Совместный выпуск

Октябрь 2020
№3. Часть 1

ВЕСТНИК
Национальной инженерной академии Республики Казахстан

ВЫЧИСЛИТЕЛЬНЫЕ ТЕХНОЛОГИИ
Федеральный исследовательский центр информационных и вычислительных технологий

Содержание

High-performance and Intelligent Computational Models for Oil Production Problems*

D. Zh. Akhmed-Zaki[1], D. Lebedev[1], T. S. Imankulov[2], Y. Kenzhebek[2], and N. Kassymbek[2]

[1] Astana IT University, Nur-Sultan, Kazakhstan
[2] Al-Farabi Kazakh National University, Almaty, Kazakhstan
kenzhebekyerzhan@gmail.com

Abstract In recent years, modern information technologies are actively used in various sectors of industry. The oil industry is no exception, since high-performance computing technologies, artificial intelligence algorithms, methods of collecting, processing and storing information are actively used to solve the problems of enhanced oil recovery.
This article discusses the application of modern approaches to solving oil production problems: the fragmented approach technology, the parallel approach technology and machine learning algorithms. The problem of multiphase multicomponent fluid flow in porous medium is solved by the Newton-ILU (0)-GMRES method. To automate the implementation of numerical models on multicomputers, fragmented programming technology and a parallel algorithm on MPI are used. The developed algorithms were tested and analyzed on the MVS-10P supercomputer. Also, the problem of oil displacement is solved using machine learning algorithms. The algorithms are classical supervised learning (linear and polynomial regression) and neural networks.

Keywords: Enhanced oil recovery, MPI, Machine learning, LUNA, Regression methods, Artificial neural network, GMRES, Preconditioner.

Introduction

The development of modern high-performance computing allows humanity to solve important scientific and applied problems in various fields. The accuracy of solving such problems requires the use of a large mesh, which in turn requires large computational resources. Currently, supercomputers are used to solve problems such as weather modeling and climate prediction, acoustic problems, hydrodynamics, economic processes and etc.Actual and at the same time quite a difficult task of hydrodynamic modeling is the simulation of multicomponent multiphase flow of liquid (oil and gas) in porous media (in oil reservoirs). In solving such problems, various methods and schemes [1,2,3] can be used, some of them are iterative methods for solving linear systems [4,5].

Various technologies are used to speed up the execution time of calculations for solving scientific problems, such as MPI [6], OpenMP [7], CUDA [8,9], OpenCL [10], and fragmented programming [11].

* Supported by National Engineering Academy

In this paper, we solve a system of equations using the Newton-Raphson method, at each iteration of which the system of algebraic equations is solved using the generalized minimum residual method (GMRES) with ILU (0) preconditioner. A program for solving this problem using a method called Newton-ILU(0)-GMRES is implemented parallel using MPI and using fragmented programming (FP) technology, which is aimed at automating the construction of parallel programs for supercomputers.

There is a lot of research related to increasing oil production using machine learning methods. In this work [12], the authors found out that the use of machine learning (ML) algorithms may turn out to be more productive in comparison with traditional calculations on a regular grid [13,14].

The work [15] considers machine learning algorithms for estimating the oil production coefficient using a combination of engineering and stratigraphic parameters. For a data set consisting of 30 parameters, linear regression models and the support vector machine (SVM) method were applied. As a result, the data obtained were very close to the results of cross-validation. Thus, the authors of this work suggest that the methods considered by them can be used to predict future production. The authors of [16,17] considered the use of artificial neural network (ANN) for predicting oil production.

The study [18] examined various machine learning methods for predicting downhole pressure, oil production, and forecasting water cut in production tasks. The authors of [19] also examined the use of machine learning methods for interpreting pressure, flow rate, and temperature data from permanent downhole sensors. In this paper, three machine learning methods were applied, such as linear regression (LR), core method, and ridge core regression.

Numerical Simulation of Multicomponent Multiphase Flow in Porous Media using MPI technology

The one-dimensional problem of compositional fluid flow in a porous medium is considered. Compositional flow involves multiple components and three phases, and there is a mass transfer between the vapor and liquid phases

A mathematical model of a multicomponent flow of a three-phase liquid in a porous medium is considered. The law of conservation of mass for each component is written down and a system of nonlinear equations is obtained. Distribution of chemical components in the hydrocarbon phase described by the K-value method [20]. The thermodynamic behaviour of fluids under reservoir conditions are described by the Peng-Robinson equations of state [21].

The system of nonlinear equations is linearized by the Newton-Raphson method. The resulting system of linear equations is reduced to the following form, as shown by [2]:

$$Ax = b, \tag{1}$$

In this work, system (1) was solved by the iterative method GMRES, which is a widely used Krylov subspace method [22].

To reduce the number of iterations in the computation, preconditioning was used as an explicit or implicit modification of the system of linear equations, making it possible to simplify the solution. We choose ILU(0) [23] as a preconditioner for the GMRES algorithm, which was found by decomposing matrix A by a fine-grained ILU factorization algorithm. As a result, for the numerical solution of the given problem for solving a system of linear equations, (1) uses the Newton-ILU0-GMRES algorithm as a fully implicit scheme.

To implement the parallel algorithm of the Newton-ILU0-GMRES, we need to determine the parts of the calculations that can be the basis of parallelization.

For instance, the ILU0-GMRES algorithm consists of the following steps:

1. Initialization. Choose x_0, compute $r = b - Ax_0$, solve $Pw = r$and compute $v_1 = \frac{w}{\|w\|_2}$
2. Arnoldi iteration. Iterate m times: j=1,2,…,m. The specific Arnoldi algorithm is as follows:

for i=1, ...,m
  Pw=Ax$_i$
 for k=1, ...,i
  $h_{k,i} = (w, v_k)$
  $w = w - h_{k,i}v_k$
 endk
 $h_{i+1,i} = \|w\|_2$
 $v_{j+1} = \frac{W}{h_{i+1,i}}$
end i

3. Compute the approximate solution $x^m = x^0$+$Q_{n,m}\ y_m$. Here, y_m minimizes $\|\beta\xi - H_{m,m}\|$; $Q_{n,k}$ is an nxk matrix, the column vector of the matrix is composed of $v_1, v_2, ..., v_k$ orthogonal vectors, and H_{mm} is the Hessenberg matrix.
4. Determination. Compute $r^m = b - x^m$, stop if the condition is met: otherwise let $x^0 = x^m$ and go back to 1) and recalculate. The convergence condition by which the method stops is given by an arbitrary number ε.

In this ILU0-GMRES algorithm, there are common operations such as matrix-matrix multiplication, matrix-vector multiplication, vector-vector multiplication and vector norm calculation in steps of a), b) and c). These operations could be implemented in parallel with data decomposition. In a) and b) steps of the GMRES algorithm, P is used as the ILU(0) preconditioner.

The testing of the program was carried out on matrices corresponding to the cases when the number of grid point schemes varied from 500 to 8000. The size of the matrix being solved is 6 times larger than the number of grid points. The size of the matrix, which corresponds to the case when the number of points in the difference grid is equal to 8000, is 48000 x 48000, since for each point 6 equations are written.

Test runs for the considered problem (1) were made on the MVS-10P supercomputer of the Interdepartmental Supercomputer Center of the Russian Academy of Sciences, which includes nodes with two Xeon X5450 processors and 8 GB of RAM for each node. The test results are shown in Table 1.

Table 1. The runtimes of the parallel program in MPI

Number of processors	Size of Matrix A				
	3000x3000	6000x6000	12000x12000	24000x24000	48000x48000
1	3.74	48.38	334.24	1 688.74	20 219.34
8	1.19	7.57	53.82	394.74	2 942.02
64	2.59	12.03	64.16	389.68	2 565.59
128	2.95	12.66	65.25	395.58	2 569.01
256	4.39	17.29	82.75	450.28	2 798.32
512	11.53	42.56	202.01	866.98	4 185.07

As we can see from the test results, the parallel program running on 8 processors achieved the shortest runtime. As the size of the task increases, the parallel application achieves the shortest calculation time running on 64 processors, but it is not much different from the runtime of a parallel program running on 8, 128 and 256 processors. The reason is because with an increase in the number of processors, the computation time on each processor decreases, and the time spent on communications increases.

It can be seen that the runtimes on 64 and 128 processors are about the same. This finding is explained by the fact that on 64 processors, the communication time is less than the computation time, and on 128 processors, the computation time is less than the communication time, but in total, the time consumed is almost the same. For matrix size 3000x3000 on 256 and 512 processors, the parallel program is slower than the sequential program. With an increase in the size of the matrix, for instance, for matrices 6000x6000 - 48000x48000, the runtime of the parallel program decreases compared to the sequential program. This regularity makes it possible to predict that with an increase in the size of the matrix, in large numbers of processors, acceleration also increases. It can be concluded that it makes sense to run the parallel algorithm of the method on a large number of processors but only with large of matrix that cannot fit in the memory of a single node.

Numerical Simulation of Multicomponent Multiphase Flow in Porous Media on LuNA Fragmented Programming System

To solve the problem presented in the previous chapter, a fragmented version of the ILU(0)-GMRES algorithm for the LuNA system was developed. A fragmented algorithm frees the programmer from parts of parallel program execution, such as data transfer and messaging, and allows him to perform these actions automatically. In the system of fragmented programming LuNA to automate the creation of parallel programs uses the so-called fragmented algorithm (FA), which consists of light processes, computational fragments (CF). The data is represented as an immutable block called data fragments (DF). Accordingly, new data, called output DFs, are computed from input DFs using CF.

In the fragmented version of the Arnoldi orthogonalization algorithm, the vectors w and v_k are divided by n the number of DFs. The resulting matrices $H_{m,m}$ and $Q_{n,k}$ are stored as separate DFs. In each new Arnoldi iteration, the defined CFs calculate the new

output DFs, using the above listed DFs as input parameters. In this case, the input DF, which will no longer be used, can be removed from memory.

You can see that calculating of the residual *r* and finding the approximate solution x^m of the ILU(0)-GMRES algorithm, which was described before, are simple matrix-vector operations, the fragmented algorithms of which can be easily implemented in the LuNA system.

Table 2. The runtimes of the fragmented program in LuNA

Number of processors	Size of Matrix A	
	3000x3000	6000x6000
8	1338,81	8412,66

Table 2 shows the running time of a fragmented program. As you can see from the table, the computation time turns out to be very long, this is due to the fact that the algorithm has a lot of connections between the nodes, it takes a lot of time to organize them, moreover, in the LuNA system, the MPICH library is used to organize parallel computations using the TCP standard / IP. Therefore, a lot of time is spent on data exchange between nodes, in contrast to the MPI program, in which the data exchange was implemented using the faster InfiniBand standard.

Using Machine Learning Methods for Oil Recovery Prediction

In this work, the obtained synthetic data from a mathematical model were divided into a training and test sample. As input parameters, various combinations of parameters of the oil production problem (porosity, viscosity of the oil phase and absolute rock permeability) were taken. And as the output parameter, the value of the oil recovery coefficient was chosen. The data was divided into a training and test set. For training, 8069 sets (80%) of the total data were used, and for the test the remaining 2017 pairs (20%). The total number of sample pairs is 10,086 models. Each sample pair consists of 40 oil recovery factor values. As a result, we have many test pairs, however, in this paper, the results will be shown only for some. Fig. 1 shows the result of one test sample pair for different regression methods. Fig. 2 shows the result of one test sample pair for ANN. The Table 3 shows the average R^2 score for 20% of test sets.

Table 3. Evaluation of R^2 for all pairs of training and test set.

Machine learning algorithms	Test sets (20%) R^2
LR	0.91
Polynomial Regression (PR) degree = 2	0.96
PR degree = 3	0.79
PR degree = 3 with L1	0.92
ANN	0.97

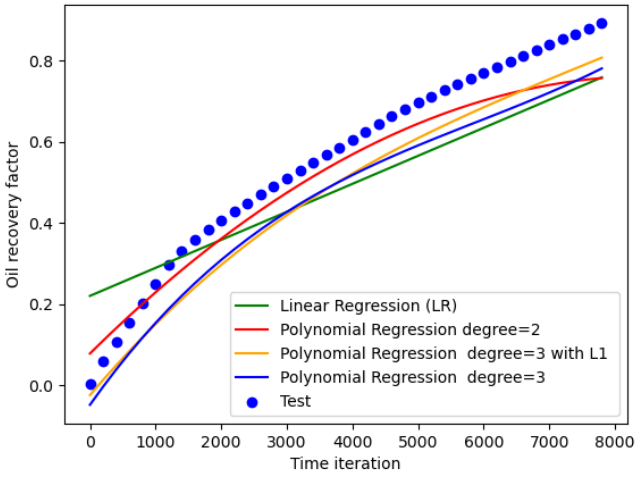

Figure 1. Oil recovery factor on different regression models.

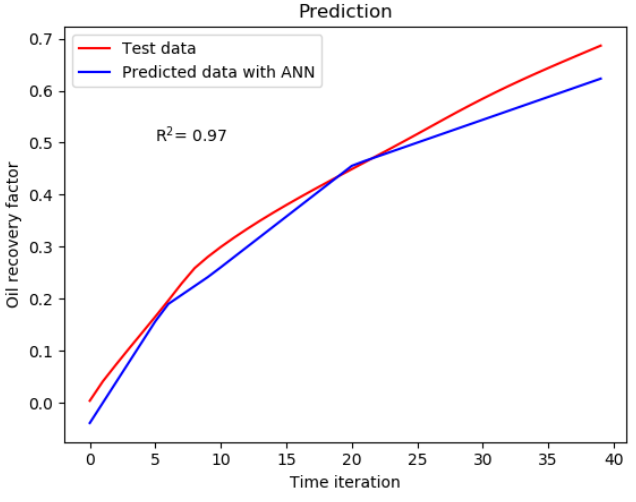

Figure 2. Oil recovery prediction using an artificial neural network.

Conclusion

This research examines the parallel implementation of the Newton-ILU0-GMRES algorithm for a multicomponent three-phase flow problem. Tests were carried out with different node sizes and numbers of processes on supercomputers. The study showed that on a large number of processes, the MPI program runs better in large-scale tasks, in addition the MPI program has higher efficiency and scalability than the LuNA program. In the future we plan to develop a fragmented program to direct control for the considered parallel algorithm, which will increase the effectiveness of the fragmented program.

This article was devoted to the application of machine learning methods for predicting oil production. Different degrees of polynomial regression were tested, and it was also revealed that for our synthetic data, the quadratic polynomial model is quite well trained and predicts the value of the oil recovery coefficient. An artificial neural network with one hidden layer with optimally selected hyperparameters was built. For the constructed neural network, the determination coefficient R^2 was 0.97, which is slightly better than the quadratic model of polynomial regression.Thus, it is assumed that the considered machine learning methods in this article may be useful for predicting oil production.

Acknowledgments

This research was funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. AP05130366).

References

1. Borisov V. E., Kritskiy B. V., Marchenko N. A., Mitrushkin D. A., & Savenkov E. B. (2013). Non-isothermal compositional flow model with chemical reactions and active solid phase for reservoir simulation. KIAM Preprint, 091.
2. Chen, Z. (2006). Reservoir Simulation: mathematical techniques in oil recovery. Society for Industrial and Applied Mathematics.
3. Akhmed-Zaki, D.Zh., Imankulov, T.S., Matkerim, B., Daribayev, B.S., Aidarov, K.A. and Turar, O.N. (2016) Large-scale simulation of oil recovery by surfactant-polymer flooding. Eurasian Journal of mathematical and computer applications, 4(1), 12-31.
4. Mittal, R.C., & Al-Kurdi, A.H. (2002). LU-decomposition and numerical structure for solving large sparse nonsymmetric linear systems. Computers & Mathematics with Applications, 185(2), 391-403. https://doi.org/10.1016/j.cam.2005.03.018
5. Lacroix, S., Vassilevski, YU., Wheeler, J., & Wheeler, M. (2003). Iterative Solution Methods for Modeling Multiphase Flow in Porous Media Fully Implicitly. SIAM Journal on Scientific Computing, 25(3), 905-926. https://doi.org/10.1137/S106482750240443X
6. Pannala, S., DAzevedo, E., Syamlal, M., & O'Brien, T. (2003). Hybrid (OpenMP and MPI) Parallelization of MFIX: A Multiphase CFD Code for Modeling Fluidized Beds. SAC '03: Proceedings of the 2003 ACM symposium on Applied computing, (pp. 199-206). Association for Computing Machinery. https://doi.org/10.1145/952532.952574

7. Iryanto, I., & Gunawan, P. H. (2017). An OpenMP parallel godunov scheme for 1D two phase oil displacement problem. 5th International Conference on Information and Communication Technology (ICoICT). https://doi.org/10.1109/ICoICT.2017.8074664
8. Zaza, A., Awotunde, A. A., Fairag, F. A., & Al-Mouhamed, M. A. (2016). A CUDA based parallel multi-phase oil reservoir simulator. Computer Physics Communications, 206, 2-16. https://doi.org/10.1016/j.cpc.2016.04.010
9. Akhmed-Zaki, D.Zh., Daribayev, B.S., Imankulov, T.S., & Turar, O.N. (2017). High-performance computing of oil recovery problem on a mobile platform using CUDA technology. Eurasian Journal of mathematical and computer applications, 5(2), 4-13. https://doi.org/10.32523/2306-3172-2017-5-2-4-13
10. Khramchenkov, E., & Khramchenkov, M. (2018). Numerical Model of Two-Phase Flow in Dissolvable Porous Media and Simulation of Reservoir Acidizing. Natural Resources Research, 27(4), 531-537. https://doi.org/10.1007/s11053-018-9371-x
11. Malyshkin, V.E., & Perepelkin, V.A. (2011). LuNA Fragmented Programming System, Main Functions and Peculiarities of Run-Time Subsystem. Proceedings of the 11th International Conference on Parallel Computing Technologies/LNCS, 6873, 53-61. https://doi.org/10.1007/978-3-642-23178-0_5
12. Akhmed-Zaki, D., Danaev, N., Mukhambetzhanov, S., Imankulov, T. [2012]. Analysis and evaluation of heat and mass transfer processes in porous media based on Darcy-Stefan's model. ECMOR 2012 - 13th European Conference on the Mathematics of Oil Recovery. https://doi.org/10.3997/2214-4609.20143274
13. Aliyuda, K., Howell, J. [2019]. Machine Learning Algorithm for Estimating Oil Recovery Factor Using a Combination of Engineering and Stratigraphic Dependent Parameters. Interpretation. 7. 1-34. 10.1190/int-2018-0211.1.
14. Imankulov, T.S., Akhmed-Zaki, D. [2016]. Computer modelling of non-isothermal, multiphase and multicomponent flow by using combined EOR technologies. 15th European Conference on the Mathematics of Oil Recovery, ECMOR 2016. https://doi.org/10.3997/2214-4609.201601765
15. Jreou, G. [2012]. Application of neural network to optimize oil field production. Asian Transactions on Engineering. 3. 1-9.
16. Krasnov, F., Glavnov, N., Sitnikov, A. [2018]. A Machine Learning Approach to Enhanced Oil Recovery Prediction. 10.1007/978-3-319-73013-4_15.
17. Mirzaei-Paiaman, A., Salavati, S. [2012]. The Application of Artificial Neural Networks for the Prediction of Oil Production Flow Rate. Energy Sources. 34. 10.1080/15567036.2010.492386.
18. Ristanto, T., Horne, R. [2018]. Machine Learning Applied to Multiphase Production Problems.
19. Tian, Ch., Horne, R. [2019]. Applying Machine-Learning Techniques To Interpret Flow-Rate, Pressure, and Temperature Data From Permanent Downhole Gauges. SPE Reservoir Evaluation & Engineering. 22. 386-401. 10.2118/174034-PA.
20. Pederson, K.S., & Christensen, P.L. (2008). Phase Behavior of Petroleum Reservoir Fluids. CRC Press.
21. Peng, D.Y., & Robinson, D.B. (1976). A new two-constant equation of state. Industrial and Engineering Chemistry Fundamentals, 15(1), 59-64. https://doi.org/10.1021/i160057a011
22. Saad, Y. (2003). Iterative methods for sparse linear systems. (2nd edn). SIAM.
23. Chow, E., & Saad, Y. (1997). Ilus: An incomplete LU preconditioner in sparse skyline format. International Journal for Numerical Methods in Fluids, 25(7), 739-748.