Implementation of a three-phase fluid flow ("oil-water-gas") numerical model in the LuNA fragmented programming system

Akhmed-Zaki D.Zh.[1], Lebedev D.V.[1], Perepelkin V.A.[2]

[1] al-Farabi Kazakh National University, Almaty, Republic of Kazakhstan
Darhan.Ahmed-Zaki@kaznu.kz, danil.lebedev.0881@gmail.com
[2]Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia
perepelkin@ssd.sscc.ru

Abstract. The fragmented programming technology and the language implementing it are briefly introduced as well as LuNA fragmented programming system, on the example of two-dimensional boundary value problem solution, for liquid filtration "oil-water-gas" system. For parallel implementation of the boundary value problem, the parallel longitudinal-transverse sweep algorithm was applied. Using this method, the fragmented program in the LuNA system has also been implemented. The calculations are made for different number of points in the spatial variables. To compare the quality of implementation the applied numerical algorithm has been implemented in several variations: the sequential program, the parallel program using MPI and the fragmented parallel program in LuNA language using LuNA programming system.

Keywords: Fragmented programming, LuNA, numerical solution, MPI, Parallel program, sweep method.

1 Introduction

Implementation of large-scale numerical models on supercomputers is often challenging, especially for scientists, who are not experienced in system parallel programming. Consequently, of great importance are systems and tools of parallel programming. Such tools are aimed at reducing the complexity of parallel programming through its automation. In appropriate circumstances they reduce the complexity of parallel programs development, lower system parallel programming skill requirements, improve quality of resulting parallel programs, and so on.

Parallel programming automation is a subject for numerous research efforts, and its importance tends to increase. Worth mentioning are the following systems of parallel programming, which closely relate to scientific numerical modeling: PaRSEC [8], libgeodecomp [9], Charm++ [10], KeLP [11].

The LuNA programming system [6-7,12] is also a system of parallel programming, aimed at elimination of parallel programming from the process of parallel implementation of large-scale numerical models on supercomputers. In this paper we discuss the parallel implementation of two-dimensional boundary value problem solution, for liquid filtration "oil-water-gas" system in LuNA programming system.

The fragmented programming technology and LuNA programming system were chosen, because they aim at reduction of complexity of implementation of large scale numerical models.

The paper is organized as follows. The next two sections describe the application problem and its mathematical formulation, section 4 studies the parallel algorithm of the problem's solution. Section 5 contains a brief description of LuNA system, and section 6 represents the results of the performance tests.

2 The problem of filtration

This paper considers the two-dimensional problem of three-phase fluid filtration in the "oil-water-gas" system. The problem is a simulation of oil recovery process for secondary methods. Practical significance of the calculations for this class of problems is quite large. This is due to the fact that a very large part of oil production is associated with the use of secondary recovery techniques, such as displacement of oil by water or solvents, thermal impact on the field, etc. Considered model describes the secondary method of oil by water displacement. It has a number of specific features that make it difficult, and in some cases impossible to use standard numerical methods, proven to be efficient for other classes of problems. General formulation of the problem can be reduced to the following form: there is an oil reservoir, in which the water is pumped under pressure through the injection well, and it is necessary to calculate how much oil will be obtained from the production well. More information can be found in [1-4].

3 Definition of the problem

Let us consider a two-dimensional boundary value problem for liquid filtration of "oil-water-gas" system. The problem is described by the following equations [1,2] in dimensionless variables.

$$\begin{cases}
\frac{\partial}{\partial x}\left[K_w\left(\frac{\partial P_w}{\partial x}-\gamma_w\frac{L}{P_H}\frac{\partial z}{\partial x}\right)\right]+\frac{\partial}{\partial y}\left[K_w\left(\frac{\partial P_w}{\partial y}-\gamma_w\frac{L}{P_H}\frac{\partial z}{\partial y}\right)\right]=\frac{\mu_w}{\mu_o}\frac{\partial S_w}{\partial \tau}+\frac{\mu_w L^2}{K\rho_w P_H}q_w \\
\frac{\partial}{\partial x}\left[K_o\left(1+C_f P_H(P_o-1)\right)\left(\frac{\partial P_o}{\partial x}-\gamma_o\frac{L}{P_H}\frac{\partial z}{\partial x}\right)\right]+\frac{\partial}{\partial y}\left[K_o\left(1+C_f P_H(P_o-1)\right)\left(\frac{\partial P_o}{\partial y}-\gamma_o\frac{L}{P_H}\frac{\partial z}{\partial y}\right)\right]= \\
=\frac{\partial}{\partial \tau}\left[S_o\left(1+C_f P_H(P_o-1)\right)\right]+\frac{\mu_o L^2}{K\rho_H P_H}q_o \\
\frac{\partial}{\partial x}\left[R_s K_o\left(1+C_f P_H(P_o-1)\right)\left(\frac{\partial P_g}{\partial x}-\frac{\partial P_{cog}}{\partial x}-\gamma_o\frac{L}{P_H}\frac{\partial z}{\partial x}\right)\right]+ \\
+\frac{\partial}{\partial y}\left[R_s K_o\left(1+C_f P_H(P_o-1)\right)\left(\frac{\partial P_g}{\partial y}-\frac{\partial P_{cog}}{\partial y}-\gamma_o\frac{L}{P_H}\frac{\partial z}{\partial y}\right)\right]+ \\
\frac{\mu_o P_H}{\mu_g\rho_H RTZ}\frac{\partial}{\partial x}\left[K_g P_g\left(\frac{\partial P_g}{\partial x}-\gamma_g\frac{L}{P_H}\frac{\partial z}{\partial x}\right)\right]+\frac{\mu_o P_H}{\mu_g\rho_H RTZ}\frac{\partial}{\partial y}\left[K_g P_g\left(\frac{\partial P_g}{\partial y}-\gamma_g\frac{L}{P_H}\frac{\partial z}{\partial y}\right)\right]= \\
=\left(1-C_f P_H\right)\frac{\partial}{\partial \tau}(R_s S_o)+C_f P_H\frac{\partial}{\partial \tau}(S_o)+\frac{P_H}{\rho_H RTZ}\frac{\partial}{\partial \tau}\left(P_g S_g\right)+\frac{\mu_o L^2}{K\rho_H P_H}\left(R_s q_o+q_g\right) \\
P_o-P_w=P_{cow} \\
P_g-P_o=P_{cog} \\
S_w+S_o+S_g=1
\end{cases} \qquad (1)$$

Initial and boundary conditions

$$\begin{aligned}
&P_0(x,y,0)=P_o^H(x,y), P_w(x,y,0)=P_w^H(x,y), P_g(x,y,0)=P_g^H(x,y) \\
&S_0(x,y,0)=S_o^H(x,y), S_w(x,y,0)=S_w^H(x,y), S_g(x,y,0)=S_g^H(x,y)
\end{aligned} \qquad (2)$$

$$\left.\frac{\partial P_o}{\partial n}\right|_\Gamma=0, \left.\frac{\partial P_w}{\partial n}\right|_\Gamma=0, \left.\frac{\partial P_g}{\partial n}\right|_\Gamma=0 \qquad (3)$$

Given equations, describe the change in pressure (P_l) and saturation (S_l) for each phase, in time and space coordinates. Influence of wells is accounted by the corresponding coefficients (q_l). The complexity of the solution is caused by presence of coefficients – functions of saturation inside of differential operator.

4 Algorithm of the Solution

To solve the system (1) an iterative method with implicit pressure and explicit saturation is used. In order to do this, first three equations of the system are summarized and using fourth and fifth ratios of the system (1). Obtained equation is solved relative to the gas pressure. Resulting equation is reduced to the implicit difference scheme, then solved by the longitudinal-transverse sweep method [5] at each iteration layer using saturation values from the previous iteration layer. The idea of the method is to calculate values for next time step through intermediary time step, where at intermediary time step the function value is calculated as derivative relative to one spatial variable, and in the second spatial variable takes the value from previous time step (longitudinal direction). In the transverse direction, the value of the function at next time step is calculated as derivative by second spatial variable and first spatial variable takes the value calculated at intermediary time step. Calculated values of gas pressure are used to find other pressure values. Then, we find saturation of oil and water, using the first and second equation of the system (1), and gas saturation value – using the last ratio of the system. Iterative process stops when the convergence condition is satisfied.

$$max\left|S_{g,i,j}^{r} - S_{g,i,j}^{r-1}\right| \leq \varepsilon_1 (4)$$

Convergence of the process is affected by number of wells, because gradients of pressure and saturation phases are rapidly changing around the wells and the rest of the area is changing smoothly and therefore the convergence condition is reached faster.
We suggest the following parallel algorithm. We make decomposition of the spatial area into rectangles, as shown in the following figure

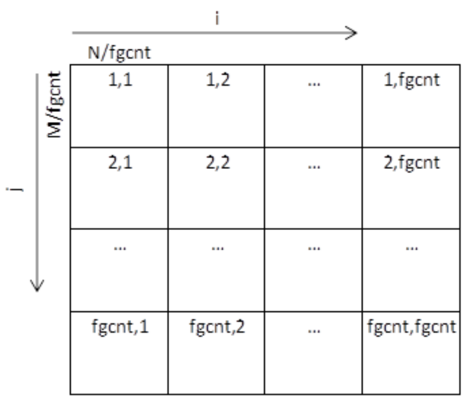

Fig. 1. Decomposition area.

Figure 1 shows that direction j has fragment size M/fgcnt, where fgcnt is the number of fragments. Direction i has fragment size N/fgcnt. Therefore, we have $fgcnt^2$ data fragments of size *N/fgcnt×M/fgcnt*. Then, the original algorithm can be represented as sequence of following steps:
Step 1. Generate variable matrices $P_o, P_w, P_g, S_w, S_o, S_g$ of size *Nx(M/fgcnt+1)* for first and last data fragment and Nx(M/fgcnt+2) for others. The values of these matrices are determined from the initial conditions (2).
Step 2. Solve the equation for the gas pressure obtained from addition of the system (1) at intermediate time step for $j = \overline{1, M/fgcnt}$, $i = \overline{1, N-1}$ and all $k = \overline{1, fgcnt}$
Step 3. Calculate values for pressure of oil and water, using ratios 4 and 5 of the system (1) for all $k = \overline{1, fgcnt}$.
Step 4. Calculate values for saturation of oil and water using equations 1 and 2 of the system (1), gas saturation can be found from ratio 6 of the system (1) for all $k = \overline{1, fgcnt}$.
Step 5. Check convergence condition (4), if it is satisfied for all $k = \overline{1, fgcnt}$ then go to step 6, otherwise override values of variables matrix $P_o, P_w, P_g, S_w, S_o, S_g$ with values calculated in steps 2-4.

Step 6. Generate variables matrix $P_o, P_w, P_g, S_w, S_o, S_g$ of size *(N/fgcnt+1)*x*M* for first and last data fragment and *(N/fgcnt+2)*x*M* for others.
Step 7. We define the values of these matrices according to the following scheme: fragments k send all l fragments submatrices of their variables$P_o, P_w, P_g, S_w, S_o, S_g$, of size (N/fgcnt+2)x(M/fgcnt+2) for $l \neq 1\ or\ l \neq fgcnt$ where $i = \overline{(l-1) * N/fgcnt - 1, l * N/fgcnt + 1}$ and size of (N/fgcnt+1)x(M/fgcnt+1) otherwise, where $i = \overline{(l-1) * N/fgcnt, l * N/fgcnt + 1}\ by\ l = 0$ and $i = \overline{(l-1) * N/fgcnt, l * N/fgcnt}\ by\ l = fgcnt$
Step 8. Solve the equation for gas pressure obtained from addition of the system (1) at intermediary time step for $i = \overline{1, N/fgcnt}$, $j = \overline{1, M-1}$ and all $l = \overline{1, fgcnt}$.
Step 9. Similar to steps 3 and 4, find the pressure of oil and water and saturation of oil, water and gas for all $l = \overline{1, fgcnt}$.
Step 10. Check convergence condition (4), if it is satisfied for all $l = \overline{1, fgcnt}$ then go to step 11, otherwise override values of variable matrices $P_o, P_w, P_g, S_w, S_o, S_g$ to values calculated in step 9.
Step 11. Fill variable matrices $P_o, P_w, P_g, S_w, S_o, S_g$, for longitudinal direction by the following scheme: l fragment sends all k fragments submatrix of their variables $P_o, P_w, P_g, S_w, S_o, S_g$ of size (*N/fgcnt+2*)x(*M/fgcnt+2*) for $k \neq 1\ or\ k \neq fgcnt$ where $i = \overline{(k-1) * N/fgcnt - 1, k * N/fgcnt + 1}$ and of size (*N/fgcnt+1*)x(*M/fgcnt+1*), otherwise, where $i = \overline{(k-1) * N/fgcnt, k * N/fgcnt + 1}\ by\ k = 0$ and $i = \overline{(k-1) * N/fgcnt, k * N/fgcnt}\ by\ k = fgcnt$ and go to step 2.
The process stops when it reaches specified time.
A feature of this parallel algorithm is the absence of data exchange within the computations by directions. Only when the iterative process converged in one direction, the calculated values will be transferred to other processes. This reduces amount of communications, while checking convergence conditions of the iterative method, but leads to sending the entire array, obtained by each process as a result of calculation, to all other processes, in order to start computation in other direction.
To compare implementation quality the applied numerical algorithm has been implemented in several forms: sequential Java program, parallel C++ program using the MPI standard, parallel fragmented program in LuNA language [6, 7] using LuNA programming system.

5 LuNA language and system of fragmented programming

LuNA (Language for Numerical Algorithms) – is a language and a system of parallel programming, aimed at automation of parallel programming of large-scale numerical models on supercomputers. LuNA language and system are being developed in the Supercomputer software department of the Institute of computational mathematics and mathematical geophysics of the Siberian branch of Russian academy of sciences.

The theoretical basis of LuNA is the theory of structured synthesis of parallel programs on computational models [13]. The main approach of LuNA can be described as follows. A user reformulates an application algorithm into the explicitly parallel form, called fragmented algorithm (FA). The FA is automatically transformed by

LuNA compiler into a parallel program, executable by LuNA run-time system on a multicomputer. LuNA compiler and run-time system take care of such problems as performing communications, data access synchronization, memory management, dynamic load balancing, and so on.

A major part of the approach is the possibility to improve the quality of FA execution by specifying "recommendations" – partial decisions on how to distribute FA among computing nodes, on what kind of workload scheduling to choose, and so on. The recommendations allow tuning the FA execution "by hand", achieving better performance without the user needing to drive into complex system parallel programming.

Main advantages, offered by LuNA, are reduction of programmer qualification requirements, reduction of parallel program development laboriousness, automatic provision of such properties of parallel program as dynamic load balancing, performing communications in parallel with computations, and so on. The programmer does not do parallel programming as such, his role is limited to algorithm decomposition, sequential programming and declaration of recommendations in a domain specific language (DSL). So, the parallel programming as such is eliminated form the process of implementation of numerical models for multicomputers.

Currently, LuNA system is implemented as a prototype. An FA, described in LuNA language, is interpreted by the run-time system, which invokes user sequential procedures, encapsulated in a traditional dynamic load library, according to the FAS. In such way, implementation of coarse-grained parallel algorithm mainly consists of native code execution and minor system overhead. On the contrary, fine-grained FA is likely to have poor performance. The LuNA run-time system is designed to be scalable to supercomuters of any size, therefore it only employs scalable distributed system algorithms with localized communications.

6 Performance Tests

Experiments were conducted for different number of mesh points in the spatial coordinates and different number of processors for parallel implementations. Parallel version and LuNA version were run on a cluster of Siberian Supercomputer Center. Time step and number of wells were not changed. The purpose of the test was to determine the efficiency of various implementations of the numerical algorithm. Test results are shown in Figure 2.

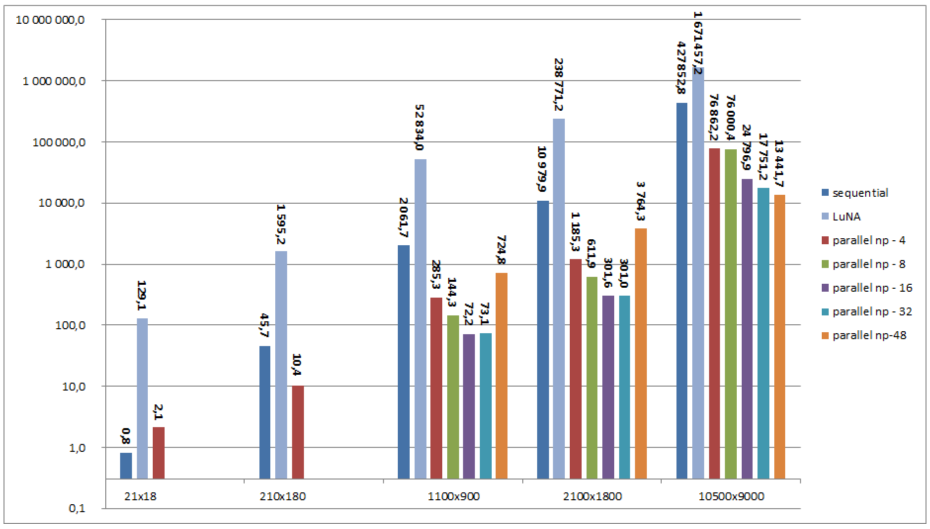

Fig. 2. The computation time (sec.) depending on the number of mesh points.

Figure 2 shows that at a small number of mesh points the minimum execution time belongs to the sequential implementation of the algorithm.

With increase of the number of points, parallel implementation is at first catching up and then overtakes the sequential implementation. This is due to the fact that, with the growth of computation load benefit from parallelism begins to outweigh the overheads arising from communications. Execution time of the LuNA program exceeds the other execution times at about half order. The reason for that is the relatively small size of the problem, which results in the fact that system overheads to organize execution of the fragmented program exceeds the "useful" computations of the algorithm. Nevertheless, despite lower quality of the implementation the LuNA program is easier to develop, because the LuNA system releases the programmer from the burden of dealing with communications, resource allocation, memory management, and number of other problems. Also, from figure 2 it can be seen, that with increase of the number of points and processors the execution time reduces, but only to a certain limit. With increase of the number of processors, more time is wasted on data transfer, and it affects the execution time of the program with small amount of computational load.

We can conclude that benefits of parallel implementations of the algorithm tend to increase with the problem size increase.

This work is supported by grant funding of scientific and technical programs and projects by the Committee of Science, Ministry of Education and Science RK, grant No. 528/GF2; and Russian Foundation for Basic Research (grants No 14-07-00381-a and 14-01-31328-mol_a).

References

1. Aziz K., Settari A. Petroleum reservoir simulation: Applied. Science Publishers Ltd., London, U.K., 1979, 407c.

2. Henry B Crichlow. Modern Reservoir Engineering – a simualation Approach. 1977, 303 p.
3. Konovalov A.N. The problem of filtration multiphase incompressible fluid. Novosibirsk: science 1988, 166p.
4. D. Zh. Akhmed-Zaki, N. T. Danaev, S. T. Mukhambetzhanov, T. Imankulov, *Analysis and Evaluation of Heat and Mass Transfer Processes in Porous Media Based on Darcy - Stefan's Model,* ECMOR XIII, 2012. http://dx.doi.org/10.3997/2214-4609.20143274
5. The method of fractional steps for solving multidimensional problems of mathematical physics. N. N. Janenko, 1967, 197 p.
6. Victor Malyshkin and Vladislav Perepelkin. Optimization methods of parallel execution of numerical programs in the LuNA fragmented programming system // The Journal of Supercomputing, Springer, Volume 61, Number 1 (2012), pp. 235-248. DOI: 10.1007/s11227-011-0649-6.
7. Kireev, S. The LuNA Library of Parallel Numerical Fragmented Subroutines / S. Kireev, V. Malyshkin, H.Fujita // Lecture Notes in Computer Science. – 2011. – Vol. 6873. – P. 290 –301.
8. http://icl.cs.utk.edu/parsec/index.html - accessed 15 jan 2015
9. Andreas Schäfer, Dietmar Fey. LibGeoDecomp: A Grid-Enabled Library for Geometric Decomposition Codes // Proceedings of the 15th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface // Springer-Verlag Berlin, Heidelberg, 2008. pp 285-294. DOI: 10.1007/978-3-540-87475-1_39
10. Laxmikant V. Kale, Sanjeev Krishnan. CHARM++: a portable concurrent object oriented system based on C++ // OOPSLA '93 Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications // ACM New York, NY, USA ©1993. pp 91-108. DOI: 10.1145/165854.165874
11. Scott Kohn, John Weare, M. Elizabeth Ong, and Scott B. Baden, Software Abstractions and Computational Issues in Parallel Structured Adaptive Mesh Methods for Electronic Structure Calculations // *IMA Volumes in Mathematics and its Applications,* Volume 117, *Structured Adaptive Mesh Refinement (SAMR) Grid Methods,* S. B. Baden, N. Chrisochoides, M. Norman, and D. Gannon, Eds., Springer-Verlag, 1999, pp. 75-95
12. Malyshkin V.E., Perepelkin V.A. *LuNA Fragmented Programming System, Main Functions and Peculiarities of Run-Time Subsystem* - In: Proceedings of the 11th Conference on Parallel Computing Technologis, LNCS 6873 - pp. 53-61, Springer, 2011
13. V.A.Valkovsky, V.E. Malyshkin. Synthesis if parallel programs and system on the basis if computational models. Nauka, Novosibirsk, 1988, 128 pp. In Russian: V.A.Valkovsky, V.E. Malyshkin. Sintez parallelnykh program i system na vychislitelnykh modelyah. Nauka, Novosibirsk, 1988, 128 str.