UDC 519.832.4

V. V. ROMANUKE

A PROGRAM FUNCTION REPRESENTATION FOR IMPLEMENTING THE METHOD OF PRACTICING THE OPTIMAL MIXED STRATEGY WITH INNUMERABLE SPECTRUM BY UNKNOWN NUMBER OF PLAYS

Khmelnytskyy National University
11, Instytutska street, Khmelnytskyy, Ukraine

Abstract. There has been projected the MATLAB program function for practicing the (optimal) mixed strategy in the antagonistic game with the unknown number of plays, where the player has the innumerable spectrum of its continuous (optimal) mixed strategy to be practiced. The projected program function works with the seven input arguments, including the numerically sampled mixed strategy and the game kernel. The information about the pure strategy to be selected in the current game play is returned into the MATLAB workspace.

Анотація. Спроектовано програмну функцію у MATLAB для практичного застосування (оптимальної) змішаної стратегії в антагоністичній грі з невідомим числом розігрувань, де гравець володіє незліченним спектром своєї неперервної (оптимальної) змішаної стратегії, яку необхідно застосувати на практиці. Спроектована програмна функція працює з сімома вхідними аргументами, включаючи чисельно дискретизовані змішану стратегію та ядро гри. Інформація про чисту стратегію, яку треба обирати у поточному розігруванні гри, повертається у робочу область MATLAB.

Аннотация. Спроектировано программную функцию в MATLAB для практического применения (оптимальной) смешанной стратегии в антагонистической игре с неизвестным числом разыгрываний, где игрок обладает неисчислимым спектром своей непрерывной (оптимальной) смешанной стратегии, которую необходимо применить на практике. Спроектированная программная функция работает с семью входными аргументами, включая численно дискретизированные смешанную стратегию и ядро игры. Информация о чистой стратегии, которую нужно выбирать в текущем разыгрывании игры, возвращается в рабочую область MATLAB.

Keywords: antagonistic game, mixed strategy, MATLAB.

THE PROBLEM PREFACE AND THE PAPER AIM

A lot of socio-economic, ecological and technical processes may be modeled in the form of the antagonistic game, since there always is something about different interests, generated by the participants of the time process. Such math form allows easily predicting, optimizing or equalizing the process, and this trustworthy predictability gives the needful certitude in the future actions for both the participants. For the present day there are some papers on the principles and methods of practicing the optimal mixed strategy in the antagonistic game, including the strategy with its innumerable spectrum [1]. The developed methods deal with the algebraic statements, needing to be upgraded under the contemporary math software. And as the continuous antagonistic games are the models of the more general conflict processes with the two participants, then the consequent paper aim is to implement programmatically the method of practicing the optimal mixed strategy with innumerable spectrum, when the number of the game plays is unknown.

PROJECTING THE FRAGMENTS OF THE CODE FOR IMPLEMENTING THE METHOD WITHIN A MATH SOFTWARE

May there be the antagonistic game with the kernel $W(x, y)$, where one of the players has the innumerable set of its pure strategies. Suppose that this player has the continuous set $Z = [0; 1]$ of the pure strategies, and has the optimal mixed strategy $f_{\mathrm{opt}}(z)$ as the continuous function of the pure strategy $z \in Z$, satisfying the condition

$$\int_Z f_{\text{opt}}(z)\,dz = \int_0^1 f_{\text{opt}}(z)\,dz = 1 \tag{1}$$

and having the innumerable spectrum $\operatorname{supp} f_{\text{opt}}(z)$. Due to the paper [1], before breaking the segment $Z = [0;\,1]$ into such equal K halfsegments, that the function $f_{\text{opt}}(z)$ within every halfsegment would be approximately constant, there should be fulfilled the conditions:

$$\frac{df_{\text{opt}}(z)}{dz} \geqslant 0 \text{ or } \frac{df_{\text{opt}}(z)}{dz} \leqslant 0 \;\; \forall\, z \in [z_{k-1};\, z_k) \text{ by } k = \overline{1,\, K}, \tag{2}$$

where $z_0 \equiv 0$ and $z_K \equiv 1$;

$$\left| \frac{df_{\text{opt}}(z)}{dz} \right| \leqslant a \;\; \forall\, z \in [z_{k-1};\, z_k) \text{ by } k = \overline{1,\, K}, \tag{3}$$

where the parameter a is the inconstancy tolerance, taken as $a \in (0.001;\, 0.1]$. The conditions of the smoothness and tolerated inconstancy of the game kernel $W(x,\, y)$ lie in the following two conditions. If the opposite player is the second, having also the unit segment $Y = [0;\, 1]$ of the pure strategies, then there should be

$$\frac{\partial^2 W(x,\, y)}{\partial x \partial y} \geqslant 0 \text{ or } \frac{\partial^2 W(x,\, y)}{\partial x \partial y} \leqslant 0 \;\; \forall\, x \in [x_{k-1};\, x_k) \text{ by } k = \overline{1,\, K} \text{ and } \forall\, y \in [0;\, 1]. \tag{4}$$

But for the discrete finite set $Y = \{y_j\}_{j=1}^{N}$ by $N \in \mathbb{N} \setminus \{1\}$ the condition (4) should be stated as

$$\frac{\partial W(x,\, y_j)}{\partial x} - \frac{\partial W(x,\, y_{j-1})}{\partial x} \geqslant 0 \text{ or } \frac{\partial W(x,\, y_j)}{\partial x} - \frac{\partial W(x,\, y_{j-1})}{\partial x} \leqslant 0$$

$$\forall\, x \in [x_{k-1};\, x_k) \text{ by } k = \overline{1,\, K} \text{ and } j = \overline{2,\, N}. \tag{5}$$

The second condition is the kernel inconstancy tolerance, which for $Y = [0;\, 1]$ is

$$\left| \frac{\partial^2 W(x,\, y)}{\partial x \partial y} \right| \leqslant b \;\; \forall\, x \in [x_{k-1};\, x_k) \text{ by } k = \overline{1,\, K} \text{ and } \forall\, y \in [0;\, 1], \tag{6}$$

and for $Y = \{y_j\}_{j=1}^{N}$ this condition is

$$\left| \frac{\partial W(x,\, y_j)}{\partial x} - \frac{\partial W(x,\, y_{j-1})}{\partial x} \right| \leqslant b \;\; \forall\, x \in [x_{k-1};\, x_k) \text{ by } k = \overline{1,\, K} \text{ and } j = \overline{2,\, N}, \tag{7}$$

where the parameter b is the kernel inconstancy tolerance, could be taken as $b \in (0.001;\, 0.1]$.

Obviously, that if the continuous set $Z = [0;\, 1]$ of the pure strategies is the second player set, then the opposite player is the first, and the conditions (4) — (7) are rewritten as the following:

$$\frac{\partial^2 W(x,\, y)}{\partial x \partial y} \geqslant 0 \text{ or } \frac{\partial^2 W(x,\, y)}{\partial x \partial y} \leqslant 0 \;\; \forall\, y \in [y_{k-1};\, y_k) \text{ by } k = \overline{1,\, K} \text{ and } \forall\, x \in [0;\, 1] \tag{8}$$

for the case of the continuous set $X = [0;\, 1]$ of the first player, and

$$\frac{\partial W(x_i, y)}{\partial y} - \frac{\partial W(x_{i-1}, y)}{\partial y} \geqslant 0 \text{ or } \frac{\partial W(x_i, y)}{\partial y} - \frac{\partial W(x_{i-1}, y)}{\partial y} \leqslant 0$$

$$\forall\ y \in [y_{k-1}; y_k) \text{ by } k = \overline{1, K} \text{ and } i = \overline{2, M} \tag{9}$$

for the discrete finite set $X = \{x_i\}_{i=1}^{M}$ by $M \in \mathbb{N} \setminus \{1\}$; the kernel inconstancy tolerance condition is

$$\left| \frac{\partial^2 W(x, y)}{\partial x \partial y} \right| \leqslant b \ \ \forall\ y \in [y_{k-1}; y_k) \text{ by } k = \overline{1, K} \text{ and } \forall\ x \in [0; 1] \tag{10}$$

for $X = [0; 1]$ and

$$\left| \frac{\partial W(x_i, y)}{\partial y} - \frac{\partial W(x_{i-1}, y)}{\partial y} \right| \leqslant b \ \ \forall\ y \in [y_{k-1}; y_k) \text{ by } k = \overline{1, K} \text{ and } i = \overline{2, M} \tag{11}$$

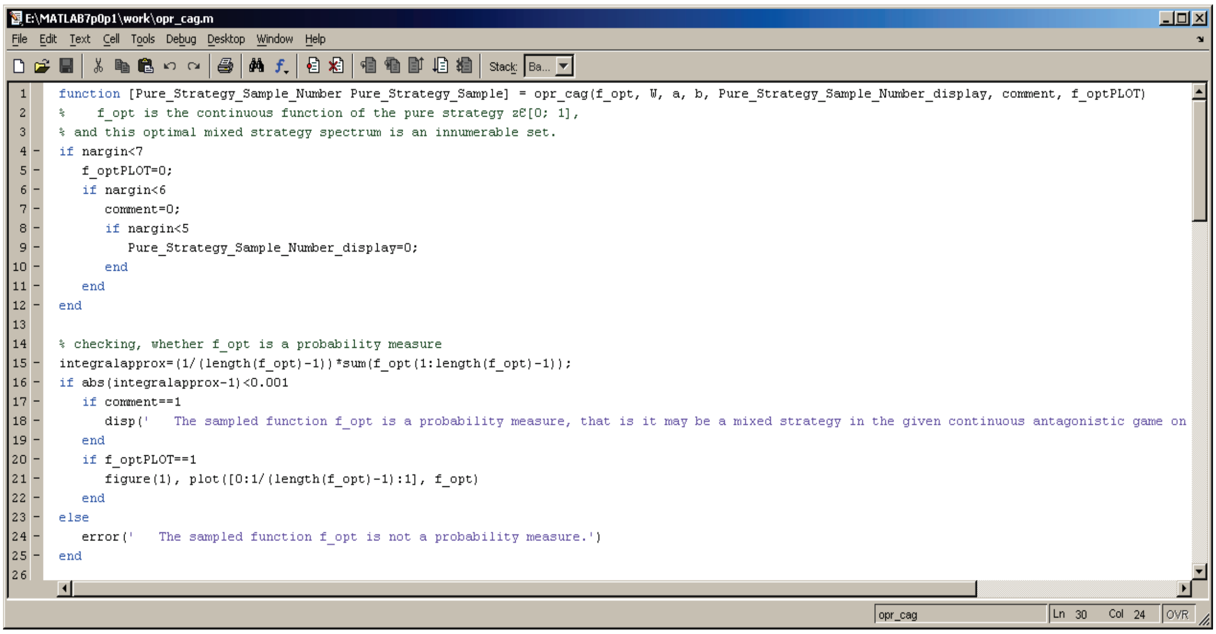
for $X = \{x_i\}_{i=1}^{M}$.

Any of the stated conditions (2) — (11) may be supported only in numerical way. One of the most grandiose numerical math applications is the registered trademark MATLAB. In this system there are boundless opportunities for coding the program functions to be run right from the command line in the MATLAB Command Window. The projected program function, been named “opr_cag”, has the seven input arguments (figure 1, line 1), each of which implies correspondingly the numerically sampled function $f_{\text{opt}}(z)$, the numerically sampled kernel $W(x, y)$, parameters a and b, the option for whether the sampled pure strategy number should be displayed, the comment option, and the option on whether the primarily sampled function $f_{\text{opt}}(z)$ should be plotted in a separate figure window. The last three options are not necessary for being typed, as by default they are not checked (figure 1, line 4 — 12).

For not working in vain with the wrong sampled data, the lines 15 — 25 (figure 1) of the code are assigned for verifying whether the primarily sampled function $f_{\text{opt}}(z)$ is a mixed strategy, that is the condition (1) is verified numerically. If L is the number of the sampled points $\{f_{\text{opt}}(z_l)\}_{l=1}^{L}$ in $f_{\text{opt}}(z)$ by the constant difference $z_l - z_{l-1}\ \ \forall\ l = \overline{2, L}$, then the approximate value

$$\frac{1}{L-1} \sum_{l=1}^{L-1} f_{\text{opt}}(z_l) \tag{12}$$

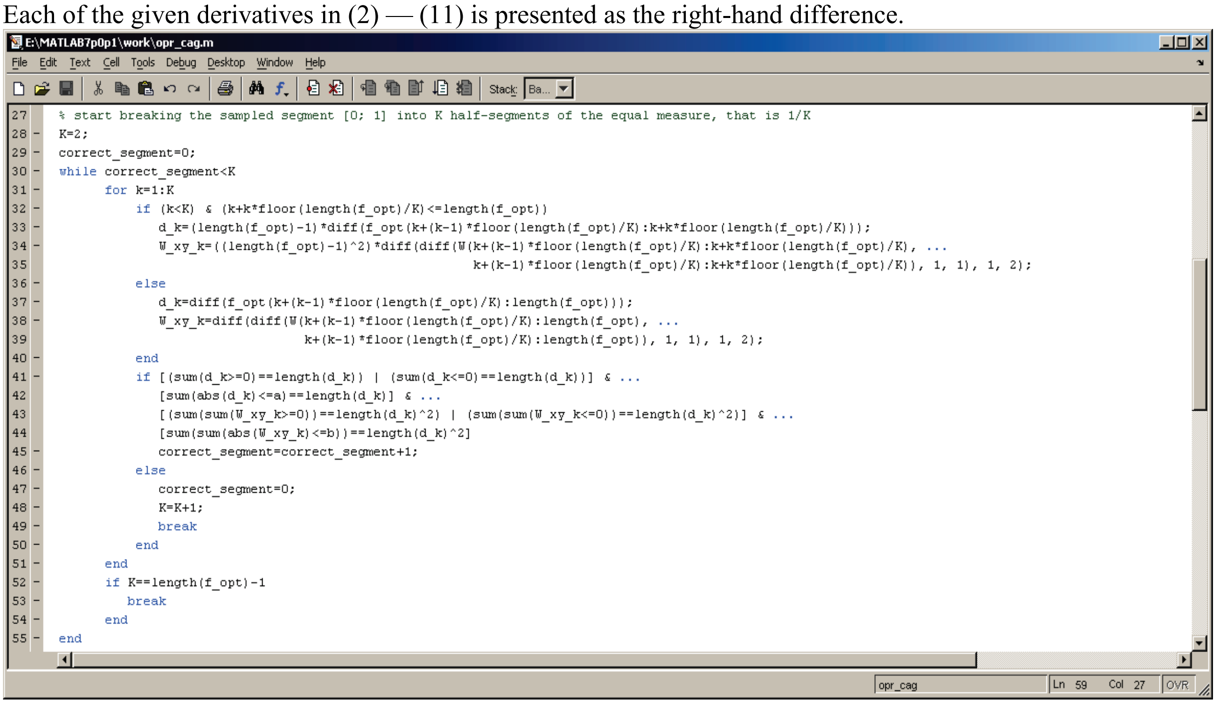
of the integral (1) mustn’t be further from the unity than for 0.001 (line 16). In the case, when it is not held true, then there is displayed the error message (line 24) and the code running stops.



```
E:\MATLAB7p0p1\work\opr_cag.m
1   function [Pure_Strategy_Sample_Number Pure_Strategy_Sample] = opr_cag(f_opt, W, a, b, Pure_Strategy_Sample_Number_display, comment, f_optPLOT)
2   %    f_opt is the continuous function of the pure strategy z∈[0; 1],
3   % and this optimal mixed strategy spectrum is an innumerable set.
4   if nargin<7
5      f_optPLOT=0;
6      if nargin<6
7         comment=0;
8         if nargin<5
9            Pure_Strategy_Sample_Number_display=0;
10        end
11     end
12  end
13
14  % checking, whether f_opt is a probability measure
15  integralapprox=(1/(length(f_opt)-1))*sum(f_opt(1:length(f_opt)-1));
16  if abs(integralapprox-1)<0.001
17     if comment==1
18        disp('   The sampled function f_opt is a probability measure, that is it may be a mixed strategy in the given continuous antagonistic game on
19     end
20     if f_optPLOT==1
21        figure(1), plot([0:1/(length(f_opt)-1):1], f_opt)
22     end
23  else
24     error('   The sampled function f_opt is not a probability measure.')
25  end
26
```

Fig.1. Beginning of the code with option on $f_{\text{opt}}\left(z\right)$ plot in a separate figure window

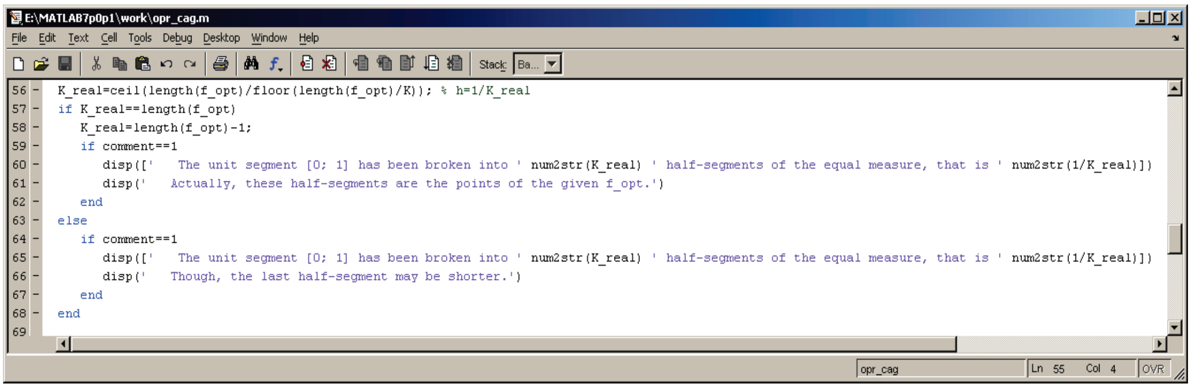
The conditions (2) — (11) are run in the lines 28 — 55 of the code (figure 2), where actually it is determined the number K of the equal measure halfsegments with the pure strategies to be selected further. Each of the given derivatives in (2) — (11) is presented as the right-hand difference.



```
E:\MATLAB7p0p1\work\opr_cag.m
27  % start breaking the sampled segment [0; 1] into K half-segments of the equal measure, that is 1/K
28  K=2;
29  correct_segment=0;
30  while correct_segment<K
31        for k=1:K
32           if (k<K) & (k+k*floor(length(f_opt)/K)<=length(f_opt))
33              d_k=(length(f_opt)-1)*diff(f_opt(k+(k-1)*floor(length(f_opt)/K):k+k*floor(length(f_opt)/K)));
34              W_xy_k=((length(f_opt)-1)^2)*diff(diff(W(k+(k-1)*floor(length(f_opt)/K):k+k*floor(length(f_opt)/K), ...
35                                          k+(k-1)*floor(length(f_opt)/K):k+k*floor(length(f_opt)/K)), 1, 1), 1, 2);
36           else
37              d_k=diff(f_opt(k+(k-1)*floor(length(f_opt)/K):length(f_opt)));
38              W_xy_k=diff(diff(W(k+(k-1)*floor(length(f_opt)/K):length(f_opt), ...
39                                 k+(k-1)*floor(length(f_opt)/K):length(f_opt)), 1, 1), 1, 2);
40           end
41           if [(sum(d_k>=0)==length(d_k)) | (sum(d_k<=0)==length(d_k))] & ...
42              [sum(abs(d_k)<=a)==length(d_k)] & ...
43              [(sum(sum(W_xy_k>=0))==length(d_k)^2) | (sum(sum(W_xy_k<=0))==length(d_k)^2)] & ...
44              [sum(sum(abs(W_xy_k)<=b))==length(d_k)^2]
45              correct_segment=correct_segment+1;
46           else
47              correct_segment=0;
48              K=K+1;
49              break
50           end
51        end
52        if K==length(f_opt)-1
53           break
54        end
55  end
```

Fig.2. The conditions (2) — (11) code part

Finally the real number K is found as the following: the fraction $\dfrac{L}{K}$ is rounded towards minus infinity, and then the number L is divided by this result; the last result is rounded towards plus infinity (figure 3, line 56). If the comment option was set on, then the accompanying message about the real number K will appear in the running MATLAB Command Window (figure 3, lines 57 — 68).


E:\MATLAB7p0p1\work\opr_cag.m

```
K_real=ceil(length(f_opt)/floor(length(f_opt)/K)); % h=1/K_real
if K_real==length(f_opt)
   K_real=length(f_opt)-1;
   if comment==1
      disp(['   The unit segment [0; 1] has been broken into ' num2str(K_real) ' half-segments of the equal measure, that is ' num2str(1/K_real)])
      disp('   Actually, these half-segments are the points of the given f_opt.')
   end
else
   if comment==1
      disp(['   The unit segment [0; 1] has been broken into ' num2str(K_real) ' half-segments of the equal measure, that is ' num2str(1/K_real)])
      disp('   Though, the last half-segment may be shorter.')
   end
end
```

Fig.3. Code part with messaging

After having got the real number K the player applies the set of its K pure strategies

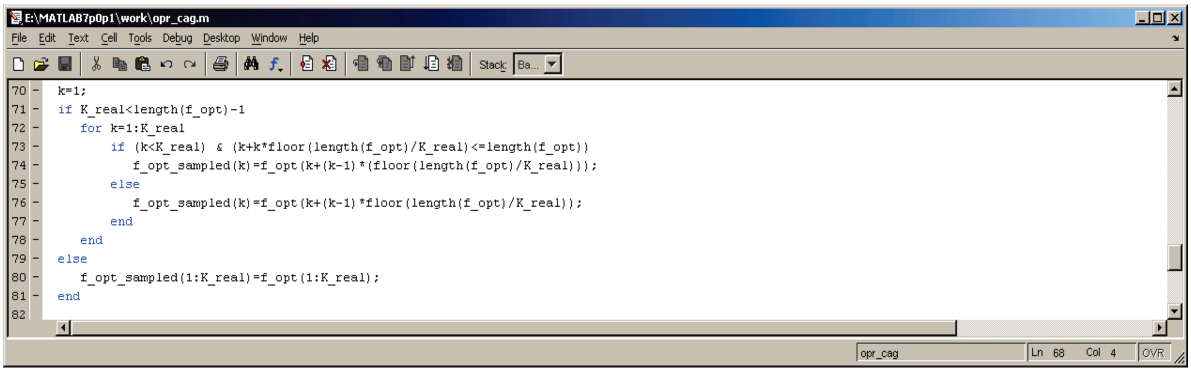
$$\left\{z_{k-1}\right\}_{k=1}^{K}=\left\{0,\,\left\{z_k\right\}_{k=1}^{K-1}\right\}=\left\{0,\,z_1,\,z_2,\,\ldots,\,z_{K-2},\,z_{K-1}\right\} \tag{13}$$

with the corresponding probabilities

$$\left\{f_{\text{opt}}\left(z_{k-1}\right)h\right\}_{k=1}^{K}=\left\{f_{\text{opt}}\left(0\right)h,\,\left\{f_{\text{opt}}\left(z_k\right)h\right\}_{k=1}^{K-1}\right\}=$$

$$=\left\{f_{\text{opt}}\left(0\right)h,\,f_{\text{opt}}\left(z_1\right)h,\,f_{\text{opt}}\left(z_2\right)h,\,\ldots,\,f_{\text{opt}}\left(z_{K-2}\right)h,\,f_{\text{opt}}\left(z_{K-1}\right)h\right\}, \tag{14}$$

being taken out in the code lines 70 — 81 (figure 4), where the constant sampling space is $h=z_k-z_{k-1}$ by $k=\overline{1,\,K-1}$.


E:\MATLAB7p0p1\work\opr_cag.m

```
70   k=1;
71   if K_real<length(f_opt)-1
72      for k=1:K_real
73          if (k<K_real) & (k+k*floor(length(f_opt)/K_real)<=length(f_opt))
74             f_opt_sampled(k)=f_opt(k+(k-1)*(floor(length(f_opt)/K_real)));
75          else
76             f_opt_sampled(k)=f_opt(k+(k-1)*floor(length(f_opt)/K_real));
77          end
78      end
79   else
80      f_opt_sampled(1:K_real)=f_opt(1:K_real);
81   end
82
```

Fig.4. Code part with taking out the probabilities (14)

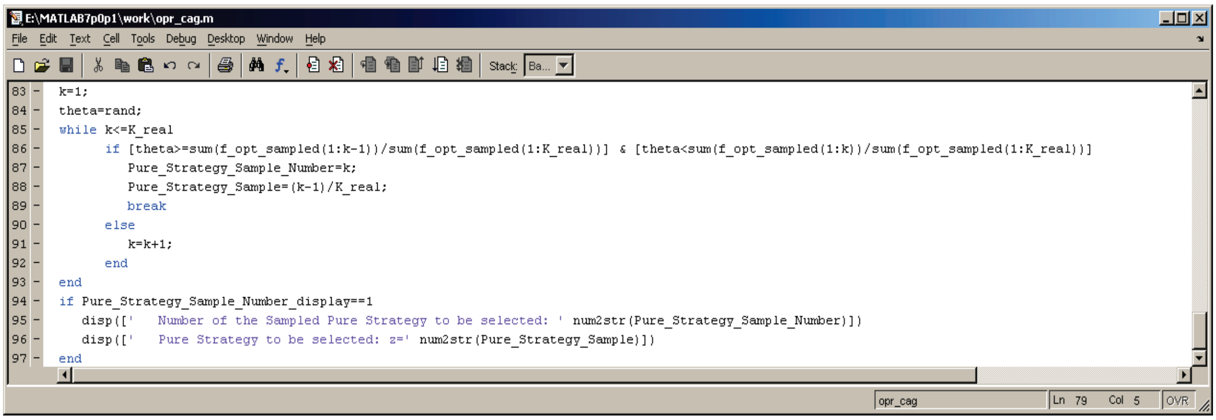
After having sampled the pure strategies set (13) and taken the probabilities (14), the player raffles the single uniformly distributed on the semisegment $[0;\,1)$ variate Θ (figure 5, line 84) with the value θ [2 — 5]. The pure strategy z_{u-1} is selected by the nonstrict inequality

$$\theta\geqslant\frac{\sum\limits_{k=1}^{u-1}f_{\text{opt}}\left(z_{k-1}\right)}{\sum\limits_{k=1}^{K}f_{\text{opt}}\left(z_{k-1}\right)} \tag{15}$$

and the strict inequality

$$\theta < \frac{\sum_{k=1}^{u} f_{\text{opt}}\left(z_{k-1}\right)}{\sum_{k=1}^{K} f_{\text{opt}}\left(z_{k-1}\right)} \tag{16}$$

for $u \in \left\{\overline{1, K}\right\}$, what is fulfilled in the lines 85 — 93 on the figure 5. If the fifth option was set on, then the message about the sampled pure strategy number and its value will appear in the running MATLAB Command Window (figure 5, lines 94 — 97). In the case of an appropriate giving the program function “opr_cag” its output, into the MATLAB current workspace there is returned the sampled pure strategy number and its value.



```
83 - k=1;
84 - theta=rand;
85 - while k<=K_real
86 -     if [theta>=sum(f_opt_sampled(1:k-1))/sum(f_opt_sampled(1:K_real))] & [theta<sum(f_opt_sampled(1:k))/sum(f_opt_sampled(1:K_real))]
87 -        Pure_Strategy_Sample_Number=k;
88 -        Pure_Strategy_Sample=(k-1)/K_real;
89 -        break
90 -     else
91 -        k=k+1;
92 -     end
93 - end
94 - if Pure_Strategy_Sample_Number_display==1
95 -    disp(['   Number of the Sampled Pure Strategy to be selected: ' num2str(Pure_Strategy_Sample_Number)])
96 -    disp(['   Pure Strategy to be selected: z=' num2str(Pure_Strategy_Sample)])
97 - end
```

Fig.5. Fulfilling the check of conditions (15) and (16)

Consider an example of applying the projected program function. May there be the continuous game on the unit square $[0; 1]\times[0; 1]$ with the kernel, imaged on the figure 6.

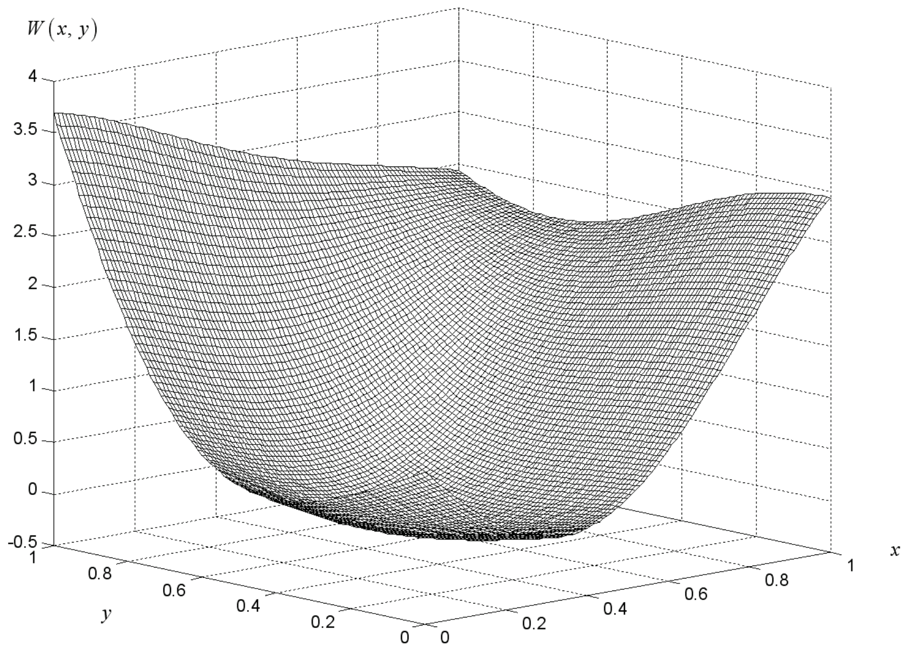


Fig.6. Example of a kernel

And may here the first player, having the mixed strategy $p_{\text{opt}}(x)$ as an example of the optimal one (though it is almost certainly nonoptimal, as it is seen from the figure 7; but the first player will use it nevertheless just as an element of its set of all the mixed strategies), applies the method of practicing the optimal mixed strategy under the program function “opr_cag”.

In the example the first input for “opr_cag” is the vector of the 101 elements (primarily sampled mixed strategy), and the second is the 101×101 -size matrix (the sampled kernel). The requirements to the smoothness or inconstancy are left minimal, that is $a = b = 0.1$ (figure 8, the line to run). For those initial conditions the five launches of “opr_cag” are displayed on the figures 9 — 13.

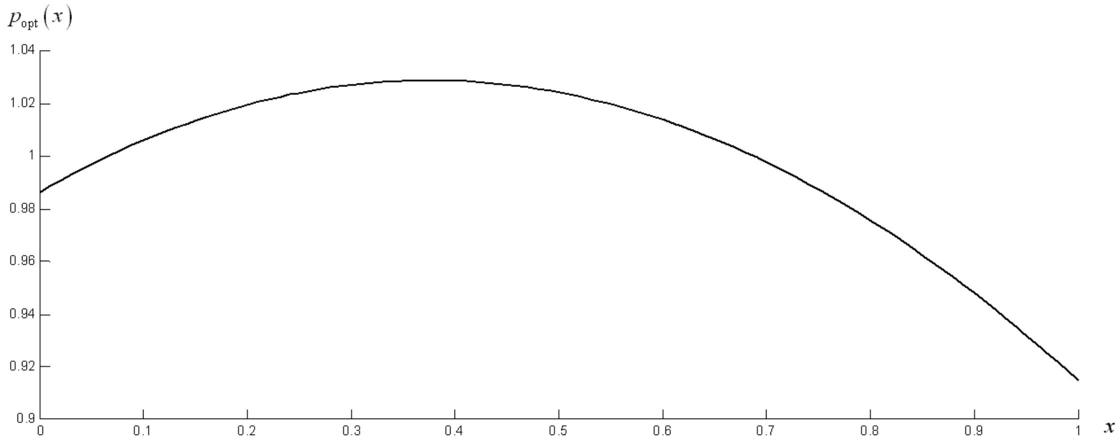


Fig.7. Example of an optimal strategy

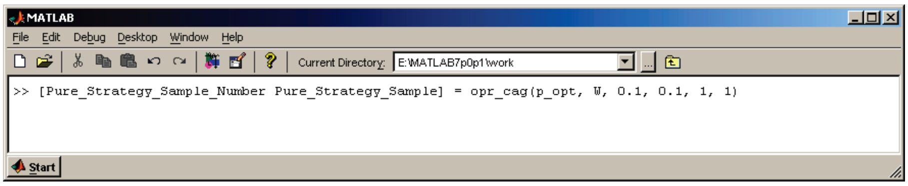


Fig.8. Launching “opr_cag”

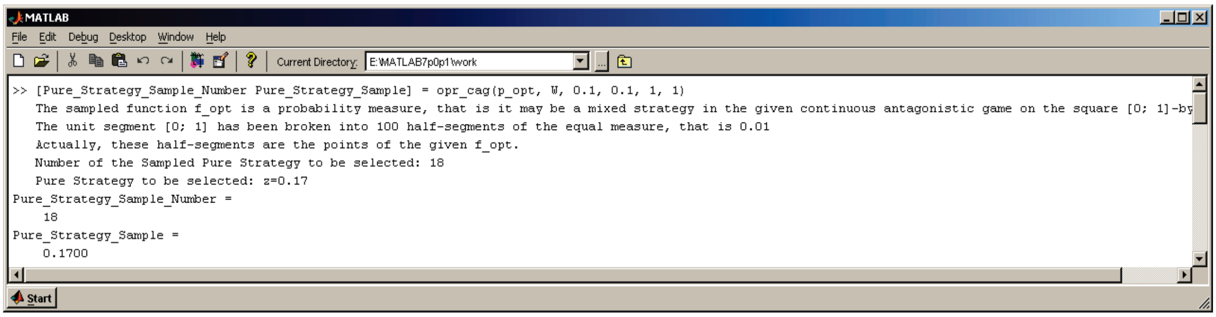


Fig.9. Results after launched “opr_cag” with the 18-th sampled pure strategy to be selected

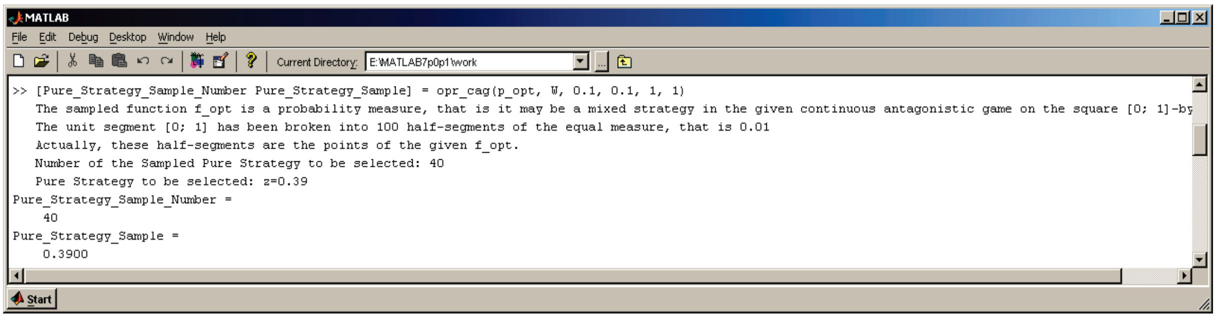


Fig.10. Results after launched “opr_cag” with the 40-th sampled pure strategy to be selected

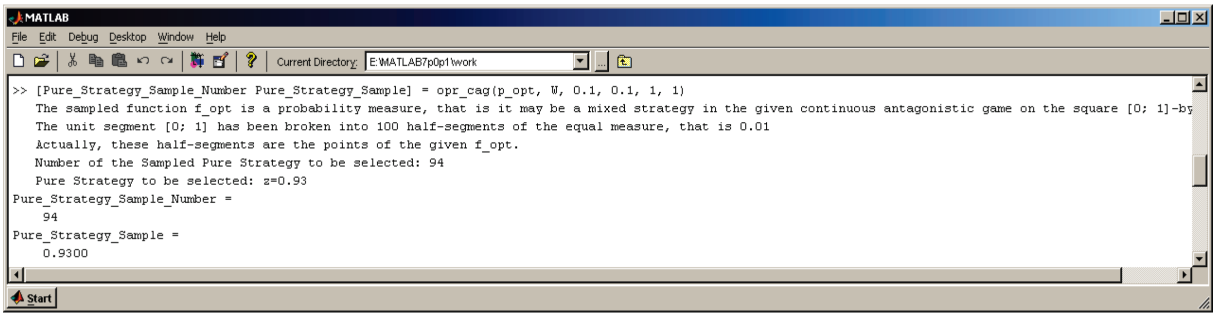


Fig.11. Results after launched “opr_cag” with the 94-th sampled pure strategy to be selected

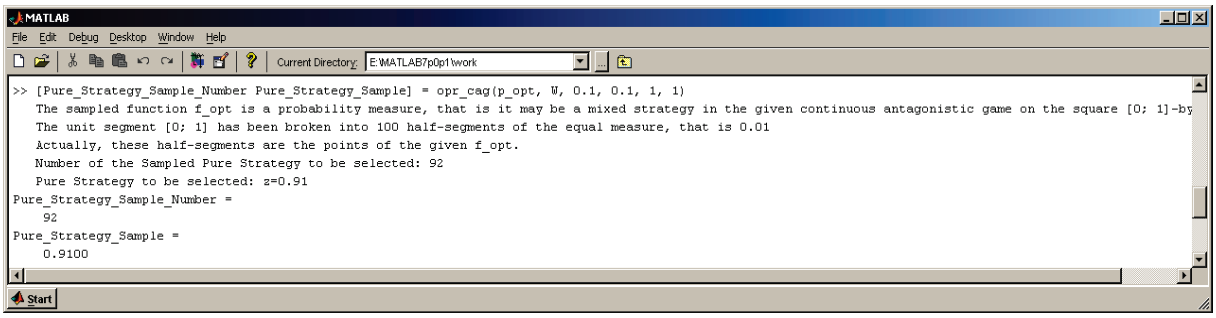


Fig.12. Results after launched “opr_cag” with the 92-nd sampled pure strategy to be selected

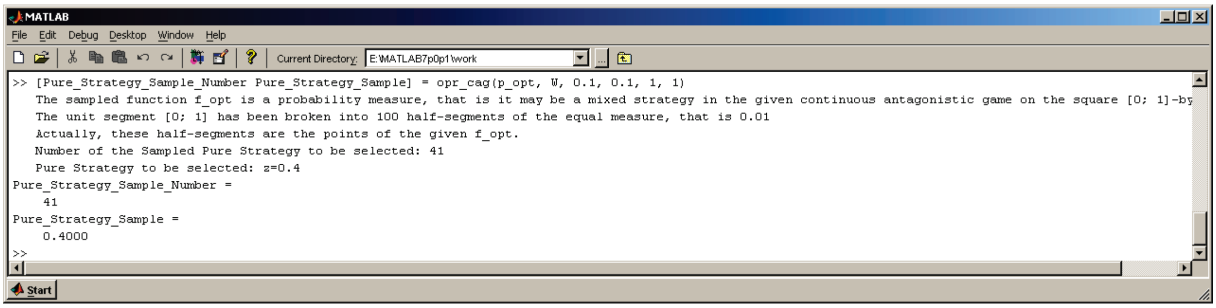


Fig.13. Results after launched “opr_cag” with the 41-st sampled pure strategy to be selected

CONCLUSION ON THE PROJECTED PROGRAM FUNCTION AND AN OUTLOOK FOR THE FURTHER PROJECTS

The projected program function, having implemented the method of practicing the optimal mixed strategy with innumerable spectrum in the MATLAB environment, may be applied only when the number of the game plays is unknown, or when it is unknown when the game stops. Speaking generally, there may be not only the optimal mixed strategy, but any of the mixed strategies of the player, though there are some troubles to check numerically the optimality of the strategy $f_{\text{opt}}(z)$ by the given kernel $W(x, y)$, and so this check was not inserted into the code for now. Having the installed MATLAB, a player define all the needful arguments for the function “opr_cag” and selects one of its pure strategies from the segment $[0; 1]$ due to the returned pure strategy number and its value into the workspace. It is important to mark, that instead of the pure strategies set (13) and probabilities (14) a player may apply the sets

$$\{z_k\}_{k=1}^{K} = \left\{\{z_k\}_{k=1}^{K-1},\, 1\right\} = \{z_1,\, z_2,\, \ldots,\, z_{K-2},\, z_{K-1},\, 1\} \tag{17}$$

and

$$\left\{f_{\text{opt}}(z_k)h\right\}_{k=1}^{K} = \left\{\left\{f_{\text{opt}}(z_k)h\right\}_{k=1}^{K-1},\, f_{\text{opt}}(1)h\right\} =$$

$$=\left\{f_{\mathrm{opt}}\left(z_1\right)h,\ f_{\mathrm{opt}}\left(z_2\right)h,\ \ldots,\ f_{\mathrm{opt}}\left(z_{K-2}\right)h,\ f_{\mathrm{opt}}\left(z_{K-1}\right)h,\ f_{\mathrm{opt}}\left(1\right)h\right\} \quad (18)$$

correspondingly. If to turn at the case, when the number of the future game plays is certainly known, then the base method of practicing the mixed strategy [1] fits only partially, as there each player will attempt to develop own tactics of the pure strategies selection [5 — 8] for ensuring the most great own payoff. Consequently, an outlook for the further math software projects is for adapting the developed program function “opr_cag” for to use it in the antagonistic game with the preliminarily known number of the future plays [9 — 16].

REFERENCES

1. Romanuke V. V. Method of practicing the optimal mixed strategy with innumerable set in its spectrum by unknown number of plays / V. V. Romanuke // Measuring and Computing Devices in Technological Processes. — 2008. — № 2. — P. 196 — 203.
2. 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.
3. 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.
4. Романюк В. В. Метод реалізації оптимальних змішаних стратегій у матричній грі з пустою множиною сідлових точок у чистих стратегіях з невідомою кількістю партій гри / В. В. Романюк // Вісник Хмельницького національного університету. Технічні науки. — 2009. — № 2. — С. 224 — 229.
5. 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.
6. Романюк В. В. Про порядок перебору чистих стратегій в одній матричній грі без сідлової точки для реалізації оптимальних змішаних стратегій / В. В. Романюк // Материалы II Международной научно-практической конференции “Ключевые аспекты научной деятельности — 2007”. Том 7. Естественные науки. — Днепропетровск: Наука и образование, 2007. — С. 12 — 14.
7. Романюк В. В. Формулювання одного з принципів оптимальності в елементарній антагоністичній грі без сідлової точки при неповній реалізації оптимальних змішаних стратегій / В. В. Романюк // Вісник Хмельницького національного університету. Технічні науки. — 2007. — № 2. — Т. 2. — С. 218 — 222.
8. 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.
9. 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.
10. 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.
11. 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.
12. Романюк В. В. Оптимізація кількості варіантів відповіді у закритих тестах з фіксованим часом за допомогою матричної гри / В. В. Романюк // Вісник Хмельницького національного університету. Технічні науки. — 2009. — № 3. — С. 187 — 192.
13. Романюк В. В. Про рівнозначність оптимальних змішаних стратегій другого гравця у вгнутій антагоністичній грі з експоненціальним ядром на одиничному гіперкубі чотиривимірного евклідового простору / В. В. Романюк // Вісник Хмельницького національного університету. Економічні науки. — 2009. — № 2. — Т. 1. — С. 113 — 121.
14. Романюк В. В. Нерівнозначні оптимальні змішані стратегії другого гравця у вгнутій антагоністичній грі з експоненціальним ядром, що задається на декартовому добутку двох одиничних кубів / В. В. Романюк // Науково-теоретичний журнал Хмельницького економічного університету “Наука й економіка”. — Випуск 3 (15), 2009. — Том 2. — С. 206 — 234.
15. Романюк В. В. Комплексне програмне забезпечення для визначення оптимальної поведінки у

конкурентних процесах з визначеними на одиничному гіперкубі простору $\mathbb{R}^4$ експоненціальними платіжними функціями / В. В. Романюк // Вісник Хмельницького національного університету. Економічні науки. — 2009. — № 2. — Т. 2. — С. 188 — 193.

16. Romanuke V. V. Optimality control in the concave antagonistic game with annihilation probability payoff function as the kernel on the unit hypercube of the six-dimensional arithmetic space / V. V. Romanuke // Информационно-вычислительные технологии и их приложения: сборник статей X Международной научно-технической конференции. — Пенза: РИО ПГСХА, 2009. — С. 236 — 241.

Надійшла до редакції 12.02.2010р.

ROMANUKE V. V. — candidate of technical science, associate professor of the Applied Mathematics and Social Informatics Department, Khmelnytskyy National University, Khmelnytskyy, Ukraine.