EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 5, Issue 2 (2017) 4 – 13

HIGH-PERFORMANCE COMPUTING OF OIL RECOVERY PROBLEM ON A MOBILE PLATFORM USING CUDA TECHNOLOGY

Akhmed-Zaki D.Zh., Daribayev B.S., Imankulov T.S., Turar O.N.

Abstract This paper describes the organization of high-performance computing by using GPUs (graphics processing unit) of different devices. In this study, the following tasks are examined: the Dirichlet problem for the Poisson equation, the problem of immiscible oil displacement (Buckley-Leverett's model) and mathematical model of oil displacement process by polymer-surfactant injection in a porous medium. Two algorithms have been tested: sequential and parallel (GPU). The calculations are performed on a mobile device Xiaomi MiPad with NVIDIA Tegra K1 processor. The results are analyzed and compared with computations on a PC (personal computer) with a NVIDIA GeForce GTX 550Ti graphics card. The research output has led the authors to come to the three main conclusions: mobile devices can be used as computers to solve practical problems as well as PCs; the global memory of mobile devices prove to be ineffective in complex calculations; there is demand for practical models that use shared memory in calculations. Therefore, authors find that it is reasonably possible to conduct calculations of complex mathematical models in real-time using mobile devices. All these allow to create a mobile prototype of the being-developed "smart" field platform for oil companies of Kazakhstan.

Key words: shared memory, mobile device, high-performance computing, parallel computing, Buckley-Leverett's model.

AMS Mathematics Subject Classification: 65D25, 65D18, 65M06, 68W10.

1 Introduction

In recent years, mobile technologies have become one of the most popular segments of global tech sector. Therefore, these technologies are being developed at a tremendous speed. Modern mobile processors are complete computational units which may be used as additional threads for computing clusters or as a platform for calculating separate tasks. In our days, there are millions of mobile devices whose resources are not used in full and that creates space for use of a number of selected computations. For example, BOINC (Berkeley Open Infrastructure for Network Computing) system provides scientists with computational resources consisting of thousands of PCs and mobile devices all over the world [1]. However, BOINC only uses the central processing unit of mobile devices.

The idea of creating a high-performance software on mobile devices have been originated in the framework of realization of the "i-fields" concept [2] to the oil fields of Kazakhstan (i.e. the goal was to develop a hydrodynamic simulator for real-time calculations on mobile platforms (Figure 1). Its purpose to work on modelling and

prediction of hydrodynamic processes using mobile devices, which allows operational decision-making despite user's location and time of day. This work is part of project aimed at design of distributed high-performance information system to analyze the development of oil and gas fields using modern mathematical models of enhanced oil recovery [3, 4, 5]. High-performance applications for mobile devices will be loaded as a service package from a web platform.

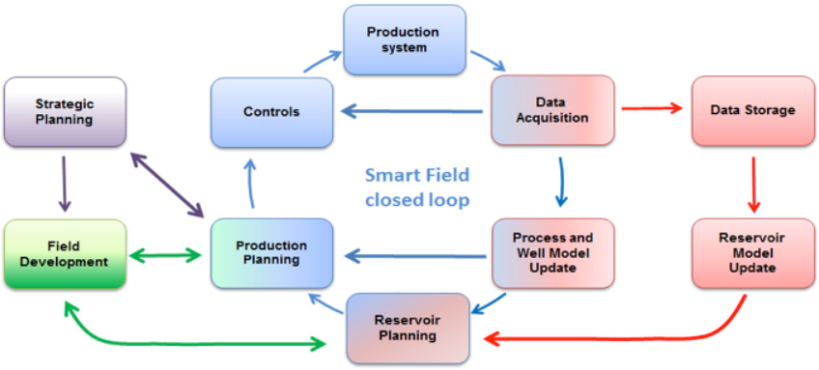

Figure 1: "Closed-loop intelligent field" scheme (Smart Fields Consortium, Stanford University, USA).

Section 2 describes three problems in order to test the parallel algorithm. The first one is solving the three-dimensional Poisson equation with a Dirichlet boundary condition [6]. This equation considered as the simplest case of elliptic equations that are frequently used for simulation of stationary physical processes. The second problem is Buckley-Leverett's model, which describes a simple immiscible oil displacement process in a porous medium. The last one is the solution of problem of oil displacement process by polymer and surfactant injection taking into account a salinity of oil reservoir and temperature effects. The last computer model described the complex real industrial problem of oil recovery. Section 3 describes a parallel algorithm on a GPU, which uses a shared memory for storage of a grid node values. Section 4 comprises of various comparative tests focusing on effectiveness of mentioned algorithms.

2 Problem Statement

The work is aimed at creation of an ubiquitous computing system for oil recovery computer model. For this purpose, a mobile application has been written, which implements high-performance parallel algorithms using GPU of the mobile device. These algorithms are used to parallelize the numerical solution of differential equations. 3 cases are considered for the models of varying complexity.

Case 1 Poisson equation is the simple example of an elliptic problem. It allows to analyze the behaviour of many stationary simulation elliptic models. The problem was numerically solved in three-dimensional domain $D = \{0 \leq x_\alpha \leq 1, \alpha = 1, 2, 3\}$.

Table 1: Computing time (sec.) of three-dimensional Poisson equation on different grids and algorithms (using CUDA technology with block size 8x8x8)

Tested algorithms	32x32x32	64x64x64	128x128x128	256x256x256
Parallel algorithm (PC)	0.03	0.25	2.14	20.59
Parallel algorithm (mobile)	0.23	0.96	14.75	62.09
Sequential algorithm (PC)	0.41	3.53	29.65	241.61

Poisson equation has the following form:

$$\Delta u = -f \tag{1}$$

Dirichlet boundary conditions are:

$$\begin{gathered} u(0, x_2, x_3) = u(1, x_2, x_3) = 0, \\ 0 \le x_2 \le 1; 0 \le x_3 \le 1, \\ u(x_1, 0, x_3) = u(x_1, 1, x_3) = 0, \\ 0 \le x_1 \le 1; 0 \le x_3 \le 1, \\ u(x_1, x_2, 0) = u(x_1, x_2, 1) = 0, \\ 0 \le x_1 \le 1; 0 \le x_2 \le 1. \end{gathered} \tag{2}$$

where Δ is the Laplace operator.

The Jacobi iteration method for numerical solutions of the Poisson equation is considered in the three-dimensional case:

$$\begin{gathered} u_{i,j,k}^{n+1} = \frac{1}{6}(u_{i-1,j,k}^{n} + u_{i+1,j,k}^{n} + u_{i,j-1,k}^{n} + \\ + u_{i,j+1,k}^{n} + u_{i,j,k-1}^{n} + u_{i,j,k+1}^{n} + h^2 * f_{i,j,k}) \end{gathered} \tag{3}$$

Case 2 Buckley-Leverett's model is chosen as an example of a real industrial problem. The model considers the following assumptions:

- fluids and porous medium are incompressible;
- the model does not take into account the capillary effects and the gravitational forces;
- the two-phase flow obeys Darcy's law;
- the process is isothermal;

Mass conservation equation for water and oil phases has the following form [7]:

$$m\frac{\partial S_o}{\partial t} + div(\vec{v}_o) = q_2 \tag{4}$$

$$m\frac{\partial S_w}{\partial t} + div(\vec{v}_w) = q_1 \tag{5}$$

$$S_w + S_o = 1 \tag{6}$$

where m is porosity, S_w, S_o are water and oil saturations, $\vec{v}_w, \vec{v}_o$ are velocities of the water and oil phases which are expressed by the Darcy's law:

$$\vec{v}_i = -K_0 \frac{f_i(s)}{\mu_i} \nabla P, i = w, o \tag{7}$$

$f_i(s), \mu_i$ is the relative permeability and viscosity of fluids, K_0 is the absolute permeability.

The pressure equation is obtained by adding (6) and (7):

$$div(\vec{v}_w) + div(\vec{v}_o) = q_1 + q_2 \tag{8}$$

The initial and boundary conditions are:

$$\begin{gathered} s|_{t=0} = s_0(x) \\ \frac{\partial s}{\partial n}|_{\partial\Omega} = 0; \frac{\partial P}{\partial n}|_{\partial\Omega} = \gamma_p.V_p; \end{gathered} \tag{9}$$

where $\partial\Omega$ is the bound of area.

The relative permeability curves are taken as follows:

$$f_w(S_w) = S_w^{3.5}; f_o(S_w) = (1 - S_w)^{3.5}; \tag{10}$$

Case 3 Mathematical model of the oil displacement process by polymer-surfactant flooding, which considers the influence of temperature effects and water salinity, consists of (4)-(10), polymer/ surfactant/salt transport equations and a heat transfer equation has the following form [8, 9, 10]:

$$\begin{gathered} m\frac{\partial}{\partial t}(c_p s_w) + \frac{\partial a_p}{\partial t} + div(\vec{v}_w c_p) = \\ = div(mD_{pw} s_w \bigtriangledown c_p) \end{gathered} \tag{11}$$

$$\begin{gathered} m\frac{\partial}{\partial t}(c_{sw} s_w + c_{so} s_o) + \frac{\partial a_{surf}}{\partial t} + \\ + div(\vec{v}_w c_{sw}) + div(\vec{v}_o c_{so}) = \\ = div(mD_{sw} s_w \bigtriangledown c_{sw} + mD_{so} s_o \bigtriangledown c_{so}) \end{gathered} \tag{12}$$

$$m\frac{\partial}{\partial t}(c_s s_w) + div(\vec{v}_w c_s) = 0, \tag{13}$$

where c_p, c_s are polymer and salt concentrations in aqueous phase, c_{sw}, c_{so} are surfactant concentration in aqueous and oleic phases, a_p, a_{surf} are polymer and surfactant adsorption functions, D_{pw}, D_{sw}, D_{so} are polymer and surfactant diffusion coefficients.

Heat transfer equation looks like the following:

$$\begin{gathered} \frac{\partial}{\partial t}(((1-m)C_r\rho_r + \\ + m(C_w s_w \rho_w + C_o s_o \rho_o))T) + \\ + div(\rho_w C_w \vec{v}_w T) + + div(\rho_o C_o \vec{v}_o T) = \\ = div((1-m)\lambda_0 + m(\lambda_1 s_w + \lambda_2 s_0) \bigtriangledown T) \end{gathered} \tag{14}$$

Table 2: Computing time (sec.) for the Case 3

Tested algorithms	16x16x16	32x32x32	64x64x64	128x128x128
Parallel algorithm (PC)	0.127	0.499	2.402	32.639
Parallel algorithm (mobile)	1.115	2.180	6.004	51.480

where C_w, C_o, C_r are specific heat of water, oil and rock, ρ_w, ρ_o, ρ_r are density of water, oil and rock, $\lambda_w, \lambda_o, \lambda_r$ are coefficients of thermal conductivity.

For sequential numerical calculation consistency of units and order of variables are important. Therefore, a system of equations (1) - (14) is converted to a dimensionless form. To solve these equations, an explicit scheme is used. First of all, fluid properties and physical parameters of reservoir are determined. Further calculations are conducted to find in the following order: distribution of pressure; distribution of saturation; distribution of salt, surfactant and polymer concentrations; distribution of temperature in the reservoir;

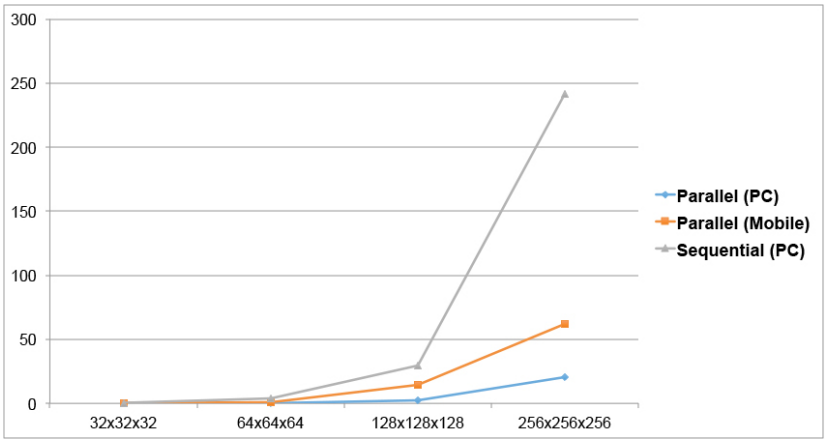

Figure 2: Comparative timeline calculations for three-dimensional Poisson equation.

3 A Parallel Algorithm using CUDA Technology on a Mobile Device

We assume that several blocks are located in one grid of a GPU. Each block has three directions (i.e. the individual blocks are arranged in a three-dimensional form). In each block, there are several threads which taken in a three-dimensional form. Further on, during the process of calculation, the data are copied from the global memory to the shared memory of the mobile device. In this case, the received data cannot be used repeatedly. Taking into account the fact that global memory is considered to be the slowest one, copying internal nodes four times will be ineffective. In order to proceed with the calculation in each subregion, it is necessary to use the data from the neighbouring subregions (i.e. one needs to copy the boundary data from a global memory [11, 12]).

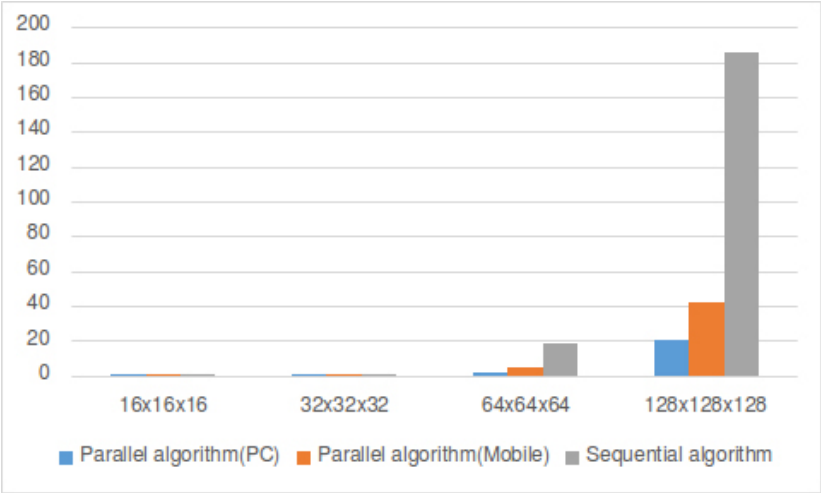

Figure 3: Comparative timeline calculations for pressure equation.

For solving problems mentioned above the parallel algorithm uses a kernel function. Within this function, an array is declared in a shared memory according to the size of the problem. Within a single block, each thread will work only with the shared memory. It is known that the shared memory has a very high throughput and there is no need to access the global memory each time which results in a considerable performance increase. In the mentioned parallel algorithm, in order to calculate the output array, the kernel function declares a temporary array in the shared memory and it sets its value to the value of the input array, and additionally sets the boundary values from the neighbouring blocks. Further on, while calculating the equation, it calls up the input array from shared memory instead of global memory. Due to the high throughput of the shared memory, the time used in calculation of the program speeds up. Upon the receipt of boundary values from the neighbouring blocks, it is important to choose the correct indices for the work with the correct data.

4 The Results and Comparison of Computational Experiments

Case 1 Let us consider results of parallel calculation on different devices to solve the three-dimensional Poisson equation. The chosen block size is equal to 8 in three dimensions. The calculation time of this algorithm on the PC is several times less than the one on the mobile device (Table 1). It can be explained by the fact that the equation has more dimensions, which affects performance during the receipt of border elements from the neighbour blocks. For the grid of the size 256x265x256, the calculation time on the PC is equal to 20.59 seconds, and 62.09 seconds for the mobile device. Depending on the size of the grid, parallel implementations on the mobile device are 200-250% slower than the ones on the PC. The parallel algorithm works 10-15 times faster than the sequential algorithm (Figure 2).

Case 2 Next, we are testing our parallel algorithm on different devices to solve the three-dimensional pressure and saturation equations of Buckley-Leverett's model. For

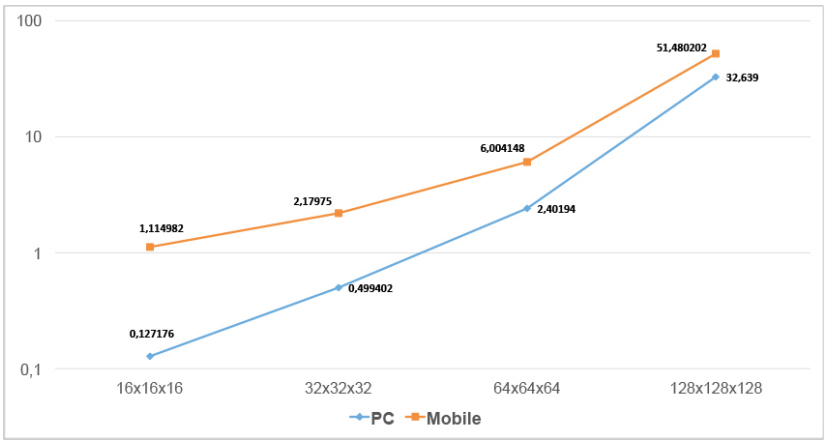

Figure 4: Computing times of mobile device and PC (polymer and surfactant flooding)

the grid size of 128x128x128, the calculation time of the parallel algorithm on the PC is equal to 21.07 seconds, and on the mobile device – 42.845 seconds.

Figure 3 presents the ratio of mobile and PC computing times. The results prove that the increase in the grid size leads to a time ratio decline.

Case 3 After testing the Buckley-Leverett's model, we are tested the model of oil displacement process by polymer and surfactant, taking into account the salinity and temperature of the reservoir. In Figure 4, we can see the ratio of calculating time on the PC and the mobile device. By increasing the grid size the time ratio decreases. Calculation times of the tested algorithm on different grids and devices are shown in Table 2.

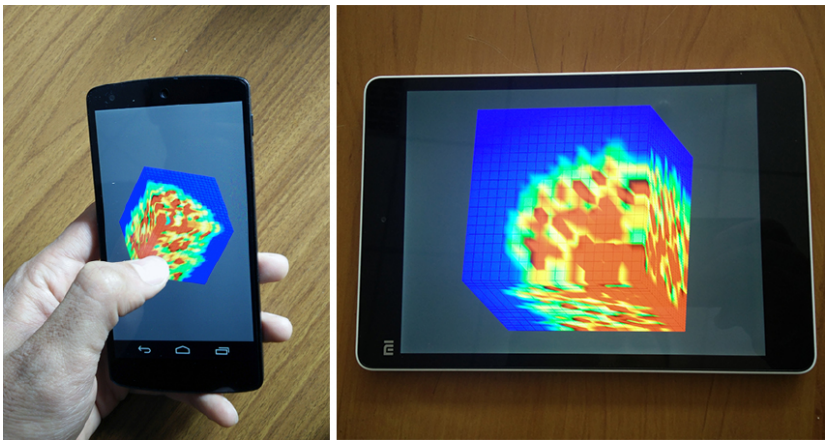

Figure 5: Demonstration of the mobile application results on the data of Kazakh oil company.

Figure 5 demonstrates the application for the mobile device on the base of the model of oil displacement process by polymer and surfactant, taking into account the salinity and temperature of the reservoir in porous medium.

The organization of mobile computing for solving practical problems that arise at various technological processes throughout the world has been developing recently. The existing high-performance mobile applications are widely developed in the game industry, as well as in the data preprocessing for image recognition, and in mobile cloud services. It is known that the power of modern mobile devices are constantly increasing; for example, the NVIDIA Tegra X1, which has a GPU with 256 cores on the NVIDIA Maxwell architecture has the ability to provide the theoretical performance at 1 Tflops [13]. The specified performance is comparable to the power of mini-supercomputers; therefore the complexity of tasks on mobile platforms will also increase in the future.

Nowadays, there are a number of companies who develop hydrodynamic simulators of oil and gas fields, such as, Schlumberger, Roxar, etc [14]. Their software is mainly oriented only for PCs and high-performance clusters as "thick client" version, which consequently reduces the degree of convenience in operation and requires additional specific knowledge for engineers who use it.

The proposed prototype of the mobile high-performance hydrodynamic simulator runs based on the known input data format (such as, Schlumberger, Roxar, etc.) and the results are easily integrated with other software packages for further use. The user is able to rapidly assess the computer model calculation outputs and make decisions on the oil fields in real-time. The latter is modified implementation of the complex IT systems component, which designed according to the "i-fields" paradigm in the modern oil and gas industry. Actually the Case 2 and Case 3 are used in a lot of major oil and gas fields around the world [15], which indicates possibility of using our mobile prototype in real oil fields.

5 Conclusion

The paper describes the results of using Tegra K1 based mobile device to develop parallel numerical solutions of chosen mathematical models.

The following conclusions can be noted:

First, mobile devices can be used as computers with significant computational capacities. Second, according to the presented tests, it becomes obvious that the global memory of the mobile device is even more ineffective than the one on a PC. As is seen in the figures, the time losses during repeating accesses to the global memory in some tests can result in a longer working period of the program than the sequential algorithm. Third, it turns out that there is a high demand for practical models in using algorithms for shared memory.

The results of research are illustrated a potential possibility to develop and use the prototype of high-performance reservoir simulator for mobile devices in the framework of "i-fields" concept for calculating of mathematical models of enhanced oil recovery in real-time. In the future, we will try to realize the idea of high-performance computing organization on networks of mobile platforms with a GPU using by wireless and plan to add in-app visualization module for drawing large volumes of data using the ray tracing algorithm with CUDA technology.

References

[1] *BOINC - Open-source software for volunteer computing and grid computing, http://boinc.berkeley.edu/.*

[2] *Smart Fields Consortium, https://smartfields.stanford.edu/.*

[3] T.S. Imankulov, D.Zh. Akhmed-Zaki, B.S. Daribayev, O.N. Turar *Organization of High-performance Computing on the Mobile Platforms to Calculate the Oil Recovery of Reservoir*, ECMOR 2016 - 15th European Conference on the Mathematics of Oil Recovery, Handout. Amsterdam, Netherlands. 2016.

[4] Akhmed-Zaki D., Lebedev D., Perepelkin. V. *Implementation of a three dimensional Three-Phase Fluid Flow ("Oil-Water-Gas") Numerical Model in LuNA Fragmented Programming System*, Journal of Supercomputing, S.I.: Parallel Computing Technologies. (2016), 1-7.

[5] D. Zh. Ahmed-Zaki, S. T. Mukhambetzhanov, T.S. Imankulov *Design of i-Fields System Component: Computer Model of Oil-Recovery by Polymer Flooding*, Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics. (2015), 510-517.

[6] A.A. Samarskij, *Teoria rasnostnyh shem*, 1983, 616.

[7] Kh. Aziz and A. Settari, *Petroleum reservoir simulation*, Applied Science Publishers, 1979.

[8] D. Akhmed-Zaki and N. Danayev and S. Mukhambetzhanov and T. Imankulov, *Analysis and evaluation of heat and mass transfer processes in porous media based on Darcy-Stafan's model*, ECMOR 2012 - 13th European Conference on the Mathematics of Oil Recovery, (2012), 122.

[9] D.Zh. Akhmed-Zaki, T.S. Imankulov, B. Matkerim, B.S. Daribayev, K.A. Aidarov, O.N. Turar, *Large-scale simulation of oil recover by surfactant-polymer flooding*, Eurasian Journal of Mathematical and Computer Applications, Vol. 4, Issue 1 (2016), 12-31.

[10] T. Imankulov and D. Lebedev and K. Aidarov and O. Turar, *Design of HPC system for analysis the gel-polymer flooding of oil fields*, Contemporary Engineering Sciences, 7 (2014), 1531-1545.

[11] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, 2010, 290.

[12] Shane Cook, *CUDA Programming. A Developer's Guide to Parallel Computing with GPUs*, 2012, 600.

[13] *NVIDIA TEGRA X1 The New Level of Mobile Performance, http://www.nvidia.com/object/tegra-x1-processor.html.*

[14] *Rock Flow Dynamics, http://rfdyn.com/ru/technology/.*

[15] L.W. Lake, *Enhanced oil recovery*, Englewoods Cliffs, New Jersey: Prentice Hall, Inc, 1989.

D.Zh.Akhmed-Zaki,
al-Farabi Kazakh National University,
al-Farabi ave., 71,
Email: darhan_a@mail.ru.

B.S.Daribayev,
al-Farabi Kazakh National University,
al-Farabi ave., 71,

Email: beimbet.daribaev@gmail.com.

T.S.Imankulov,
al-Farabi Kazakh National University,
al-Farabi ave., 71,
Email: imankulov_ts@mail.ru.

O.N. Turar,
al-Farabi Kazakh National University,
al-Farabi ave., 71,
Email: turar_olzhas@mail.ru.

Received 22.02.2017, Accepted 28.04.2017