Chaos and Game Theory

"Chaos in the Rock-Paper-Scissors Game"

By Rudy Medina

Summary of Problem:

The game of rock-paper-scissors is played all over the earth by young children. The game is played by having two players simultaneously choose one of three options which are rock, paper, or scissors. The option the player chooses is called the throw. The winner is decided based on the throws of the two individuals. In this game, a throw of rock beats a throw of scissors. A throw of paper beats a throw of rock, and a throw of scissors beats a throw of paper. In the event that both people have the same throw, it can be agreed on that it is a tie or perhaps some other decision can be agreed on by the players prior to the start of the game. In the case where equal throws are called ties, this game is an example of a zero-sum game. By definition, a zero-sum game is a game where a win by one participant is balanced by the loss of the other participant. Let me define a throw by p and its complement by p'. The complement of a throw is the throw that is defeated by using p. The complement of rock is scissors, as rock defeats scissors. In this case, the following is the matrix breakdown of the payouts for all nine possibilities. As an example, $rock_1$ vs. $paper_2$ results in a loss(-1) for $player_1$ and a win(+1) for $player_2$. δ_x and δ_y are the payouts in the event that the players tie.

$$\begin{pmatrix} & Rock_2 & Paper_2 & Scissors_2 \\ Rock_1 & \delta_x, \delta_y & -1,1 & 1,-1 \\ Paper_1 & 1,-1 & \delta_x, \delta_y & -1,1 \\ Scissors_1 & -1,1 & 1,-1 & \delta_x, \delta_y \end{pmatrix}$$

In this problem the sample space is {rock$_1$, paper$_1$, scissors$_1$, rock$_2$, paper$_2$, scissors$_2$}. At the beginning of the game, each of these events can have equal probability of 1/3. If two participants play randomly, then the probabilities at each turn will stay at 1/3 for each of the options. It is this strategy that would satisfy the mixed Nash-Equilibrium strategy for this game. The Nash equilibrium is named after mathematician and Nobel Laureate who first proposed this on his dissertation. If one of the players has chosen a strategy and the other player can't benefit from changing his strategy, then the current strategy and its payoff are the Nash equilibrium. In this game, two players would meet the mixed Nash equilibrium if they randomly choose between the three options with equal probability. Because the game is played repeatedly, players are more likely to employ a strategy than play to the mixed Nash Equilibrium. In fact the rock-paper-scissors association describes the participants as using psychological methods and tendencies of the other person in choosing their next throw. As an example, three consecutive throws of the same type is usually not used by most players. Thus, the compliment of that throw should not be used after two consecutive equal throws. Some players may place greater probabilities on certain throws based on recent throws and their successes or failures of those throws. It is this factors that make the creation of a model complex for this simple game. Different models have been suggested for updating the probabilities of the players in this game. For this project, I chose to use the following model for updating the probability of learning players. The goal of the project was to investigate the probability trajectories for rock and observe its Lyapunov exponents for various strategies.

Learning algorithm used for probability updates:

The model I chose is an iterative model that updates the probability of choosing an event by taking into account the success(or failure) of that event in the last throw and a parameter α called the learning rate. Consequently, the probability of the compliment is also updated, and the probability of the third strategy is one minus the sum of the other two probabilities. The purpose of this project is to investigate the probability trajectories of a person and its connection to dynamical systems and chaos. While the overall distribution of the strategies may be close to 1/3,1/3,1/3 for the different strategies, the probability trajectories may model behavior that is chaotic. Let $throw_t$ be the throw at time t for person number one. The following are the updated probabilities at time (t+1) for $player_1$

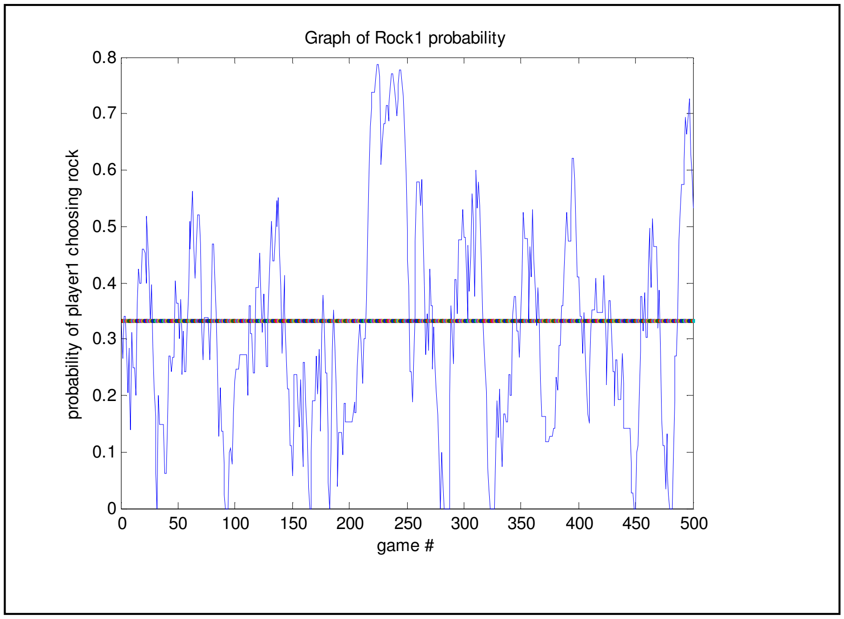
$$
\begin{aligned}
P(rock)_{t+1} &= P(rock)_t + w_t\alpha\left(1-P\left(rock\right)_t\right)\\
P(scissors)_{t+1} &= P\left(scissors\right)_t + \alpha\left(1-w_t\right)\left(1-P\left(scissors\right)_t\right)\\
P\left(paper\right)_{t+1} &= 1-P(rock)_{t+1}-P(scissors)_{t+1}
\end{aligned}
$$

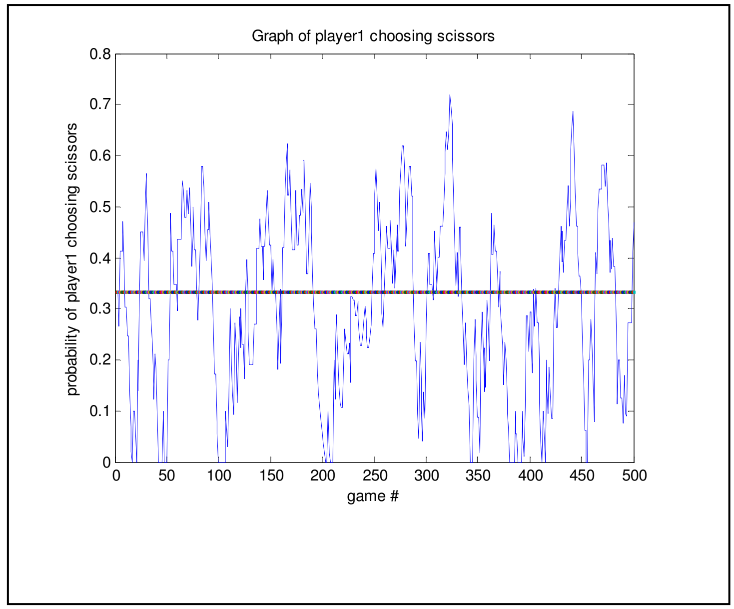
If $throw_t$ =rock and $player_1$ wins game# t, then w_t =1, and the probability of $player_1$ choosing rock at (t+1) will go up by $\alpha\left(1-P\left(rock\right)_t\right)$. Consequently, the probability of $player_1$ choosing scissors, the compliment of rock, will decrease. In all cases w_t =1 represents a success at the nth stage of the game for $throw_t$ and w_t = -1 represents a loss at the t^{th} game . The probability update equations for the other options follow in a similar fashion.

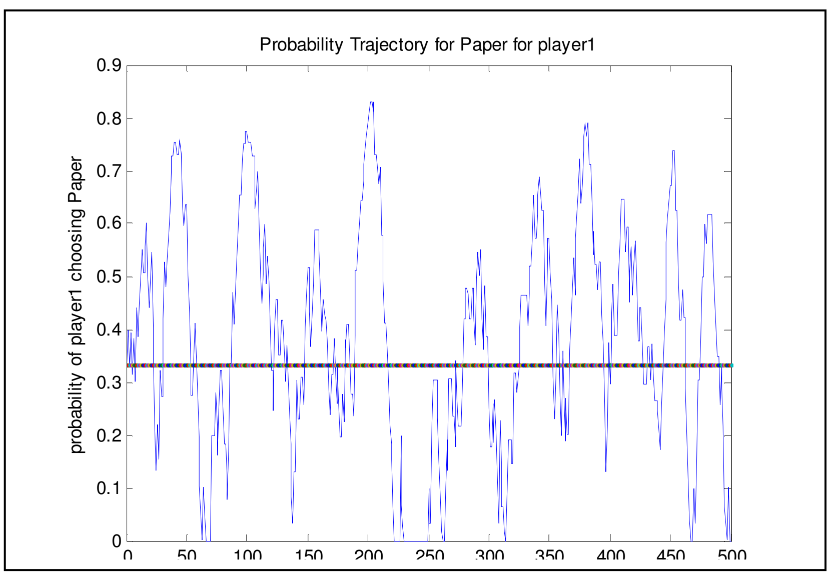
Case 1: Player$_1$ uses the updated probability model and Player$_2$ uses the mixed Nash equilibrium of random throws. Equal throws are considered a tie with a payout=w_t=0 for both participants. The learning rate for player$_1$ is varied and the game is played 500 times in each case. The following tables show how the number of wins of player$_1$ is approximately 1/3 of the total number of games. The distribution of {rock,paper,scissors} in the following is also approximately {1/3,1/3,1/3}. The following tables demonstrate the number of wins and the average Lyapunov exponent for different learning rates for the option of Rock. In all cases, the average number of wins is close to the expected value of this game. A couple of indicators of chaos are a positive Lyapunov exponent and orbits that are not asymptotically periodic. In [1], the term Lyapunov number is used to show the average rate of separation between close points. The definition of a chaotic orbit is an orbit that does not tend toward periodicity and whose Lyapunov number is greater than 1 or whose Lyapunov exponent is positive.

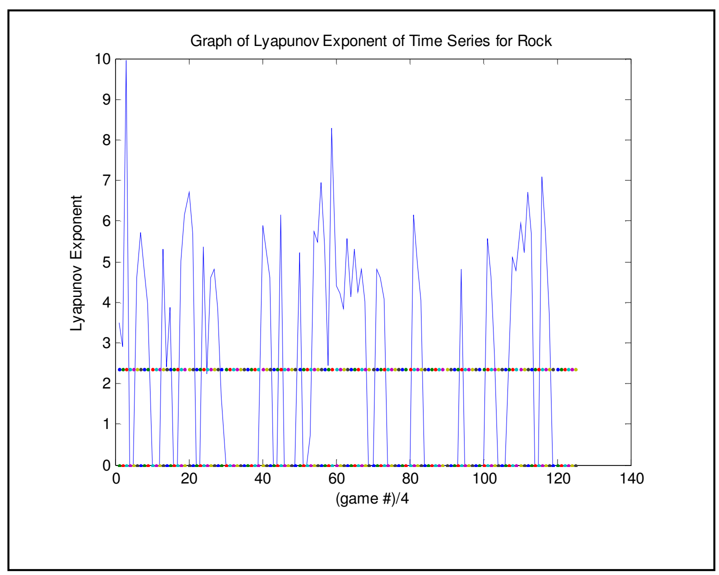
Learning Rate	Average Lyapunov Exponent	Number of Wins by player 1
.1	2.355135831901934	163
.1	2.264368923225087	167
.1	1.914481668820223	186
.1	2.882957294232538	175
.1	1.928999931504547	160
.1	2.321838741644572	167
.1	3.093934052475729	161
.1	1.699753248070484	175
.1	2.686139337364106	162
.1	2.199049831039435	174

Learning Rate	Average Lyapunov Exponent	Standard Deviation of Average Lyapnov Exponent	Average Number of wins	Standard Deviation of average number of wins
.05	2.298942	.310734	175.5	8.910044
.075	2.361211	.180118	164.4	10.4158
.100	2.339034	.366926	163.8	12.96834
.150	2.32206	.446279	168.6	10.88526



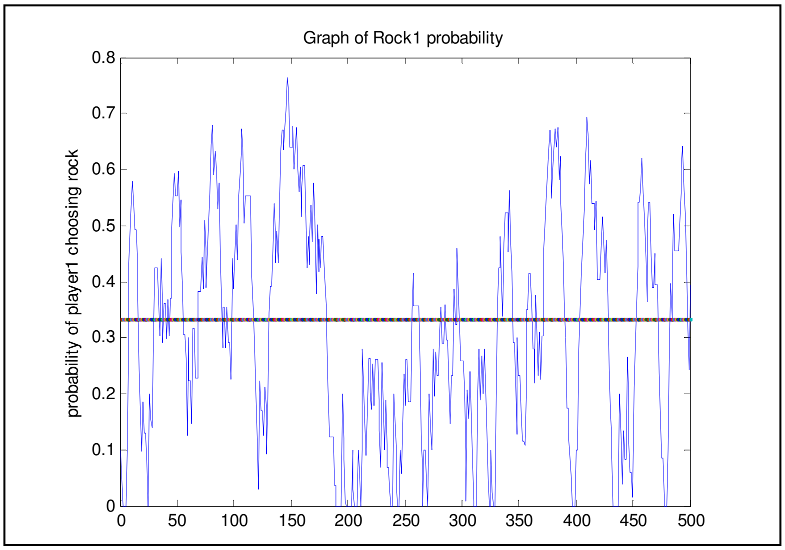


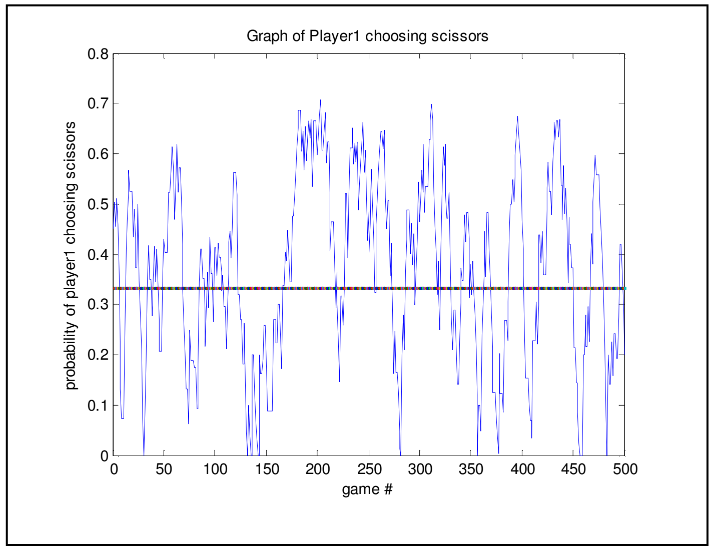


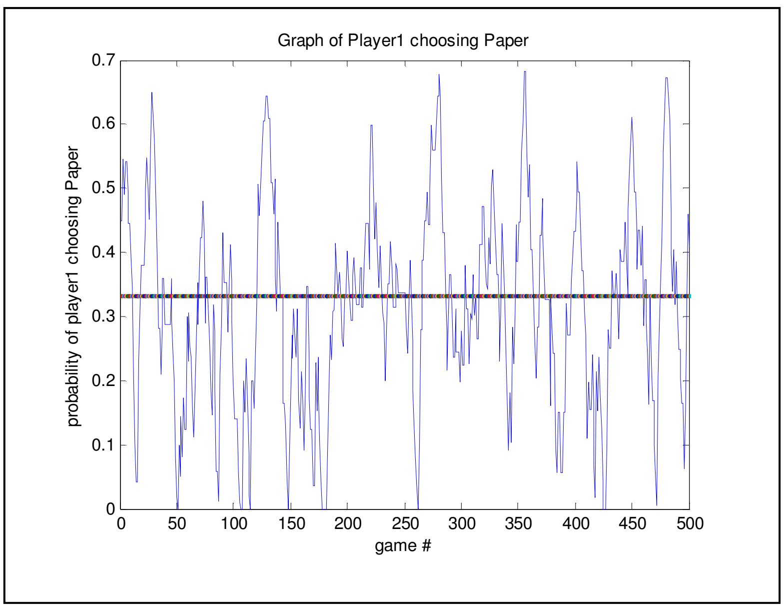


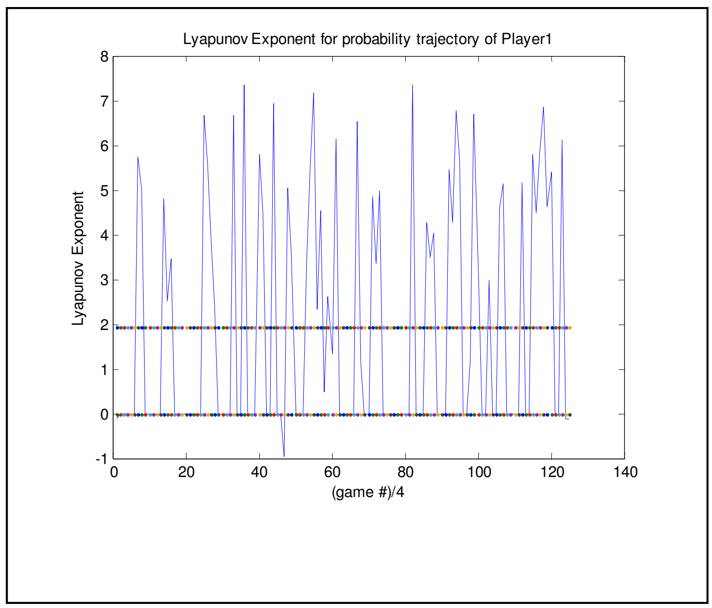
Case 1B:

Player1 uses the learning algorithm and player2 uses the random approach. Ties have the same payout, but player1's initial probability values have been changed to {0.10,0.45,0.45} for rock, paper, and scissors.



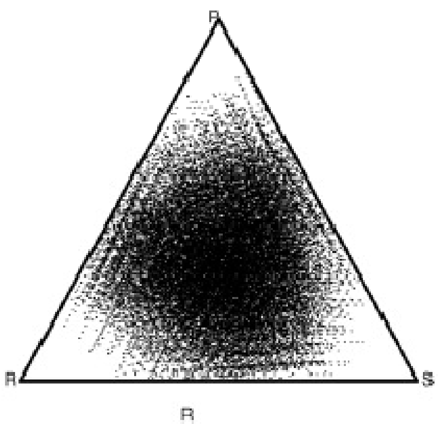


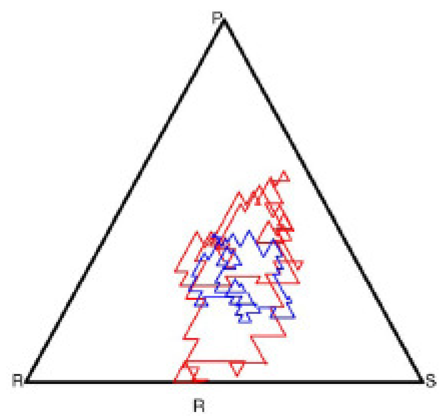




Case 2:

Both players use the learning algorithm as described above. I will investigate the graphs, Lyapunov exponents as the payoff for ties will change. In all cases, the learning rate is kept constant at 0.10. From [6] is the following trajectory of 20,000 points in the probability simplex that arises from two players whose learning rate is 0.10. By visual inspection, it appears that there is random scatter of the probability distributions. The next diagram depicts the same trajectory in short intervals of time. This graph doesn't show random scatter but orbits that are associated more with chaotic behavior.



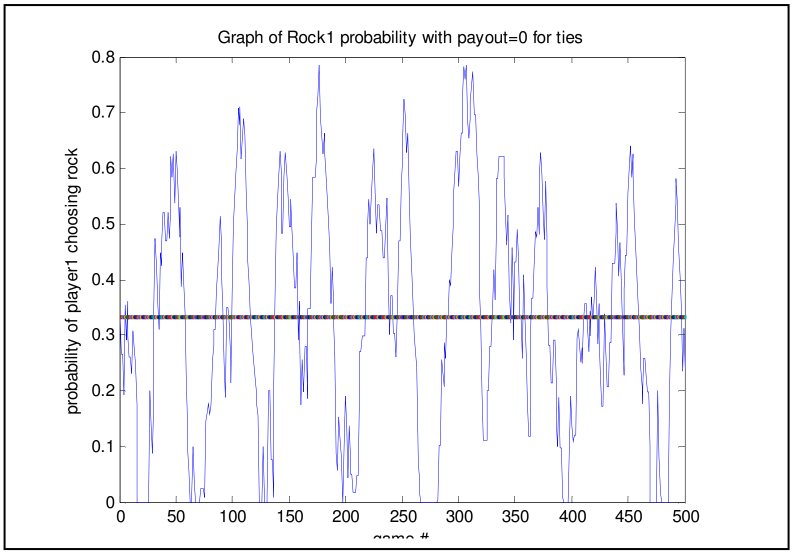


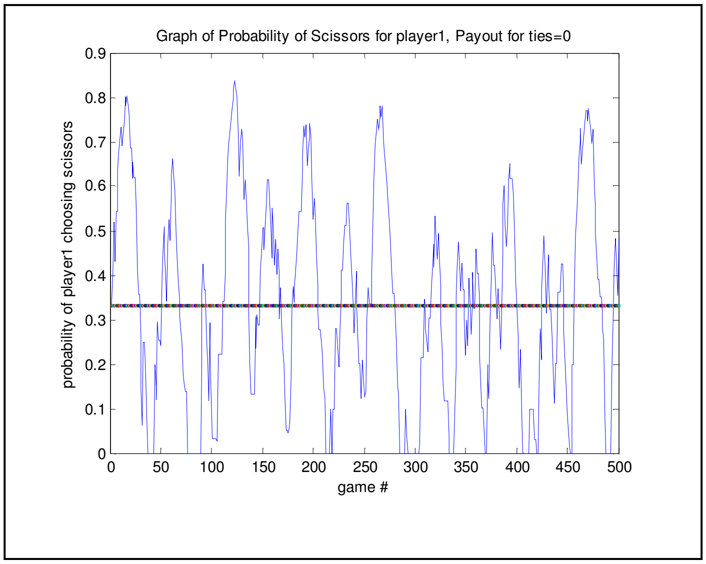
The following is the table for the various payoffs for ties and the average Lyapunov exponent.

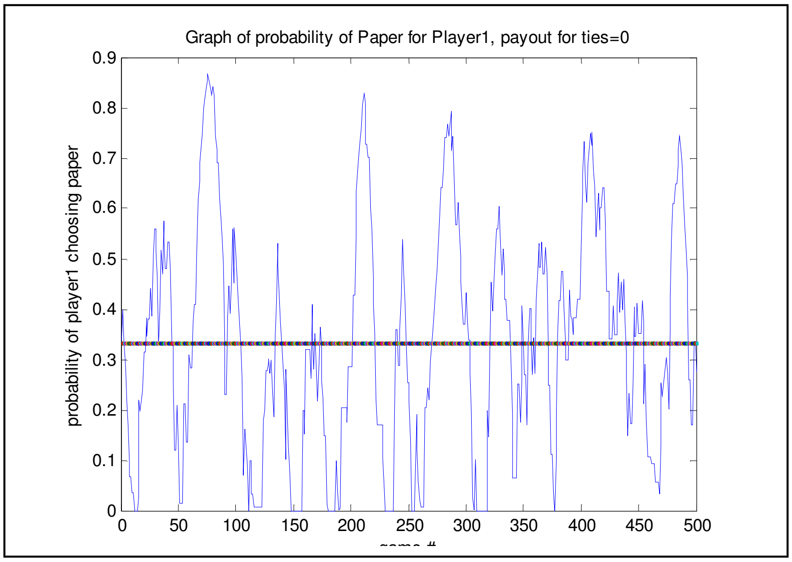
δ_x	δ_y	$\bar{\lambda}$ average Lyapunov Exponent	σ_λ (std. deviation)
0	0	2.301302	.208565
-0.1	0.05	2.762462	.312775
-0.2	0.04	2.886217	.651439
-0.3	0.1	2.797475	.307176
0.3	-0.1	2.83807	.291466

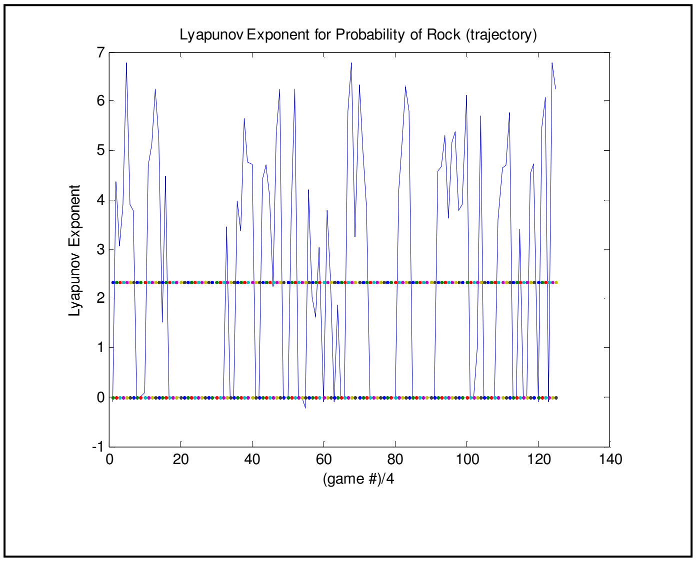
Case 2.A

The following are the graphs of rock, paper, and scissors, respectively for player1. The payoff for ties in this case were equal to 0, i.e. $\delta_x = \delta_y = 0$. The graphs show the trajectories of the probabilities for each option. A straight line at y=1/3 is also graphed. According to the table above, the positive lyapunov exponent indicates that the trajectories of the rock probability are chaotic.



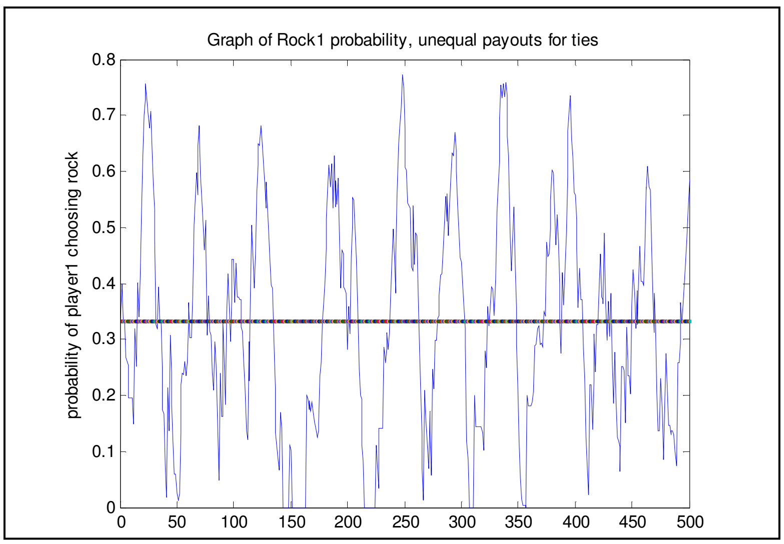


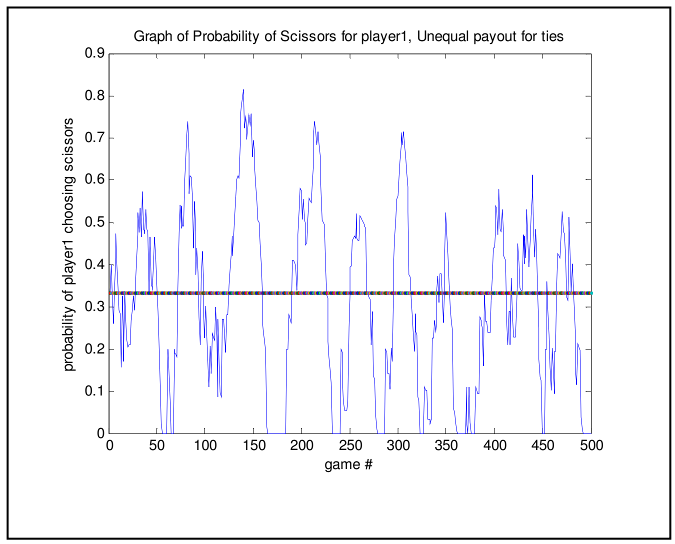


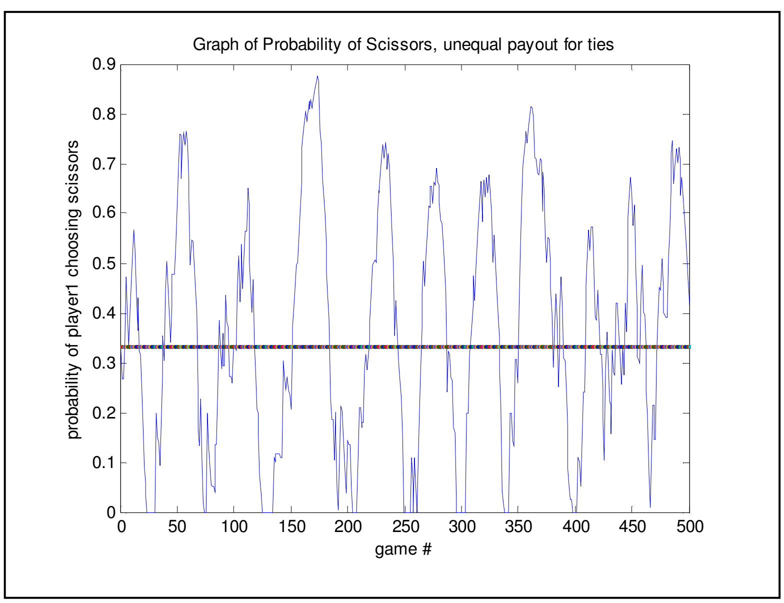


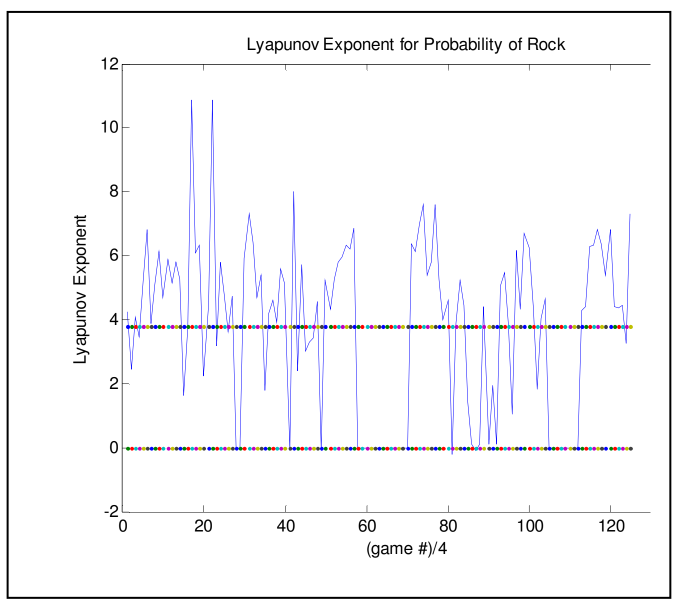
Case 2.B

For the following, assume that $\delta_x = -0.1$ and $\delta_y = 0.05$. Thus $\delta_x + \delta_y < 0$. The Lyapunov exponent is still positive (2.675233851958696), and the graph of the probability of rock appears to have peaks near the top and a valleys near 0. In a similar experiment, Sato, Akiyama, and Farmer used a learning algorithm with the same payoffs for ties. They concluded that the orbits of the probability trajectories appear to be heteroclinic orbits. It appears that if $\delta_x + \delta_y < 0$, this type of behavior will occur. Let x_0 and x_1 be equilibrium points. A heteroclinic orbit is an orbit that is contained in the stable manifold of x_0 and the unstable manifold of x_1.



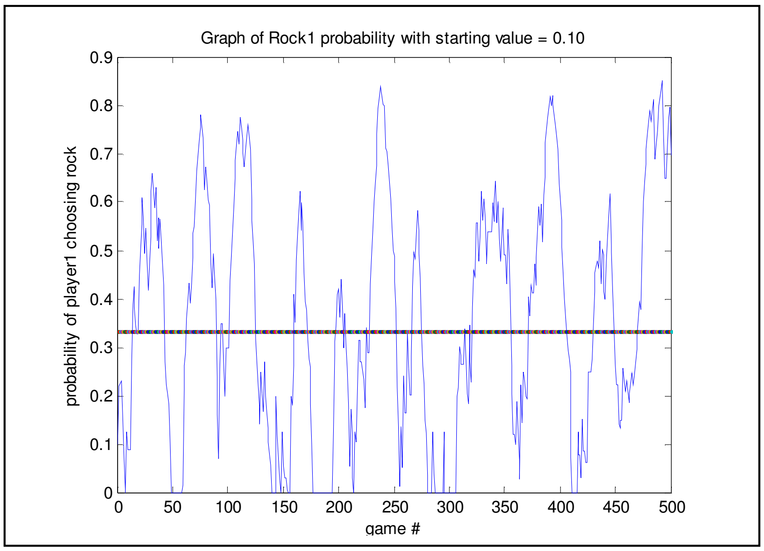


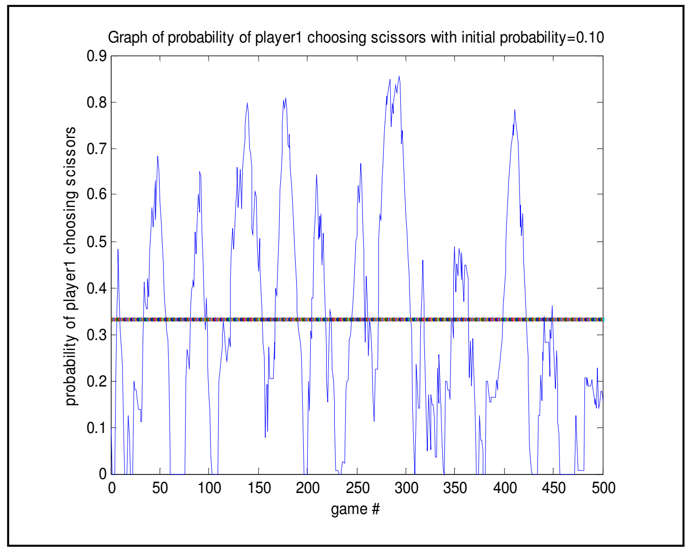


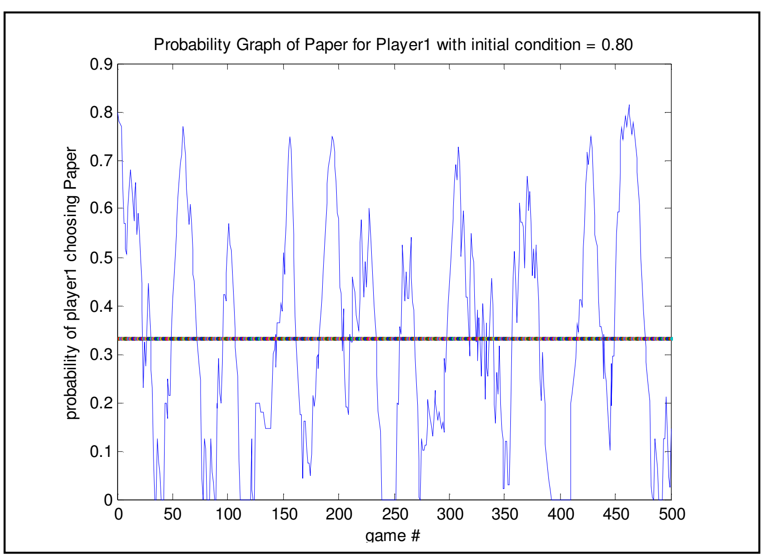


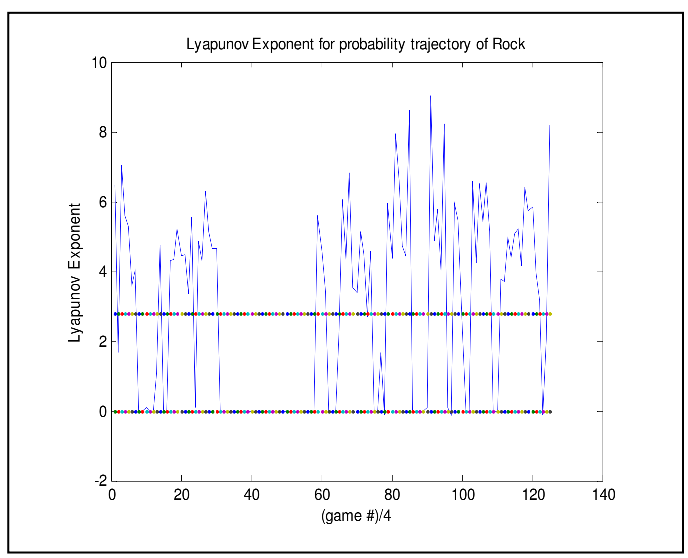
Case 3:

After working with the above, I decided to change the starting probability values and see if there were any changes in the Lyapunov exponents or perhaps in the graphs. The payout for ties is -0.25 and 0.25 respectively for player1 and player2. Five thousand iterations were done in each case. Player2 starting probabilities were held constant {1/3,1/3,1/3}.









Starting rock probability	Starting paper probability	Starting scissors probability	Avg. Lyapunov exponent
.5	.01	.49	4.461717005165243
.5	.05	.45	4.152725462710674
.5	.10	.40	4.401065445226129
.5	.20	.30	4.203167869689596
.1	.45	.45	4.328645065069649
.1	.1	.8	4.346800347790457
.3	.6	.1	4.694951362993801
.3	.1	.6	4.345611505612644
.9	.05	.05	4.274006136791124

Conclusion

My conjecture is that chaos occurs in the probability trajectories when a learning algorithm is used in the game of rock-paper-scissors. I decided on a learning algorithm that both players would use in an attempt to improve their winning percentages. If a throw was successful, then the probability of that same throw would increase and consequently the probability of the compliment would decrease. I first chose to have player1 use the learning algorithm and player2 using random throws. I then chose to have both players use the learning algorithm with the payoff for ties being changed. When the payoff for ties was not equal to 0, the probability trajectories appeared to follow heteroclinic orbits. Lastly, I chose to vary the starting probabilities for player1. In all cases above the Lyapunov exponents were positive, which is a good indicator of expansion of neighboring points and chaos.

References

1) Alligood, Sauer, and Yorke, CHAOS*: An introduction to Dynamical Systems*, Springer, 1996

2) Benaim, Michel & Hofbauer, Josef, *Learning in Games with Unstable Equilibria*, http://www.gtcenter.org/Archive/Conf06/Downloads/Conf/Hopkins121.pdf

3) Cason, Friedman, and Hopkins, *Testing the TASP: An experimental Investigation of Learning in Games with Unstable Equilibria*, http://www.econ.ed.ac.uk/papers/TASPtest.pdf

4) http://william-king.www.drexel.edu/top/eco/game/nash.html

5) http://www.math.tamu.edu/~mpilant/math614/

6) Salvetti, Franco & Patelli, Paolo & Nicolo,Simone, *Chaotic Time Series Prediction for the Game Rock-Paper-Scissors*, University of Colorado at Boulder, 2006

7) Sato, Yuzuro & Akiyama, Eizo & Farmer,J. , *Chaos in Learning a Simple Two-Person Game,* Proceedings of the National Academcy of Science of the United States of America, 2002

Appendix: Matlab Code for RPS (rock-paper-scissors)

```
function Finalproj=rps
% Rudy Medina, Math 614, Spring 2009
% This program runs the game of rock-paper-scissors
% it uses a learning algorithm for both players
% it graphs the trajectories of the learned probabilities
% and computes the lyapunov exponent using Dr. Pilant's
% method.

format long
N=20000;

pplay1r=zeros(N,1);
pplay1s=zeros(N,1);
pplay1p=zeros(N,1);
pplay2r=zeros(N,1);
pplay2s=zeros(N,1);
pplay2p=zeros(N,1);
throw1=zeros(N,1);
throw2=zeros(N,1);
win1=zeros(N,1);
win2=zeros(N,1);
x=zeros(N,1);
y=zeros(N,1);
data=zeros(N,1);
g=0;
rock1=0;
rock2=0;
paper1=0;
paper2=0;
scissors1=0;
scissors2=0;
totalwins1=0;
totalwins2=0;
lyp=zeros(N,1);
total=0;

% initial probability conditions for both players
pplay1r(1)=0.1;
pplay1s(1)=.1;
pplay1p(1)=.8;
pplay2r(1)=1/3;
pplay2s(1)=1/3;
pplay2p(1)=1/3;
```

```matlab
for i=1:500   % began the iteration of the games
    a=.1;     % learning rate
    z=randi(1000,1);   %choose a random number from 1-1000
    g=z/1000;
    b=randi(1000,1);   %choose a random number from 1-1000
    h=b/1000;

    data(i)=pplay1r(i);  %store the rock probability trajectories



    % the probability of player1 throwing a rock
    if g < pplay1r(i)
        throw1(i)=1;
        rock1=rock1+1;
    end;

    % the probability of player1 throwing Paper
    if g > pplay1r(i) & g < (pplay1r(i)+pplay1p(i))
        throw1(i)=2;
        paper1=paper1+1;
    end;

    % the probability of player1 throwing scissors
    if g > (pplay1r(i)+pplay1p(i))
        throw1(i)=3;
        scissors1=scissors1 + 1;
    end;

    % the probability of player2 throwing rock
    if h < pplay2r(i)
        throw2(i)=1;
        rock2=rock2+1;
    end;

    % the probability of player2 throwing paper
    if h > pplay2r(i) & h < (pplay2r(i)+pplay2p(i))
        throw2(i)=2;
        paper2=paper2+1;
    end;

    % the probability of player2 throwing scissors
    if h > (pplay2r(i)+pplay2p(i))
        throw2(i)=3;
        scissors2=scissors2 + 1;
    end;



    %Payout for a tie between both players
    if throw1(i)==throw2(i)
        win1(i)=-0.25;
        win2(i)=0.25;
```

```
    end;


    %Payout for rock vs scissors
    if throw1(i)==1 & throw2(i)==3
        win1(i)=1;
        win2(i)=-1;
        totalwins1=totalwins1+1;
    end;

    %Payout for rock vs paper
    if throw1(i)==1 & throw2(i)==2
        win1(i)=-1;
        win2(i)=1;
        totalwins2=totalwins2+1;
    end;

    %Payout for paper vs rock
    if throw1(i)==2 & throw2(i)==1
        win1(i)=1;
        win2(i)=-1;
        totalwins1=totalwins1+1;
    end;

    %Payout for paper vs scissors
    if throw1(i)==2 & throw2(i)==3
        win1(i)=-1;
        win2(i)=1;
        totalwins2 = totalwins2 + 1;
    end;

    %Payout for scissors vs rock
    if throw1(i)==3 & throw2(i)==1
        win1(i)=-1;
        win2(i)=1;
        totalwins2=totalwins2+1;
    end;

    %Payout for scissors vs paper
    if throw1(i)==3 & throw2(i)==2
        win1(i)=1;
        win2(i)=-1;
        totalwins1=totalwins1+1;
    end;


  %The following is the learning algorithm for each player
% probability update for Rock for player 1
    if throw1(i)==1
        pplay1r(i+1)=pplay1r(i)+win1(i)*(a*(1-pplay1r(i)));
        pplay1s(i+1)=pplay1s(i)+(1-win1(i))*a*(1-pplay1s(i));

            if pplay1r(i+1)<0
                pplay1r(i+1)=0;
            end;
```

```
            if pplay1s(i+1)<0
                pplay1s(i+1)=0;
            end;

        if pplay1r(i+1) + pplay1s(i+1) > 1
            pplay1s(i+1)= 1 - pplay1r(i+1);

        end;
        pplay1p(i+1)= 1 - pplay1r(i+1) - pplay1s(i+1);

    end;


% probability update for Paper for player 1
    if throw1(i)==2
        pplay1p(i+1)=pplay1p(i)+win1(i)*(a*(1-pplay1p(i)));
        pplay1r(i+1)=pplay1r(i)+(1-win1(i))*a*(1-pplay1r(i));

            if pplay1p(i+1)<0
                pplay1p(i+1)=0;
            end;

            if pplay1r(i+1)<0
                pplay1r(i+1)=0;
            end;

        if (pplay1p(i+1)+pplay1r(i+1))>1
            pplay1r(i+1)=1-pplay1p(i+1);
        end;

        pplay1s(i+1)= 1 - pplay1p(i+1) - pplay1r(i+1);
    end;

% probability update for scissors for player 1
    if throw1(i)==3
        pplay1s(i+1)=pplay1s(i)+win1(i)*(a*(1-pplay1s(i)));
        pplay1p(i+1)=pplay1p(i)+(1-win1(i))*a*(1-pplay1p(i));

            if pplay1s(i+1)<0
                pplay1s(i+1)=0;
            end;

            if pplay1p(i+1)<0
                pplay1p(i+1)=0;
            end;

        if (pplay1s(i+1)+pplay1p(i+1))>1
            pplay1p(i+1)=1-pplay1s(i+1);
        end;

        pplay1r(i+1)= 1 - pplay1s(i+1) - pplay1p(i+1);
    end;
```

```
%probability update for Rock for player 2
    if throw2(i)==1
        pplay2r(i+1)= pplay2r(i)+win2(i)*(a*(1-pplay2r(i)));
        pplay2s(i+1)=pplay2s(i)+(1-win2(i))*a*(1-pplay2s(i));

        if pplay2r(i+1)<0
                pplay2r(i+1)=0;
            end;

            if pplay2s(i+1)<0
                pplay2s(i+1)=0;
            end;

        if pplay2r(i+1) + pplay2s(i+1) > 1
            pplay2s(i+1)= 1 - pplay2r(i+1);

        end;

        pplay2p(i+1)= 1 - pplay2r(i+1) - pplay2s(i+1);


    end;


%probability update for paper for player 2
    if throw2(i)==2
        pplay2p(i+1)=pplay2p(i)+win2(i)*(a*(1-pplay2p(i)));
        pplay2r(i+1)=pplay2r(i)+(1-win2(i))*a*(1-pplay2r(i));

        if pplay2p(i+1)<0
                pplay2p(i+1)=0;
            end;

            if pplay2r(i+1)<0
                pplay2r(i+1)=0;
            end;

        if (pplay2p(i+1)+pplay2r(i+1))>1
            pplay2r(i+1)=1-pplay2p(i+1);
        end;

        pplay2s(i+1)= 1 - pplay2p(i+1) - pplay2r(i+1);
    end;


%probability update for scissors for player 2
    if throw2(i)==3
        pplay2s(i+1)=pplay2s(i)+win2(i)*(a*(1-pplay2s(i)));
        pplay2p(i+1)=pplay2p(i)+(1-win2(i))*a*(1-pplay2p(i));

        if pplay2s(i+1)<0
                pplay2s(i+1)=0;
```

```
            end;

            if pplay2p(i+1)<0
                pplay2p(i+1)=0;
            end;

        if (pplay2s(i+1)+pplay2p(i+1))>1
            pplay2p(i+1)=1-pplay2s(i+1);
        end;

        pplay2r(i+1)= 1 - pplay2s(i+1) - pplay2p(i+1);
    end;




end;


fprintf('rock \t  scissors \t   paper\n')
for w=1:i
    fprintf('%f\t  %f\t %f\t %f\t %f\t
%f\n',pplay1r(w),pplay1s(w),pplay1p(w),pplay1r(w)+pplay1s(w)+pplay1p(w),throw
1(w),win1(w));
end;



m=1:i;
pplay2r(m);
pplay2s(m);
pplay2p(m);
plot(m,pplay1r(m),m,.333333);
% Title for the graphs
title(sprintf('Graph of Rock1 probability'))
pause;
plot(m,pplay1s(m),m,.3333333);
pause;
plot(m,pplay1p(m),m,.3333333);
pause;
plot(pplay1p(m),pplay1s(m),'*');
pause;

% computing the Lyapunov Exponents
N2 = floor(i/2);
N4 = floor(i/4);
% find mid point 'of orbit sequence
k=N2;
% create space for exponents
exponent = zeros(N4,1);
% look at 1/4 of the points
for (j=1:N4)
    % set distance initially
    d = abs(data(k+1)-data(k));
    index = k+1;
    for (i=2:N-1)
        % see if there is a closer point
```

```
        if (i ~= k) && (abs(data(i)-data(k)))<d
            % if so, store index and distance
            d = abs(data(i)-data(k));
            index = i;
        end
    end
    % write log of quotient as difference of logs to get better accuracy!
    if (data(k) ~= data(index)) && (data(k+1) ~= data(index+1))
        exponent(j) = log( abs(data(k+1)-data(index+1)))-log(abs(data(k)-
data(index)));
    end
    % repeat with the next point
    k = k+1;
end

% now plot the lyapunov exponents
t = 1:N4;
lyapunov = exponent(1:N4);
exp_avg = 0.0;
% find the average value for lyapunov exponent
for (i=1:N4)
    exp_avg = exp_avg + exponent(i);
end


% plot the exponents, the average, and the baseline
exp_avg = exp_avg/N4;
plot(t,lyapunov,t,0,t,exp_avg);
display 'average value for lyapunov exponent is:'
exp_avg
paper1
    scissors1
    rock2
     totalwins1
```