УДК 004.272.2:517.951
https://doi.org/10.47533/2020.1606-146X.07

Б. С. [*] ***ДАРИБАЕВ*** [1], ***Т. С. ИМАНКУЛОВ*** [1], ***Д. Ж. АХМЕД-ЗАКИ*** [2]

[1]*Казахский национальный университет имени аль-Фараби*
[2]*Университет Международного Бизнеса*

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ НА CUDA ДЛЯ РЕШЕНИЯ ЗАДАЧ МНОГОФАЗНОЙ ФИЛЬТРАЦИИ МНОГОКОМПОНЕНТНОЙ ЖИДКОСТИ В ПОРИСТЫХ СРЕДАХ

Рассматривается реализация параллельного алгоритма продольно-поперечной прогонки с использованием технологии CUDA для решения задач многофазной фильтрации многокомпонентной жидкости в пористых средах. Для решения систем трехдиагональных уравнений продольно-поперечной прогонки были использованы методы циклической и параллельной циклической редукции. Результаты исследования показали, что реализация алгоритмов циклической и параллельной циклической редукции на современных графических процессорах намного эффективней, чем на центральных процессорах.

Ключевые слова: *CUDA, метод продольно-поперечной прогонки, циклическая редукция, параллельная циклическая редукция, многофазная фильтрация.*

Введение. Гидродинамическое моделирование процессов, которые протекают в нефтяных пластах, является одной из сложных задач механики жидкости. Связано это с тем, что процессы, протекающие в подземных резервуарах, могут быть очень сложны. Развитие современных графических процессоров (GPU) открыло возможность для вычисления такого рода задач. Учитывая этот факт, нами был рассмотрен параллельный алгоритм продольно-поперечной прогонки (Alternating Direction Implicit - ADI) [1] на GPU для решения задачи многофазной фильтрации многокомпонентной жидкости в пористых средах. ADI – это конечно-разностный численный метод для решения параболических, гиперболических и эллиптических уравнений и широко используется в научных и инженерных областях [2-5]. В методе ADI каждый численный шаг разбивается на несколько подэтапов, основанных на пространственном измерении задачи, и система линейного уравнения решается неявно в одном направлении при явной схеме в другом направлении. В метода ADI на каждом подшаге решаемые уравнения имеют трехдиагональную структуру и могут быть решены с помощью трехдиагонального матричного алгоритма (TDMA) [6].

Для решения систем трехдиагональных уравнений было разработано немало параллельных алгоритмов: циклическая редукция (CR) [7], параллельная циклическая

*Адрес для переписки. E-mail: beimbet.daribayev@gmail.com

редукция (PCR) [8]. Для распараллеливания метода ADI мы использовали существующие алгоритмы CR и PCR на GPU. Для реализации последовательного алгоритма использовали неявный метод прогонки [9].

Для ускорения такого рода параллельных алгоритмов отлично подходят платформы с неспециализированными вычислениями на GPU (GPGPU). Графический процессор достигает высокой производительности, выполняя более тысячи потоков одновременно, и каждый из них обрабатывает различные наборы данных. В разных областях исследований было много успешных реализации на GPU, таких как анализ медицинский изображений [10] и вычислительная гидродинамика [11, 12].

Математическая модель. Рассмотрим математическую модель движения многофазной многокомпонентной жидкости в пористой среде. В модели учитываются 3 фазы. Номер фазы будем обозначать буквой α. Индекс $\alpha = o$ соответствует нефтяной фазе: нефтяная, газовая и водная, $\alpha = g$ – газовой фазе, $\alpha = w$ – водяной фазе. Номер компонента в фазе α будем обозначать буквой j. Уравнение неразрывности для отдельного компонента с номером j, содержащегося в фазе α, в дифференциальной форме и фазовые скорости $\vec{U}_\alpha$ по закону Дарси имеют вид [13-15]:

$$\frac{\partial}{\partial t}\left(\sum\nolimits_\alpha m\rho_\alpha C_{\alpha j}\right) + div\left(\sum\nolimits_\alpha \rho_\alpha C_{\alpha j}\vec{U}_\alpha\right) = div\left(mD_{\alpha j}S_\alpha grad\left(\rho_\alpha C_{\alpha j}\right)\right)$$
$$\alpha = \overline{w,o,g}; \quad j = \overline{1,n} \tag{1}$$

$$u_\propto = -K\frac{k_{ra}}{\mu_\propto}\nabla(p_\propto - p_\propto gz), \quad \alpha = w,o,g; \tag{2}$$

где, m – пористость породы, S_α – насыщенности фаз, $\vec{U}_\alpha$ – скорости фильтрации фаз, $D_{j\propto}$ – коэффициент молекулярной диффузия компонента j в фазе α, K – абсолютная проницаемость, $k_{ra} = k_{ra}(S, T)$ – относительная фазовая проницаемость фазы α, $\mu_\alpha = \mu_\alpha(p_\alpha, T, C_{i\alpha})$ – вязкость фазы, $p_\propto = p_\propto(S)$ – давление фазы, $\rho_\propto$ – массовая плотность фазы, g – ускорение свободного падения, z – глубина.

Давления фаз связаны между собой следующими выражениями:

$$p_o = p_w + p_{cow}(S_w),\ p_g = p_o + p_{cog}(S_g), \tag{3}$$

где $p_{cow}(S_w, T)$, $p_{cog}(S_g, T)$ капиллярные давления на границе вода – нефть (*cow*) и нефть – газовой (*cog*). Эти зависимости считаются заданными функциями насыщенностей.

Циклическая и параллельная циклическая редукция. Для последовательной реализации метода ADI был использован алгоритм прогонки [9].

Для реализации параллельного метода ADI были выбраны алгоритмы циклической (CR) и параллельной циклической редукции (PCR). Алгоритм CR состоит из двух этапов: прямой и обратный ход. На прямом ходе система последовательно уменьшается до меньшей системы с половиной числа неизвестных, пока не будет достигнута система из одного неизвестного. На обратном ходе последовательно находятся остальные неизвестные, используя ранее найденные значения.

На каждом шаге прямого хода мы обновляем все уравнения с четным индексом параллельно с уравнением текущей системы как линейную комбинацию уравнений i, $i+1$ и $i-1$, так что мы выводим только систему неизвестных с четными индексами.

Уравнение i имеет вид $a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i$. Обновленные значения a_i, b_i, c_i и d_i находятся по следующим формулам:

$$a_i' = -a_{i-1}k_1, b_i' = b_i - c_{i-1}k_1 - a_{i+1}k_2$$

$$c_i' = -c_{i+1}k_2, d_i' = d_i - d_{i-1}k_1 - d_{i+1}k_2$$

$$k_1 = \frac{a_i}{b_{i-1}}, k_2 = \frac{c_i}{b_{i+1}}$$

На каждом шаге обратного хода мы решаем параллельно все x_i с нечетными индексами, подставляя уже решенные значения x_{i-1} и x_{i+1} в уравнение i:

$$x_i = \frac{d_i' - a_i' x_{i-1} - c_i' x_{i+1}}{b_i'}$$

Заметим, что для простоты в приведенном описании мы пренебрегаем специальной обработкой последнего уравнения и первого неизвестного, соответственно, в двух фазах алгоритма. Кроме того, мы решаем систему двух уравнений между двумя фазами алгоритма (Рисунок 1).

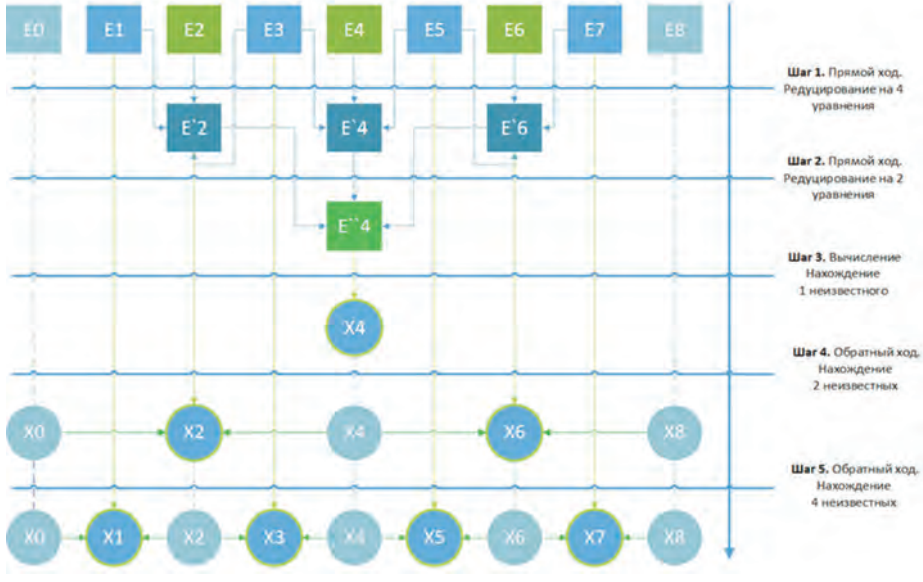

Рисунок 1 – Схема связей алгоритма CR для системы с восемью уравнениями

Параллельный алгоритм CR и последовательный алгоритм прогонки выполняют ряд операций, которые являются линейными по числу неизвестных. Алгоритм прогонки выполняет $8n$ операций, в то время как CR выполняет $17n$ операций. Однако на параллельном компьютере с n процессорами CR требует $2\log_2 n$ шагов, в то время как алгоритм прогонки требует $2n$ шагов.

PCR является усовершенствованным вариантом CR. В отличие от CR, PCR имеет только прямой ход. Хотя механизм редукции и формулы вычисления такие же, как у CR, на каждом этапе редукции PCR уменьшает каждую из существующих систем до

двух систем с половинным размером (Рисунок 2). PCR для завершения выполняет $12n\log_2 n$ операции и $\log_2 n$ шагов.

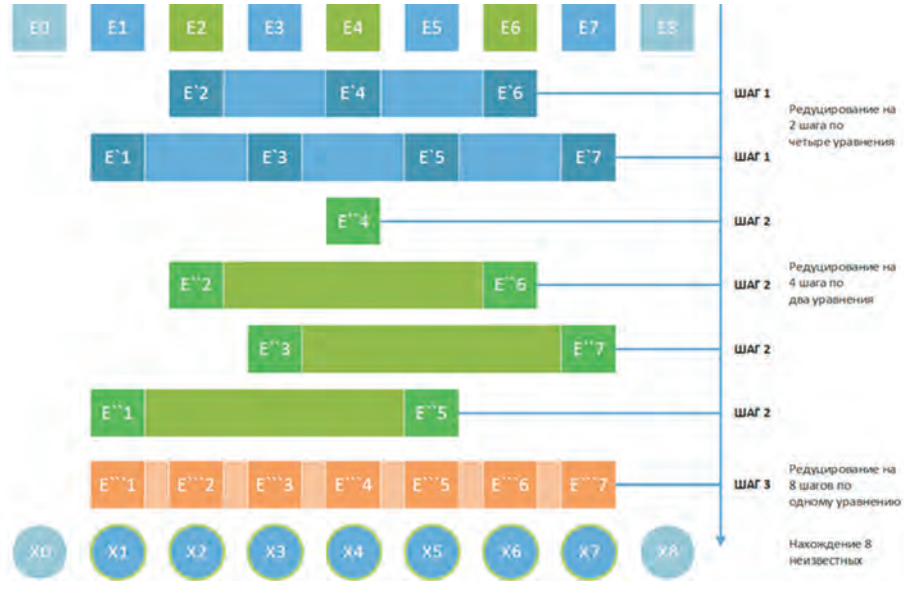

Рисунок 2 – Схема связей алгоритма PCR для системы с восемью уравнениями

Параллельный метод ADI. При параллельной реализации метода ADI на GPU с использованием технологий CUDA для всех реализаций копируем данные один раз с CPU на GPU и сохраняем все данные в памяти GPU, пока не вернем результат в CPU.

При реализации CR на CUDA сначала устанавливаем параметры выполнения ядра. Количество блоков CUDA равно количествам системы, так как каждая отдельная система сопоставляется с одним блоком CUDA. Количество потоков в одном блоке равно половине числа уравнений для каждой системы, так как мы начинаем с обновления только уравнения с четными индексами на первом алгоритмическом шаге.

Используем пять глобальных массивов для хранения трех диагоналей матрицы: нижняя диагональ, основная диагональ, верхняя диагональ, правая сторона и вектор решения. Объем разделяемой памяти, которую выделяем для каждого CUDA блока или каждой трехдиагональной системы равно system_size * 5 * sizeof (float) байт.

Прямой ход выполняет $\log_2$(system_size/2) алгоритмические шаги. Все пять массивов alist, blist, clist, dlist и xlist представляют собой массивы разделяемой памяти с данными, загружаемыми из глобальной памяти. Шаг начинается с 1 и удваивает значение каждого алгоритмического шага. Количество активных потоков уменьшается на половину на каждом алгоритмическом шаге.

В начале обратного хода мы решаем систему уравнений с двумя уравнениями, полученную на заключительном этапе прямого хода. Затем выполняем $\log_2$(system_size/2) алгоритмические шагов для решения всех остальных неизвестных. Соответ-

ственно алгоритму удваиваем количество активных потоков на каждом алгоритмическом шаге.

При реализации PCR на CUDA, как и в методе CR, устанавливаем количество CUDA блоков на количество систем. Но в отличие от CR устанавливаем количество потоков в каждом блоке на количество уравнений в каждой системе, так как в PCR обновляем все уравнения, а не половинные уравнения, на алгоритмическом этапе. Кроме того, число активных потоков остается постоянным на каждом алгоритмическом этапе.

На персональных компьютерах с графическими ускорителями, где вычислительные части программы выгружаются на GPU через шину PCI Express, программа должна минимизировать передачу данных между хостом и устройством. Поэтому лучшим сценарием является то, что данные копируются на устройство только один раз в начале и сохраняются до конца вычисления.

Результаты. В этом разделе мы тестируем параллельный алгоритм ADI, используя двумерную задачу многофазной фильтрации многокомпонентной жидкости в пористых средах на различных размерах сетки. Для тестирования программы использовали персональный компьютер с восьмиядерным процессором Intel Core i7-3770 с тактовой частотой 3.4 ГГц и памятью 8 Гб, графической картой Nvidia RTX 2080 с видеопамятью 8 ГБ, с операционной системой Windows 10 с установленной CUDA 10. В качестве языка программирования был выбран C/C++.

Параметры задачи установлены следующим образом: размер сетки (N) является однородным в обоих направлениях $ch = \Delta x = \Delta y = 1 / (N - 1)$, числовой временной шаг Δt составляет 0.001, а время составляет 0.1, поэтому количество итерации по времени составляет 100. Размер сетки должен быть равен степени двойки плюс один. Размер CUDA блока зависит от размера N, из-за этого есть ограничение на размер сетки.

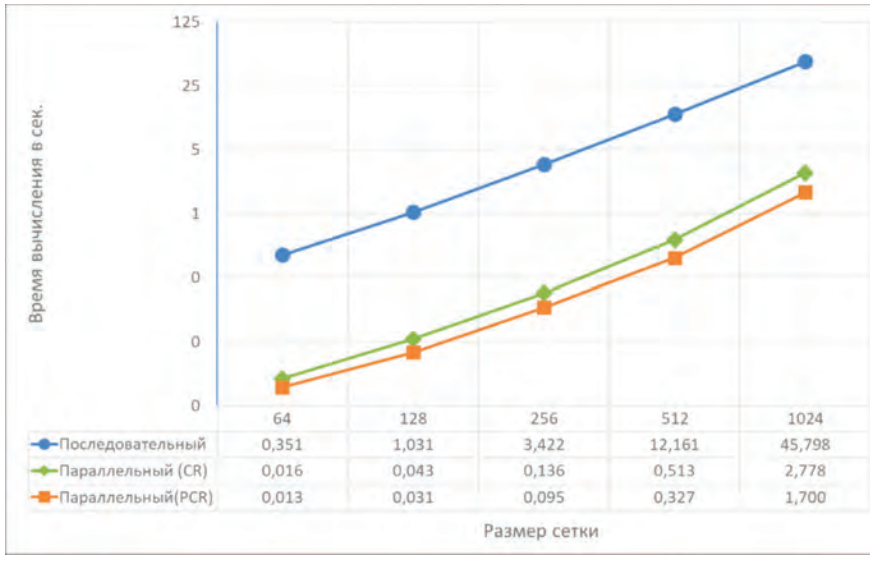

Рисунок 3 – Сравнительный график результатов времени вычисления на различных размерах сетки

В результате мы сравнили время вычисления трех программ: последовательная программа, реализованная с помощью метода прогонки и параллельные программы, реализованные с помощью CR и PCR на CUDA. Результаты тестирования показали, что параллельные программы на графическом процессоре по сравнению с последовательной программой 16-37 раз быстрее работает (рисунок 3). Из рисунка 3 можно заметить, что параллельная PCR программа быстрее работает чем параллельная программа, реализованная с помощью CR. Рисунок 2 показывает, что алгоритм PCR имеет только один прямой ход и на каждом шаге редукции количество уравнении сокращается. За счет этого по сравнению с CR программой уменьшается время вычисления.

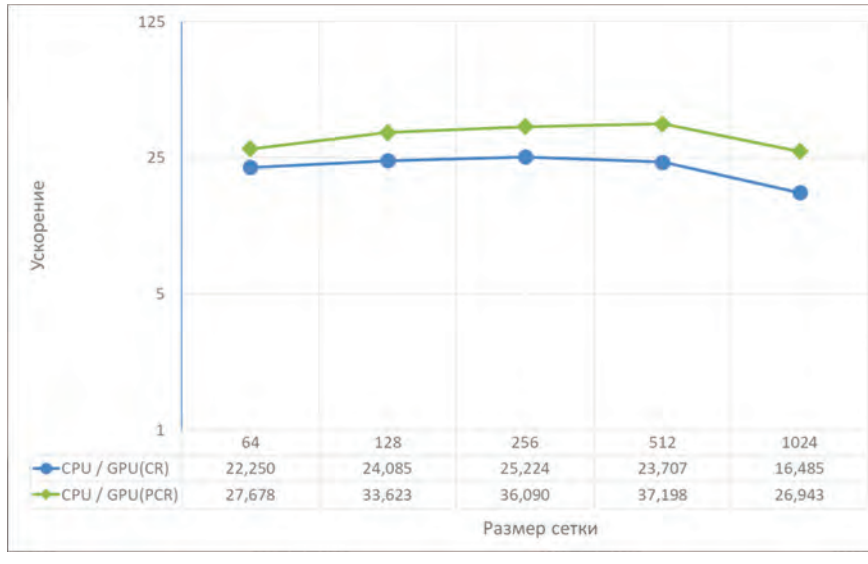

Рисунок 4 – Сравнительный график ускорении параллельных алгоритмов (CR и PCR) на различных размерах сетки

Рисунок 4 показывает, что ускорение по увеличению размера сетки уменьшается. Это связана с накладными расходами при синхронизации данных между хостом и девайсом. В параллельной программе на девайсе выделяются память для 51 двумерного массива с типом float. Примерная требуемая память для этих массивов равна 51 * system_size * system_size * sizeof(float). Этот параметр нужно учитывать перед каждым запуском.

Заключение. В работе были реализованы параллельные алгоритмы ADI на CUDA для решения задач многофазной фильтрации многокомпонентной жидкости в пористых средах. Результаты тестирования показали, что алгоритм PCR на CUDA быстрее работает за счет меньшего количества операции по сравнению с алгоритмом CR. На всех алгоритмах есть ограничения по памяти. Учитывая эти ограничения, следующим этапом работы является:

– реализация распределения данных одной системы на несколько CUDA блоков для снятия ограничения размерности вычислительной сетки;

– тестирование разработанных алгоритмов на мобильных устройствах с поддержкой технологий CUDA;

– реализация гибридных алгоритмов для задач многофазной фильтрации многокомпонентной жидкости в пористых средах;

– разработка алгоритма запуска параллельных программ на нескольких графических процессорах.

Работа выполнена в рамках проекта № AP05130366-OT-19 «Разработка интеллектуальной высокопроизводительной информационной системы анализа технологий повышения нефтеотдачи пласта iFields-II» за счет грантового финансирования МОН РК.

ЛИТЕРАТУРА

1 Peaceman D., Rachford H. The numerical solution of parabolic and elliptic diferential equations // Journal of the Society for Industrial & Applied Mathematics. – 1955. – V.3. – N 1. – P. 28-41.

2 Casulli V., Cheng R. Semi-implicit finite diference methods for three-dimensional shallow water flow // International Journal for numerical methods in fluids. – 2005. – V.15. – N 6. – P. 629-648.

3 Ellner N., Wachspress E. Alternating direction implicit iteration for systems with complex spectra // SIAM journal on numerical analysis. – 1991. – V.28. – N 3. – P. 859-870.

4 Lindemuth I., Killeen J. Alternating direction implicit techniques for two-dimensional magnetohydrodynamic calculations // Journal of Computational Physics. – 1972. – V.13 – N 2. – P. 181-208.

5 Namiki T. A new FDTD algorithm based on alternating-direction implicit method // Microwave Theory and Techniques, IEEE Transactions on 47 (10) (1999) 2003–2007.

6 Wu W. Computational river dynamics // CRC. – 2007.

7 Hockney R. A fast-direct solution of Poisson's equation using Fourier analysis // Journal of the ACM. – 1995. – V.12 – N 1. – P. 95-113. 8 Hockney R., Jesshope C. Parallel computers. – 1981.

9 Thomas L.H. Elliptic Problems in Linear Differential Equations over a Network // Watson Sci. Comput. Lab Report. – New York, 1949.

10 Jang B., Kaeli D., Do S., Pien H. Multi GPU implementation of iterative tomographic reconstruction algorithms // IEEE International Symposium on Biomedical Imaging: From Nano to Macro. – 2009. – P. 185-188.

11 Zhang Y., Jia Y. Parallelization of implicit CCHE2D model using CUDA programming techniques // World Environment & Water Resources Congress 2013. – USA, 2013. – P. 1777-1792.

12 Zhang Y., Jia Y. Parallelized CCHE2D model with CUDA Fortran on graphics process units // Computers and Fluids. – 2013. – P. 359-368.

13 Zhangxin Chen. Reservoir Simulation: Mathematical Techniques in Oil Recovery. – Society for Industrial and Applied Mathematics, 2007. – 250 p.

14 Борисов В.Е., Критский Б.В. и др. Композиционная неизотермическая модель фильтрации в пористой среде с учетом химических реакций и активной твердой фазы. – Москва, 2013. – 32 с.[Borisov V.E., Kritskij B.V. i dr. Kompozicionnaya neizotermicheskaya model' fil'tracii v poristoj srede s uchetom himicheskih reakcij i aktivnoj tverdoj fazy. – Moskva, 2013. – 32 s.]

15 Imankulov T.S., Akhmed-Zaki D.Zh., Daribayev B.S. and et al. Intellectual system for analyzing thermal compositional modeling with chemical reactions // 16th European Conference on the Mathematics of Oil Recovery. – 2018.

Б. С. ДӘРІБАЕВ[1], Т. С. ИМАНКУЛОВ[1], Д. Ж. АХМЕД-ЗАКИ[2]

[1]әл-Фараби атындағы Қазақ ұлттық университеті
[2]Халықаралық бизнес университеті

КЕУЕК ОРТАДА КӨПФАЗАЛЫ КӨПКОМПОНЕНТТІ СҰЙЫҚТЫҚТЫҢ ФИЛЬТРАЦИЯ ЕСЕПТЕРІН ШЕШУГЕ АРНАЛҒАН CUDA ПАРАЛЛЕЛЬ АЛГОРИТМІ

Мақалада кеуек орталарда көп компонентті сұйықтықтағы көп фазалы фильтрация есебін шешу үшін CUDA технологиясын қолдану арқылы бойлық-көлденең қуалау әдісінің параллель алгоритмін жүзеге асыру қарастырылады. Бойлық-көлденең қуалаудың үш диагоналды теңдеулер жүйесін шешу үшін циклдік және параллель циклдік редукция әдістері қолданылды. Зерттеудің нәтижелері циклдік және параллель циклдік редукция әдістері орталық процессорларға қарағанда заманауи графикалық процессорларда әлдеқайда тиімді жүзеге асырылатынын көрсетті.

Түйін сөздер: *CUDA, бойлық-көлденең қуалау әдісі, циклдік редукция, параллель циклдік редукция, көп фазалық фильтрация.*

B. S. DARIBAYEV[1], T. S. IMANKULOV[1], D. ZH. AKHMED-ZAKI[2]

[1]al-Farabi Kazakh National University
[2]University of International Business

PARALLEL ALGORITHM ON CUDA FOR SOLVING MULTIPHASE, MULTICOMPONENT FLUID FILTRATION PROBLEMS IN POROUS MEDIA

The paper discusses the implementation of alternating direction implicit parallel algorithm with CUDA technology to solve the problems of multiphase filtration of a multicomponent fluid in porous media. To solve the tridiagonal equations systems of alternating direction implicit method, cyclic and parallel cyclic reduction methods were used. The results of the study showed that the implementation of cyclic and parallel cyclic reduction algorithms on modern GPUs is much more efficient than on CPUs.

Key words: *CUDA, alternating direction implicit method, cyclic reduction, parallel cyclic reduction, multiphase filtration.*