w_s^v and w_s of vengeful and non-vengful types of Self are:

$$w_s^v = (p_1(1-a)p_2 + p_1 a p_3) * 1 + (p_1(1-a)(1-p_2) + p_1 a(1-p_3)) * (-1-v) \tag{2}$$

$$w_s = (p_o(1-a)p_3 + p_o a p_2) * 1 + (p_o(1-a)(1-p_3) + p_o a(1-p_2)) * (-1) \tag{3}$$

Equation (2) is derived as follows. If vengeful Self does not trust (probability $1-p_1$), she receives a payoff of zero. On the other hand, if she trusts (probability p_1), she gets payoff 1 or $-1-v$ depending on Other's decision and perception. Her payoff is 1 when Other perceives correctly (probability $(1-a)$) a vengeful type and cooperates (probability p_2), and also when Other misperceives (probability a) and cooperates (probability p_3). She gets $-1-v$ when Other perceives correctly $(1-a)$ and defects $(1-p_2)$, and also when she misperceives (a) and defects $(1-p_3)$. Similar logic yields the expression for non-vengeful Self's payoff w_s.

The expected payoffs w_o^s and w_o for Other when he perceives a vengeful or a non-vengeful type are:

$$\begin{aligned} w_o^s = \; & (x(1-a)p_1p_2 + (1-x)ap_op_2) * (1) + (x(1-a)p_1(1-p_2)) * (2-v/c) + \\ & ((1-x)ap_o(1-p_2)) * 2 \end{aligned} \tag{4}$$

$$\begin{aligned} w_o = \; & (xap_1p_3 + (1-x)(1-a)p_op_3) * (1) + (xap_1(1-p_3)) * (2-v/c) + \\ & ((1-x)(1-a)p_o(1-p_3)) * 2 \end{aligned} \tag{5}$$

To derive (4), note that Other who perceives a vengeful type gets payoff 2 when a non-vengeful Self (fraction $1-x$ of that population) trusts (probability p_o), is misperceived as vengeful (probability a) and he defects (probability $1-p_2$). A second possible payoff for him is $2-v/c$, obtained when Self is vengeful (fraction x) and trusts (probability p_1) while Other perceives correctly (probability $1-a$) and defects (probability $1-p_2$). The remaining possible payoff is 1, obtained when a vengeful type Self (x) trusts (p_1) is correctly perceived $(1-a)$ and Other cooperates (p_2) or, alternatively, when a non-vengeful Self $(1-x)$ trusts (p_o) and Other misperceives (a) and cooperates (p_2). Similar logic yields the expected payoff w_o when Other perceives a non-vengeful type Self.

4 Adjustment dynamics

Recall that the state space is six dimensional, and specifies the fraction of vengeful type (x), the degree of vengefulness (v), and four mixing probabilities (p_i). We therefore specify dynamics as a system of six coupled ordinary differential equations (ODEs), derived from expected payoffs using standard evolutionary principles.

With two alternative types, the basic principle of evolution is that the type with higher payoff (= fitness) will increase its share of the population. This could be due to individual learning or imitation—players are more likely to switch to the more successful strategy and less likely to switch away from it. Or it might be due to demographic change—more rapid entry to and slower exit from the more successful strategy, or even (as in biology) a lower death rate or higher birth rate for the more successful strategy.

To implement this general principle, we use standard continous time replicator dynamics (Taylor and Jonker, 1978; Hofbauer and Sigmund, 1988). For the share x of vengeful types in the Self population, replicator dynamics postulate that the growth rate $\dot{x}/x$ is proportional (with rate constant β_x) to its own payoff w_s^v relative to the population average $xw_s^v + (1-x)w_s$. Since $w_s^v - [xw_s^v + (1-x)w_s] = (1-x)(w_s^v - w_s)$, we obtain equation (6) below from standard replicator dynamics.

The remaining state variables involve a continuum of alternatives, not just two or a finite number of alternatives. Here we rely on gradient dynamics, which are standard for continuous evolution in biology (e.g., Wright (1949), Eshel (1983) and Kauffman (1993)), and seem more common than other possible specifications in economics (e.g., Sonnenschein (1982)). Thus the degree of vengefulness $v = v_H$ changes at a rate proportional to its gradient $\frac{\partial w_s^v}{\partial v}$. (Note that the gradient is positive when v is negative, so we can harmlessly impose the boundary condition $v \geq 0$.)

We use hybrid gradient-replicator dynamics for each mixing probability p_i. Its adjustment rate is proportional to its payoff gradient $\frac{\partial w_s^{[v]}}{\partial p_i}$. To shrink the range to $[e, 1-e]$, we include factors $(1-e-p_i)(p_i-e)$, analogous to the replicator factors $(1-x)x$ that keep x in the interval $[0,1]$. Thus our system of six ODEs is:

$$\dot{x} = \beta_x(1-x)x(w_s^v - w_s) \tag{6}$$

$$\dot{v} = \beta_v\left(\frac{\partial w_s^v}{\partial v}\right) \tag{7}$$

$$\dot{p}_o = \beta_o(1-e-p_o)(p_o-e)\left(\frac{\partial w_s}{\partial p_o}\right) \tag{8}$$

$$\dot{p}_1 = \beta_1(1-e-p_1)(p_1-e)\left(\frac{\partial w_s^v}{\partial p_1}\right) \tag{9}$$

$$\dot{p}_2 = \beta_2(1-e-p_3)(p_2-e)\left(\frac{\partial w_o^s}{\partial p_2}\right) \quad (10)$$

$$\dot{p}_3 = \beta_3(1-e-p_3)(p_3-e)\left(\frac{\partial w_o}{\partial p_3}\right) \quad (11)$$

We assume that p_i, representing individual learning, adjusts more rapidly than does x, representing entry and exit, or type switching, and that v, representing genetic disposition, adjusts least rapidly. Thus $0 < \beta_v < \beta_x < \beta_0 = \beta_1 = \beta_2 = \beta_3$. To complete the dynamic specification, take the initial state as given and impose the boundary conditions $0 \leq x \leq 1, v \geq 0$ and $e \leq p_i \leq 1-e$.

Two remarks are in order before proceeding. It might seem that we are dealing with a system of partial differential equations, but the gradients on the right hand side can be expressed in terms of the state variables only, using equations (2-4). Indeed, in the $\dot{p}_i$ equations the gradients are of functions that are linear in p_i. Hence these gradients can be replaced by fitness difference terms, and then the last four equations look more like standard replicator equations. In the equation for $\dot{p}_o$, for example, we see from equation (3) that $\frac{\partial w_s}{\partial p_o} = w_s|_{[p_o=1]} - w_s|_{[p_o=0]}$.

The other remark is that for standard replicator dynamics in dimension 2 and higher, it has been recognized for some time that Liouville's theorem precludes convergence to any interior equilibrium from an open basin of attraction (e.g., see Fudenberg and Levine, 1998, p.95). For that reason, we shall focus on convergence in time average of the state variable $S = (x, v, p_o, p_1, p_2, p_3)$. That is, we shall look at approximations of $\lim_{t\to\infty} t^{-1}\int_0^t S(u)du$, rather than of $\lim_{t\to\infty} S(t)$ directly, when investigating the stability of the interior ("good") EPBE.

5 Results

We solve the ODE system numerically using the Matlab shown in Appendix A. The baseline parameters are $k = 0.2$, c=0.8, e=0.1, β_v=0.01, β_x=0.10 and $\beta_{1,2,3} = 2$.

Panel A figure 3 shows typical numerical solutions for baseline parameters and initial conditions not far from the EPBE. The state indeed cycles around the "good" EPBE with constant amplitude, consistent with Liouville's theorem. Panel B confirms convergence in time average.

Table 1 shows convergence in time average for two departures from baseline parameters, to c=0.2 instead of c=0.8, and to $k = 0.4$ instead $k = 0.2$. More generally, numerical simulations indicate that the "good" EPBE is locally asymptotically stable for all parameters values within the set $c \in (0, 1)$, $e \in (\hat{e}(k))$ and $k \in (0, 0.6)$.

There are three caveats. First, of course, is that we are referring to convergence in time average. Second, we are talking about local stability, so we don't drastically alter the initial state. For baseline parameters we have confirmed convergence from initial states $v^* - 0.04 \leq v(0) \leq v^* + 0.01$,

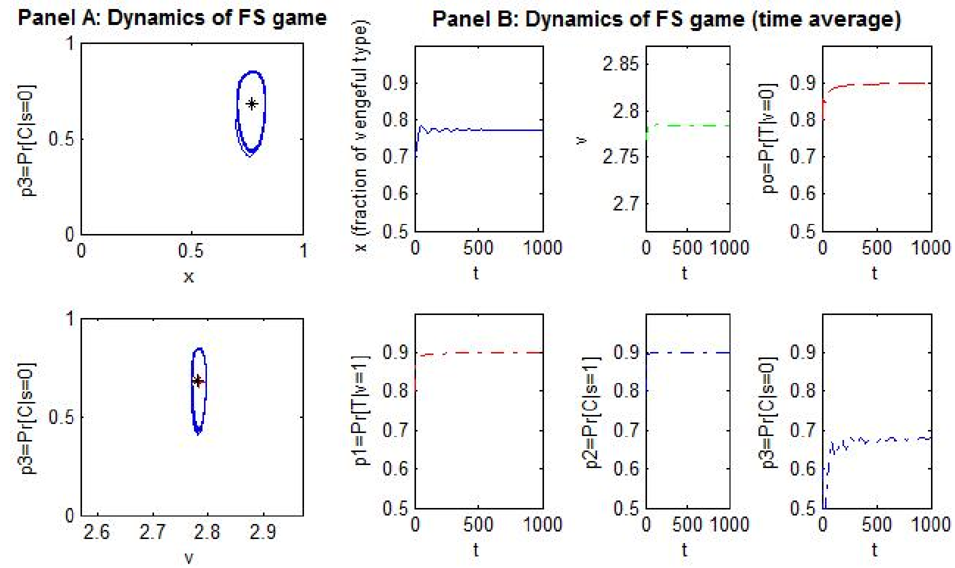

Figure 3: Dynamics (Panel A) and Time-Average dynamics (Panel B) of the FS09 game. Matlab parameters include maximum time step 1/4, relative error tolerance $1e-8$ and absolute error tolerance $1e-9$. In Panel A, the "Good" EPBE is indicated by * and the time average by +.

$x^* - 0.2 \leq x(0) \leq x^* + 0.15$, and $0.45 \leq p_i(0) \leq 1 - e$. Third, we must restrict the adjustment speeds appropriately. The proportion x of the vengeful type should adjust more rapidly than the degree v of vengefulness, i.e., $\beta_v < \beta_x$. This restriction is consistent with the idea from FS09 that slow genetic adjustment controls v, while exit and entry control x.

Table 1: Numerical results

	c=0.2			k=0.4		
	x	v	p_3	x	v	p_3
Initial	0.50	2.81	0.60	0.75	1.65	0.60
EPBE	0.40	2.78	0.68	0.82	1.67	0.74
Time average	0.40	2.79	0.69	0.82	1.67	0.74

Note The other p_i's start from 0.8 and achieve the upper bound $1 - e$.

The "Bad" EPBE is at the corner of the state space, where the mixing probabilities are at the lower bound e and the fraction of vengeful type x goes to zero. Liouville's theorem does not preclude direct convergence to a corner equilibrium. Indeed, we find direct covergence to the "Bad" EPBE as in Figure 4, given initial values of $x < 0.2$ (not many vengeful types) and $p_2 < 0.2$ (a low probability that Other cooperates).

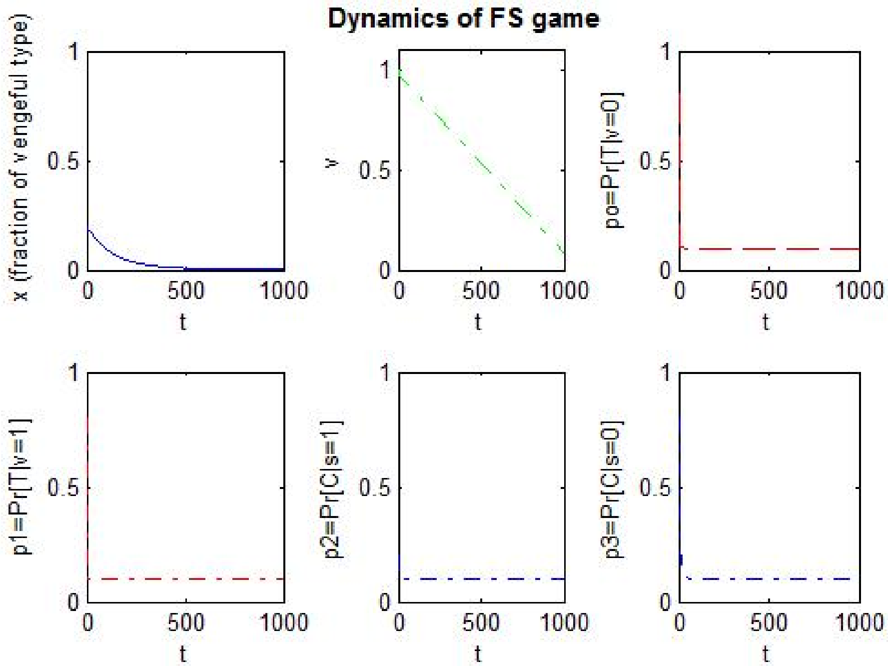

Figure 4: Convergence to the “Bad” EPBE

6 Discussion

The preceding establishes a “proof of concept”. We considered a moderately complicated game of incomplete information with two populations (distinct player roles). The underlying strategy sets were discrete ($\{N, T\}$ and $\{C, D\}$) but the state space was six dimensional and continuous. To assess dynamic stability of an equilibrium refinement called EPBE, we specified replicator dynamics or gradient dynamics or hybrids, as appropriate for each of the six state variables.

We focused on convergence in time average to the interior (“good”) EPBE of our example game, and obtained very positive results. We also found a sizeable basin of attraction for the corner (“bad”) EPBE. Thus numerical simulations indicate that, among the plethora of Perfect Bayesian equilibria, the static refinement EPBE indeed identified those that are locally asymptotically stable.

The point of our exercise is to encourage the use of long-run dynamic stability to refine the multiplicity of Bayesian Nash equilibria. In our example application, there were seven fat families of perfect Bayesian equilibria, but only the two EPBE were found to be stable and therefore detectable in stationary data. We hope that other investigators can use our methods to sharpen the testable predictions in a variety of other applications.

References

[1] **Eshel, Ilan**. "Evolutionary and Continuous Stability." *Journal of Theoretical Biology*, 103, pp. 99-111, 1983.

[2] **Friedman, D. and N. Singh.** "Equilibrium vengeance." *Games and Economic Behavior*, 66, pp. 813-829, 2009.

[3] **Fudenberg, D. and D. Levine**. *The Theory of Learning in Games*, MIT Press, 1998.

[4] **Hofbauer, J. and K. Sigmund**. *The Theory of Evolution and Dynamical Systems*, Cambridge University Press, 1988.

[5] **Kaufman, Stuart.** *The Origins of Order: Self-Organization and Selection in Evolution*, NY: Oxford U Press, 1993.

[6] **Maynard Smith, J. and G.R. Price** "The Logic of Animal Conflict." *Nature*, 246, pp. 15-18, 1973.

[7] **Sonnenschein, H.** "Price dynamics based on the adjustment of firms. " *American Economic Review*, 72 (5), pp. 1088-1096, 1982.

[8] **Taylor, P.D. and L.B. Jonker.** "Evolutionary stable strategies and game dynamics." *Mathematical Biosciences*, 40, pp. 145-156, 1978.

[9] **Wright, Sewall.** "Adaption and Selection," in L. Jepsen, G.G. Simpson, and E. Mayr eds., *Genetics, Paleontology, and Evolution.* Princeton, N.J.: Princeton University Press, 1949.

7 Appendix A

```
%On the Dynamic Stability of Bayesian Nash Equilibrium
%Authors: Dan Friedman and Jean Paul Rabanal
%Economics Department
%UC Santa Cruz
%December 2009
% This m.file solves the dynamic system specified in venge2.m The user
% should incorporate exogenous parameters, and ODE solver options.
%=======================================================================
clear;
% Pick what series what to draw
% time average (1) or levels 0.
taverage=1;
%=======================================================================
%Exogeneous parameters
global e c v k emin a velo g
e=0.10;        % tremble
emin = e;
c=0.8;         % cost
k=0.2;         % k is  used in the expression a=0.5*exp(-k*v^2)
tf = 1000;     % Time span


global betax betav betao beta1 beta2 beta3
velo=2;
betax=0.1;     %speed of adjument for equation 1 (x)
betav=0.01;    %speed of adjument for equation 2 (v)
betao= velo;   %speed of adjument for equation 3 (po)
beta1= velo;   %speed of adjument for equation 4 (p1)
beta2= velo;   %speed of adjument for equation 5 (p2)
beta3= velo;   %speed of adjument for equation 6 (p3)

global xo vo
xo = 0.6;     %initial value for x
```

```
vo = 2.77;     %initial value for v

%Mixing probabilities
%po = Pr[T|v=0]=  t(1-e)+(1-t)e  = y(3)
%p1 = Pr[T|v=1]=  u(1-e)+(1-u)e  = y(4)
%p2 = Pr[C|s=1]= (1-r)(1-e) + re = y(5)
%p3 = Pr[C|s=0]= (1-q)(1-e) + qe = y(6)

global poo p1o p2o p3o
pp=0.8;
poo= pp;       %initial value po
p1o= pp;       %initial value p1
p2o= pp;       %initial value p2
p3o= 0.6;      %initial value p3

%Solver
tspan=[0 tf];   %Time span
options = odeset('NonNegative',[2], 'MaxStep',1/4,'RelTol',1e-8,'AbsTol',[1e-9 1e-9 1e-9 ...
1e-9 1e-9 1e-9]);
[t,y] = ode45(@venge2,tspan,[xo vo poo p1o p2o p3o],options);

%=========================================================================
%Computation the equilibrium values that will appear in the plots.
po=1-e;
p1=1-e;
p2=1-e;
v=fzero('nlem',5); % (5 is initial guess)
R=(k*v*(2+v)-0.5)*exp(-k*v^2);
q=e/((1-2*e)*R);
qcode=1-((1-2*e)*q+e);
ae=0.5*exp(-k*v^2);
xe=(1-ae)/(1+(v/c-2)*ae);
wos=(xe*(1-ae)*p1*p2+(1-xe)*ae*po*p2)*1+...
    (xe*(1-ae)*p1*(1-p2))*(2-v/c)+...
```

```
    ((1-xe)*ae*po*(1-p2))*2;
wo=(xe*ae*p1*qcode+(1-xe)*(1-ae)*po*qcode)*(1) +...
    (xe*ae*p1*(1-qcode))*(2-v/c)+((1-xe)*(1-ae)*po*(1-qcode))*2;

wsve=(p1*(1-ae)*p2+p1*a*qcode)*1 + (p1*(1-ae)*(1-p2)+...
    p1*ae*(1-qcode))*(-1-v);
wse=(po*(1-ae)*qcode+po*ae*p2)*1 +...
    (po*(1-ae)*(1-qcode)+po*ae*(1-p2))*(-1);
ek= R/(2-2*ae+2*R);

%Time average computation
l=length(y)
z=y;
z(:,:)=0;
z(1,:)=y(1,:)
for j= 2:l
  z(j,:)=z(j-1,:).*(j-1)./(j)+y(j,:)./(j);
end
%===========================================================================
% Graphs (time average or levels)
if taverage==1
    z=z;
else
    z=y;
end

h=title('Dynamics of Friedman and Singh Paper')
subplot(2,3,1)
plot(t,z(:,1),'-');
hold on
plot(t,xe,'--k');
axis([tspan 0 1])
xlabel('t'), ylabel('x (fraction of vengeful type)')
hold off
```

```
subplot(2,3,3)
plot(t,z(:,3),'--r');
hold on
axis([tspan 0 1])
plot(t,1-e,'--k');
xlabel('t'), ylabel('po=Pr[T|v=0]')
hold off

subplot(2,3,4)
plot(t,z(:,4),'-.r');
hold on
plot(t,1-e,'--k');
axis([tspan 0 1])
xlabel('t'), ylabel('p1=Pr[T|v=1]')
hold off

subplot(2,3,2)
plot(t,z(:,2),'-.g');
hold on
plot(t,v,'--k');
axis([tspan vo-0.5 vo+0.5])
xlabel('t'), ylabel('v')
hold off

subplot(2,3,5)
plot(t,z(:,5),'-.');
hold on
plot(t,1-e,'--k');
axis([tspan 0 1])
xlabel('t'), ylabel('p2=Pr[C|s=1]')
hold off

subplot(2,3,6)
```

```
plot(t,z(:,6),'-.');
hold on
plot(t,qcode,'--k');
axis([tspan 0 1])
xlabel('t'), ylabel('p3=Pr[C|s=0]')
hold off

axes('position',[.01 0.88 1.0 .05],'Box','off','Visible','off');
title(['Panel B: Dynamics of FS game (time average)']);

set(get(gca,'Title'),'Visible','On');
set(get(gca,'Title'),'FontSize',11);
set(get(gca,'Title'),'FontWeight','bold');

figure(2)
subplot(2,1,1)
%plot3(t,z(:,1),z(:,6)); %3-D
plot(y(:,1),y(:,6)); %2-D
hold on
plot(z(length(z),1),z(length(z),6),'r:+');
hold on
plot(xe,qcode,'*k');
axis([0 1 0 1])
xlabel('x'), ylabel('p3=Pr[C|s=0]')

subplot(2,1,2)
plot(y(:,2),y(:,6)); %2-D
hold on
plot(z(length(z),2),z(length(z),6),'r:+');
hold on
plot(v,qcode,'*k');
axis([vo-0.2 vo+0.2 0 1])
hold off
xlabel('v'), ylabel('p3=Pr[C|s=0]')
```

```
axes('position',[.01 0.88 1.0 .05],'Box','off','Visible','off');
title(['Panel A: Dynamics of FS game']);
set(get(gca,'Title'),'Visible','On');
set(get(gca,'Title'),'FontSize',11);
set(get(gca,'Title'),'FontWeight','bold');

%==========================================================================
%On the Dynamic Stability of Bayesian Nash Equilibrium
%Authors: Dan Friedman and Jean Paul Rabanal
%Economics Department
%UC Santa Cruz
%December 2009
% This m.file specifies the adjustment dynamics (ODE system).
%=======================================================================
function dy = venge2(t,y)
dy=zeros(6,1);

% Variables that code uses
global xo e v k wsv ws wos wo c
global betax betav betao beta1 beta2 beta3
global po p1 p2 p3 a emin

%The code uses the following variables
%x = y(1)                                     ' Fraction of vengeful type
%v = y(2)                                     ' Grade of venge
%po = Pr[T|v=0]=  t(1-e)+(1-t)e  = y(3)      ' Prob. of playing T given v=0
%p1 = Pr[T|v=1]=  u(1-e)+(1-u)e  = y(4)      ' Prob. of playing T given v=1
%p2 = Pr[C|s=1]= (1-r)(1-e) + re = y(5)      ' Prob. of playing C given s=1
%p3 = Pr[C|s=0]= (1-q)(1-e) + qe = y(6)      ' Prob. of playing C given s=0
% The prob. are a convex combination between [e,emin].
%See Figure 4 in the FS (2009) paper.

%a  'The misperception prob. is endogenized according to the expression (1)
%in FS (2009): a=0.5*exp(-k*v^2)
```

```
a=0.5*exp(-k*(y(2)^2));
%==========================================================================
%Expected pay-off

%SELFs fitness
%v>0
%wsv= (p1*(1-a)*p2+p1*a*p3)*1 + (p1*(1-a)*(1-p2)+p1*a*(1-p3))*(-1-v)
wsv=(y(4).*(1-a).*y(5)+y(4).*a.*y(6)).*1 +...
 (y(4).*(1-a).*(1-y(5))+y(4).*a.*(1-y(6))).*(-1-y(2));
%v=0
%ws=  (po*(1-a)*p3+po*a*p2)*1 + (po*(1-a)*(1-p3)+po*a*(1-p2))*(-1)
ws=(y(3).*(1-a).*y(6)+y(3).*a.*y(5)).*1 + ...
    (y(3).*(1-a)*(1-y(6))+ y(3).*a.*(1-y(5))).*(-1);

%OTHERs fitness
%s=1
%wos = (x*(1-a)*p1*p2+(1-x)*a*po*p2)*(1) + (x*(1-a)*p1*(1-p2))*(2-v/c) + ...
% ((1-x)*a*po*(1-p2))*2
%s=0
%wo = (x*a*p1*p3+(1-x)*(1-a)*po*p3)*(1) + (x*a*p1*(1-p3))*(2-v/c) + ...
%((1-x)*(1-a)*po*(1-p3))*2

%==========================================================================
%ODE system
% 1 xdot=betax*(1-x)*x*(wsv-ws)
% 2 vdot=betav* (d wsv / d v)
% 3 p0dot= beta0*(1-emin-p(0))*(p(0)-emin)*(partial ws / partial p0)
% 4 p1dot= beta1*(1-emin-p(1))*(p(1)-emin)*(partial wsv / partial p1)
% 5 p2dot= beta2*(1-emin-p(2))*(p(2)-emin)*(partial wos / partial p2)
% 6 p3dot= beta0*(1-emin-p(3))*(p(3)-emin)*(partial wo / partial p3)

dy(1)= betax*(wsv-ws)*(y(1))*(1-y(1));
dy(2)= betav*((-2*y(2)*k*a)*(-y(4)*y(5)+y(4)*y(6)-(1-y(5))*y(4)*(-1-y(2))+...
   y(4)*(1-y(6))*(-1-y(2)))-(y(4)*(1-a)*(1-y(5))+y(4)*a*(1-y(6))));
```

```
% Remember that da/dv= -k*a
dy(3)= betao*(((1-a)*y(6)+a*y(5))*1 + ((1-a)*(1-y(6))+a*(1-y(5)))*(-1))*...
    (y(3)-emin)*(1-emin-y(3));
dy(4)= (y(4)-emin)*(1-emin-y(4))*beta1*...
    (((1-a)*y(5)+a*y(6))*1 + ((1-a)*(1-y(5))+a*(1-y(6)))*(-1-y(2)));
dy(5)= (y(5)-emin)*(1-emin-y(5))*beta2*...
    ((y(1)*(1-a)*y(4)+(1-y(1))*a*y(3))*(1) + ...
    (y(1)*(1-a)*y(4)*(-1))*(2-y(2)/c) + ((1-y(1))*a*y(3)*(-1))*2);
dy(6)= (y(6)-emin)*(1-emin-y(6))*beta3*...
    ((y(1)*a*y(4)+(1-y(1))*(1-a)*y(3))*(1) +...
    (y(1)*a*y(4)*(-1))*(2-y(2)/c) + ((1-y(1))*(1-a)*y(3)*(-1))*2);
end
```