PAVLOV AND THE PRISONER'S DILEMMA

By

David Kraines

and

Vivian Kraines

IMA Preprint Series # 369

December 1987

INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

514 Vincent Hall

206 Church Street S.E.

Minneapolis, Minnesota 55455

PAVLOV AND THE PRISONER'S DILEMMA

by David Kraines* and Vivian Kraines**

Abstract: Our Pavlov learns by conditioned response: through rewards and punishments, to cooperate or defect. We analyze the behavior of an extended play Prisoner's Dilemma with Pavlov against various opponents and compute the time and cost to train Pavlov to cooperate. Among our results is that Pavlov and his clone would learn to cooperate more rapidly than if Pavlov played against the Tit for Tat strategy. This fact has implications for the evolution of cooperation.

1. Introduction

Our Pavlov is an individual who learns to respond by positive and negative reinforcement. Given two choices, he will initially respond to them randomly. If he is rewarded for choosing A, then he will be somewhat more likely to choose A again. Similarly if he is punished for choosing B, then he will be somewhat less likely to choose B again and thus more likely to choose A. To ensure that Pavlov will consistently choose A, he must be given a succession of rewards and punishments to reinforce this choice.

The Prisoner's Dilemma that we will consider in this paper is given by the payoff matrix

$$(1)\qquad \text{Opp}\quad \begin{array}{c} \\ C \\ D \end{array}\begin{array}{c} \text{Pavlov} \\ \begin{array}{cc} C & D \end{array} \\ \left|\begin{array}{cc} 1,1 & -2,2 \\ 2,-2 & -1,-1 \end{array}\right| \end{array}$$

This means that that if Opp and Pavlov both choose C, (so that the outcome is CC), then each gets one unit of reward. If Opp chooses C but Pavlov chooses D, (with a CD outome), then Opp Y loses 2 units while Pavlov gains two units. Each player can maximize his return or minimize his loss if he plays the D or defect strategy.

* Dept of Mathematics, Duke University and University Minnesota
** Dept of Mathematics, Meredith College and Inst Math Applications

The payoff for this equilibrium strategy is -1 to each rather than the +1 that each would receive were the outcome CC. In an extended series of plays between the same players, other strategies improve the chances that players will cooperate and so that each will gain. Perhaps the most simple and most successful of these strategies is Tit for Tat. TT starts the game by cooperating and thereafter plays what his opponent played the previous round. This strategy has repeatedly "won" against many other rational strategies. See [Ra], and [Ax] for a history of this problem as well as an in depth discussion and a large bibliography.

Our Pavlov is not, however, a rational player. He learns, gradually, to respond in a particular way through a series of rewards and punishments, but does not think or plan ahead. How can Pavlov be trained to cooperate? What is the most efficient method of training? What is the cost to Pavlov and to the trainer?

We answer these questions for seven opponents in a quantitative manner. The opponents we study are:

Stalin	who always defects
Monte	with totally random behavior
Gandhi	who always cooperates
TT	who uses the Tit for Tat strategy
PT	our perfect trainer, who has deep psychological insight
NT	a normal trainer, without the special talent of PT
Pavlov	a clone of Pavlov

In our model, the last four opponents will eventually be able to train Pavlov to cooperate with certainty. We also show that it is foolish to try to double cross a trained Pavlov by defecting even once. In most cases the Tit for Tat strategy takes longer and is more expensive than the others. Indeed in the Pavlov vs

Pavlov model, each player trains the other by this conditioned response method even faster than TT would train Pavlov. One could argue that this is a suggestive, if simplistic, model for the evolution of cooperation in a biological environment [AH].

Our method is to model this extended Prisoner's Dilemma as a finite Markov chain with states representing the probability that Pavlov will cooperate. Standard techniques in matrix algebra will then produce the training time and cost for the different opponents. The matrix algebra and the tables and figures in this paper were generated using PC-MATLAB [ML], a powerful programmable matrix manipulator. The programs that we have used are included in the appendix to this paper as well as the figures that MATLAB has generated.

2. Preliminaries

At any given time, Pavlov will have a certain propensity to cooperate. He may be certain to defect, certain to cooperate or make his choice completely randomly. Let p be this probability state. So p=0 means that Pavlov will certainly defect, p=1 means that he will certainly cooperate, and p=1/2 means that he could flip a coin to decide. The greater the value of p, the more likely he is to cooperate.

First let us examine how the outcome of a play affects Pavlov's probability state. Assume that before a given play, Pavlov will cooperate with probability p. If the outcome is CC, then Pavlov will be more likely to cooperate on the next play. We need to quantize this greater likeliness.

The **learning increment** for Pavlov is defined to be a fixed positive number x. We assume that by collecting this one unit of reward for the CC outcome, the probability that Pavlov cooperates on the next play increases from p to p+x, or more precisely, the minimum of p+x and 1 since no probability can be larger than 1. In this paper we will consider learning increments of x ranging from 1/4 to 1/30 with emphasis on x=1/10. We will also restrict attention to those values of p which can be written k/n for some integer k from 0 to n. Thus, with x=1/10, if Pavlov was only 30% likely to cooperate and yet both he and his opponent did cooperate, then on the next play he will be 40% likely to cooperate.

If the result of the play is DD then Pavlov is given 1 unit of punishment for defecting. This negative reinforcement will make him less likely to defect, and so more likely to cooperate, on the next play. We assume that it increases his probability of cooperation to min(p+x,1), precisely as in the CC outcome. On the other hand, the CD and DC states will, perversely, reward Pavlov 2 units for defecting or punish him 2 units for cooperating and thus will lower his probability of cooperating next round to max(p-2x,0). In the 30% example above, the CD or DC outcome will reduce Pavlov's probability to cooperate to 10%. If this is followed by still another CD or DC outcome, then the new probability will be 0%; Pavlov will defect with certainty.

The long term effect on Pavlov's propensity to cooperate can be modelled by a Markov chain whose n+1 states, the **probability states**, {0, 1/n, 2/n,...,1}, represent the probability that Pavlov will cooperate on the next play. The Markov transition matrix is the (n+1)x(n+1) matrix, P, whose (i,j)th entry is the probability that Pavlov will change from state i/n to state j/n. Here i and j run from 0 to n and represent the probability state i/n and j/n respectively. Thus if n=10 and p=0.3, then the (3,4) entry will represent the probability that the outcome of the play was CC or DD which means that Pavlov's probability state increases from 0.3 to 0.4. The (3,1) entry will represent the probability that the outcome was DC or CD, causing p to decrease to 0.1. All other entries in that row are 0. Note that the sum of the two nonzero entries in each row is 1.

This Markov matrix could be described more succinctly if we introduce some notation. Let pr(CC,p) be the probability that the outcome of a play is CC given that Pavlov will cooperate with probability p. This number depends, of course, on the strategy of Pavlov's opponent. Similarly define pr(CD,p), pr(DC,p), and pr(DD,p). The probability that Pavlov's probability state will increase from p to p+x is thus

$$(2) \quad Tr^{+}(p) = pr(CC,p) + pr(DD,p).$$

This sum is the transition probability which will appear one step to the right of the diagonal in the Markov matrix, e.g. in the (i/n,(i+1)/n) place. The only other non zero element in the i^{th} row of this Markov matrix is two units to the left of the diagonal, in the (i/n,(i-2)/n), representing a change from p to p-2x. This occurs with probability

$$(3) \quad Tr^{-}(p) = pr(CD,p) + pr(DC,p).$$

These formulas must be adjusted at the top and the bottom of the matrix since probability states must be between 0 and 1.

Using Markov chains one can make statements about the long term variance of p. An absorbing state is one which once entered is never left. If p=1, so Pavlov cooperates with certainty, then we say that Pavlov is trained. If Pavlov's opponent will cooperate whenever Pavlov is trained then the new p is also 1, i.e., the p=1 state is absorbing. Using matrix algebra one can easily compute the expected number of plays until arriving at the absorbing state given that Pavlov starts in state p. See Chapter 3 [KS] and section 5 below for more details.

Two of our opponents, Stalin and Monte, will not always cooperate when Pavlov is trained. For these models, there is no absorbing state. Instead we compute the equilibrium vector which represents the proportion of time that Pavlov is in any given state p in the long run. When Pavlov plays Gandhi, two absorbing states are possible. We will compute the likelihood of ending in the trained state as opposed to the CD state. For each of the other four opponents, there will be a single CC absorbing state, the state p=1 in which Pavlov is trained.

3. Stalin vs. Pavlov

Elementary analysis of the single play Prisoner's Dilemma shows that a rational player does better by defecting, although this is not so in an extended game. Our Stalin will always defect regardless of his opponent's actions. Thus Pavlov will be punished heavily for cooperating (DC), thereby reducing p, and also punished, though more lightly, for defecting (DD), thereby increasing p. Pavlov will play, suffer, change his probability state, play again and suffer some more.

Let us now consider this model quantitatively. Assume that Pavlov's learning increment is x=1/n. If Pavlov is in state p=k/n, then with probability k/n he will move to state (k-2)/n (DC) while with probability 1-k/n he will move to state (k+1)/n (DD). Outcomes CC and CD are not possible as Stalin never cooperates. For n=10, the Markov matrix is given as Table 1.

		Probability for next play										
	Prob	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
c												
u	0.0	0	1.0	0	0	0	0	0	0	0	0	0
r	0.1	0.1	0	0.9	0	0	0	0	0	0	0	0
r	0.2	0.2	0	0	0.8	0	0	0	0	0	0	0
e	0.3	0	0.3	0	0	0.7	0	0	0	0	0	0
n	0.4	0	0	0.4	0	0	0.6	0	0	0	0	0
t	0.5	0	0	0	0.5	0	0	0.5	0	0	0	0
_	0.6	0	0	0	0	0.6	0	0	0.4	0	0	0
p	0.7	0	0	0	0	0	0.7	0	0	0.3	0	0
r	0.8	0	0	0	0	0	0	0.8	0	0	0.2	0
o	0.9	0	0	0	0	0	0	0	0.9	0	0	0.1
b	1.0	0	0	0	0	0	0	0	0	1.0	0	0

Table 1 MARKOV MATRIX n=10 Stalin vs Pavlov

For example, if p=0.3, then Pavlov will cooperate with 30% probability. If he does cooperate, then he will be punished two units and so will only cooperate with 10% probability on the next play. If he defects (70% likely) then he is punished and so will cooperate with 40% probability on the next play. Thus in the row labelled 0.3, we see an entry of 0.3 under 0.1, representing the probability that Pavlov's probability state will be lowered from

30% to 10%, and we see an entry of 0.7 under the 0.4, representing the probability that Pavlov will defect and thus move to the 40% likely state.

This Markov process is called ergodic, meaning that Pavlov can get to any probability state starting from any other. For such processes, one can compute the equilibrium probability vector $R=(r_o,r_1,...,r_n)$ where r_k represents the proportion of plays for which Pavlov will be in state k/n. The sum of the entries of this vector must be 1 since he must be in some state. It is well known that this vector is independent of the initial value of p. R may be computed as the unique positive solution to the vector matrix equation RP = R which also satisfies sum(R) = 1, or as the normalized left eigenvalue of P with eigenvalue 1 (Chapter 4, [KS]).

Let p = i/n and let $r^p = r_i$, the proportion of plays in which the probability state is p. Define the payoff function to Pavlov by

(4) $\quad \mathrm{PyP}(p) = \mathrm{pr}(CC,p) - \mathrm{pr}(DD,p) + 2\mathrm{pr}(CD,p) - 2\mathrm{pr}(DC,p)$

and $\mathrm{PyP} = \mathrm{sum}\ r_p * \mathrm{PyP}(p)$

This quantity measures the units of reward less the units of punishment that Pavlov can expect on the average per play. The quantity, PyP(p) is the expected payoff given that Pavlov is in state p.

(5) $\quad \mathrm{PyO}(p) = \mathrm{pr}(CC,p) - \mathrm{pr}(DD,p) - 2\mathrm{pr}(CD,p) + 2\mathrm{pr}(DC,p)$

and $\mathrm{PyP} = \mathrm{sum}\ r_p * \mathrm{PyO}(p)$

The difference between PyP and PyO is in the sign of the CD and DC terms, since each receive the same payoff for CC and DD but opposite payoffs for CD and DC.

For example, the estimated payoff per play to Pavlov and to Stalin, i.e., the total rewards and punishments averaged over many plays, is computed as follows. Assume that Pavlov is in state p. Then on the average, Pavlov will lose one unit if the outcome is DD, which occurs with probability 1-p, i.e. pr(DD,p) = 1-p. He will lose 2 units for the DC outcome which occurs with probability p. Thus he can expect to gain PyP(p) = -(1-p) - 2p = -1-p units. Note that pr(CC,p) = pr(CD,p) = 0 as Stalin never cooperates. We multiply this number by the proportion of time he is in state p, which we read off from the equilibrium vector, and then add up over all states. The payoff to Stalin, on the other hand is -1 for DD and +2 for DC, so that he can expect to win PyO(p) = -(1-p) + 2p = -1 + 3p units times the proportion of time Pavlov is in state p, summed over all states.

For n = 6 to 16, we compute and display the equilibrium vectors in the column labelled by the corresponding n in Table 2. The average payoff to each player for each n is displayed below the respective column.

n, learning increment = 1/n

.		6	7	8	9	10	11	12	13	14	15	16
%												
	0/n	.13	.10	.08	.06	.05	.04	.03	.02	.02	.01	.01
e	1/n	.23	.20	.16	.13	.11	.09	.07	.06	.05	.04	.03
n	2/n	.27	.25	.23	.20	.18	.15	.13	.11	.09	.08	.07
d	3/n	.21	.22	.23	.22	.21	.20	.18	.16	.14	.13	.11
		.11	.14	.17	.19	.19	.20	.19	.19	.18	.16	.15
i	e	.04	.00	.09	.12	.14	.16	.17	.17	.18	.17	.17
n	t	.01	.02	.04	.06	.08	.10	.12	.13	.15	.15	.16
	c		.00	.01	.02	.03	.05	.07	.08	.10	.12	.13
s	.			.00	.00	.01	.02	.03	.04	.06	.07	.09
t	.				.00	.00	.01	.01	.02	.03	.04	.05
a	.					.00	.00	.00	.01	.01	.02	.02
t							.00	.00	.00	.00	.01	.01
e								.00	.00	.00	.00	.00
									.00	.00	.00	.00
										.00	.00	.00
											.00	.00
												.00
average payoff per play to:												
Pavlov		-1.35	-1.34	-1.34	-1.34	-1.34	-1.34	-1.34	-1.33	-1.33	-1.33	-1.33
Stalin		0.04	0.03	0.02	0.01	0.011	0.008	0.006	0.005	0.003	0.003	0.002

Table 2 EQUILIBRIUM VECTORS Stalin vs Pavlov

Figure 1 in Appendix A plots the values of the equilibrium vectors for various values of n. Note that the distribution of p is centered about p=1/3, meaning that Pavlov will tend to cooperate only 1/3 of the time. Although it cannot be seen from the above chart due to rounding off, Pavlov will be trained, i.e., p will be 1, for a small proportion of the time. For n=10, this proportion is 0.0002 or 1/500 plays. Note that from Figure 2 Stalin ends up with a near 0 payoff forever while Pavlov loses 1.33 units per play, worse than if he were to continually defect. This is consistent with the payoff that would accrue were Pavlov permanently in state 1/3, i.e. if he used a 1/3 mixed strategy.

4. Monte vs Pavlov

Our Monte randomly cooperates and defects regardless of what his opponent does. In this game, as against Stalin, there is no absorbing state, for if Pavlov were temporarily trained, then Monte will defect, though not necessarily on the next play. This defection will cause Pavlov to lower his probability of cooperating and thus to become untrained.

The computation of the entries of the Markov chain is somewhat different from the Stalin model as now all four possible outcomes will occur if p is not 0 or 1. Recall that p will be increased, that cooperation will be reinforced, if the outcome of the play is CC or DD. As Pavlov will cooperate with probability p and Monte with probability 1/2, it is clear that CC occurs with probability p/2. Similarly DD occurs with probability (1-p)/2. Thus one or the other occur with probability

$$(6) \quad Tr^{+}(p) = p/2 + (1-p)/2 = 1/2.$$

Likewise, CD or DC must occur with probability 1/2. After making the appropriate adjustments for p = 0 and 1, we see that the Markov matrix for Monte vs Pavlov and n=10 is as in Table 3.

Probability for next play

current prob

Prob	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.0	0.5	0.5	0	0	0	0	0	0	0	0	0
0.1	0.5	0	0.5	0	0	0	0	0	0	0	0
0.2	0.5	0	0	0.5	0	0	0	0	0	0	0
0.3	0	0.5	0	0	0.5	0	0	0	0	0	0
0.4	0	0	0.5	0	0	0.5	0	0	0	0	0
0.5	0	0	0	0.5	0	0	0.5	0	0	0	0
0.6	0	0	0	0	0.5	0	0	0.5	0	0	0
0.7	0	0	0	0	0	0.5	0	0	0.5	0	0
0.8	0	0	0	0	0	0	0.5	0	0	0.5	0
0.9	0	0	0	0	0	0	0	0.5	0	0	0.5
1.0	0	0	0	0	0	0	0	0	0.5	0	0.5

TABLE 3 MARKOV MATRIX n=10 Monte vs. Pavlov

The equilibrium vector is computed as before, see Table 4 and Figure 3. Note that Pavlov is far more likely to defect against Monte than he is against Stalin. Also in Table 4 and Figure 4, we give the average payoff to each player for various values of n. The payoff to Pavlov is computed as

(7) $$\text{PyP} = \text{sum } r_i * \{ \text{pr(CC)} - \text{pr(DD)} + 2\text{pr(CD)} - 2\text{pr(DC)} \} =$$
$$\text{sum } r_i * \{ p/2 - (1-p)/2 + 2(1-p)/2 - 2p/2 \} =$$
$$\text{sum } r_i * (1/2-p)$$

while the payoff to Monte is

(8) $$\text{PyO} = \text{sum } r_i * \{\text{pr(CC)} - \text{pr(DD)} - 2\text{pr(CD)} + 2\text{pr(DC)} \} =$$
$$\text{sum } r_i * (-3+6p)/2)$$

Although Monte loses on the average, Pavlov will gain considerably, especially if he is a slow learner. Yet Pavlov will not do nearly as well as if he and his opponent perpetually cooperated.

		n, learning increment = 1/n										
.		6	7	8	9	10	11	12	13	14	15	16
p												
r	0/n	.39	.39	.39	.38	.38	.38	.38	.38	.38	.38	.38
o	1/n	.24	.24	.24	.24	.24	.24	.24	.24	.24	.24	.24
b	2/n	.15	.15	.15	.15	.15	.15	.15	.15	.15	.15	.15
_	3/n	.09	.09	.09	.09	.09	.09	.09	.09	.09	.09	.09
e	.	.06	.06	.06	.06	.06	.06	.06	.06	.06	.06	.06
n	.	.03	.04	.03	.03	.03	.03	.03	.03	.03	.03	.03
d	.	.03	.02	.02	.02	.02	.02	.02	.01	.02	.02	.02
_	.		.02	.01	.01	.01	.01	.01	.01	.01	.01	.01
i				.01	.01	.01	.01	.01	.01	.01	.01	.01
n					.01	.00	.01	.00	.01	.01	.01	.01
_						.00	.00	.00	.00	.00	.00	.00
s							.00	.00	.00	.00	.00	.00
t								.00	.00	.00	.00	.00
a									.00	.00	.00	.00
t										.00	.00	.00
e											.00	.00
_												.00
Average payoff per play to:												
Pavlov		0.268	0.291	0.311	0.328	0.343	0.356	0.367	0.377	0.385	0.393	0.399
Monte		-0.80	-0.87	-0.93	-0.98	-1.03	-1.07	-1.10	-1.13	-1.16	-1.18	-1.20

TABLE 4 EQUILIBRIUM VECTORS Monte vs. Pavlov

5. Gandhi vs Pavlov.

Gandhi is a player of infinite patience who hopes for the best in others. He will always cooperate regardless of what his opponent does. If Pavlov initially is quite likely to cooperate, then the probable CC outcome will reinforce his propensity to cooperate. Indeed if ever p increases to 1, so that Pavlov is trained, then since Gandhi will always cooperate, Pavlov will stay trained and an absorbing state of the Markov chain is reached; Pavlov and Gandhi will continue to cooperate and will each be rewarded one unit per play for this action. By his nature, Pavlov cannot plan ahead to see that he might benefit even more by defecting.

On the other hand if Pavlov is initially unlikely to cooperate, then he will be richly rewarded for his defection and will be even less likely to cooperate the next play. Even more quickly, the plays degenerate into the CD state in which Pavlov always defects and wins and Gandhi invariably cooperates and loses. Thus this Markov process has two absorbing states, p = 0 and p = 1. This is reflected in the entry of 1 in the (0,0) and the (n,n) position of the Markov matrix.

probability for next play

current prob

Prob	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.0	1.0	0	0	0	0	0	0	0	0	0	0
0.1	0.9	0	0.1	0	0	0	0	0	0	0	0
0.2	0.8	0	0	0.2	0	0	0	0	0	0	0
0.3	0	0.7	0	0	0.3	0	0	0	0	0	0
0.4	0	0	0.6	0	0	0.4	0	0	0	0	0
0.5	0	0	0	0.5	0	0	0.5	0	0	0	0
0.6	0	0	0	0	0.4	0	0	0.6	0	0	0
0.7	0	0	0	0	0	0.3	0	0	0.7	0	0
0.8	0	0	0	0	0	0	0.2	0	0	0.8	0
0.9	0	0	0	0	0	0	0	0.1	0	0	0.9
1.0	0	0	0	0	0	0	0	0	0	0	1.0

TABLE 5 MARKOV MATRIX n=10 Gandhi vs Pavlov

The theory behind the computation of the expected time until Pavlov arrives in an absorbing state and the payoff he and his opponent can expect during this time is as follows. See Chapter 3 [KS] for more details. Let Q be the matrix formed from the Markov matrix P by deleting the rows and the columns which contain an absorbing state (a 1 on the diagonal). Thus for Gandhi vs Pavlov, Q consists of the (n-1)x(n-1) matrix with entries the $p_{i,j}$ of P for $1\leq i,j\leq n-1$, i.e. we delete the p=0 and the p=1 states.

The matrix (I-Q) will be nonsingular and its inverse, N, is called the fundamental matrix. The entry, $n_{i,j}$, of N measures the expected number of times that Pavlov will be in state j/n given that he was initially in state i/n. Since Pavlov must eventually end up in one of the absorbing states, p=0 or p=1, this expected number will be finite. Thus the expected number of times before Pavlov arrives at an absorbing state given that he started in state i/n is N_i=sum n_{ij}.

The total payoff to Pavlov until he arrives in an absorbing state depends on his initial state p=i/n. It is computed as the expected payoff, given that he is in state j/n, times the expected number of times that he is in state j/n and then summed over j.

$$(9) \quad \text{TPyP}(i/n) = \text{sum } n_{ij} \{\text{pr}(CC,j/n) - \text{pr}(DD,j/n) + 2\text{pr}(CD,j/n) - 2\text{pr}(DC,j/n)\}$$

Compare this with equation (4). The average expected payoff per play is thus

$$(10) \quad \text{APyP}(i/n) = \text{TPyP}(i/n)/N_i.$$

Similarly we have the total and average payoff for Pavlov's opponent:

$$\text{(11)} \quad TPyO(i/n) = \text{sum } n_{ij} \{pr(CC,j/n) - pr(DD,j/n) - 2pr(CD,j/n) + 2pr(DC,j/n)\}$$

$$\text{(12)} \quad APyO(i/n) = TPyO(i/n)/N_i.$$

If his initial state p is small, i.e. if Pavlov is initially likely to defect then it is still possible that he will end up in the CC absorbing state. We list the values of these probabilities for x=1/10 (Table 6) and graph these values for n=10, 20 and 30 (Figure 5). Note that the dividing line for which absorbing state is reached is at about p = 2/3. In particular if the initial probability for cooperating is 1/2 then it is much more likely that they will end up in the CD state, especially if Pavlov is a slow learner. Even with x = 1/10, the chance that Pavlov will end up in the mutually cooperating state is less than one out of four (23.73%).

state	expected time to absorption	total payoff Pavlov	average payoff Pavlov	total payoff Gandhi	average payoff Gandhi	prob. to terminate in cooperation
0	0	0	0	0	0	0
0.1	1.1608	2.1839	1.8814	-1.9085	-1.6441	0.0006
0.2	1.6078	2.8386	1.7655	-2.0846	-1.2965	0.0060
0.3	3.0392	5.1932	1.7088	-3.4230	-1.1263	0.0300
0.4	4.0888	6.5484	1.6016	-3.2902	-0.8047	0.0985
0.5	5.3101	8.1130	1.5278	-3.0985	-0.5835	0.2373
0.6	5.5811	8.0328	1.4393	-1.7741	-0.3179	0.4445
0.7	4.9093	6.6891	1.3625	-0.4300	-0.0876	0.6752
0.8	3.3090	4.2217	1.2758	0.5708	0.1725	0.8629
0.9	1.4909	1.7689	1.1864	0.6570	0.4407	0.9675
1.0	0	0	0	0	0	1.0

TABLE 6 Time and Payoff to Train Gandhi vs Pavlov

6. Tit for Tat vs Pavlov.

TT plays what his opponent did on the previous play. This strategy appears to be best in many situations [Ax, Ho, Ra]. However, we will see that TT's strategy is quite poor when used against our arational but trainable player, Pavlov.

We model this game with a 2n+2 Markov process with n+1 states, Cp, and n+1 state, Dp. State Cp means that Pavlov will cooperate with probability p **and** that he cooperated last play so that TT will cooperate next play. State Dp is interpreted similarly. This game has a single absorbing state of C1.

To compute the transition probabilities, consider first the state Cp. As Pavlov cooperated to arrive at that state, TT will cooperate on the next round. Thus with probability p the outcome will be CC and so the new state will be C(p+x), i.e., $Tr^{+}(Cp)=p$. Similarly with probability $Tr^{+}(Cp)=(1-p)$, the outcome will be DC and so the new probability state will be D(p-2x). The same reasoning shows that Dp will transform to state D(p+x) with probability $Tr^{+}(Dp)=1-p$ and to state C(p-2x) with probability $Tr^{-}(Dp)=p$. The Markov matrix is in Table 7. In the table, Di [Ci] denotes the state Di/10 [Ci/10].

probability for next play

current_prob	D0	C0	D1	C1	D2	C2	D3	C3	D4	C4	D5	C5	D6	C6	D7	C7	D8	C8	D9	C9	D10	C10
D0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	.1	0	0	.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C1	.9	0	0	0	0	.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	.2	0	0	0	0	.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2	.8	0	0	0	0	0	0	.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D3	0	0	0	.3	0	0	0	0	.7	0	0	0	0	0	0	0	0	0	0	0	0	0
C3	0	0	.7	0	0	0	0	0	0	.3	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	.4	0	0	0	0	.6	0	0	0	0	0	0	0	0	0	0	0
C4	0	0	0	0	.6	0	0	0	0	0	0	.4	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	.5	0	0	0	0	.5	0	0	0	0	0	0	0	0	0
C5	0	0	0	0	0	0	.5	0	0	0	0	0	0	.5	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	.6	0	0	0	0	.4	0	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	.4	0	0	0	0	0	0	.6	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	.7	0	0	0	0	.3	0	0	0	0	0
C7	0	0	0	0	0	0	0	0	0	0	.3	0	0	0	0	0	0	.7	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	.8	0	0	0	0	.2	0	0	0
C8	0	0	0	0	0	0	0	0	0	0	0	0	.2	0	0	0	0	0	0	.8	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.9	0	0	0	0	.1	0
C9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.1	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
C10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

TABLE 7 Markov Matrix n=10 Tit for Tat vs Pavlov

The expected time until Pavlov is fully trained, i.e, in the C1 state, is given in Table 8 for n=10. The payoffs to Pavlov and TT incurred in the training process are in corresponding columns. The formulas for these quantities are as in equations (4) and (5). See also the MATLAB programs in the appendix.

State	expected time to train	total payoff Pavlov	average payoff Pavlov	total payoff Tit for Tat	average payoff TT
D 0	216.8659	-108.9504	-0.5024	-104.9572	-0.4840
C 0	217.8659	-106.9504	-0.4909	-106.9572	-0.4909
D 0.1	215.8659	-107.9504	-0.5001	-103.9572	-0.4816
C 0.1	217.8786	-106.7458	-0.4899	-106.7527	-0.4900
D 0.2	214.5325	-106.8393	-0.4980	-102.8461	-0.4794
C 0.2	216.9929	-105.9050	-0.4881	-105.9119	-0.4881
D 0.3	212.4492	-105.3115	-0.4957	-101.3183	-0.4769
C 0.3	212.5012	-102.7237	-0.4834	-102.7304	-0.4834
D 0.4	208.6938	-102.8396	-0.4928	-98.8465	-0.4736
C 0.4	201.3170	-96.1948	-0.4778	-96.2012	-0.4779
D 0.5	201.4943	-98.4627	-0.4887	-94.4695	-0.4688
C 0.5	178.9936	-84.2281	-0.4706	-84.2340	-0.4706
D 0.6	188.4874	-91.2016	-0.4839	-87.2086	-0.4627
C 0.6	143.5380	-66.1448	-0.4608	-66.1496	-0.4609
D 0.7	**166.7431**	-79.7119	-0.4781	-75.7197	-0.4541
C 0.7	**98.4341**	-44.0149	-0.4472	-44.0184	-0.4472
D 0.8	134.8253	-63.5073	-0.4710	-59.5198	-0.4415
C 0.8	52.8369	-22.5373	-0.4265	-22.5393	-0.4266
D 0.9	94.9744	-43.9571	-0.4628	-40.0005	-0.4212
C 0.9	17.6743	-6.8712	-0.3888	-6.8720	-0.3888
D 1.0	53.8369	-24.4373	-0.4539	-20.8393	-0.3871
C 1.0	0	0	0	0	0

TABLE 8 Time and Payoff to Train Tit for Tat vs Pavlov

The expected time to train Pavlov is shorter if TT starts with the C strategy. This benefit is significant if Pavlov's initial probability state p is greater that .5. For example, if p = 0.7, then the expected time until Pavlov is trained is 167 plays if TT initially defects but 98 plays if he initially cooperates. It is interesting to observe that the average payoff to Pavlov and to TT are nearly the same. Each player suffers the same amount during the training process.

7. Perfect Trainer vs Pavlov

Our perfect trainer, PT, knows, through comprehensive psychological observation and sagacious mathematical analysis, exactly the mental state of Pavlov before each play. This means **only** that PT knows with what probability Pavlov will cooperate, **not** whether he will cooperate. He is no mind reader. We first describe PT's strategy to train Pavlov most efficiently. In other words, what PT should play to maximize the propensity for p to increase.

If PT were to cooperate, then p would increase to p+x only if Pavlov also cooperates, which happens with probability p. If PT were to defect, then p would also increase to p+x if Pavlov defects (the negative reinforcement). As pr(DD,p)=(1-p), PT should cooperate only if p>(1-p), i.e. if p>1/2. In other words, PT should do just what Pavlov is more likely to do. In this way he will reward anticipated good behavior and punish anticipated bad behavior. Note that if p=1 then PT and Pavlov will each assuredly cooperate if PT used this strategy, i.e. a trained Pavlov is an absorbing state.

Note that if p is close to 1/2, then Pavlov will frequently do the unexpected, resulting in a CD or DC outcome and thus a significant regression in the training process. Define an index depending on p, the expected reinforcement function, as follows:

$$(13)\quad ER(p)=pr(CC,p) + pr(DD,p) - 2pr(CD,p) - 2pr(DC,p).$$

$$= \begin{cases} 1 - 3p & \text{if } p\leq 1/2 \\ -2 + 3p & \text{if } p>1/2 \end{cases}$$

This index is different from the payoff functions (4) and (5) in that it measures negative as well as positive reinforcement toward the goal of being trained. If this index is positive, then on the average, Pavlov will become more cooperative, while if it is negative, then Pavlov will likely regress. For the perfect trainer, this index is given by the V shaped curve graphed in Figure 6.

For example if $p=1/2$, then half the time p will increase by x while the other half it will decrease by $2x$. Thus we may expect an average change of $xER(1/2) = (x-2x)/2=-x/2$ in the p state. If $p=2/3$, then PT will play C and thus the probability of a CC outcome is 2/3 while that of a CD outcome is 1/3. Thus

$$(14) \quad ER(2/3) = 1*2/3 - 2*1/3 = 0,$$

meaning that the expected increase in p balances with the expected decrease. However if p is slightly less than 2/3, then $ER(p)<0$ and so p will likely decrease away from 2/3, away from cooperation, while if $p>2/3$, $ER(p)>0$ and so p will also likely increase away from 2/3 but toward cooperation. This likelihood is even greater as p gets closer to 1 so that p should soon reach 1; Pavlov will then be trained.

If $p=1/3$ then PT will play D. Thus a DC outcome will result with probability 1/3 and a DD with probability 2/3. The expected reinforcement is thus

$$(15) \quad ER(1/3) = 2/3 - 2*1/3 = 0.$$

Slightly below 1/3, the index will be positive, pushing p up toward 1/3 while slightly above 1/3 it will be negative, pushing p back toward 1/3. Thus the state p=1/3, may be called an attracting state in the sense that nearby states tend toward it.

It may be helpful to look at this training process as a swimmer in a stream with varying current. The swimmer randomly strokes either toward the D shore or toward the C shore with strength of his stroke proportional to the learning increment x. The current will tend to push the swimmer away from the D shore with a force decreasing with distance. At the 1/3 mark, the current will shift, tending to push the swimmer back to the attracting 1/3 mark. This counter current is greatest in the middle, and weakens near the 2/3 mark. Beyond the 2/3 mark, the current will aid the swimmer to reach the C shore at which place he remains. Thus we can expect to see Pavlov swim randomly about the 1/3 mark until a rare succession of chance strokes causes him to bucks the current, gets him past the 2/3 mark and then, with the aid of the current, to the C shore.

We illustrate this phenomena quantitatively for the learning increment, x =1/10. First note that the Markov matrix (Table 9) looks like that with Stalin when p is small and like that with Gandhi when p is large.

	probability for next play										
current_prob: Prob	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	0	1.0	0	0	0	0	0	0	0	0	0
0.1	0.1	0	0.9	0	0	0	0	0	0	0	0
0.2	0.2	0	0	0.8	0	0	0	0	0	0	0
0.3	0	0.3	0	0	0.7	0	0	0	0	0	0
0.4	0	0	0.4	0	0	0.6	0	0	0	0	0
0.5	0	0	0	0.5	0	0	0.5	0	0	0	0
0.6	0	0	0	0	0.4	0	0	0.6	0	0	0
0.7	0	0	0	0	0	0.3	0	0	0.7	0	0
0.8	0	0	0	0	0	0	0.2	0	0	0.8	0
0.9	0	0	0	0	0	0	0	0.1	0	0	0.9
1.0	0	0	0	0	0	0	0	0	0	0	1.0

TABLE 9 MARKOV MATRIX n=10 Perfect Trainer vs Pavlov

The expected number of plays and the total and average payoffs to Pavlov and to PT are given in Table 10 for n=10.

prob.	Total time to train	Total pay to Pavlov	av. pay per play Pavlov	Total pay to PT	av.pay per play for PT
0	51.6739	-29.0458	-0.5621	-13.2089	-0.2556
0.1	50.6739	-28.0458	-0.5535	-12.2089	-0.2409
0.2	49.4517	-26.7125	-0.5402	-11.3200	-0.2289
0.3	47.6461	-24.6291	-0.5169	-10.3478	-0.2172
0.4	44.9199	-21.3077	-0.4743	-9.4073	-0.2094
0.5	**40.2321**	-15.3712	-0.3821	**-8.4655**	-0.2104
0.6	30.8181	-9.1133	-0.2957	-5.5833	-0.1812
0.7	19.7502	-3.3170	-0.1679	-2.7006	-0.1367
0.8	**9.5436**	-0.0080	-0.0008	**-0.3727**	-0.0391
0.9	2.9750	0.7683	0.2583	0.4299	0.1445
1.0	0	0	0	0	0

TABLE 10 Time and Payoff to Train PT vs Pavlov

Note that the average payoff to PT during the training process for x=1/10 is slightly negative. Clearly, in the short run, he would do better always defecting as did Stalin. On the other hand, once Pavlov is trained, then PT collects +1 on each successive play and so rapidly overtakes Stalin. Indeed, if p=1/2 initially, it will take on the average 40 plays to train Pavlov. The total payoff to PT during that training period is of the order of -8.5 as compared with -0.8 for Stalin. Thus 8 steps after Pavlov is trained, or 48 steps on the average from the start, PT will have a higher payoff than Stalin. From then on he will do far better.

The expected training time and training cost or payoff depends, of course, on how quickly Pavlov learns. See Table 11 and Figure 8 for times and payoffs for n from 4 to 30..

n	expected time to train	total payoff Pavlov	average payoff Pavlov	total payoff PT	average payoff PT
4.0	6.4222	-0.2000	-0.0311	-2.6889	-0.4187
6.0	12.4857	-2.7571	-0.2208	-3.8429	-0.3078
8.0	20.7631	-7.2840	-0.3508	-5.1750	-0.2492
10.0	31.6077	-14.2529	-0.4509	-6.6532	-0.2105
12.0	45.5331	-24.3577	-0.5349	-8.3029	-0.1823
14.0	63.1870	-38.4339	-0.6083	-10.1372	-0.1604
16.0	85.3872	-57.5299	-0.6738	-12.1818	-0.1427
18.0	113.1425	-82.9328	-0.7330	-14.4619	-0.1278
20.0	147.6925	-116.2256	-0.7869	-17.0066	-0.1151
22.0	190.5518	-159.3464	-0.8362	-19.8463	-0.1042
24.0	243.5666	-214.6645	-0.8813	-23.0139	-0.0945
26.0	308.9832	-285.0712	-0.9226	-26.5447	-0.0859
28.0	389.5299	-374.0892	-0.9604	-30.4766	-0.0782
30.0	488.5169	-486.0047	-0.9949	-34.8507	-0.0713

TABLE 11 Training Payoff for n = 4 to 30

8. To swindle or not to swindle:

Once Pavlov is trained, PT might be tempted to defect, collect an extra unit, and then retrain Pavlov by returning to the strategy of cooperation. For example with x=1/10, PT knows that even after a double-cross, Pavlov will still cooperate 80% of the time on the next play, and, given that he does cooperate, he will be trained 90% of the time after the second play. Thus with probability 0.72 = 0.8 * 0.9, Pavlov will be retrained in two steps and PT will be one unit richer for his action.

However a reading of Table 10 shows that this defection is quite ill advised for PT. Although it is indeed unlikely that Pavlov will defect in response to PT's swindle, if Pavlov does defect (he will do so about 28% of the time), then it will be very costly to PT. The expected number of plays to train Pavlov from this p=0.8 state is 9.5 and the expected total payoff to PT is

-0.37. Thus in these 10 or 11 plays from the initial defection until Pavlov is retrained, PT has gained two units from the DC outcome but can expect to lose 0.37. If he had not defected, then PT would have collected one unit for each of these 10 or 11 plays. So by cheating, PT will on the average be about 9 units poorer than he would have been had he always cooperated.

If Pavlov is a slow learner, i.e., n is large, then PT has a better chance for getting away with a bit of cheating. However our computations show that the break even point does not occur until n is approximately 30. At this learning increment, the average time to train Pavlov is of the order of 500 plays, a very slow learner indeed (See Table 11 and Figure 8). In summary, against a Pavlov with an even marginally realistic learning increment, it does not pay to cheat.

9. Normal Trainer vs Pavlov

One may argue that it is unrealistic for anyone to know the state of mind of Pavlov. Our normal trainer, NT, has no extraordinary observational skills that enable him to know the exact value of Pavlov's probability state. Yet he can guess whether $p>1/2$ by looking at Pavlov's level of cooperation on the last few plays. If p is actually 0, then NT should be able to predict with certainty from the last few plays that p is at least less than 1/2. Similarly if $p=1$, NT could be expected to know that $p>1/2$. However if p is near 1/2 then NT would have only a 50-50 chance of guessing right.

Let $f(p)$ be the probability that NT guesses that $p>1/2$. We have argued above that $f(0)=0$, $f(1/2)=1/2$, and $f(1)=1$. For simplicity we assume that $f(p)=p$ for all p. Thus if $p=2/3$, we assume that NT will correctly guess that $p>1/2$ just two out of every three times. This is a quite weak constraint on NT. A mathematically or psychologically inclined trainer should be able to do far better. Note that the similar probability function for PT is given by $f(p)=0$ if $p<1/2$ and $f(p)=1$ if $p\geq 1/2$, i.e., PT "guesses" with absolute certainty.

The best strategy for NT is the closest that he can come to the strategy of PT. He should cooperate if he guesses that Pavlov is more likely to cooperate than defect and he should defect otherwise. Assume that Pavlov is in state p. Then pr(CC,p), the probability that CC will be the outcome, is the probability that Pavlov actually does cooperate, p, times the probability that NT thinks that Pavlov is likely to cooperate, $f(p)=p$. Thus pr(CC,p) $= p^2$. Similarly pr(DD,p) $= (1-p)^2$. Since p increases after either a CC or a DD outcome, the probability that p will increase to $p+x$ is $p^2 + (1-p)^2 = 1 - 2p + 2p^2$. The complementary probability, $2p - 2p^2$, is thus the probability that p will decrease to $p-2x$. The expected reinforcement index (13) is

$$(16)\quad ER(p) = 1 - 2p + 2p^2 - 2(2p - 2p^2) = 1 - 6p + 6p^2.$$

This is the equation of a parabola with minimum at (1/2, -1/2) and which is 0 for $p = (3 \pm (3))/6 \simeq 0.21$ or 0.79 (see Figure 7). In our stream analogy, the current turns against the swimmer for a considerably larger interval, 0.21 to 0.79 instead of 0.33 to 0.67, although the strength of the current is the same at the shores and the center of the stream. The Markov matrix is given in Table 12 for n = 10.

		Probability for next play									
current prob	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	0	1.0	0	0	0	0	0	0	0	0	0
0.1	.18	0	.82	0	0	0	0	0	0	0	0
0.2	.32	0	0	.68	0	0	0	0	0	0	0
0.3	0	.42	0	0	.58	0	0	0	0	0	0
0.4	0	0	.48	0	0	.52	0	0	0	0	0
0.5	0	0	0	.50	0	0	.50	0	0	0	0
0.6	0	0	0	0	.48	0	0	.52	0	0	0
0.7	0	0	0	0	0	.42	0	0	.58	0	0
0.8	0	0	0	0	0	0	.32	0	0	.68	0
0.9	0	0	0	0	0	0	0	.18	0	0	.82
1.0	0	0	0	0	0	0	0	0	0	0	1.0

TABLE 12 Markov Matrix Normal Trainer vs Pavlov

The expected time and payoff is displayed in Table 13. The payoff for Pavlov will be exactly the same as to NT since

pr(CD,p) = pr(DC,p) = p(1-p) in equations (4) and (5) and so

PyP(y) = PyO(p) = pr(CC,p) - pr(DD,p) = 2p - 1.

prob	Total time to train	Total payoff	average payoff
0	**128.6799**	-59.1004	-0.4593
0.1	127.6799	-58.1004	-0.4550
0.2	126.2409	-56.9052	-0.4508
0.3	123.6226	-54.9899	-0.4448
0.4	118.9603	-52.0478	-0.4375
0.5	110.3167	-47.1794	-0.4277
0.6	95.0108	-39.3690	-0.4144
0.7	70.9804	-28.0501	-0.3952
0.8	40.7715	-14.8874	-0.3651
0.9	13.7765	-4.2490	-0.3084
1.0	0	0	0

TABLE 13 Time and Payoff to Train NT vs Pavlov

10. Pavlov vs Pavlov:

Pavlov's final opponent is his clone. We call these two players P1 and P2 and assume that they have the same learning increment $x=1/n$. Initially we assume that P1 will cooperate with probability p1 and P2 with probability p2. Without loss of generality we may assume that $p1 \geq p2$. This may be modelled by a Markov process with $(n+1)^2$ states (p1,p2). For example if n=10 and the joint probability state is (0.7, 0.4), then pr(CC,p) = 0.7*0.4 while pr(DD,p) = 0.3*0.6. Either of these outcomes will increase the likelihood of cooperation by both P1 and P2. That is the new state will be (0.8,0.5) with probability 0.28 + 0.18 = 0.46. The DC and CD outcomes will lower each probability by two increments. That is the new state will be (0.5,0.2) with probability 1-0.46 = 0.54. Note that p1-p2=0.3 in each of these new states as well as in the original state.

Now assume that the initial state was (1,0), i.e., P1 cooperated and P2 defected with certainty. P1 will be punished for cooperating and so will be less likely to do so next time. P2 will be rewarded for defecting and thus will again defect with certainty on the next play. Thus the next state will be (0.8, 0). A similar argument shows that the following state will be either (0.6, 0) or (0.9, 0.1). From state (0.9,0.1) the next state would be either (1.0,0.2) if CC or DD is the outcome or (0.7,0) if CD or DC is the outcome. In the latter case, note that 0.1 could not be reduced by two increments.

Graphically what happens is that the states either move up two or down one learning increments along the diagonals p1-p2 = const, or they move to the right one along the lower side, p1=1, or they move up 2 increments along the right hand side, p2 =0. A movement along the side must transform the state (p1,p2) to a diagonal x-y = const closer to the main diagonal, so that the new values of p1 and p2 are closer together. Once we leave a diagonal p1-p2 = const, we can never return to it. See Figure 10.

It is quite likely that the state (1,0) or the state (0.7,0.4) will transform to the (0,0) state of mutual assured defection in a relatively few plays. Once in that state, we will be guaranteed that p1 = p2 for all successive states, i.e. the clones will decide with the same probability whether or not to cooperate. Their actual decisions will often be different.

The transition probabilities along this main diagonal are precisely the same as for the NT vs. Pavlov game, i.e., p=p1=p2 will increase with probability $1-p1-p2+2p1*p2 = 1-2p+2p^2$. Thus for n=10, starting at (0,0) we may expect mutual training in 129 moves, the expected training time for NT vs Pavlov from the p=0 state (see Table 13).

In our Markov model for P1 vs P2, we need only consider the states with p1≥p2 since it is impossible to cross the main diagonal. Thus there are (n+1)(n+2)/2 states, that is 66 states for n=10. Due to the large size, we will not present the Markov matrix here as we did for the TT, PT and NT games.

The matrix in Table 14 represents the expected time to train P1 and P2. For example if the initial state is (1,0) then 137 plays are expected until mutual training. If the initial state is (0.7,0.4) then 119 plays should be expected. As mentioned above, 129 plays are expected if one starts in the (0,0) state. If p1 and p2 are each initially quite large, there will be a reasonable chance to achieve mutual cooperation, (1,1), without first descending to the (0,0) state of mutual defection, thus considerably shortening the expected time.

Probability State for P2

State of P1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	**129**	129	130	131	133	134	135	135	136	137	137
0.1	129	128	128	129	131	133	134	135	136	137	137
0.2	130	128	126	126	128	130	132	134	135	136	137
0.3	131	129	126	124	123	124	126	129	131	132	133
0.4	133	131	128	123	119	117	117	119	121	123	124
0.5	134	133	130	124	117	110	106	104	104	104	105
0.6	135	134	132	126	117	106	95	87	81	79	78
0.7	135	135	134	129	**119**	104	87	71	59	51	48
0.8	136	136	135	131	121	104	81	59	41	29	23
0.9	137	137	136	132	123	104	79	51	29	14	7
1.0	**137**	137	137	133	124	105	78	48	23	7	0

TABLE 14 Expected time to train

The matrix in Table 15 gives the payoff to P1 for the mutual training process. The payoff to P2 is the symmetric entry. For example if the initial state was (1,0), then the payoff to the cooperating P1 is -75 while the payoff to the defecting P2 is the other corner entry, -46. For initial state (0.7,0.4), the payoff to P1 is -59 and that to P2 is -42.

Probability State for P2

State of P1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	-59	-57	-55	-53	-52	-51	-50	-49	-48	-47	**-46**
0.1	-61	-58	-55	-53	-52	-51	-49	-48	-47	-46	-46
0.2	-64	-60	-57	-53	-51	-50	-49	-48	-47	-46	-45
0.3	-66	-63	-60	-55	-51	-48	-47	-46	-45	-44	-43
0.4	-68	-66	-63	-57	-52	-47	-44	**-42**	-41	-40	-39
0.5	-69	-68	-66	-60	-54	-47	-41	-37	-34	-33	-32
0.6	-71	-70	-68	-63	-57	-48	-39	-32	-27	-24	-22
0.7	-72	-72	-70	-66	**-59**	-49	-38	-28	-20	-15	-12
0.8	-73	-73	-72	-69	-62	-51	-37	-24	-15	-8	-5
0.9	-74	-74	-73	-71	-64	-52	-37	-22	-11	-4	-1
1.0	**-75**	-75	-74	-72	-65	-54	-37	-21	-8	-2	0

TABLE 15 Expected payoff for P1

Certain joint probability states are more likely to be assumed than others. If the intial state is (1,0), then the next state is (0.8,0) and states with p1-p2 = 0.9 are no longer possible. From here the next state will be (0.6,0) with 80% probability. This implies that the joint states tend to move toward the (0,0) mutual defection state. Table 16 displays the expected number of times that the joint probability will be (p1,p2), given that the initial state was (1,0). Note that (0.7,0.4) will be a joint state in only 16% of all such extended Prisoner's Dilemma games. On the other hand, we can expect that P1 and P2 will each defect with certainty over 12 times in each extended game before the mutual training will be achieved. For the details on the computation of this number, see Section 5, the MATLAB program in the Appendix B, and Theorem 3.5.4 [KS].

Probability State for Pavlov 2

	.	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
	0	**12.56**	0	0	0	0	0	0	0	0	0	0
	0.1	1.41	21.38	0	0	0	0	0	0	0	0	0
	0.2	1.19	1.92	24.49	0	0	0	0	0	0	0	0
p	0.3	.86	1.31	1.86	20.99	0	0	0	0	0	0	0
a	0.4	.97	.77	1.08	1.41	14.50	0	0	0	0	0	0
v	0.5	.49	.70	.53	.71	.89	8.68	0	0	0	0	0
l	0.6	.98	.28	.40	.31	.41	.51	4.84	0	0	0	0
	0.7	.18	.42	.13	.20	**.16**	.22	.28	2.71	0	0	0
	0.8	1.03	.06	.16	.06	.10	.09	.13	.17	1.57	0	0
	0.9	0	.21	.02	.05	.02	.04	.04	.07	.10	1.07	0
	1.0	1.00	0	0	.01	.02	.02	.03	.04	.08	.14	0

TABLE 16 **Time in each state, start in (1,0)**

The same pattern is presented for the initial state of (0.7,0.4) in Table 17. Note that no states on diagonals lower than p1-p2=0.3 can ever be realized; P1 and P2 will never diverge in their probability states. Although the more likely route to mutual cooperation is still through mutual defection, (they can expect to be in the (0,0) state nearly 11 times on the average), a sequence of six increasingly likely CC outcomes would lead to the (1,1) state. As the expected time to train P1 and P2 from the initial (0.7,0.4) state is 119 while it is 129 from the (0,0) state, it will indeed occasionally happen that this route of mutual cooperation will be followed.

		Probability State for P2										
	.	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
	0	**10.87**	0	0	0	0	0	0	0	0	0	0
	0.1	1.25	18.51	0	0	0	0	0	0	0	0	0
	0.2	.95	1.71	21.20	0	0	0	0	0	0	0	0
P	0.3	.87	1.06	1.67	18.17	0	0	0	0	0	0	0
a	0.4	0	1.32	.89	1.27	12.55	0	0	0	0	0	0
v	0.5	0	0	1.74	.60	.80	7.51	0	0	0	0	0
l	0.6	0	0	0	1.32	.36	.46	4.19	0	0	0	0
	0.7	0	0	0	0	**1.80**	.20	.26	2.35	0	0	0
	0.8	0	0	0	0	0	.91	.15	.17	1.36	0	0
	0.9	0	0	0	0	0	0	.45	.08	.10	.93	0
	1.0	0	0	0	0	0	0	0	.26	.24	.27	0

TABLE 17 Time in each state, start in (0.7,0.4)

11. Comparison of Training methods and generalizations:

Each of the opponents, Tit for Tat, Perfect Trainer, Normal Trainer and Pavlov's clone, will eventually train Pavlov to cooperate. This is an immediate consequence of the fact that the Markov chain in our model has exactly one absorbing state (see Chapter 3 [KS]). Figure 9 plots the expected training times for these different trainers for values of n from 4 to 11. The training time appears to be exponential in n, at least for NT, TT and PP. Note that if the original payoff matrix (1) is replaced by one in which all rewards and punishments are doubled, then this has the same affect as doubling the learning increment or halving n.

The effect of changing the relative size of the rewards and punishments has a more significant qualitative effect on this model. For example if the payoff matrix is changed to

$$\begin{vmatrix} 9,9 & -10,10 \\ 10,-10 & -1,-1 \end{vmatrix}$$

(see p37 [Ra]), then the expected reinforcement function (9), which measures the tendency to move towards cooperation, should be modified to

(17) $ER(p) = 9pr(CC) + pr(DD) - 10pr(DC) - 10pr(CD)$.

If the opponent were to always cooperate, then

$ER(p) = 9p - 10(1-p)$,

while if he were to consistently defect, then

$ER(p) = (1-p) - 10p$.

Thus to increase the propensity to cooperate, our Perfect Trainer should cooperate only when doing so is more likely to reinforce cooperation than his defection would be, i.e., when

$9p - 10(1-p) > (1-p) - 10p$ or $p>11/30$.

With this strategy ER(p) is negative only for p between 1/11 and 10/19, a much shorter interval than that in section 6. That mutual cooperation will evolve quite rapidly in this game has been verified by simulations [Ra].

A similar analysis can be applied to other variations of the Prisoner's Dilemma and even to other non zero sum 2 person games. The Markov chains can be constructed precisely as in this paper to get quantitative results.

12. Conclusions:

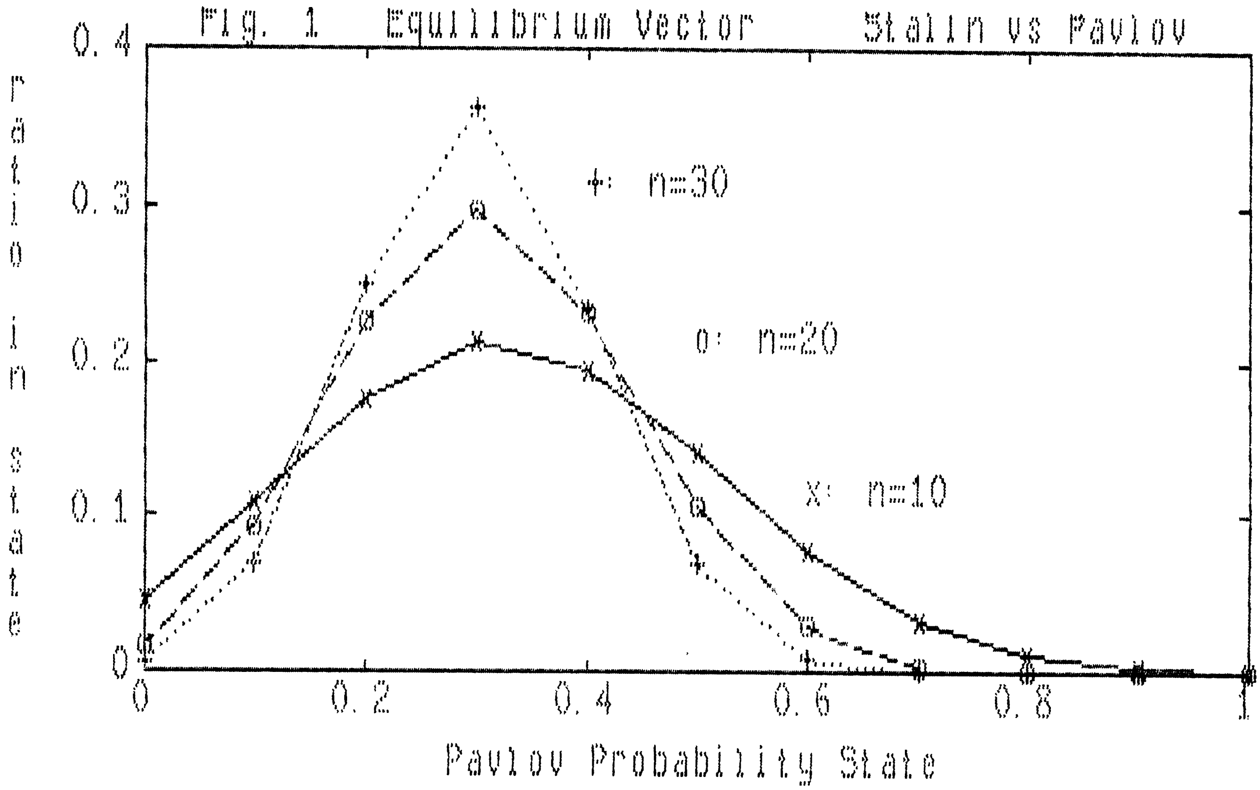
Drawing conclusions about the real world from these simple models is a precarious undertaking. Life is not a succession of identical Prisoner's Dilemmas with only two players. However our Markov approach does suggest some qualitative strategies. Totally inflexible or totally random strategies tend to do poorly, as we saw with Stalin, Monte and Gandhi. Indeed Gandhi's strategy of always cooperating is seen to work well only if his opponent is already inclined to cooperate. More often it results in a disaster for him, i.e., an absorbing CD state.

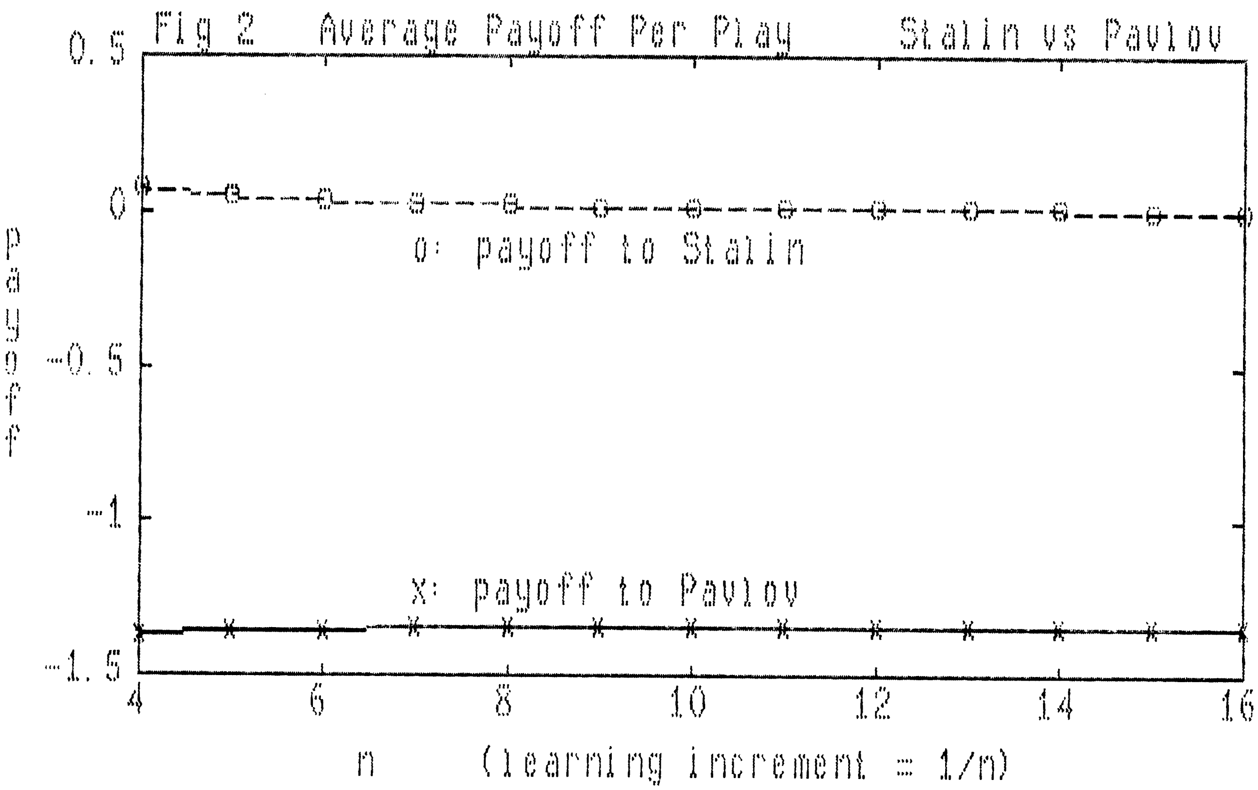
The Tit for Tat strategy, so successful against rational opponents, is inefficient against Pavlov. One should be more forgiving of the occasional slip by the basically cooperative opponent and more strict toward the habitual defector. An understanding of the degree of rationality of your opponent is essential in designing an effective strategy.

The training time depends quite heavily (exponentially) on the learning increment. Although it may not be possible to make your opponent smarter, it might be possible to scale up the rewards and punishments, which would have the same effect. Other modifications in the payoff matrix will have even more dramatic effect on the training time.

Perhaps our most surprising result is how quickly the Pavlov vs Pavlov game converges to identical probability states, p1=p2. Even though the players frequently degenerate into the (0,0) state of mutual defection, from that joint state, they will tend toward mutual cooperation as rapidly as if a normal trainer were the opponent. Do species in the wild tend to behave with equal aggressiveness in competitive situation because of this conditioned response or by some sort of evolutionary instinct? Is it not the case that in human interaction, each party tends to exhibit the same degree of belligerence and friendliness? Perhaps one should have more faith in the efficiency of natural forces.

APPENDIX A FIGURES





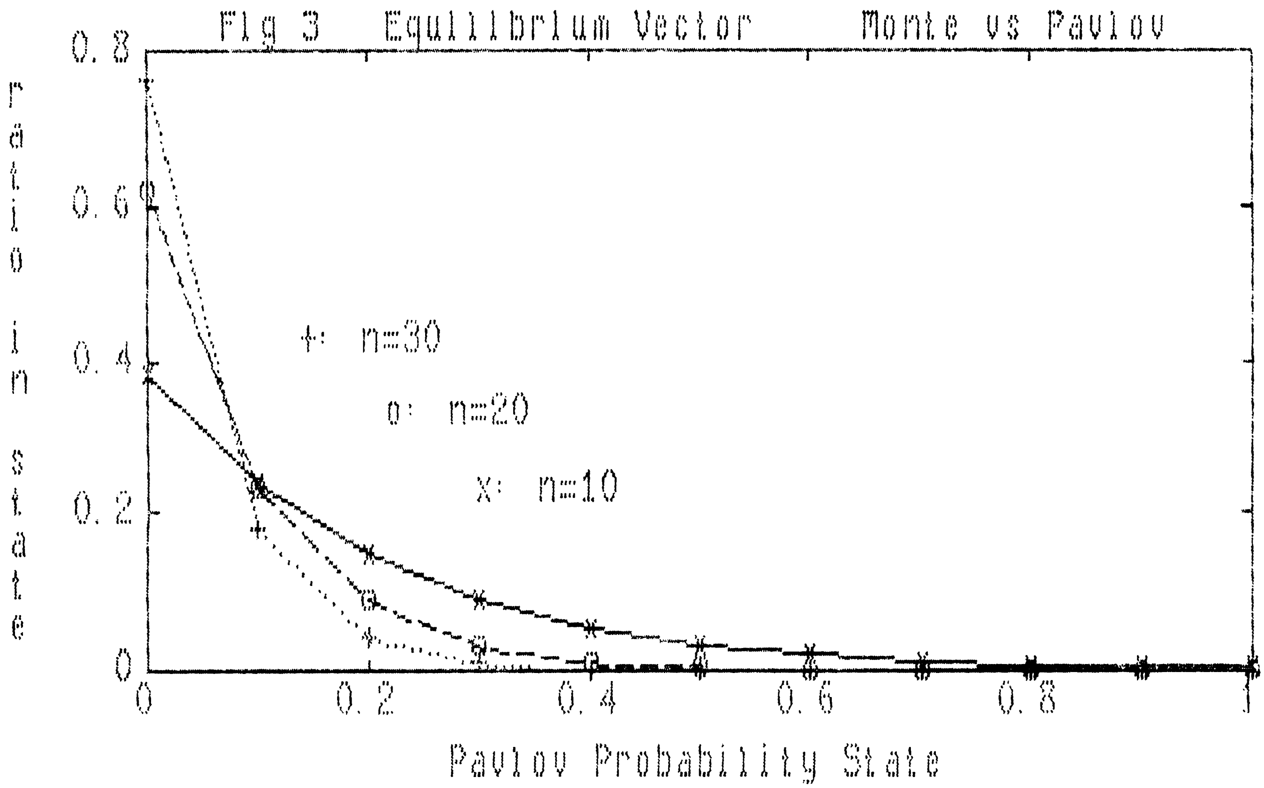


Fig 3 Equilibrium Vector Monte vs Pavlov
ratio in state
+: n=30
o: n=20
x: n=10
Pavlov Probability State
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1

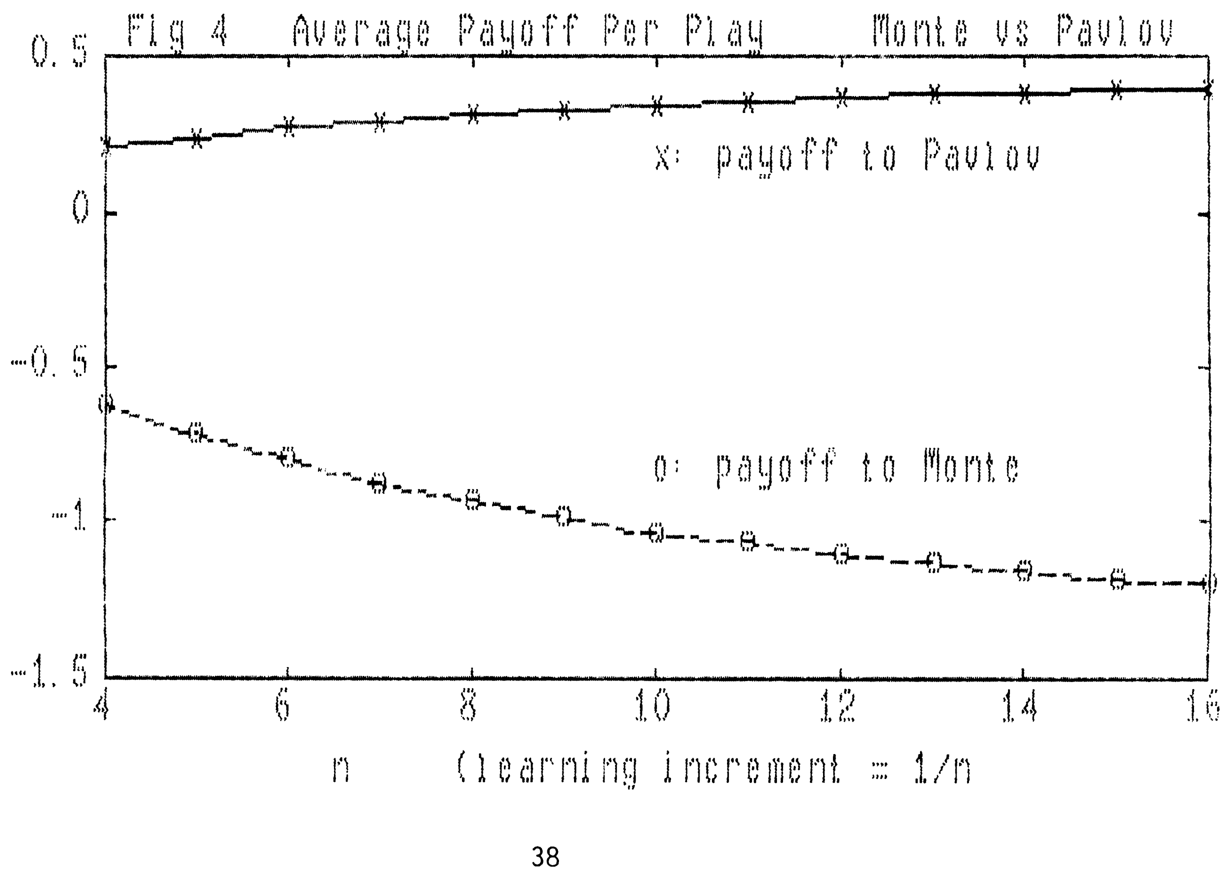


Fig 4 Average Payoff Per Play Monte vs Pavlov
payoff
x: payoff to Pavlov
o: payoff to Monte
n (learning increment = 1/n)
0.5
0
-0.5
-1
-1.5
4
6
8
10
12
14
16

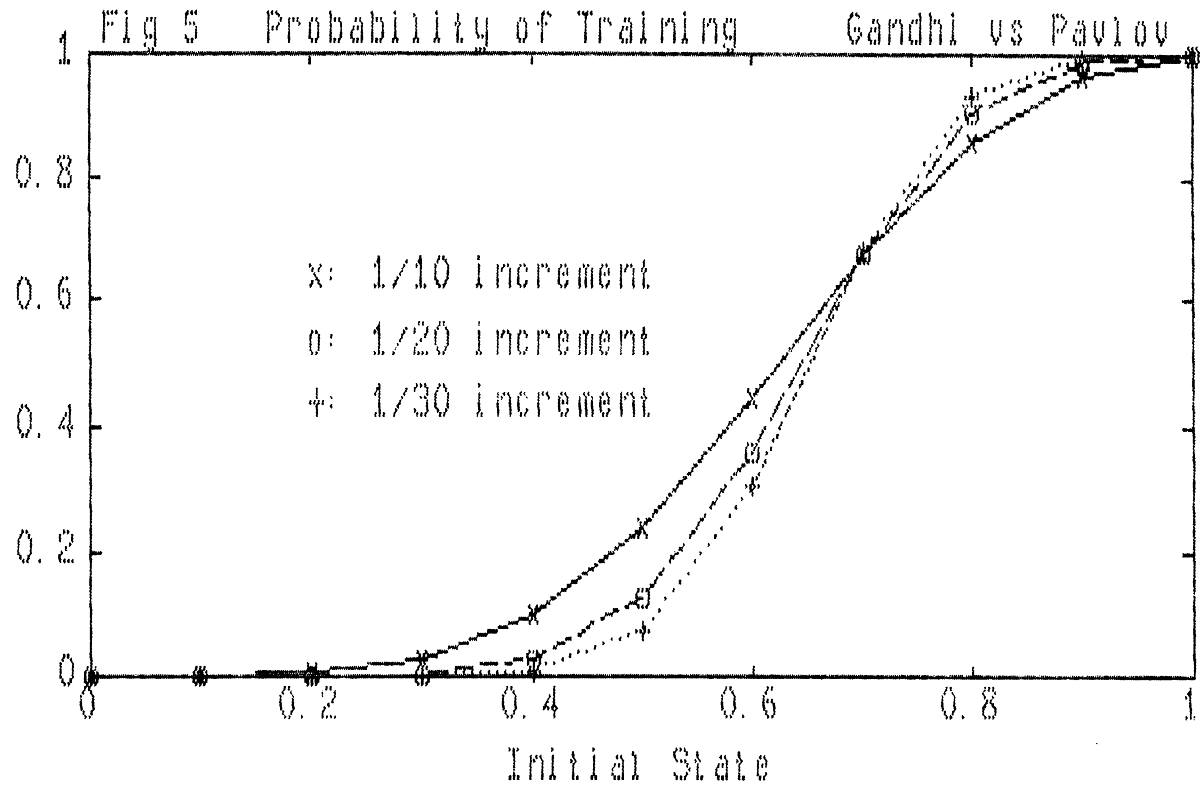


Fig 5 Probability of Training Gandhi vs Pavlov
x: 1/10 increment
o: 1/20 increment
+: 1/30 increment
Initial State
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

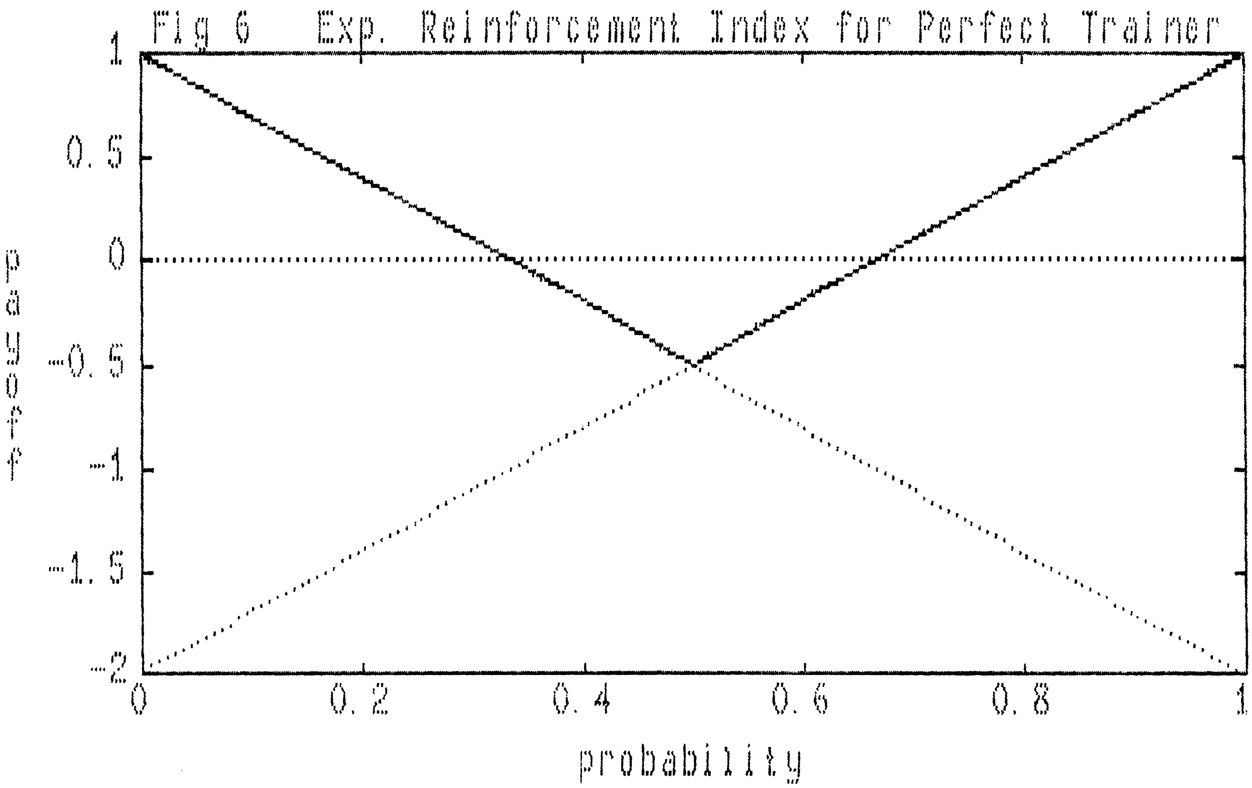


Fig 6 Exp. Reinforcement Index for Perfect Trainer
payoff
probability
1
0.5
0
-0.5
-1
-1.5
-2
0
0.2
0.4
0.6
0.8
1

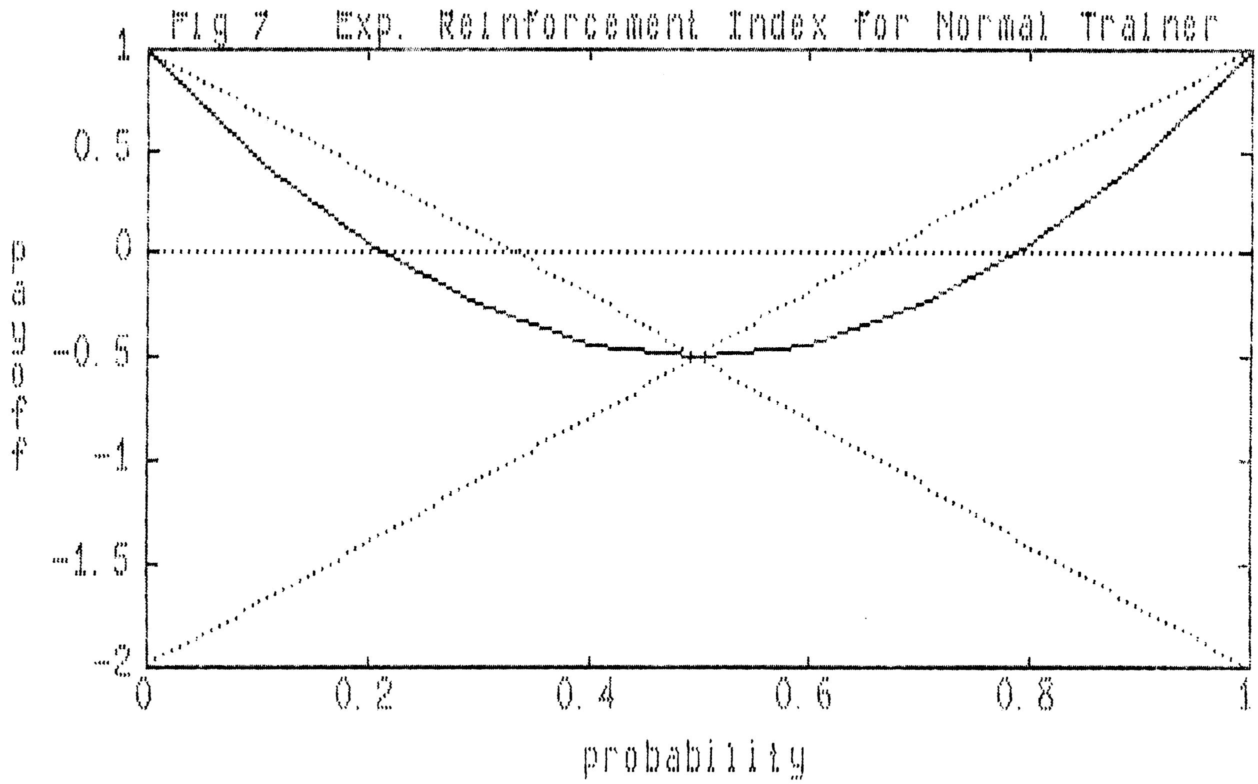


Fig 7 Exp. Reinforcement Index for Normal Trainer
payoff
probability
1
0.5
0
-0.5
-1
-1.5
-2
0
0.2
0.4
0.6
0.8
1

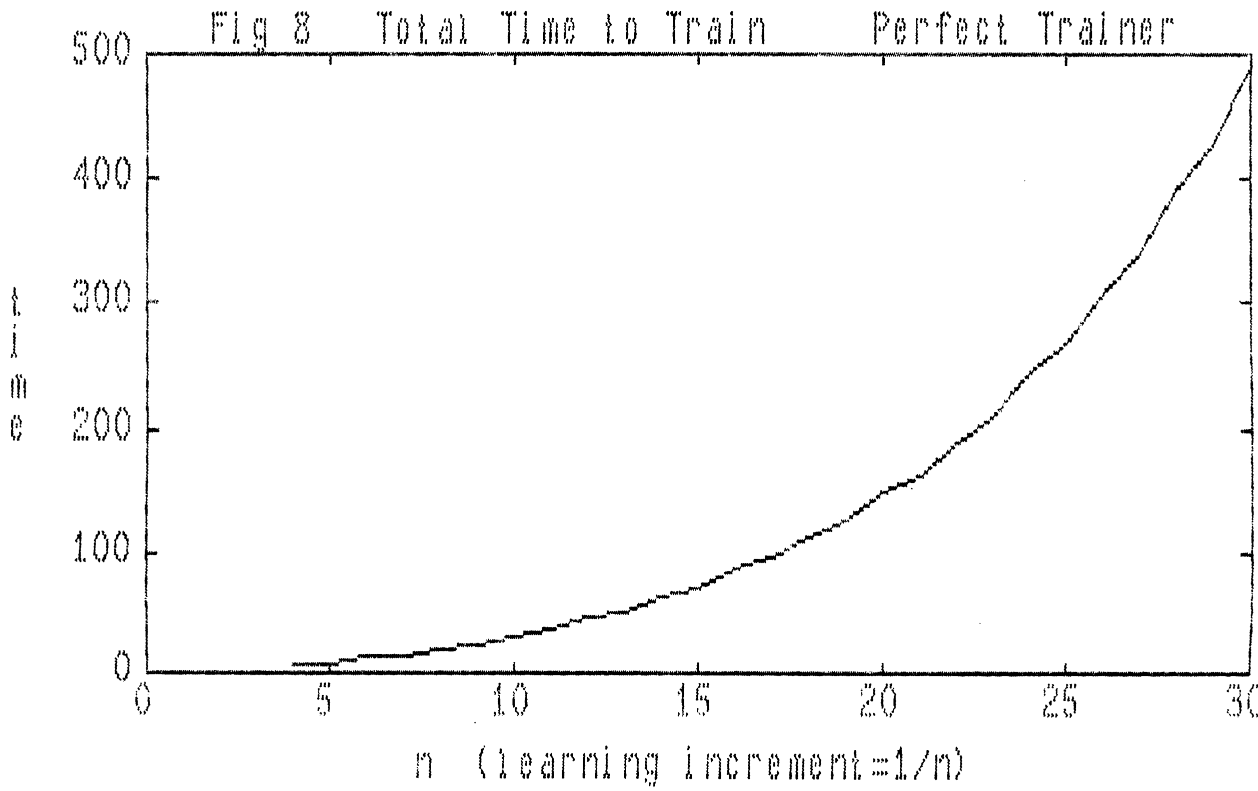


Fig 8 Total Time to Train Perfect Trainer
time
n (learning increment=1/n)
500
400
300
200
100
0
0
5
10
15
20
25
30

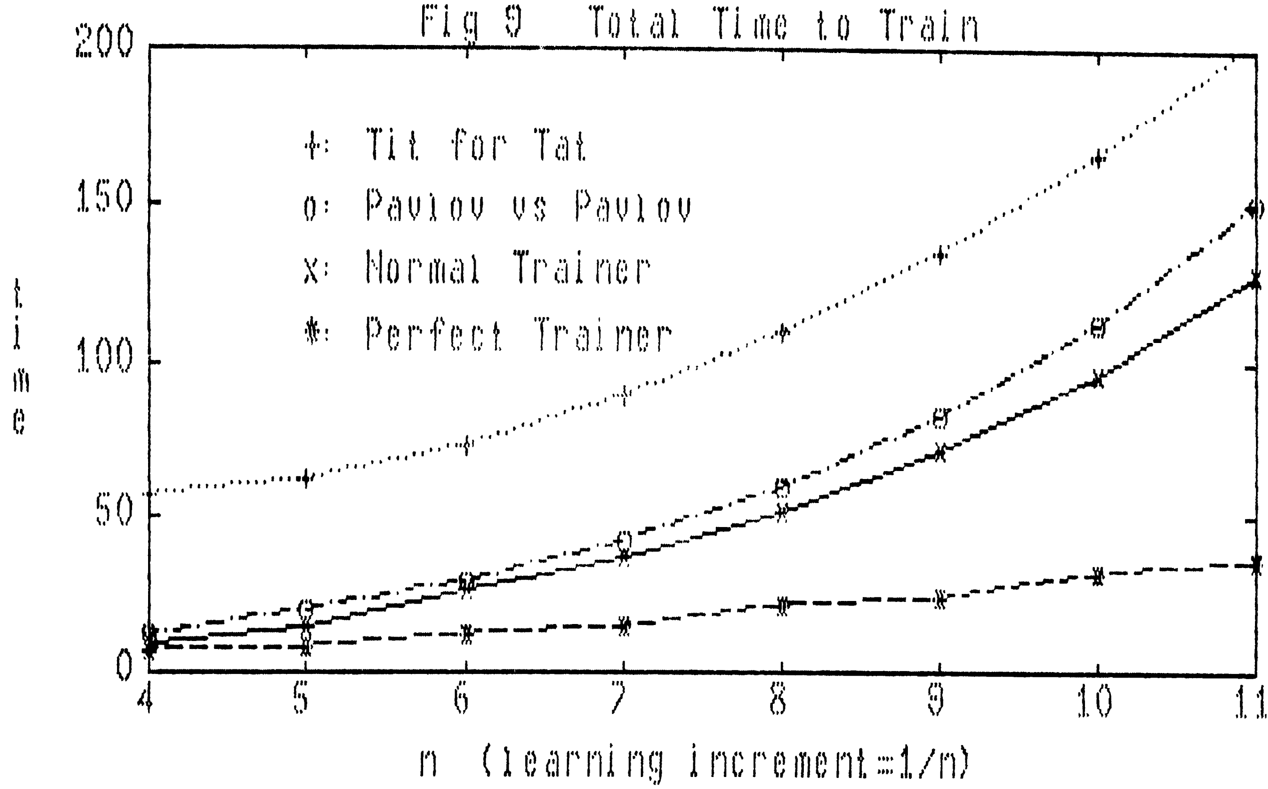
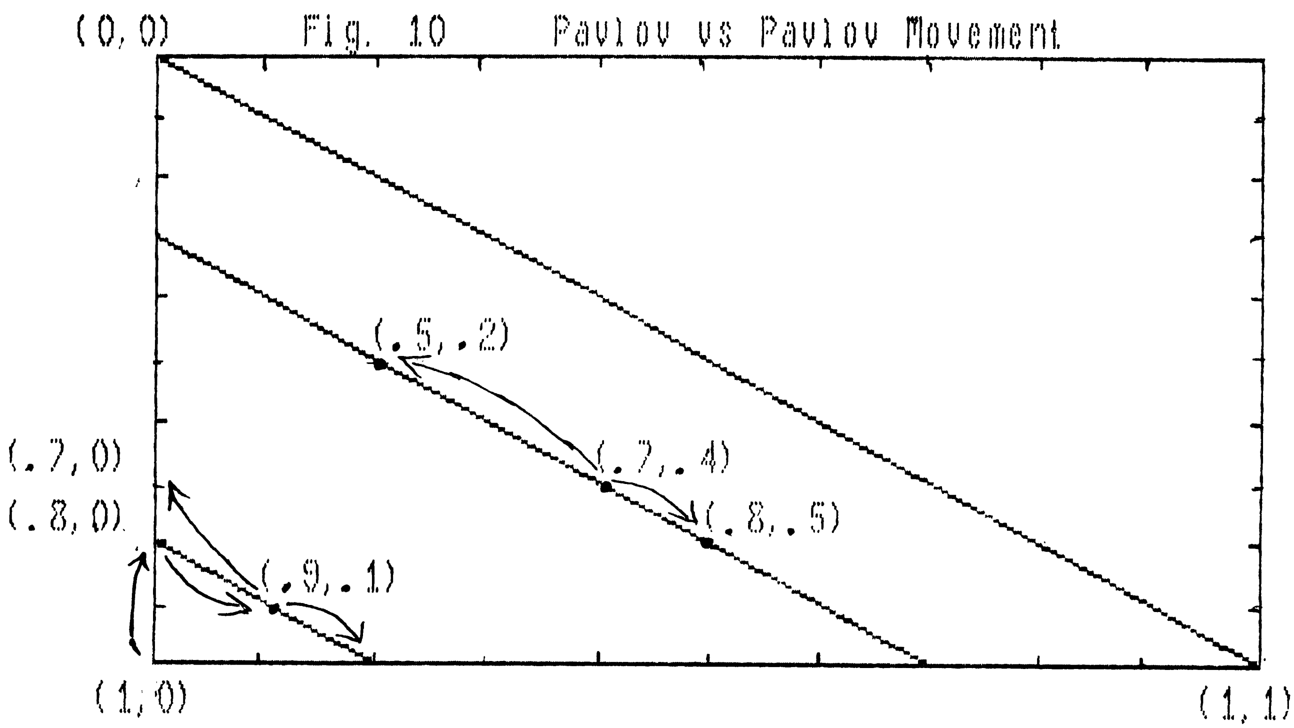


Fig 9 Total Time to Train
200
150
100
50
0
time
+: Tit for Tat
o: Pavlov vs Pavlov
x: Normal Trainer
*: Perfect Trainer
4
5
6
7
8
9
10
11
n (learning increment=1/n)


(0,0)
Fig. 10 Pavlov vs Pavlov Movement
(.5,.2)
(.7,.4)
(.8,.5)
(.7,0)
(.8,0)
(.9,.1)
(1,0)
(1,1)

APPENDIX B MATLAB Programs

```
function [equiSt, pPay, tPay] = pavsta(n)
% Stalin model, opponent always defect
% input - n, learning increment x=1/n
% output - equiSt = equilibrium state, see Table 2
%         pPay = payoff to Pavlov
%         tPay = payoff to Stalin (opponent)

%%% set up Markov matrix P.  See Table 1
P=zeros(n+1);
for i=1:n, P(i,i+1)=1-(i-1)/n; end;    %Tr^+(p) = 1-p, see eq(2)
for i=3:n,  P(i,i-2) =(i-1)/n; end;    %Tr^-(p) = p, see eq(3)
P(2,1)=1/n; P(n+1,n-1)=1;              %compute edge states separately

%%% Compute equilibrium vector
[a,d]=eig(p');              %first find the eigen vectors/values
pos=1; true=0;              %find position of the first unit eigenval.
eps=10*eps;                 %needed to avoid round off errors
while abs(true)<eps
  if abs(d(pos,pos)-1) < eps,  true=1;
  else pos=pos+1;  end;
  if abs(pos-n-2)<eps error('no eigen value=1'); end;
end;

counter=zeros(n+1,1);       %column vector that will pick out
counter(pos)=1;             %the correct eigen vector
a=a*counter;
equiSt=a./norm(a,1);        %normalize vector to be a prob. vector

%%% compute payoffs
pPay=0;  tPay=0;
for i=0:n,
  pPay=pPay+a(i+1)*(-1-i/n);        %see eq(4), PyP(p)=-1-p for p=i/n
  tPay=tPay+a(i+1)*(-1+3*i/n);      %see eq(5), PyO(p)=-1+3p.
  end;
```

* * * * * * * * * * * * * * * *

```
function [equiSt,pPay,tPay] = pavmont(n)
% Random trainer, opponent uses Monte Carlo method
% input - n, learning increment =1/n
% output - equiSt = equilibrium state , see Table 4
%          pPay = payoff to Pavlov
%          tPay = payoff to Monte

%%% compute Markov matrix P.   See Table 3.
P=zeros(n+1);
for i=1:n, P(i,i+1)=1/2; end;                                %tr^+(p)=1/2
for i=3:n,  P(i,i-2) =1/2; end;                              %tr^-(p)=1/2
P(1,1)=1/2; P(2,1)=1/2; P(n+1,n-1)=1/2; P(n+1,n+1)=1/2; %edge states

%%% compute equilibrium vector.  Left eigen vector of P is right
%eigen vector of P'
[a,d]=eig(p');              %first find the eigen vectors/values
```

```
pos=1; true=0;              % find position of first unit eigen value
eps=10*eps;                 %for round off error
while abs(true)<eps
 if abs(d(pos,pos)-1) < eps,  true=1;
  else pos=pos+1;  end;
  if abs(pos-n-2)<eps error('no eigen value=1');  end;
end;

counter=zeros(n+1,1);       %column vector that will pick out the
counter(pos)=1;             %correct eigen vector
a=a*counter;
equiSt=a./norm(a,1);

%%% compute payoffs.
pPay=0; tPay=0;
for i=0:n
  pPay=pPay+a(i+1)*(1/2-i/n);      %see eq(7),
  tPay=tPay+a(i+1)*(6*i/n-3)/2;    %see eq(8)
  end;
```

* * * * * * * * * * * * * * *

```
function oMat = pavgan(n)
%Gandhi model, opponent always cooperates
%input - n, learning increment x = 1/n
%output - oMat = table of results.  See Table 6.
%                column 1 = probability of state
%                column 2 = Total expected time to absorption
%                column 3 = total expected payoff to Pavlov
%                column 4 = average payoff per play to Pavlov
%                column 5 = total expected payoff to Gandhi
%                column 6 = average payoff per play to Gandhi
%                column 7 = prob. terminating in cooperation

%%%  Markov matrix.  The two absorbing states are put on top for
%easier computation.  Row i corresponds to prob p=(i-2)/n.  Table 5
P=zeros(n+1);
for i=4:n, P(i,i+1)=(i-2)/n;              %PyP(p)=p
           P(i,i-2) =1-P(i,i+1); end;     %PoO(p)=1-p
P(3,2)=1-1/n;  P(3,4)=1/n;                %edge states
P(1,1)=1; P(2,2)=1;
P(n+1,1)=1-1/n;  P(n+1,n-1)=1/n;

%%% computation see Section 5 and chapter 3 of [KS]
q=P(3:n+1,3:n+1);                %(n-1)x(n-1) transient states
r=P(3:n+1,1:2);
numSt=inv(eye(n-1) - q);         %time in state j before absorption,
                                 %if initially state i. Thm 3.2.4[KS]
b=numSt*r;                       %(n-1)x2 matrix of probability starting
                                 %state i, abs. in state j. Thm3.3.7[KS]
tot=numSt*ones(n-1,1);           %total time before absorption
```

```
%%% compute payoff before absorption
probSt=[1/n:1/n:1-1/n+10*eps];
pPayrate=2-probSt;              %PyP(p)=2-p
pPay=numSt*pPayrate';           %total expected payoff to Pavlov, eq(9)
tPayrate=3*probSt-2;            %PyO(p)=-2+3p
tPay=numSt*tPayrate';           %total expected payoff to Gandhi, eq(11)
apPay=pPay./tot;                %ave payoff per play to Pavlov, eq(10)
atPay=tPay./tot;                %ave payoff per play to Gandhi,eq (12)
a=[probSt' tot pPay apPay tPay atPay b(:,1)];      %table of results
last=[1 0 0 0 0 0 1];           %entries for p=1 absorbing state
oMat=[zeros(1,7);a;last];       %add last state to table
```

* * * * * * * * * * * * * * *

```
function oMat = pavtft(n);
%Tit for tat model
%input - n, learning increment = 1/n
%output - oMat = table of results.  See Table 8
%                column 1 = prob. of state: p for Dp, p+1/2n for Cp
%                column 2 = Total time to train
%                column 3 = total expected payoff to Pavlov
%                column 4 = average payoff per play to Pavlov
%                column 5 = total expected payoff to Tit for tat
%                column 6 = ave payoff per play to Tit for tat

%%% compute Markov matrix, P, a 2(n+1) square matrix
P=zeros(2*(n+1),2*(n+1));          %see Table 7
 for i=2:n-1,
    P(2*i+1,2*i-2)=i/n;            %Tr-+(Dp)=p
    P(2*i+1,2*i+3)=1-i/n;          %Tr++(Dp)=1-p
    P(2*i+2,2*i+4)=i/n;            %Tr+-(Cp)=p
    P(2*i+2,2*i-3)=1-i/n;          %Tr--(Cp)=1-p
  end;
  P(1,3)=1;  P(2,1)=1;             %edge states
  P(3,2)=1/n;  P(3,5)=1-1/n;
  P(4,1)=1-1/n;  P(4,6)=1/n;
  P(2*n+1,2*n-2)=1;  P(2*n+2,2*n+2)=1;

  q=P(1:2*n+1,1:2*n+1);            %transient portion.  See Sec. 5
  numSt = inv(eye(q) - q);         %the funcamental matrix
  total = numSt * ones(2*n+1,1); %total time before absorption
  probSt=[0:1/(2*n):1+10*eps];    %state prob.: p for Dp, p+1/2n for Cp
```

```
  %%% computation of payoffs
  Pavlov=zeros(2*n+1,1);  opponent=zeros(2*n+1,1);
  for i=0:n-1,
    x=i/n;
    Pavlov(2*i+1,1)=-1-x;          %PyP(Dp) = -1-p
    Pavlov(2*i+2,1)=2-x;           %PyP(Cp) = 2-p
    opponent(2*i+1,1)=-1+3*x;      %PyO(Dp) = -1+3p
    opponent(2*i+2,1)=-2+3*x;      %PyO(Cp) = -2+3p
    end;
  x=1-1/n;                         %separate computation for D1
  Pavlov(2*n+1,1)=-1-x;  opponent(2*n+1,1)=-1+3*x;
  pPay=numSt*Pavlov;               %expected total payoffs, eq(9)
  tPay=numSt*opponent;             %eq (11)
  Pav=pPay./total;                 %ave payoff per play, eq (10)
  Tav=tPay./total;                 %eq (12)
  oMat=[probSt',total,pPay,Pav,tPay,Tav];   %table of results
  last=[1+1/(2*n) 0 0 0 0 0];      %data for absorbing C1 state
  oMat=[oMat;last];                %add last state
```

function oMat = pavper(n)

```
%Perfect Trainer model
%input - n, learning increment  is x=1/n
%output - a table of results, as in pavtft.m, see Table 10.

%%% compute Markov matrix, P.  See Table 9.
P=zeros(n+1,n+1);
k=round(n/2);
P(1,2)=1; P(2,1)=1/n; P(2,3)=1-1/n;
for i=2:k     P(i,i+1)=1-(i-1)/n;    end;    %for p≤.5, Tr^+(p)=1-p
for i=3:k     P(i,i-2)=(i-1)/n;    end;      %for p≤.5, Tr^-(p)=p
for i=k+1:n    P(i,i+1)=(i-1)/n;    end;     %for p>.5, Tr^+(p)=p
for i=k+1:n    P(i,i-2)=1-(i-1)/n;    end;   %for p>.5, Tr^-(p)=1-p
P(n+1,n+1)=1;                                %absorbing state
q=P(1:n,1:n);                %transient states.  See Section 5
numSt=inv(eye(q)-q);         %the funcamental matrix
tot=numSt*ones(n,1);         %total time before absorption

%%% compute payoffs
pRate=zeros(n,1); tRate=zeros(n,1);
for i=1:k, x=(i-1)/n;                        %payoff rates for p≤1/2
  pRate(i)=-1-x; tRate(i)=3*x-1; end;        %PyP(p)=-1-p, PyO(p)=3p-1
for i=k+1:n, x=(i-1)/n;                      %payoff rates for p>1/2
  pRate(i)=2-x; tRate(i)=3*x-2; end;         %PyP(p)=2-p, PyO(p)=3p-2
ppayoff=numSt*pRate; tpayoff=numSt*tRate;    %eq (9) and (11)
tavct=tpayoff./tot; pavct=ppayoff./tot;      %eq (10) and (12)
probSt=(0:1/n:1-1/n+10*eps);

oMat=[probSt' tot ppayoff pavct tpayoff tavct];   %table of output
oMat=[oMat;[1 0 0 0 0 0]];                 %add data of absorbing state
```

```
function oMat = pavntr(n)
% function to compute cost of training Pavlov with normal trainer
%input - n, learning increment  is x=1/n
%output - a table of results.  See Table 13.
%          Column 1 : probability of state
%          Column 2 : expected time to train
%          Column 3 : total payoff to Pavlov or NT to train
%          Column 4 : average payoff per play to Pavlov or NT

%%% compute Markov matrix P.  See Table 12.
P=zeros(n+1,n+1);
for i=2:n x=(i-1)/n;
  P(i,i+1)=1-2*x+2*x*x; end;                    %Tr+(p)=1-2p+2p^2
for i=3:n P(i,i-2)=1-P(i,i+1); end;             %Tr-(p)=1-Tr+(p)
P(1,2)=1; P(2,1)=2/n - 2/(n*n);  P(n+1,n+1)=1;

q=P(1:n,1:n);                                   %transient states
numSt=inv(eye(n)-q);                            %fundamental matrix
count=[0:.1:1-1/n];
total = numSt*ones(n,1);
srate=zeros(n,1);
for i = 1:n, x = (i-1)/n;
  srate(i) = (-1 + 2*x); end;                   %PyP(p)=2p-1=PyO(p)
spayoff=numSt*srate;                            %eq (9)
avpayoff=spayoff./total;                        %eq (10)

oMat = [count',total,spayoff,avpayoff];         %table of results
oMat = [oMat;[1 0 0 0] ];                       %add data of abs. state
```

* * * * * * * * * * * * * * *

```
function oMat=pavsub(n,k);
%creates a submatrix for the two Pavlov version for motion along
%subdiagonal starting from (k,0).  Motion is similar to pavnrt

m=n-k+1;  q=zeros(m);                         %m is the dim of the block
for i=1:m-1
  x=(i-1)/n;  y=(k+i-1)/n;                    %prob for P1 and P2
  q(i,i+1)=x*y+(1-x)*(1-y); end;              %Tr+ = p1*p2+(1-p1)(1-p2)
for i=3:m-1  q(i,i-2)=1-q(i,i+1); end;        %Tr- = 1-Tr+
x=(m-1)/n; y=(k+m-1)/n; q(m,m-2)=x+y-2*x*y;     %end state
oMat=q;
```

```
function [time,xpayoff,numst]=pavpav(n);
%Pavlov vs Pavlov
%input - n, learning increment is x=1/n
%output - time = (n+1) square matrix, gives expected time to train
%              starting in any state (p1,p2)=(i,j).  See Table 14
%     xpayoff = (n+1) square matrix, gives expected payoff for P1
%                   See Table 15.
%    numst = The fundamental matrix.  An (n+1)(n+2)/2 square matrix
%            giving expected time before trained in each state
%            (i_1,j_1), starting from state(i_0,j_0).  Used in
%                 constructing Tables 16 and 17.

%%% compute Markov matrix P.  The states are grouped by the
%diagonals they are in, shortest one first.  There are
%1+2+...+n+(n+1) = (n+1)(n+2)/2 states

P=zeros((n+1)*(n+2)/2);
P(1,4)=1;                               %initial edge states
P(2,3)=1/n; P(2,7)=1-1/n;
P(3,4)=1-1/n; P(3,6)=1/n;

ct=4;                         %regular pattern starts in row 4
for k=3:n+1                   %to create kxk block corresponding to
                              %the subdiagonal of length k.
  P(ct:ct+k-1,ct:ct+k-1)=pavsub(n,n-k+1);    %pavsub is a program
                                       %that creates the k by k block
  if k<n, P(ct,ct+2*k+1)=1-sum(P(ct,:));     %insert the ones
                                             %outside the block.
    elseif k==n, P(ct,ct+k)=1-sum(P(ct,:));
    else P(ct,ct+1)=1;
    end;
  if k<n+1,
     P(ct+1,ct+k)=1-sum(P(ct+1,:));
     P(ct+k-1,ct+2*k)=1-sum(P(ct+k-1,:));
    else P(ct+1,ct)=1-sum(P(ct+1,:));
    end;

  ct=ct+k;
  end;

pos=(n+1)*(n+2)/2; P(pos,pos)=1;             %the absorbing state
q=P(1:pos-1,1:pos-1);                        %transient states
clear p;                                 %make room for other
                                         %computation for large n (~ 10)
numst=inv(eye(q)-q);                     %see section 5
tot=numst*ones(pos-1,1);

%%% to compute the costs, reorder the n+1 square matrix of
%probabilities (p1,p2) into a (n+1)(n+1)/2 list, ordered by
%subdiagonals as in the creation of the Markov matrix above
xrate=[0:1/n:1]'*ones(1,n+1);            %index matrix for Pav1
yrate=ones(n+1,1)*[0:1/n:1];             %index matrix for Pav2
xrtlist=zeros((n+1)*(n+2)/2+1,1);        %create lists
yrtlist=zeros((n+1)*(n+2)/2+1,1);
```

```
for i=1:n+1, for j=1:i,
  ct=(n+j-i+1)*(n+j-i)/2+j;    %index function to convert (i,j)
                               %position in the square matrix to
                               %corresponding position in list
  xrtlist(ct)=xrate(i,j);      %now the lists have the probability
  yrtlist(ct)=yrate(i,j);      %of the corresponding entry in
  end; end;                    %matrix
max=(n+1)*(n+2)/2-1;           %take out the absorbing state
xrtlist=xrtlist(1:max);  yrtlist=yrtlist(1:max);
rate1=3*yrtlist-xrtlist-ones(max,1);      %PyO(p1,p2)=3p2-p1-1
rate2=3*xrtlist-yrtlist-ones(max,1);      %PyP(p1,p2)=3p1-p2-1
clist1=numst*rate1; clist2=numst*rate2;   % eq(9) and (11)

%%% finally, convert the lists back to matrices for display
time=zeros(n+1,n+1); xcost=zeros(n+1,n+1);ycost=zeros(n+1,n+1);
ct=1;
for i=1:n
  for j=1:i
    time(n+1-i+j,j)=tot(ct,1);          %insert into the matrix
    xcost(n+1-i+j,j)=clist1(ct);        %by the  subdiagonals,
    ycost(n+1-i+j,j)=clist2(ct);          %shortest one first
    ct=ct+1;
    end;  end;
for j=1:n
  time(j,j)=tot(ct,1);             %now for the main diagonal
  xcost(j,j)=clist1(ct);
  ycost(j,j)=clist2(ct);
  ct=ct+1;
  end;
%the upper triangle of time can be filled by symmetry
for i=1:n+1    for j=i+1:n+1    time(i,j)=time(j,i); end; end;
%the upper triangle of P1 is the lower triangle of P2
xcost = xcost + ycost' -diag(diag(xcost));
```

function a = grsta1

```
%function to graph the Equilibrium vector, n=10, 20 and 30 for
%Stalin vs Pavlov.  See Figure 1.

[a10 b c] = pavsta(10);          %get data.  Since we do not need
[a20 b c] = pavsta(20);          %payoffs, store them in dummy
[a30 b c] = pavsta(30);          %variables b and c

a2pk=zeros(11,1);  a3pk=zeros(11,1);
for k=1:10,                      %pack a20 and a30 into vectors of
  a2pk(k)=a20(2*k-1)+a20(2*k);   %length 11 so we can plot them
  a3pk(d)=a30(3*k-2)+a30(3*k-1)+a30(3*k);      %together.
  end;
a2pk(11)=a20(21);  a3pk(11)=a30(31);
```

```
equivec=[a10 a2pk a3pk];
count=0:.1:1;
plot(count,equivec);hold;
plot(count,a10,'x');plot(count,a2pk,'o');plot(count,a3pk,'+');
title('Fig. 1   Equilibrium Vector     Stalin vs Pavlov');
xlabel('Pavlov Probability State');ylabel('ratio in state');
text(0.6,0.1,'x: n=10');text(0.5,0.2,'o: n=20');
text(.4,0.3,'+: n=30');
hold
```

REFERENCES

[Ax] Axelrod, R. The Evolution of Cooperation. Basic Books, New York. 1984

[AH] Axelrod, R. and Hamilton, W.D., "The evolution of Cooperation" Science 211:1390-96, 1981

[Ho] Hofstadter, D.R., Computer Tournaments of the Prisoner's Dilemma. Scientific American 248:16-26, May, 1983

[KS] Kemeny, J.G. and Snell, J.L., Finite Markov Chains, Springer Verlag, New York, 1976

[ML] PC-MATLAB, The Math Works, 20 N. Main St., Suite 250, Sherborn, MA 01770, 1985, 1987

[Ra] Rapaport, A. and A.M. Chammah, Prisoner's Dilemma. Univ of Michigan Press, Ann Arbor, 1965

Recent IMA Preprints

#	Author/s Title
280	**R. Duran**, On the Approximation of Miscible Displacement in Porous Media by a Method of Characteristics Combined with a Mixed Method
281	**H. Aixiang**, The Convergence for Nodal Expansion Method Functional Integrals
282	**V. Twersky**, Dispersive Bulk Parameters for Coherent Propagation in Correlated Random Distributions
283	**F. den Hollander**, Mixing Properties for Random Walk in Random Scenery
284	**H. R. Jauslin**, Nondifferentiable Potentials for Nonequilibrium Steady States
285	**K. Meyer, G. R. Sell**, Homoclinic Orbits and Bernoulli Bundles in Almost Periodic Systems
286	**J. Douglas, Jr., Y. Yuan**, Finite Difference Methods for the Transient Behavior of a Semiconductor Device
287	**Li Kaitai, Yan Ningning**, The Extrapolation for Boundary Finite Elements
288	**R. Durrett, R. Schonmann**, Stochastic Growth Models
289	**D. Kinderlehrer**, Remarks about Equilibrium Configurations of Crystals
290	**D. G. Aronson, J. L. Vazquez**, Eventual C^{∞}-Regularity and Concavity for Flows in One-Dimensional Porous Media
291	**L. R. Scott, J. M. Boyle, B. Bagheri**, Distributed Data Structures for Scientific Computation
292	**J. Douglas, Jr., P. J. Paes Leme, T. Arbogast, T. Schmitt**, Simulation of Flow in Naturally Fractured Petroleum Reservoirs
293	**D. G. Aronson, L. A. Caferelli**, Optimal Regularity for One-Dimensional Porous Medium Flow
294	**Haim Brezis**, Liquid Crystals and Energy Estimates for S^2-Valued Maps
295	**T. Arbogast**, Analysis of the Simulation of Single Phase Flow through a Naturally Fractured Reservoir
296	**H. Yinnian, L. Kaitai**, The Coupling Method of Finite Elements and Boundary Elements for Radiation Problems
297	**T. Cazenave, A. Haraux, L. Vazquez, F. B. Weissler**, Nonlinear Effects in Wave Equation with a Cubic Restoring Force
298	**M. Chipot, F. B. Weissler**, Some Blow-Up Results for a Nonlinear Parabolic Equation with a Gradient Term
299	**L. Kaitai**, Perturbation Solutions of Simple and Double Bifurcation Problems for Navier-Stokes Equations
300	**C. Zhangxin, L. Kaitai**, The Convergence on the Multigrid Algorithm for Navier-Stokes Equations
301	**A. Gerardi, G. Nappo**, Martingale Approach for Modeling DNA Synthesis
302	**D. N. Arnold, L. Ridgway, M. Vogelius**, Regular Inversion of the Divergence Operator with Dirichlet Boundary Conditions on a Polygon Divergence Operator with Dirichlet Boundary Conditions on a Polygon
303	**R. G. Duran**, Error Analysis in $L^p, 1 \leq p \leq \infty$, for Mixed Definite Element Methods for Linear and Quasi-Linear Elliptic Problems
304	**R. Nochetto, C. Verdi**, An Efficient Linear Scheme to Approximate Parabolic Free Free Boundary Problems: Error Estimates and Implementation
305	**K. A. Pericak-Spector, S. J. Spector**, Nonuniqueness for a Hyperbolic System: Cavitation in Nonlinear Elastodynamics
306	**E. G. Kalnins, W. Miller, Jr.**, q-Series and Orthogonal Polynomials Associate with Barnes' First Lemma
307	**D. N. Arnold, R. S. Falk**, A Uniformly Accurate Finite Element Method for Mindlin-Reissner Plate
308	**Chi-Wang Shu**, TVD Properties of a Class of Modified Eno Schemes for Scalar Conservation Laws
309	**E. Dikow, U. Hornung**, A Random Boundary Value Problem Modeling Spatial Variability in Porous Media Flow
310	**J. K. Hale**, Compact Attractors and Singular Perturbations
311	**A. Bourgeat, B. Cockburn**, The TVD-Projection Method for Solving Implicit Numeric Schemes for Scalar Conservation Laws: A Numerical Study of a Simple Case
312	**B. Muller, A. Rizzi**, Navier-Stokes Computation of Transonic Vortices over a Round Leading Edge Delta Wing
313	**J. Thomas Beale**, On the Accuracy of Vortex Methods at Large Times
314	**P. Le Talle, A. Lotfi**, Decomposition Methods for Adherence Problems in Finite Elasticity
315	**J. Douglas, Jr., J. E. Santos**, Approximation of Waves in Composite Media
316	**T. Arbogast**, The Double Porosity Model for Single Phase Flow in Naturally Fractured Reservoirs
317	**T. Arbogast, J. Douglas, Jr., J. E. Santos**, Two-Phase Immiscible Flow in Naturally Fractured Reservoirs
318	**J. Douglas, Jr., Y. Yirang**, Numerical Simulation of Immiscible Flow in Porous Media Based on Combining the Method of Characteristics with Finite Element Procedures
319	**R. Duran, R. H. Nochetto, J. Wang**, Sharp Maximum Norm Error Estimates for Finite Element Approximations of the Stokes Problem in 2-D
320	**A. Greven**, A Phase Transition for a System of Branching Random Walks in a Random Environment
321	**J. M. Harrison, R. J. Williams**, Brownian Models of Open Queueing Networks with Homogeneous Customer Populations
322	**Ana Bela Cruzeiro**, Solutions ET mesures invariantes pour des equations d'evolution Stochastiques du type Navier-Stokes
323	**Salah-Eldin A. Mohammed**, The Lyapunov Spectrum and Stable Manifolds for Stochastic Linear Delay Equations
324	**Bao Gia Nguyen**, Typical Cluster Size for 2-DIM Percolation Processes (Revised)
325	**R. Hardt, D. Kinderlehrer, F.-H. Lin**, Stable Defects of Minimizers of Constrained Variational Principles
326	**M. Chipot, D. Kinderlehrer**, Equilibrium Configurations of Crystals

Recent IMA Preprints (Continued)

\# Author/s Title

327 **Kiyosi Itô**, Malliavin's C^∞ functionals of a centered Gaussian system
328 **T. Funaki**, Derivation of the hydrodynamical equation for one-dimensional Ginzburg-Landau model
329 **Y. Masaya**, Schauder Expansion by some Quadratic Base Function
330 **F. Brezzi, J. Douglas, Jr.**, Stabilized Mixed Methods for the Stokes Problem
331 **J. Mallet-Paret, G. R. Sell**, Inertial Manifolds for Reaction Diffusion Equations in Higher Space Dimensions
332 **San-Yih Lin, M. Luskin**, Relaxation Methods for Liquid Crystal Problems
333 **H. F. Weinberger**, Some Remarks on Invariant Sets for Systems
334 **E. Miersemann, H. D. Mittelmann**, On the Continuation for Variational Inequalities Depending on an Eigenvalue Parameter
335 **J. Hulshof, N. Wolanski**, Monotone Flows in N-Dimensional Partially Saturated Porous Media: Lipschitz Continuity of the Interface
336 **B. J. Lucier**, Regularity Through Approximation for Scalar Conservation Laws
337 **B. Sturmfels**, Totally Positive Matrices and Cyclic Polytopes
338 **R. G. Duran, R. H. Nochetto**, Pointwise Accuracy of a Stable Petrov-Galerkin Approximation to Stokes Problem
339 **L. Gastaldi**, Sharp Maximum Norm Error Estimates for General Mixed Finite Element Approximations to to Second Order Elliptic Equations
340 **L. Hurwicz, H. F. Weinberger**, A Necessary Condition for Decentralizability and an Application to Intemporal Allocation
341 **G. Chavent, B. Cockburn**, The Local Projection P^0P^1-Discontinuous-Galerkin-Finite Element Method for Scalar Conservation Laws
342 **I. Capuzzo-Dolcetta, P.-L. Lions**, Hamilton-Jacobi Equations and State-Constraints Problems
343 **B. Sturmfels, N. White**, Gröbner Bases and Invariant Theory
344 **J. L. Vazquez**, C^∞-Regularity of Solutions and Interfaces of the Porous Medium Equation
345 **C. Beattie, W. M. Greenlee**, Improved Convergence Rates for Intermediate Problems
346 **H. D. Mittelmann**, Continuation Methods for Parameter-Dependent Boundary Value Problems
347 **M. Chipot, G. Michaille**, Uniqueness Results and Monotonicity Properties for Strongly Nonlinear Elliptic Variational Inequalities
348 **Avner Friedman, Bei Hu** The Stefan Problem for a Hyperbolic Heat Equation
349 **Michel Chipot, Mitchell Luskin** Existence of Solutions to the Elastohydrodynamical Equations for Magnetic Recording Systems
350 **R.H. Nochetto, C. Verdi**, The Combined Use of a Nonlinear Chernoff Formula with a Regularization Procedure for Two-Phase Stefan Problems
351 **Gonzalo R. Mendieta** Two Hyperfinite Constructions of the Brownian Bridge
352 **Victor Klee, Peter Kleinschmidt** Geometry of the Gass-Saaty Parametric Cost LP Algorithm
353 **Joseph O'Rourke** Finding A Shortest Ladder Path: A Special Case
354 **J. Gretenkort, P. Kleinschmidt, Bernd Sturmfels**, On the Existence of Certain Smooth Toric Varieties
355 **You-lan Zhu** On Stability & Convergence of Difference Schemes for Quasilinear Hyperbolic Initial-Boundary-Value Problems
356 **Hamid Bellout, Avner Friedman** Blow-Up Estimates for Nonlinear Hyperbolic Heat Equation
357 **P. Gritzman, M. Lassak** Helly-Test for the Minimal Width of Convex Bodies
358 **K.R. Meyer, G.R. Sell** Melnikov Transforms, Bernoulli Bundles, and Almost Periodic Perturbations
359 **J.-P. Puel, A. Raoult** Buckling for an Elastoplastic Plate with An Increment Constitutive Relation
360 **F.G. Garvan** A Beta Integral Associated with the Root System G_2
361 **L. Chihara, D. Stanton** Zeros of Generalized Krawtchouk Polynomials
362 **Hisashi Okamoto** O(2)-Equivariant Bifurcation Equations with Two Modes Interaction
363 **Joseph O'Rourke, Catherine Schevon** On the Development of Closed Convex Curves on 3-Polytopes
364 **Weinan E** Analysis of Spectral Methods for Burgers' Equation
365 **Weinan E** Analysis of Fourier Methods for Navier-Stokes Equation
366 **Paul Lemke** A Counterexample to a Conjecture of Abbott
367 **Peter Gritzmann** A Characterization of all Loglinear Inequalities for Three Quermassintegrals of Convex Bodies
368 **David Kinderlehrer** Phase transitions in crystals: towards the analysis of microstructure
369 **David Kraines, Vivian Kraines** Pavlov and the Prisoner's Dilemma
370 **F.G. Garvan** A Proof of the MacDonald-Morris Root System Conjecture for F_4
371 **Neil L. White** Cayley Factorization
372 **Bernd Sturmfels** Applications of Final Polynomials and Final Syzygies
373 **Avner Friedman, Michael Vogelius** Identification of Small Inhomogeneities of Extreme Conductivity by Boundary Measurements: A Continuous Dependence Result
374 **Jan Kratochvíl, Mirko Křivánek** On the Computational Complexity of Codes in Graphs
375 **Thomas I. Seidman** The Transient Semiconductor Problem with Generation Terms, II

Recent IMA Preprints (Continued)

#	Author/s Title
327	**Kiyosi Itô**, Malliavin's C^∞ functionals of a centered Gaussian system
328	**T. Funaki**, Derivation of the hydrodynamical equation for one-dimensional Ginzburg-Landau model
329	**Y. Masaya**, Schauder Expansion by some Quadratic Base Function
330	**F. Brezzi, J. Douglas, Jr.**, Stabilized Mixed Methods for the Stokes Problem
331	**J. Mallet-Paret, G. R. Sell**, Inertial Manifolds for Reaction Diffusion Equations in Higher Space Dimensions
332	**San-Yih Lin, M. Luskin**, Relaxation Methods for Liquid Crystal Problems
333	**H. F. Weinberger**, Some Remarks on Invariant Sets for Systems
334	**E. Miersemann, H. D. Mittelmann**, On the Continuation for Variational Inequalities Depending on an Eigenvalue Parameter
335	**J. Hulshof, N. Wolanski**, Monotone Flows in N-Dimensional Partially Saturated Porous Media: Lipschitz Continuity of the Interface
336	**B. J. Lucier**, Regularity Through Approximation for Scalar Conservation Laws
337	**B. Sturmfels**, Totally Positive Matrices and Cyclic Polytopes
338	**R. G. Duran, R. H. Nochetto**, Pointwise Accuracy of a Stable Petrov-Galerkin Approximation to Stokes Problem
339	**L. Gastaldi**, Sharp Maximum Norm Error Estimates for General Mixed Finite Element Approximations to to Second Order Elliptic Equations
340	**L. Hurwicz, H. F. Weinberger**, A Necessary Condition for Decentralizability and an Application to Intemporal Allocation
341	**G. Chavent, B. Cockburn**, The Local Projection P^0P^1-Discontinuous-Galerkin-Finite Element Method for Scalar Conservation Laws
342	**I. Capuzzo-Dolcetta, P.-L. Lions**, Hamilton-Jacobi Equations and State-Constraints Problems
343	**B. Sturmfels, N. White**, Gröbner Bases and Invariant Theory
344	**J. L. Vazquez**, C^∞-Regularity of Solutions and Interfaces of the Porous Medium Equation
345	**C. Beattie, W. M. Greenlee**, Improved Convergence Rates for Intermediate Problems
346	**H. D. Mittelmann**, Continuation Methods for Parameter-Dependent Boundary Value Problems
347	**M. Chipot, G. Michaille**, Uniqueness Results and Monotonicity Properties for Strongly Nonlinear Elliptic Variational Inequalities
348	**Avner Friedman, Bei Hu** The Stefan Problem for a Hyperbolic Heat Equation
349	**Michel Chipot, Mitchell Luskin** Existence of Solutions to the Elastohydrodynamical Equations for Magnetic Recording Systems
350	**R.H. Nochetto, C. Verdi**, The Combined Use of a Nonlinear Chernoff Formula with a Regularization Procedure for Two-Phase Stefan Problems
351	**Gonzalo R. Mendieta** Two Hyperfinite Constructions of the Brownian Bridge
352	**Victor Klee, Peter Kleinschmidt** Geometry of the Gass-Saaty Parametric Cost LP Algorithm
353	**Joseph O'Rourke** Finding A Shortest Ladder Path: A Special Case
354	**J. Gretenkort, P. Kleinschmidt, Bernd Sturmfels**, On the Existence of Certain Smooth Toric Varieties
355	**You-lan Zhu** On Stability & Convergence of Difference Schemes for Quasilinear Hyperbolic Initial-Boundary-Value Problems
356	**Hamid Bellout, Avner Friedman** Blow-Up Estimates for Nonlinear Hyperbolic Heat Equation
357	**P. Gritzman, M. Lassak** Helly-Test for the Minimal Width of Convex Bodies
358	**K.R. Meyer, G.R. Sell** Melnikov Transforms, Bernoulli Bundles, and Almost Periodic Perturbations
359	**J.-P. Puel, A. Raoult** Buckling for an Elastoplastic Plate with An Increment Constitutive Relation
360	**F.G. Garvan** A Beta Integral Associated with the Root System G_2
361	**L. Chihara, D. Stanton** Zeros of Generalized Krawtchouk Polynomials
362	**Hisashi Okamoto** O(2)–Equivariant Bifurcation Equations with Two Modes Interaction
363	**Joseph O'Rourke, Catherine Schevon** On the Development of Closed Convex Curves on 3-Polytopes
364	**Weinan E** Analysis of Spectral Methods for Burgers' Equation
365	**Weinan E** Analysis of Fourier Methods for Navier-Stokes Equation
366	**Paul Lemke** A Counterexample to a Conjecture of Abbott
367	**Peter Gritzmann** A Characterization of all Loglinear Inequalities for Three Quermassintegrals of Convex Bodies
368	**David Kinderlehrer** Phase transitions in crystals: towards the analysis of microstructure
369	**David Kraines, Vivian Kraines** Pavlov and the Prisoner's Dilemma
370	**F.G. Garvan** A Proof of the MacDonald-Morris Root System Conjecture for F_4
371	**Neil L. White, Tim McMillan** Cayley Factorization
372	**Bernd Sturmfels** Applications of Final Polynomials and Final Syzygies
373	**Avner Friedman, Michael Vogelius** Identification of Small Inhomogeneities of Extreme Conductivity by Boundary Measurements: A Continuous Dependence Result
374	**Jan Kratochvíl, Mirko Křivánek** On the Computational Complexity of Codes in Graphs
375	**Thomas I. Seidman** The Transient Semiconductor Problem with Generation Terms, II