

APPROACH TO EFFECTIVE IMPLEMENTATION OF NUMERICAL ALGORITHMS

V. N. Aleeva

South Ural State University (National Research University),
454080, Chelyabinsk, Russia

DOI: 10.24412/2073-0667-2025-1-29-44

EDN: QALLET

The problem of effective implementation of numerical algorithms is actual. The paper proposes a solution to this problem. The solution uses the author's concept of a Q -determinant. In the paper, we describe the notions of the concept of a Q -determinant used for research. These are the following notions.

An algorithmic problem has the form $\bar{y} = F(N, B)$, where $N = \{n_1, \dots, n_k\}$ is the set of parameters of the problem dimension or N is the empty set, B is the set of input data, $\bar{y} = \{y_1, \dots, y_m\}$ is the set of output data. Here n_i ($i \in \{1, \dots, k\}$) is any positive integer. If $N = \{n_1, \dots, n_k\}$, then by $\bar{N} = \{\bar{n}_1, \dots, \bar{n}_k\}$ we denote a set of k positive integers, where $\bar{n}_i$ is some given value of the parameter n_i for each $i \in \{1, \dots, k\}$. We denote the set of all possible k -tuples $\bar{N}$ by $\{\bar{N}\}$. An algorithm for solving an algorithmic problem is denoted by $\mathcal{A}$, and the set of operations used by algorithm $\mathcal{A}$ is denoted by Q .

An expression over B and Q is a term in the standard sense of mathematical logic. A chain of length n is the result of applying some associative operation from Q to n expressions. If $N = \emptyset$, then any expression w over B and Q is an unconditional Q -term. If $N \neq \emptyset$ and V is the set of all expressions over B and Q , then any mapping $w : \{\bar{N}\} \rightarrow V \cup \emptyset$ is also called an unconditional Q -term. $w(\bar{N}) = \emptyset$ means that the value of $w(\bar{N})$ is undefined. If $N = \emptyset$ and the expression w over B and Q has a value of logical type under any interpretation of the variables B , then we call the unconditional Q -term w an unconditional logical Q -term. Let $N \neq \emptyset$ and w be an unconditional Q -term. Suppose also that the expression $w(\bar{N})$ for each $\bar{N} \in \{\bar{N}\}$ has a value of logical type under any interpretation of the variables B . In this case, we call the unconditional Q -term w an unconditional logical Q -term. Suppose that $u_1, \dots, u_l$ are unconditional logical Q -terms, $w_1, \dots, w_l$ are unconditional Q -terms. Then the set l of pairs $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$ is a conditional Q -term of length l . Let $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i=1, 2, \dots}$ be a countable set of pairs of unconditional Q -terms. Suppose that $\{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$ is a conditional Q -term of length l for any $l < \infty$. Then we call $(\hat{u}, \hat{w})$ a conditional infinite Q -term. Q -terms can be calculated.

Suppose that algorithm $\mathcal{A}$ consists in finding for each $i \in \{1, \dots, m\}$ the value y_i by computing the value of the Q -term f_i . Then the set of Q -terms $\{f_i \mid i \in \{1, \dots, m\}\}$ we call the Q -determinant of the algorithm $\mathcal{A}$. Moreover, we call the system of equations $\{y_i = f_i \mid i \in \{1, \dots, m\}\}$ a representation of the algorithm $\mathcal{A}$ in the form of a Q -determinant.

The process of computing Q -terms $\{f_i \mid i \in \{1, \dots, m\}\}$ of algorithm $\mathcal{A}$ is called an implementation of algorithm $\mathcal{A}$. An implementation of algorithm $\mathcal{A}$ is called parallel if there are operations of the algorithm that are executed simultaneously. An implementation of algorithm $\mathcal{A}$ is called Q -effective if Q -terms $\{f_i \mid i \in \{1, \dots, m\}\}$ are computed simultaneously, operations are executed as they are ready, and chain operations are computed using the doubling scheme. A Q -effective implementation

uses the entire parallelism resource of the algorithm. The concepts of height and width of an algorithm characterize the parallelism resource of the algorithm. An implementation of the algorithm $\mathcal{A}$ is called realizable if it is such that a finite number of operations must be executed simultaneously. There are algorithms such that the Q -effective implementation is not realizable.

We can use the Q -determinant of a numerical algorithm for design parallel programs implementing the algorithm, including effective programs. On the base of this idea, the author has developed a method for designing effective programs. In this paper, we describe the proposed method. The method involves the following steps: construction of the Q -determinant of the algorithm, description of the Q -effective implementation of the algorithm, development of a program for an realizable Q -effective implementation of the algorithm. We call Q -effective programs developed using the effective program design method. Therefore, we can also call this method the method of designing Q -effective programs. Note that Q -effective programs use the parallelism resource of algorithms completely. In the paper, we provide an overview of the numerical algorithms for which Q -effective programs have been developed. We also describe the features of applying the design method of Q -effective programs to algorithms. Note that we are considering algorithms that solve various algorithmic problems and have a different structure of Q -determinants.

In addition, we present the following results of a study of Q -effective programs. 1) We introduce the concept of the computing infrastructure of the program. This is a set of conditions for the development and execution of a program. We prove that each of the Q -effective programs for a given numerical algorithm is the most efficient for its computing infrastructure among programs implementing the same algorithm.

2) We show that any numerical algorithm can be implemented using a potentially infinite set of Q -effective programs for various computing infrastructures. From this, we can conclude that of all the Q -effective programs for this algorithm, there is probably not the most efficient one. However, each of the Q -effective programs is effective for its computing infrastructure.

So, we can say that the method of designing Q -effective programs has theoretical and practical importance.

Key words: improving parallel computing efficiency, Q -determinant of algorithm, representation of algorithm in form of Q -determinant, Q -effective implementation of algorithm, parallelism resource of algorithm, Q -effective program.

References

1. Aleeva V., Aleev R. Investigation and Implementation of Parallelism Resources of Numerical Algorithms // ACM Transactions on Parallel Computing. 2023. Vol. 10. N 2, Article number 8. P. 1–64. DOI: 10.1145/3583755.
2. Aleeva V. N. Analiz parallel'nyx chislennykh algoritmov. Preprint № 590. Novosibirsk: VC SO AN SSSR, 1985. 23 s. (in Russian)
3. Ershov YU. L., Palyutin E. A. Matematicheskaya logika. M.: Nauka, 1987. 336 s. (in Russian)
4. Aleeva V. N., Zotova P. S., Skleznev D. S. Rasshirenije vozmozhnostej issledovaniya resursa parallelizma chislennykh algoritmov s pomoshh'yu programmnoj Q -sistemy // Vestnik YuUrGU. Seriya: Vychislitel'naya matematika i informatika. 2021. T. 10, N 2. S. 66–81. DOI: 10.14529/cmse210205. (in Russian)
5. Aleeva V. N. Improving Parallel Computing Efficiency // Proceedings — 2020 Global Smart Industry Conference, GloSIC 2020. IEEE. 2020. P. 113–120. Article number 9267828. DOI: 10.1109/GloSIC50886.2020.9267828.
6. Aleeva V. N., Aleev R. Zh. High-Performance Computing Using Application of Q -determinant of Numerical Algorithms // Proceedings — 2018 Global Smart Industry Conference, GloSIC 2018. IEEE. 2018. 8 p. Article number 8570160. DOI: 10.1109/GloSIC.2018.8570160.

7. Aleeva V., Bogatyreva E., Skleznev A., et al. Software Q -system for the Research of the Resource of Numerical Algorithms Parallelism // Supercomputing. RuSCDays 2019. Communications in Computer and Information Science. 2019. Vol. 1129. P. 641–652. DOI: 10.1007/978-3-030-36592-9_52.
8. McColl W.F. General Purpose Parallel Computing // Lectures on Parallel Computation, Cambridge International Series on Parallel Computation. USA: Cambridge University Press, 1993. P. 337–391.
9. Valiant L. G. A bridging model for parallel computation // Communications of the ACM. 1990. Vol. 33, no. 8. P. 103–111. DOI: 10.1145/79173.79181.
10. Leung J. Y.-T., Zhao H. Scheduling problems in master-slave model // Annals of Operations Research. 2008. Vol. 159. P. 215–231. DOI: 10.1007/s10479-007-0271-4.
11. Aleeva V. Designing a Parallel Programs on the Base of the Conception of Q -Determinant // Supercomputing. RuSCDays 2018. Communications in Computer and Information Science. 2019. Vol. 965. P. 565–577. DOI: 10.1007/978-3-030-05807-4_48.
12. Superkomp'yuter "Tornado YuUrGU". [Electron. Res.]: <http://supercomputer.susu.ru/computers/tornado/>. Accessed: 17.09.2024. (in Russian)
13. Otkrytaya enciklopediya svoystv algoritmov. [Electron. Res.]: <https://algowiki-project.org/ru>. Accessed: 17.09.2024. (in Russian).
14. Val'kevich N. V. Q -effektivnaya realizaciya algoritma ymnozheniya matric na superkomp'yutere "Tornado YuUrGU" // Vyp. kvalif. rabota bakalavra po napravleniyu "Fundamental'naya informatika i informacionnye texnologii": 02.03.02 / Yuzhnoural'skij gosudarstvennyj universitet. Chelyabinsk. 2017. 33 l. [Electron. Res.]: <http://omega.sp.susu.ru/publications/bachelorthesis/17-Valkevich.pdf>. Accessed: 17.09.2024. (in Russian)
15. Tarasov D. E. Q -effektivnyj kodizain realizacii metoda Gaussa–Zhordana na superkomp'yutere "Tornado YuUrGU" // Vyp. kvalif. rabota magistra po napravleniyu "Fundamental'naya informatika i informacionnye texnologii": 02.04.02 / Yuzhnoural'skij gosudarstvennyj universitet. Chelyabinsk. 2017. 41 l. [Electron. Res.]: <http://omega.sp.susu.ru/publications/masterthesis/17-Tarasov.pdf>. Accessed: 17.09.2024. (in Russian)
16. Lapteva Yu. S. Q -effektivnaya realizaciya metoda Yakobi dlya resheniya SLAU na superkomp'yutere "Tornado YuUrGU" // Vyp. kvalif. rabota bakalavra po napravleniyu "Fundamental'naya informatika i informacionnye texnologii": 02.03.02 / Yuzhnoural'skij gosudarstvennyj universitet. Chelyabinsk. 2017. 30 l. [Electron. Res.]: <http://omega.sp.susu.ru/publications/bachelorthesis/17-Lapteva.pdf>. Accessed: 20.09.2024. (in Russian)
17. Bazhenova L. A. Primenenie metoda proektirovaniya Q -effektivnoj programmy dlya resheniya sistemy setochnyx uravnenij // Vyp. kvalif. rabota bakalavra po napravleniyu «Fundamental'naya informatika i informacionnye texnologii»: 02.03.02 / Yuzhnoural'skij gosudarstvennyj universitet. Chelyabinsk. 2018. 30 l. [Electron. Res.]: <http://omega.sp.susu.ru/publications/bachelorthesis/18-Bazhenova.pdf>. Accessed: 20.09.2024. (in Russian)
18. Kondakova A. S. Razrabotka Q -effektivnoj programmy dlya resheniya pyatitochechnyx raznostnyx uravnenij i issledovanie eyo dinamicheskix xarakteristik // Vyp. kvalif. rabota magistra po napravleniyu "Fundamental'naya informatika i informacionnye texnologii": 02.04.02 / Yuzhnoural'skij gosudarstvennyj universitet. Chelyabinsk. 2019. 40 l. [Electron. Res.]: http://omega.sp.susu.ru/publications/masterthesis/2019_220_kondakovaas.pdf. Accessed: 20.09.2024. (in Russian)
19. Necheporenko A. D. Razrabotka Q -effektivnoj programmy dlya resheniya SLAU metodom Gaussa–Zejdelya // Vyp. kvalif. rabota bakalavra po napravleniyu "Fundamental'naya informatika i informacionnye texnologii": 02.03.02 / Yuzhnoural'skij gosudarstvennyj universitet. Chelyabinsk. 2018. 32 l. [Electron. Res.]: <http://omega.sp.susu.ru/publications/bachelorthesis/18-Necheporenko.pdf>. Accessed: 20.09.2024. (in Russian)

20. Aleeva V. N., Shatov M. B. Primenenie koncepcii Q -determinanta dlya effektivnoj realizacii chislennykh algoritmov na primere metoda copyazyonnykh gradientov dlya resheniya sistemy linejnykh uravnenij // Vestnik YuUrGU. Seriya: Vychislitel'naya matematika i informatika. 2021. T. 10, N 3. S. 56–71. DOI: 10.14529/cmse210304. (in Russian)
21. Dijkstra E. W. A note on two problems in connexion with graphs // Numerische Mathematik. 1959. Vol. 1. P. 269–271. DOI: 10.1007/BF01386390.
22. Aleeva V. N., Manatin P. A. Primenenie metoda proektirovaniya Q -effektivnykh programm dlya algoritma Dejkstry // Vestnik YuUrGU. Seriya: Vychislitel'naya matematika i informatika. 2023. T. 12, N 2. S. 62–77. DOI: 10.14529/cmse230203. (in Russian)
23. SCISPACE. Answer to the question “How does parallel computing improve the efficiency of AI algorithms?”. [Electron. Res.]: <https://typeset.io/questions/how-does-parallel-computing-improve-the-efficiency-of-ai-5gkdumloct>. Accessed: 23.09.2024.
24. Sapozhnikov A. S. Primenenie metoda proektirovaniya Q -effektivnykh programm k metodu stoxasticheskogo gradientnogo spuska dlya obucheniya nejronnykh setej // Vyp. kvalif. rabota magistra po napravleniyu. Programmnyaya inzheneriya: 09.04.04 / Yuzhnoural'skij gosudarstvennyj universitet. Chelyabinsk. 2024. 43 l. [Electron. Res.]: https://sp.susu.ru/student/masterthesis/2024_229_sapoznikovas.pdf. Accessed: 23.09.2024. (in Russian)
25. Voevodin V. V., Voevodin V. V. Parallel'nye vychisleniya. SPb.: BXV-Peterburg, 2002. 608 s. (in Russian)
26. Aleeva V. N., Avtomatizirovannoe proektirovanie i ispolnenie effektivnykh programm dlya chislennykh algoritmov // Vestnik YuUrGU. Seriya: Vychislitel'naya matematika i informatika. 2023. T. 12, N 3. S. 31–49. DOI: 10.14529/cmse230303. (in Russian)

ПОДХОД К ЭФФЕКТИВНОЙ РЕАЛИЗАЦИИ ЧИСЛЕННЫХ АЛГОРИТМОВ

В. Н. Алеева

Южно-Уральский государственный университет (НИУ),
454080, Челябинск, Россия

УДК 004.021, 004.032.24, 004.051, 004.272

DOI: 10.24412/2073-0667-2025-1-29-44

EDN: QALLET

В статье рассмотрен метод проектирования эффективных программ для реализации численных алгоритмов, основанный на авторской концепции Q -детерминанта. Разработанные с помощью метода эффективные программы называются Q -эффективными. Они используют ресурс параллелизма алгоритмов полностью. Приводится обзор численных алгоритмов, для которых разработаны Q -эффективные программы. Введено понятие вычислительной инфраструктуры программы. Доказано, что каждая из Q -эффективных программ, реализующих один и тот же численный алгоритм, является самой эффективной для своей вычислительной инфраструктуры среди программ, реализующих тот же алгоритм. Проведенное исследование обосновывает теоретическую и практическую значимость метода проектирования эффективных программ для реализации численных алгоритмов.

Ключевые слова: повышение эффективности параллельных вычислений, Q -детерминант алгоритма, представление алгоритма в форме Q -детерминанта, Q -эффективная реализация алгоритма, ресурс параллелизма алгоритма, Q -эффективная программа.

Введение. Основная цель параллельных вычислений — ускорение решения вычислительных задач. Она достигается за счет распараллеливания алгоритмов, применяемых для решения задач. Ускорение решения задачи зависит от того, насколько реализация алгоритма использует ресурс параллелизма алгоритма. Самая эффективная реализация алгоритма использует ресурс параллелизма алгоритма полностью. При выполнении она требует больше вычислителей (процессоров, ядер) параллельной вычислительной системы (ПВС), чем любая другая реализация алгоритма, так как одновременно должно выполняться больше операций. Следовательно, постоянный рост вычислительных мощностей ПВС способствует эффективной реализации алгоритмов.

На практике, как правило, параллельные программы выполняют не самые эффективные реализации алгоритмов. Повышение эффективности параллельных вычислений дает возможность уменьшить время их выполнения и увеличить быстродействие ПВС. Таким образом, проблема повышения эффективности параллельных вычислений является актуальной. В работе [1] показано, что решение данной проблемы заключается в исследовании ресурсов параллелизма алгоритмов и реализации ресурсов параллелизма алгоритмов на ПВС. Статья посвящена результатам исследований, полученным при реализации ресурсов параллелизма алгоритмов на ПВС на основе авторского подхода к распараллеливанию численных алгоритмов — концепции Q -детерминанта.

Цель исследования, изложенного в статье, состоит в обосновании теоретической и практической значимости метода проектирования эффективных программ для реализации численных алгоритмов, основанного на концепции Q -детерминанта. Она предполагает решение следующих задач.

- 1) Применение метода проектирования эффективных программ для коллекции алгоритмов, реализующих различные алгоритмические проблемы.
- 2) Исследование свойств разработанных эффективных программ и описание его результатов.

Статья организована следующим образом. Раздел 1 содержит некоторые основные понятия концепции Q -детерминанта, используемые в статье. В разделе 2 изложен метод проектирования эффективных программ. В разделе 3 показано применение метода проектирования эффективных программ для конкретных алгоритмов, реализующих различные алгоритмические проблемы. В разделе 4 описаны результаты исследования свойств программ, разработанных с помощью метода проектирования эффективных программ. В заключении подводятся итоги работы и формулируется направление развития исследований на основе концепции Q -детерминанта.

1. Концепция Q -детерминанта. Концепция Q -детерминанта является теоретической основой данного исследования. Впервые она изложена в работе автора [2], затем в последующих работах, при этом наиболее полно в [1]. Опишем понятия концепции Q -детерминанта, используемые при исследовании.

Обозначения. Алгоритмическую проблему будем представлять в виде $\bar{y} = F(N, B)$, где $N = \{n_1, \dots, n_k\}$ — множество параметров размерности проблемы или N — пустое множество, B — множество входных данных, $\bar{y} = \{y_1, \dots, y_m\}$ — множество выходных данных, при этом целое число t является либо константой, либо значением вычислимой функции параметров N при условии, что $N \neq \emptyset$. Здесь n_i ($i \in \{1, \dots, k\}$) равно любому положительному целому числу. Если $N = \{n_1, \dots, n_k\}$, то через $\bar{N} = \{\bar{n}_1, \dots, \bar{n}_k\}$ обозначим набор из k положительных целых чисел, где $\bar{n}_i$ — некоторое заданное значение параметра n_i для каждого $i \in \{1, \dots, k\}$. Через $\{\bar{N}\}$ обозначим множество всех возможных k -наборов $\bar{N}$. Пусть $\mathcal{A}$ — алгоритм для решения алгоритмической проблемы, Q — множество операций, используемых алгоритмом $\mathcal{A}$.

Определение 1. Выражение над B и Q определим как терм в стандартном смысле математической логики [3].

Определение 2. Цепочкой длины n будем называть выражение, представляющее собой результат применения некоторой ассоциативной операции из Q к n выражениям.

Определение 3. Если $N = \emptyset$, то любое выражение w над B и Q мы называем безусловным Q -термом. Пусть $N \neq \emptyset$ и V — множество всех выражений над B и Q . Тогда любое отображение $w : \{\bar{N}\} \rightarrow V \cup \emptyset$ мы также называем безусловным Q -термом. $w(\bar{N}) = \emptyset$ означает, что значение $w(\bar{N})$ не определено.

Определение 4. Пусть $N = \emptyset$ и w — безусловный Q -терм. Кроме того, пусть выражение w над B и Q имеет значение логического типа при любой интерпретации переменных B . Тогда безусловный Q -терм w мы называем безусловным логическим Q -термом. Пусть $N \neq \emptyset$ и w — безусловный Q -терм. Предположим также, что выражение $w(\bar{N})$

для каждого $\bar{N} \in \{\bar{N}\}$ имеет значение логического типа при любой интерпретации переменных B . В этом случае безусловный Q -терм w мы называем безусловным логическим Q -термом.

Определение 5. Предположим, что $u_1, \dots, u_l$ — безусловные логические Q -термы, $w_1, \dots, w_l$ — безусловные Q -термы. Тогда множество l пар $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$ мы называем условным Q -термом длины l .

Определение 6. Пусть $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i=1,2,\dots}$ — счетное множество пар безусловных Q -термов. Предположим, что $\{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$ является условным Q -термом длины l для любого $l < \infty$. Тогда мы называем $(\hat{u}, \hat{w})$ условным бесконечным Q -термом.

Q -термы можно вычислять. Опишем, как найти значение безусловного Q -терма w при интерпретации переменных B . Если $N = \emptyset$, то нахождение значения выражения w является нахождением значения безусловного Q -терма w при любой интерпретации переменных B . Если $N \neq \emptyset$ и $w(\bar{N}) \neq \emptyset$, то $w(\bar{N})$ является выражением над B и Q . Можно найти значение выражения $w(\bar{N})$ при любой интерпретации переменных B , которое является значением безусловного Q -терма w . Конечно, мы опускаем значение $w(\bar{N}) = \emptyset$, так как в этом случае значение безусловного Q -терма w не определено.

Теперь опишем, как найти значение условного Q -терма $(\hat{u}, \hat{w})$ при интерпретации переменных B . Пусть $N = \emptyset$. Находим значения выражений u_i, w_i для $i \in \{1, \dots, l\}$. При этом мы можем найти пару u_{i_0}, w_{i_0} такую, что u_{i_0} имеет значение **true**, тогда считаем, что $(\hat{u}, \hat{w})$ имеет значение w_{i_0} . В противном случае считаем, что значение $(\hat{u}, \hat{w})$ при интерпретации переменных B не определено. Пусть $N \neq \emptyset$ и $\bar{N} \in \{\bar{N}\}$. Находим выражения $u_i(\bar{N}), w_i(\bar{N})$ для $i \in \{1, \dots, l\}$. При этом мы можем найти пару $u_{i_0}(\bar{N}), w_{i_0}(\bar{N})$ такую, что $u_{i_0}(\bar{N})$ имеет значение **true**, тогда считаем, что $(\hat{u}, \hat{w})$ имеет значение $w_{i_0}(\bar{N})$. В противном случае считаем, что значение $(\hat{u}, \hat{w})$ для $\bar{N}$ и при такой интерпретации переменных B не определено. Аналогично можно определить значение условного бесконечного Q -терма.

Определение 7. Пусть $M = \{1, \dots, t\}$. Предположим, что алгоритм $\mathcal{A}$ состоит в нахождении для каждого $i \in M$ значения y_i путем вычисления значения Q -терма f_i . Тогда множество Q -термов $\{f_i \mid i \in M\}$ мы называем Q -детерминантом алгоритма $\mathcal{A}$, а систему уравнений $\{y_i = f_i \mid i \in M\}$ представлением алгоритма $\mathcal{A}$ в форме Q -детерминанта.

Определение 8. Процесс вычисления Q -термов $\{f_i \mid i \in M\}$ алгоритма $\mathcal{A}$ называется реализацией алгоритма $\mathcal{A}$. Реализация алгоритма $\mathcal{A}$ называется параллельной, если существуют операции, которые выполняются одновременно.

Определение 9. Реализация алгоритма $\mathcal{A}$ называется Q -эффективной, если Q -термы $\{f_i \mid i \in M\}$ вычисляются одновременно, операции при их вычислении выполняются по мере готовности, при этом, если несколько операций цепочки готовы к выполнению, то они выполняются по схеме сдваивания.

Замечание 1. Определение Q -эффективной реализации показывает, что она полностью использует ресурс параллелизма алгоритма.

Ресурс параллелизма алгоритма характеризуют его высота и ширина. Эти понятия рассматриваются в работах [4–7]. В [4, 7] описана программная Q -система, разработанная для автоматизированного исследования ресурса параллелизма алгоритмов, а также сравнения ресурсов параллелизма алгоритмов, решающих одну и ту же алгоритмическую проблему.

Определение 10. *Реализация алгоритма A называется выполнимой, если одновременно должно выполняться конечное (непустое) множество операций.*

Замечание 2. Существуют алгоритмы, Q -эффективная реализация которых невыполнима. Пример такого алгоритма приведен в работе [4].

2. Метод проектирования эффективных программ. Хорошо известно, что блок-схему алгоритма можно применять для разработки последовательной программы, реализующей алгоритм. В основе метода проектирования эффективных программ лежит аналогичная идея, заключающаяся в том, что Q -детерминант численного алгоритма можно использовать для разработки параллельных программ, реализующих алгоритм, в том числе эффективных.

Определение 11. *Модель концепции Q -детерминанта, описанную в разделе 1, будем называть базовой моделью концепции Q -детерминанта.*

С помощью базовой модели концепции Q -детерминанта можно изучать машинно-независимые свойства алгоритмов, но исследовать особенности реализации алгоритмов на реальных ПВС нельзя, так как она не содержит информации о ПВС. Поэтому расширим базовую модель и назовем расширенной моделью концепции Q -детерминанта.

Определение 12. *Расширенная модель концепции Q -детерминанта — это результат добавления к базовой модели концепции Q -детерминанта моделей параллельных вычислений PRAM [8] для общей памяти и BSP [9] для распределенной памяти.*

Выбор этих моделей объясняется тем, что деление ПВС на системы с общей памятью и с распределенной памятью общепринято, кроме того, автор имеет возможность проводить исследования с использованием ПВС как с общей, так и с распределенной памятью.

Мы используем расширенную модель концепции Q -детерминанта, чтобы предложить метод проектирования эффективных программ. Метод состоит из трех этапов: 1) построение Q -детерминанта алгоритма; 2) описание Q -эффективной реализации алгоритма; 3) разработка программы для выполнимой Q -эффективной реализации алгоритма.

На первом и втором этапах метода используется базовая модель концепции Q -детерминанта, а на третьем этапе расширенная модель.

Определение 13. *Программу будем называть Q -эффективной, если она разработана с помощью данного метода.*

Дадим еще одно определение Q -эффективной программы.

Определение 14. *Программу будем называть Q -эффективной, если она выполняет Q -эффективную реализацию алгоритма.*

Определения 13 и 14 определяют одно и то же множество программ. Следовательно, понятие Q -эффективной программы можно определять как с помощью определения 13, так и с помощью определения 14.

Определение 15. *Процесс проектирования Q -эффективной программы будем называть Q -эффективным программированием.*

Q -эффективная программа полностью использует ресурс параллелизма алгоритма, так как выполняет его Q -эффективную реализацию. Следовательно, Q -эффективная программа имеет самый высокий параллелизм среди программ, реализующих алгоритм. Метод проектирования эффективных программ мы также называем методом проектирования Q -эффективных программ или методом проектирования программ для Q -эффективных реализаций алгоритмов.

Описания Q -эффективной реализации алгоритма, полученного на втором этапе метода, достаточно для разработки Q -эффективной программы для общей памяти. При разработке Q -эффективной программы для распределенной памяти необходимо учитывать, что данные должны обладать свойством локальности, в противном случае могут возникать многочисленные промахи кэша при их считывании из памяти, что снижает быстродействие. Обеспечить локальность данных позволяет распределение вычислений между вычислительными узлами ПВС, использующее описание Q -эффективной реализации алгоритма. Для распределения вычислений между вычислительными узлами мы применяем принцип MASTER-SLAVE [10]. Для исследований используется один вычислительный узел MASTER и несколько вычислительных узлов SLAVE.

Более подробно метод проектирования Q -эффективных программ изложен в работах [1, 11].

3. Применение метода проектирования эффективных программ. Метод проектирования Q -эффективных программ предложен автором статьи в 2017 году. В настоящее время метод апробирован на множестве алгоритмов, мы называем его коллекцией алгоритмов. Их обзор и результаты апробации рассматриваются в данном разделе.

Метод проектирования Q -эффективных программ был применен в ряде выпускных квалификационных работ (ВКР) студентов кафедры системного программирования Южно-Уральского государственного университета, у которых автор статьи являлась научным руководителем. В этих работах описана разработка Q -эффективных программ для алгоритмов, имеющих различные структуры Q -детерминантов, а также приведены результаты экспериментального исследования динамических характеристик разработанных программ.

Исследования проводились на суперкомпьютере «Торнадо» Южно-Уральского государственного университета [12]. При разработке программ применялся язык программирования C++. Для общей памяти использовался один вычислительный узел, который содержит два центральных процессора Intel Xeon X5680 с частотой 3.33 GHz, каждый из которых имеет 6 ядер и поддерживает 12 потоков, оперативная память узла 24 Гб ECC DDR3 Full buffered. Для разработки программ применялась технология OpenMP. Для распределенной памяти использовалось несколько вычислительных узлов. Для разработки программ применялись технологии OpenMP и MPI.

Разработанные Q -эффективные программы полностью используют ресурс параллелизма алгоритмов. Однако ресурсов вычислительной системы было недостаточно для исполь-

зования всего параллелизма при вычислительных экспериментах. В этих условиях были оценены динамические характеристики программ [13]. Ускорение программы вычислялось по формуле

$$S_p = \frac{T_1}{T_p},$$

здесь T_1 — время выполнения программы на одном вычислительном ядре, T_p — время выполнения программы на p вычислительных ядрах. Затем для вычисления эффективности программы использовалась формула

$$E_p = \frac{S_p}{p},$$

где S_p — ускорение программы, p — количество используемых вычислительных ядер.

Далее дадим обзор алгоритмов, к которым применялся метод проектирования Q -эффективных программ.

Первыми были исследованы алгоритмы умножения плотных и разреженных матриц [14]. Для хранения в памяти разреженных матриц использовался формат CSR. Q -детерминант алгоритма умножения плотных матриц состоит из безусловных Q -термов. Алгоритм умножения разреженных матриц имеет Q -детерминант, состоящий из условных Q -термов, которые учитывают доступ к входным данным в памяти.

В ВКР [15] рассматривается алгоритм для реализации метода Гаусса–Жордана решения систем линейных уравнений. Алгоритм состоит в выборе в качестве ведущего элемента на шаге i первого ненулевого элемента строки i . В связи с этим исследованием хотелось бы отметить очень интересный факт. Q -детерминант алгоритма, реализующего метод Гаусса–Жордана, состоит из условных Q -термов длины $n!$, где n — размер матрицы системы. Поэтому важно было понять, возможно ли при такой большой вычислительной сложности разработать Q -эффективную программу и провести вычислительные эксперименты. В [15] показано, что это возможно. Особенности Q -эффективной реализации, благодаря которым практическая реализация оказалась возможной, описаны в работе [1].

Первым алгоритмом с Q -детерминантом, состоящим из условных бесконечных Q -термов, к которому был применен метод проектирования Q -эффективных программ, стал алгоритм, реализующий метод Якоби решения систем линейных уравнений. Результаты исследования этого алгоритма представлены в ВКР [16].

Также было решено апробировать метод проектирования Q -эффективных программ на алгоритмах с малым и большим ресурсом параллелизма. Поэтому в ВКР [17] был рассмотрен метод прогонки решения систем линейных трехточечных уравнений, обладающий небольшим ресурсом параллелизма. Q -детерминант реализующего его алгоритма состоит из безусловных Q -термов. Исследование показало, что в данном случае не следует использовать распределенную память, поскольку это может привести к снижению производительности. Кроме того, в [17] рассмотрен метод Фурье решения систем разностных уравнений, обладающий большим ресурсом параллелизма. Q -детерминант алгоритма, реализующего этот метод, состоит из безусловных Q -термов. Он может быть эффективно реализован на распределенной памяти с применением принципа MASTER-SLAVE.

Исследование практического применения проектирования Q -эффективных программ показывает, что принцип MASTER-SLAVE непригоден для некоторых алгоритмов, так как приводит к увеличению потерь времени на обмен между вычислительными узлами. Алгоритм, реализующий метод Якоби решения систем пятиточечных разностных уравнений

[18], является примером такого алгоритма. Его Q -детерминант состоит из условных бесконечных Q -термов.

Алгоритм, реализующий метод Гаусса–Зейделя решения систем линейных уравнений, следующий, к которому применен был метод проектирования Q -эффективных программ [19]. Его Q -детерминант состоит из условных бесконечных Q -термов и является сложным, как и алгоритм, реализующий метод Гаусса–Жордана. Хотя Q -детерминанты этих алгоритмов сложные, но они хорошо структурированы. Многие другие известные алгоритмы обладают аналогичными свойствами. Хорошо структурированные Q -детерминанты облегчают создание Q -эффективных программ.

Затем метод проектирования Q -эффективных программ был применен к алгоритму, реализующему метод сопряженных градиентов решения систем линейных уравнений [20]. Его Q -детерминант состоит из условных бесконечных Q -термов. Результаты проведенных экспериментальных исследований с использованием различных реализаций алгоритма подтвердили теоретические выводы (см. раздел 4).

Алгоритм Дейкстры для вычисления кратчайших расстояний от выделенной вершины графа до всех остальных [21] является одним из самых используемых на практике алгоритмов на графах. Это первый алгоритм на графах, который был исследован с помощью метода проектирования Q -эффективных программ. Q -детерминант алгоритма Дейкстры состоит из условных Q -термов. Проведенное исследование описано в работе [22]. Оно дает возможность сделать вывод, что концепция Q -детерминанта может применяться для разработки эффективных программ, реализующих алгоритмы на графах.

Повышение эффективности алгоритмов, применяемых для решения задач искусственного интеллекта, является важной задачей. На онлайн-сервисе SCISPACE работа [5] автора статьи рекомендована в числе лучших пяти работ для ответа на вопрос: «How does parallel computing improve the efficiency of AI algorithms?» [23]. Метод проектирования Q -эффективных программ был апробирован на алгоритмах, реализующих метод стохастического градиентного спуска и метод обратного распространения ошибки, применяемых для обучения нейронных сетей, также проведена оценка динамических характеристик разработанных Q -эффективных программ. Q -детерминанты исследованных алгоритмов состоят из безусловных Q -термов. Эти исследования описаны в ВКР [24].

Представления в форме Q -детерминантов, а также результаты экспериментального исследования алгоритмов, приведенных в обзоре, можно посмотреть в работах, на которые даны ссылки.

Приведенный обзор показывает, что метод проектирования Q -эффективных программ может успешно применяться на практике.

Коллекция алгоритмов, к которым применен метод проектирования Q -эффективных программ, постоянно пополняется.

4. Результаты исследования Q -эффективных программ. Так как Q -эффективная программа выполняет Q -эффективную реализацию алгоритма, то она полностью использует ресурс параллелизма алгоритма. Поэтому динамические характеристики Q -эффективной программы зависят от ресурса параллелизма алгоритма. Определить ресурс параллелизма алгоритма можно с помощью его Q -детерминанта и программной Q -системы.

Другим подходом к вычислению границ для параллельных алгоритмов, близким к нашему подходу, является классический закон Амдала [25]. По сути, закон Амдала связан с исследованием ресурса параллелизма алгоритма, а точнее его высоты. Если Q -

эффективная реализация алгоритма имеет операции, которые необходимо выполнять последовательно, то высота алгоритма не меньше максимального уровня вложенности этих операций. В этом случае Q -эффективная программа содержит последовательные инструкции для выполнения операций, которые должны выполняться последовательно. Время выполнения Q -эффективной программы не может быть меньше, чем требуется для выполнения последовательных инструкций. Другими словами, ускорение Q -эффективной программы ограничено временем, необходимым для выполнения последовательных инструкций. Таким образом, предлагаемое нами исследование сравнимо с теоретическими границами, установленными классическим законом Амдала.

Условия разработки и выполнения параллельных программ могут быть разными. Например, при разработке можно использовать различные языки программирования, технологии параллельного программирования, архитектуры вычислительных систем и т. д. Кроме того, выполнение параллельных программ зависит от компиляторов языков программирования, от ПВС, ее вычислительных ресурсов, выделенных для параллельной программы, и т. д.

Определение 16. *Совокупность условий разработки и выполнения программы будем называть вычислительной инфраструктурой программы.*

Пример. Приведем вычислительную инфраструктуру Q -эффективных программ, описанных в разделе 3. Условиями разработки программ являются язык программирования C++, технологии параллельного программирования OpenMP и MPI, архитектуры вычислительных систем с общей и распределенной памятью, принцип MASTER-SLAVE. Условиями выполнения программ являются компилятор языка C++, суперкомпьютер «Торнадо» Южно-Уральского государственного университета, при этом для выполнения параллельной программы для общей памяти используется один вычислительный узел, а для выполнения параллельной программы для распределенной памяти задействовано несколько вычислительных узлов.

Динамические характеристики Q -эффективной программы зависят не только от ресурса параллелизма алгоритма, но и от вычислительной инфраструктуры программы, в том числе от следующих факторов.

1) Как известно, время выполнения параллельных программ увеличивается за счет передачи данных между вычислительными узлами ПВС.

2) Синхронизация вычислений также увеличивает время выполнения параллельных программ.

3) Если для выполнения Q -эффективной реализации алгоритма вычислительных ресурсов ПВС не достаточно, то динамические характеристики Q -эффективной программы, реализующей алгоритм, будут хуже, чем, если вычислительных ресурсов достаточно. Например, Q -эффективные программы, описанные в разделе 3, полностью используют ресурс параллелизма алгоритмов. Однако выделенных вычислительных ресурсов ПВС было не достаточно для выполнения Q -эффективных реализаций алгоритмов, поэтому достигнутые при вычислительных экспериментах динамические характеристики Q -эффективных программ хуже, чем могли бы быть при достаточных вычислительных ресурсах.

4) Обеспечение локальности данных при параллельных вычислениях важно, так как это приводит к увеличению производительности программ.

Теорема. Предположим, что для численного алгоритма имеются Q -эффективная и не Q -эффективная программы с одной и той же вычислительной инфраструктурой. Тогда динамические характеристики Q -эффективной программы не хуже динамических характеристик не Q -эффективной программы.

Доказательство. Для доказательства применим закон Амдала. Не Q -эффективная программа выполняет последовательно некоторые из тех операций, которые Q -эффективная программа выполняет параллельно. Поэтому не Q -эффективная программа содержит дополнительные последовательные инструкции по сравнению с Q -эффективной программой. Следовательно, время выполнения не Q -эффективной программы не меньше времени выполнения Q -эффективной программы. Тогда ускорение не Q -эффективной программы не превышает ускорение Q -эффективной программы. Таким образом, динамические характеристики Q -эффективной программы не хуже динамических характеристик не Q -эффективной программы.

Из теоремы получаем важное следствие.

Следствие. Каждая из Q -эффективных программ, реализующих один и тот же численный алгоритм, является самой эффективной для своей вычислительной инфраструктуры среди программ, реализующих тот же алгоритм.

Таким образом, метод проектирования Q -эффективных программ гарантирует, что разработанная с его помощью программа является самой эффективной для реализуемого программой алгоритма и вычислительной инфраструктуры программы.

Дальнейшим развитием расширенной модели концепции Q -детерминанта может быть добавление моделей параллельных вычислений для вычислительных систем с другими архитектурными особенностями. В настоящее время разработаны десятки моделей параллельных вычислений и создаются новые. Также для разработки Q -эффективных программ возможно использование различных языков программирования и технологий параллельного программирования. В результате для любого численного алгоритма можно разработать потенциально бесконечное множество Q -эффективных программ, которым соответствует потенциально бесконечное множество вычислительных инфраструктур. По-видимому, из всех Q -эффективных программ для данного алгоритма нет самой эффективной, так как их множество потенциально бесконечно. Однако каждая из этих программ эффективна для своей вычислительной инфраструктуры.

Следовательно, метод проектирования Q -эффективных программ имеет теоретическую и практическую значимость.

Заключение. Метод проектирования Q -эффективных программ могут применять разработчики программного обеспечения для проектирования эффективных программ, предназначенных для любых ПВС — от персональных компьютеров с многоядерными процессорами до суперкомпьютеров. Делать это несложно, учитывая, что коллекция алгоритмов, к которым применен метод проектирования Q -эффективных программ, создана студентами. Применение метода приведет к уменьшению времени выполнения программного обеспечения и повышению быстродействия вычислительных систем.

В случае, если значения динамических характеристик Q -эффективной программы не устраивают, их можно улучшить, изменив вычислительную инфраструктуру программы или заменив алгоритм алгоритмом с меньшей высотой, возможны и оба изменения одновременно.

Одним из направлений развития исследований на основе концепции Q -детерминанта в настоящее время является автоматизированное проектирование и исполнение Q -

эффективных программ. Первые системы автоматизированного проектирования, использующие разные подходы к реализации, уже разработаны. Одна из них описана в работе [26].

Список литературы

1. Aleeva V., Aleev R. Investigation and Implementation of Parallelism Resources of Numerical Algorithms // ACM Transactions on Parallel Computing. 2023. Vol. 10. N 2, Article number 8. P. 1–64. DOI: 10.1145/3583755.
2. Алеева В. Н. Анализ параллельных численных алгоритмов. Препринт № 590. Новосибирск: ВЦ СО АН СССР, 1985. 23 с.
3. Ершов Ю. Л., Палютин Е. А. Математическая логика. М.: Наука, 1987. 336 с.
4. Алеева В. Н., Зотова П. С., Склезнев Д. С. Расширение возможностей исследования ресурса параллелизма численных алгоритмов с помощью программной Q -системы // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 66–81. DOI: 10.14529/cmse210205.
5. Aleeva V. N. Improving Parallel Computing Efficiency // Proceedings — 2020 Global Smart Industry Conference, GloSIC 2020. IEEE. 2020. P. 113–120. Article number 9267828. DOI: 10.1109/GloSIC50886.2020.9267828.
6. Aleeva V. N., Aleev R. Zh. High-Performance Computing Using Application of Q -determinant of Numerical Algorithms // Proceedings — 2018 Global Smart Industry Conference, GloSIC 2018. IEEE. 2018. 8 p. Article number 8570160. DOI: 10.1109/GloSIC.2018.8570160.
7. Aleeva V., Bogatyreva E., Skleznev A., et al. Software Q -system for the Research of the Resource of Numerical Algorithms Parallelism // Supercomputing. RuSCDays 2019. Communications in Computer and Information Science. 2019. Vol. 1129. P. 641–652. DOI: 10.1007/978-3-030-36592-9_52.
8. McColl W. F. General Purpose Parallel Computing // Lectures on Parallel Computation, Cambridge International Series on Parallel Computation. USA: Cambridge University Press, 1993. P. 337–391.
9. Valiant L. G. A bridging model for parallel computation // Communications of the ACM. 1990. Vol. 33, no. 8. P. 103–111. DOI: 10.1145/79173.79181.
10. Leung J. Y.-T., Zhao H. Scheduling problems in master-slave model // Annals of Operations Research. 2008. Vol. 159. P. 215–231. DOI: 10.1007/s10479-007-0271-4.
11. Aleeva V. Designing a Parallel Programs on the Base of the Conception of Q -Determinant // Supercomputing. RuSCDays 2018. Communications in Computer and Information Science. 2019. Vol. 965. P. 565–577. DOI: 10.1007/978-3-030-05807-4_48.
12. Суперкомпьютер «Торнадо ЮУрГУ». [Электрон. Рес.]: <http://supercomputer.susu.ru/computers/tornado/>. Дата обращения: 17.09.2024.
13. Открытая энциклопедия свойств алгоритмов. [Электрон. Рес.]: <https://algowiki-project.org/ru>. Дата обращения: 17.09.2024.
14. Валькевич Н. В. Q -эффективная реализация алгоритма умножения матриц на суперкомпьютере «Торнадо ЮУрГУ» // Вып. квалиф. работа бакалавра по направлению «Фундаментальная информатика и информационные технологии»: 02.03.02 / Южно-Уральский государственный университет. Челябинск. 2017. 33 л. [Электрон. Рес.]: <http://omega.sp.susu.ru/publications/bachelorthesis/17-Valkevich.pdf>. Дата обращения: 17.09.2024.
15. Тарасов Д. Е. Q -эффективный кодизайн реализации метода Гаусса–Жордана на суперкомпьютере «Торнадо ЮУрГУ» // Вып. квалиф. работа магистра по направлению «Фундаментальная информатика и информационные технологии»: 02.04.02 / Южно-Уральский государственный

университет. Челябинск. 2017. 41 л. [Электрон. Рес.]: <http://omega.sp.susu.ru/publications/masterthesis/17-Tarasov.pdf>. Дата обращения: 17.09.2024.

16. Лаптева Ю. С. Q -эффективная реализация метода Якоби для решения СЛАУ на суперкомпьютере «Торнадо ЮУрГУ» // Вып. квалиф. работа бакалавра по направлению «Фундаментальная информатика и информационные технологии»: 02.03.02 / Южно-Уральский государственный университет. Челябинск. 2017. 30 л. [Электрон. Рес.]: <http://omega.sp.susu.ru/publications/bachelorthesis/17-Lapteva.pdf>. Дата обращения: 20.09.2024.

17. Баженова Л. А. Применение метода проектирования Q -эффективной программы для решения системы сеточных уравнений // Вып. квалиф. работа бакалавра по направлению «Фундаментальная информатика и информационные технологии»: 02.03.02 / Южно-Уральский государственный университет. Челябинск. 2018. 30 л. [Электрон. Рес.]: <http://omega.sp.susu.ru/publications/bachelorthesis/18-Bazhenova.pdf>. Дата обращения: 20.09.2024.

18. Кондакова А. С. Разработка Q -эффективной программы для решения пятиточечных разностных уравнений методом простой итерации и исследование ее динамических характеристик // Вып. квалиф. работа магистра по направлению «Фундаментальная информатика и информационные технологии»: 02.04.02 / Южно-Уральский государственный университет. Челябинск. 2019. 40 л. [Электрон. Рес.]: http://omega.sp.susu.ru/publications/masterthesis/2019_220_kondakovaas.pdf. Дата обращения: 20.09.2024.

19. Нечепоренко А. Д. Разработка Q -эффективной программы для решения СЛАУ методом Гаусса–Зейделя // Вып. квалиф. работа бакалавра по направлению «Фундаментальная информатика и информационные технологии»: 02.03.02 / Южно-Уральский государственный университет. Челябинск. 2018. 32 л. [Электрон. Рес.]: <http://omega.sp.susu.ru/publications/bachelorthesis/18-Necheporenko.pdf>. Дата обращения: 20.09.2024.

20. Алеева В. Н., Шатов М. Б. Применение концепции Q -детерминанта для эффективной реализации численных алгоритмов на примере метода сопряженных градиентов для решения систем линейных уравнений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 3. С. 56–71. DOI: 10.14529/cmse210304.

21. Dijkstra E. W. A note on two problems in connexion with graphs // Numerische Mathematik. 1959. Vol. 1. P. 269–271. DOI: 10.1007/BF01386390.

22. Алеева В. Н., Манатин П. А. Применение метода проектирования Q -эффективных программ для алгоритма Дейкстры. // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2023. Т. 12, № 2. С. 62–77. DOI: 10.14529/cmse230203.

23. SCISPACE. Ответ на вопрос «How does parallel computing improve the efficiency of AI algorithms?». [Электрон. Рес.]: <https://typeset.io/questions/how-does-parallel-computing-improve-the-efficiency-of-ai-5gkdumloct>. Дата обращения: 23.09.2024.

24. Сапожников А. С. Применение метода проектирования Q -эффективных программ к методу стохастического градиентного спуска для обучения нейронных сетей // Вып. квалиф. работа магистра по направлению «Программная инженерия»: 09.04.04 / Южно-Уральский государственный университет. Челябинск. 2024. 43 л. [Электрон. Рес.]: https://sp.susu.ru/student/masterthesis/2024_229_sapoznikovas.pdf. Дата обращения: 23.09.2024.

25. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.

26. Алеева В. Н. Автоматизированное проектирование и исполнение эффективных программ для численных алгоритмов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2023. Т. 12, № 3. С. 31–49. DOI: 10.14529/cmse230303.



Алеева Валентина Николаевна. E-mail: aleevavn@susu.ru; тел.: +7-351-267-90-89.

В 1972 году окончила Новосибирский государственный университет с присвоением квалификации «Математика, прикладная математика». В 1986 году решением Совета при Вычислительном центре СО АН СССР ей присуждена ученая степень кандидата физико-математических наук. В настоящее время она является доцентом кафедры системного программирования Южно-Уральского государственного университета (Национальный исследовательский университет).

Имеет 2 изобретения по архитектуре параллельных вычислительных систем и более 40 публикаций по распараллеливанию численных алгоритмов. Научные интересы: архитектура вычислительных систем, ресурсы параллелизма алгоритмов, проектирование эффектив-

ных программ, эффективность параллельных вычислений, в том числе в автоматизированном режиме.

Valentina Nikolaevna Aleeva. E-mail: aleevavn@susu.ru; tel.: +7-351-267-90-89.

Graduated from Novosibirsk State University with the Master of Mathematics and Applied Mathematics degree in 1972 and defended his candidate dissertation in physical and mathematical sciences in Computing Centre of SB RAS in 1986. Currently, she is an Associate Professor of the Department of System Programming at South Ural State University (National Research University).

She has 2 inventions on the architecture of parallel computing systems and more than 40 publications on parallelization of numerical algorithms. The scientific interests include the architecture of computing systems, parallelism resources of algorithms, the design of effective programs, the efficiency of parallel computing, particularly in automated mode.

Дата поступления — 15.10.2024