UDC 519.832.4

V. V. ROMANUKE
Khmelnytskyy National University

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

$$P(x, y) = \exp\left[-\alpha(x-y)^2\right] \tag{1}$$

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,

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

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

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

and its second derivative is

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

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

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

then the inequality

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

is identical to the inequality

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

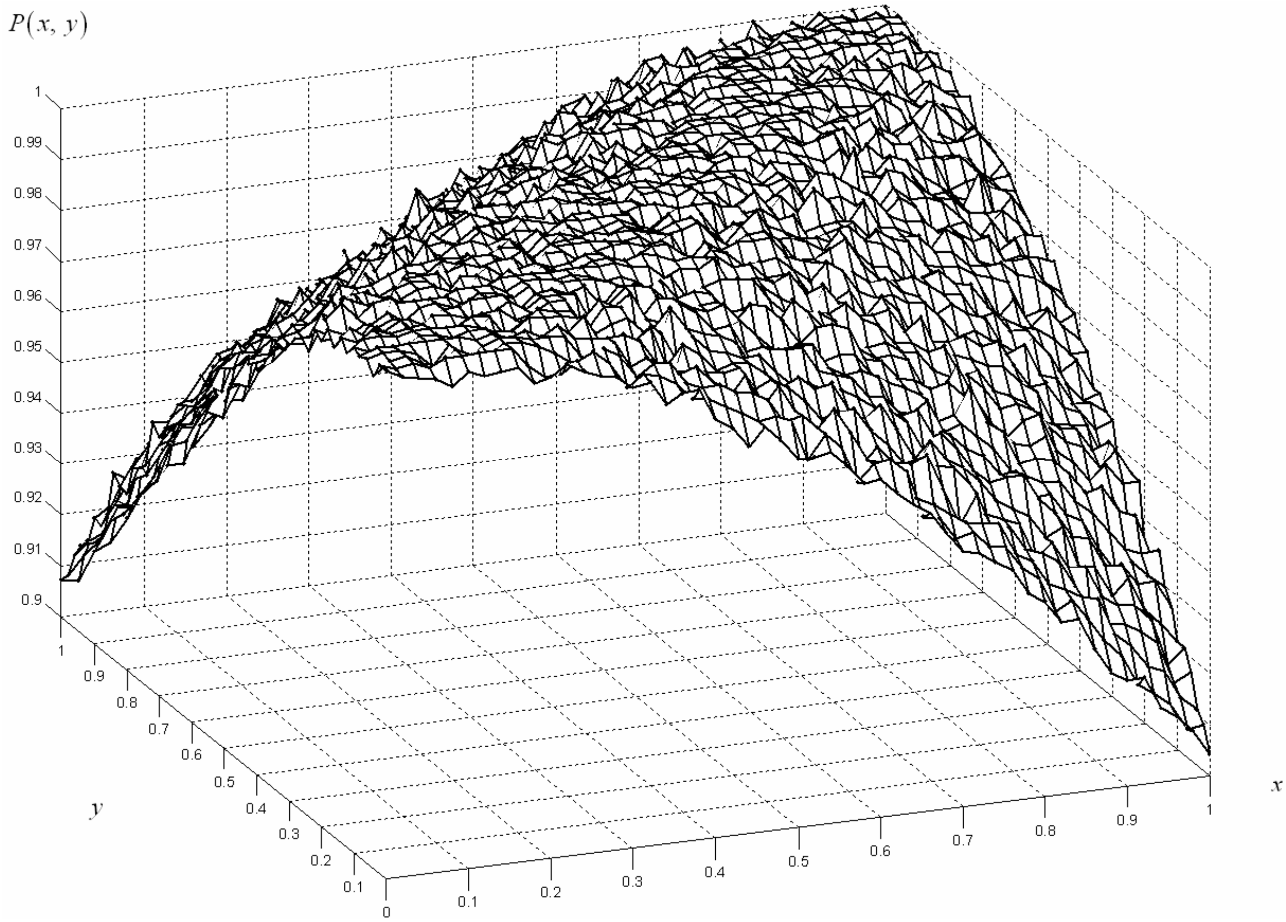


Figure 1. The surface (2) by $\alpha = 0.1$

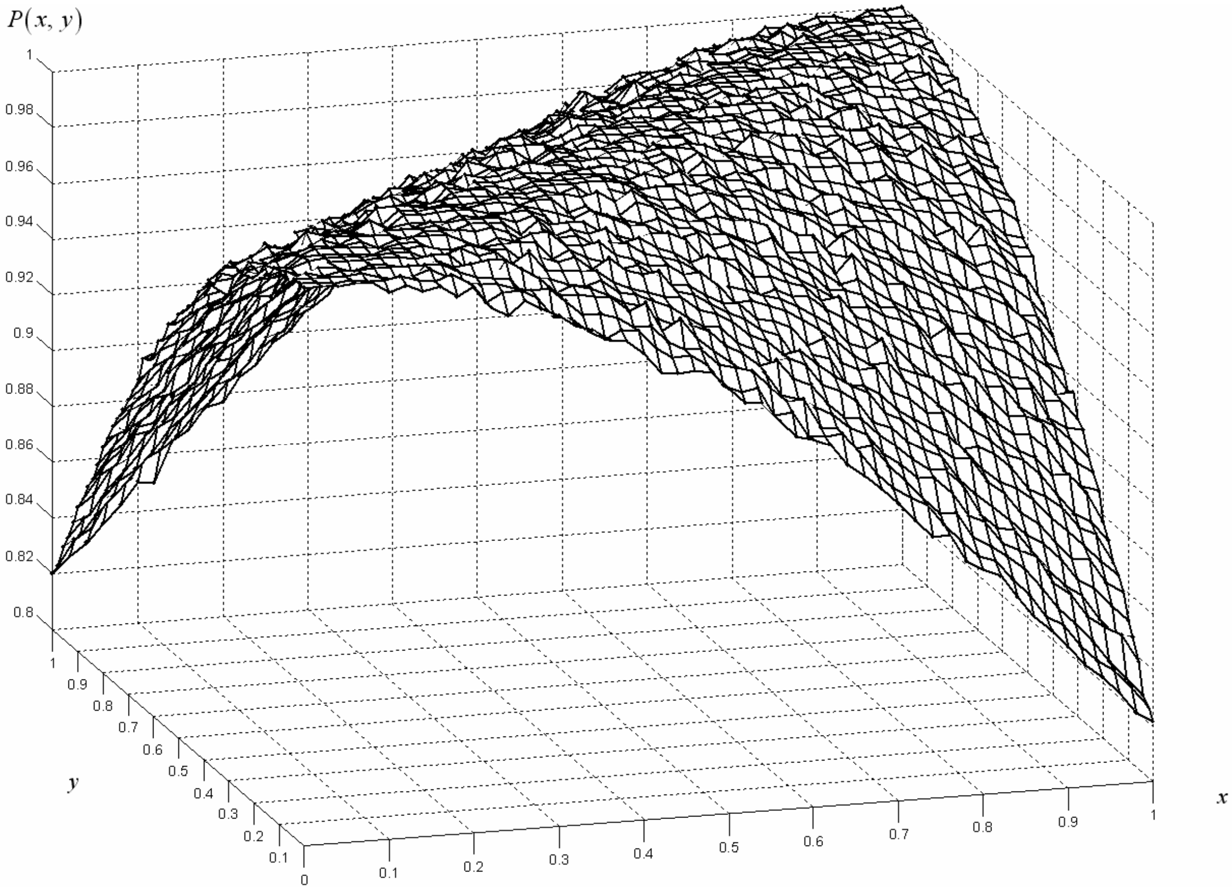


Figure 2. The surface (2) by $\alpha = 0.2$

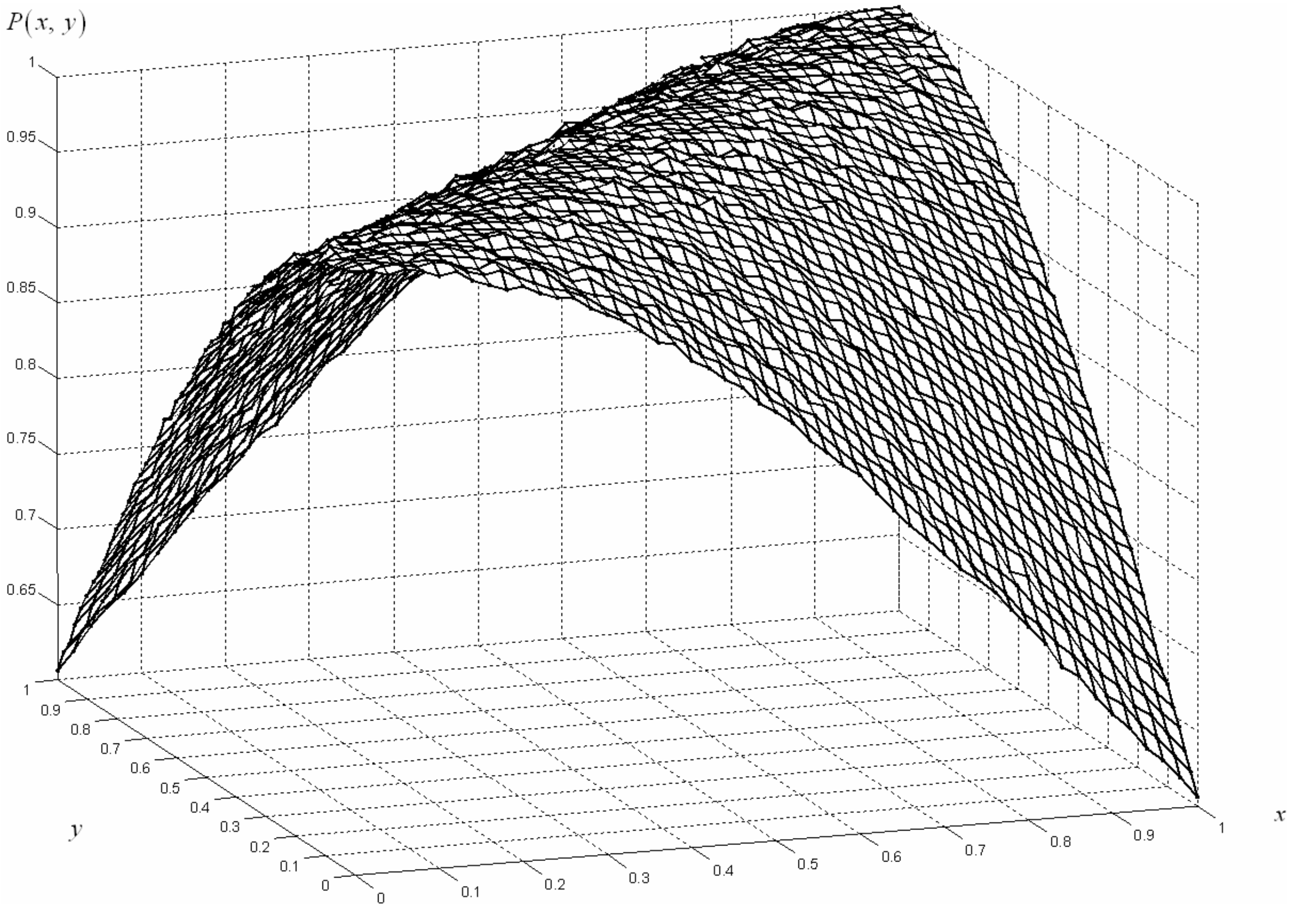


Figure 3. The surface (2) by $\alpha = 0.5$

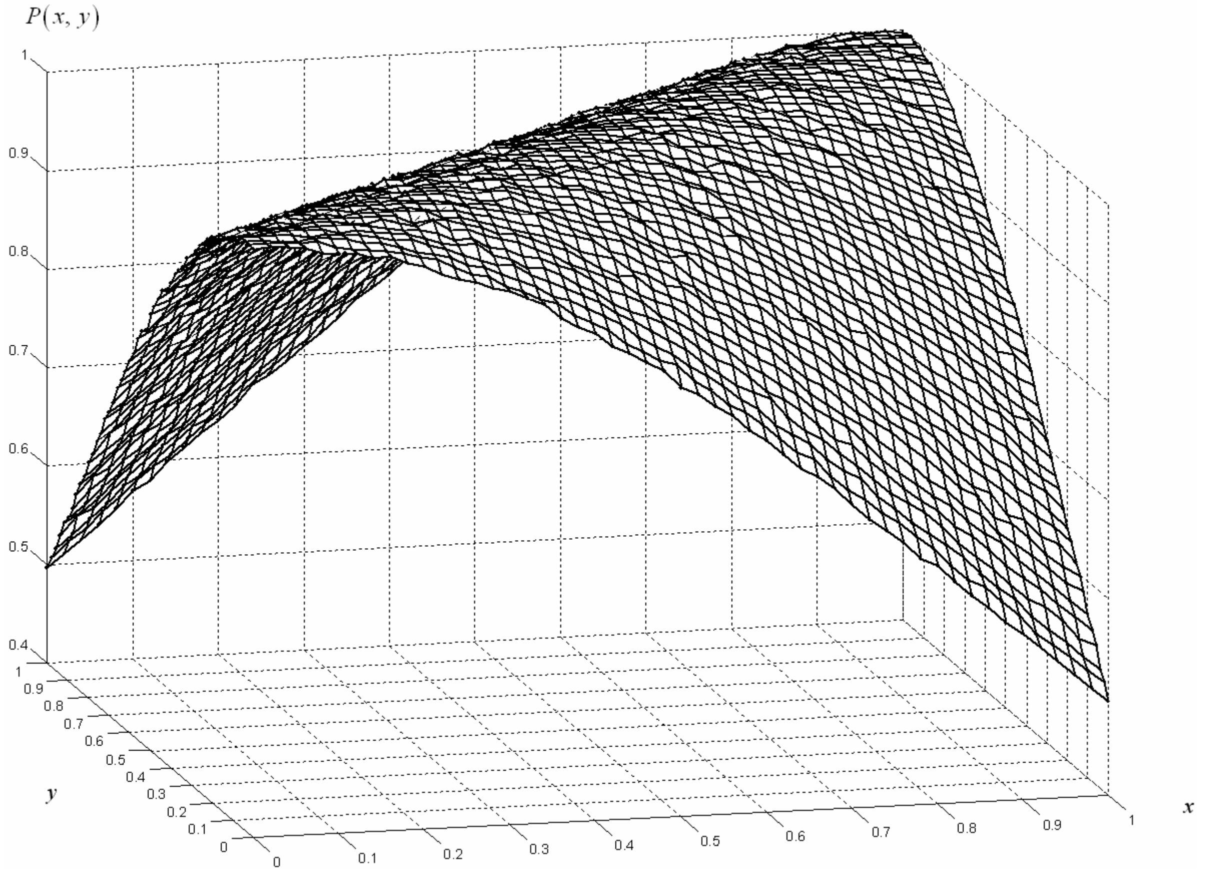


Figure 4. The surface (2) by $\alpha = 0.7$

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

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

and its second derivative

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

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

$$2\alpha(x-y)^2 - 1 \geqslant 0 \tag{10}$$

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

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

and

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

The optimal game value

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

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

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

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

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

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

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

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

$$\exp(-\alpha x^2) > \exp[-\alpha(x-1)^2] \quad (17)$$

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

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

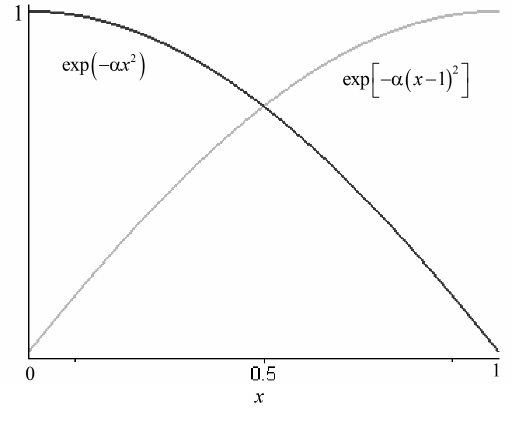


Figure 5. Relation between two exponents in the left side of the inequality (16)

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

$$\frac{\exp\left(-\frac{\alpha}{4}\right)-\exp\left[-\alpha\left(x-1\right)^2\right]}{\exp\left(-\alpha x^2\right)-\exp\left[-\alpha\left(x-1\right)^2\right]}$$

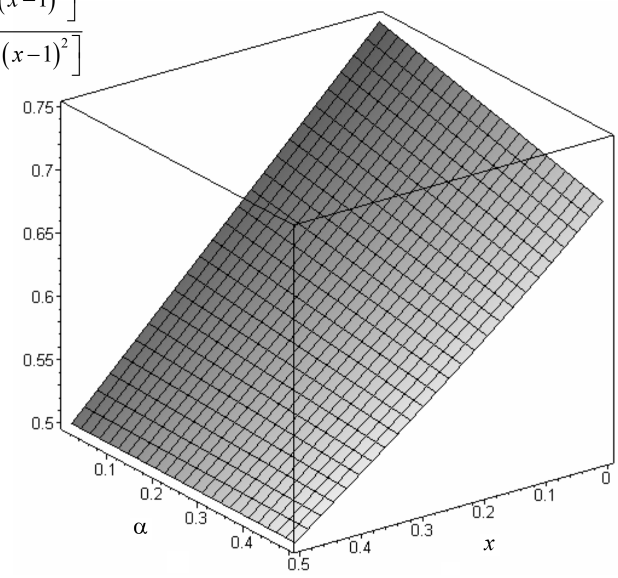


Figure 6. The right side of the inequality (18), that imaged as a surface

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

$$=\lim_{\substack{x\to\frac{1}{2}-\varepsilon\\ \varepsilon>0}}\left(\frac{2\alpha\left(x-1\right)\exp\left[-\alpha\left(x-1\right)^2\right]}{-2\alpha x\exp\left(-\alpha x^2\right)+2\alpha\left(x-1\right)\exp\left[-\alpha\left(x-1\right)^2\right]}\right)=\lim_{\substack{x\to\frac{1}{2}-\varepsilon\\ \varepsilon>0}}\left(\frac{-\alpha\exp\left(-\frac{\alpha}{4}\right)}{-\alpha\exp\left(-\frac{\alpha}{4}\right)-\alpha\exp\left(-\frac{\alpha}{4}\right)}\right)=\frac{1}{2} \qquad (19)$$

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

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

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

$$\exp\left(-\alpha x^2\right)<\exp\left[-\alpha\left(x-1\right)^2\right] \qquad (21)$$

to the inequality

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

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

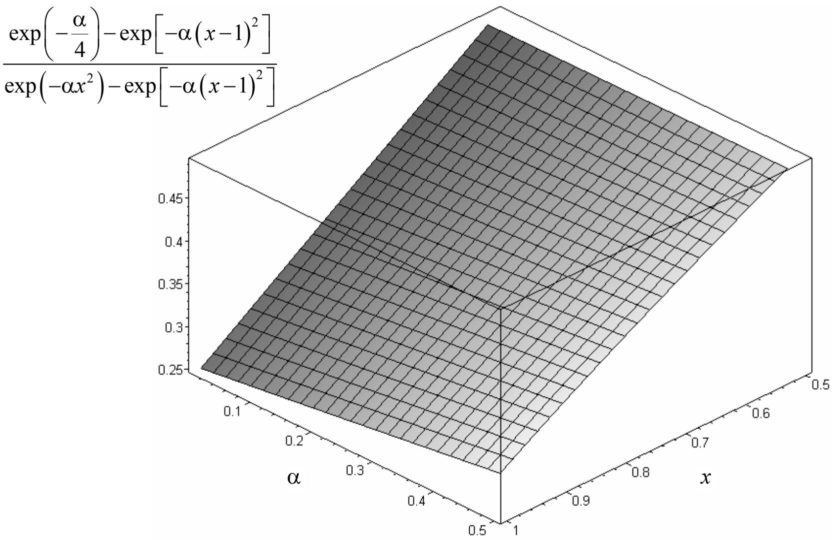


Figure 7. The right side of the inequality (22), that imaged as a surface

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

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

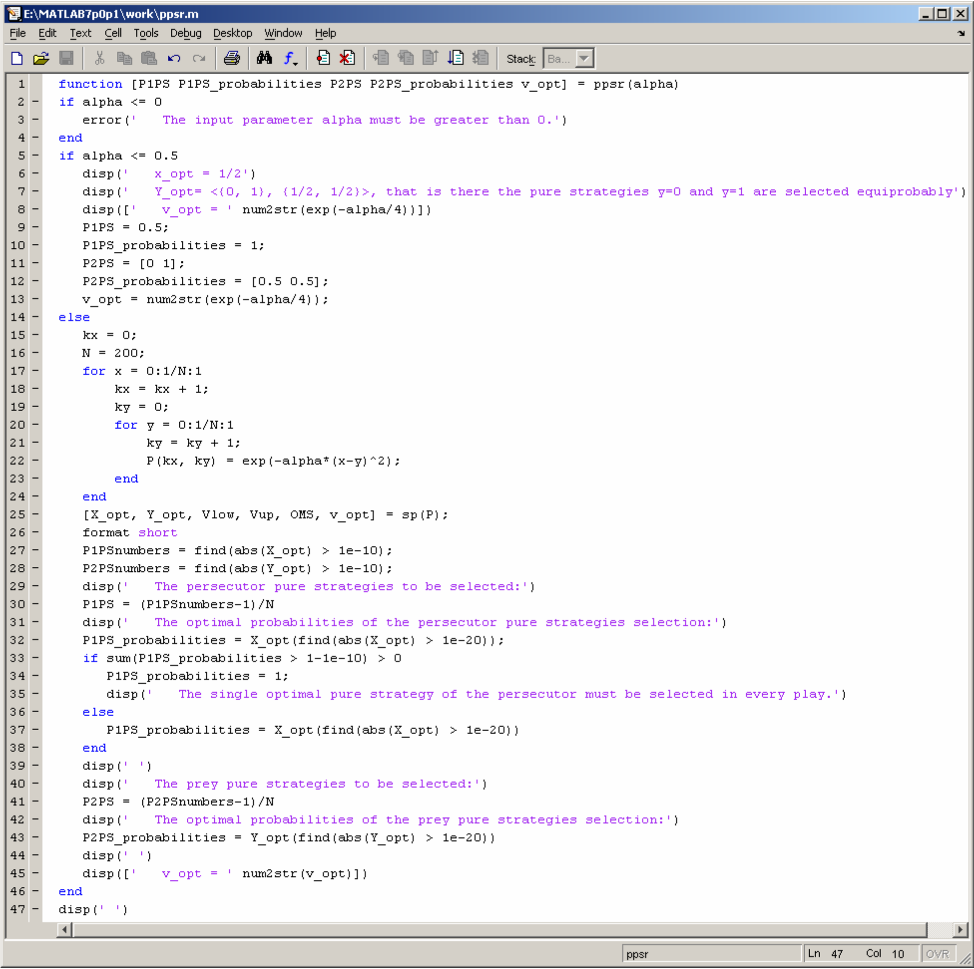
$$\varphi(0) \in \left[\frac{1}{2};\, 1\right] = \Phi_2\,. \tag{24}$$

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

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

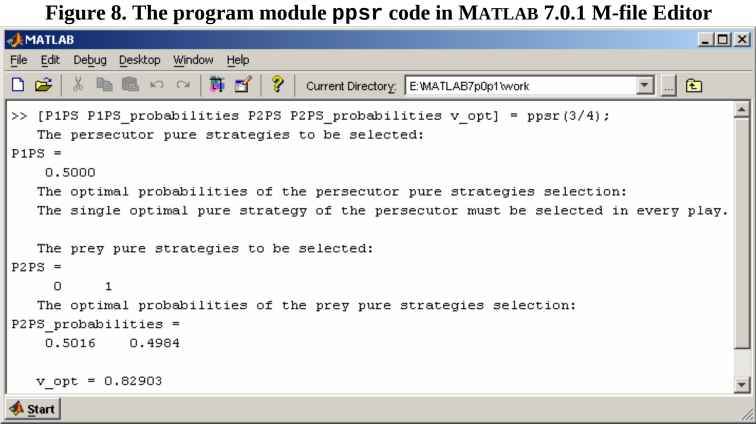
Hence the optimal probability of the pure strategy $y_2 = 1$ selection is $\varphi(1) = \frac{1}{2}$. Then the game with the defined on the unit square $D_P = [0;\, 1] \times [0;\, 1]$ kernel (1) by the parameter $\alpha \in \left(0;\, \frac{1}{2}\right]$ is solved in the single optimal strategy of the 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).



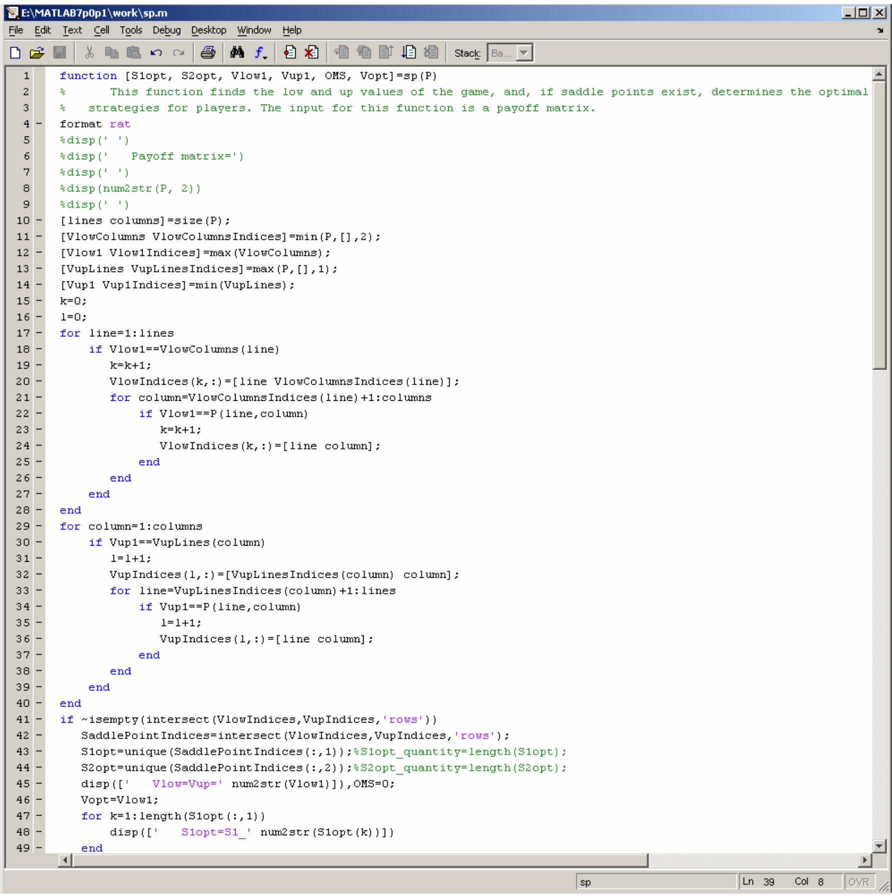
```
function [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(alpha)
if alpha <= 0
   error('   The input parameter alpha must be greater than 0.')
end
if alpha <= 0.5
   disp('   x_opt = 1/2')
   disp('   Y_opt= <(0, 1), (1/2, 1/2)>, that is there the pure strategies y=0 and y=1 are selected equiprobably')
   disp(['   v_opt = ' num2str(exp(-alpha/4))])
   P1PS = 0.5;
   P1PS_probabilities = 1;
   P2PS = [0 1];
   P2PS_probabilities = [0.5 0.5];
   v_opt = num2str(exp(-alpha/4));
else
   kx = 0;
   N = 200;
   for x = 0:1/N:1
       kx = kx + 1;
       ky = 0;
       for y = 0:1/N:1
           ky = ky + 1;
           P(kx, ky) = exp(-alpha*(x-y)^2);
       end
   end
   [X_opt, Y_opt, Vlow, Vup, OMS, v_opt] = sp(P);
   format short
   P1PSnumbers = find(abs(X_opt) > 1e-10);
   P2PSnumbers = find(abs(Y_opt) > 1e-10);
   disp('   The persecutor pure strategies to be selected:')
   P1PS = (P1PSnumbers-1)/N
   disp('   The optimal probabilities of the persecutor pure strategies selection:')
   P1PS_probabilities = X_opt(find(abs(X_opt) > 1e-20));
   if sum(P1PS_probabilities > 1-1e-10) > 0
      P1PS_probabilities = 1;
      disp('   The single optimal pure strategy of the persecutor must be selected in every play.')
   else
      P1PS_probabilities = X_opt(find(abs(X_opt) > 1e-20))
   end
   disp(' ')
   disp('   The prey pure strategies to be selected:')
   P2PS = (P2PSnumbers-1)/N
   disp('   The optimal probabilities of the prey pure strategies selection:')
   P2PS_probabilities = Y_opt(find(abs(Y_opt) > 1e-20))
   disp(' ')
   disp(['   v_opt = ' num2str(v_opt)])
end
disp(' ')
```

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



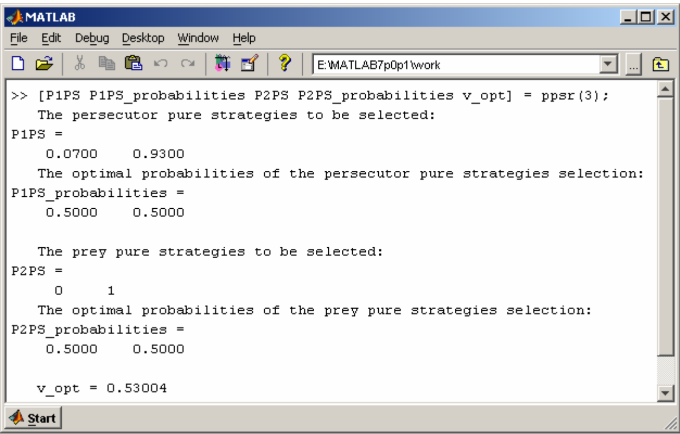
```
>> [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr(3/4);
   The persecutor pure strategies to be selected:
P1PS =
    0.5000
   The optimal probabilities of the persecutor pure strategies selection:
   The single optimal pure strategy of the persecutor must be selected in every play.

   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 $\alpha = 0.75$



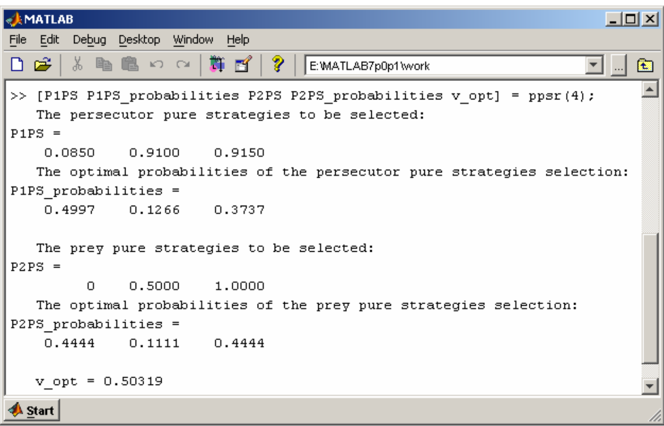

```
function [S1opt, S2opt, Vlow1, Vup1, OMS, Vopt]=sp(P)
%       This function finds the low and up values of the game, and, if saddle points exist, determines the optimal
%    strategies for players. The input for this function is a payoff matrix.
format rat
%disp(' ')
%disp('    Payoff matrix=')
%disp(' ')
%disp(num2str(P, 2))
%disp(' ')
[lines columns]=size(P);
[VlowColumns VlowColumnsIndices]=min(P,[],2);
[Vlow1 Vlow1Indices]=max(VlowColumns);
[VupLines VupLinesIndices]=max(P,[],1);
[Vup1 Vup1Indices]=min(VupLines);
k=0;
l=0;
for line=1:lines
    if Vlow1==VlowColumns(line)
       k=k+1;
       VlowIndices(k,:)=[line VlowColumnsIndices(line)];
       for column=VlowColumnsIndices(line)+1:columns
           if Vlow1==P(line,column)
              k=k+1;
              VlowIndices(k,:)=[line column];
           end
       end
    end
end
for column=1:columns
    if Vup1==VupLines(column)
       l=l+1;
       VupIndices(l,:)=[VupLinesIndices(column) column];
       for line=VupLinesIndices(column)+1:lines
           if Vup1==P(line,column)
              l=l+1;
              VupIndices(l,:)=[line column];
           end
       end
    end
end
if ~isempty(intersect(VlowIndices,VupIndices,'rows'))
   SaddlePointIndices=intersect(VlowIndices,VupIndices,'rows');
   S1opt=unique(SaddlePointIndices(:,1));%S1opt_quantity=length(S1opt);
   S2opt=unique(SaddlePointIndices(:,2));%S2opt_quantity=length(S2opt);
   disp(['    Vlow=Vup=' num2str(Vlow1)]),OMS=0;
   Vopt=Vlow1;
   for k=1:length(S1opt(:,1))
       disp(['    S1opt=S1_' num2str(S1opt(k))])
   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

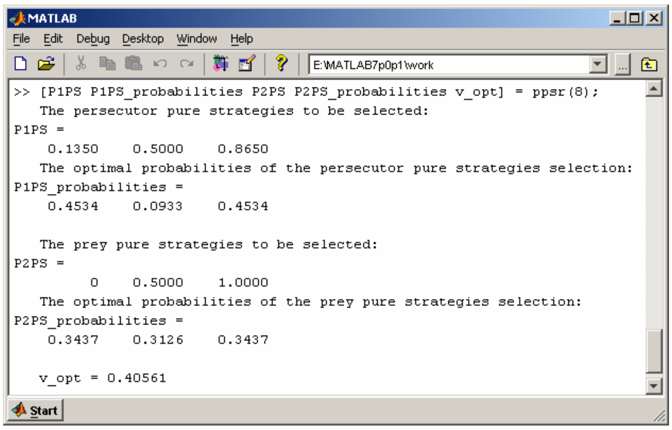


Figure 13. Symmetrization of the three pure strategies and their selection probabilities by $\alpha = 8$

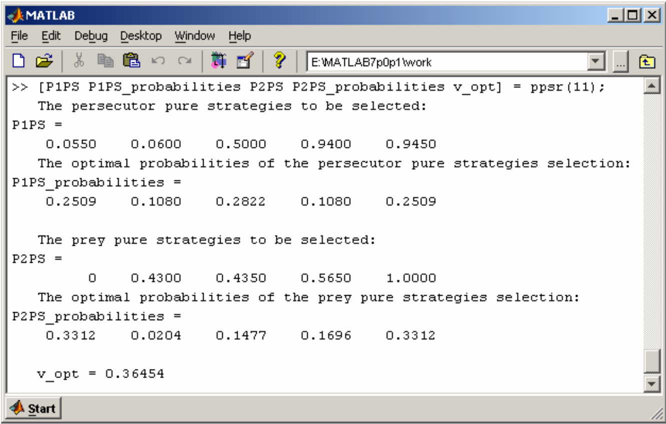


Figure 14. Appearance of the five pure strategies for their selection by $\alpha = 11$

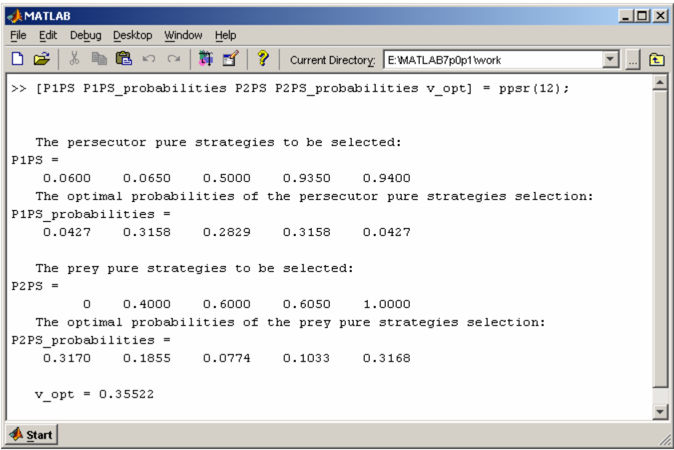


Figure 15. Some asymmetry of the five pure strategies of the prey by $\alpha = 12$

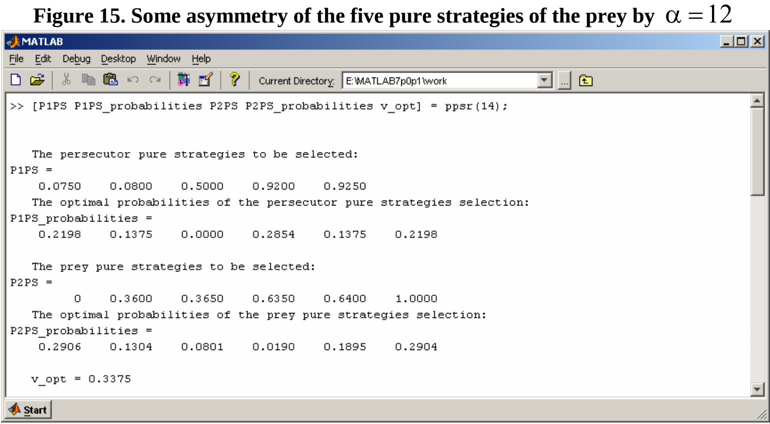


Figure 16. By $\alpha = 14$ the persecutor selects its five pure strategies (the zero probability must be expunged), and the prey selects its six pure strategies

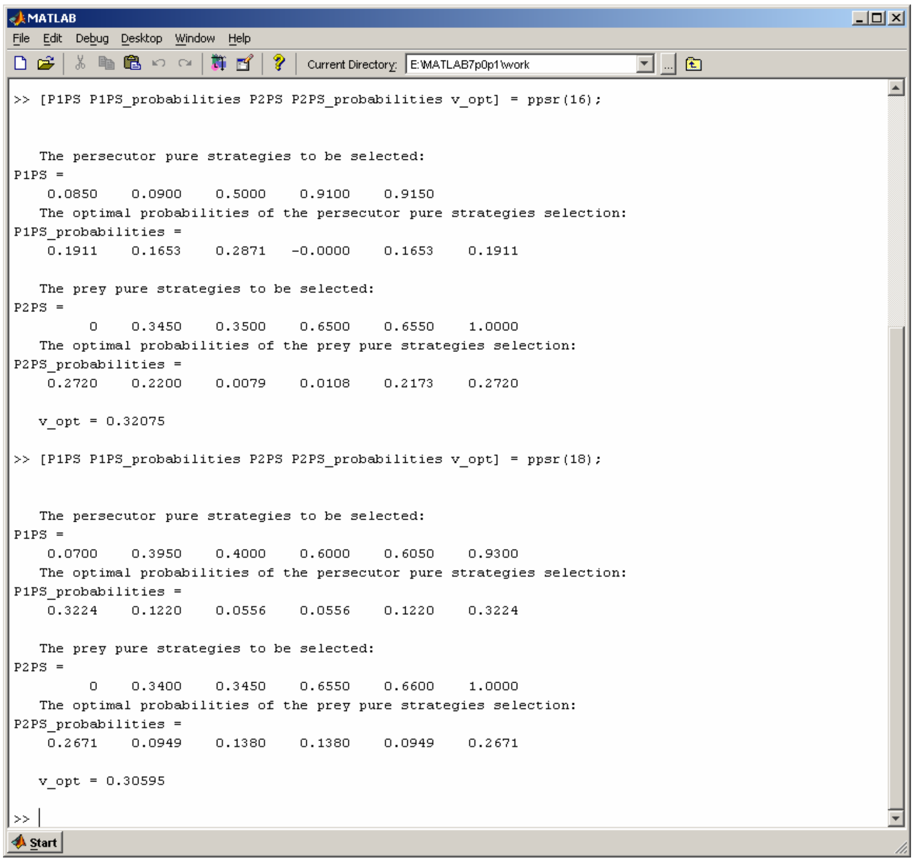


Figure 17. Pure strategies and their probabilities by $\alpha = 16$ and $\alpha = 18$

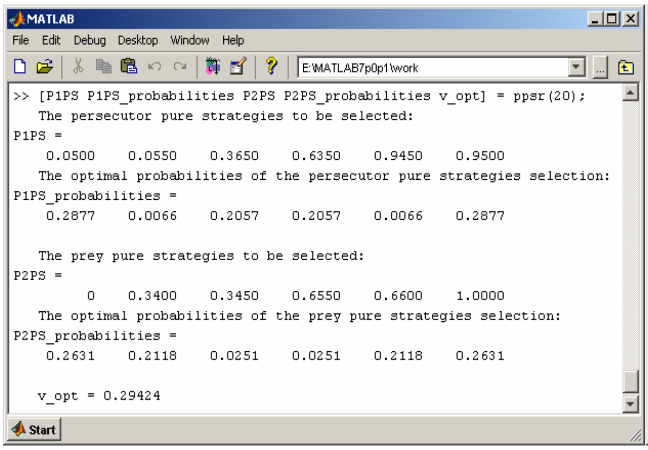


Figure 18. Symmetrization of the six pure strategies for their selection by $\alpha = 20$

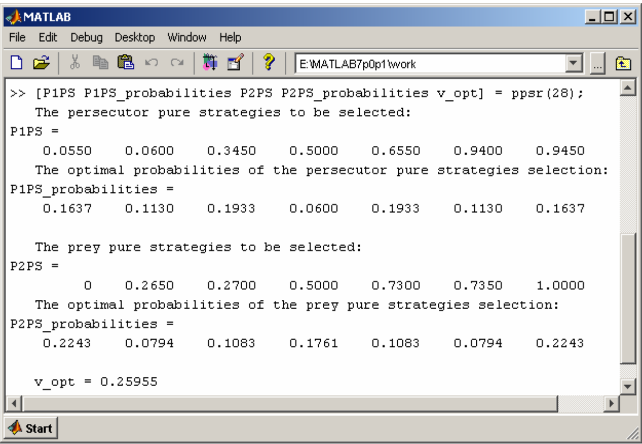


Figure 19. Appearance of the seven pure strategies for their selection by $\alpha = 28$

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

$$\mathbf{V} = \begin{bmatrix} v_1 & v_2 & \dots & v_G \end{bmatrix} \tag{26}$$

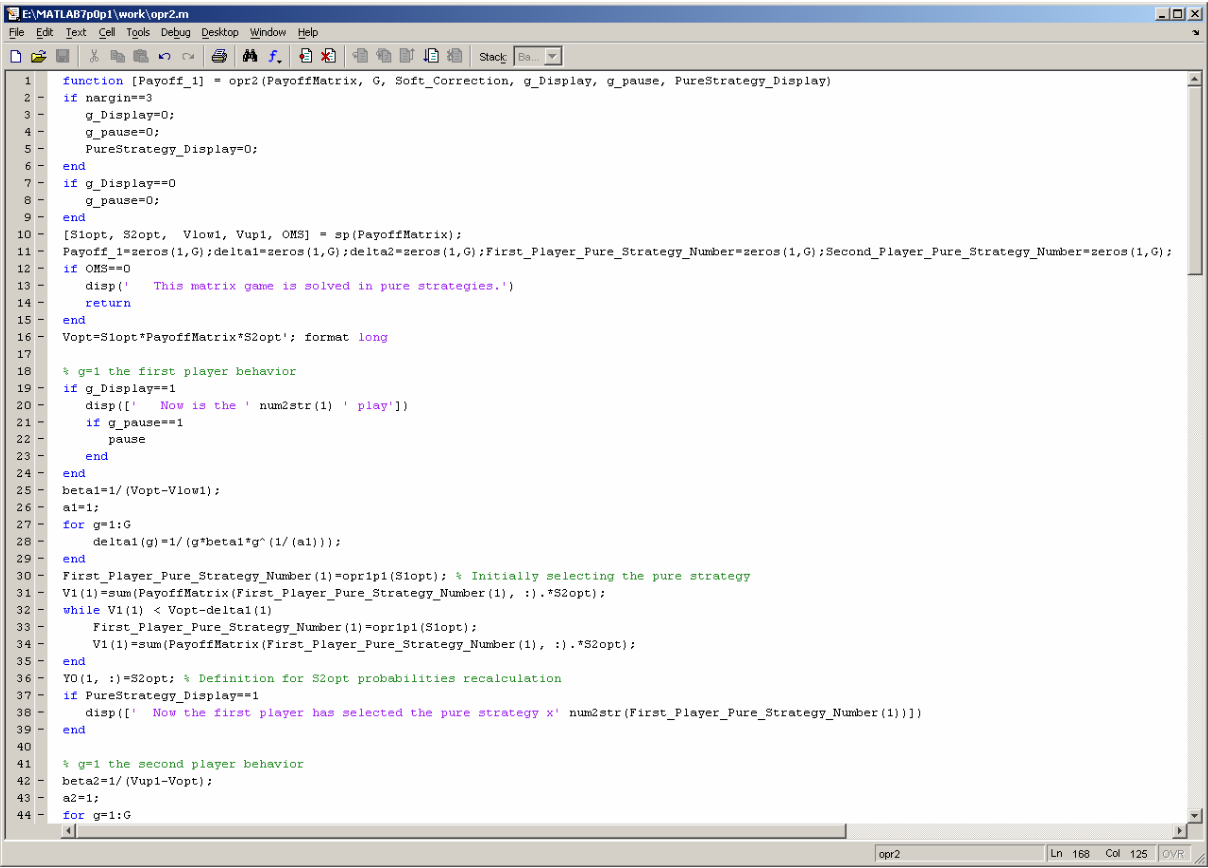
of the persecutor payoffs $\left\{v_j\right\}_{j=1}^{G}$, where v_j is the kernel (2) value in the shot j by $j=\overline{1, G}$, and actually their average is converging to v_{opt} as the number of the shots is tending to infinity:

$$\lim_{j\to\infty}\frac{1}{G}\sum_{j=1}^{G}v_j=\frac{1}{G}\lim_{j\to\infty}\sum_{j=1}^{G}v_j=v_{\text{opt}} \tag{27}$$

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

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

of the averaged persecutor payoff.




```
function [Payoff_1] = opr2(PayoffMatrix, G, Soft_Correction, g_Display, g_pause, PureStrategy_Display)
if nargin==3
   g_Display=0;
   g_pause=0;
   PureStrategy_Display=0;
end
if g_Display==0
   g_pause=0;
end
[S1opt, S2opt,  Vlow1, Vup1, OMS] = sp(PayoffMatrix);
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);
if OMS==0
   disp('   This matrix game is solved in pure strategies.')
   return
end
Vopt=S1opt*PayoffMatrix*S2opt'; format long

% g=1 the first player behavior
if g_Display==1
   disp(['   Now is the ' num2str(1) ' play'])
   if g_pause==1
      pause
   end
end
beta1=1/(Vopt-Vlow1);
a1=1;
for g=1:G
    delta1(g)=1/(g*beta1*g^(1/(a1)));
end
First_Player_Pure_Strategy_Number(1)=opr1p1(S1opt); % Initially selecting the pure strategy
V1(1)=sum(PayoffMatrix(First_Player_Pure_Strategy_Number(1), :).*S2opt);
while V1(1) < Vopt-delta1(1)
    First_Player_Pure_Strategy_Number(1)=opr1p1(S1opt);
    V1(1)=sum(PayoffMatrix(First_Player_Pure_Strategy_Number(1), :).*S2opt);
end
Y0(1, :)=S2opt; % Definition for S2opt probabilities recalculation
if PureStrategy_Display==1
   disp(['  Now the first player has selected the pure strategy x' num2str(First_Player_Pure_Strategy_Number(1))])
end

% g=1 the second player behavior
beta2=1/(Vup1-Vopt);
a2=1;
for g=1:G
```

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

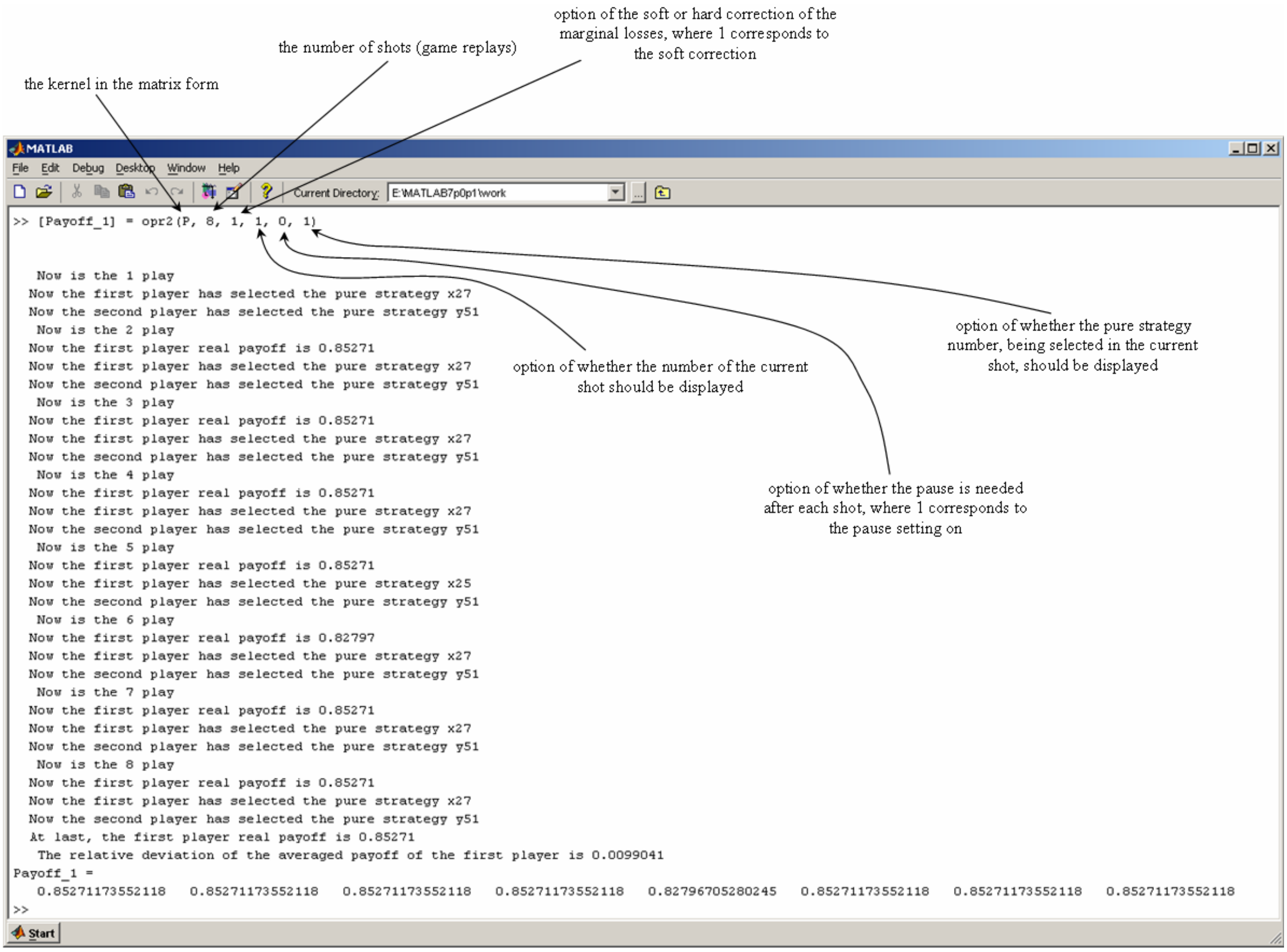


Figure 21. The example of running the program module `opr2` code with $G = 8$ and some kernel (2) in the matrix form

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.

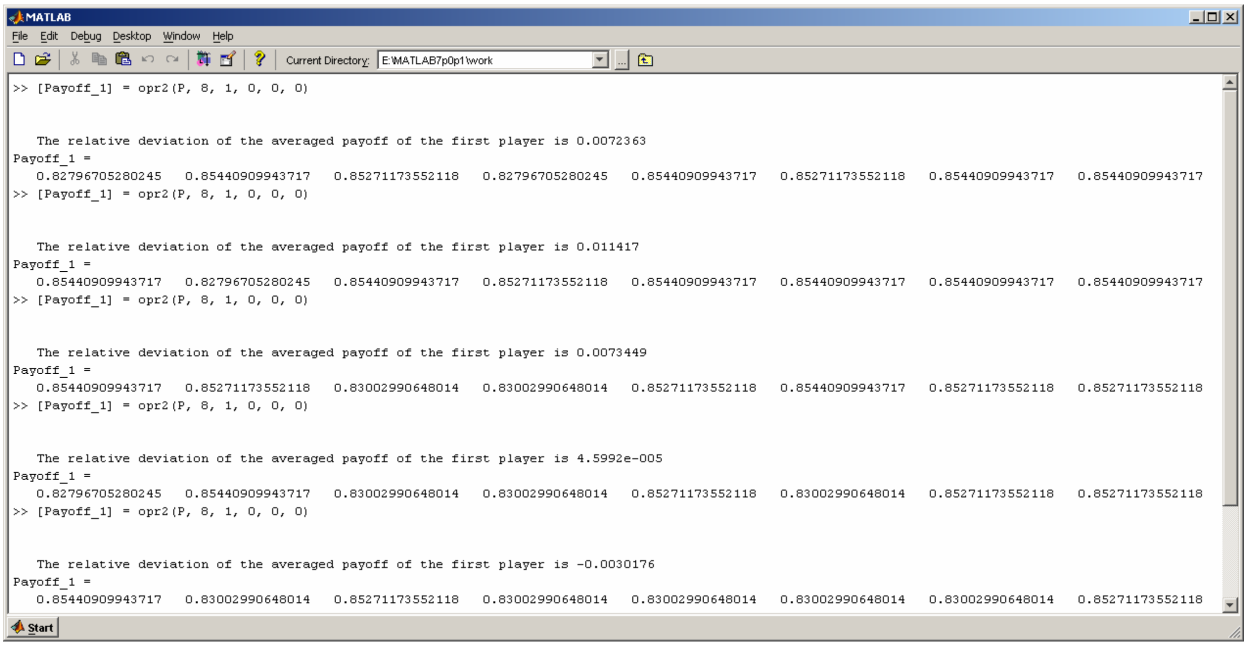


Figure 22. The vectors (26) by $G = 8$ and the same kernel (2) in the matrix form

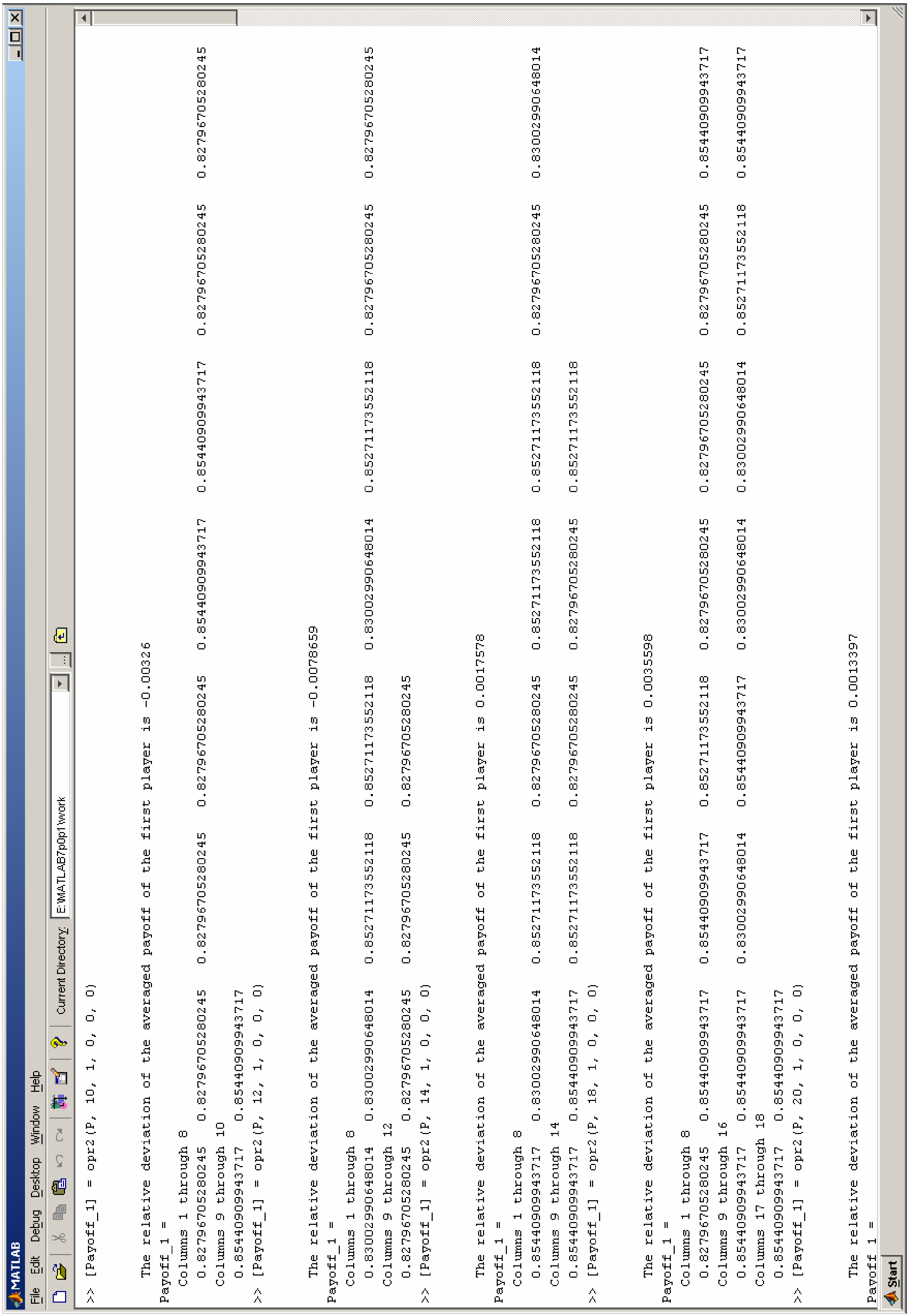


Figure 23. The vectors (26) by $G \in \{10, 12, 14, 18\}$ **and the same kernel (2) in the matrix form**

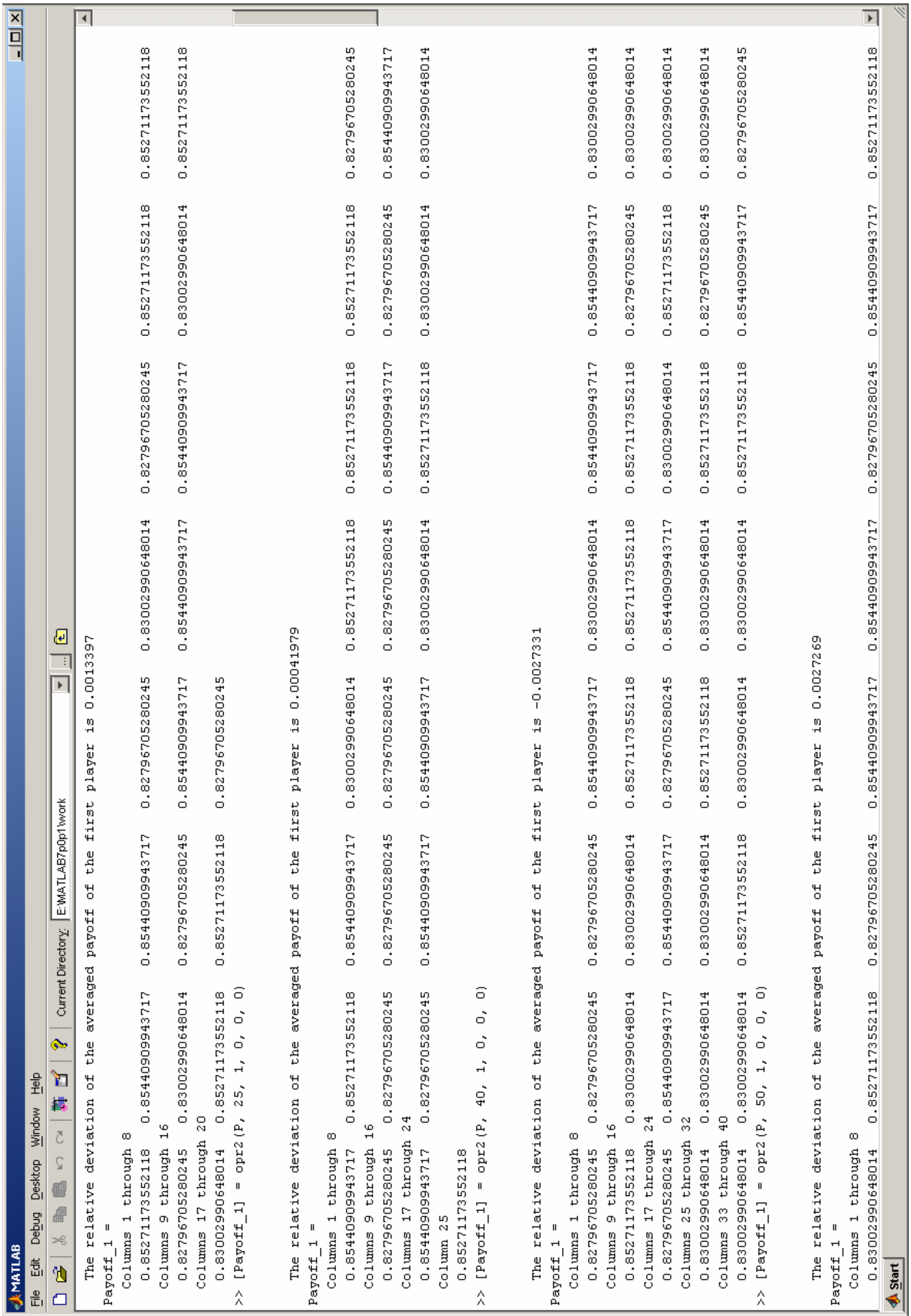


Figure 24. The vectors (26) by $G \in \{20, 25, 40\}$ and the same kernel (2) in the matrix form

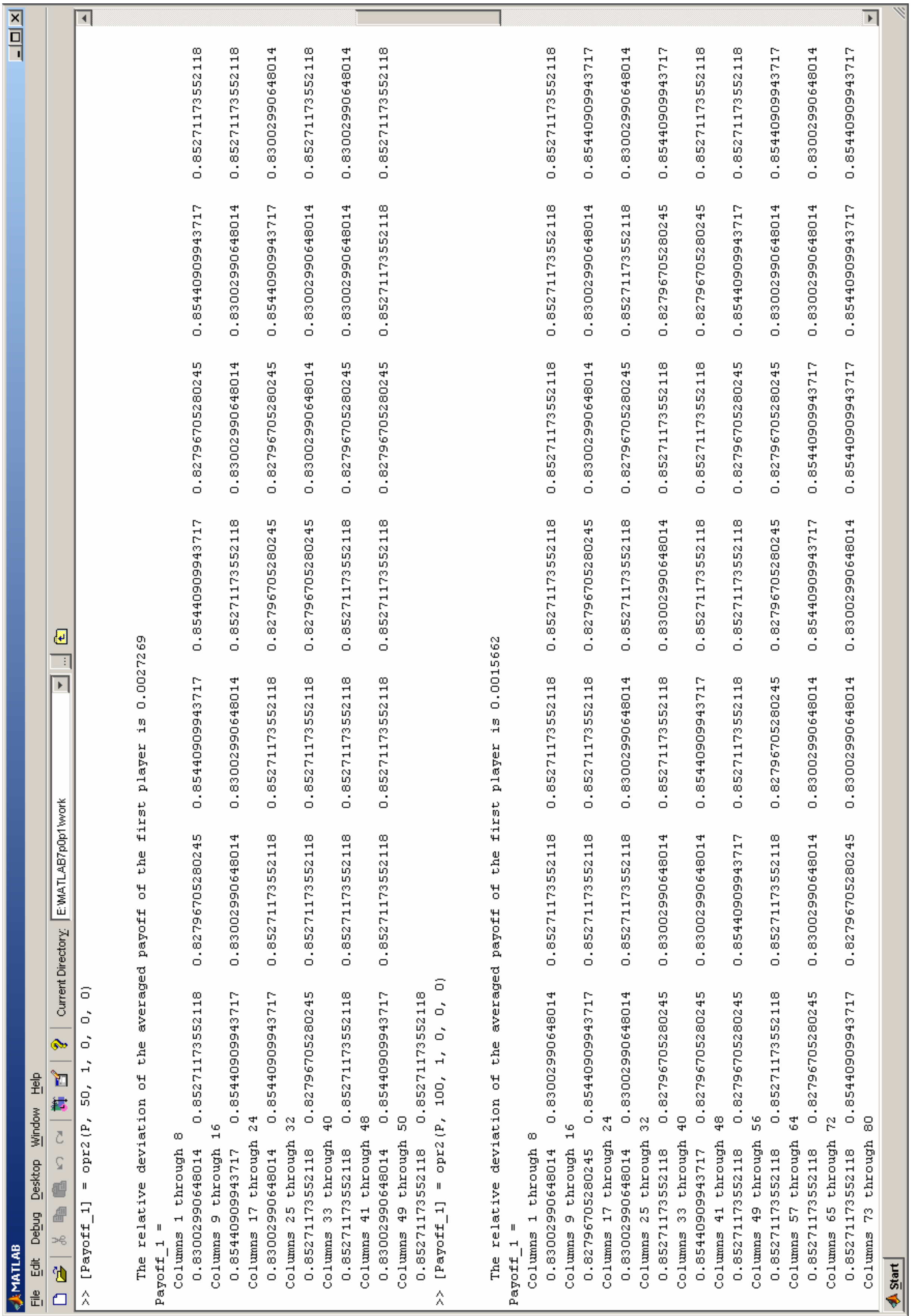


Figure 25. The vector (26) by $G = 50$ and the same kernel (2) in the matrix form, and the first portion of the vector (26) by $G = 100$

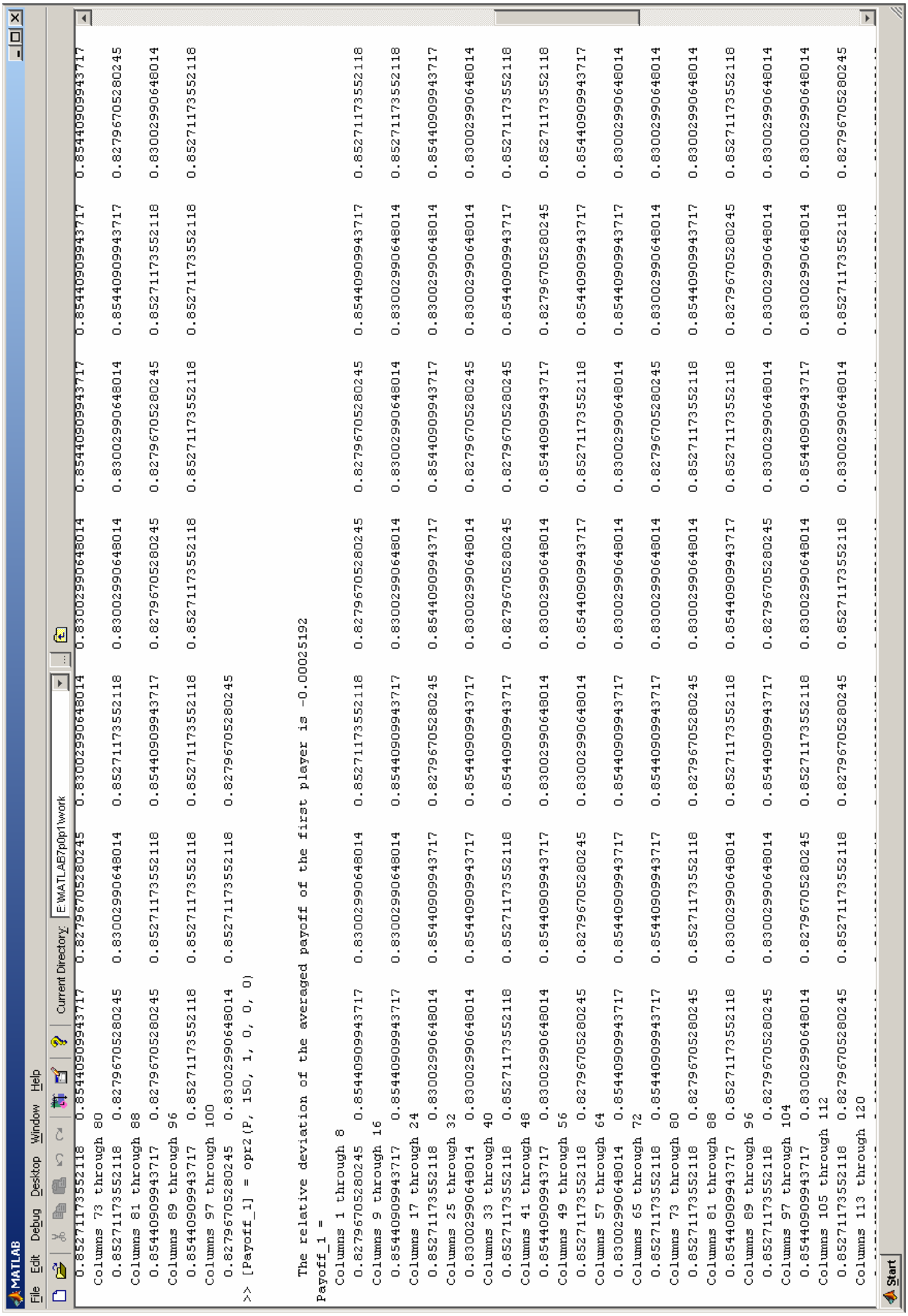


Figure 26. The second portion of the vector (26) by $G = 100$ and the same kernel (2) in the matrix form, and the first portion of the vector (26) by $G = 150$

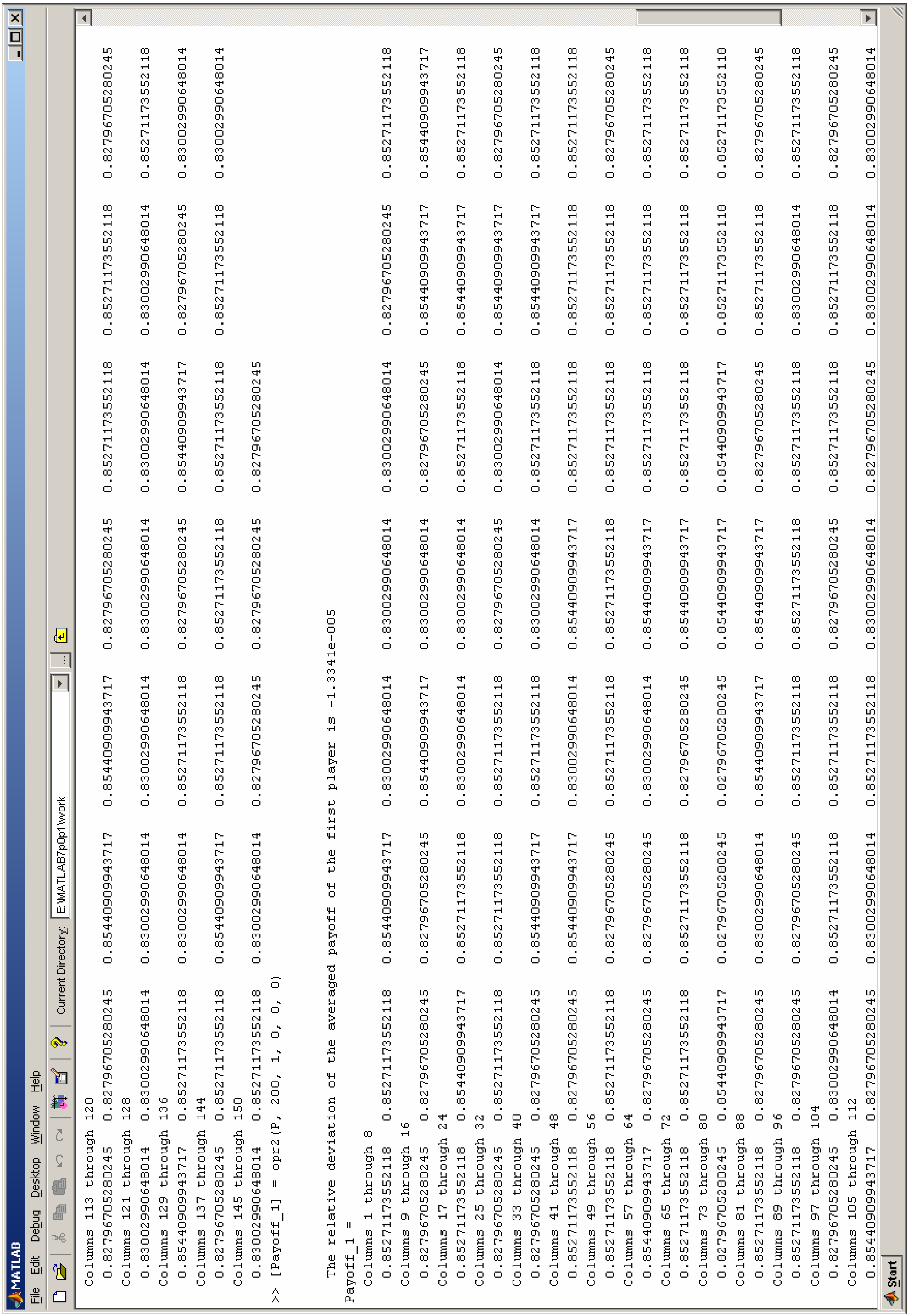
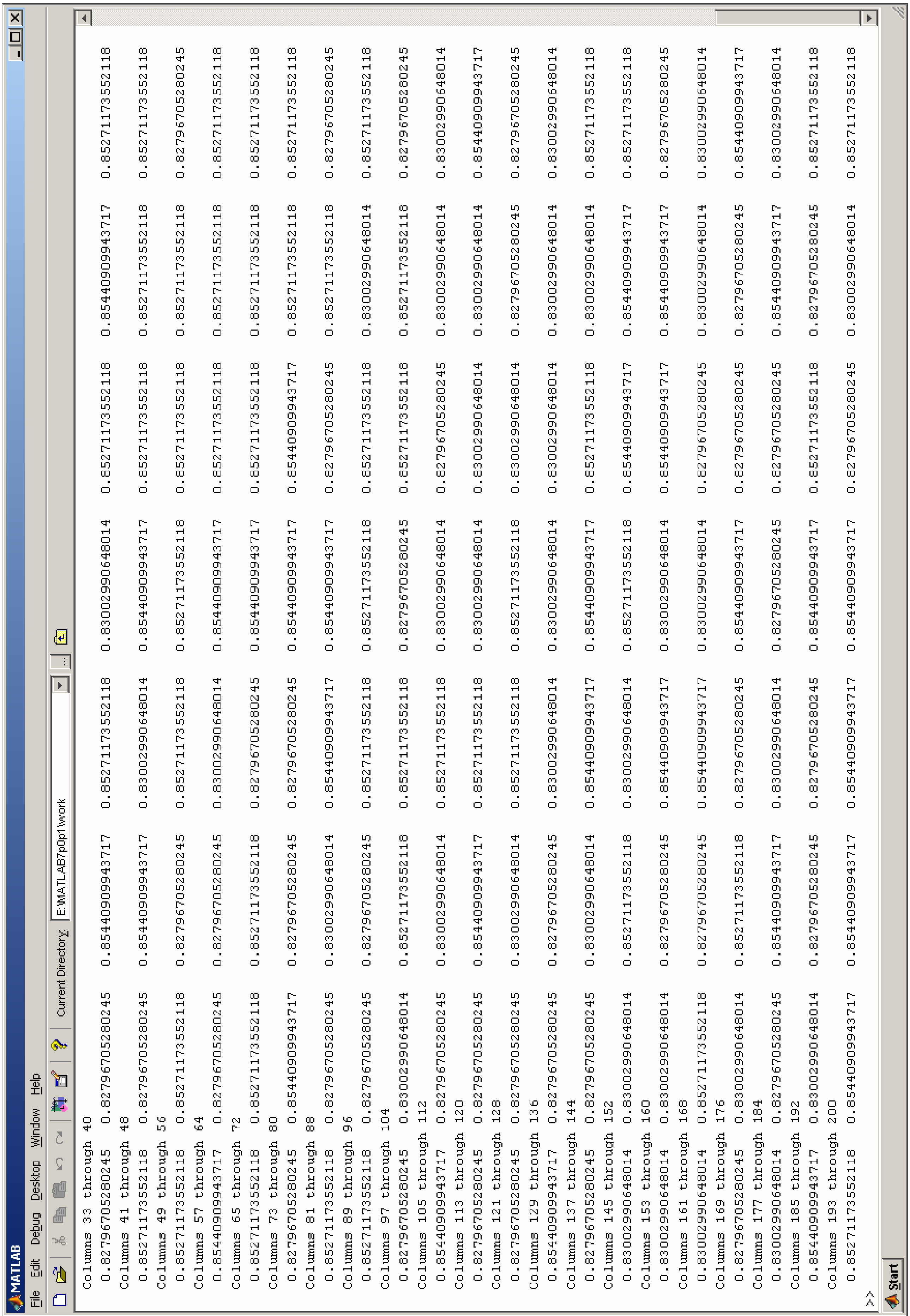


Figure 27. The second portion of the vector (26) by $G = 150$ and the same kernel (2) in the matrix form, and the first portion of the vector (26) by $G = 200$



```
MATLAB
File  Edit  Debug  Desktop  Window  Help
Current Directory: E:\MATLAB7p0p1\work
  Columns 33 through 40
   0.82796705280245   0.82796705280245   0.85440909943717   0.85271173552118   0.83002990648014   0.85271173552118   0.85440909943717   0.85271173552118
  Columns 41 through 48
   0.85271173552118   0.82796705280245   0.85440909943717   0.83002990648014   0.85440909943717   0.85271173552118   0.85271173552118   0.85271173552118
  Columns 49 through 56
   0.85271173552118   0.85271173552118   0.82796705280245   0.85271173552118   0.85271173552118   0.85271173552118   0.85271173552118   0.82796705280245
  Columns 57 through 64
   0.85440909943717   0.82796705280245   0.82796705280245   0.83002990648014   0.85440909943717   0.85271173552118   0.85271173552118   0.85271173552118
  Columns 65 through 72
   0.85271173552118   0.85271173552118   0.85271173552118   0.82796705280245   0.85440909943717   0.85271173552118   0.85271173552118   0.85271173552118
  Columns 73 through 80
   0.82796705280245   0.85440909943717   0.82796705280245   0.82796705280245   0.85440909943717   0.85440909943717   0.85271173552118   0.85271173552118
  Columns 81 through 88
   0.85271173552118   0.82796705280245   0.83002990648014   0.85440909943717   0.85440909943717   0.82796705280245   0.85271173552118   0.82796705280245
  Columns 89 through 96
   0.85271173552118   0.82796705280245   0.82796705280245   0.85271173552118   0.85271173552118   0.85271173552118   0.83002990648014   0.85271173552118
  Columns 97 through 104
   0.82796705280245   0.83002990648014   0.85271173552118   0.85271173552118   0.82796705280245   0.85271173552118   0.85271173552118   0.82796705280245
  Columns 105 through 112
   0.85440909943717   0.82796705280245   0.83002990648014   0.85271173552118   0.83002990648014   0.82796705280245   0.83002990648014   0.83002990648014
  Columns 113 through 120
   0.82796705280245   0.82796705280245   0.85440909943717   0.85271173552118   0.83002990648014   0.83002990648014   0.83002990648014   0.85440909943717
  Columns 121 through 128
   0.82796705280245   0.82796705280245   0.83002990648014   0.85271173552118   0.85271173552118   0.83002990648014   0.82796705280245   0.82796705280245
  Columns 129 through 136
   0.85440909943717   0.82796705280245   0.82796705280245   0.83002990648014   0.83002990648014   0.83002990648014   0.83002990648014   0.83002990648014
  Columns 137 through 144
   0.82796705280245   0.82796705280245   0.83002990648014   0.85440909943717   0.85440909943717   0.85271173552118   0.83002990648014   0.85271173552118
  Columns 145 through 152
   0.83002990648014   0.83002990648014   0.85271173552118   0.83002990648014   0.85271173552118   0.85440909943717   0.85440909943717   0.85271173552118
  Columns 153 through 160
   0.83002990648014   0.83002990648014   0.82796705280245   0.85440909943717   0.83002990648014   0.85440909943717   0.85440909943717   0.82796705280245
  Columns 161 through 168
   0.83002990648014   0.85271173552118   0.82796705280245   0.85440909943717   0.83002990648014   0.82796705280245   0.83002990648014   0.83002990648014
  Columns 169 through 176
   0.82796705280245   0.83002990648014   0.85271173552118   0.82796705280245   0.85440909943717   0.82796705280245   0.82796705280245   0.85440909943717
  Columns 177 through 184
   0.83002990648014   0.82796705280245   0.85440909943717   0.83002990648014   0.82796705280245   0.82796705280245   0.85440909943717   0.83002990648014
  Columns 185 through 192
   0.85440909943717   0.83002990648014   0.82796705280245   0.82796705280245   0.85440909943717   0.85271173552118   0.82796705280245   0.85271173552118
  Columns 193 through 200
   0.85271173552118   0.85440909943717   0.85440909943717   0.85440909943717   0.85440909943717   0.82796705280245   0.83002990648014   0.85271173552118
>>
Start
```

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

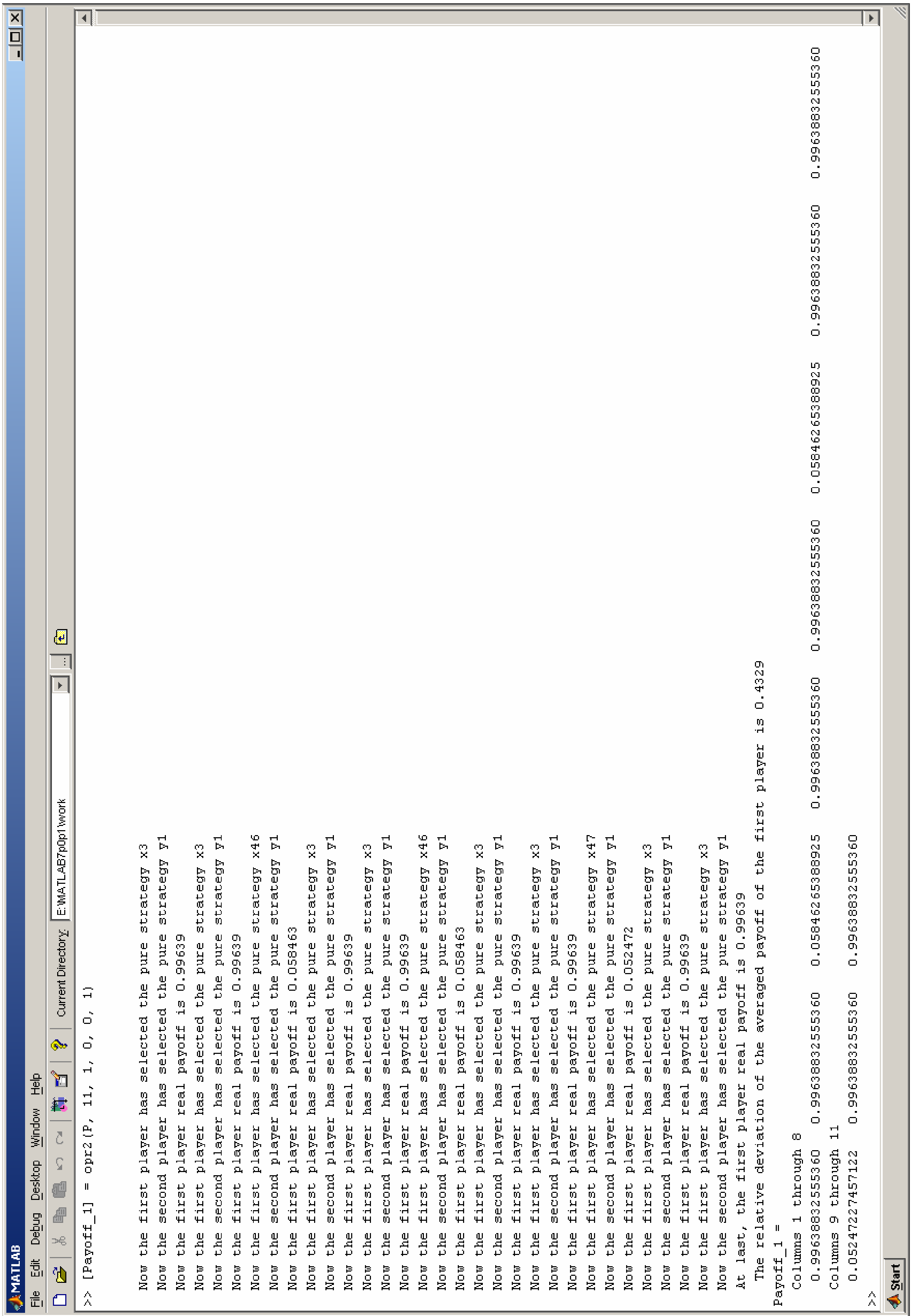


Figure 29. The hints of what pure strategy to be selected and the vector (26) by $G = 11$ and another kernel (2) in the matrix form

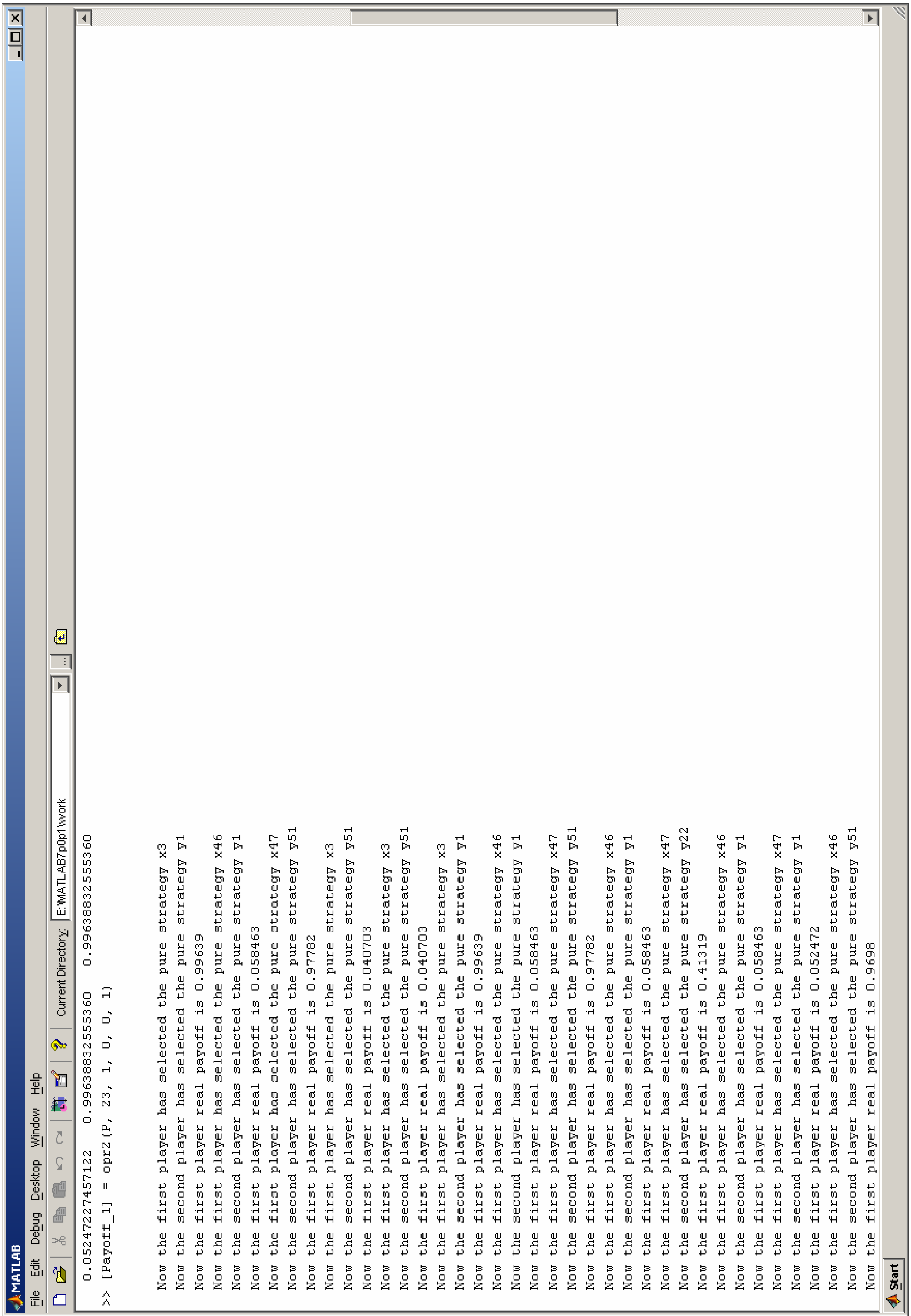


Figure 30. The first portion of the hints of what pure strategy to be selected and the vector (26) by $G = 23$ and another kernel (2) in the matrix form

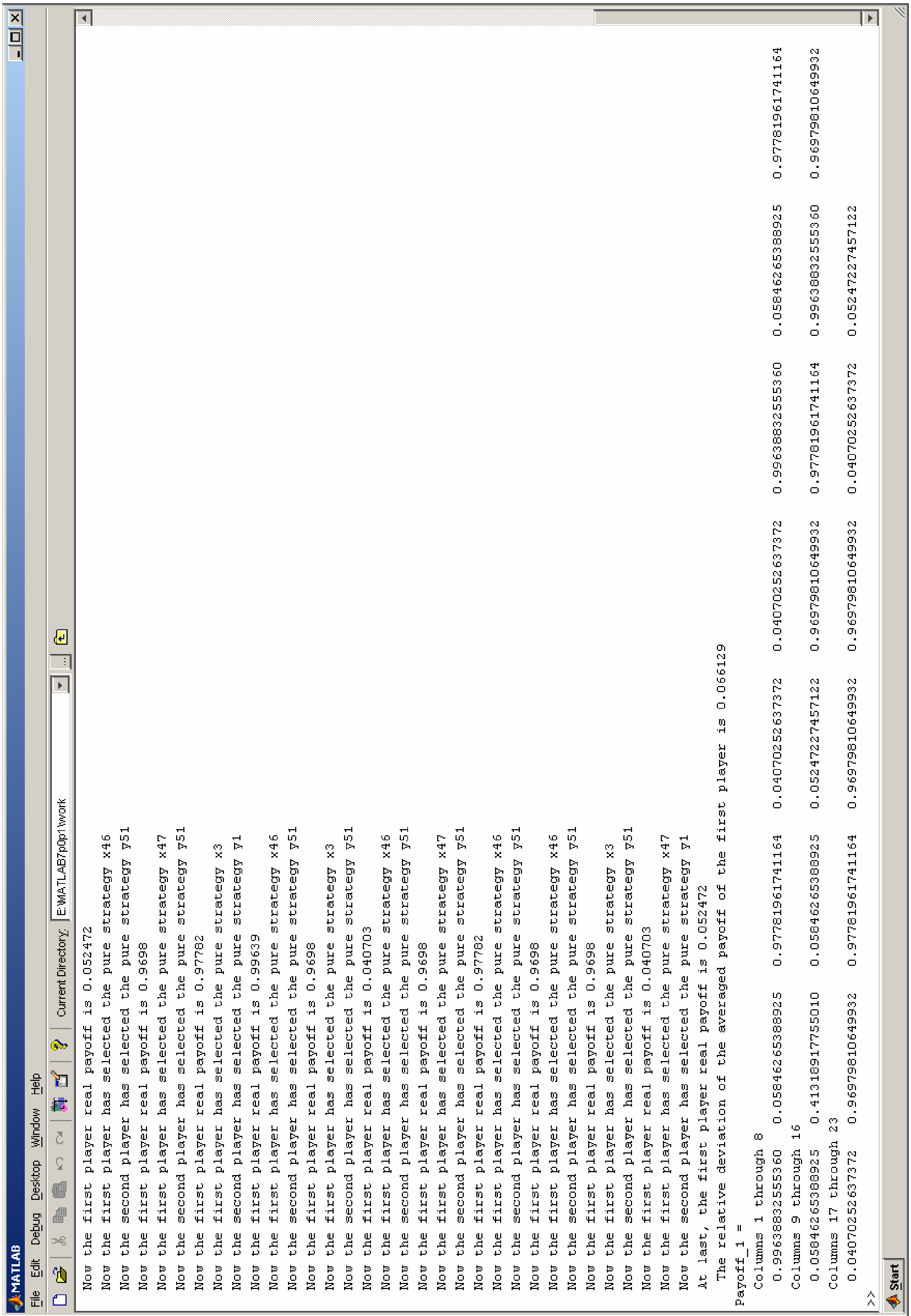


Figure 31. The second portion of the hints of what pure strategy to be selected and the vector (26) by $G = 23$ and another kernel (2) in the matrix form

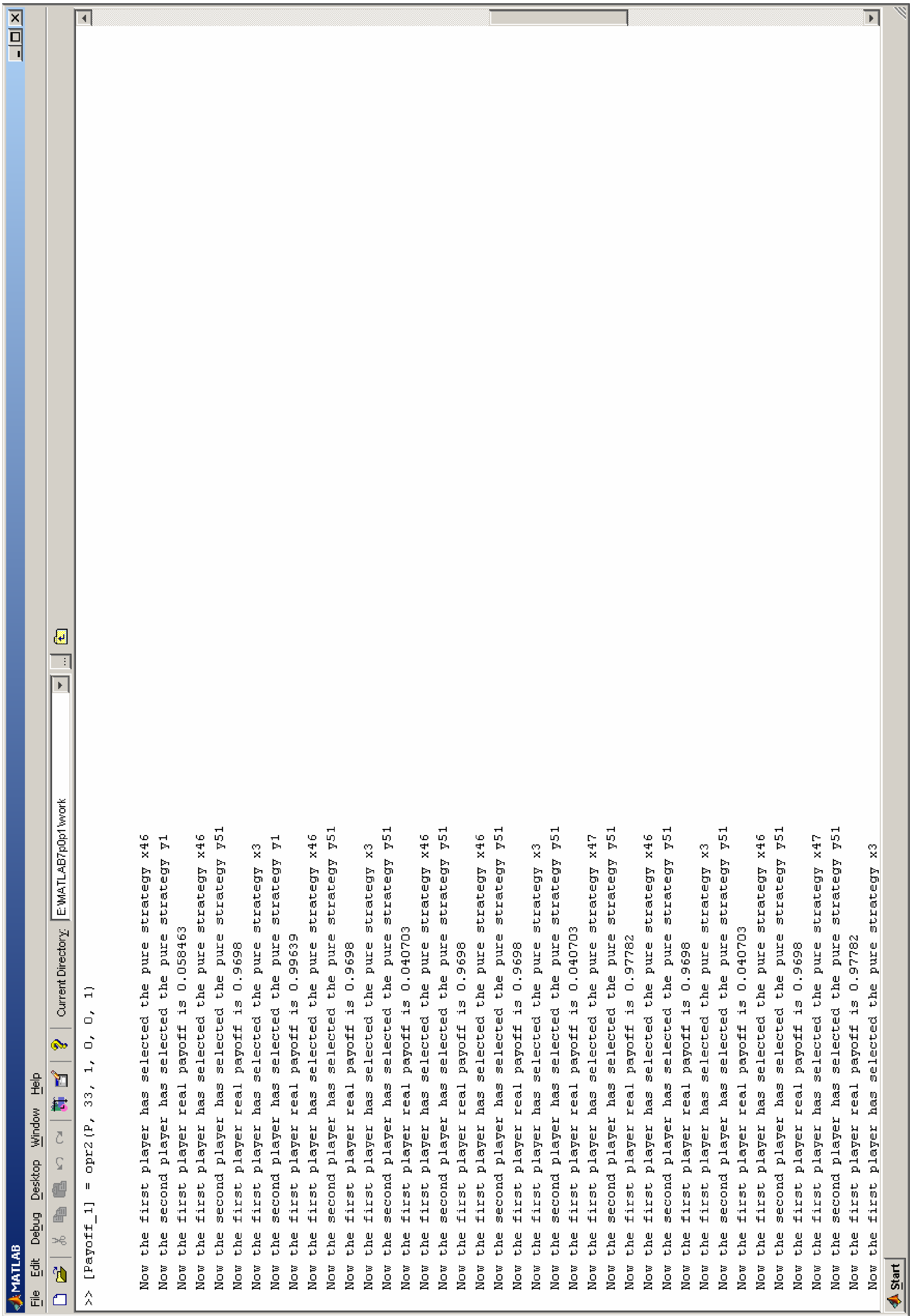


Figure 32. The first portion of the hints of what pure strategy to be selected and the vector (26) by $G = 33$ and another kernel (2) in the matrix form

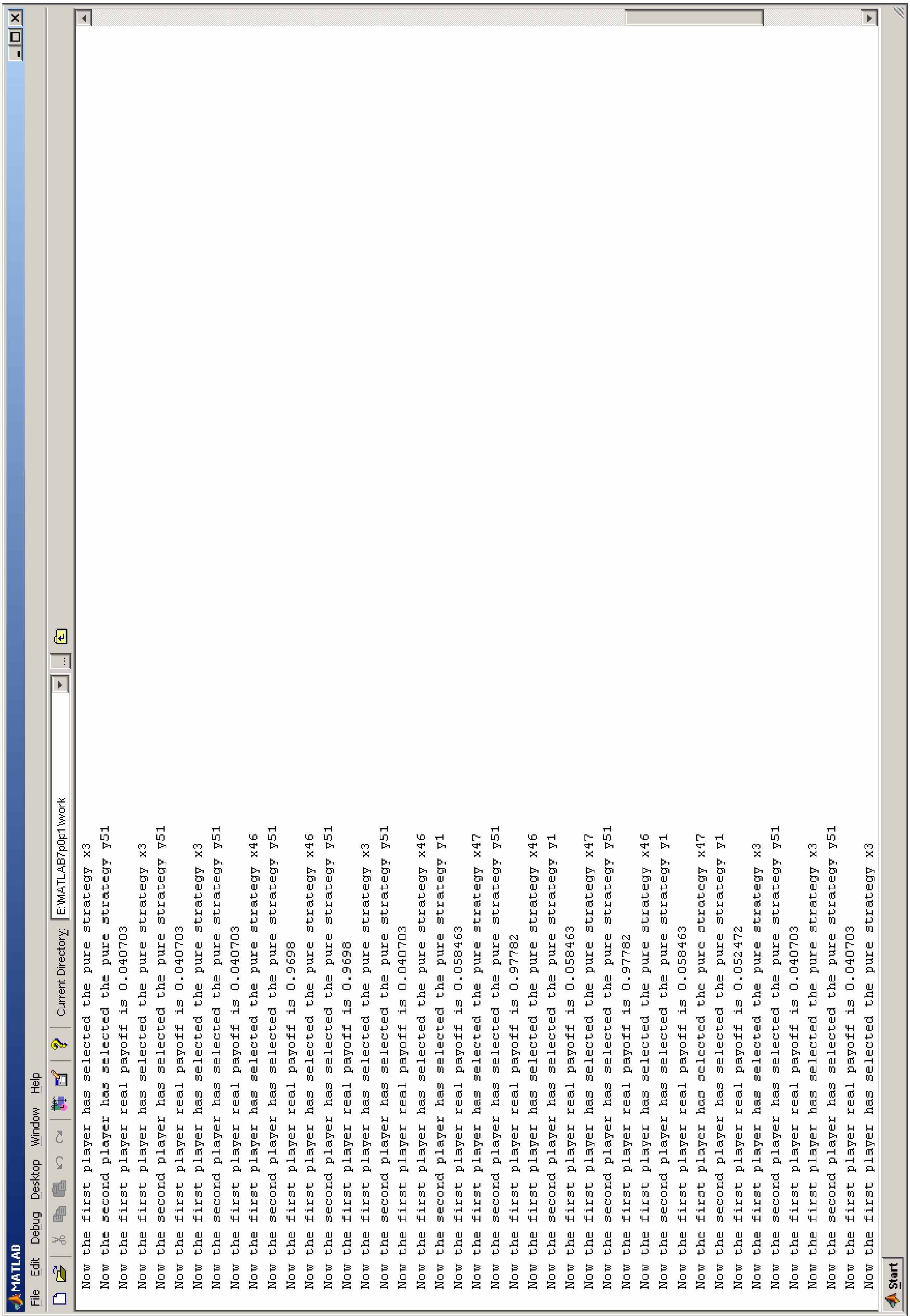


Figure 33. The second portion of the hints of what pure strategy to be selected and the vector (26) by $G = 33$ and another kernel (2) in the matrix form

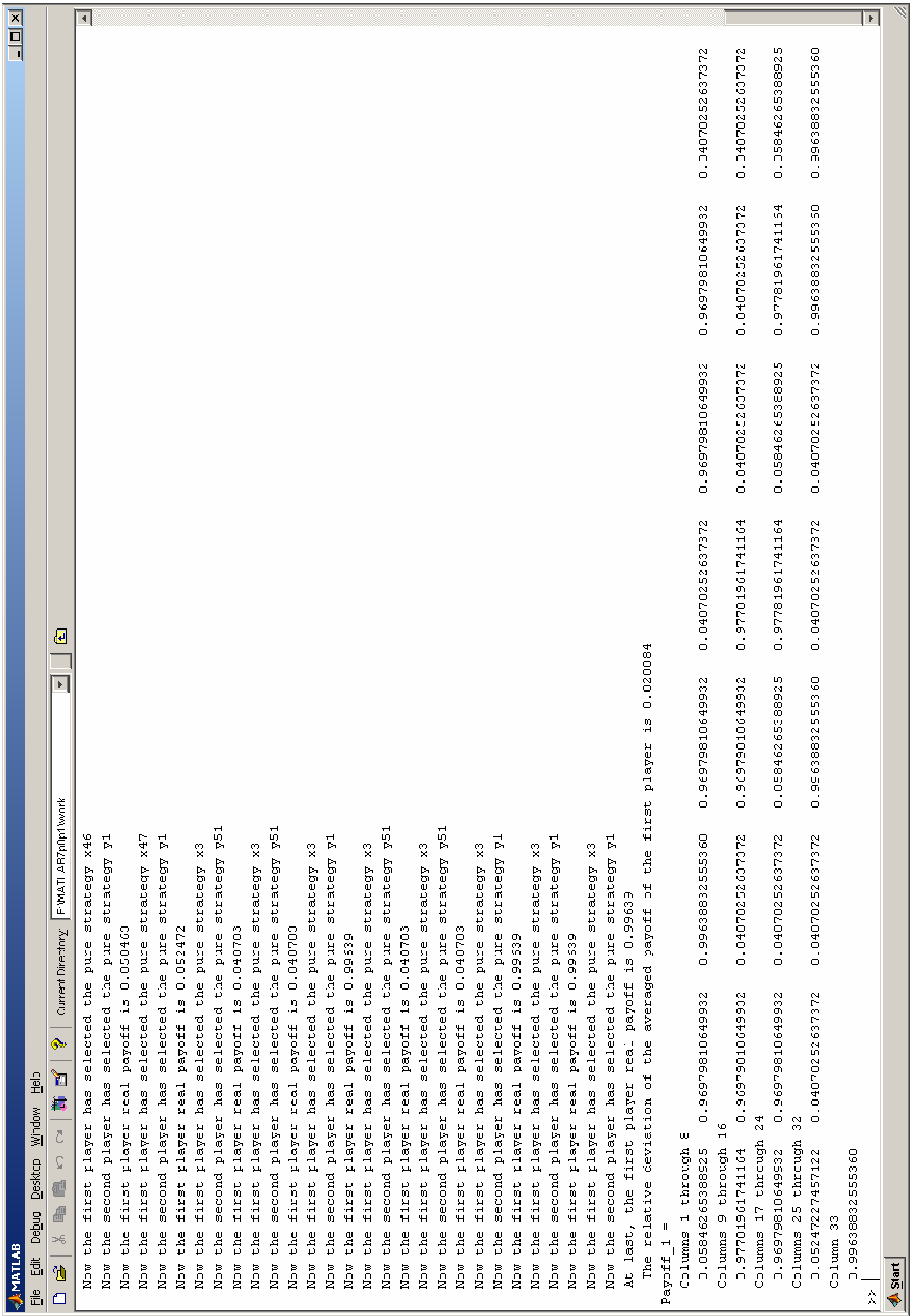


Figure 34. The third portion of the hints of what pure strategy to be selected and the vector (26) by $G = 33$ and another kernel (2) in the matrix form

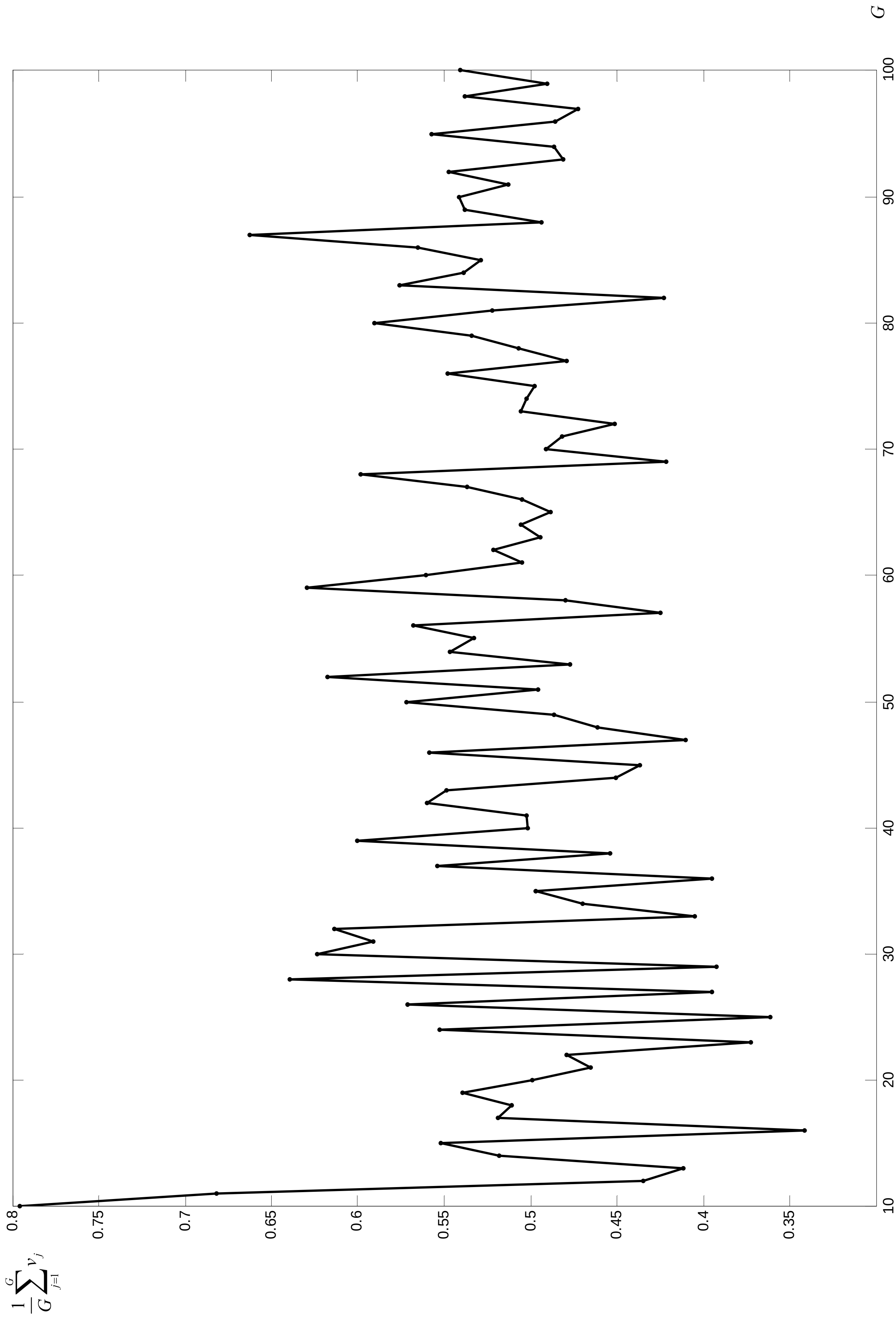


Figure 35. The averaged probability of striking the prey dependence from the number of shots $G \in \left\{\overline{10, 100}\right\}$

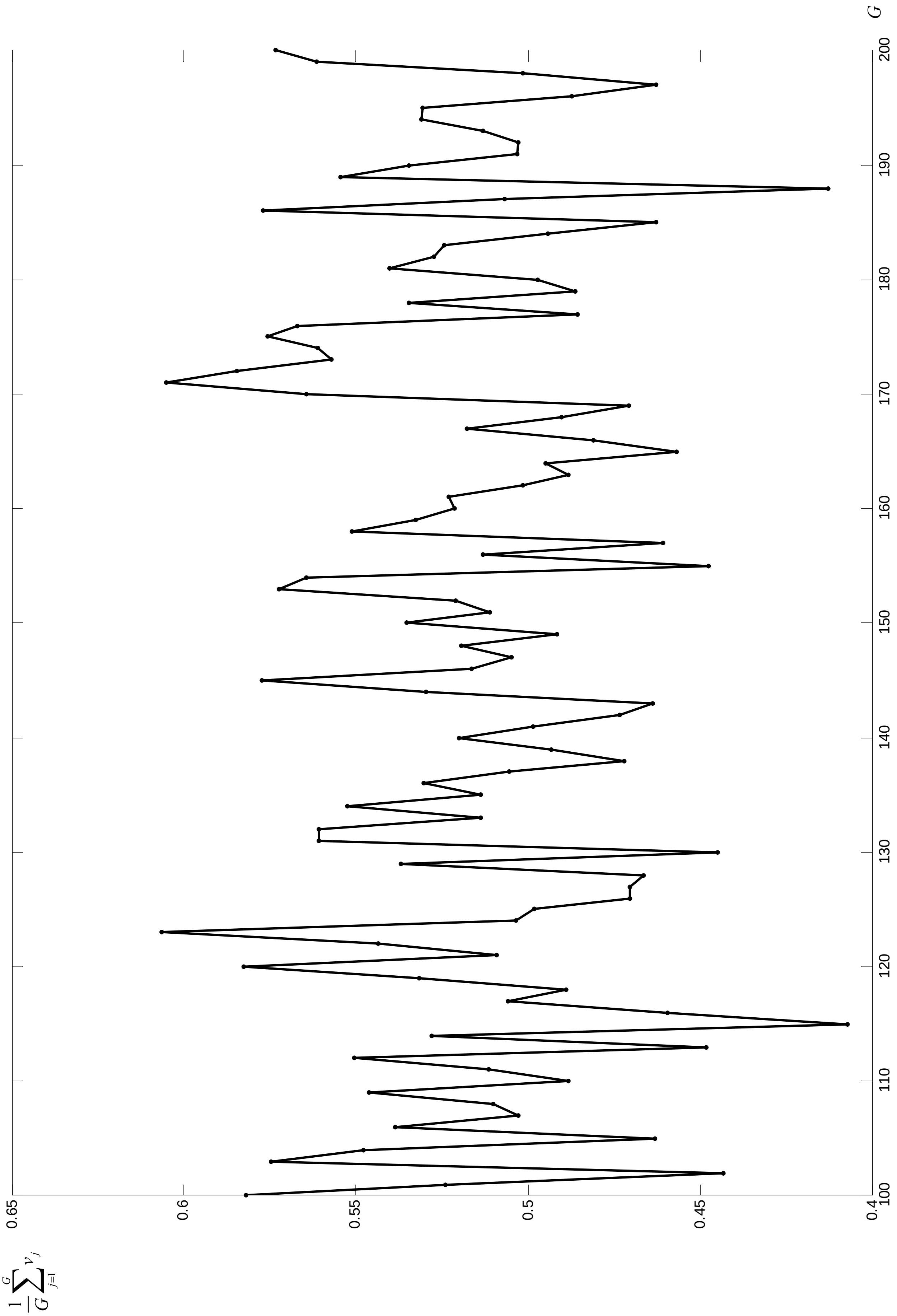


Figure 36. The averaged probability of striking the prey dependence from the number of shots $G \in \left\{\overline{100,\ 200}\right\}$

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

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

and its second derivative is

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

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

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

and its second derivative is

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

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

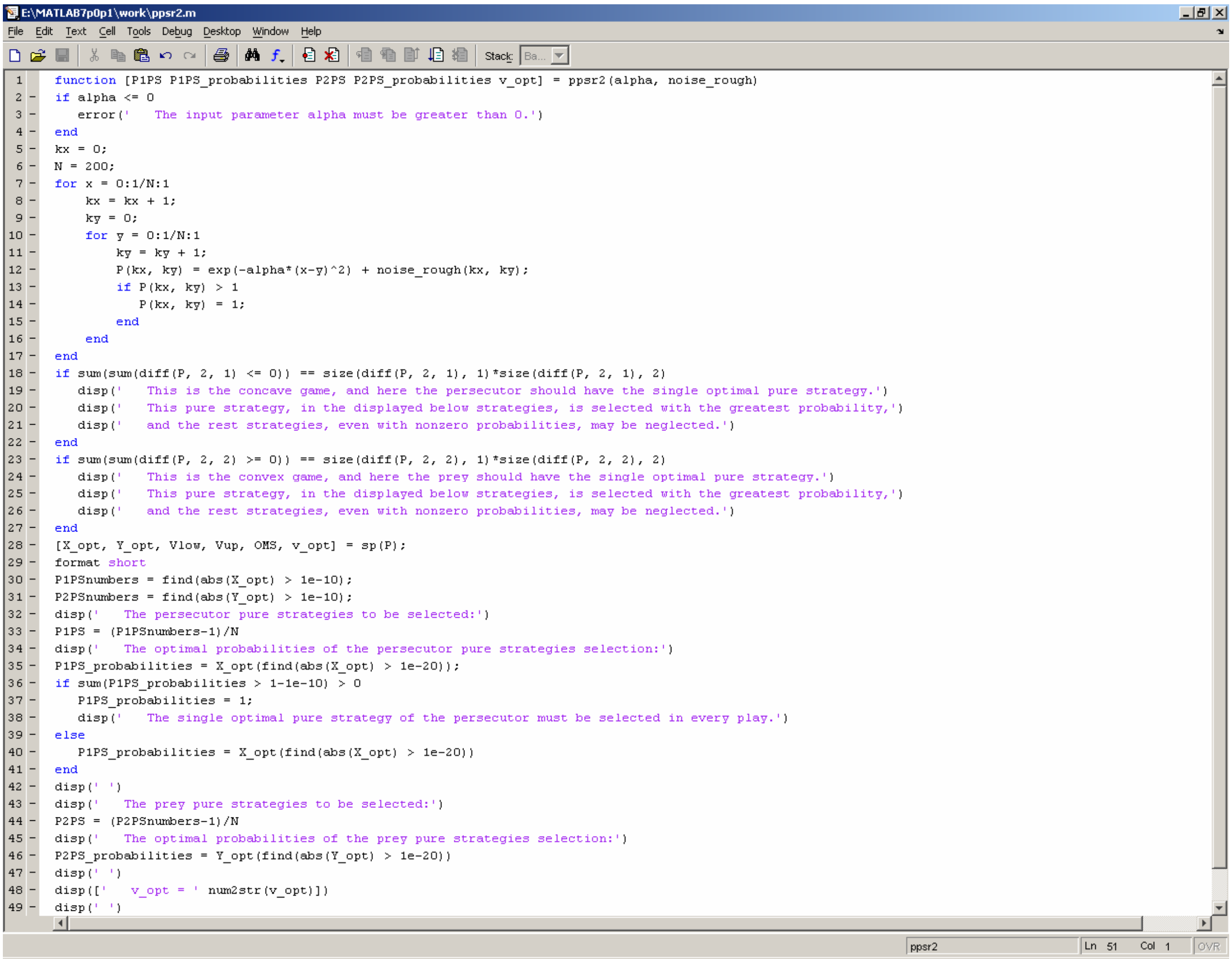
$$2\alpha\left(2\alpha(x-y)^2 - 1\right)\exp\left[-\alpha(x-y)^2\right] + \frac{\partial^2 n(x,\, y)}{\partial x^2} \leqslant 0 \quad (33)$$

and the convexity condition

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

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.



```
function [P1PS P1PS_probabilities P2PS P2PS_probabilities v_opt] = ppsr2(alpha, noise_rough)
if alpha <= 0
   error('   The input parameter alpha must be greater than 0.')
end
kx = 0;
N = 200;
for x = 0:1/N:1
     kx = kx + 1;
     ky = 0;
     for y = 0:1/N:1
         ky = ky + 1;
         P(kx, ky) = exp(-alpha*(x-y)^2) + noise_rough(kx, ky);
         if P(kx, ky) > 1
            P(kx, ky) = 1;
         end
     end
end
if sum(sum(diff(P, 2, 1) <= 0)) == size(diff(P, 2, 1), 1)*size(diff(P, 2, 1), 2)
   disp('   This is the concave game, and here the persecutor should have the single optimal pure strategy.')
   disp('   This pure strategy, in the displayed below strategies, is selected with the greatest probability,')
   disp('   and the rest strategies, even with nonzero probabilities, may be neglected.')
end
if sum(sum(diff(P, 2, 2) >= 0)) == size(diff(P, 2, 2), 1)*size(diff(P, 2, 2), 2)
   disp('   This is the convex game, and here the prey should have the single optimal pure strategy.')
   disp('   This pure strategy, in the displayed below strategies, is selected with the greatest probability,')
   disp('   and the rest strategies, even with nonzero probabilities, may be neglected.')
end
[X_opt, Y_opt, Vlow, Vup, OMS, v_opt] = sp(P);
format short
P1PSnumbers = find(abs(X_opt) > 1e-10);
P2PSnumbers = find(abs(Y_opt) > 1e-10);
disp('   The persecutor pure strategies to be selected:')
P1PS = (P1PSnumbers-1)/N
disp('   The optimal probabilities of the persecutor pure strategies selection:')
P1PS_probabilities = X_opt(find(abs(X_opt) > 1e-20));
if sum(P1PS_probabilities > 1-1e-10) > 0
   P1PS_probabilities = 1;
   disp('   The single optimal pure strategy of the persecutor must be selected in every play.')
else
   P1PS_probabilities = X_opt(find(abs(X_opt) > 1e-20))
end
disp(' ')
disp('   The prey pure strategies to be selected:')
P2PS = (P2PSnumbers-1)/N
disp('   The optimal probabilities of the prey pure strategies selection:')
P2PS_probabilities = Y_opt(find(abs(Y_opt) > 1e-20))
disp(' ')
disp(['   v_opt = ' num2str(v_opt)])
disp(' ')
```

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

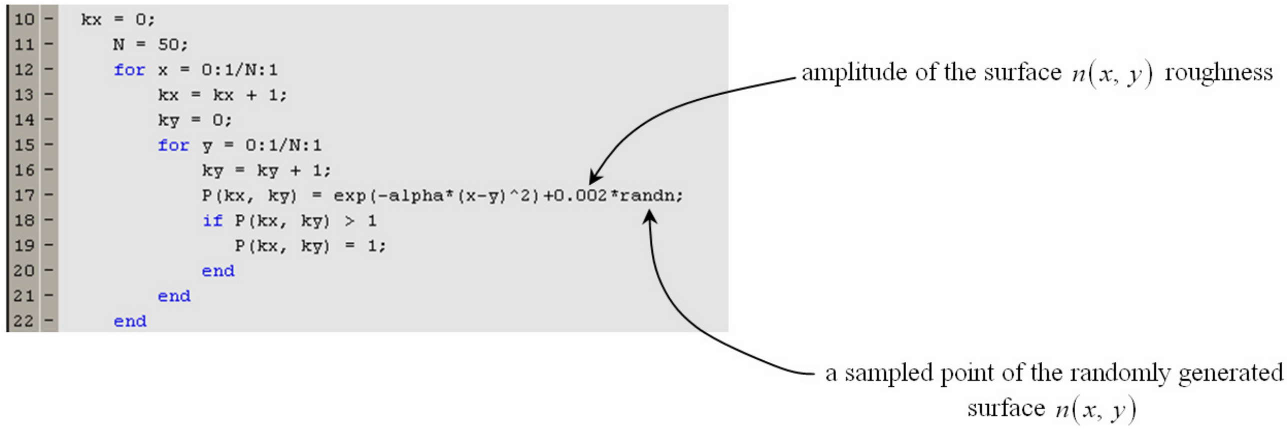


Figure 38. The fragment of the script code, which returns the sampled surface $n(x, y)$, what had been used by plotting the figures 1 — 4

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.

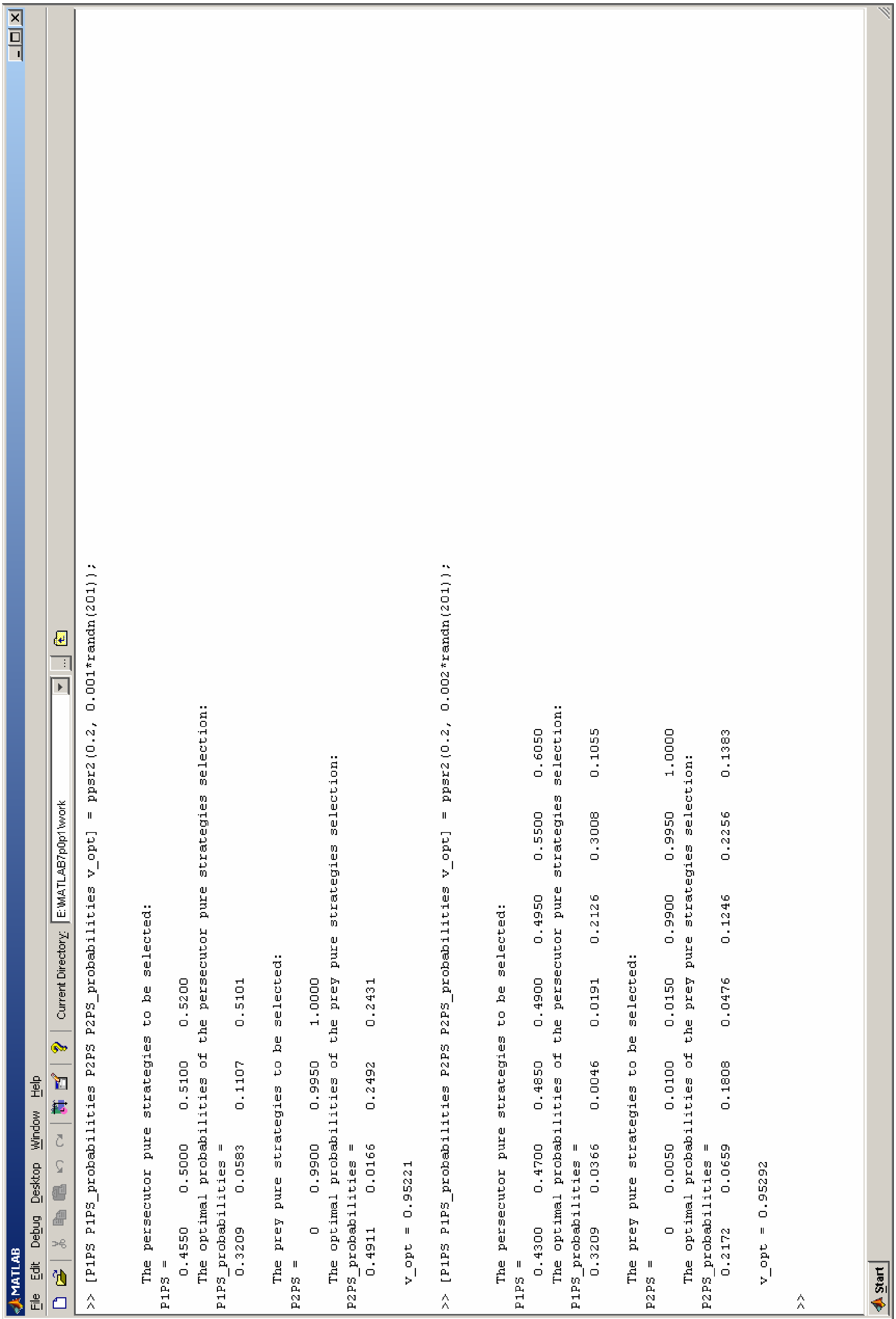


Figure 39. Program module ppsr2 application with $\alpha = 0.2$ and some low noise amplitudes

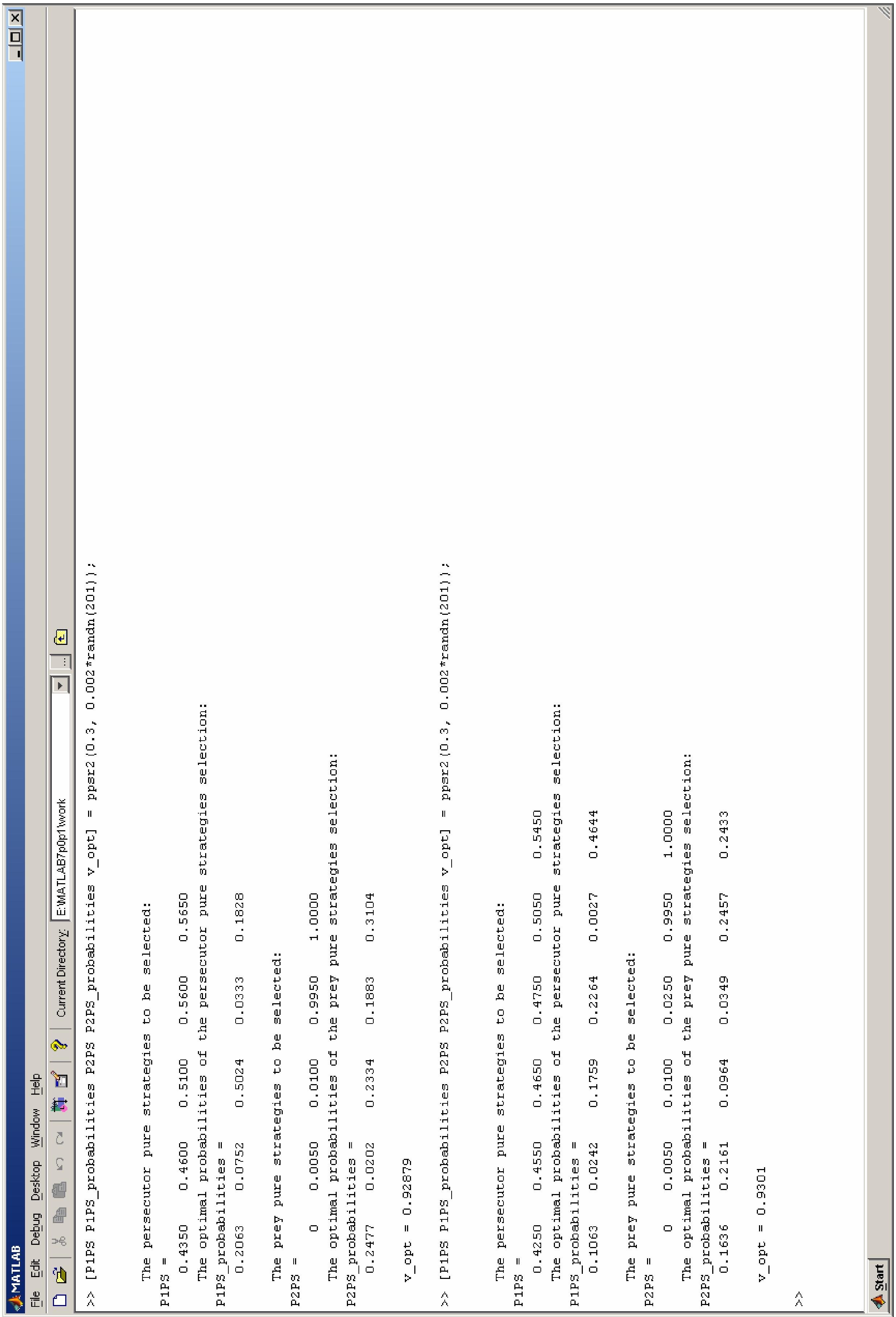


Figure 40. Program module `ppsr2` application with $\alpha = 0.3$ and the twice greater noise amplitude

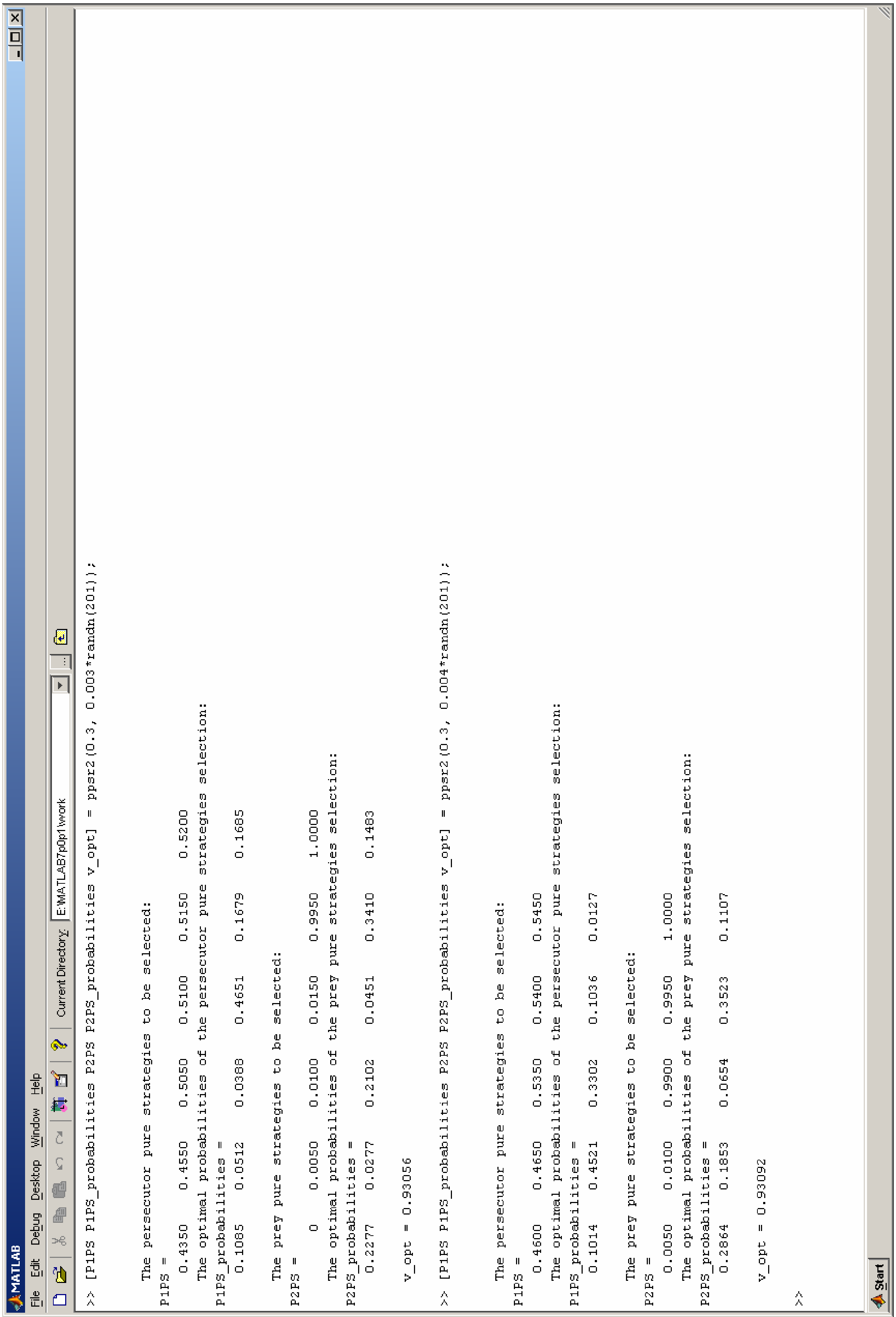


Figure 41. Program module `ppsr2` application with $\alpha = 0.3$ and more great noise amplitudes

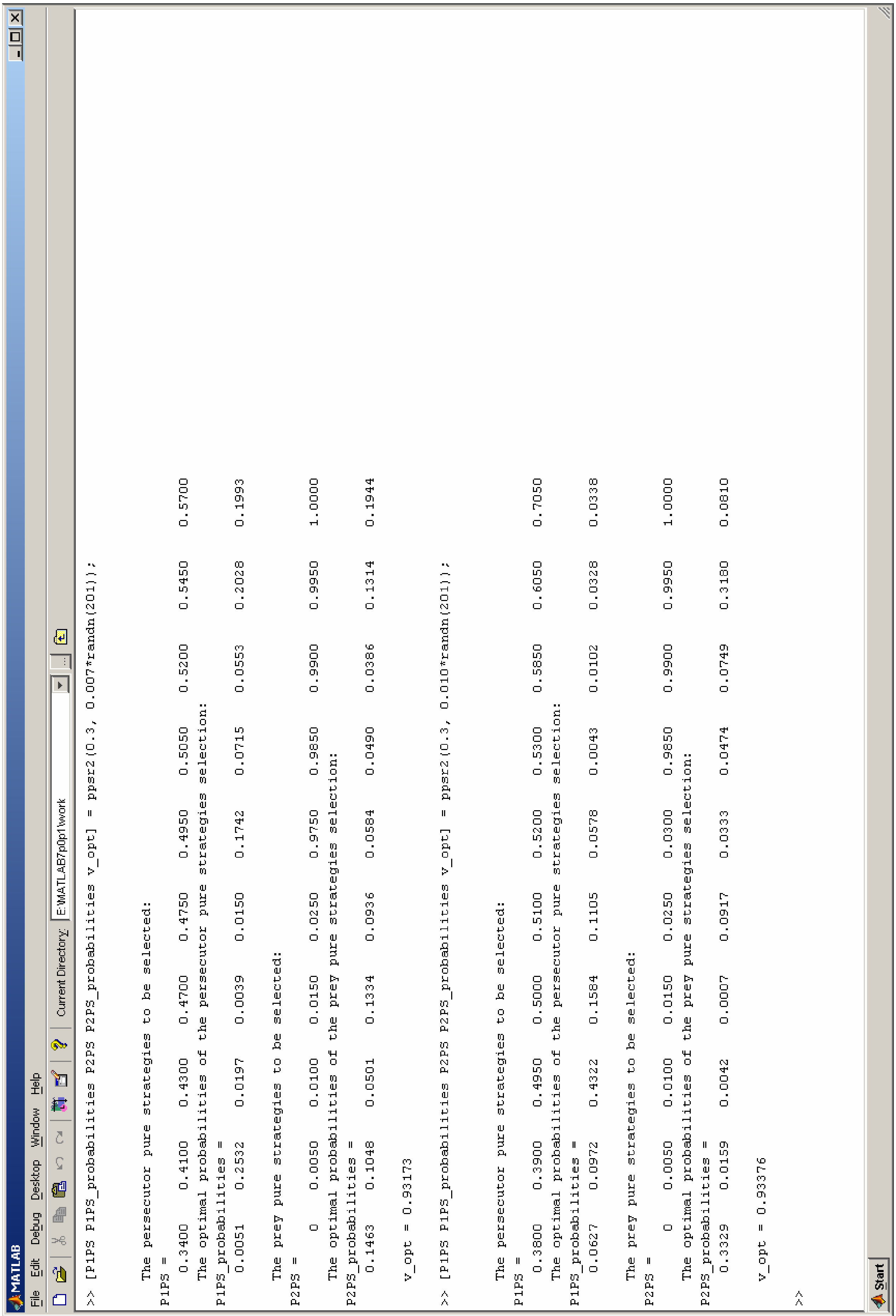


Figure 42. Program module `ppsr2` application with $\alpha = 0.3$ and some high noise amplitude

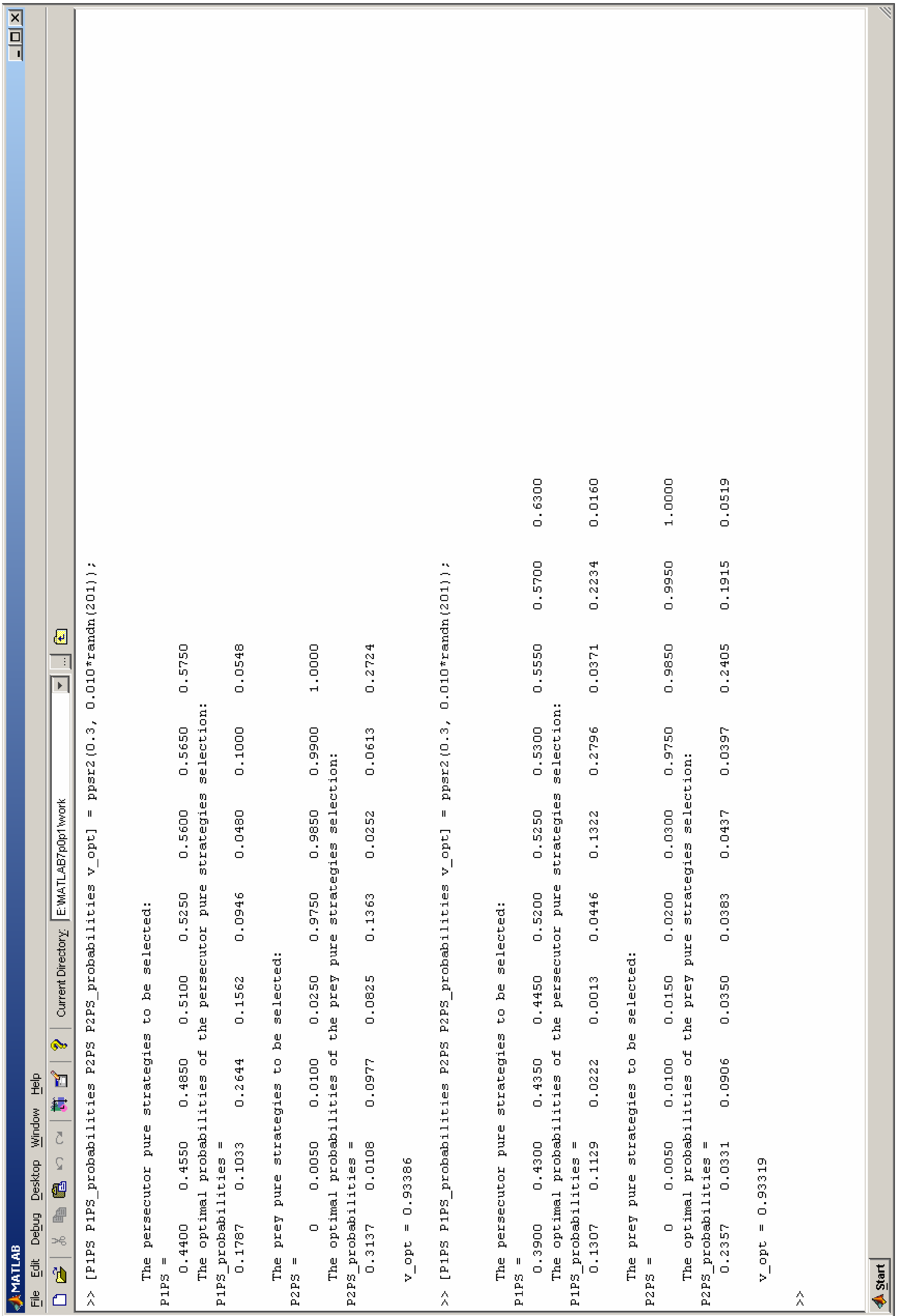


Figure 43. The two repeats of the previous runs

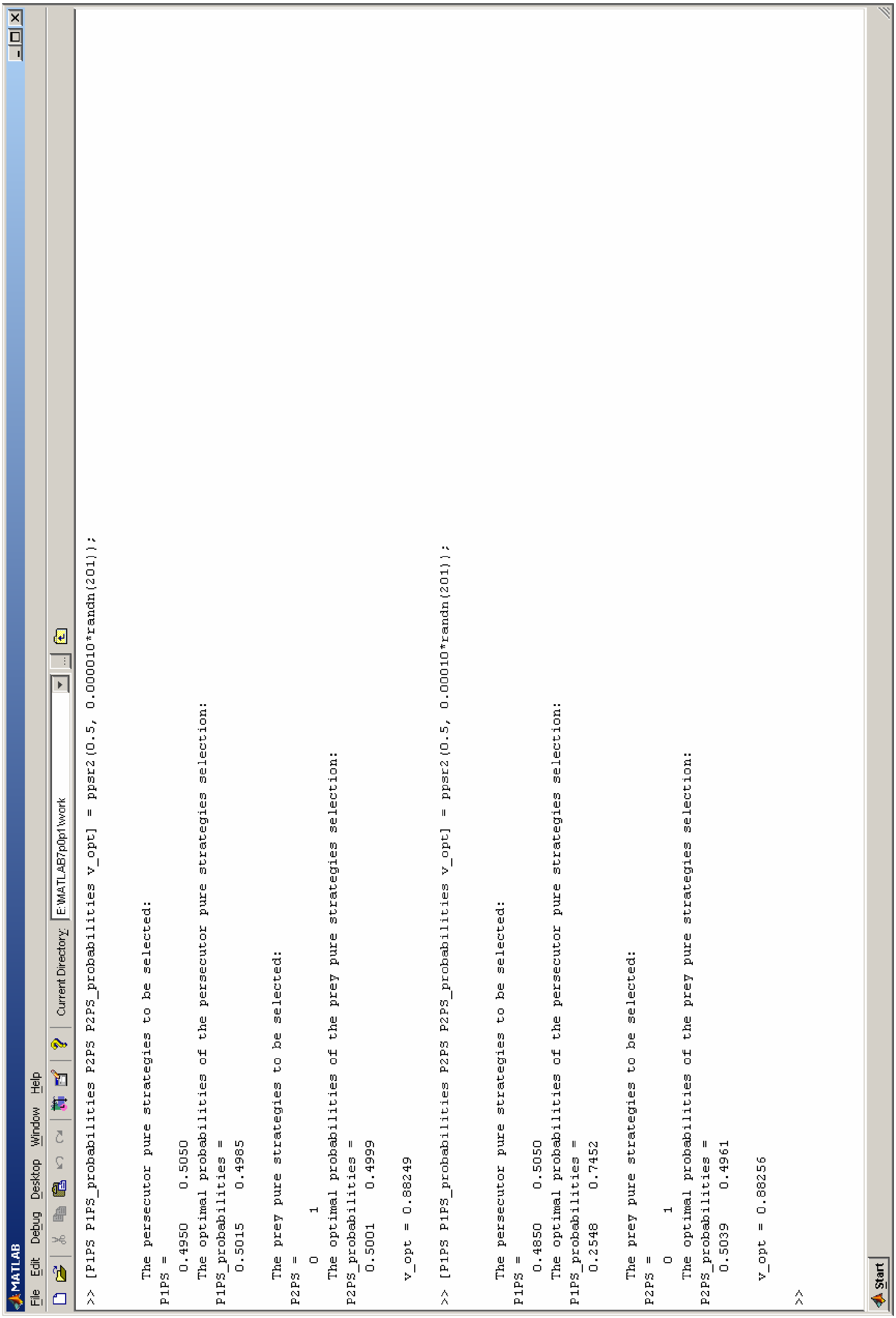


Figure 44. Program module `ppsr2` application with $\alpha = 0.5$ by very low noise amplitude and ten times greater one

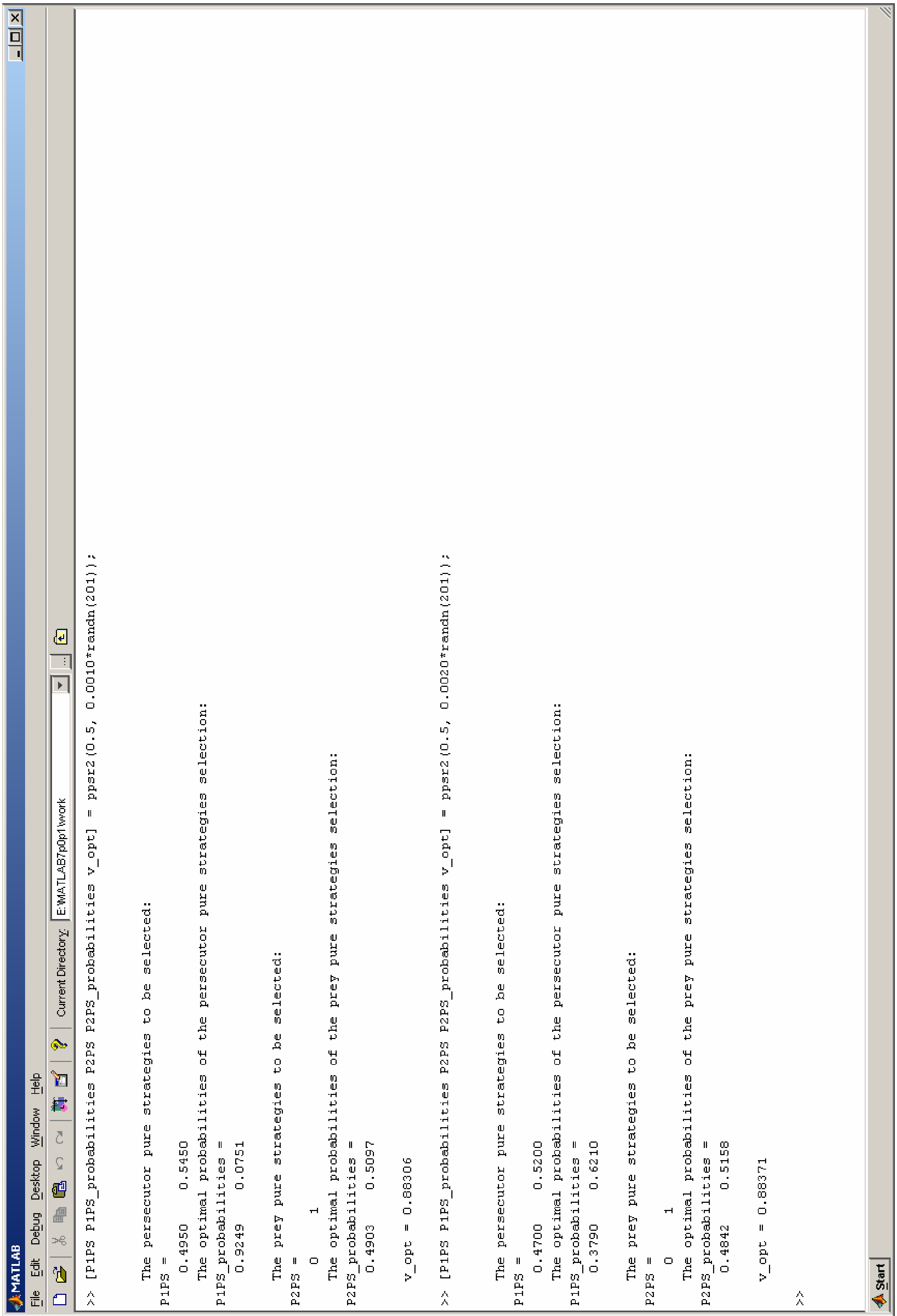


Figure 45. Continuing to increase the noise amplitude by $\alpha = 0.5$

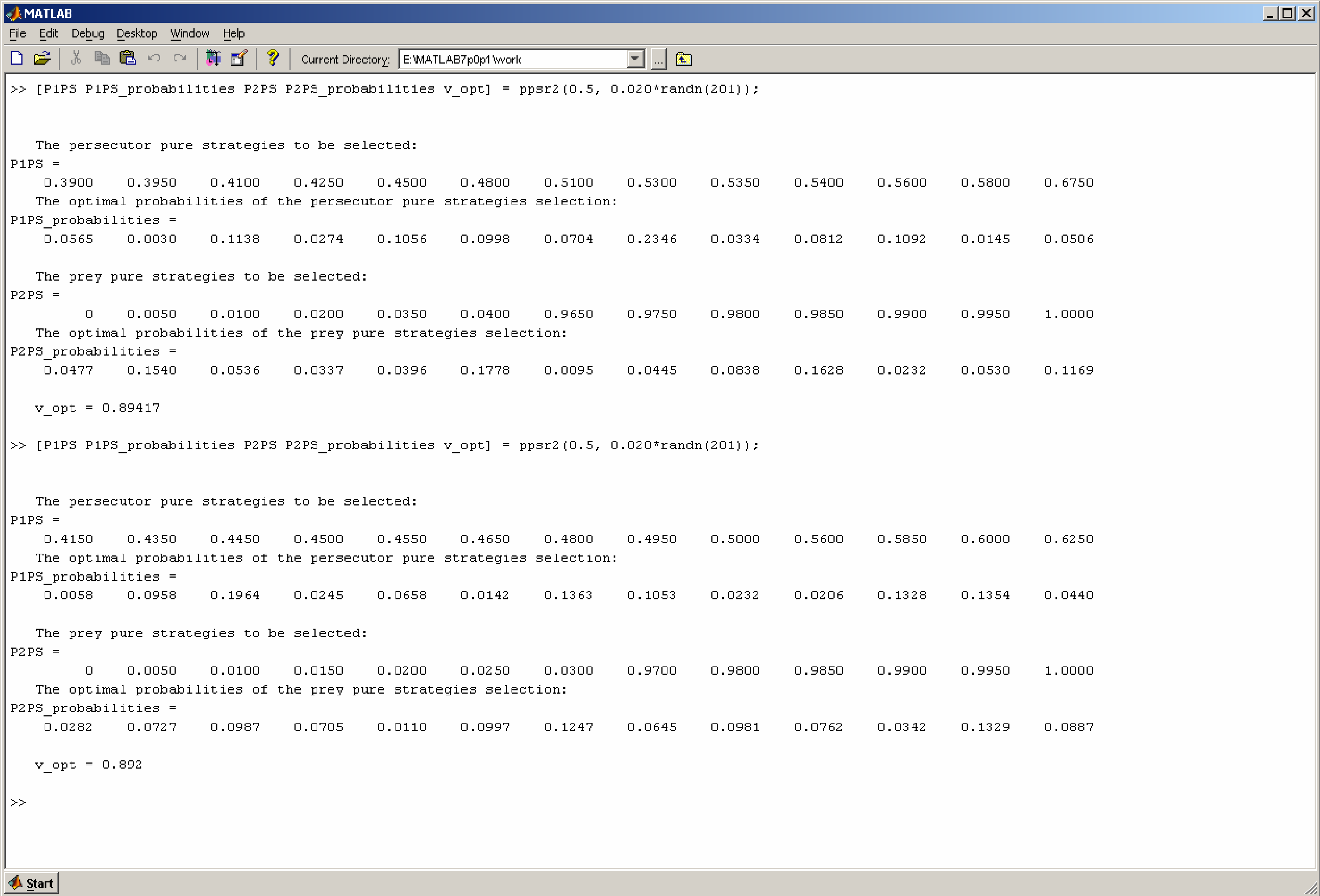
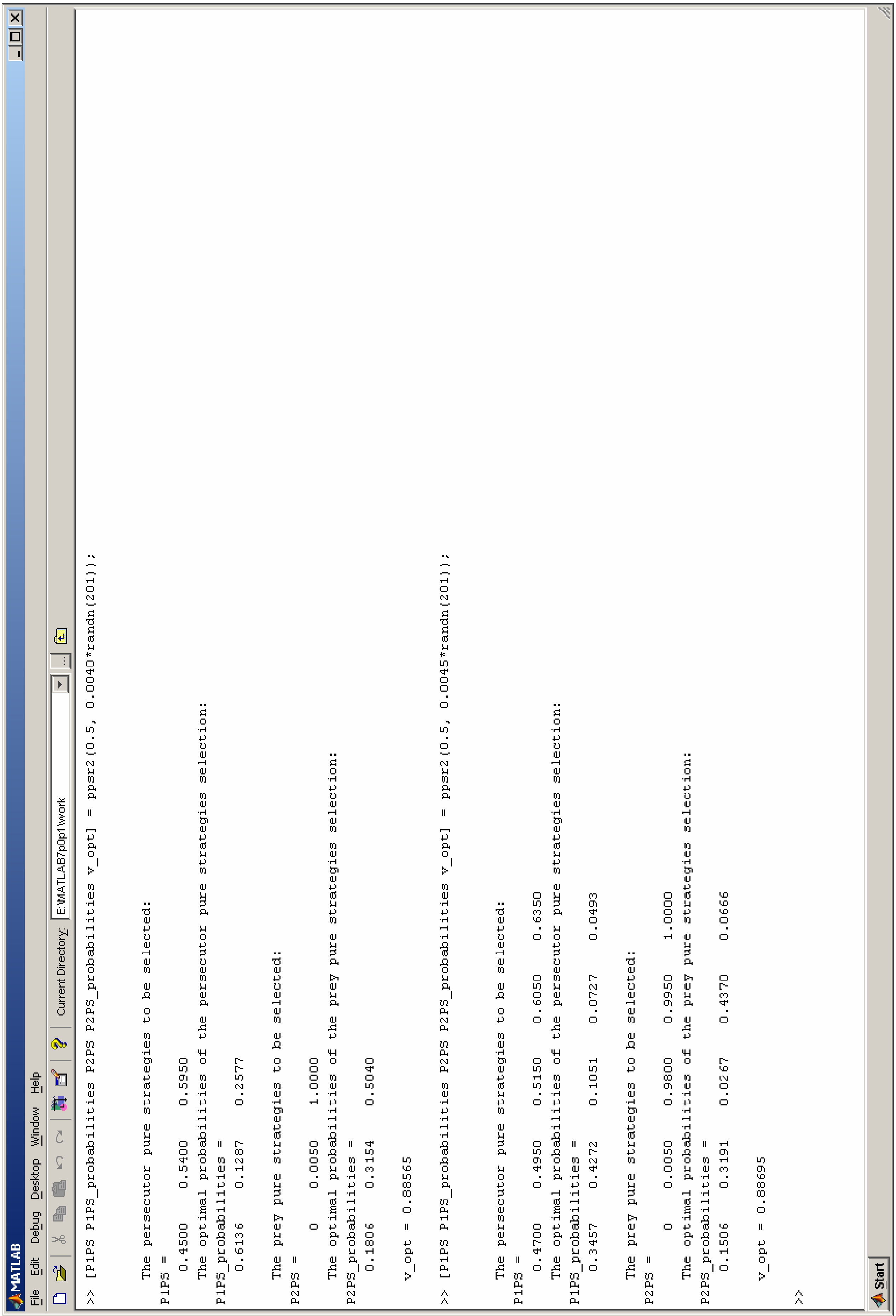


Figure 46. Program module ppsr2 application with $\alpha = 0.5$ by pretty high noise amplitude



```
MATLAB
File  Edit  Debug  Desktop  Window  Help
Current Directory: E:\MATLAB7p0p1\work

>> [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

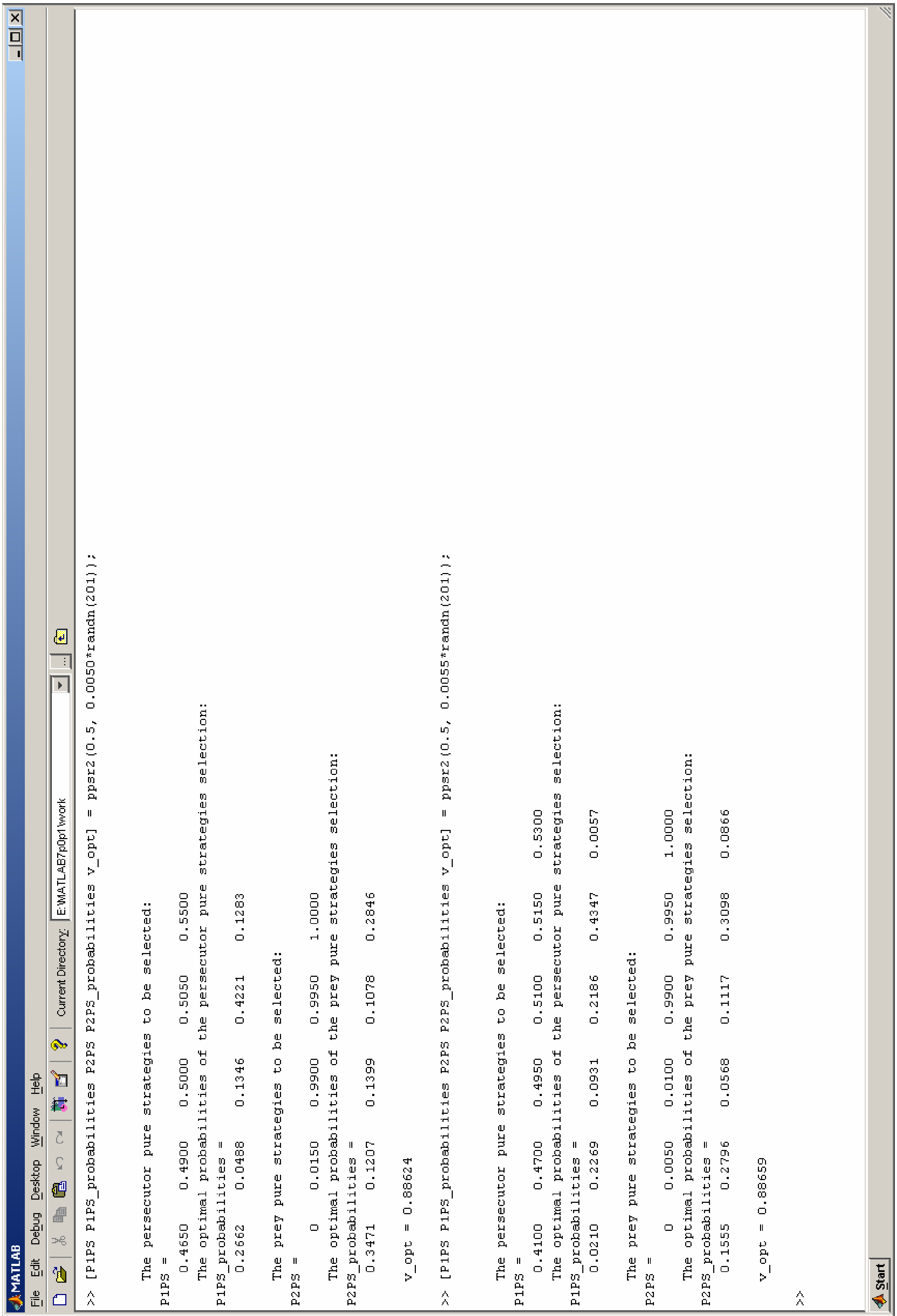


Figure 48. Further increasing the noise amplitude, and the spectrums of the optimal strategies of both the persecutor and prey keep widen

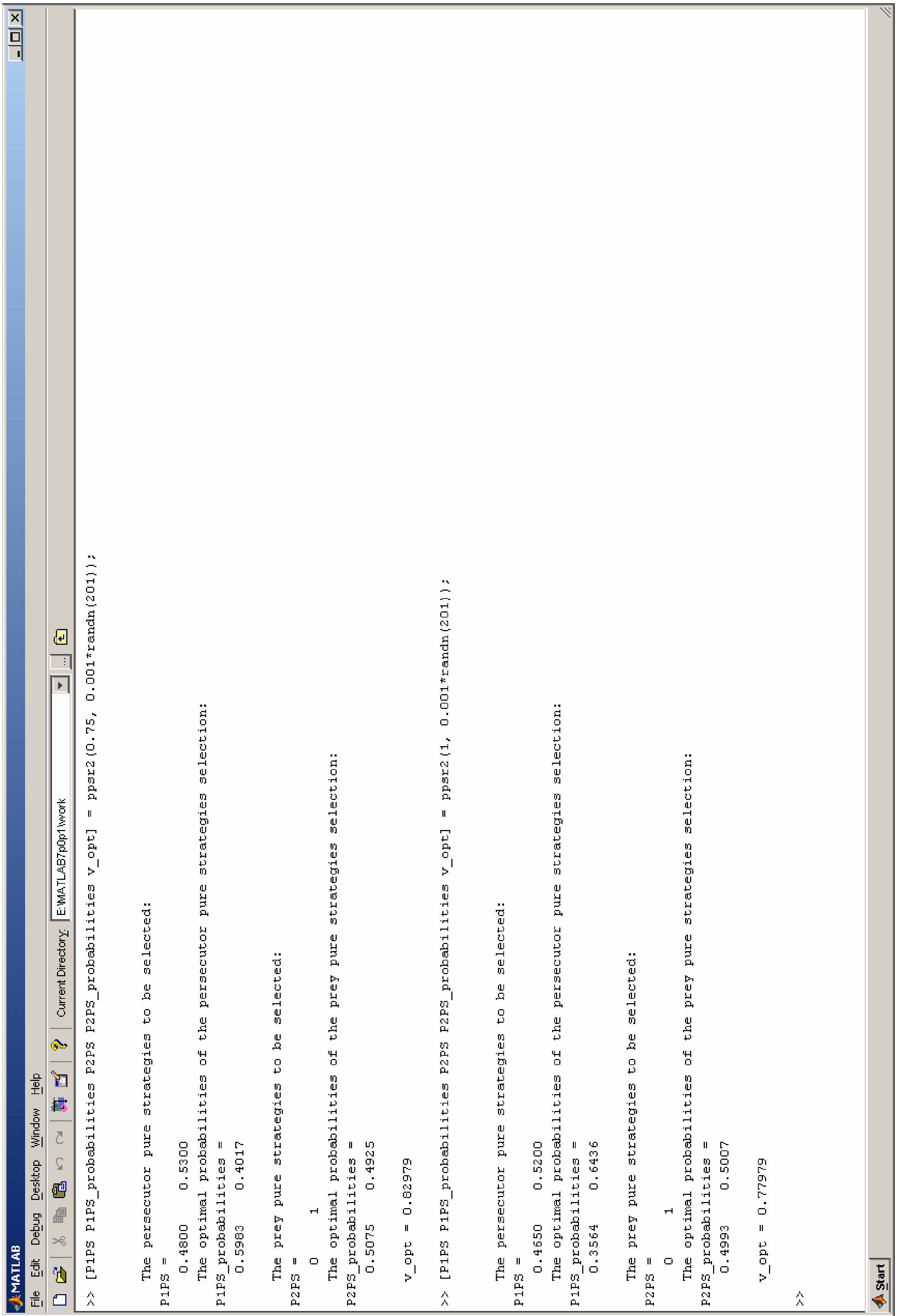


Figure 49. Taking $\alpha = 0.75$ or $\alpha = 1$ at the low noise amplitude

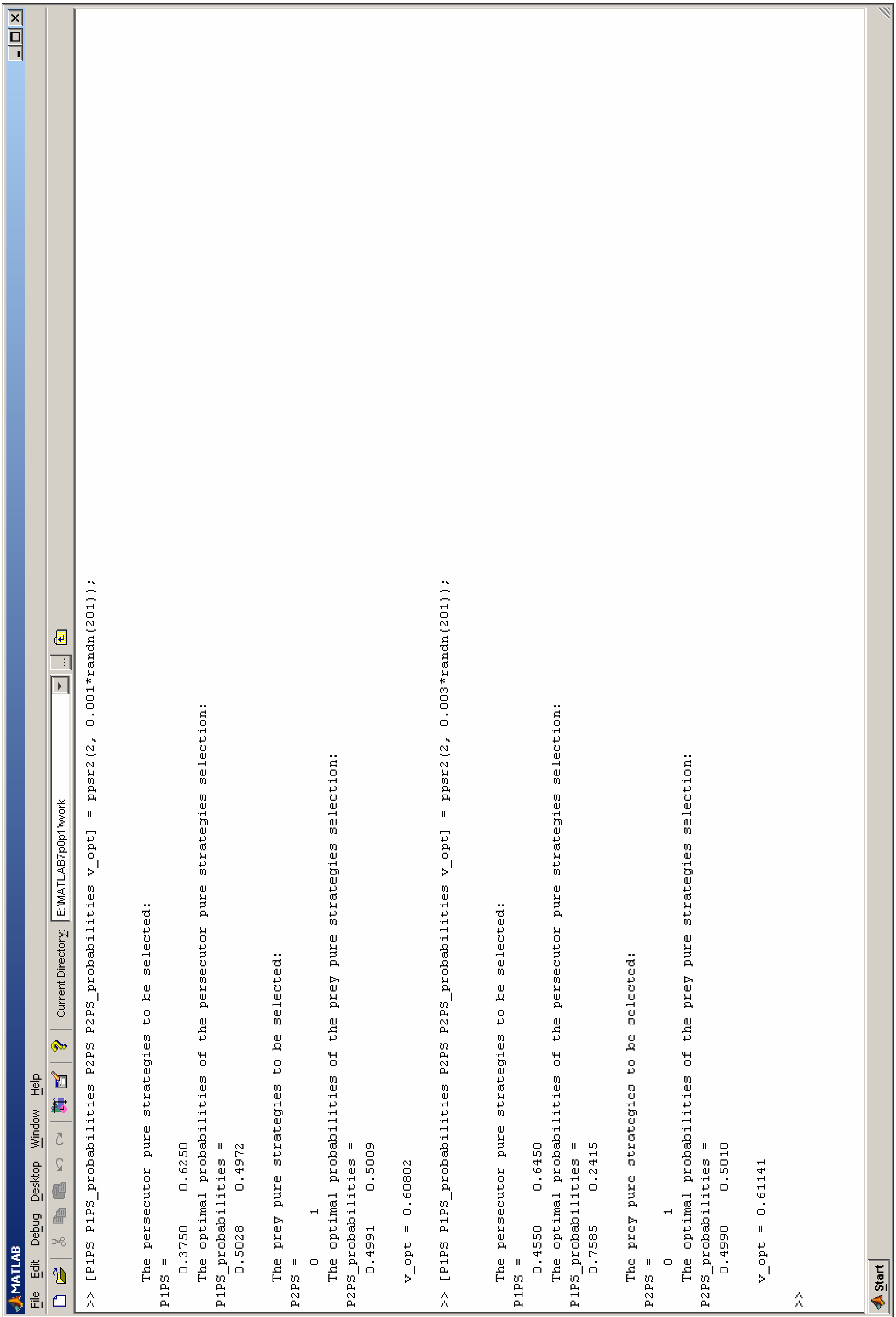


Figure 50. Antisymmetrization for $\alpha = 2$ by increasing the noise amplitude

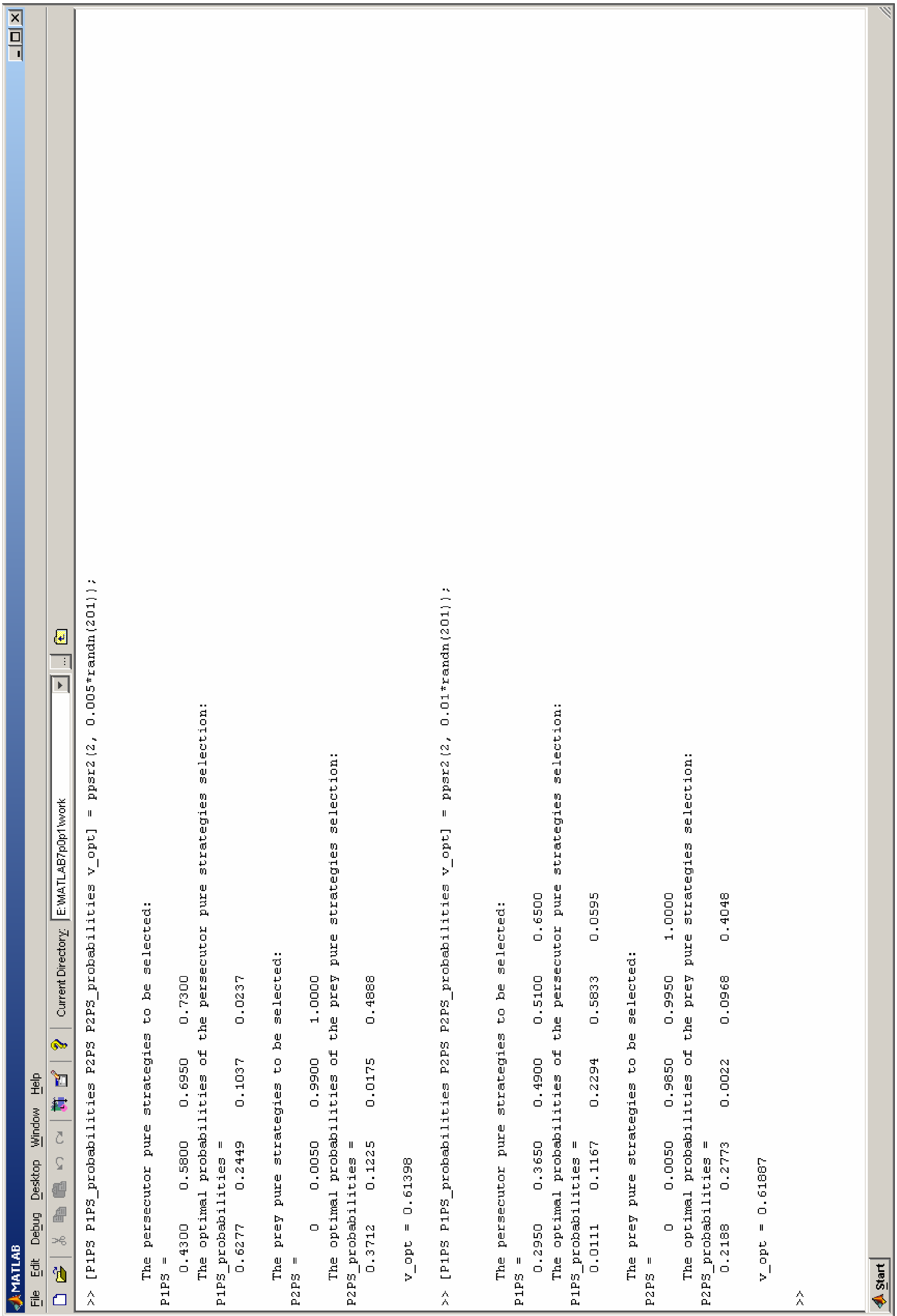


Figure 51. Further increasing the noise amplitude at $\alpha = 2$, and the spectrums of the optimal strategies of both the players keep widening

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   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   0.81688518383199   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

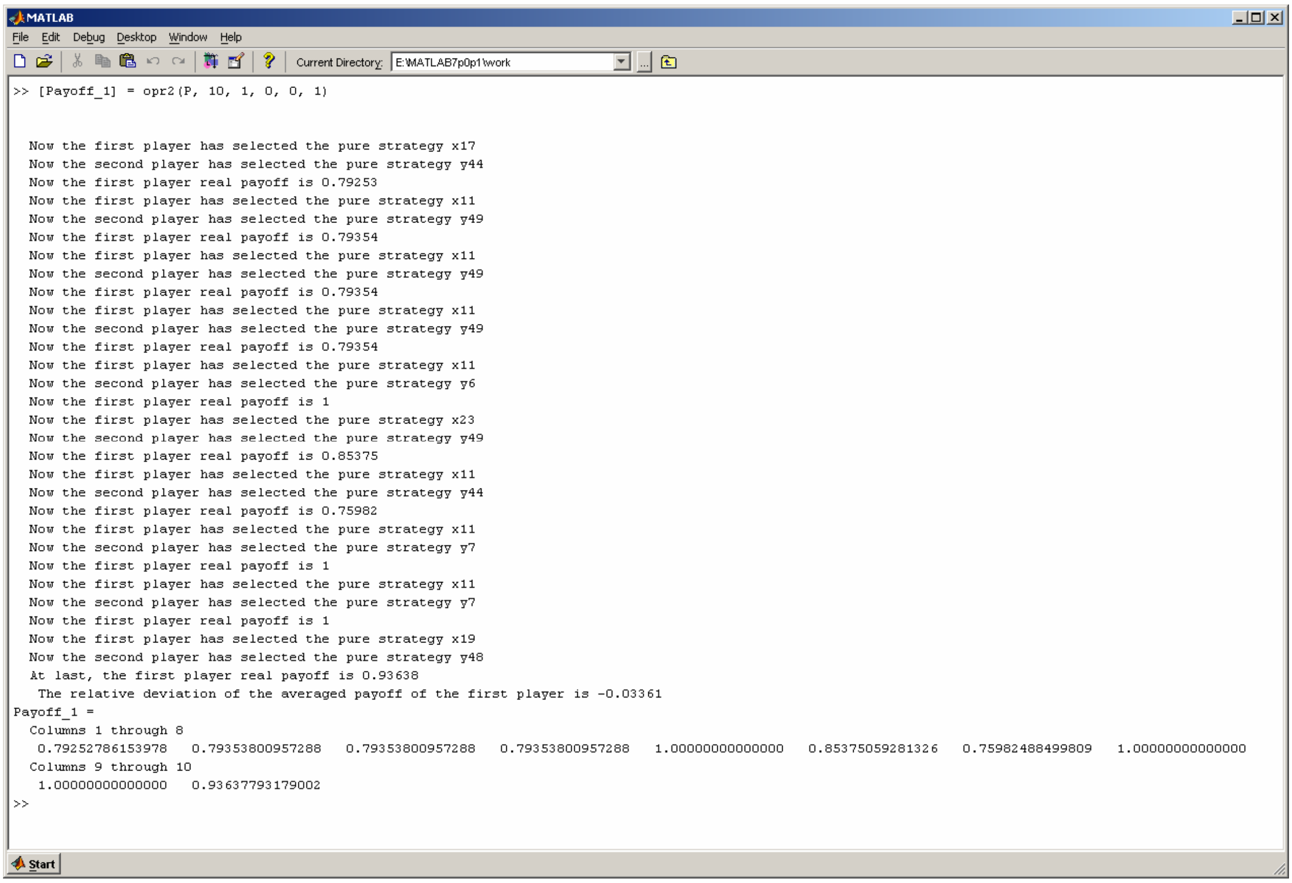


Figure 54. The module `opr2` application with the 10 shots by some high noise level and the scheme of selecting the pure strategies

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 `ppsr2` and running it.

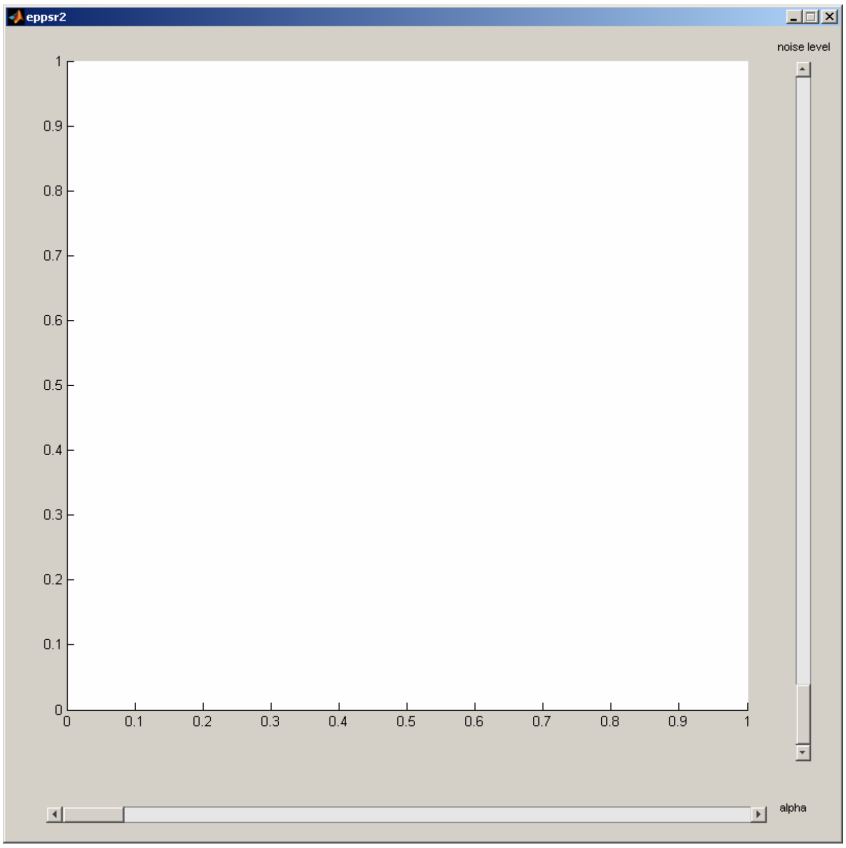


Figure 55. The executive form `eppsr2` general view as it just has been loaded by the typing `eppsr2` and entering it from the MATLAB 7.0.1 Command Window line

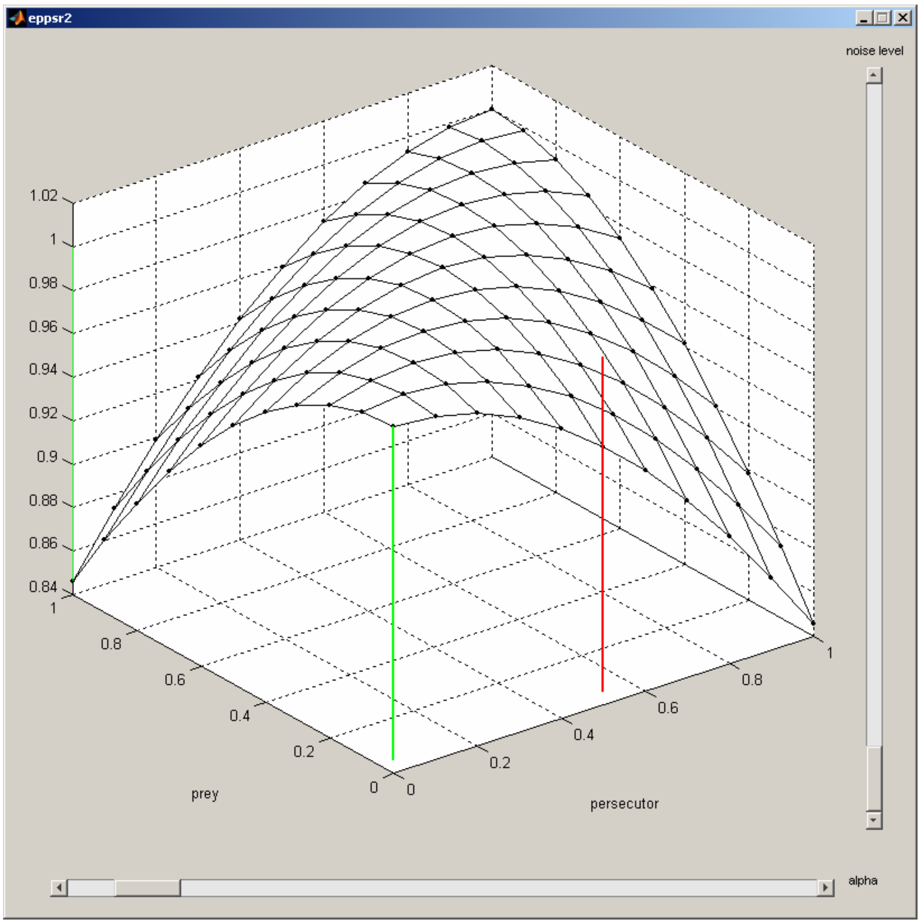


Figure 56. The single optimal pure strategy of the persecutor and the two prey pure strategies to be selected by the low α and the almost lowest noise level

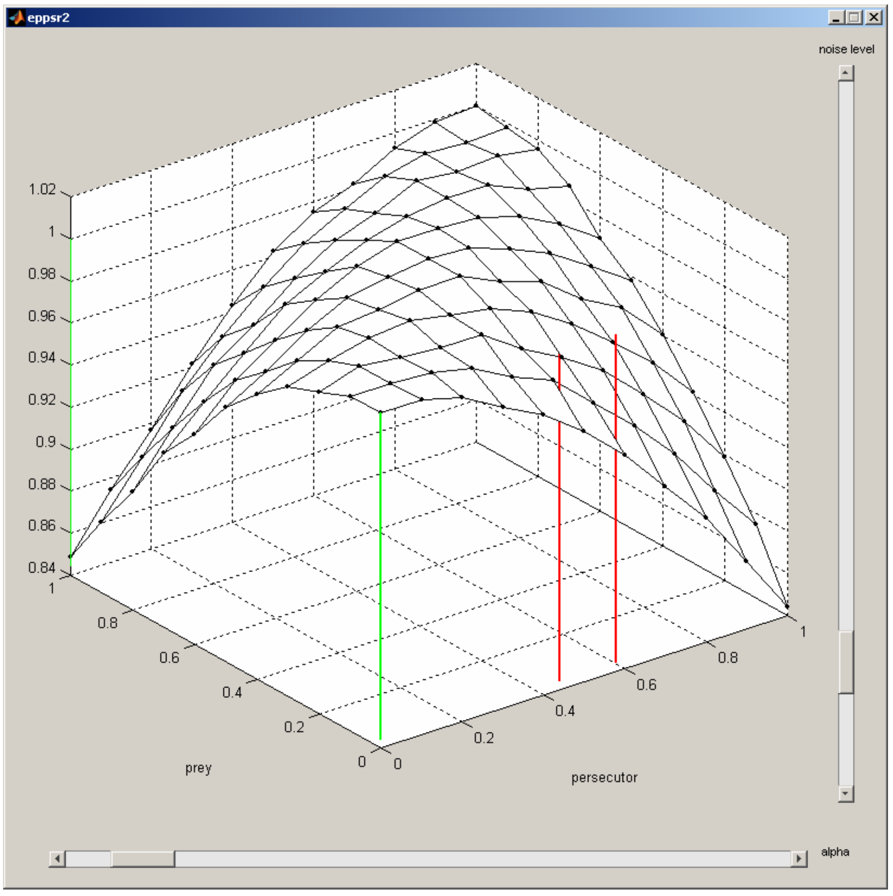


Figure 57. The equilibrium situation by some noise level increment

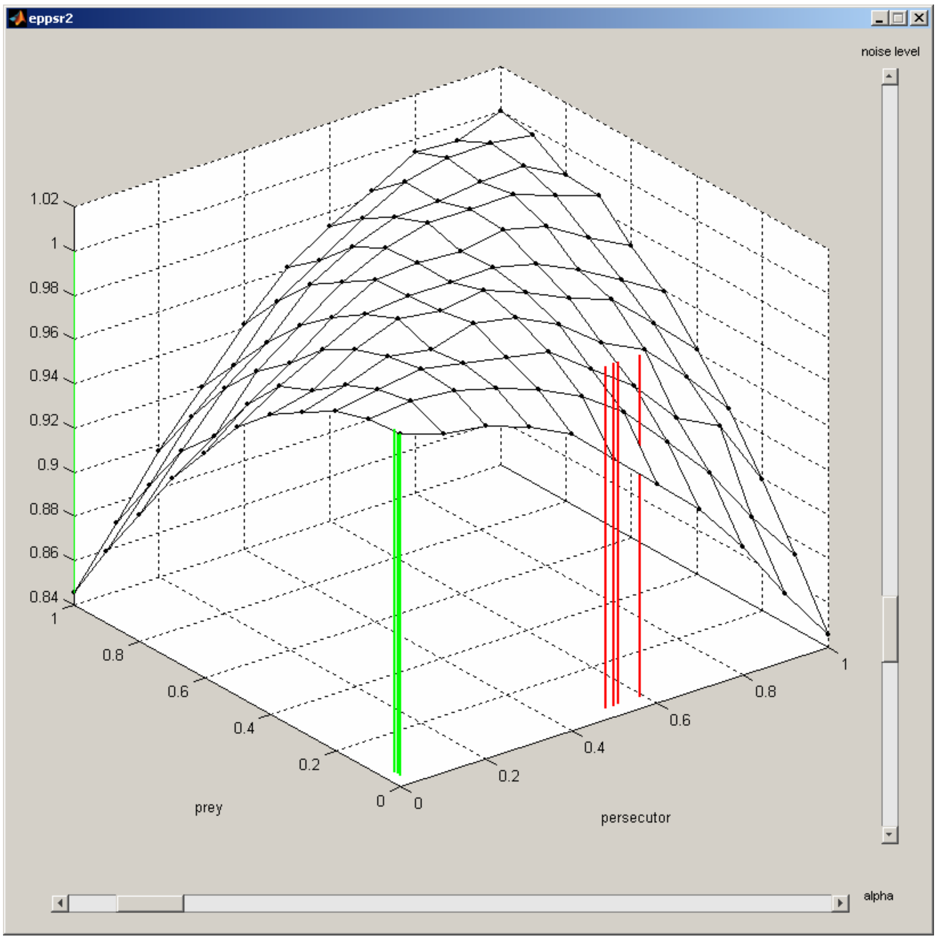


Figure 58. Appearance of the four pure strategies to be selected for each player by further noise level increasing

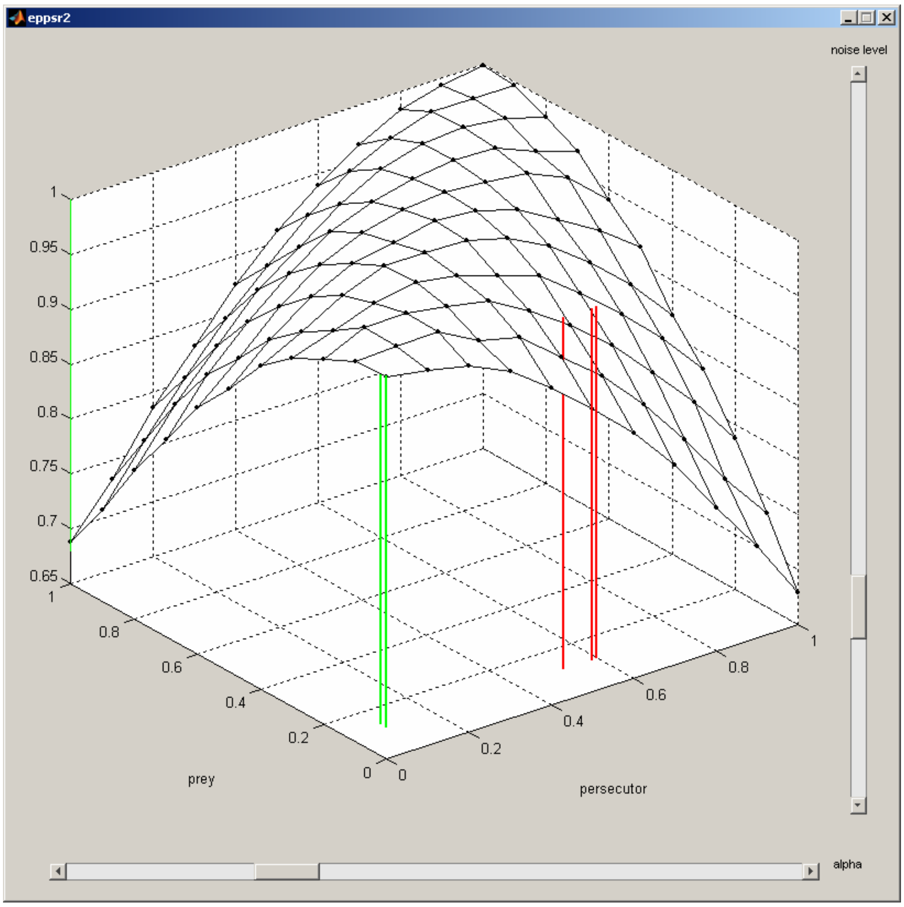


Figure 59. Disappearance of the one pure strategy to be selected for each player by some α increment

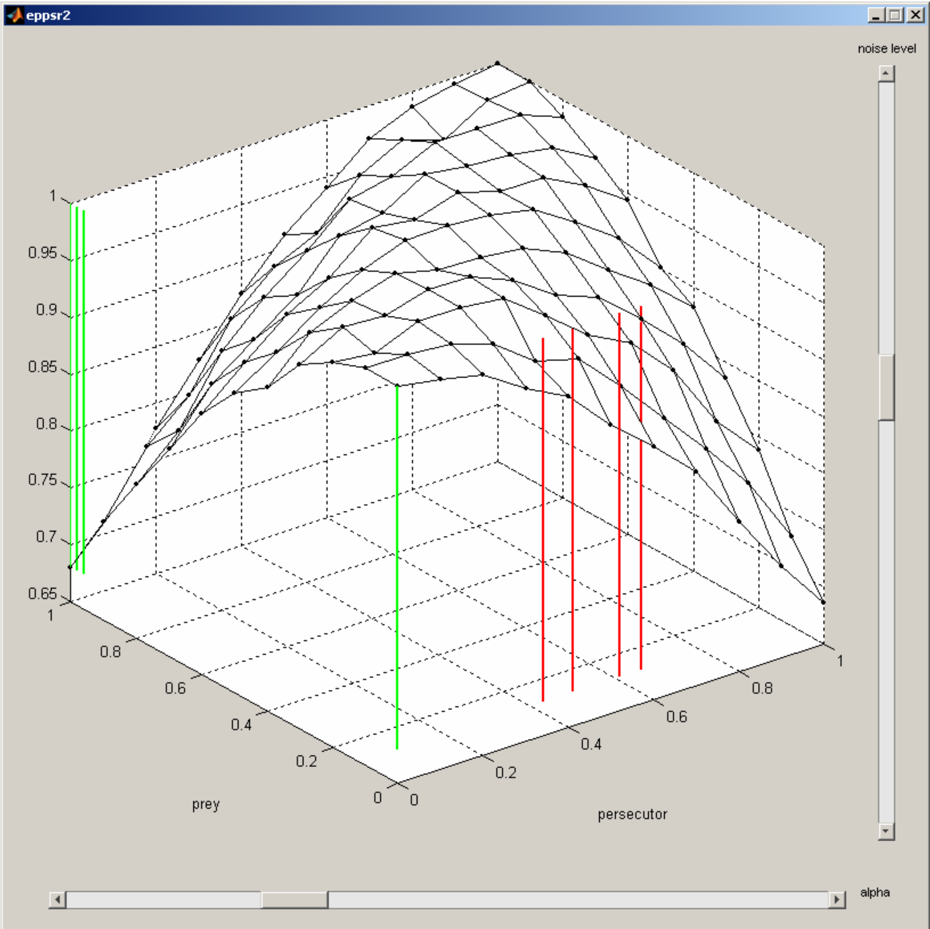


Figure 60. Again each player has the four pure strategies to be selected by further noise level increasing

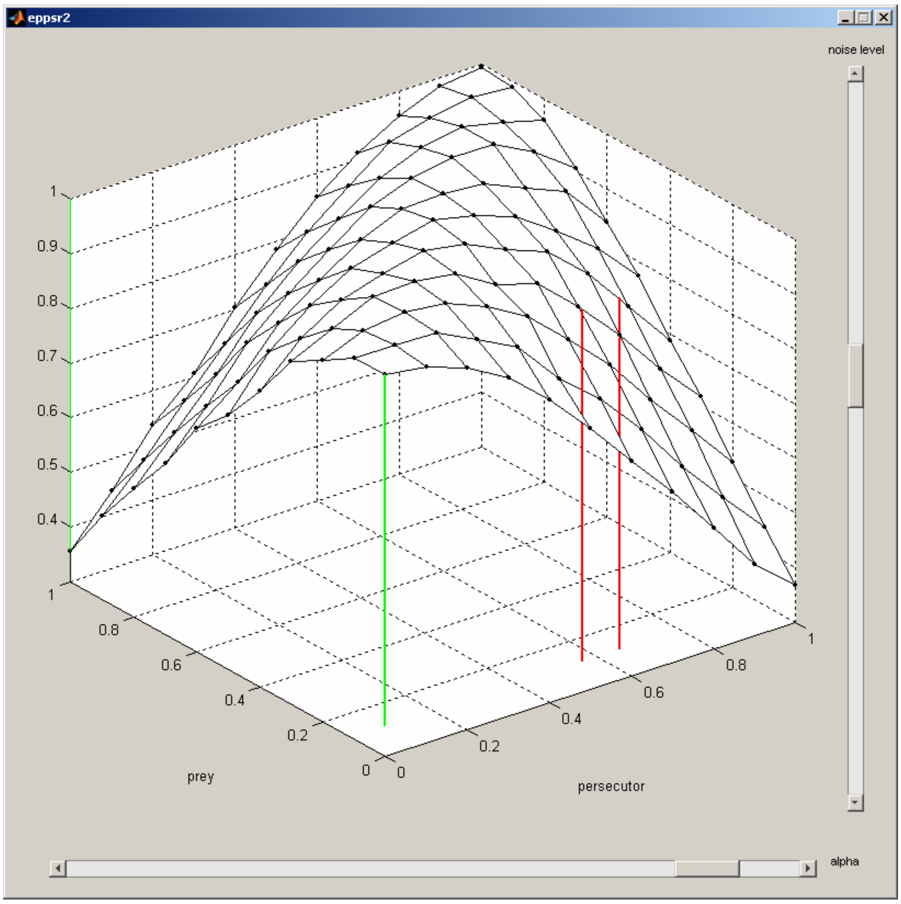


Figure 61. Further increasing α drives to the pure strategies to be selected reduction

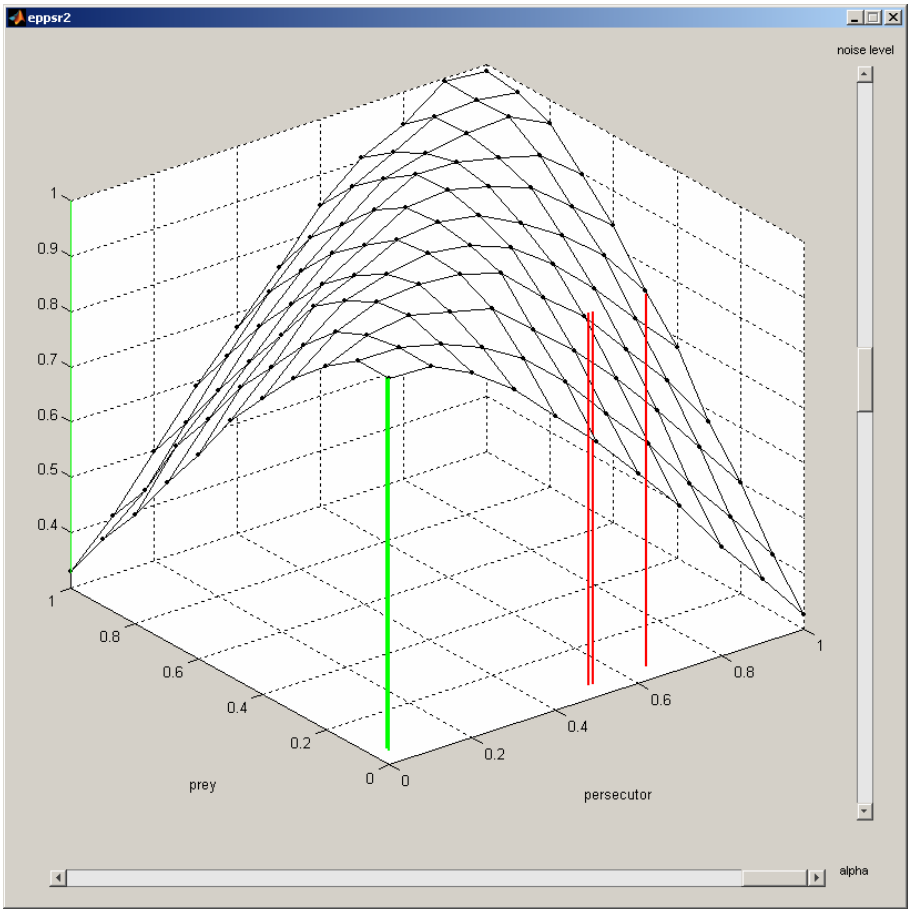


Figure 62. By the maximal α there may be some stratification of the pure strategies to be selected

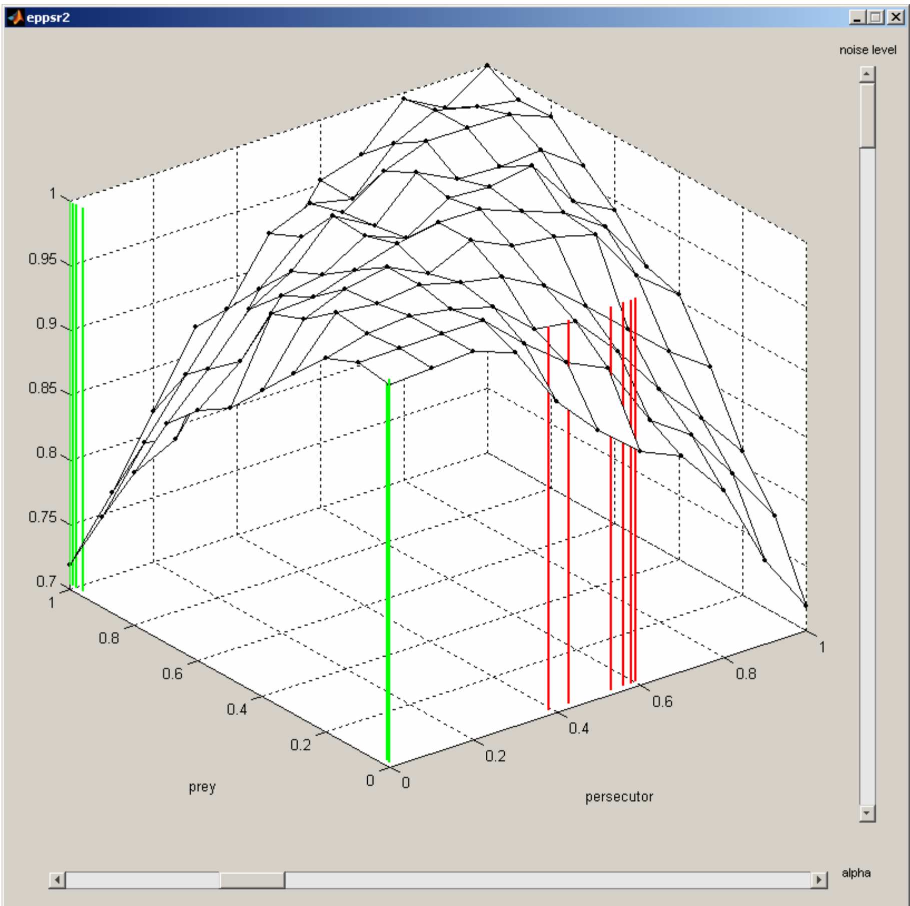


Figure 63. At the maximal noise level and not great α there appear more of the pure strategies to be selected

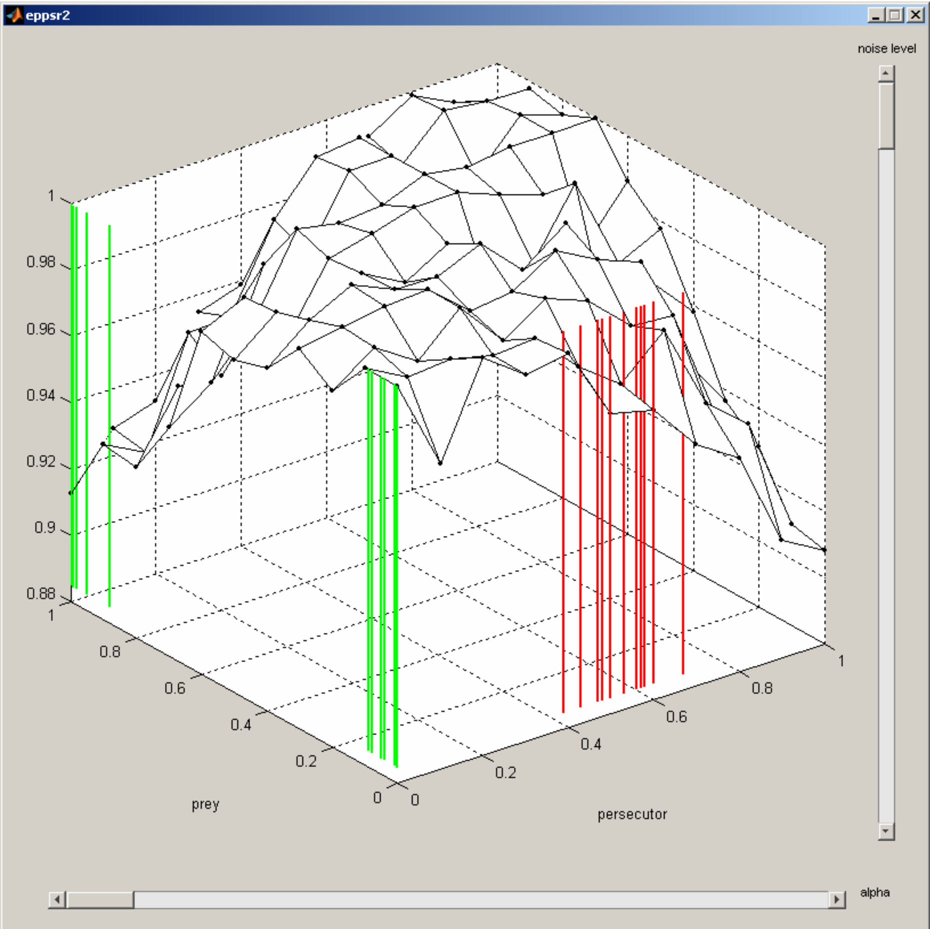


Figure 64. At the maximal noise level and the lowest α the number of the pure strategies to be selected becomes maximal for each player

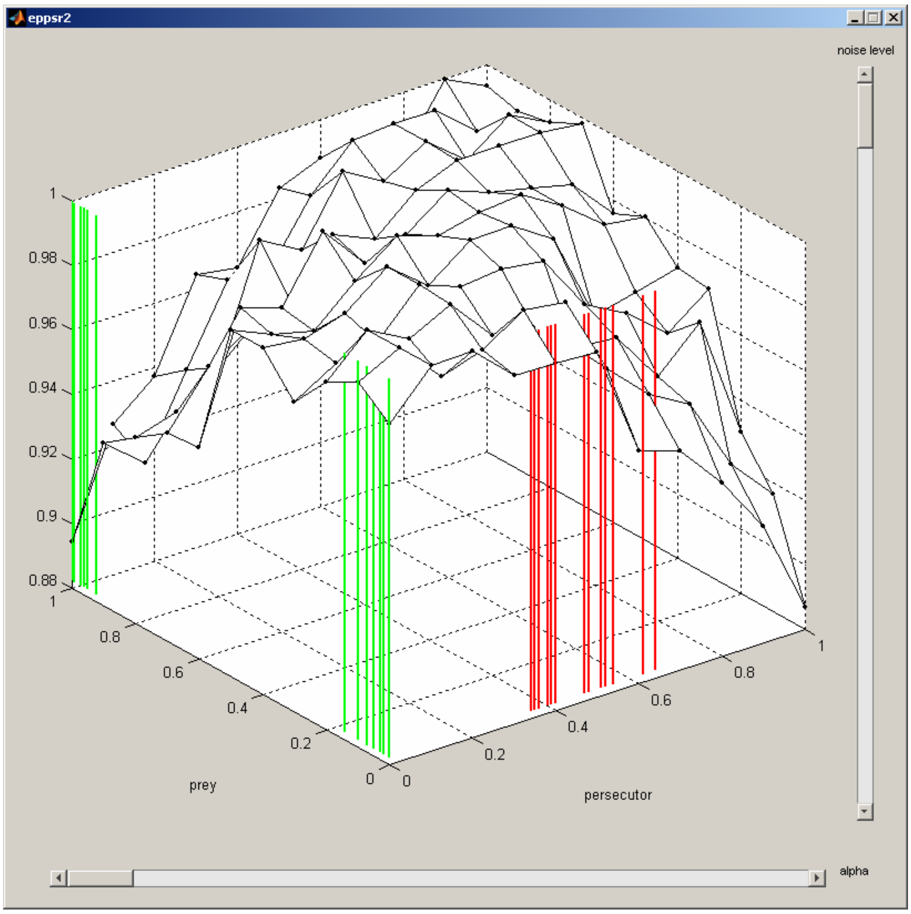


Figure 65. Another example with the maximal noise level and the lowest α

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. Воробьёв 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-грі без сідлової 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 сідлової точки / В. В. Романюк // Вісник НТУ "ХПІ". Тематичний випуск: Інформатика та моделювання. — Харків: НТУ "ХПІ", 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.