

**PERSECUTOR — PREY SYSTEM RESOLUTION FOR THE PERSECUTOR
 \mathbb{R}^2 -DEFINED EXPONENTIAL PROBABILITY OF STRIKING THE PREY WITH
ADDED $[0; 1] \times [0; 1]$ -SQUARE KNOWN WHITE NOISE**

*There has been considered the persecutor — prey antagonistic system with the non-ideal exponential kernel on the $[0; 1] \times [0; 1]$ -square. The updated program module *ppsr2*, working with the accuracy parameter α and taking into account the kernel roughness, has been elaborated. Besides, there has been formed the MATLAB 7.0.1 application *eppsr2* for the visualized praxis in working with the investigated simplest antagonistic game persecution model. All the being returned solutions of the antagonistic games, if only they are not exact analytically, may be re-obtained once again in any high precision, that needed.*

*Рассмотрено антагонистическую систему преследователь — добыча с неидеальным экспоненциальным ядром на $[0; 1] \times [0; 1]$ -квадрате. Разработан обновленный программный модуль *ppsr2*, работающий с параметром меткости α и учитывающий шероховатость ядра. Кроме того, в MATLAB 7.0.1 сделано форму приложения *eppsr2* для визуализированных упражнений при работе с исследованной простейшей моделью преследования в виде антагонистической игры. Все возвращаемые решения антагонистических игр, если только они не являются аналитически точными, могут быть получены каждый раз снова с как угодно высокой точностью.*

A problem disposition and the referred origins

Conflict events and processes are modeled with various branches of the game theory. In the persecution models there take their solid place the differential games. But the solution process of the differential games is generally harder, than solving the antagonistic game. Hence, the persecution models reduction to the antagonistic game is pretty preferable to building the differential game model and searching its solution. Then may it be considered the following problem. There is the first player, shooting the goal, maneuvering with some overload y . For striking the goal there is applied the special gadget with its overload x , that is founded on a hypothesis about the goal move. Normalizing those overloads, may they be $x \in [0; 1]$ and $y \in [0; 1]$. The task of the first player, for further being called the persecutor, is to strike the opponent, for further being called the prey. And the task of the prey is to stay unstruck. This is the known persecution problem, modeled as the antagonistic game, in which the kernel is the probability

$$P(x, y) = \exp[-\alpha(x - y)^2] \quad (1)$$

of striking the prey, defined on the unit square $D_p = [0; 1] \times [0; 1]$ by some parameter $\alpha > 0$ [1, p. 62 — 63]. Though there are some definite solutions [1, p. 63 — 66] for this game, given for fixed values of the parameter α , the problem point until the paper [2] promulgation had laid in resolving that conflict system for any $\alpha > 0$. The needful resolution had been found and formed within the MATLAB 7.0.1. However, it would be ideal to have certainly the surface (1) as the kernel of the corresponding antagonistic game. Truly the kernel only reminds the surface (1), and the real kernel is the surface (1) with the $[0; 1] \times [0; 1]$ -square uniformly distributed rough edges, which may be called the white noise $n(x, y)$. So,

$$P(x, y) = \exp[-\alpha(x - y)^2] + n(x, y) \quad (2)$$

is the kernel of the persecution antagonistic game by taking into account the noised exponential probability (1). The four examples of the surface (2), plotted by the different α at the fixed noise level, are given on the figures 1 — 4.

The paper aim

Due to the said, there are the known parameter α and the surface (2), describing the nearly exponential probability of striking the prey. This simplest persecution model may reflect some important features of the conflict pair, which are very hard to be detected by the appropriate differential game model. Therefore, the disposed problem paper aim is to solve the antagonistic game with the kernel (2) in the analytic way for as wide as possible the parameter α range, and the rest of this range must be solved numerically. The antagonistic game resolution should be formed within a program module for the fast visualized representation of the optimal behavior of the persecutor and prey.

The exponential probability kernel

Now turn backward the paper [2] and resume the obtained there results. Setting $n(x, y) = 0$, start recalling, that it had been reasonable to check if the stated game with the kernel (1) is either convex or concave. That will help to solve it with the known algorithm for convex-concave antagonistic games [3 — 5]. The first derivative of the function (1) by the variable x is

$$\frac{\partial P(x, y)}{\partial x} = \frac{\partial}{\partial x} \left(\exp[-\alpha(x-y)^2] \right) = -2\alpha(x-y) \exp[-\alpha(x-y)^2] \quad (3)$$

and its second derivative is

$$\begin{aligned} \frac{\partial^2 P(x, y)}{\partial x^2} &= \frac{\partial}{\partial x} \left(-2\alpha(x-y) \exp[-\alpha(x-y)^2] \right) = -2\alpha \exp[-\alpha(x-y)^2] + 4\alpha^2(x-y)^2 \exp[-\alpha(x-y)^2] = \\ &= 2\alpha(2\alpha(x-y)^2 - 1) \exp[-\alpha(x-y)^2]. \end{aligned} \quad (4)$$

The concavity condition $\frac{\partial^2 P(x, y)}{\partial x^2} \leq 0$ must be true $\forall x \in [0; 1]$ and $\forall y \in [0; 1]$. As there is the triple inequality

$$0 < \exp(-\alpha) \leq \exp[-\alpha(x-y)^2] \leq 1, \quad (5)$$

then the inequality

$$2\alpha(2\alpha(x-y)^2 - 1) \exp[-\alpha(x-y)^2] \leq 0 \quad (6)$$

is identical to the inequality

$$2\alpha(x-y)^2 - 1 \leq 0, \quad (7)$$

$P(x, y)$

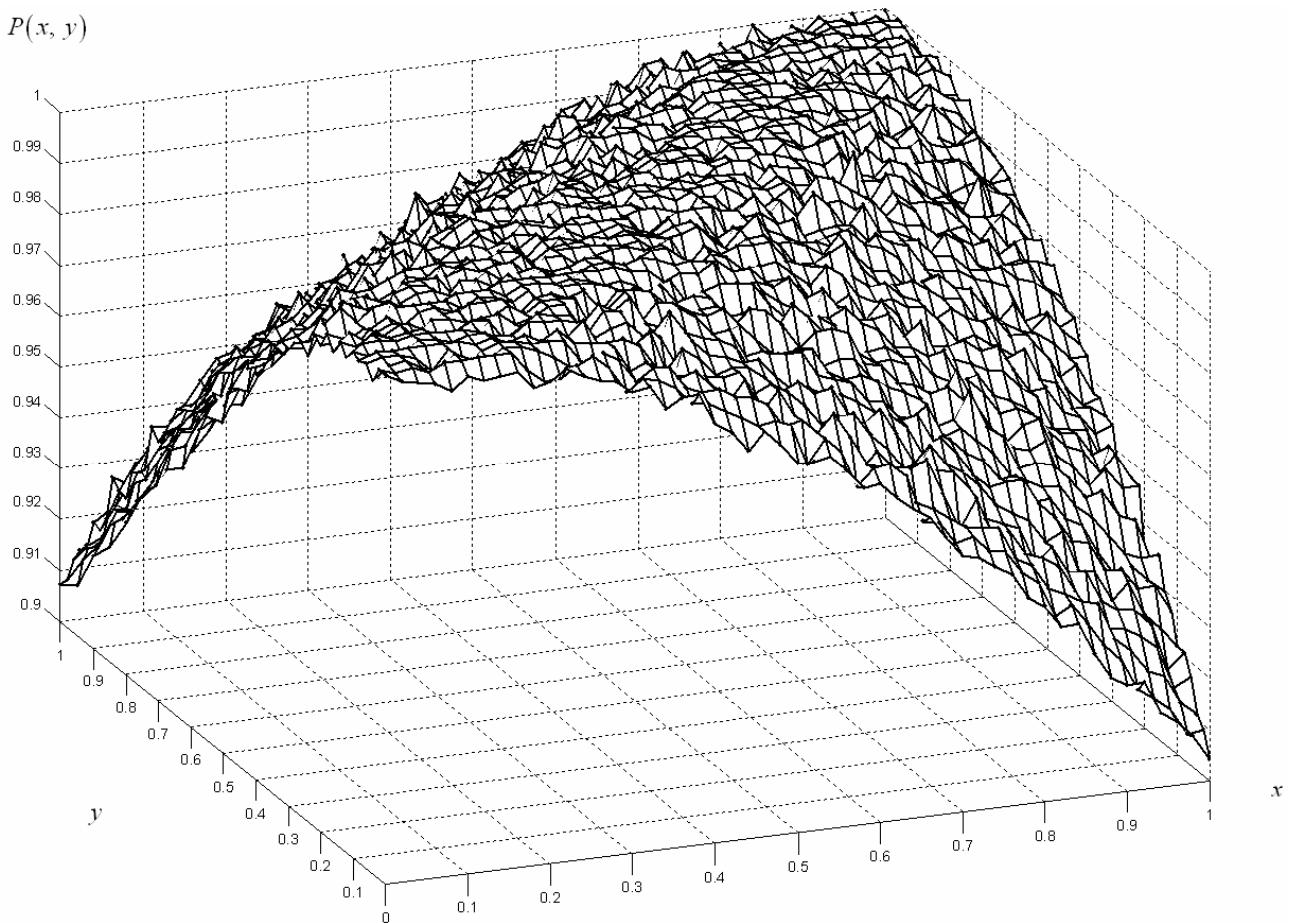


Figure 1. The surface (2) by $\alpha = 0.1$

$P(x, y)$

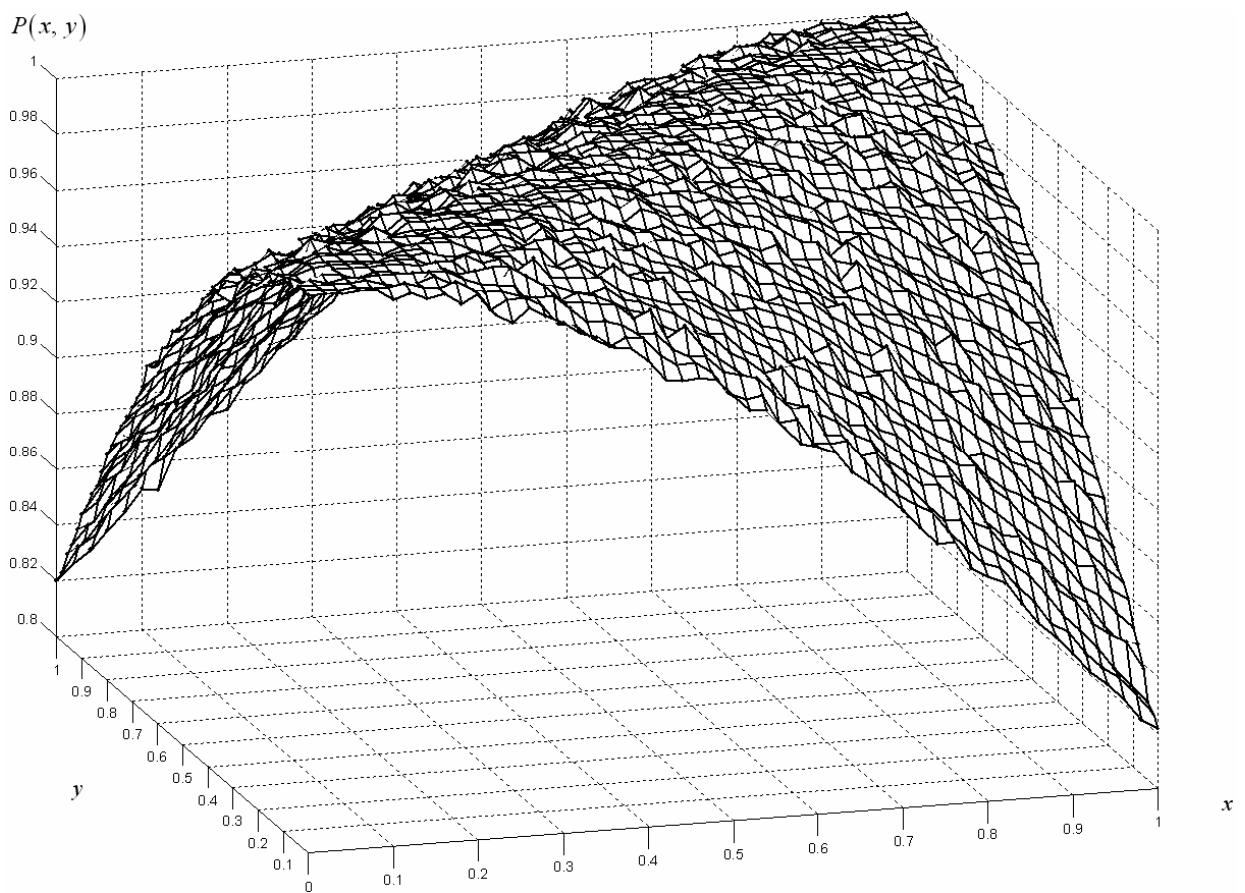


Figure 2. The surface (2) by $\alpha = 0.2$

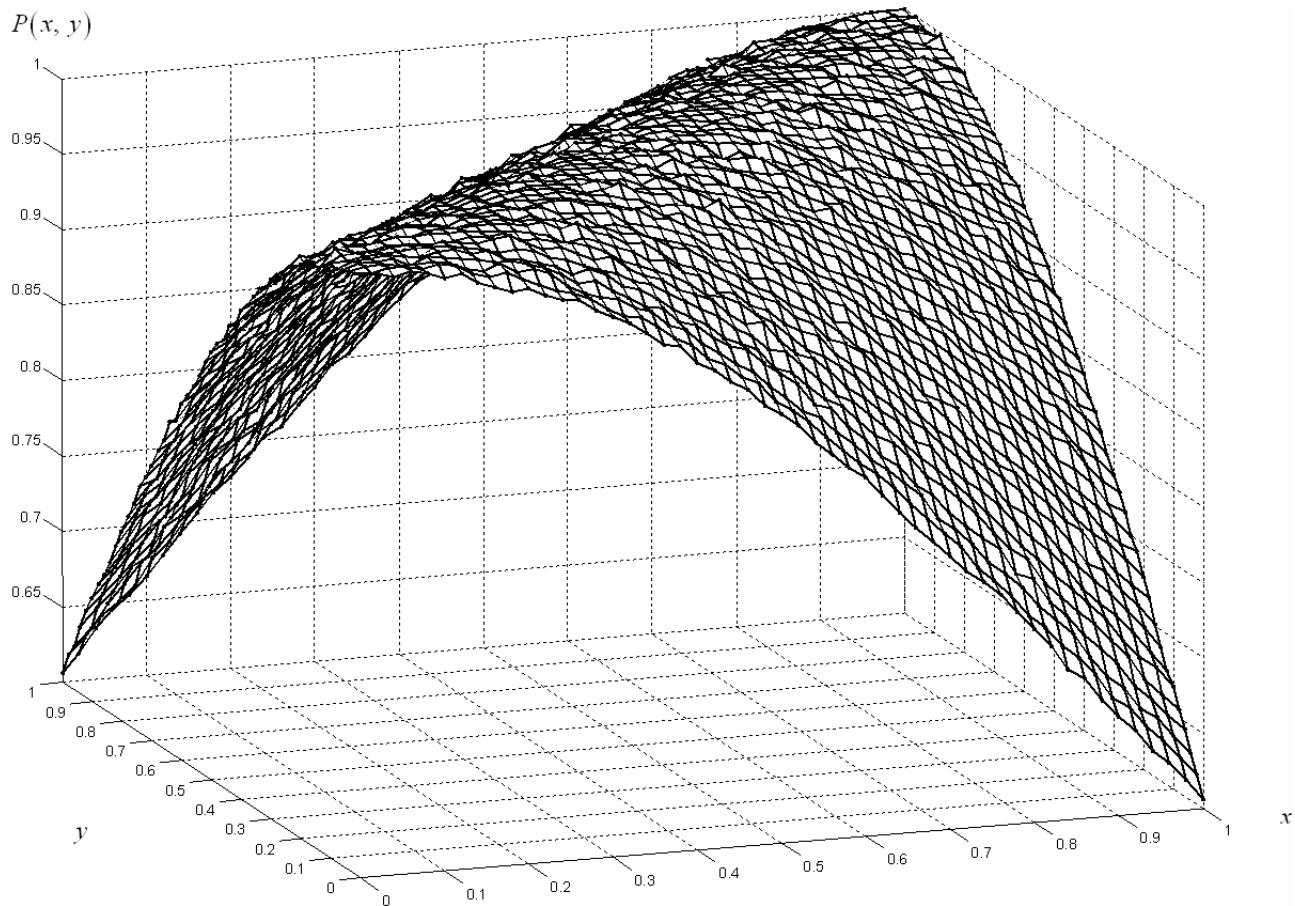


Figure 3. The surface (2) by $\alpha = 0.5$

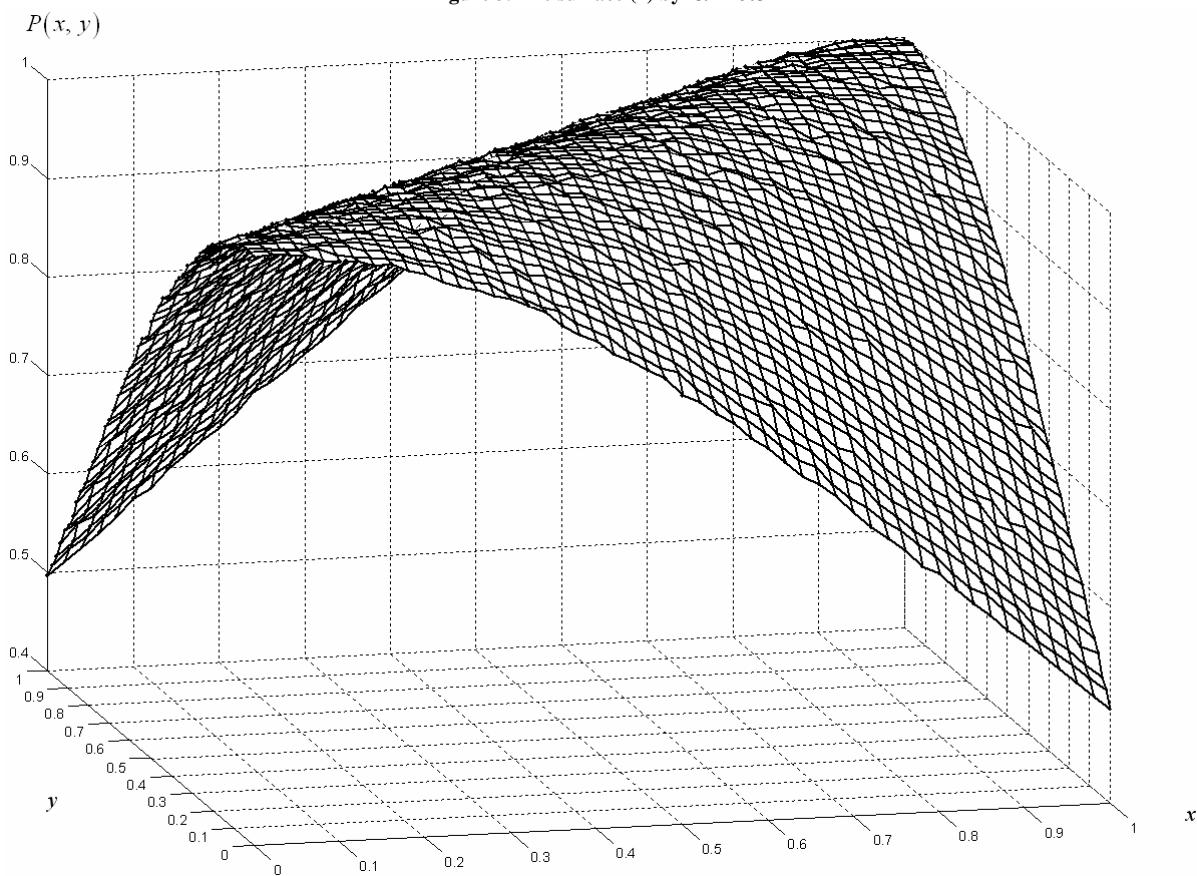


Figure 4. The surface (2) by $\alpha = 0.7$

whence, minding the obvious factor $(x-y)^2 \in [0; 1]$, the parameter $\alpha \in \left(0; \frac{1}{2}\right]$ determines the concavity of the antagonistic game with the kernel (1).

Continuing, now clear out whether this game is convex or not. The first derivative of the function (1) by the variable y is

$$\frac{\partial P(x, y)}{\partial y} = \frac{\partial}{\partial y} \left(\exp[-\alpha(x-y)^2] \right) = 2\alpha(x-y) \exp[-\alpha(x-y)^2] \quad (8)$$

and its second derivative

$$\begin{aligned} \frac{\partial^2 P(x, y)}{\partial y^2} &= \frac{\partial}{\partial y} \left(2\alpha(x-y) \exp[-\alpha(x-y)^2] \right) = -2\alpha \exp[-\alpha(x-y)^2] + 4\alpha^2(x-y)^2 \exp[-\alpha(x-y)^2] = \\ &= 2\alpha \left(2\alpha(x-y)^2 - 1 \right) \exp[-\alpha(x-y)^2] \end{aligned} \quad (9)$$

turns to be identical to the second derivative (4) of the function (1) by the variable x . Then the corresponding convexity condition

$$2\alpha(x-y)^2 - 1 \geq 0 \quad (10)$$

is impracticable for any $\alpha > 0$ as this fails yet for $x = y$.

Accepting the parameter $\alpha \in \left(0; \frac{1}{2}\right]$ after the stated (3) — (10), the minimum of the surface (1) as the function of the variable y on the segment $[0; 1]$ is

$$\min_{y \in [0; 1]} P(x, y) = \min_{y \in [0; 1]} \exp[-\alpha(x-y)^2] = P(x, 1) = \exp[-\alpha(x-1)^2] \quad \forall x \in \left[0; \frac{1}{2}\right] \quad (11)$$

and

$$\min_{y \in [0; 1]} P(x, y) = \min_{y \in [0; 1]} \exp[-\alpha(x-y)^2] = P(x, 0) = \exp(-\alpha x^2) \quad \forall x \in \left[\frac{1}{2}; 1\right]. \quad (12)$$

The optimal game value

$$\begin{aligned} v_{\text{opt}} &= \max_{x \in [0; 1]} \min_{y \in [0; 1]} P(x, y) = \max \left\{ \max_{x \in \left[0; \frac{1}{2}\right]} P(x, 1), \max_{x \in \left[\frac{1}{2}; 1\right]} P(x, 0) \right\} = \\ &= \max \left\{ \max_{x \in \left[0; \frac{1}{2}\right]} \left(\exp[-\alpha(x-1)^2] \right), \max_{x \in \left[\frac{1}{2}; 1\right]} \left[\exp(-\alpha x^2) \right] \right\} = \exp\left(-\alpha\left(\frac{1}{2}\right)^2\right) = \exp\left(-\frac{\alpha}{4}\right) \end{aligned} \quad (13)$$

is reached on the optimal pure strategy $x_{\text{opt}} = \frac{1}{2}$ of the first player. The essential pure strategies of the second player are the roots of the standard equation

$$v_{\text{opt}} = \exp\left(-\frac{\alpha}{4}\right) = P(x_{\text{opt}}, y) = \exp\left[-\alpha(x_{\text{opt}} - y)^2\right] = P\left(\frac{1}{2}, y\right) = \exp\left[-\alpha\left(\frac{1}{2} - y\right)^2\right]. \quad (14)$$

Going on, as $-\frac{\alpha}{4} = -\frac{\alpha}{4} + \alpha y - \alpha y^2$ and $y(1-y)=0$ then the roots of the equation (14) are the pure strategies $y = y_1 = 0$ and $y = y_2 = 1$.

Further on, may $\varphi(y)$ be the optimal probability of the pure strategy y selection by the second player. Then the optimal probability of the pure strategy $y_1 = 0$ selection is determined from the equilibrium situation nonstrict inequality

$$\begin{aligned}
 P(x, y_1)\varphi(y_1) + P(x, y_2)\varphi(y_2) &= P(x, y_1)\varphi(y_1) + P(x, y_2)[1 - \varphi(y_1)] = \\
 &= P(x, 0)\varphi(0) + P(x, 1)[1 - \varphi(0)] = \varphi(0)\exp(-\alpha x^2) + [1 - \varphi(0)]\exp[-\alpha(x-1)^2] \leqslant \\
 &\leqslant P(x_{\text{opt}}, y_1)\varphi(y_1) + P(x_{\text{opt}}, y_2)\varphi(y_2) = P(x_{\text{opt}}, y_1)\varphi(y_1) + P(x_{\text{opt}}, y_2)[1 - \varphi(y_1)] = \\
 &= \varphi(0)\exp\left(-\frac{\alpha}{4}\right) + [1 - \varphi(0)]\exp\left(-\frac{\alpha}{4}\right) = \exp\left(-\frac{\alpha}{4}\right) = v_{\text{opt}}
 \end{aligned} \tag{15}$$

by $x \neq x_{\text{opt}} = \frac{1}{2}$, where just has been used the equilibrium point conception with the condition $\varphi(y_1) + \varphi(y_2) = 1$ [6 — 10]. In order of (15) there is the subsequent inequality

$$\varphi(0)\left(\exp(-\alpha x^2) - \exp[-\alpha(x-1)^2]\right) \leqslant \exp\left(-\frac{\alpha}{4}\right) - \exp[-\alpha(x-1)^2], \tag{16}$$

which components should be evaluated for $x \in \left[0; \frac{1}{2}\right]$ and $x \in \left[\frac{1}{2}; 1\right]$ by the corresponding minima (11) and (12) of the kernel (1).

As it is easy to see (figure 5), there is the relation

$$\exp(-\alpha x^2) > \exp[-\alpha(x-1)^2] \tag{17}$$

$\forall x \in \left[0; \frac{1}{2}\right]$ and that conditionally gives

$$\varphi(0) \leqslant \frac{\exp\left(-\frac{\alpha}{4}\right) - \exp[-\alpha(x-1)^2]}{\exp(-\alpha x^2) - \exp[-\alpha(x-1)^2]}. \tag{18}$$

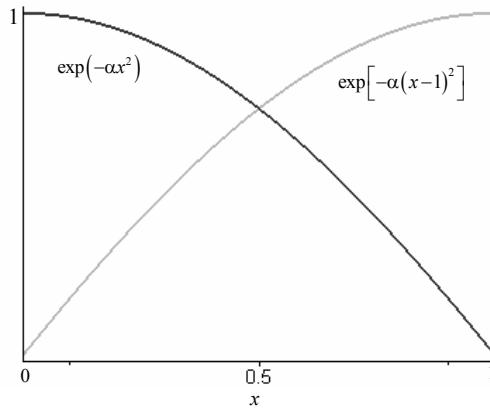


Figure 5. Relation between two exponents in the left side of the inequality (16)

The ratio in the right side of the inequality (18) is the monotonously decreasing curve by any fixed $\alpha \in \left(0; \frac{1}{2}\right]$ and $\forall x \in \left[0; \frac{1}{2}\right]$ (figure 6). But the limit

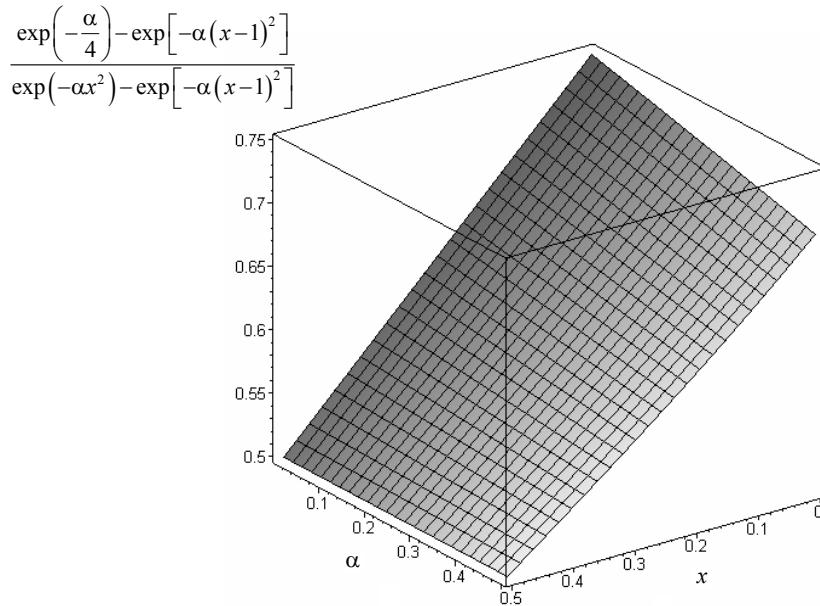


Figure 6. The right side of the inequality (18), that imaged as a surface

$$\begin{aligned}
 & \lim_{\substack{x \rightarrow \frac{1}{2} - \varepsilon \\ \varepsilon > 0}} \left(\frac{\exp\left(-\frac{\alpha}{4}\right) - \exp[-\alpha(x-1)^2]}{\exp(-\alpha x^2) - \exp[-\alpha(x-1)^2]} \right) = \lim_{\substack{x \rightarrow \frac{1}{2} - \varepsilon \\ \varepsilon > 0}} \left(\frac{\frac{\partial}{\partial x} \left(\exp\left(-\frac{\alpha}{4}\right) - \exp[-\alpha(x-1)^2] \right)}{\frac{\partial}{\partial x} \left(\exp(-\alpha x^2) - \exp[-\alpha(x-1)^2] \right)} \right) = \\
 & = \lim_{\substack{x \rightarrow \frac{1}{2} - \varepsilon \\ \varepsilon > 0}} \left(\frac{2\alpha(x-1)\exp[-\alpha(x-1)^2]}{-2\alpha x \exp(-\alpha x^2) + 2\alpha(x-1)\exp[-\alpha(x-1)^2]} \right) = \lim_{\substack{x \rightarrow \frac{1}{2} - \varepsilon \\ \varepsilon > 0}} \left(\frac{-\alpha \exp\left(-\frac{\alpha}{4}\right)}{-\alpha \exp\left(-\frac{\alpha}{4}\right) - \alpha \exp\left(-\frac{\alpha}{4}\right)} \right) = \frac{1}{2}
 \end{aligned} \tag{19}$$

displays that $\forall x \in \left[0; \frac{1}{2}\right]$ the optimal probability of the pure strategy $y_1 = 0$ selection satisfies the condition

$$\varphi(0) \in \left[0; \frac{1}{2}\right] = \Phi_1. \tag{20}$$

Taking $x \in \left(\frac{1}{2}; 1\right]$ leads from the statement (16) through the relation

$$\exp(-\alpha x^2) < \exp[-\alpha(x-1)^2] \tag{21}$$

to the inequality

$$\varphi(0) \geq \frac{\exp\left(-\frac{\alpha}{4}\right) - \exp[-\alpha(x-1)^2]}{\exp(-\alpha x^2) - \exp[-\alpha(x-1)^2]}. \tag{22}$$

The ratio in the right side of the inequality (22) is the monotonously decreasing curve by any fixed $\alpha \in \left(0; \frac{1}{2}\right]$ and $\forall x \in \left(\frac{1}{2}; 1\right]$ (figure 7). But the limit

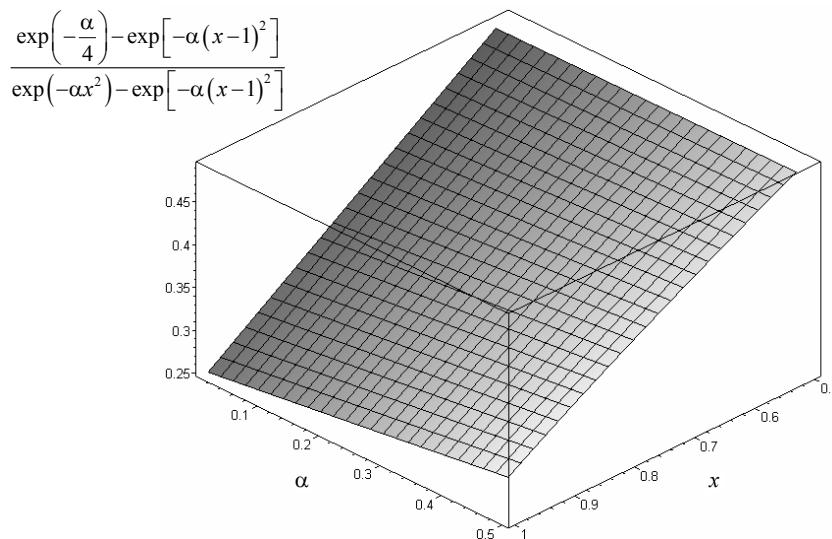


Figure 7. The right side of the inequality (22), that imaged as a surface

$$\lim_{\substack{x \rightarrow \frac{1}{2} + \varepsilon \\ \varepsilon > 0}} \left(\frac{\exp\left(-\frac{\alpha}{4}\right) - \exp\left[-\alpha(x-1)^2\right]}{\exp(-\alpha x^2) - \exp\left[-\alpha(x-1)^2\right]} \right) = \frac{1}{2} \quad (23)$$

displays that $\forall x \in \left(\frac{1}{2}; 1\right]$ the optimal probability of the pure strategy $y_1 = 0$ selection satisfies the condition

$$\varphi(0) \in \left[\frac{1}{2}; 1\right] = \Phi_2. \quad (24)$$

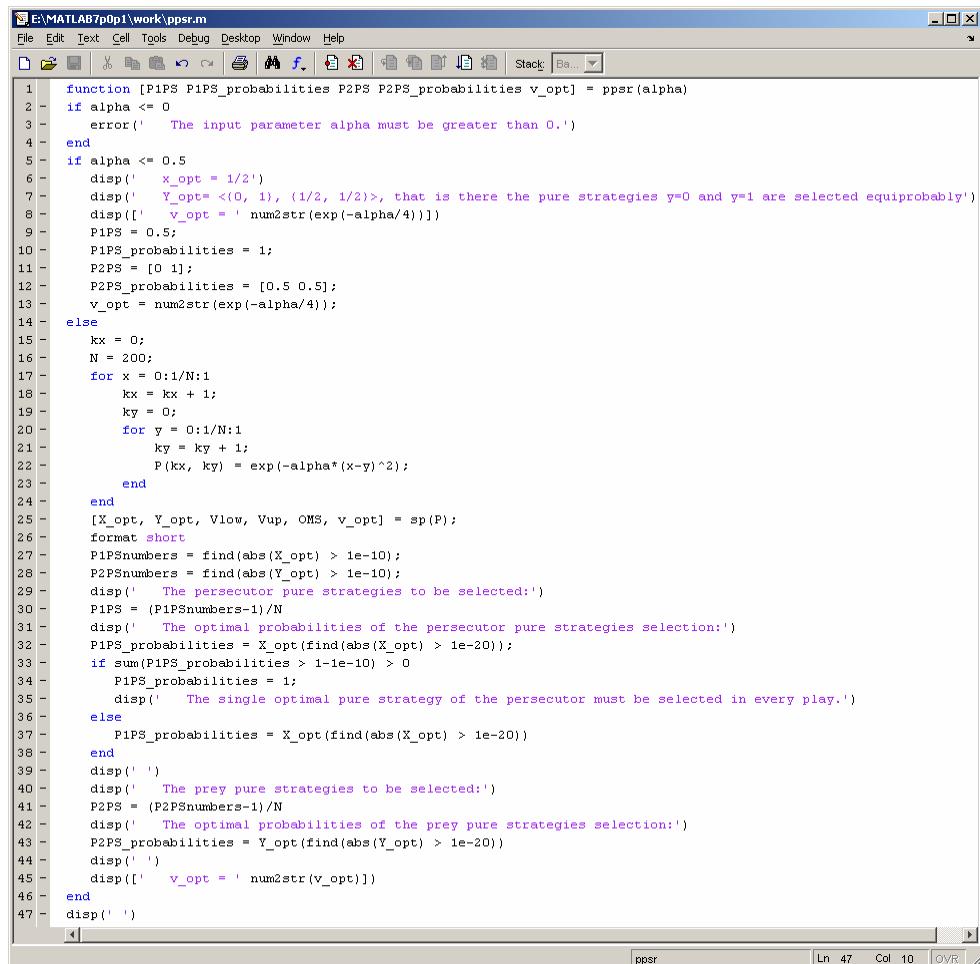
Consequently, the optimal probability of the pure strategy $y_1 = 0$ selection is

$$\varphi(0) \in \Phi_1 \cap \Phi_2 = \left[0; \frac{1}{2}\right] \cap \left[\frac{1}{2}; 1\right] = \left\{\frac{1}{2}\right\}. \quad (25)$$

Hence the optimal probability of the pure strategy $y_2 = 1$ selection is $\varphi(1) = \frac{1}{2}$. Then the game with the defined on the unit square $D_p = [0; 1] \times [0; 1]$ kernel (1) by the parameter $\alpha \in \left(0; \frac{1}{2}\right]$ is solved in the single optimal strategy of the first player $x_{\text{opt}} = \frac{1}{2}$ and the single mixed optimal strategy of the second player, which consists in the equiprobable selection of the two pure strategies $y_1 = 0$ and $y_2 = 1$ due to (19) — (25). Such equilibrium situation gives the optimal game value $v_{\text{opt}} = \exp\left(-\frac{\alpha}{4}\right)$. And by this system persecutor — prey configuration the likelihood of striking the prey is sufficiently great, as even in the worst case, that is by $\alpha = \frac{1}{2}$, the probability of the strike is $v_{\text{opt}} = \exp\left(-\frac{1}{8}\right) \approx 0.8825$.

However, the parameter $\alpha \in \left(0; \frac{1}{2}\right]$ is a feature of the high-technology equipment for the persecutor. So, the case $\alpha > \frac{1}{2}$ should be solved carefully as well as the previous. The exact analytic methods of solving such system persecutor — prey configuration are unavailable. Then for getting the needful solution there may be applied the MATLAB 7.0.1 software for numerical computations. The viewed on the figure 8 program module ppsr (Persecutor — Prey System Resolution) handles the single input α and returns the result as the solution of the game with the defined on the unit square $D_p = [0; 1] \times [0; 1]$ kernel (1). For instance, if $\alpha = \frac{3}{4}$ then the solution is much the same as for the

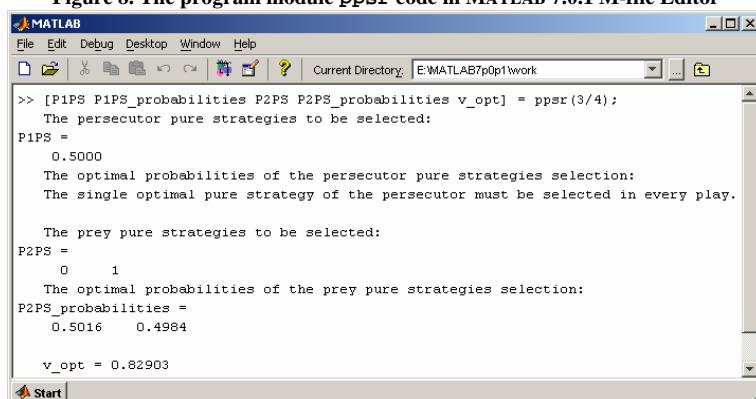
cases with $\alpha \in \left(0; \frac{1}{2}\right]$ (figure 9). Actually, the carried investigations prompt that for $\alpha \in (0; 2]$ the solution stays nearly stable. Some insignificant deviation from the equiprobable selection of the two pure strategies $y_1 = 0$ and $y_2 = 1$ is explained with the finite precision of calculations, which mainly are in the program submodule `sp`, accepting the kernel in the matrix form and returning the whole game solution (figure 10). Nevertheless, the single optimal pure strategy of the persecutor $x_{\text{opt}} = \frac{1}{2}$ remains for $\alpha \in (0; 2]$ definitely. And only for $\alpha > 2$ the number of pure strategies of the players, selected with nonzero optimal probabilities, starts increasing (figures 11 — 19).



```

1 function [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(alpha)
2 if alpha <= 0
3     error(' The input parameter alpha must be greater than 0.')
4 end
5 if alpha <= 0.5
6     disp(' x_opt = 1/2')
7     disp([' Y_opt<(0, 1), (1/2, 1/2)>, that is there the pure strategies y=0 and y=1 are selected equiprobably'])
8     disp([' v_opt = ' num2str(exp(-alpha/4))])
9     P1PS = 0.5;
10    P1PS_probabilities = 1;
11    P2PS = [0 1];
12    P2PS_probabilities = [0.5 0.5];
13    v_opt = num2str(exp(-alpha/4));
14 else
15     kx = 0;
16     N = 200;
17     for x = 0:1/N:1
18         kx = kx + 1;
19         ky = 0;
20         for y = 0:1/N:1
21             ky = ky + 1;
22             P(kx, ky) = exp(-alpha*(x-y)^2);
23         end
24     end
25     [X_opt, Y_opt, Vlow, Vup, OMS, v_opt] = sp(P);
26     format short
27     P1PSnumbers = find(abs(X_opt) > 1e-10);
28     P2PSnumbers = find(abs(Y_opt) > 1e-10);
29     disp(' The persecutor pure strategies to be selected:')
30     P1PS = (P1PSnumbers-1)/N
31     disp(' The optimal probabilities of the persecutor pure strategies selection:')
32     P1PS_probabilities = X_opt(find(abs(X_opt) > 1e-20));
33     if sum(P1PS_probabilities > 1e-10) > 0
34         P1PS_probabilities = 1;
35     else
36         disp(' The single optimal pure strategy of the persecutor must be selected in every play.')
37     else
38         P1PS_probabilities = X_opt(find(abs(X_opt) > 1e-20));
39     end
40     disp(' ')
41     disp(' The prey pure strategies to be selected:')
42     P2PS = (P2PSnumbers-1)/N
43     disp(' The optimal probabilities of the prey pure strategies selection:')
44     P2PS_probabilities = Y_opt(find(abs(Y_opt) > 1e-20));
45     disp([' v_opt = ' num2str(v_opt)])
46 end
47 disp(' ')

```

Figure 8. The program module `ppsr` code in MATLAB 7.0.1 M-file Editor


```

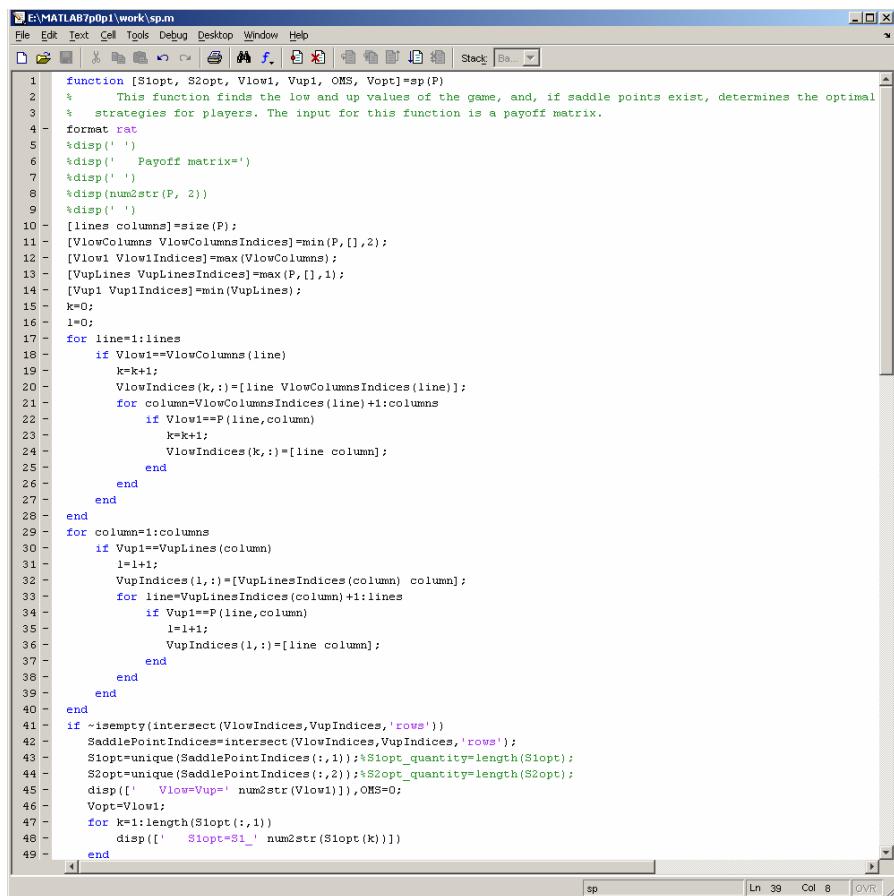
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(3/4);
The persecutor pure strategies to be selected:
P1PS =
0.5000
The optimal probabilities of the persecutor pure strategies selection:
The single optimal pure strategy of the persecutor must be selected in every play.

The prey pure strategies to be selected:
P2PS =
0 1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
0.5016 0.4984

v_opt = 0.82903

```

Figure 9. The persecutor — prey system resolution for $\alpha = 0.75$

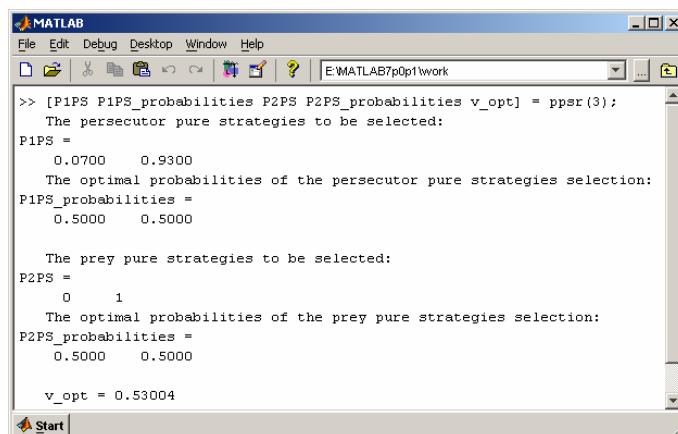


```

1 function [S1opt, S2opt, Vlowl, Vup1, OMS, Vopt]=sp(P)
2 % This function finds the low and up values of the game, and, if saddle points exist, determines the optimal
3 % strategies for players. The input for this function is a payoff matrix.
4 format rat
5 %disp(' ')
6 %disp(' Payoff matrix=')
7 %disp(' ')
8 %disp(num2str(P, 2))
9 %disp(' ')
10 [lines columns]=size(P);
11 [VlowColumns VlowColumnsIndices]=min(P,[],2);
12 [Vlow1 Vlow1Indices]=max(VlowColumns);
13 [VupLines VupLinesIndices]=max(P,[],1);
14 [Vup1 Vup1Indices]=min(VupLines);
15 k=0;
16 l=0;
17 for line=1:lines
18 if Vlow1==VlowColumns(line)
19 k=k+1;
20 VlowIndices(k,:)=[line VlowColumnsIndices(line)];
21 for column=VlowColumnsIndices(line)+1:columns
22 if Vlow1==P(line,column)
23 k=k+1;
24 VlowIndices(k,:)=[line column];
25 end
26 end
27 end
28 for column=1:columns
29 if Vup1==VupLines(column)
30 l=l+1;
31 VupIndices(l,:)=[VupLinesIndices(column) column];
32 for line=VupLinesIndices(column)+1:lines
33 if Vup1==P(line,column)
34 l=l+1;
35 VupIndices(l,:)=[line column];
36 end
37 end
38 end
39 end
40 end
41 if ~isempty(intersect(VlowIndices,VupIndices,'rows'))
42 SaddlePointIndices=intersect(VlowIndices,VupIndices,'rows');
43 S1opt=unique(SaddlePointIndices(:,1));%S1opt_quantity=length(S1opt);
44 S2opt=unique(SaddlePointIndices(:,2));%S2opt_quantity=length(S2opt);
45 disp([' Vlow=Vup=' num2str(Vlow1)]);
46 Vopt=Vlowl;
47 for k=1:length(S1opt(:,1))
48 disp([' S1opt=' num2str(S1opt(k))]);
49 end

```

Figure 10. The program submodule **sp** code in MATLAB 7.0.1 M-file Editor



```

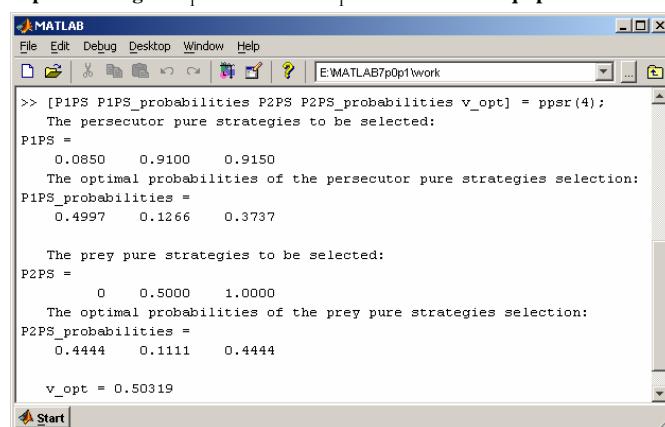
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(3);
The persecutor pure strategies to be selected:
P1PS =
0.0700 0.9300
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
0.5000 0.5000

The prey pure strategies to be selected:
P2PS =
0 1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
0.5000 0.5000

v_opt = 0.53004

```

Figure 11. Appearance of the two pure strategies $x_1 = 0.03$ and $x_1 = 0.97$ for the equiprobable selection of the persecutor by $\alpha = 3$



```

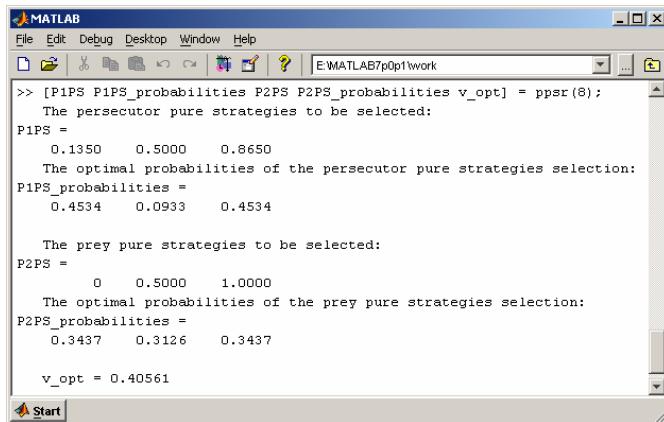
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(4);
The persecutor pure strategies to be selected:
P1PS =
0.0850 0.9100 0.9150
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
0.4997 0.1266 0.3737

The prey pure strategies to be selected:
P2PS =
0 0.5000 1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
0.4444 0.1111 0.4444

v_opt = 0.50319

```

Figure 12. By $\alpha = 4$ both the players have the three pure strategies for their selection



```

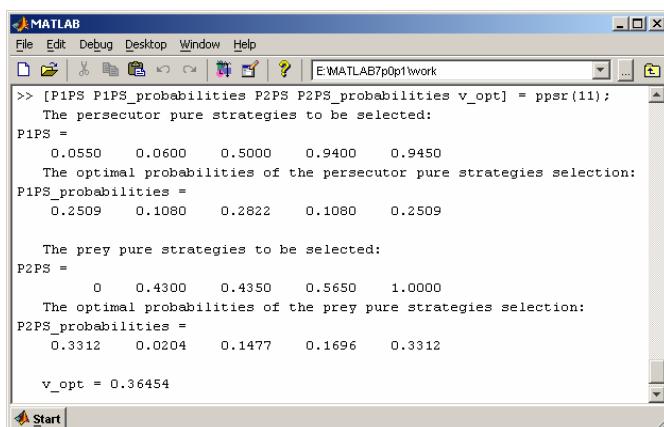
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(8);
The persecutor pure strategies to be selected:
P1PS =
    0.1350    0.5000    0.8650
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.4534    0.0933    0.4534

The prey pure strategies to be selected:
P2PS =
    0    0.5000    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.3437    0.3126    0.3437

v_opt = 0.40561

```

Figure 13. Symmetrization of the three pure strategies and their selection probabilities by $\alpha = 8$



```

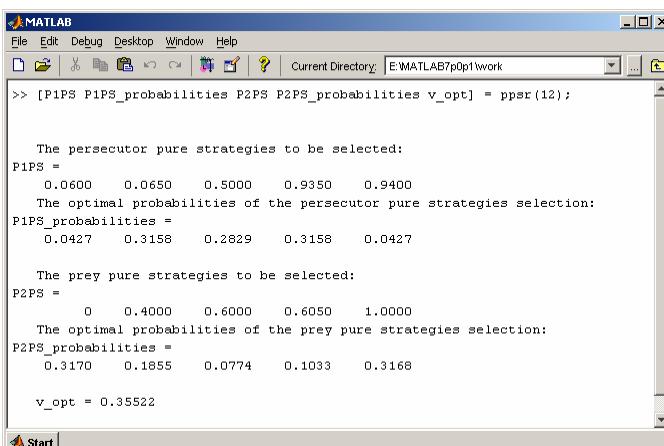
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(11);
The persecutor pure strategies to be selected:
P1PS =
    0.0550    0.0600    0.5000    0.9400    0.9450
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.2509    0.1080    0.2822    0.1080    0.2509

The prey pure strategies to be selected:
P2PS =
    0    0.4300    0.4350    0.5650    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.3312    0.0204    0.1477    0.1696    0.3312

v_opt = 0.36454

```

Figure 14. Appearance of the five pure strategies for their selection by $\alpha = 11$



```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(12);

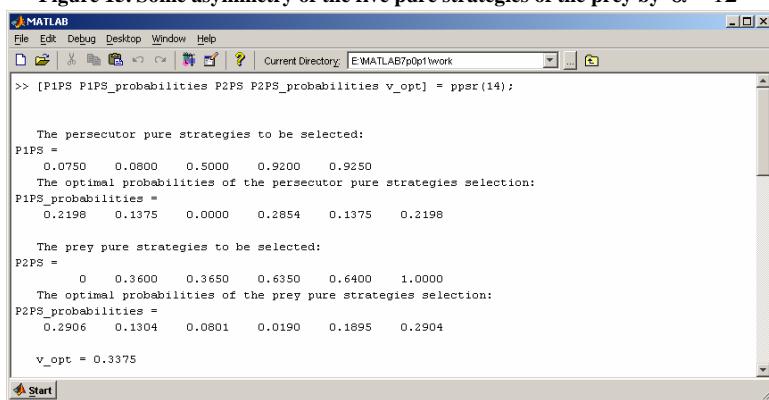
The persecutor pure strategies to be selected:
P1PS =
    0.0600    0.0650    0.5000    0.9350    0.9400
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.0427    0.3158    0.2829    0.3158    0.0427

The prey pure strategies to be selected:
P2PS =
    0    0.4000    0.6000    0.6050    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.3170    0.1855    0.0774    0.1033    0.3168

v_opt = 0.35522

```

Figure 15. Some asymmetry of the five pure strategies of the prey by $\alpha = 12$



```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(14);

The persecutor pure strategies to be selected:
P1PS =
    0.0750    0.0800    0.5000    0.9200    0.9250
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.2198    0.1375    0.0000    0.2854    0.1375    0.2198

The prey pure strategies to be selected:
P2PS =
    0    0.3600    0.3650    0.6350    0.6400    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2906    0.1304    0.0801    0.0190    0.1895    0.2904

v_opt = 0.33375

```

Figure 16. By $\alpha = 14$ the persecutor selects its five pure strategies (the zero probability must be expunged), and the prey selects its six pure strategies

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(16);

The persecutor pure strategies to be selected:
P1PS =
    0.0850    0.0900    0.5000    0.9100    0.9150
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.1911    0.1653    0.2871   -0.0000    0.1653    0.1911

The prey pure strategies to be selected:
P2PS =
    0    0.3450    0.3500    0.6500    0.6550    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2720    0.2200    0.0079    0.0108    0.2173    0.2720

v_opt = 0.32075

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(18);

The persecutor pure strategies to be selected:
P1PS =
    0.0700    0.3950    0.4000    0.6000    0.6050    0.9300
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.3224    0.1220    0.0556    0.0556    0.1220    0.3224

The prey pure strategies to be selected:
P2PS =
    0    0.3400    0.3450    0.6550    0.6600    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2671    0.0949    0.1380    0.1380    0.0949    0.2671

v_opt = 0.30595

>>

```

Figure 17. Pure strategies and their probabilities by $\alpha = 16$ and $\alpha = 18$

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(20);
The persecutor pure strategies to be selected:
P1PS =
    0.0500    0.0550    0.3650    0.6350    0.9450    0.9500
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.2877    0.0066    0.2057    0.2057    0.0066    0.2877

The prey pure strategies to be selected:
P2PS =
    0    0.3400    0.3450    0.6550    0.6600    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2631    0.2118    0.0251    0.0251    0.2118    0.2631

v_opt = 0.29424

```

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(28);
The persecutor pure strategies to be selected:
P1PS =
    0.0550    0.0600    0.3450    0.5000    0.6550    0.9400    0.9450
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.1637    0.1130    0.1933    0.0600    0.1933    0.1130    0.1637

The prey pure strategies to be selected:
P2PS =
    0    0.2650    0.2700    0.5000    0.7300    0.7350    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2243    0.0794    0.1083    0.1761    0.1083    0.0794    0.2243

v_opt = 0.25955

```

Figure 18. Symmetrization of the six pure strategies for their selection by $\alpha = 20$ Figure 19. Appearance of the seven pure strategies for their selection by $\alpha = 28$

It ought to be underlined, that the persecutor — prey systems with the optimal probability $v_{\text{opt}} < \frac{1}{2}$ of the prey

strike are out of practical interest. Then the case on the figure 12 is nearly the worst, where both the competitors should select the three pure strategies with the corresponding optimal probabilities.

The being determined within the module `ppsr` optimal probabilities may be easily practiced [11 — 15] by the prey or the persecutor, if it applies the earlier developed MATLAB 7.0.1 program module `opr2` (figure 20, [16, 17]). It is sufficient for the prey (or the persecutor) to type in the MATLAB 7.0.1 Command Window the line with the six arguments of this module, knowing the number G of the future shots on itself (or the future shots on the target), and run this line by pressing the enter key, as it is shown on the figure 21. The module `opr2` will compute the vector

$$\mathbf{V} = [v_1 \quad v_2 \quad \dots \quad v_G] \quad (26)$$

of the persecutor payoffs $\{v_j\}_{j=1}^G$, where v_j is the kernel (2) value in the shot j by $j = \overline{1, G}$, and actually their average is converging to v_{opt} as the number of the shots is tending to infinity:

$$\lim_{j \rightarrow \infty} \frac{1}{G} \sum_{j=1}^G v_j = \frac{1}{G} \lim_{j \rightarrow \infty} \sum_{j=1}^G v_j = v_{\text{opt}} \quad (27)$$

Besides, in the MATLAB 7.0.1 Command Window there are returned the hints of what pure strategy to be selected, and the relative deviation

$$\delta_v = \frac{\frac{1}{G} \sum_{j=1}^G v_j - v_{\text{opt}}}{v_{\text{opt}}} \quad (28)$$

of the averaged persecutor payoff.

```

1 function [Payoff_1] = opr2(PayoffMatrix, G, Soft_Correction, g_Display, g_pause, PureStrategy_Display)
2 if nargin==3
3     g_Display=0;
4     g_pause=0;
5     PureStrategy_Display=0;
6 end
7 if g_Display==0
8     g_pause=0;
9 end
10 [S1opt, S2opt, V1ow1, Vup1, OMS] = sp(PayoffMatrix);
11 Payoff_1=zeros(1,G);delta1=zeros(1,G);delta2=zeros(1,G);First_Player_Pure_Strategy_Number=zeros(1,G);Second_Player_Pure_Strategy_Number=zeros(1,G);
12 if OMS==0
13     disp(' This matrix game is solved in pure strategies.')
14     return
15 end
16 Vopt=S1opt*PayoffMatrix*S2opt'; format long
17
18 % g=1 the first player behavior
19 if g_Display==1
20     disp([' Now is the ' num2str(1) ' play'])
21     if g_pause==1
22         pause
23     end
24 end
25 beta1=1/(Vopt-V1ow1);
26 a1=1;
27 for g=1:G
28     delta1(g)=1/(g*beta1*g^(1/(a1)));
29 end
30 First_Player_Pure_Strategy_Number(1)=opr1p1(S1opt); % Initially selecting the pure strategy
31 V1(1)=sum(PayoffMatrix(First_Player_Pure_Strategy_Number(1), :).*S2opt);
32 while V1(1) < Vopt-delta1(1)
33     First_Player_Pure_Strategy_Number(1)=opr1p1(S1opt);
34     V1(1)=sum(PayoffMatrix(First_Player_Pure_Strategy_Number(1), :).*S2opt);
35 end
36 V0(1, :)=S2opt; % Definition for S2opt probabilities recalculations
37 if PureStrategy_Display==1
38     disp([' Now the first player has selected the pure strategy x' num2str(First_Player_Pure_Strategy_Number(1))])
39 end
40
41 % g=1 the second player behavior
42 beta2=1/(Vup1-Vopt);
43 a2=1;
44 for g=1:G
45
46 end

```

Figure 20. The starting part of the program module **opr2** code in MATLAB 7.0.1 M-file Editor

```

the kernel in the matrix form
option of the soft or hard correction of the
marginal losses, where 1 corresponds to
the soft correction

the number of shots (game replays)

option of whether the pure strategy
number, being selected in the current
shot, should be displayed

option of whether the pause is needed
after each shot, where 1 corresponds to
the pause setting on

>> [Payoff_1] = opr2(P, 8, 1, 1, 0, 1)

Now is the 1 play
Now the first player has selected the pure strategy x27
Now the second player has selected the pure strategy y51
Now is the 2 play
Now the first player real payoff is 0.85271
Now the first player has selected the pure strategy x27
Now the second player has selected the pure strategy y51
Now is the 3 play
Now the first player real payoff is 0.85271
Now the first player has selected the pure strategy x27
Now the second player has selected the pure strategy y51
Now is the 4 play
Now the first player real payoff is 0.85271
Now the first player has selected the pure strategy x27
Now the second player has selected the pure strategy y51
Now is the 5 play
Now the first player real payoff is 0.85271
Now the first player has selected the pure strategy x25
Now the second player has selected the pure strategy y51
Now is the 6 play
Now the first player real payoff is 0.82797
Now the first player has selected the pure strategy x27
Now the second player has selected the pure strategy y51
Now is the 7 play
Now the first player real payoff is 0.85271
Now the first player has selected the pure strategy x27
Now the second player has selected the pure strategy y51
Now is the 8 play
Now the first player real payoff is 0.85271
Now the first player has selected the pure strategy x27
Now the second player has selected the pure strategy y51
At last, the first player real payoff is 0.85271
The relative deviation of the averaged payoff of the first player is 0.0099041
Payoff_1 =
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.82796705280245 0.85271173552118 0.85271173552118 0.85271173552118
>>

```

Figure 21. The example of running the program module **opr2** code with $G = 8$ and some kernel (2) in the matrix form

Some examples of the module **opr2** application are shown below on the figures 22 — 34, and the mean probability of striking the prey dependence is plotted on the figures 35 — 36. Here it must be marked, that the program module **opr2** does not require the program module **ppsr** running before or later, as it works fully without assistance. But instead of putting there the kernel (2) in the matrix form as the payoff matrix, there may be put just the corresponding submatrix, which lines correspond to the optimal strategy spectrum of the persecutor, and columns correspond to the optimal strategy spectrum of the prey.

```

the kernel in the matrix form
option of the soft or hard correction of the
marginal losses, where 1 corresponds to
the soft correction

the number of shots (game replays)

option of whether the pure strategy
number, being selected in the current
shot, should be displayed

option of whether the pause is needed
after each shot, where 1 corresponds to
the pause setting on

>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.0072363
Payoff_1 =
0.82796705280245 0.85440909943717 0.85271173552118 0.82796705280245 0.85440909943717 0.85271173552118 0.85440909943717 0.85440909943717
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.011417
Payoff_1 =
0.85440909943717 0.82796705280245 0.85440909943717 0.85271173552118 0.85440909943717 0.85440909943717 0.85440909943717 0.85440909943717
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.0073449
Payoff_1 =
0.85440909943717 0.85271173552118 0.83002990648014 0.83002990648014 0.85271173552118 0.85440909943717 0.85271173552118 0.85271173552118
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 4.5992e-005
Payoff_1 =
0.82796705280245 0.85440909943717 0.83002990648014 0.83002990648014 0.85271173552118 0.83002990648014 0.85271173552118 0.85271173552118
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.0030176
Payoff_1 =
0.85440909943717 0.83002990648014 0.85271173552118 0.83002990648014 0.83002990648014 0.83002990648014 0.85271173552118 0.85271173552118

```

Figure 22. The vectors (26) by $G = 8$ and the same kernel (2) in the matrix form

```

MATLAB
File Edit Debug Desktop Window Help
Current Directory: E:\MATLAB\37\pdq\1\work
Payoff_1 =
Columns 1 through 8
0.82796705280245 0.82796705280245 0.82796705280245 0.82796705280245
Columns 9 through 10
0.85440909943717 0.85440909943717
>> [Payoff_1] = opr2(P, 10, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.00326

Payoff_1 =
Columns 1 through 8
0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 9 through 12
0.82796705280245 0.82796705280245 0.82796705280245
>> [Payoff_1] = opr2(P, 12, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.0078659

Payoff_1 =
Columns 1 through 8
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118
Columns 9 through 14
0.85440909943717 0.85440909943717 0.85440909943717 0.85440909943717
>> [Payoff_1] = opr2(P, 14, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.0017578

Payoff_1 =
Columns 1 through 8
0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 9 through 18
0.82796705280245 0.82796705280245 0.82796705280245 0.82796705280245
>> [Payoff_1] = opr2(P, 18, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.0035598

Payoff_1 =
Columns 1 through 8
0.85440909943717 0.85440909943717 0.85440909943717 0.85440909943717
Columns 9 through 16
0.82796705280245 0.82796705280245 0.82796705280245 0.82796705280245
Columns 17 through 18
0.85440909943717 0.85440909943717 0.85440909943717 0.85440909943717
>> [Payoff_1] = opr2(P, 20, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.0013397
Payoff_1 =

```

Figure 23. The vectors (26) by $G \in \{10, 12, 14, 18\}$ and the same kernel (2) in the matrix form

```

MATLAB File Edit Debug Desktop Window Help
File Edit Desktop Window Help
Current Directory E:\MATLAB\B7\p04\work
Payoff_1 =
Columns 1 through 8
0.85271173552118 0.85440909943717 0.85440909943717 0.82796705280245 0.83002990648014 0.82796705280245 0.85271173552118 0.85271173552118
Columns 9 through 16
0.82796705280245 0.83002990648014 0.82796705280245 0.85440909943717 0.85440909943717 0.85440909943717 0.83002990648014 0.85271173552118
Columns 17 through 20
0.83002990648014 0.85271173552118 0.85271173552118 0.82796705280245
>> [Payoff_1] = opr2(P, 25, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.0013397

Payoff_1 =
Columns 1 through 8
0.85440909943717 0.85271173552118 0.85440909943717 0.83002990648014 0.85271173552118 0.85271173552118 0.85271173552118 0.82796705280245
Columns 9 through 16
0.82796705280245 0.82796705280245 0.82796705280245 0.82796705280245 0.82796705280245 0.82796705280245 0.82796705280245 0.85440909943717
Columns 17 through 24
0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.85271173552118 0.83002990648014 0.83002990648014
Column 25
0.85271173552118
>> [Payoff_1] = opr2(P, 40, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.0027331

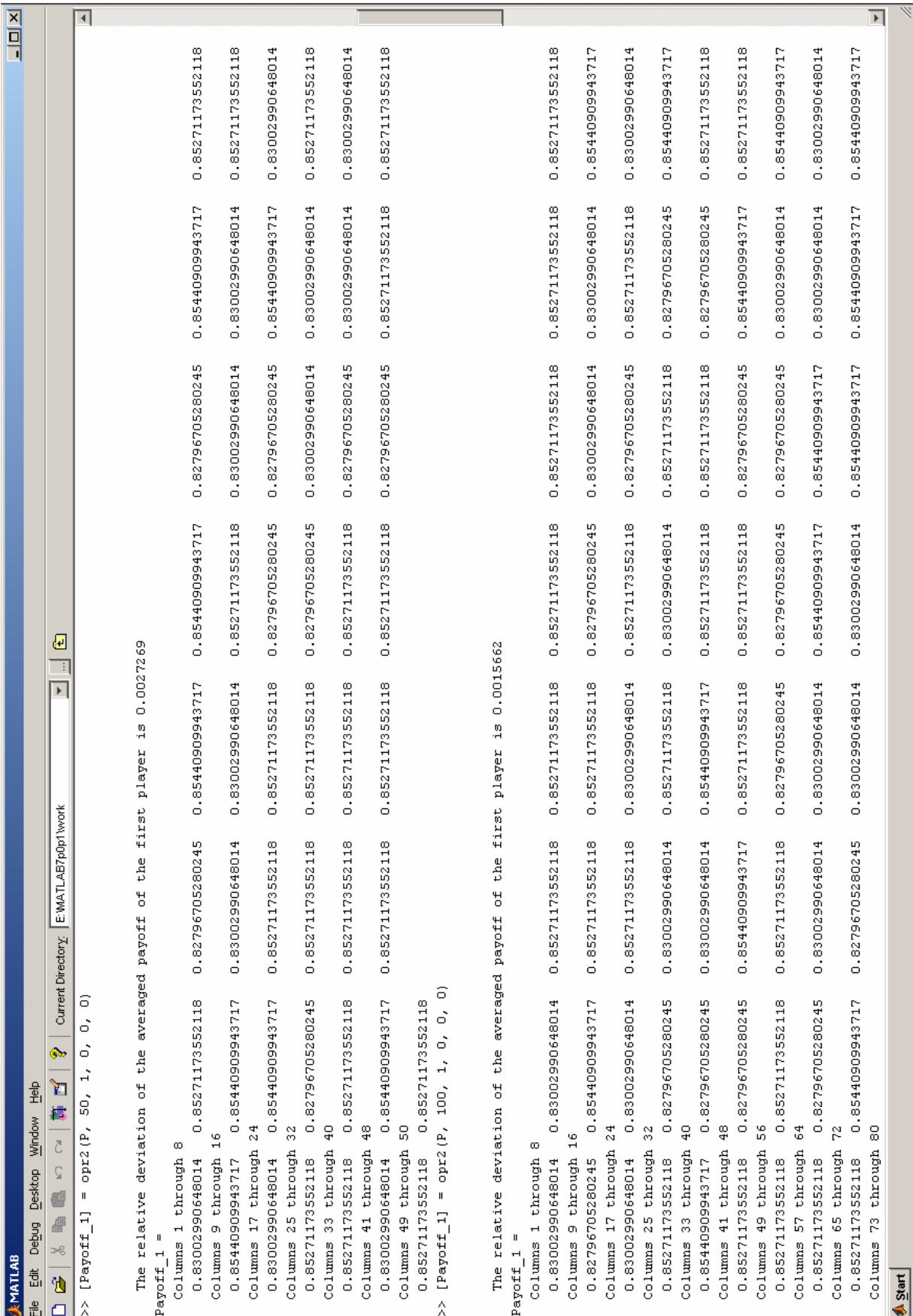
Payoff_1 =
Columns 1 through 8
0.82796705280245 0.82796705280245 0.82796705280245 0.85440909943717 0.83002990648014 0.85440909943717 0.85440909943717 0.83002990648014
Columns 9 through 16
0.85271173552118 0.83002990648014 0.83002990648014 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.83002990648014
Columns 17 through 24
0.82796705280245 0.85440909943717 0.85440909943717 0.82796705280245 0.85440909943717 0.83002990648014 0.85271173552118 0.83002990648014
Columns 25 through 32
0.83002990648014 0.83002990648014 0.83002990648014 0.85271173552118 0.83002990648014 0.85271173552118 0.82796705280245 0.83002990648014
Columns 33 through 40
0.83002990648014 0.83002990648014 0.85271173552118 0.83002990648014 0.83002990648014 0.85271173552118 0.85440909943717 0.82796705280245
>> [Payoff_1] = opr2(P, 50, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.0027269

Payoff_1 =
Columns 1 through 8
0.83002990648014 0.85271173552118 0.82796705280245 0.85440909943717 0.85440909943717 0.85440909943717 0.82796705280245 0.85271173552118

```

Figure 24. The vectors (26) by $G \in \{20, 25, 40\}$ and the same kernel (2) in the matrix form



The relative deviation of the averaged payoff of the first player is 0.0027269

```

Payoff_1 =
Columns 1 through 8
0.83002990648014 0.85271173552118 0.82796705280245 0.85440909943717 0.82796705280245 0.85440909943717 0.85271173552118
Columns 9 through 16
0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 17 through 24
0.83002990648014 0.85440909943717 0.85271173552118 0.85271173552118 0.82796705280245 0.82796705280245 0.85440909943717 0.83002990648014
Columns 25 through 32
0.85271173552118 0.82796705280245 0.85271173552118 0.85271173552118 0.82796705280245 0.83002990648014 0.85271173552118
Columns 33 through 40
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.82796705280245 0.82796705280245 0.83002990648014 0.85271173552118
Columns 41 through 48
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.82796705280245 0.83002990648014 0.83002990648014
Columns 49 through 56
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.82796705280245 0.83002990648014 0.83002990648014
>> [Payoff_1] = opr2(P, 50, 1, 0, 0)

```

The relative deviation of the averaged payoff of the first player is 0.0015662

```

Payoff_1 =
Columns 1 through 8
0.83002990648014 0.83002990648014 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118
Columns 9 through 16
0.82796705280245 0.85440909943717 0.85271173552118 0.85271173552118 0.82796705280245 0.83002990648014 0.83002990648014 0.85440909943717
Columns 17 through 24
0.83002990648014 0.83002990648014 0.85271173552118 0.83002990648014 0.85271173552118 0.82796705280245 0.85271173552118 0.83002990648014
Columns 25 through 32
0.85271173552118 0.82796705280245 0.83002990648014 0.83002990648014 0.83002990648014 0.85271173552118 0.82796705280245 0.85440909943717
Columns 33 through 40
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.82796705280245 0.85271173552118
Columns 41 through 48
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.82796705280245 0.83002990648014
Columns 49 through 56
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85440909943717 0.85440909943717
Columns 57 through 64
0.85271173552118 0.82796705280245 0.83002990648014 0.83002990648014 0.83002990648014 0.85440909943717 0.85440909943717 0.83002990648014
Columns 65 through 72
0.85271173552118 0.85440909943717 0.82796705280245 0.83002990648014 0.83002990648014 0.85440909943717 0.85440909943717 0.85440909943717
Columns 73 through 80
0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118 0.85440909943717 0.85440909943717

```

Figure 25. The vector (26) by $G = 50$ and the same kernel (2) in the matrix form, and the first portion of the vector (26) by $G = 100$

The relative deviation of the averaged payoff of the first player is -0.00025192

```

Payoff_1 =
Columns 1 through 8
0.85271173552118 0.85440909943717 0.82796705280245 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 9 through 16
0.82796705280245 0.85440909943717 0.83002990648014 0.85271173552118 0.82796705280245 0.82796705280245 0.82796705280245 0.82796705280245
Columns 17 through 24
0.85440909943717 0.85440909943717 0.83002990648014 0.85440909943717 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 25 through 32
0.85271173552118 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014
Columns 33 through 40
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 41 through 48
0.85271173552118 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014
Columns 49 through 56
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 57 through 64
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014
Columns 65 through 72
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 73 through 80
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014
Columns 81 through 88
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014
Columns 89 through 96
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014
Columns 97 through 104
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014
Columns 105 through 112
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014
Columns 113 through 120
0.83002990648014 0.83002990648014 0.85440909943717 0.82796705280245 0.85440909943717 0.85440909943717 0.83002990648014 0.83002990648014

```

Figure 26. The second portion of the vector (26) by $G = 100$ and the same kernel (2) in the matrix form, and the first portion of the vector (26) by $G = 150$

```

File Edit Desktop Window Help
□ | X | Current Directory E:\MATLAB7\p01\work
Columns 113 through 120
0.82796705280245 0.82796705280245 0.85440909943717 0.85440909943717 0.82796705280245 0.83271173552118 0.85271173552118 0.82796705280245
Columns 121 through 128
0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 129 through 136
0.85440909943717 0.85271173552118 0.83002990648014 0.83002990648014 0.85271173552118 0.82796705280245 0.85440909943717 0.82796705280245
Columns 137 through 144
0.82796705280245 0.85271173552118 0.85440909943717 0.85440909943717 0.85271173552118 0.85271173552118 0.85271173552118 0.83002990648014
Columns 145 through 150
0.83002990648014 0.85271173552118 0.83002990648014 0.83002990648014 0.82796705280245 0.82796705280245 0.82796705280245
>> [Payoff_1] = opr2(P, 200, 1, 0, 0, 0)

```

The relative deviation of the averaged payoff of the first player is -1.3341e-005

```

Payoff_1 =
Columns 1 through 8
0.85271173552118 0.85271173552118 0.85440909943717 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014 0.83002990648014
Columns 9 through 16
0.82796705280245 0.82796705280245 0.82796705280245 0.85440909943717 0.83002990648014 0.82796705280245 0.85440909943717 0.85440909943717
Columns 17 through 24
0.85271173552118 0.85440909943717 0.85271173552118 0.83002990648014 0.83002990648014 0.85271173552118 0.85440909943717 0.85271173552118
Columns 25 through 32
0.82796705280245 0.82796705280245 0.85271173552118 0.85271173552118 0.82796705280245 0.83002990648014 0.85440909943717 0.82796705280245
Columns 33 through 40
0.82796705280245 0.82796705280245 0.85440909943717 0.85440909943717 0.85271173552118 0.83002990648014 0.85271173552118 0.85440909943717
Columns 41 through 48
0.85271173552118 0.82796705280245 0.85440909943717 0.83002990648014 0.85440909943717 0.85271173552118 0.85440909943717 0.85271173552118
Columns 49 through 56
0.85271173552118 0.85271173552118 0.82796705280245 0.82796705280245 0.85271173552118 0.85271173552118 0.85271173552118 0.85271173552118
Columns 57 through 64
0.85440909943717 0.82796705280245 0.82796705280245 0.83002990648014 0.85440909943717 0.85440909943717 0.85271173552118 0.85271173552118
Columns 65 through 72
0.85271173552118 0.85271173552118 0.85271173552118 0.82796705280245 0.82796705280245 0.85271173552118 0.85271173552118 0.85271173552118
Columns 73 through 80
0.85440909943717 0.82796705280245 0.85440909943717 0.83002990648014 0.85440909943717 0.85440909943717 0.85271173552118 0.85271173552118
Columns 81 through 88
0.85271173552118 0.82796705280245 0.83002990648014 0.85440909943717 0.85440909943717 0.85440909943717 0.85271173552118 0.85271173552118
Columns 89 through 96
0.85271173552118 0.82796705280245 0.83002990648014 0.85271173552118 0.82796705280245 0.85271173552118 0.85271173552118 0.85271173552118
Columns 97 through 104
0.85440909943717 0.82796705280245 0.83002990648014 0.85271173552118 0.85271173552118 0.85271173552118 0.83002990648014 0.85271173552118
Columns 105 through 112
0.82796705280245 0.83002990648014 0.85271173552118 0.82796705280245 0.82796705280245 0.85271173552118 0.85271173552118 0.82796705280245
Columns 113 through 120
0.85440909943717 0.82796705280245 0.83002990648014 0.83002990648014 0.82796705280245 0.83002990648014 0.83002990648014 0.83002990648014

```

Figure 27. The second portion of the vector (26) by $G = 150$ and the same kernel (2) in the matrix form, and the first portion of the vector (26) by $G = 200$

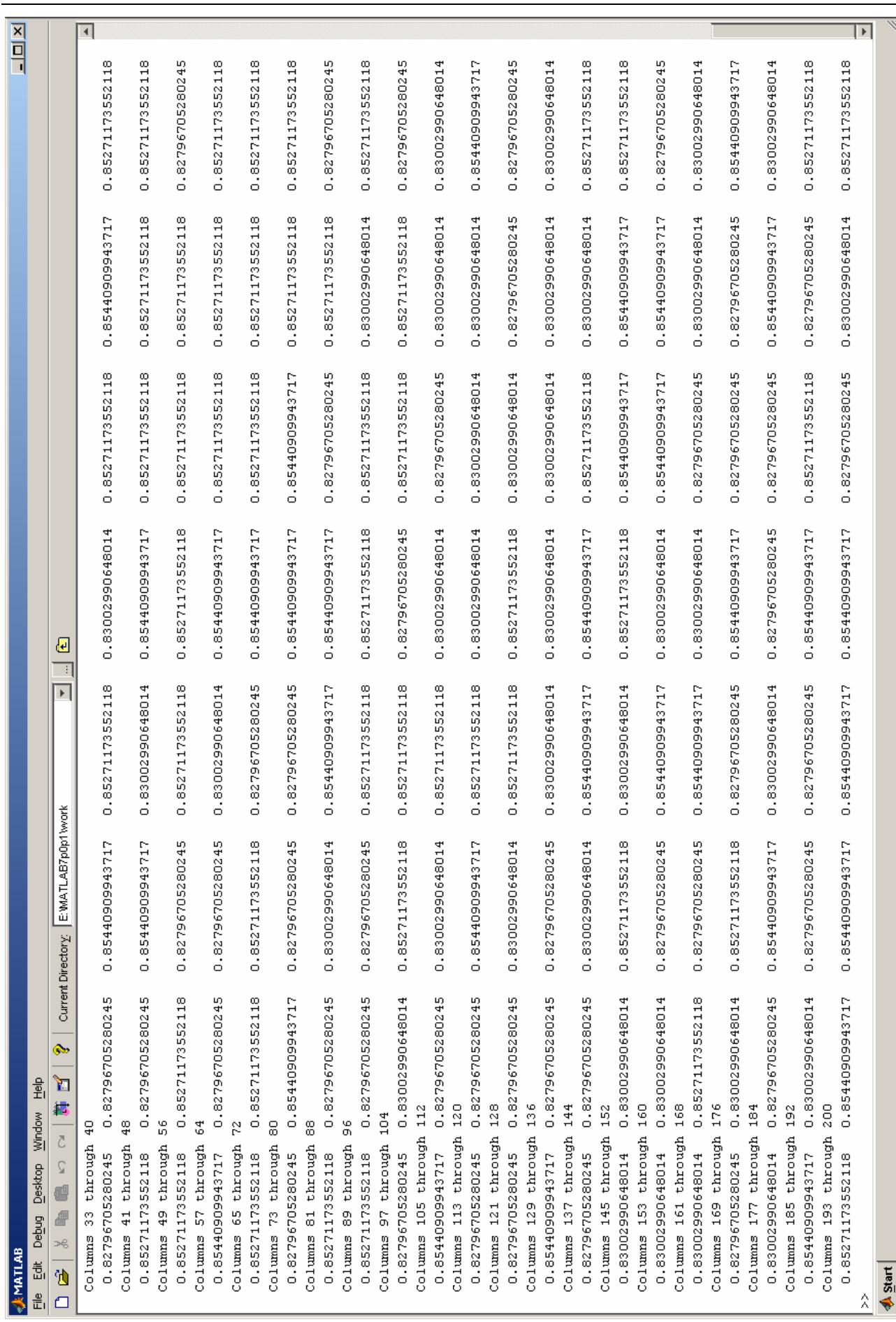


Figure 28. The second portion of the vector (26) by $G = 200$ and the same kernel (2) in the matrix form

The screenshot shows a MATLAB command window with the following text output:

```

Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.99639
Now the second player has selected the pure strategy x3
Now the first player real payoff is 0.99639
Now the first player has selected the pure strategy y1
Now the second player has selected the pure strategy x46
Now the first player real payoff is 0.99639
Now the first player has selected the pure strategy y1
Now the second player has selected the pure strategy x46
Now the first player real payoff is 0.058463
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.99639
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.99639
Now the second player has selected the pure strategy x3
Now the first player real payoff is 0.058463
Now the first player has selected the pure strategy y1
Now the second player has selected the pure strategy x3
Now the first player real payoff is 0.99639
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.99639
Now the second player has selected the pure strategy x3
Now the first player real payoff is 0.058463
Now the first player has selected the pure strategy y1
Now the second player has selected the pure strategy x3
Now the first player real payoff is 0.99639
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.052472
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.99639
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.99639
At last, the first player real payoff is 0.99639
The relative deviation of the averaged payoff of the first player is 0.4329
Payoff_1 =
Columns 1 through 8
0.9963883255360 0.9963883255360 0.0584626538925 0.9963883255360 0.0584626538925 0.9963883255360 0.9963883255360 0.9963883255360
Columns 9 through 11
0.05247227457122 0.9963883255360 0.9963883255360
>>

```

Figure 29. The hints of what pure strategy to be selected and the vector (26) by $G = 11$ and another kernel (2) in the matrix form

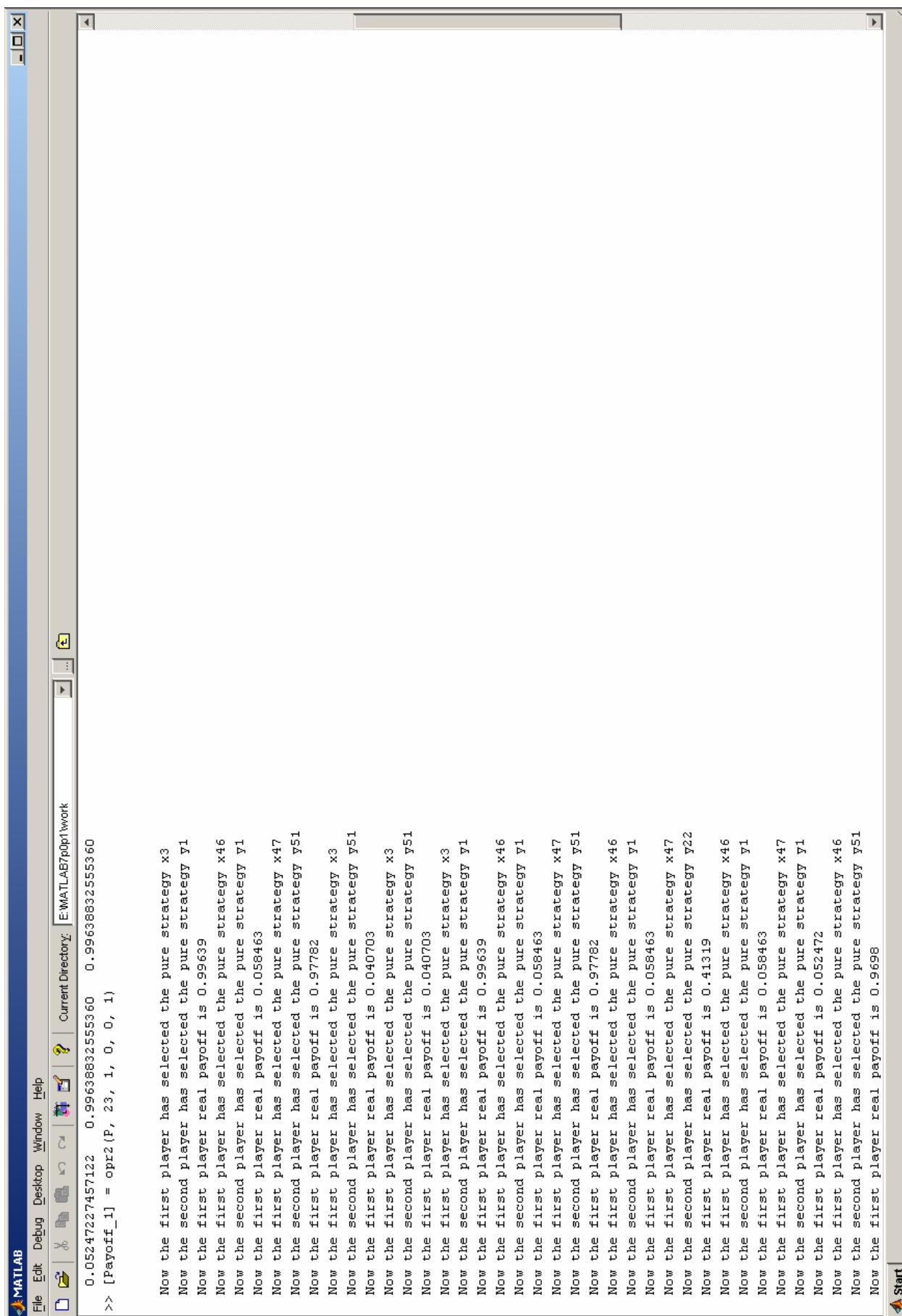


Figure 30. The first portion of the hints of what pure strategy to be selected and the vector (26) by $G = 23$ and another kernel (2) in the matrix form

The screenshot shows a MATLAB interface with the following content:

```

Now the first player real payoff is 0.052472
Now the first player has selected the pure strategy x46
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.9698
Now the first player has selected the pure strategy x47
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.97782
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.99639
Now the first player has selected the pure strategy x46
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.9698
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.040703
Now the first player has selected the pure strategy x46
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.9698
Now the first player has selected the pure strategy x47
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.97782
Now the first player has selected the pure strategy x46
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.9698
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.040703
Now the first player has selected the pure strategy x47
Now the second player has selected the pure strategy y1
At last, the first player real payoff is 0.052472
The relative deviation of the averaged payoff of the first player is 0.066129
Payoff_1 =
Columns 1 through 8
0.9963883555360 0.05846265388925 0.97781961741164 0.04070252637372 0.04070252637372 0.9963883255360 0.05846265388925 0.97781961741164
Columns 9 through 16
0.05846265388925 0.4131891755010 0.05846265388925 0.05247227457122 0.96979810649932 0.97781961741164 0.9963883255360 0.96979810649932
Columns 17 through 23
0.04070252637372 0.96979810649932 0.97781961741164 0.96979810649932 0.96979810649932 0.04070252637372 0.05247227457122
>>

```

Figure 31. The second portion of the hints of what pure strategy to be selected and the vector (26) by $G = 23$ and another kernel (2) in the matrix form

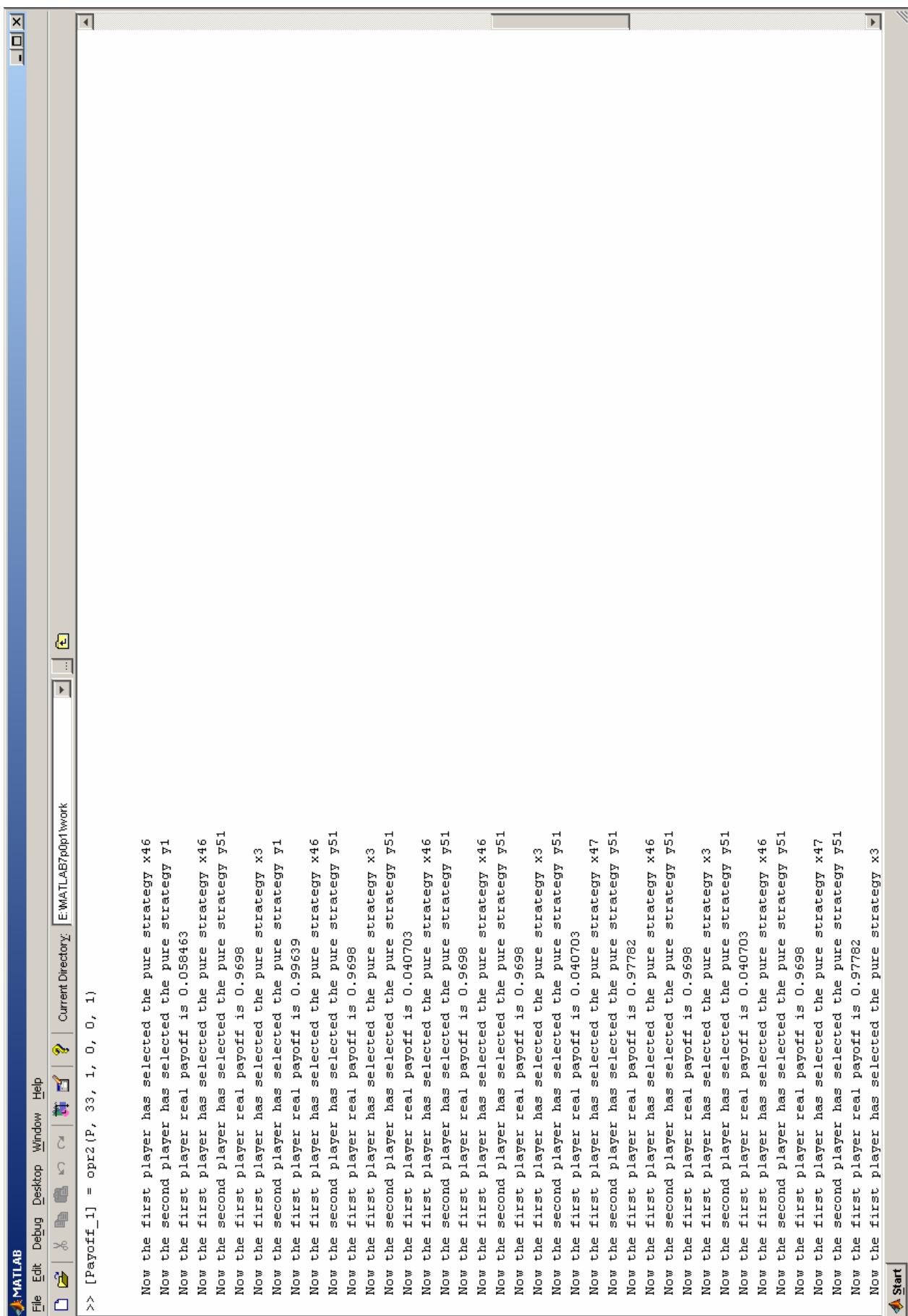


Figure 32. The first portion of the hints of what pure strategy to be selected and the vector (26) by $G = 33$ and another kernel (2) in the matrix form

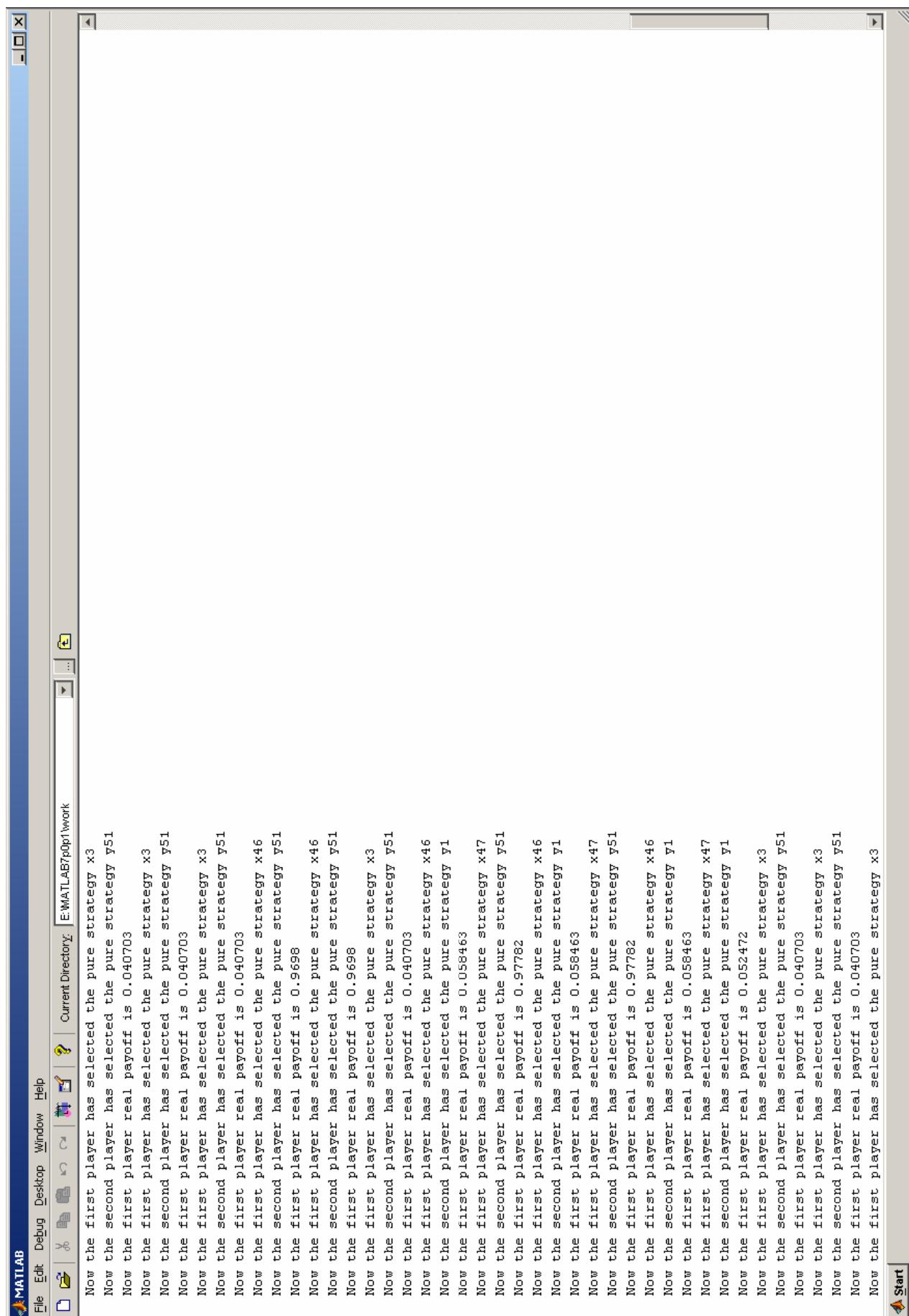


Figure 33. The second portion of the hints of what pure strategy to be selected and the vector (26) by $G = 33$ and another kernel (2) in the matrix form

The screenshot shows a MATLAB interface with the following details:

- MenuBar:** File, Edit, Debug, Desktop, Window, Help.
- ToolBar:** Standard MATLAB icons for file operations, zoom, and help.
- Current Directory:** E:\MATLAB7\p01\work
- Command Window:**

```

Now the first player has selected the pure strategy x46
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.058463
Now the first player has selected the pure strategy x47
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.052472
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.040703
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.040703
Now the first player has selected the pure strategy y1
Now the second player has selected the pure strategy x3
Now the first player has selected the pure strategy y51
Now the second player has selected the pure strategy x3
Now the first player real payoff is 0.99639
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.040703
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y51
Now the first player real payoff is 0.040703
Now the second player has selected the pure strategy x3
Now the first player real payoff is 0.99639
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
Now the first player real payoff is 0.99639
Now the first player has selected the pure strategy x3
Now the second player has selected the pure strategy y1
At last, the first player real payoff is 0.99639
The relative deviation of the averaged payoff of the first player is 0.020084

```
- Matrix Display:**

Payoff_1 =
Columns 1 through 8
0.99638832553360 0.96979810649932 0.96979810649932 0.96979810649932 0.96979810649932 0.96979810649932 0.96979810649932 0.96979810649932
0.05846265388925 0.96979810649932 0.96979810649932 0.96979810649932 0.96979810649932 0.96979810649932 0.96979810649932 0.96979810649932
Columns 9 through 16
0.97781961741164 0.96979810649932 0.04070252637372 0.96979810649932 0.97781961741164 0.04070252637372 0.04070252637372 0.04070252637372
Columns 17 through 24
0.96979810649932 0.96979810649932 0.04070252637372 0.05846265388925 0.97781961741164 0.05846265388925 0.97781961741164 0.05846265388925
Columns 25 through 32
0.05247227457122 0.04070252637372 0.04070252637372 0.99638832553360 0.04070252637372 0.04070252637372 0.99638832553360 0.99638832553360
Column 33
0.99638832553360
- Command Line:** >> Start

Figure 34. The third portion of the hints of what pure strategy to be selected and the vector (26) by $G = 33$ and another kernel (2) in the matrix form

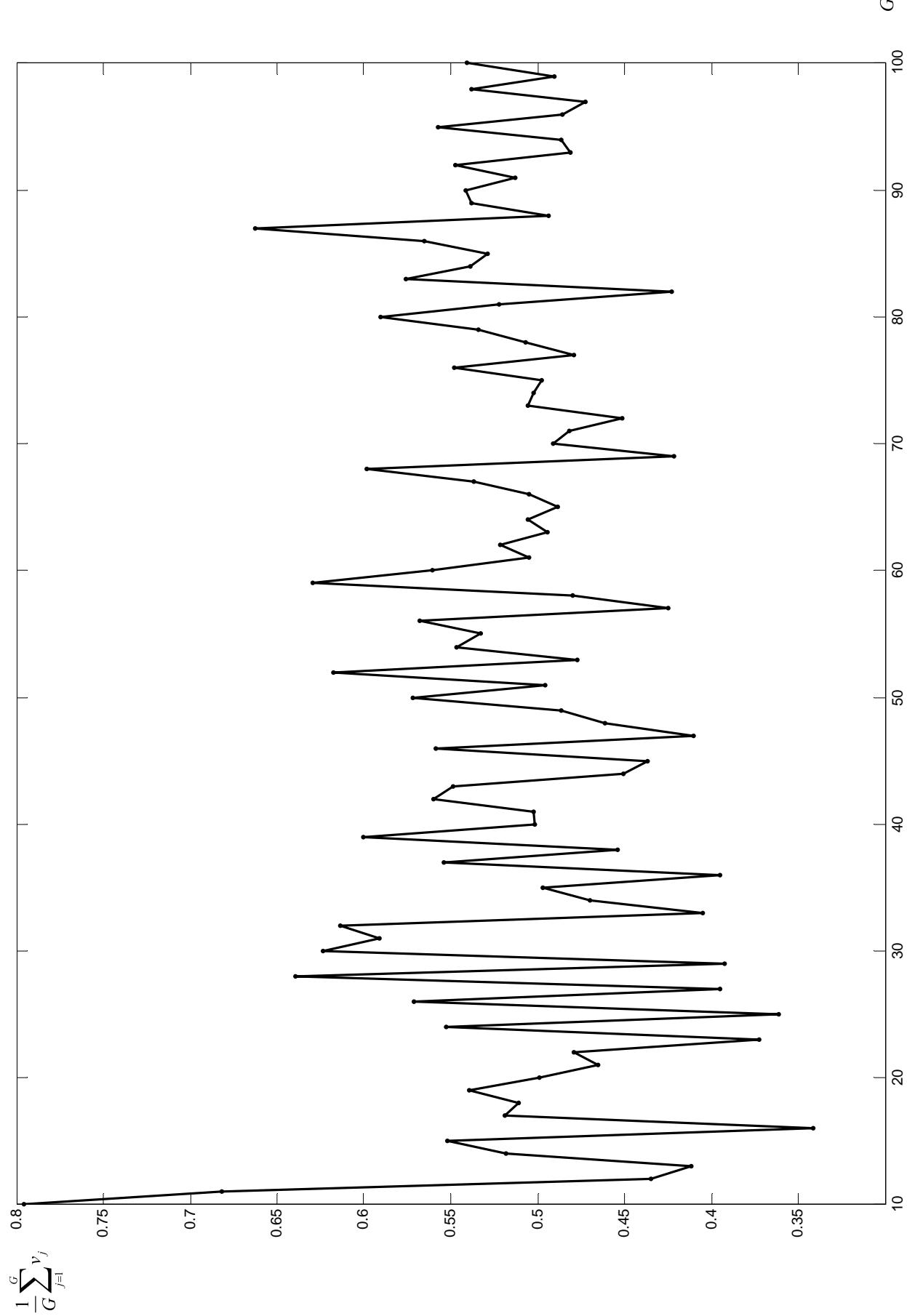


Figure 35. The averaged probability of striking the prey dependence from the number of shots $G \in \{10, 100\}$

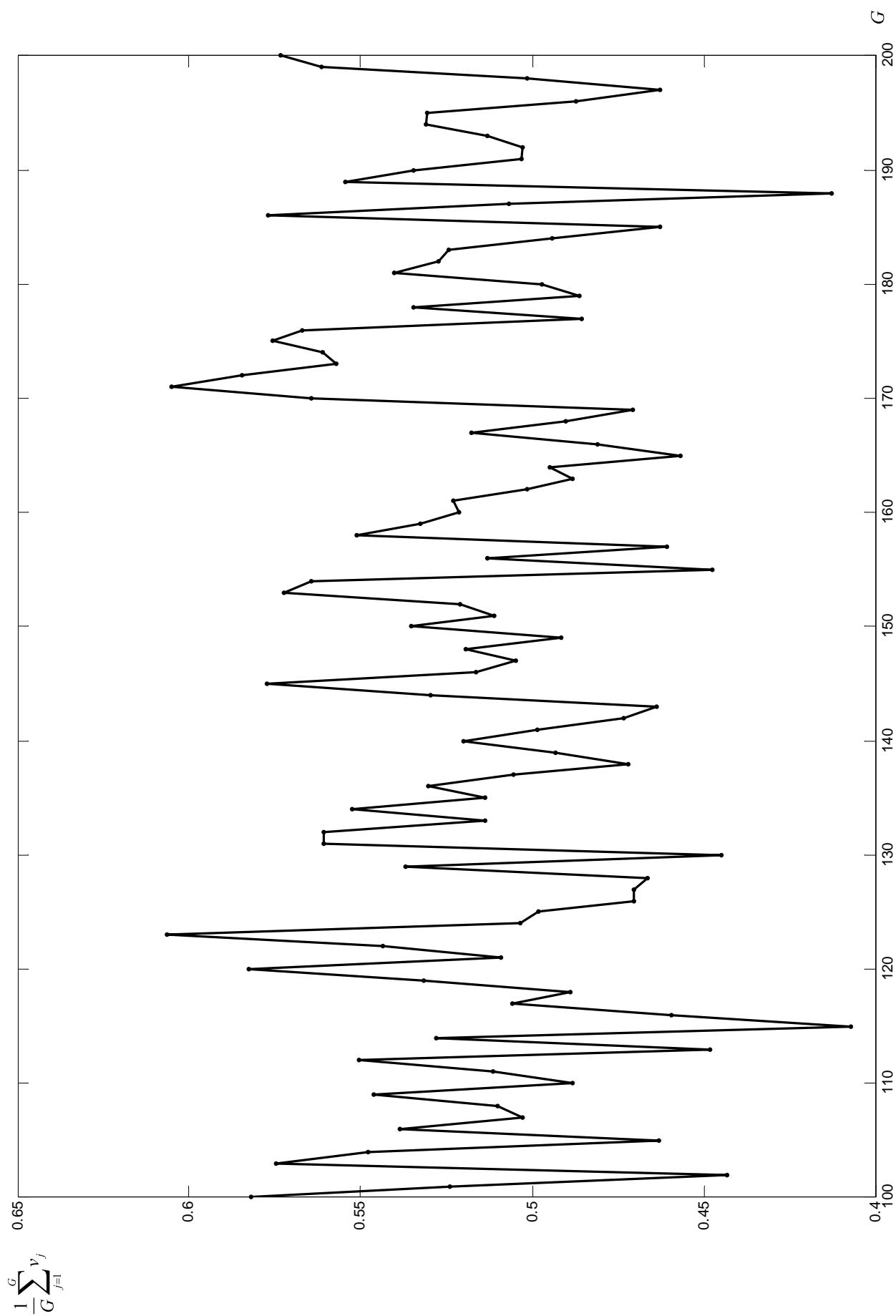


Figure 36. The averaged probability of striking the prey dependence from the number of shots $G \in \{\overline{100, 200}\}$

The rugged exponential probability kernel

Again check the real kernel (2) for the game convexity and concavity. The first derivative of the function (2) by the variable x is

$$\frac{\partial P(x, y)}{\partial x} = \frac{\partial}{\partial x} \left(\exp[-\alpha(x-y)^2] + n(x, y) \right) = \frac{\partial}{\partial x} n(x, y) - 2\alpha(x-y) \exp[-\alpha(x-y)^2] \quad (29)$$

and its second derivative is

$$\begin{aligned} \frac{\partial^2 P(x, y)}{\partial x^2} &= \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} n(x, y) - 2\alpha(x-y) \exp[-\alpha(x-y)^2] \right) = \\ &= \frac{\partial^2 n(x, y)}{\partial x^2} - 2\alpha \exp[-\alpha(x-y)^2] + 4\alpha^2(x-y)^2 \exp[-\alpha(x-y)^2] = \\ &= 2\alpha(2\alpha(x-y)^2 - 1) \exp[-\alpha(x-y)^2] + \frac{\partial^2 n(x, y)}{\partial x^2}. \end{aligned} \quad (30)$$

The first derivative of the function (2) by the variable y is

$$\frac{\partial P(x, y)}{\partial y} = \frac{\partial}{\partial y} \left(\exp[-\alpha(x-y)^2] + n(x, y) \right) = 2\alpha(x-y) \exp[-\alpha(x-y)^2] + \frac{\partial}{\partial y} n(x, y) \quad (31)$$

and its second derivative is

$$\begin{aligned} \frac{\partial^2 P(x, y)}{\partial y^2} &= \frac{\partial}{\partial y} \left(2\alpha(x-y) \exp[-\alpha(x-y)^2] + \frac{\partial}{\partial y} n(x, y) \right) = \\ &= \frac{\partial^2 n(x, y)}{\partial y^2} - 2\alpha \exp[-\alpha(x-y)^2] + 4\alpha^2(x-y)^2 \exp[-\alpha(x-y)^2] = \\ &= 2\alpha(2\alpha(x-y)^2 - 1) \exp[-\alpha(x-y)^2] + \frac{\partial^2 n(x, y)}{\partial y^2}. \end{aligned} \quad (32)$$

Again, the concavity condition $\frac{\partial^2 P(x, y)}{\partial x^2} \leq 0$ must be true $\forall x \in [0; 1]$ and $\forall y \in [0; 1]$, and, reversely, the convexity condition $\frac{\partial^2 P(x, y)}{\partial y^2} \geq 0$ must be true $\forall x \in [0; 1]$ and $\forall y \in [0; 1]$. The concavity condition

$$2\alpha(2\alpha(x-y)^2 - 1) \exp[-\alpha(x-y)^2] + \frac{\partial^2 n(x, y)}{\partial x^2} \leq 0 \quad (33)$$

and the convexity condition

$$2\alpha(2\alpha(x-y)^2 - 1) \exp[-\alpha(x-y)^2] + \frac{\partial^2 n(x, y)}{\partial y^2} \geq 0 \quad (34)$$

may be verified numerically with as high precision as needed. The concavity of the being explored game will point at that the persecutor has the pure strategy, and the convexity will point at the prey having the pure strategy. Further, within here, it will be programmed within the updated MATLAB 7.0.1 program module `ppsr`. This updated module should accept minimally the two inputs, first of which is the parameter α . The second input is the roughness surface $n(x, y)$, being added to the exponential probability (1). Naturally, that by $n(x, y) = 0$ the updated program module, that may be called `ppsr2`, must function as the prime original module `ppsr`. The screenshot of the module `ppsr2` code is on the figure 37. Clearly, the precision of the convexity and concavity verification depends on the number of the input points of the roughness surface $n(x, y)$.

The examples of the program module `ppsr2` application require to have the surface $n(x, y)$. To produce this surface artificially there has been coded a script line (figure 38), which returning had been used by plotting the figures 1 — 4. With the artificial roughness surface $n(x, y)$ there are the 13 live results (figures 39 — 51) of having applied the

program module `ppsr2` by different α and amplitude of the surface $n(x, y)$ roughness.

```

1 function [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(alpha, noise_rough)
2 if alpha <= 0
3     error(' The input parameter alpha must be greater than 0.')
4 end
5 kx = 0;
6 N = 200;
7 for x = 0:1/N:1
8     kx = kx + 1;
9     ky = 0;
10    for y = 0:1/N:1
11        ky = ky + 1;
12        P(kx, ky) = exp(-alpha*(x-y)^2) + noise_rough(kx, ky);
13        if P(kx, ky) > 1
14            P(kx, ky) = 1;
15        end
16    end
17 end
18 if sum(sum(diff(P, 2, 1) <= 0)) == size(diff(P, 2, 1), 1)*size(diff(P, 2, 1), 2)
19 disp(' This is the concave game, and here the persecutor should have the single optimal pure strategy.')
20 disp(' This pure strategy, in the displayed below strategies, is selected with the greatest probability.')
21 disp(' and the rest strategies, even with nonzero probabilities, may be neglected.')
22 end
23 if sum(sum(diff(P, 2, 2) >= 0)) == size(diff(P, 2, 2), 1)*size(diff(P, 2, 2), 2)
24 disp(' This is the convex game, and here the prey should have the single optimal pure strategy.')
25 disp(' This pure strategy, in the displayed below strategies, is selected with the greatest probability.')
26 disp(' and the rest strategies, even with nonzero probabilities, may be neglected.')
27 end
28 [X_opt, Y_opt, Vlow, Vup, OMS, v_opt] = sp(P);
29 format short
30 P1PSnumbers = find(abs(X_opt) > 1e-10);
31 P2PSnumbers = find(abs(Y_opt) > 1e-10);
32 disp(' The persecutor pure strategies to be selected:')
33 P1PS = (P1PSnumbers-1)/N
34 disp(' The optimal probabilities of the persecutor pure strategies selection:')
35 P1PS_probabilities = X_opt(find(abs(X_opt) > 1e-20));
36 if sum(P1PS_probabilities > 1-1e-10) > 0
37     P1PS_probabilities = 1;
38     disp(' The single optimal pure strategy of the persecutor must be selected in every play.')
39 else
40     P1PS_probabilities = X_opt(find(abs(X_opt) > 1e-20));
41 end
42 disp(' ')
43 disp(' The prey pure strategies to be selected:')
44 P2PS = (P2PSnumbers-1)/N
45 disp(' The optimal probabilities of the prey pure strategies selection:')
46 P2PS_probabilities = Y_opt(find(abs(Y_opt) > 1e-20));
47 disp(' ')
48 disp([' v_opt = ' num2str(v_opt)])
49 disp(' ')

```

Figure 37. The screenshot of the module `ppsr2` full code

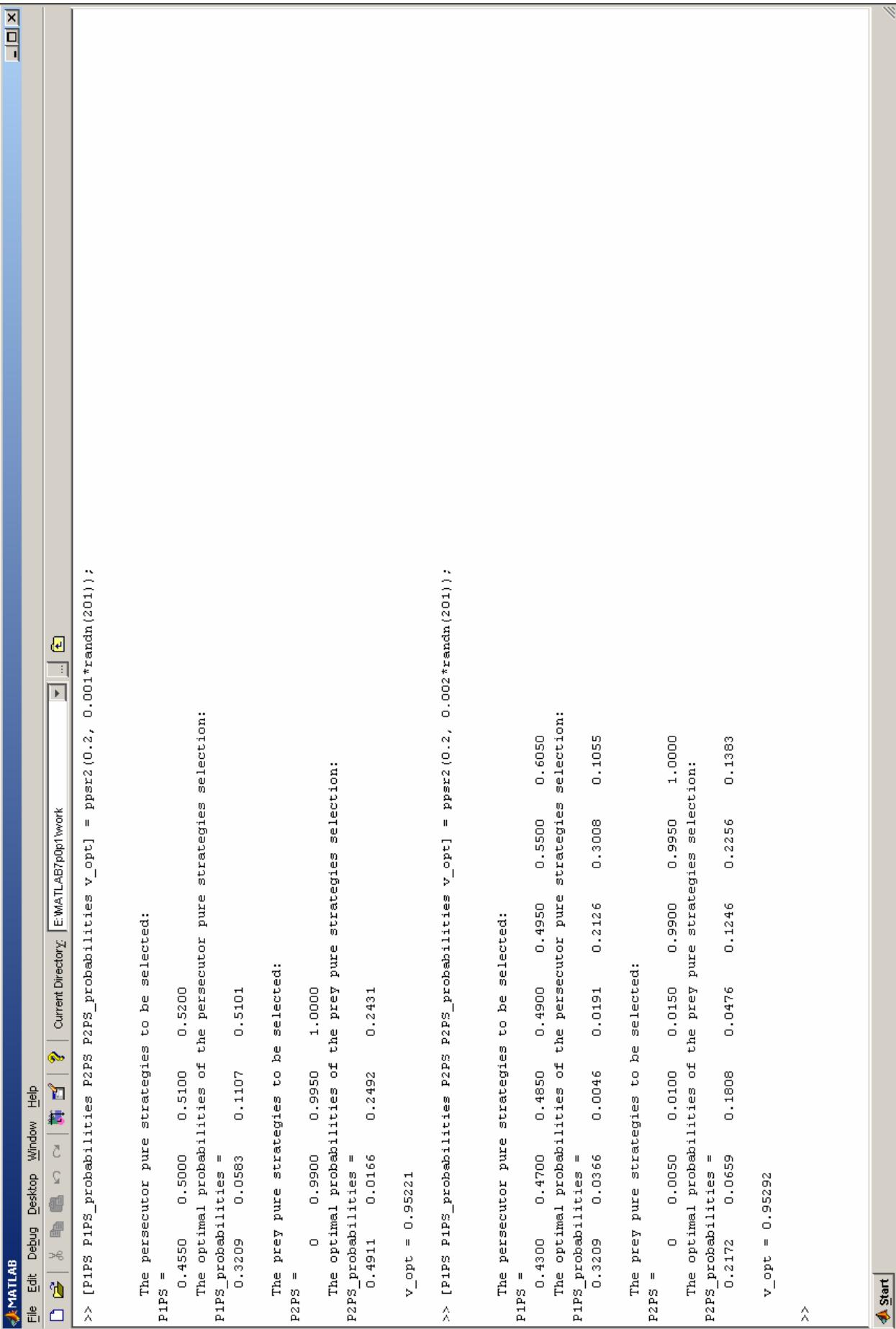
```

10 - kx = 0;
11 - N = 50;
12 - for x = 0:1/N:1
13 -     kx = kx + 1;
14 -     ky = 0;
15 -     for y = 0:1/N:1
16 -         ky = ky + 1;
17 -         P(kx, ky) = exp(-alpha*(x-y)^2)+0.002*randn;
18 -         if P(kx, ky) > 1
19 -             P(kx, ky) = 1;
20 -         end
21 -     end
22 - end

```

Figure 38. The fragment of the script code, which returns the sampled surface $n(x, y)$, what had been used by plotting the figures 1 — 4

With the number of the future shots G , for practicing the been determined optimal probabilities, the prey or the persecutor should apply the earlier mentioned MATLAB 7.0.1 program module `opr2` by typing in the MATLAB 7.0.1 Command Window the line with the number G and running this line by the pressed enter key. The module `opr2` will compute the vector (26) and the value (28), where actually (27) stays. All the hints, obviously, appear also, if they are checked with the corresponding option.



```

MATLAB
File Edit Debug Desktop Window Help
Current Directory E:\MATLAB7p0\p1\work
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.2, 0.001*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4550    0.5000    0.5100    0.5200
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.3209    0.0583    0.1107    0.5101

The prey pure strategies to be selected:
P2PS =
    0    0.9900    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.4911    0.0166    0.2492    0.2431
v_opt = 0.95221

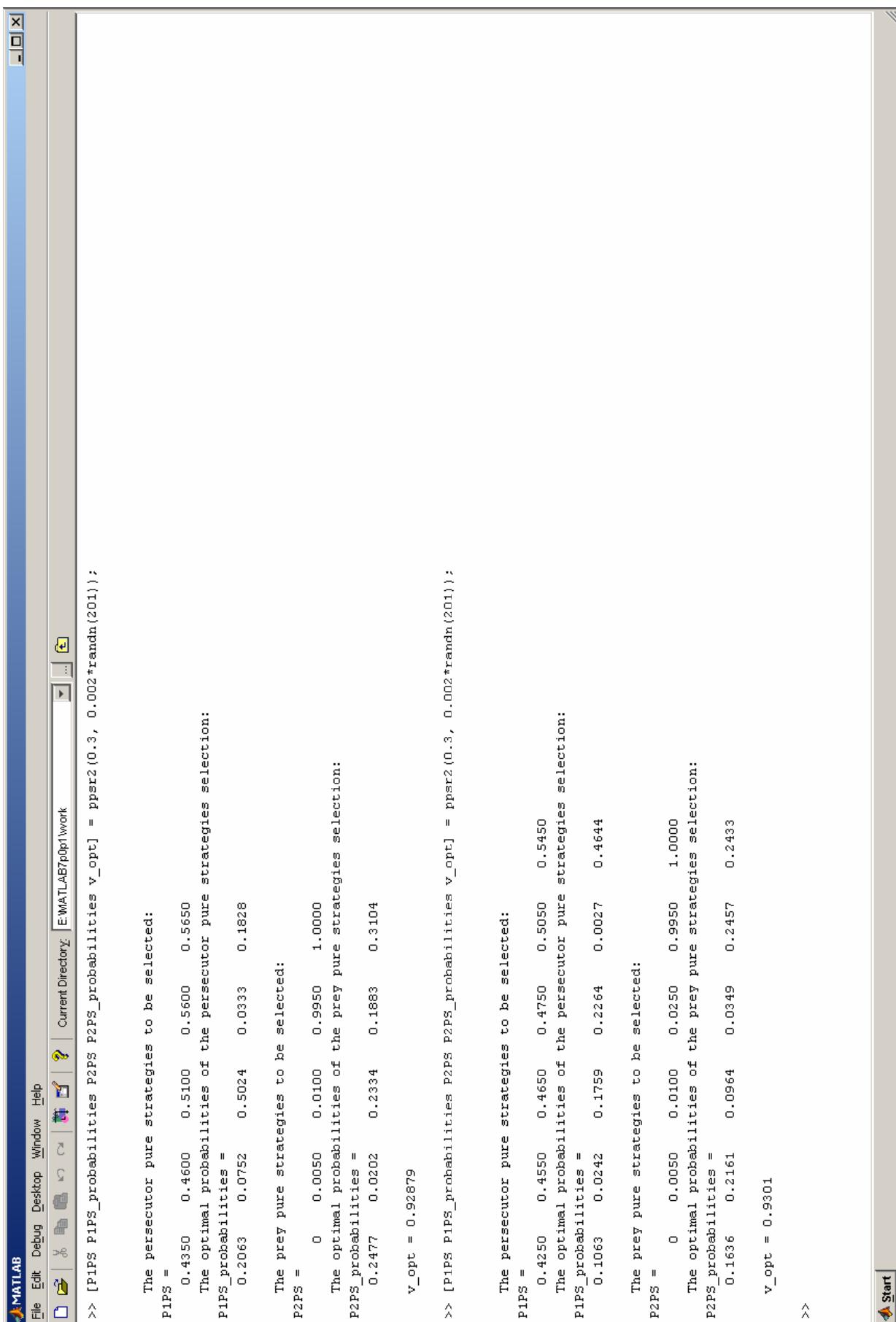
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.2, 0.002*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4300    0.4700    0.4850    0.4900    0.4950    0.5500    0.6050
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.3209    0.0366    0.0046    0.0191    0.2126    0.3008    0.1055

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.0150    0.0900    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2172    0.0659    0.1808    0.0476    0.1246    0.2256    0.1383
v_opt = 0.95292
>>

```

Figure 39. Program module ppsr2 application with $\alpha = 0.2$ and some low noise amplitudes



The screenshot shows a MATLAB application window with the following content:

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.3, 0.002*randn(201));

```

The persecutor pure strategies to be selected:

P1PS	0.4350	0.4600	0.5100	0.5600	0.5650
------	--------	--------	--------	--------	--------

The optimal probabilities of the persecutor pure strategies selection:

P1PS_probabilities	0.2063	0.0752	0.5024	0.0333	0.1828
--------------------	--------	--------	--------	--------	--------

The prey pure strategies to be selected:

P2PS	0	0.0050	0.0100	0.9950	1.0000
------	---	--------	--------	--------	--------

The optimal probabilities of the prey pure strategies selection:

P2PS_probabilities	0.2477	0.0202	0.2334	0.1883	0.3104
--------------------	--------	--------	--------	--------	--------

$$v_{opt} = 0.92879$$

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.3, 0.002*randn(201));

```

The persecutor pure strategies to be selected:

P1PS	0.4250	0.4550	0.4650	0.4750	0.5050	0.5450
------	--------	--------	--------	--------	--------	--------

The optimal probabilities of the persecutor pure strategies selection:

P1PS_probabilities	0.1063	0.0242	0.1759	0.2264	0.0027	0.4644
--------------------	--------	--------	--------	--------	--------	--------

The prey pure strategies to be selected:

P2PS	0	0.0050	0.0100	0.0250	0.9950	1.0000
------	---	--------	--------	--------	--------	--------

The optimal probabilities of the prey pure strategies selection:

P2PS_probabilities	0.1636	0.2161	0.0964	0.0349	0.2457	0.2433
--------------------	--------	--------	--------	--------	--------	--------

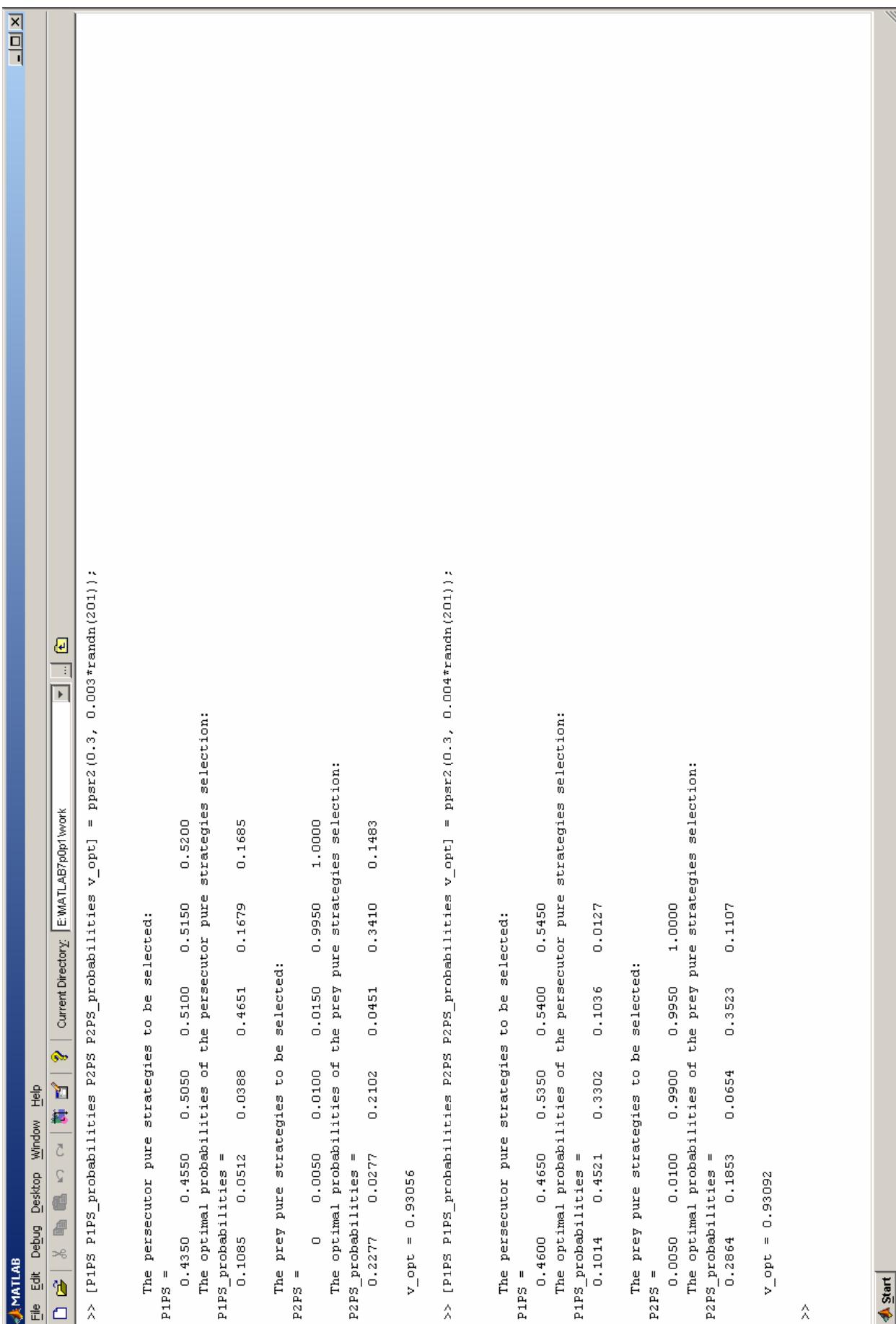
$$v_{opt} = 0.9301$$

```

>>

```

Figure 40. Program module `ppsr2` application with $\alpha = 0.3$ and the twice greater noise amplitude



The screenshot shows the MATLAB graphical user interface. The title bar of the active window says "Applied Math". The menu bar includes File, Edit, Debug, Desktop, Window, Help. The toolbar has icons for file operations like Open, Save, and Print. The current directory is set to "E:\MATLAB7\ppsr1\work". The code window displays a MATLAB script with comments and output text.

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.3, 0.003*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4350    0.4550    0.5050    0.5100    0.5150    0.5200
    The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.1085    0.0512    0.0388    0.4651    0.1679    0.1685

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.0150    0.9950    1.0000
    The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2277    0.0277    0.2102    0.0451    0.3410    0.1483
v_opt = 0.93056

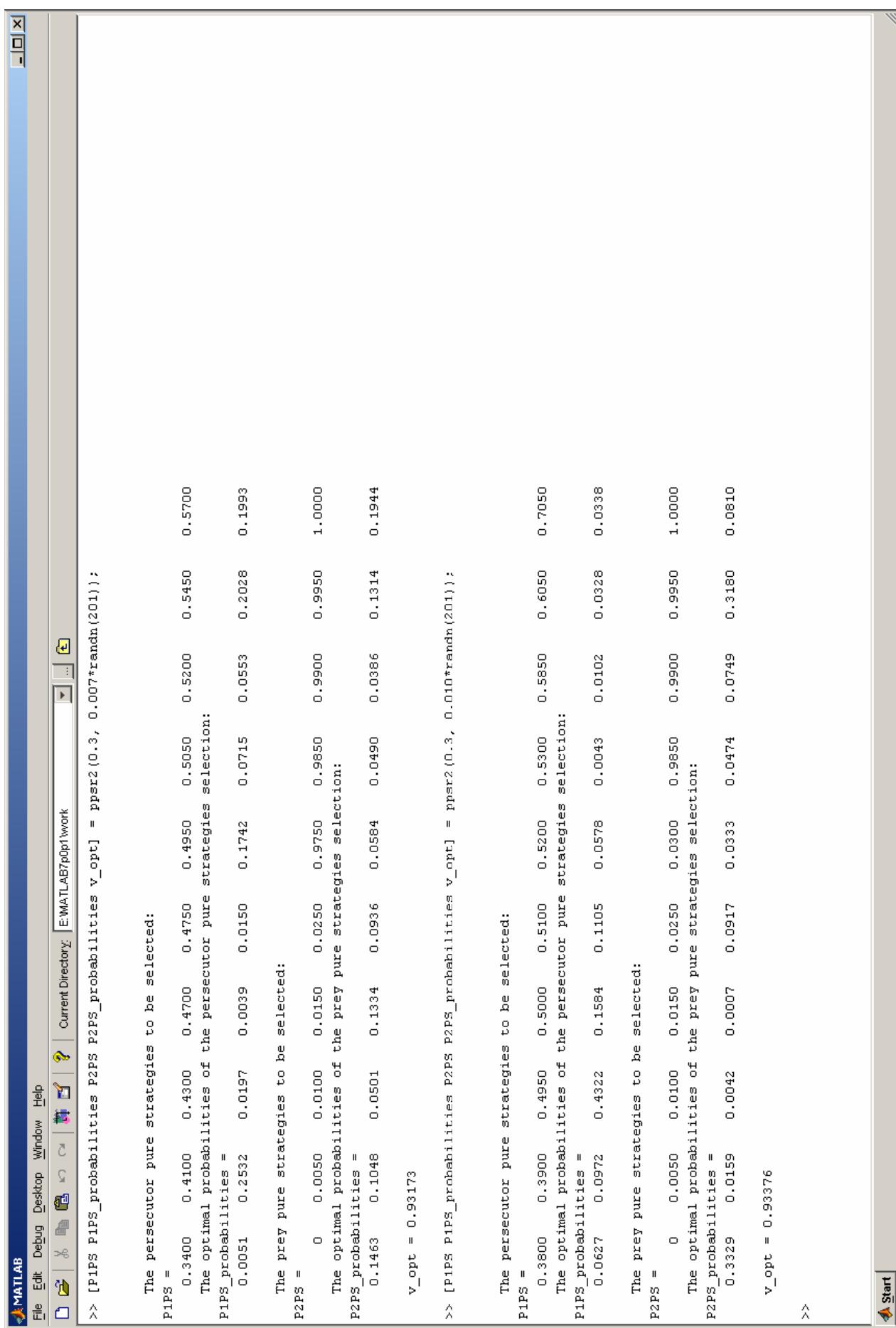
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.3, 0.004*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4600    0.4650    0.5350    0.5400    0.5450
    The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.1014    0.4521    0.3302    0.1036    0.0127

The prey pure strategies to be selected:
P2PS =
    0.0050    0.0100    0.9900    0.9950    1.0000
    The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2864    0.1853    0.0654    0.3523    0.1107
v_opt = 0.93092
>>

```

Figure 41. Program module ppsr2 application with $\alpha = 0.3$ and more great noise amplitudes



The screenshot shows a MATLAB application window with the title bar "Applied Math". The menu bar includes File, Edit, Debug, Desktop, Window, Help, and several icons. The current directory is set to E:\MATLAB\B\pp01\work. The command window displays the following code and its execution results:

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.3, 0.007*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.3400    0.4100    0.4300    0.4700    0.4750    0.4950    0.5050    0.5200    0.5450    0.5700
    The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.0051    0.2532    0.0197    0.0039    0.0150    0.1742    0.0715    0.0553    0.2028    0.1993

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.0150    0.0250    0.9750    0.9950    0.9900    0.9950    1.0000
    The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.1463    0.1048    0.0501    0.1334    0.0936    0.0584    0.0490    0.0386    0.1314    0.1944

v_opt = 0.93173

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.3, 0.010*randn(201));

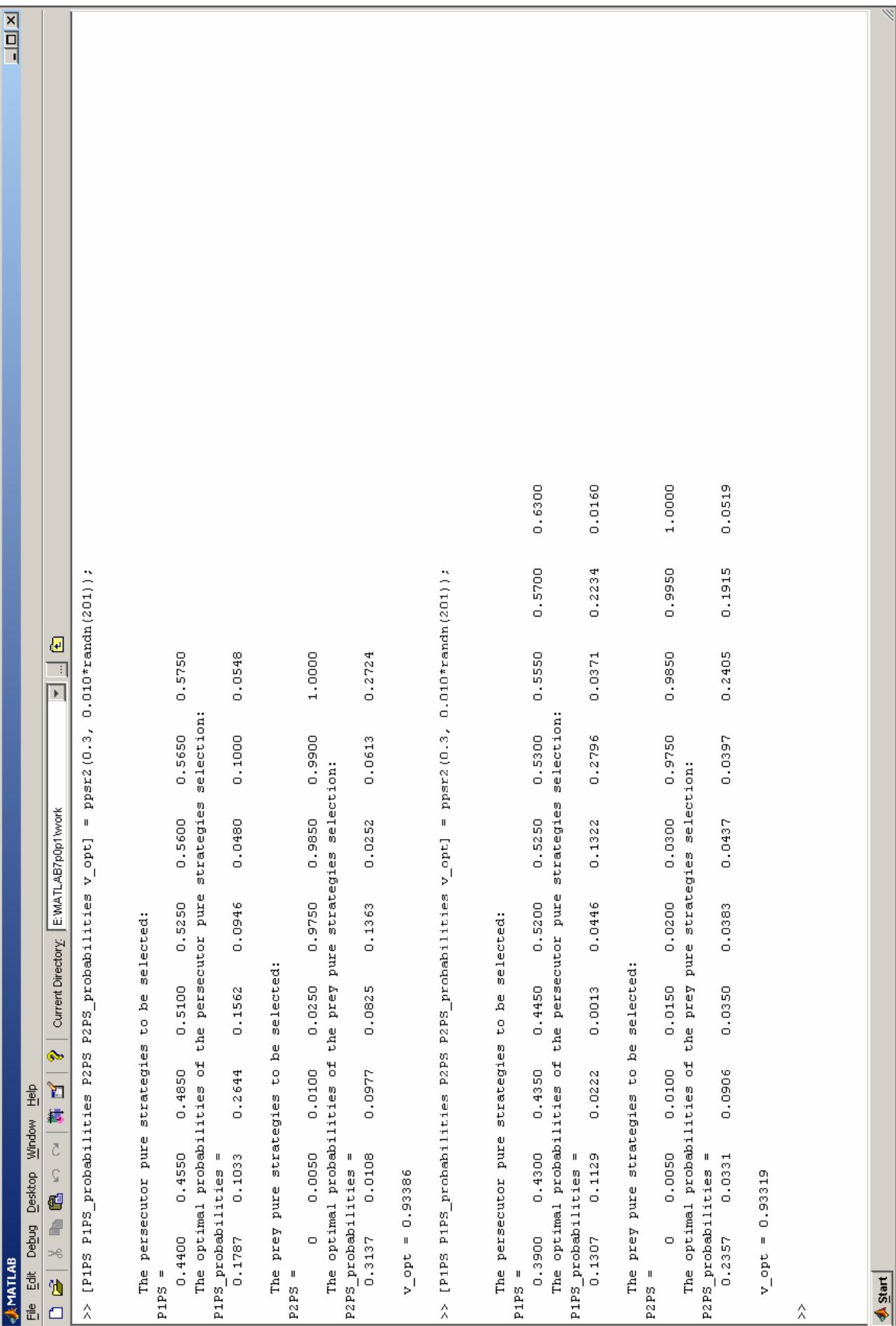
The persecutor pure strategies to be selected:
P1PS =
    0.3800    0.3900    0.4950    0.5000    0.5100    0.5200    0.5300    0.5850    0.6050    0.7050
    The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.0627    0.0972    0.4322    0.1584    0.1105    0.0578    0.0043    0.0102    0.0328    0.0338

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.0150    0.0250    0.0300    0.9850    0.9900    0.9950    1.0000
    The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.3329    0.0159    0.0042    0.0007    0.0917    0.0333    0.0474    0.0749    0.3180    0.0810

v_opt = 0.93376
>>

```

Figure 42. Program module ppsr2 application with $\alpha = 0.3$ and some high noise amplitude



The screenshot shows a MATLAB window with two identical command-line sessions. Both sessions start with the command `>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppst2(0.3, 0.010*randn(201));`. The first session continues with:

```

The persecutor pure strategies to be selected:
P1PS =
    0.4400    0.4550    0.4850    0.5100    0.5250    0.5600    0.5650    0.5750
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.1787    0.1033    0.2644    0.1562    0.0946    0.0480    0.1000    0.0548

```

The second session continues with:

```

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.0250    0.9750    0.9850    0.9900    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.3137    0.0108    0.0977    0.0825    0.1363    0.0252    0.0613    0.2724
v_opt = 0.93386

```

Both sessions then repeat the command `>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppst2(0.3, 0.010*randn(201));`. The third session continues with:

```

The persecutor pure strategies to be selected:
P1PS =
    0.3900    0.4300    0.4350    0.4450    0.5200    0.5250    0.5300    0.5550    0.5700    0.6300
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.1307    0.1129    0.0222    0.0013    0.0446    0.1322    0.2796    0.0371    0.2234    0.0160

```

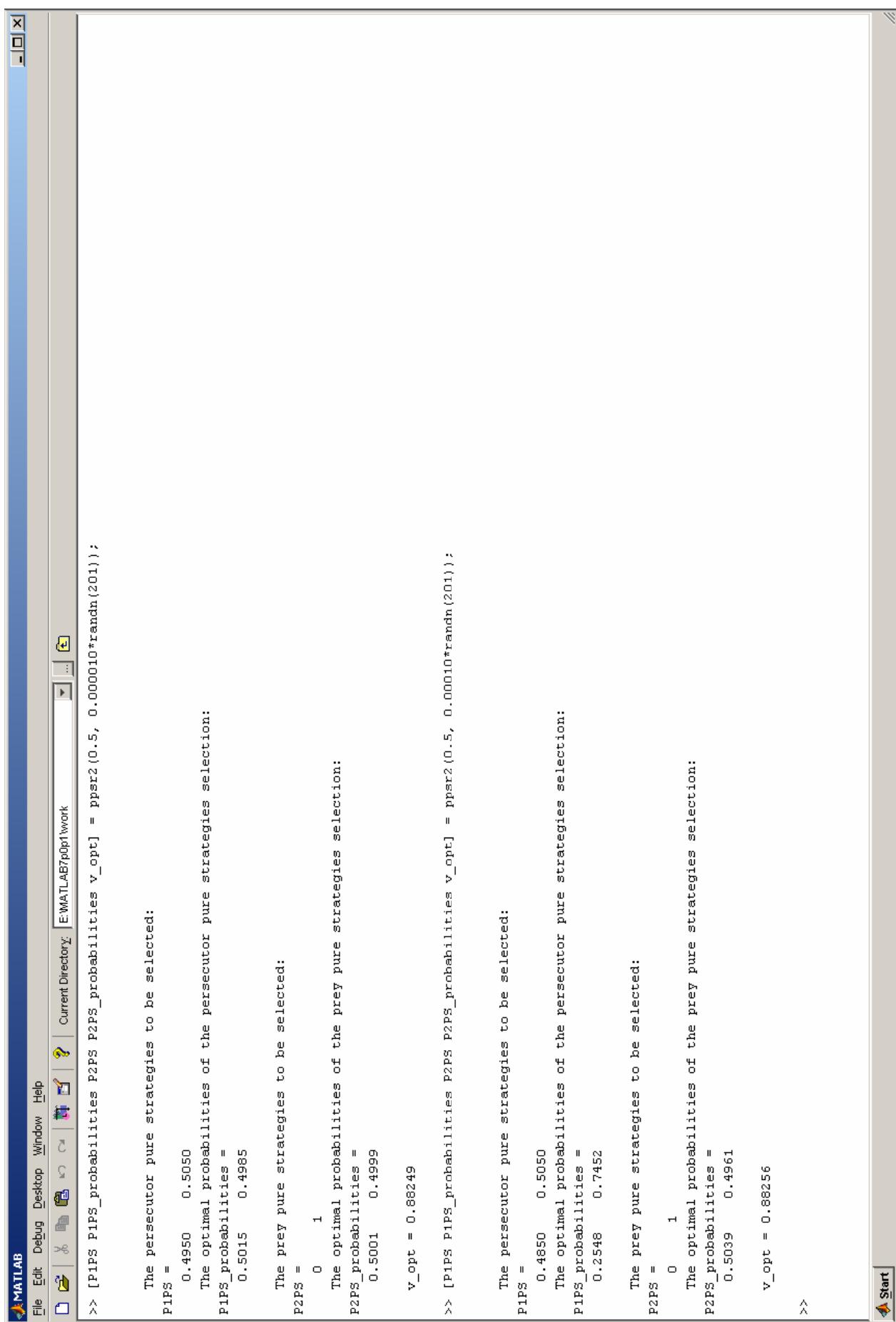
The fourth session continues with:

```

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.0150    0.0200    0.0300    0.9750    0.9850    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2357    0.0331    0.0906    0.0350    0.0383    0.0437    0.0397    0.2405    0.1915    0.0519
v_opt = 0.93319
>>

```

Figure 43. The two repeats of the previous runs



The screenshot shows a MATLAB application window with the title bar "Applied Math". The menu bar includes File, Edit, Debug, Desktop, Window, Help, and a Current Directory set to E:\MATLAB7p0\pd1\work. The central workspace displays the following code and output:

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.5, 0.000010*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4950    0.5050
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.5015    0.4985

The prey pure strategies to be selected:
P2PS =
    0         1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.5001    0.4999
v_opt = 0.88249

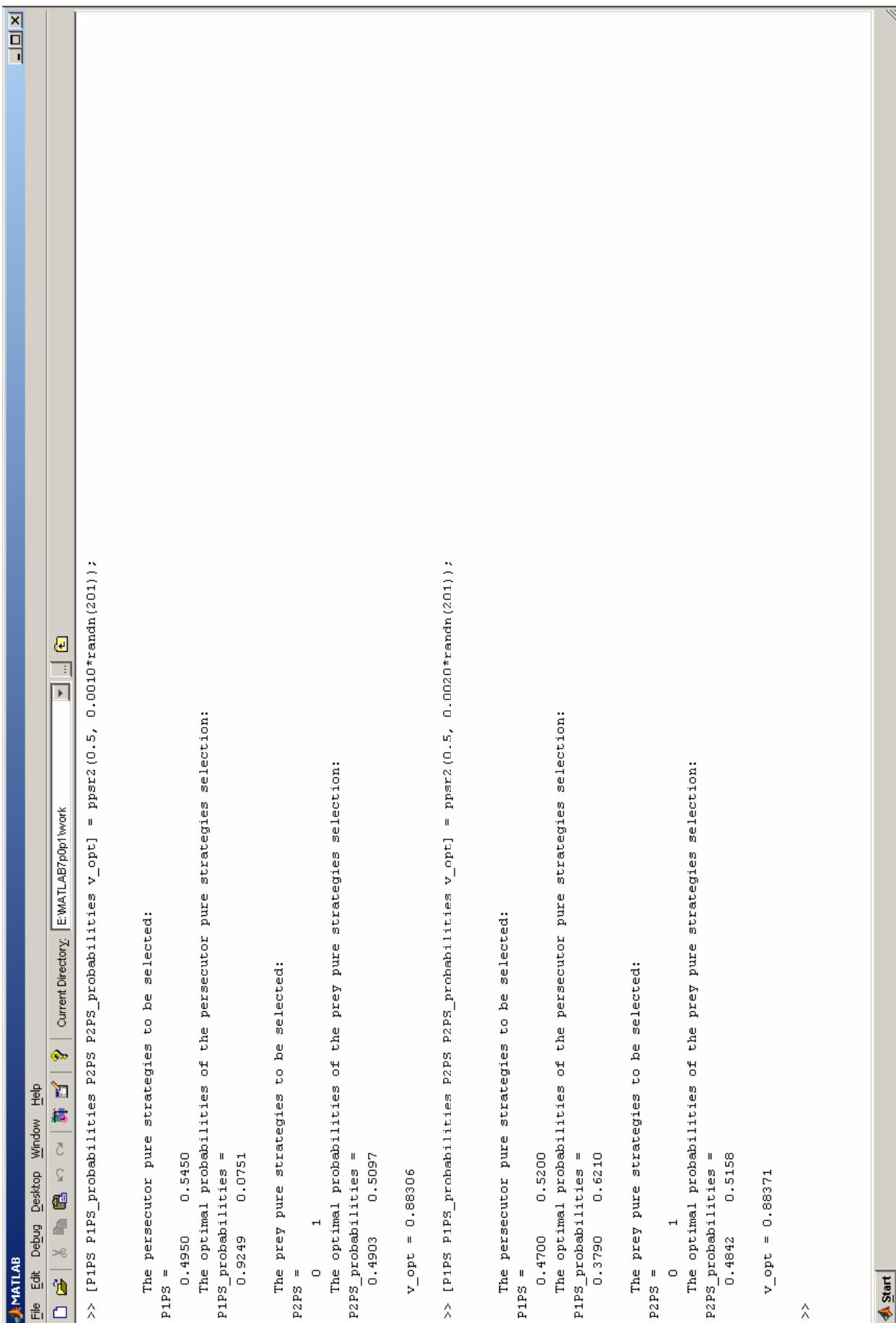
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.5, 0.000010*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4850    0.5050
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.2548    0.7452

The prey pure strategies to be selected:
P2PS =
    0         1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.5039    0.4961
v_opt = 0.88256
>>

```

Figure 44. Program module ppsr2 application with $\alpha = 0.5$ by very low noise amplitude and ten times greater one



The persecutor pure strategies to be selected:

```
P1PS =
0.4950 0.5450
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
0.9249 0.0751
```

The prey pure strategies to be selected:

```
P2PS =
0 1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
0.4903 0.5097
v_opt = 0.88306
```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = rpsr2(0.5, 0.0010*randn(201));

The persecutor pure strategies to be selected:

```
P1PS =
0.4700 0.5200
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
0.3790 0.6210
```

The prey pure strategies to be selected:

```
P2PS =
0 1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
0.4842 0.5158
v_opt = 0.88371
>>
```

Figure 45. Continuing to increase the noise amplitude by $\alpha = 0.5$

Applied Math

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.5, 0.020*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.3900    0.3950    0.4100    0.4250    0.4500    0.4800    0.5100    0.5300    0.5350    0.5400    0.5600    0.5800    0.6750
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.0565    0.0030    0.1138    0.0274    0.1056    0.0998    0.0704    0.2346    0.0334    0.0812    0.1092    0.0145    0.0506

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.0200    0.0350    0.0400    0.9650    0.9750    0.9800    0.9850    0.9900    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.0477    0.1540    0.0536    0.0337    0.0396    0.1778    0.0095    0.0445    0.0838    0.1628    0.0232    0.0530    0.1169

v_opt = 0.89417

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.5, 0.020*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4150    0.4350    0.4450    0.4500    0.4550    0.4650    0.4800    0.4950    0.5000    0.5600    0.5850    0.6000    0.6250
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.0058    0.0958    0.1964    0.0245    0.0658    0.0142    0.1363    0.1053    0.0232    0.0206    0.1328    0.1354    0.0440

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.0150    0.0200    0.0250    0.0300    0.9700    0.9800    0.9850    0.9900    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.0282    0.0727    0.0987    0.0705    0.0110    0.0997    0.1247    0.0645    0.0981    0.0762    0.0342    0.1329    0.0887

v_opt = 0.892
>>

```

Figure 46. Program module ppsr2 application with $\alpha = 0.5$ by pretty high noise amplitude

The screenshot shows a MATLAB 7.0 interface with the following code and output:

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.5, 0.0040*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4500    0.5400    0.5950
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.6136    0.1287    0.2577

The prey pure strategies to be selected:
P2PS =
    0         0.0050    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.1806    0.3154    0.5040
v_opt = 0.88565

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(0.5, 0.0045*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4700    0.4950    0.5150    0.6050    0.6350
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.3457    0.4272    0.1051    0.0727    0.0493

The prey pure strategies to be selected:
P2PS =
    0         0.0050    0.9800    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.1506    0.3191    0.0267    0.4370    0.0666
v_opt = 0.88695
>>

```

Figure 47. Optimal probabilities by $\alpha = 0.5$ for some decreased noise amplitude

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = pnsr2(0.5, 0.0050*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4650    0.4900    0.5000    0.5050    0.5500
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.2662    0.0488    0.1346    0.4221    0.1283

The prey pure strategies to be selected:
P2PS =
    0    0.0150    0.9900    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.3471    0.1207    0.1399    0.1078    0.2846
v_opt = 0.88624

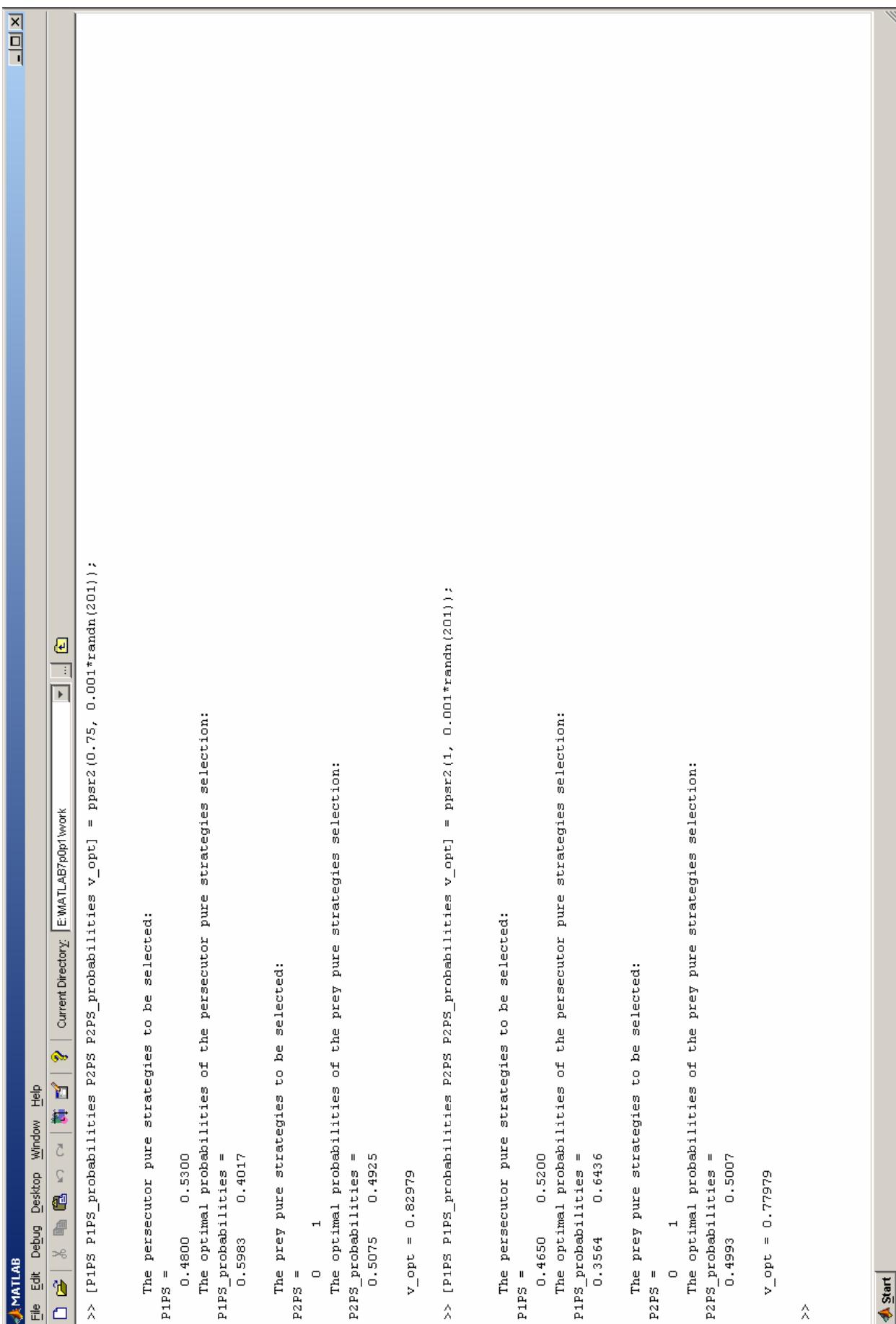
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = pnsr2(0.5, 0.0055*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4100    0.4700    0.4950    0.5100    0.5150    0.5300
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.0210    0.2269    0.0931    0.2186    0.4347    0.0057

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.0100    0.9900    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.1555    0.2796    0.0568    0.1117    0.3098    0.0866
v_opt = 0.88659
>>

```

Figure 48. Further increasing the noise amplitude, and the spectrums of the optimal strategies of both the persecutor and prey keep widen



The persecutor pure strategies to be selected:

```
P1PS =
    0.4800    0.5300
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.5983    0.4017
```

The prey pure strategies to be selected:

```
P2PS =
    0         1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.5075    0.4925
v_opt = 0.82979
```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = rpsr2 (0.75, 0.001*randn(201));

The persecutor pure strategies to be selected:

```
P1PS =
    0.4650    0.5200
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.3564    0.6436
```

The prey pure strategies to be selected:

```
P2PS =
    0         1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.4993    0.5007
v_opt = 0.77979
>>
```

Figure 49. Taking $\alpha = 0.75$ or $\alpha = 1$ at the low noise amplitude

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(2, 0.001*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.3750    0.6250
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.5028    0.4972

The prey pure strategies to be selected:
P2PS =
    0         1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.4991    0.5009
v_opt = 0.60802

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(2, 0.003*randn(201));

The persecutor pure strategies to be selected:
P1PS =
    0.4550    0.6450
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.7585    0.2415

The prey pure strategies to be selected:
P2PS =
    0         1
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.4990    0.5010
v_opt = 0.61141
>>

```

Figure 50. Antisymmetrization for $\alpha = 2$ by increasing the noise amplitude

The screenshot shows a MATLAB window with the title bar "Applied Math". The menu bar includes File, Edit, Debug, Desktop, Window, Help. The toolbar has icons for New, Open, Save, Print, Copy, Paste, Find, and Current Directory. The current directory is set to E:\MATLAB7\p0p1\work.

The code in the command window is:

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = opr2(2, 0.005*randn(201));

```

The output is:

```

The persecutor pure strategies to be selected:
P1PS =
    0.4300    0.5800    0.6950    0.7300
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.6277    0.2449    0.1037    0.0237

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.9900    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.3712    0.1225    0.0175    0.4888
v_opt = 0.61398

```

The next part of the code is:

```

>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = opr2(2, 0.01*randn(201));

```

The output is:

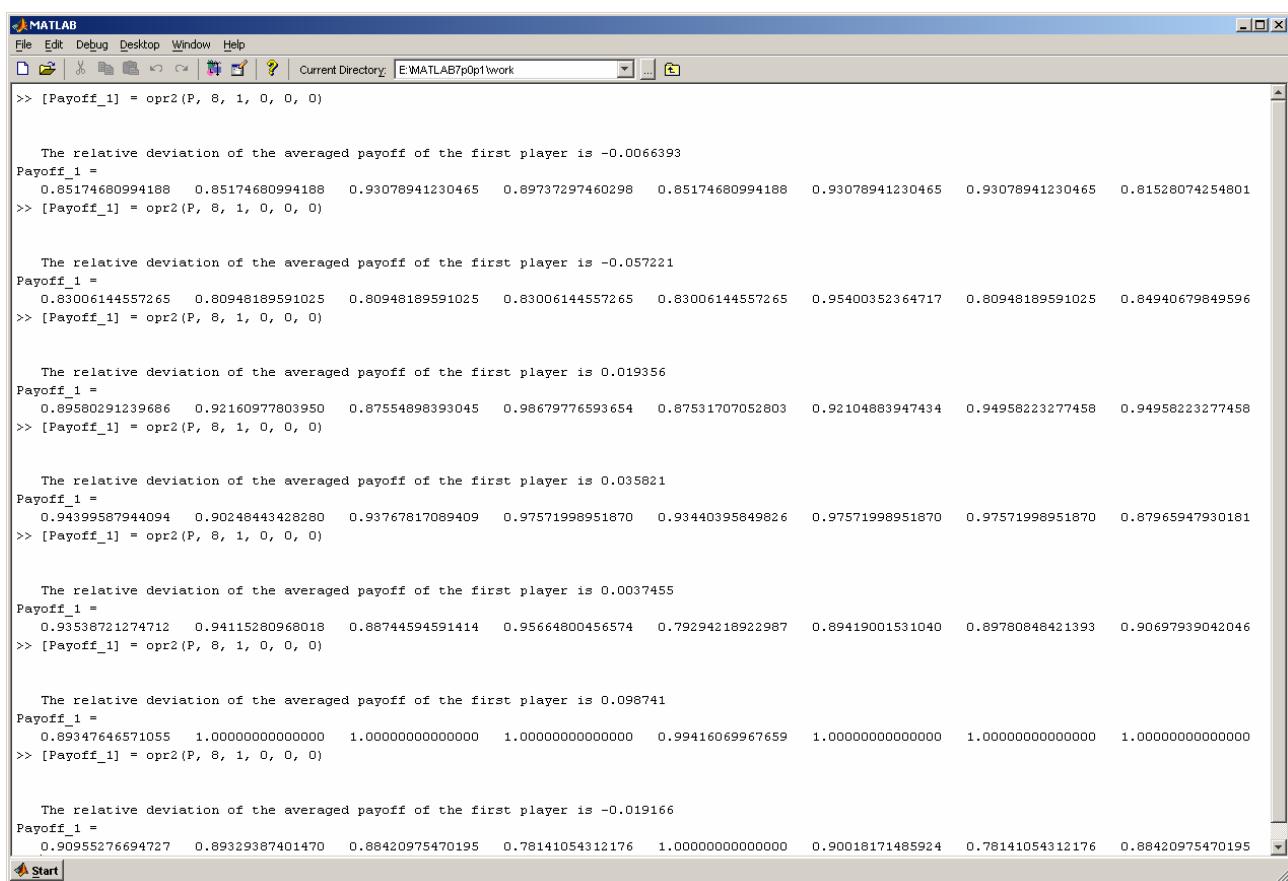
```

The persecutor pure strategies to be selected:
P1PS =
    0.2950    0.3650    0.4900    0.5100    0.6500
The optimal probabilities of the persecutor pure strategies selection:
P1PS_probabilities =
    0.0111    0.1167    0.2294    0.5833    0.0595

The prey pure strategies to be selected:
P2PS =
    0    0.0050    0.9850    0.9950    1.0000
The optimal probabilities of the prey pure strategies selection:
P2PS_probabilities =
    0.2186    0.2773    0.0022    0.0968    0.4048
v_opt = 0.61887
>>

```

Figure 51. Further increasing the noise amplitude at $\alpha = 2$, and the spectrums of the optimal strategies of both the players keep widening
Another new three examples of the module opr2 application are shown on the figures 52 — 54.



```

>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.0066393
Payoff_1 =
0.85174680994188 0.85174680994188 0.93078941230465 0.89737297460298 0.85174680994188 0.93078941230465 0.93078941230465 0.81528074254801
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.057221
Payoff_1 =
0.83006144557265 0.80948189591025 0.80948189591025 0.83006144557265 0.83006144557265 0.95400352364717 0.80948189591025 0.84940679849596
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.019356
Payoff_1 =
0.89580291239686 0.92160977803950 0.87554898393045 0.98679776593654 0.87531707052803 0.92104883947434 0.94958223277458 0.94958223277458
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.035821
Payoff_1 =
0.94399587944094 0.90248443428280 0.93767817089409 0.97571998951870 0.93440395849826 0.97571998951870 0.97571998951870 0.87965947930181
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

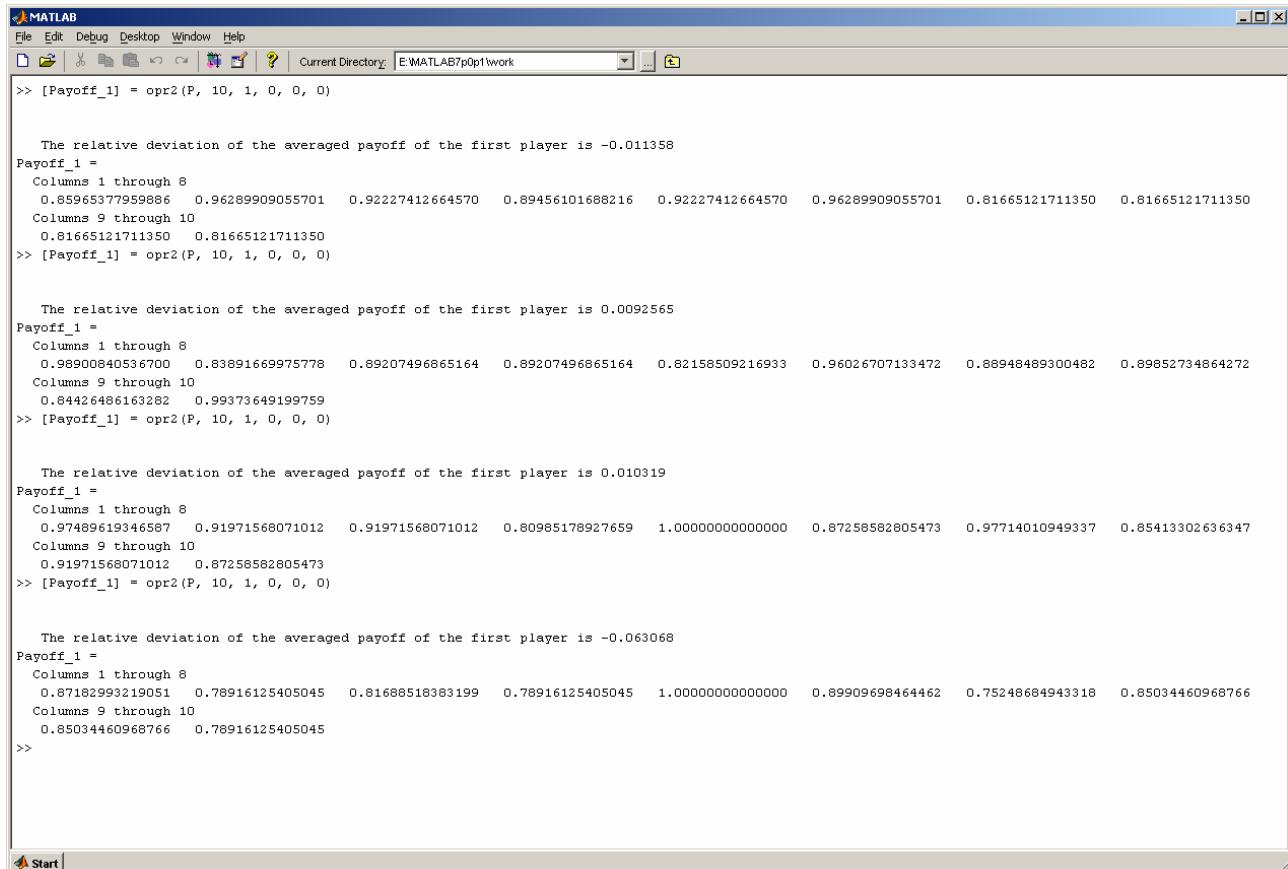
The relative deviation of the averaged payoff of the first player is 0.0037455
Payoff_1 =
0.93538721274712 0.94115280968018 0.88744594591414 0.95664800456574 0.79294218922987 0.89419001531040 0.89780848421393 0.90697939042046
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.098741
Payoff_1 =
0.89347646571055 1.00000000000000 1.00000000000000 0.99416069967659 1.00000000000000 1.00000000000000 1.00000000000000
>> [Payoff_1] = opr2(P, 8, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.019166
Payoff_1 =
0.90955276694727 0.89329387401470 0.88420975470195 0.78141054312176 1.00000000000000 0.90018171485924 0.78141054312176 0.88420975470195

```

Figure 52. The module opr2 application with the eight shots by increasing the noise level



```

>> [Payoff_1] = opr2(P, 10, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.011358
Payoff_1 =
Columns 1 through 8
0.85965377959886 0.96289909055701 0.92227412664570 0.89456101688216 0.92227412664570 0.96289909055701 0.81665121711350 0.81665121711350
Columns 9 through 10
0.81665121711350 0.81665121711350
>> [Payoff_1] = opr2(P, 10, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.0092565
Payoff_1 =
Columns 1 through 8
0.98900840536700 0.83891669975778 0.89207496865164 0.89207496865164 0.82158509216933 0.96026707133472 0.88948489300482 0.89852734864272
Columns 9 through 10
0.84426486163282 0.99373649199759
>> [Payoff_1] = opr2(P, 10, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is 0.010319
Payoff_1 =
Columns 1 through 8
0.97489619346587 0.91971568071012 0.91971568071012 0.80985178927659 1.00000000000000 0.87258582805473 0.97714010949337 0.85413302636347
Columns 9 through 10
0.91971568071012 0.87258582805473
>> [Payoff_1] = opr2(P, 10, 1, 0, 0, 0)

The relative deviation of the averaged payoff of the first player is -0.063068
Payoff_1 =
Columns 1 through 8
0.87182993219051 0.78916125405045 0.81688516383199 0.78916125405045 1.00000000000000 0.89909698464462 0.75248684943318 0.85034460968766
Columns 9 through 10
0.85034460968766 0.78916125405045
>>

```

Figure 53. The module opr2 application with the 10 shots by increasing the noise level

```

>> [Payoff_1] = opr2(P, 10, 1, 0, 0, 1)

Now the first player has selected the pure strategy x17
Now the second player has selected the pure strategy y44
Now the first player real payoff is 0.79253
Now the first player has selected the pure strategy x11
Now the second player has selected the pure strategy y49
Now the first player real payoff is 0.79354
Now the first player has selected the pure strategy x11
Now the second player has selected the pure strategy y49
Now the first player real payoff is 0.79354
Now the first player has selected the pure strategy x11
Now the second player has selected the pure strategy y49
Now the first player real payoff is 0.79354
Now the first player has selected the pure strategy x11
Now the second player has selected the pure strategy y6
Now the first player real payoff is 1
Now the first player has selected the pure strategy x23
Now the second player has selected the pure strategy y49
Now the first player real payoff is 0.85375
Now the first player has selected the pure strategy x11
Now the second player has selected the pure strategy y44
Now the first player real payoff is 0.75982
Now the first player has selected the pure strategy x11
Now the second player has selected the pure strategy y7
Now the first player real payoff is 1
Now the first player has selected the pure strategy x11
Now the second player has selected the pure strategy y7
Now the first player real payoff is 1
Now the first player has selected the pure strategy x19
Now the second player has selected the pure strategy y48
At last, the first player real payoff is 0.93638
The relative deviation of the averaged payoff of the first player is -0.03361

Payoff_1 =
Columns 1 through 8
0.79252786153978 0.79353800957288 0.79353800957288 0.79353800957288 1.0000000000000000 0.85375059281326 0.75982488499809 1.0000000000000000
Columns 9 through 10
1.0000000000000000 0.93637793179002
>>

```

Figure 54. The module opr2 application with the 10 shots by some high noise level and the scheme of selecting the pure strategies

The executive form within MATLAB 7.0.1, gathered all the parameters and options of the explored antagonistic game persecution model, is viewed on the figure 55. This application, named `eppsr2`, works only with the mouseclicks by pulling the two scrolls to define the parameter α and the roughness $n(x, y)$ amplitude (root-mean-square deviation). The returned pure strategies in the optimal strategies of the persecutor and the prey are visualized on the surface (2) $[0; 1] \times [0; 1]$ -square plot, helping to conceive them more physically (figures 56 — 65). Here, on the being presented form, a user does not define the roughness $n(x, y)$ precisely by its sampled points, but holds or controls the level of that noise. This feature may occur useful, when a user just trains to possess the described simplest antagonistic game persecution model, for its further application from the MATLAB 7.0.1 Command Window line, inputting the real roughness $n(x, y)$ sampled points into the program module `eppsr2` and running it.

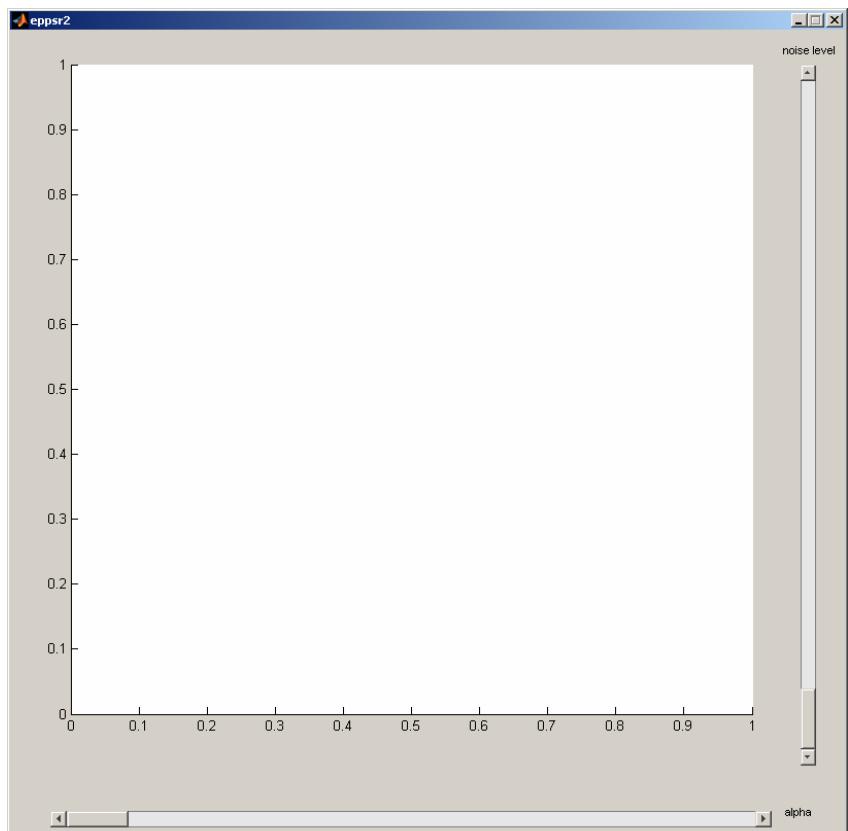


Figure 55. The executive form **eppsr2** general view as it just has been loaded by the typing **eppsr2** and entering it from the MATLAB 7.0.1 Command Window line

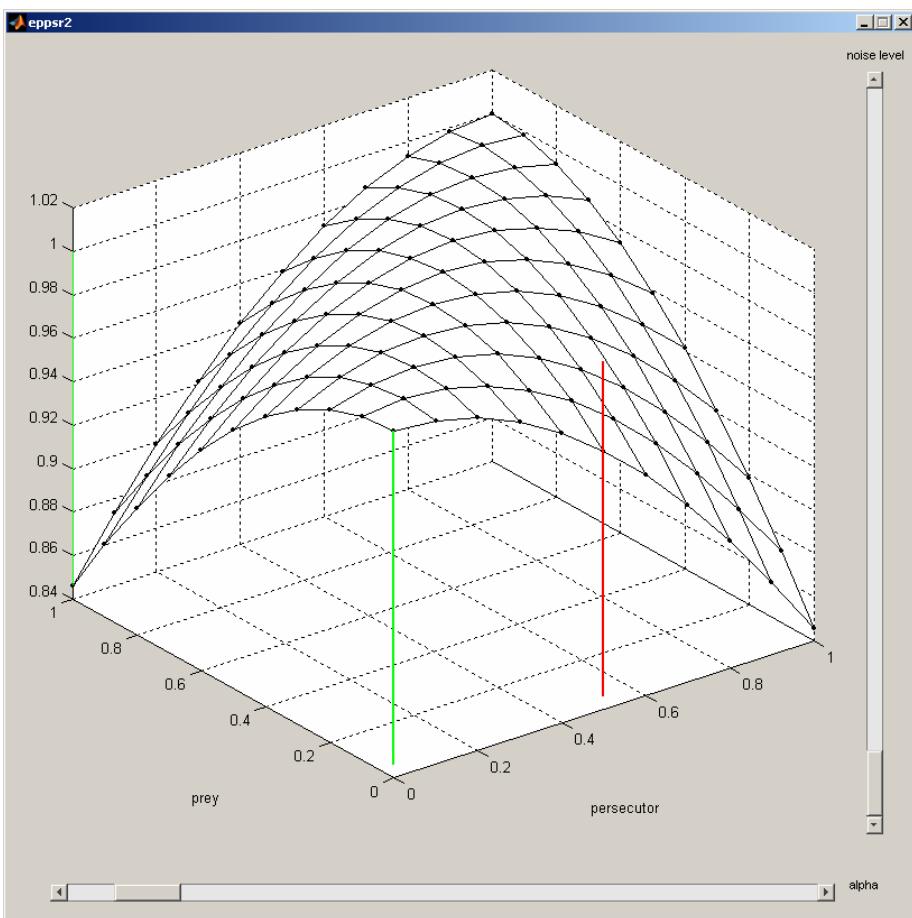


Figure 56. The single optimal pure strategy of the persecutor and the two prey pure strategies to be selected by the low α and the almost lowest noise level

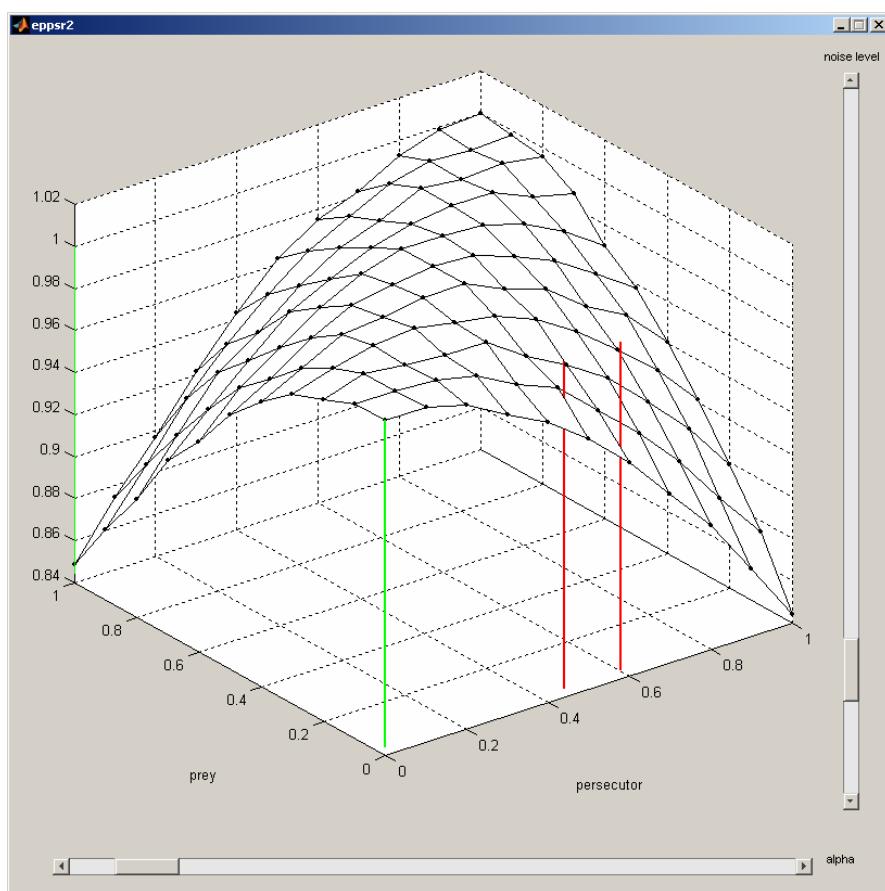


Figure 57. The equilibrium situation by some noise level increment

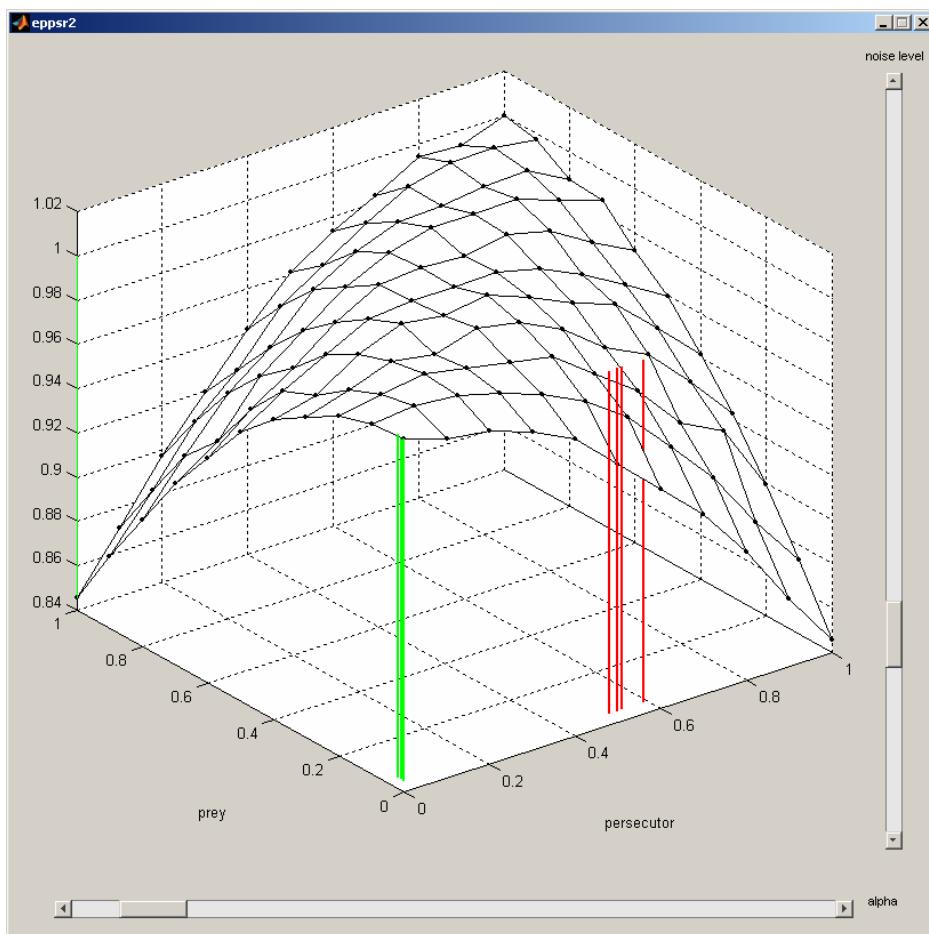


Figure 58. Appearance of the four pure strategies to be selected for each player by further noise level increasing

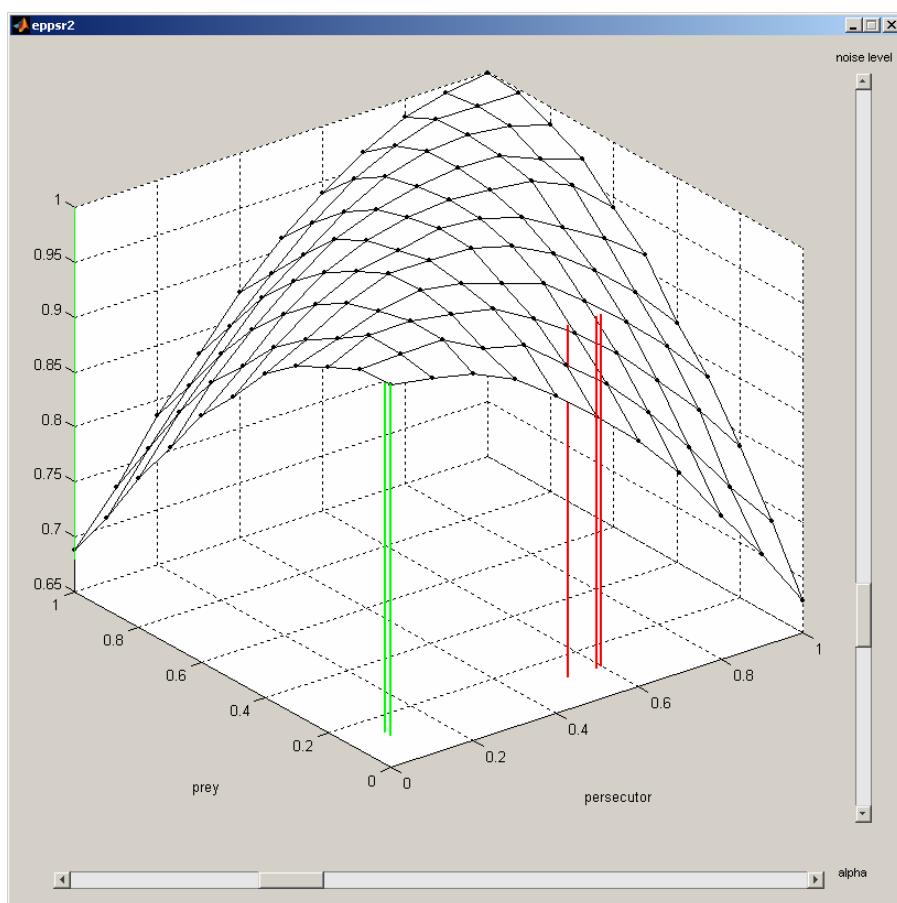


Figure 59. Disappearance of the one pure strategy to be selected for each player by some α increment

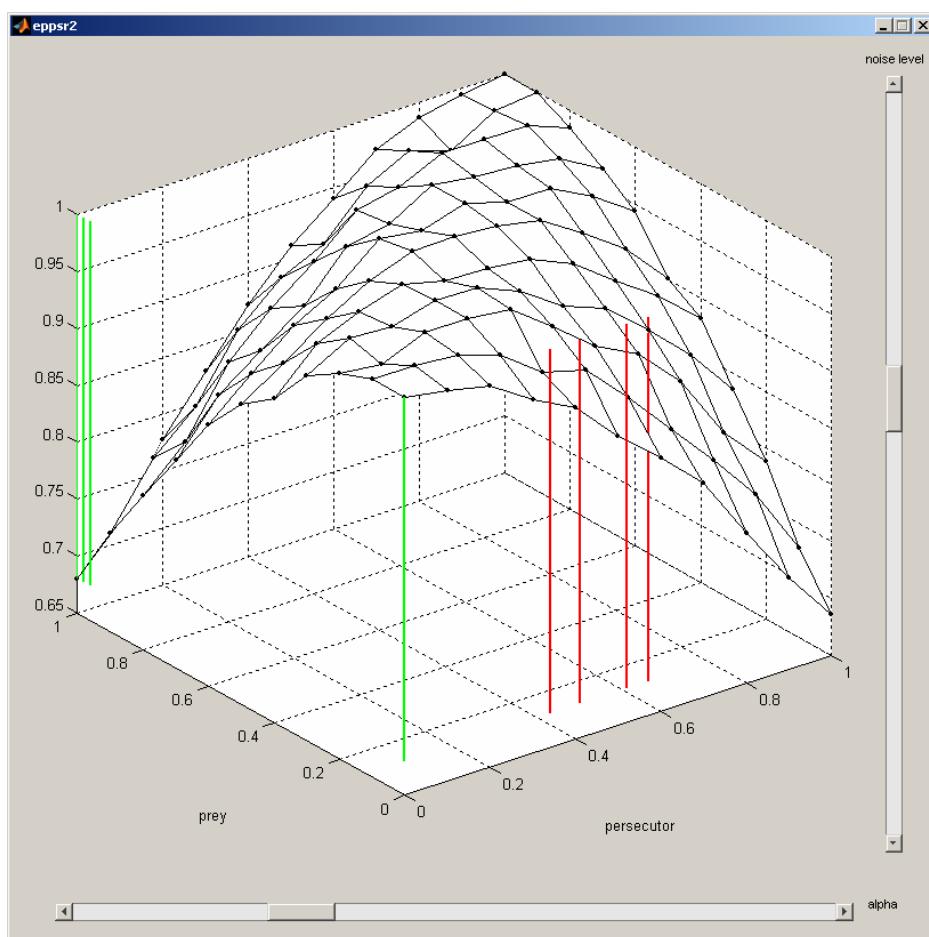


Figure 60. Again each player has the four pure strategies to be selected by further noise level increasing

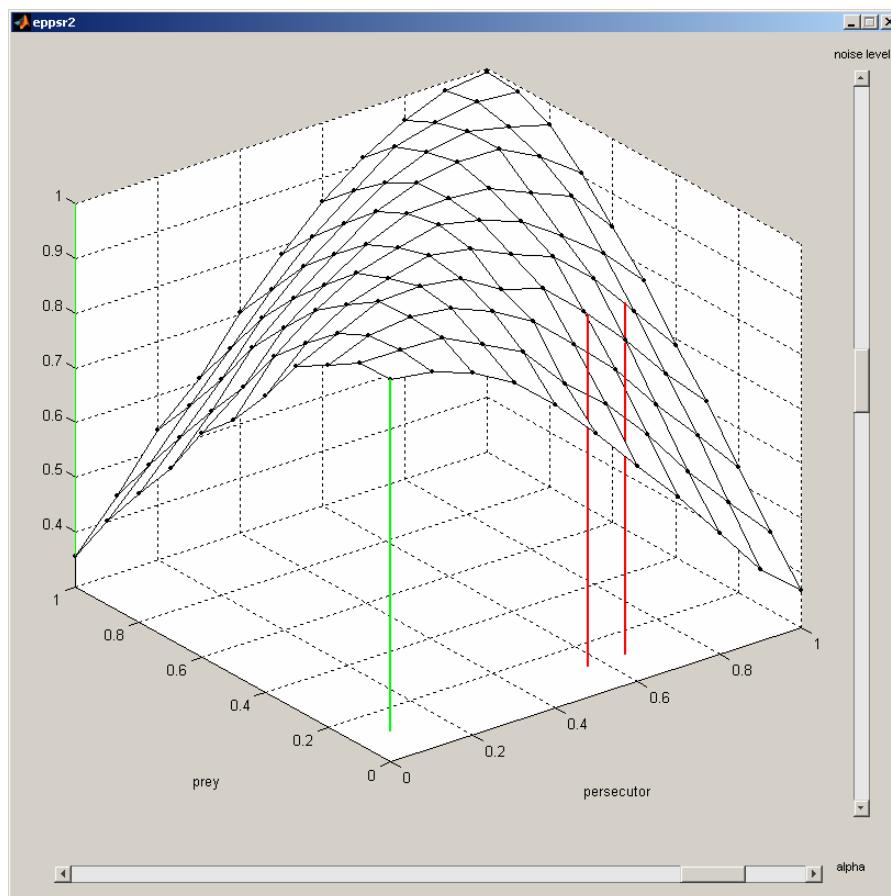


Figure 61. Further increasing α drives to the pure strategies to be selected reduction

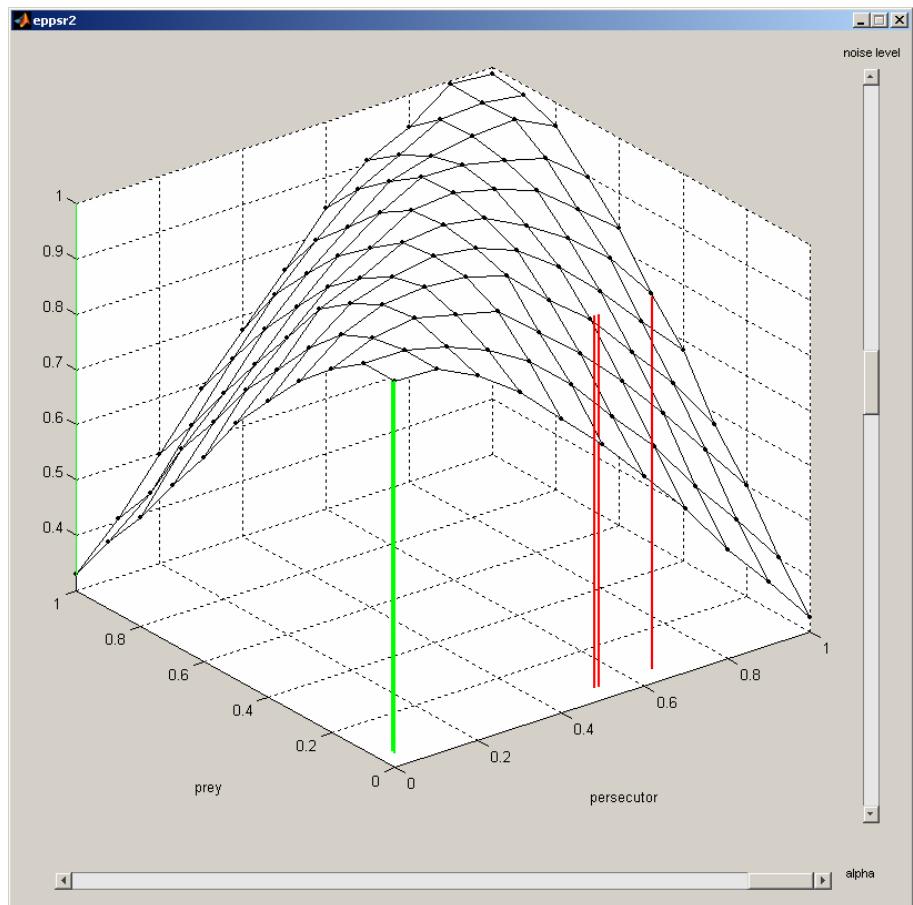


Figure 62. By the maximal α there may be some stratification of the pure strategies to be selected

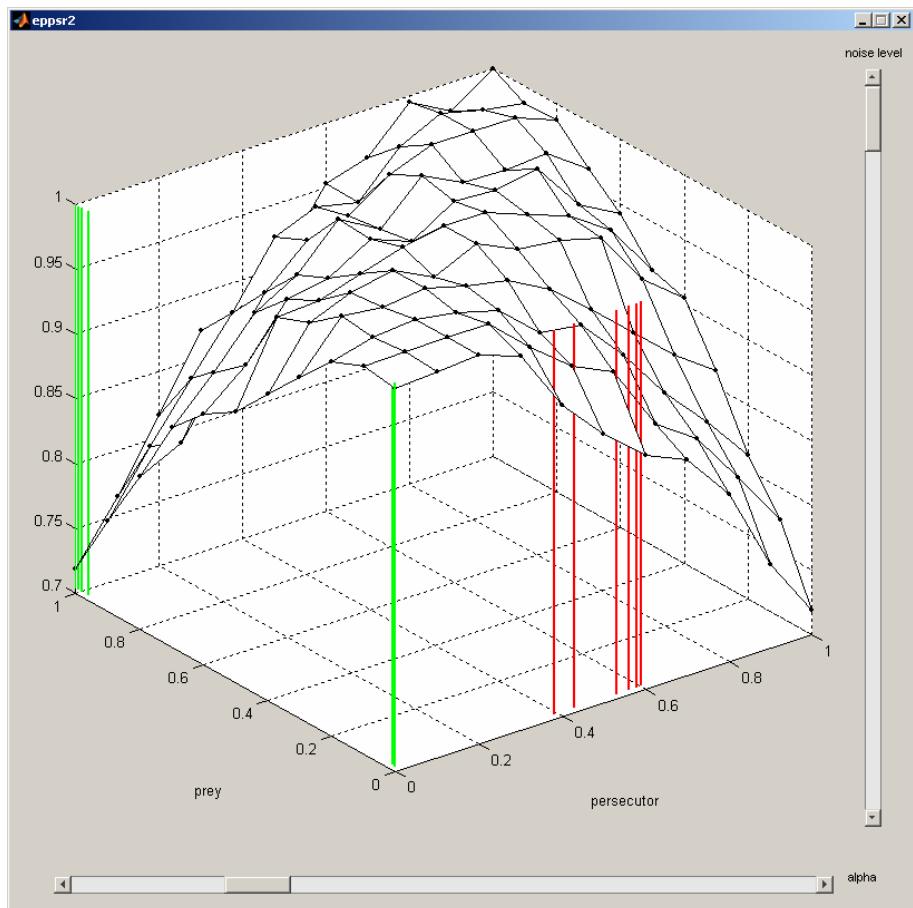


Figure 63. At the maximal noise level and not great α , there appear more of the pure strategies to be selected

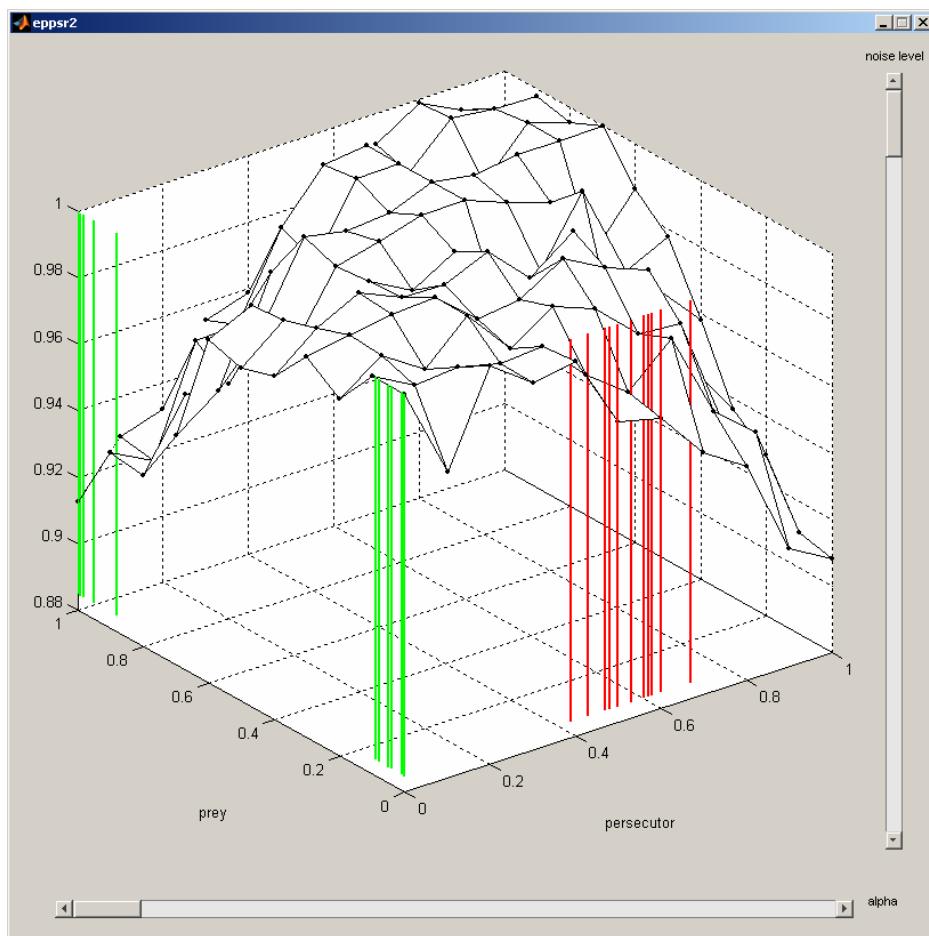


Figure 64. At the maximal noise level and the lowest α the number of the pure strategies to be selected becomes maximal for each player

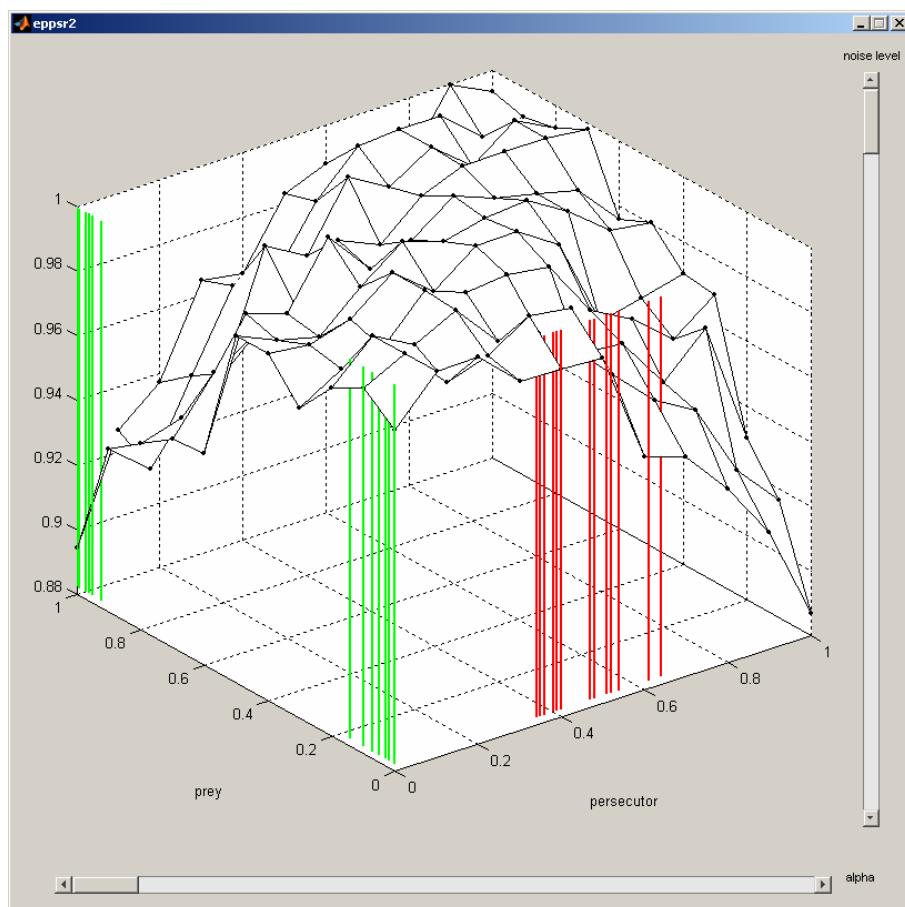


Figure 65. Another example with the maximal noise level and the lowest α

Conclusion

This paper generalizes the persecution model in the form of the antagonistic game with the kernel (2), that is where the kernel is the non-ideal exponential surface. For the generalization there has been updated the program module `ppsr2`, taking on the input α and the roughness $n(x, y)$. The module `ppsr2` returns the exact analytic solution, if the game is either concave or convex (maybe, even concave-convex), and it returns the approximate solution for other cases, presenting the optimal strategies of the players (persecutor and prey). To practice those optimal probabilities the players should be involved into the program module `opr2` procedure, accepting the optimal mixed strategy (or its spectrum) and returning the pure strategy number to be selected in the current game (current replay). The executive form `eppsr2` helps to understand deeper the whole realization process of the antagonistic game persecution model, returning in the end the optimal strategies of the persecutor and the prey, and plotting them on the surface (2) graph.

List of the used references

1. Вентцель Е. С. Элементы теории игр / Вентцель Е. С. — М. : Государственное издательство физико-математической литературы, 1961. — 67 с. — (Популярные лекции по математике ; вып. 32).
2. Романюк В. В. Разрешение системы преследователь — добыча для экспоненциальной вероятности поражения добычи преследователем / В. В. Романюк // Вестник НТУ "ХПИ". Тематический выпуск: Информатика и моделирование. — Харьков: НТУ "ХПИ", 2009. — № 13. — С. 138 — 149.
3. Воробьев Н. Н. Теория игр для экономистов-кибернетиков. — М.: Наука, Главная редакция физико-математической литературы, 1985. — 272 с.
4. Теория игр: Учеб. пособие для ун-тов / Л. А. Петросян, Н. А. Зенкевич, Е. А. Семина. — М.: Высшая школа, Книжный дом "Университет", 1998. — 304 с.: ил.
5. Оуэн Г. Теория игр: Пер. с англ. Изд. 2-е. — М.: Едиториал УРСС, 2004. — 216 с.
6. Romanuke V. V. A strictly convex game on the unit square and its solution five versions / V. V. Romanuke // Науково-теоретичний журнал Хмельницького економічного університету "Наука й економіка". — Випуск 4 (12), 2008. — С. 381 — 388.
7. Романюк В. В. Загальні розв'язки однієї неперервної антагоністичної гри / В. В. Романюк // Науково-теоретичний журнал Хмельницького економічного університету "Наука й економіка". — Випуск 4 (8), 2007. — С. 73 — 100.
8. Романюк В. В. Базові вісім співвідношень для шести видів розв'язку однієї неперервної антагоністичної строго випуклої гри / В. В. Романюк // Вісник Хмельницького національного університету. Технічні науки. — 2009. — № 1. — С. 261 — 268.
9. Романюк В. В. Чотири опорних співвідношення для чотирьох видів розв'язку однієї строго випуклої неперервної антагоністичної гри / В. В. Романюк // Вимірювальна та обчислювальна техніка в технологічних процесах. — 2008. — № 1. — С. 169 — 174.
10. Romanuke V. V. The 12 situations in the kernel of a continuous strictly convex antagonistic game and the nine game solution forms / V. V. Romanuke // Информационно-вычислительные технологии и их приложения: сборник статей IX Международной научно-технической конференции. — Пенза: РИО ПГСХА, 2008. — С. 247 — 257.
11. Романюк В. В. Програмний модуль для оптимізації використання гравцями тактики перебору чистих стратегій у 2×2 -гри без сідової точки / В. В. Романюк // Вісник Хмельницького національного університету. Економічні науки. — 2008. — № 6. — Т. 3. — С. 138 — 141.
12. Романюк В. В. Моделювання реалізації оптимальних змішаних стратегій в антагоністичній грі з двома чистими стратегіями в кожного з гравців / В. В. Романюк // Наукові вісті НТУУ "КПІ". — 2007. — № 3. — С. 74 — 77.
13. Romanuke V. V. On the issue of applying the pure strategies selection tactics in the matrix 2×2 -game / V. V. Romanuke // Збірник наукових праць факультету прикладної математики та комп'ютерних технологій Хмельницького національного університету. — 2008. — № 1. — С. 25 — 37.
14. Романюк В. В. Метод реалізації принципу оптимальності у матричних іграх без сідової точки / В. В. Романюк // Вісник НТУ "ХПИ". Тематичний випуск: Інформатика та моделювання. — Харків: НТУ "ХПИ", 2008. — № 49. — С. 146 — 154.
15. Романюк В. В. Тактика перебору чистих стратегій як теоретичне підґрунтя для дослідження ефективності різних способів реалізації оптимальних змішаних стратегій / В. В. Романюк // Наукові вісті НТУУ "КПІ". — 2008. — № 3. — С. 61 — 68.
16. Романюк В. В. Метод реалізації оптимальних змішаних стратегій у матричній грі з порожньою множиною сідлових точок у чистих стратегіях з відомою кількістю партій гри / В. В. Романюк // Наукові вісті НТУУ "КПІ". — 2009. — № 2. — С. 45 — 52.
17. Романюк В. В. Адаптація методу реалізації оптимальних змішаних стратегій у матричній грі з порожньою множиною ситуацій рівноваги з відомою наперед кількістю раундів гри у програмному середовищі MATLAB / В. В. Романюк // Вісник Хмельницького національного університету. Технічні науки. — 2009. — № 4. — С. 57 — 67.