ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)

ACADEMIC SCIENTIFIC JOURNAL OF COMPUTER SCIENCE

№2
2026

ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)

CENTRAL ASIAN ACADEMIC
RESEARCH CENTER

ACADEMIC SCIENTIFIC JOURNAL OF COMPUTER SCIENCE

2 (358)

APRIL – JUNE 2026

PUBLISHED SINCE JANUARY 1963
PUBLISHED 4 TIMES A YEAR

ALMATY, NAS RK

Academic Scientific Journal of Computer Science
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)
Owner: «Central Asian Academic Research Center» LLP (Almaty).
Certificate **№ KZ77VPY00121154** on the re-registration of the periodical printed and online publication of the information agency, issued on **05.06.2025** by the Republican State Institution «Information Committee» of the Ministry of Culture and Information of the Republic of Kazakhstan
Subject area*: information and communication technologies.*
Currently: *included in the list of journals recommended by the CCSES MSHE RK in the direction of «Information and communication technologies».*
Periodicity: *4 times a year.*
http://www.physico-mathematical.kz/index.php/en/

Academic Scientific Journal of Computer Science
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)
Меншіктеуші: «Орталық Азия академиялық ғылыми орталығы» ЖШС (Алматы).
Ақпарат агенттігінің мерзімді баспасөз басылымын, ақпарат агенттігін және желілік басылымды қайта есепке қою туралы ҚР Мәдениет және Ақпарат министрлігі «Ақпарат комитеті» Республикалық мемлекеттік мекемесі **05.06.2025** ж. берген **№ KZ77VPY00121154** Куәлік.
Тақырыптық бағыты: *ақпараттық-коммуникациялық технологиялар*
Қазіргі уақытта: *«ақпараттық-коммуникациялық технологиялар» бағыты бойынша ҚР БҒМ БҒСБК ұсынған журналдар тізіміне енді.*
Мерзімділігі: *жылына 4 рет.*
http://www.physico-mathematical.kz/index.php/en/

Academic Scientific Journal of Computer Science
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)
Собственник: *ТОО «Центрально-азиатский академический научный центр» (г. Алматы).*
Свидетельство о постановке на переучет периодического печатного издания, информационного агентства и сетевого издания **№ KZ77VPY00121154.** Дата выдачи **05.06.2025**
Тематическая направленность: *информационно-коммуникационные технологии.*
В настоящее время: *вошел в список журналов, рекомендованных КОКСНВО МНВО РК по направлению «информационно-коммуникационные технологии».*
Периодичность: *4 раза в год.*
http://www.physico-mathematical.kz/index.php/en/

CONTENTS

COMPUTER SCIENCE

МАЗМҰНЫ

КОМПЬЮТЕРЛІК ҒЫЛЫМДАР

СОДЕРЖАНИЕ

КОМПЬЮТЕРНЫЕ НАУКИ

ACADEMIC SCIENTIFIC JOURNAL OF COMPUTER SCIENCE
ISSN 1991-346X
Volume 2.
Namber 358 (2026). 335–346

https://doi.org/10.32014/2026.2518-1726.442

IRSTI 50.37.15
UDC 004.272:004.42

© Sarsembayev M, Urmashev B. *, 2026.
Al-Farabi Kazakh National University, Almaty, Kazakhstan.
E-mail: baidaulet.urmashev@kaznu.kz

OPTIMIZATION OF THE CALCULATION OF KINETIC EQUATIONS OF COMBUSTION PROCESSES ON GPU USING GLOBAL MEMORY AND SHARED MEMORY

Sarsembayev Magzhan — researcher, Department of Computer Science, Al-Farabi Kazakh National University, Almaty, Kazakhstan,
E-mail: magjan@kaznu.kz, https://orcid.org/0000-0003-2139-2456;
Urmashev Baydaulet — Candidate of Physical and Mathematical Sciences, Professor, Department of Computer Science, Al-Farabi Kazakh National University, Almaty, Kazakhstan,
E-mail: baidaulet.urmashev@kaznu.kz, https://orcid.org/ 0000-0002-0085-1708.

Abstract. This paper considers the problem of improving computational performance when solving kinetic equations of combustion processes on graphics processing units. Detailed chemical kinetic mechanisms used for modeling the combustion of real fuels may include hundreds of components and thousands of elementary reactions, which leads to a significant increase in computation time when traditional CPU-based methods are used. To accelerate computations, a CUDA-oriented approach is proposed, based on distributing reactions among GPU threads and optimizing access to GPU memory. The paper compares three computational variants: sequential calculation on the CPU, parallel calculation on the GPU using global memory, and parallel calculation on the GPU using shared memory. The Aramco_Mech mechanism is used as a test mechanism. Numerical experiments were carried out for mechanisms of different sizes: from 33 components and 118 reactions to 493 components and 2716 reactions. The calculations were performed on a system with an AMD Ryzen 9 5950X 16-Core Processor, 128 GB of RAM, and an NVIDIA GeForce RTX 4080 Ti graphics adapter. The results show that as the number of reactions increases, the advantage of GPU-based computations becomes more pronounced. For the mechanism containing 493 components and 2716 reactions, the CPU computation time was 1 day 4 hours 2 minutes 4 seconds 796 milliseconds, while the GPU computation time using global memory was 21 minutes 10 seconds 591 milliseconds, and using shared memory it was 10 minutes

51 seconds 585 milliseconds. The speedup was 79.43 times for global memory and 154.89 times for shared memory. The obtained results confirm the efficiency of GPU architecture and the importance of optimal memory organization when solving large systems of chemical kinetics.

Keywords: chemical kinetics, combustion, GPU, CUDA, shared memory, global memory, Runge–Kutta, Aramco_Mech, parallel computing, kinetic solver

For citations: Sarsembayev M, Urmashev B. Optimization of the calculation of kinetic equations of combustion processes on GPU using global memory and shared memory. Academic Scientific Journal of Computer Science, 2026. — No.2. – P. 335-346. DOI https://doi.org/10.32014/2026.2518-1726.442

Әл-Фараби атындағы Қазақ Ұлттық университеті, Алматы, Қазақстан.
E-mail: magjan@kaznu.kz, baidaulet.urmashev@kaznu.kz

GLOBAL MEMORY ЖӘНЕ SHARED MEMORY ҚОЛДАНУ АРҚЫЛЫ GPU-ДА ЖАНУ ПРОЦЕСТЕРІНІҢ КИНЕТИКАЛЫҚ ТЕҢДЕУЛЕРІН ЕСЕПТЕУДІ ОҢТАЙЛАНДЫРУ

Сарсембаев Магжан — зерттеуші, Компьютерлік ғылымдар кафедрасы, әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан,
E-mail: magjan@kaznu.kz, ORCID: https://orcid.org/0000-0003-2139-2456;
Урмашев Байдаулет — физика-математика ғылымдарының кандидаты, профессор, Компьютерлік ғылымдар кафедрасы, әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан,
E-mail: baidaulet.urmashev@kaznu.kz, ORCID: https://orcid.org/0000-0002-0085-1708.

Аннотация. Бұл мақалада графикалық процессорларда жану процестерінің кинетикалық теңдеулерін шешу кезінде есептеу өнімділігін арттыру мәселесі қарастырылады. Нақты отындардың жануын модельдеу үшін қолданылатын егжей-тегжейлі химиялық кинетикалық механизмдер жүздеген компоненттер мен мыңдаған элементар реакцияларды қамтуы мүмкін. Бұл дәстүрлі CPU негізіндегі әдістерді қолданған кезде есептеу уақытының едәуір артуына алып келеді. Есептеулерді жеделдету үшін реакцияларды GPU ағындары арасында үлестіруге және GPU жадысына қатынауды оңтайландыруға негізделген CUDA-бағытталған тәсіл ұсынылады. Мақалада есептеудің үш нұсқасы салыстырылады: CPU-да тізбекті есептеу, global memory қолдану арқылы GPU-да параллель есептеу және shared memory қолдану арқылы GPU-да параллель есептеу. Сынақ механизмі ретінде Aramco_Mech механизмі қолданылды. Сандық тәжірибелер әртүрлі өлшемдегі механизмдер үшін жүргізілді: 33 компонент пен 118 реакциядан бастап 493 компонент пен 2716 реакцияға дейін. Есептеулер AMD Ryzen 9 5950X 16-Core Processor процессоры, 128 ГБ жедел жады және NVIDIA GeForce RTX

4080 Ti графикалық адаптері бар есептеу жүйесінде орындалды. Алынған нәтижелер реакциялар саны артқан сайын GPU негізіндегі есептеулердің артықшылығы айқынырақ байқалатынын көрсетті. 493 компоненттен және 2716 реакциядан тұратын механизм үшін CPU есептеу уақыты 1 тәулік 4 сағ 2 мин 4 с 796 мс болды. Ал global memory қолданылған GPU есептеу уақыты 21 мин 10 с 591 мс, shared memory қолданылған жағдайда 10 мин 51 с 585 мс құрады. Жылдамдату коэффициенті global memory үшін 79,43 есе, ал shared memory үшін 154,89 есе болды. Алынған нәтижелер үлкен өлшемді химиялық кинетика жүйелерін шешу кезінде GPU архитектурасының тиімділігін және жадыны оңтайлы ұйымдастырудың маңыздылығын растайды.

Түйін сөздер: химиялық кинетика, жану, GPU, CUDA, shared memory, global memory, Рунге–Кутта, Aramco_Mech, параллель есептеу, кинетикалық солвер

© Сарсембаев М., Урмашев Б.*, 2026.
Казахский национальный университет имени аль-Фараби,
Алматы, Казахстан.
E-mail: magjan@kaznu.kz, baidaulet.urmashev@kaznu.kz

ОПТИМИЗАЦИЯ РАСЧЕТА КИНЕТИЧЕСКИХ УРАВНЕНИЙ ПРОЦЕССОВ ГОРЕНИЯ НА GPU С ИСПОЛЬЗОВАНИЕМ GLOBAL MEMORY И SHARED MEMORY

Сарсембаев Магжан — исследователь кафедры компьютерных наук, Казахский национальный университет имени аль-Фараби, Алматы, Казахстан,
E-mail: magjan@kaznu.kz, https://orcid.org/0000-0003-2139-2456;
Урмашев Байдаулет — кандидат физико-математических наук, профессор кафедры компьютерных наук, Казахский национальный университет имени аль-Фараби, Алматы, Казахстан,
E-mail: baidaulet.urmashev@kaznu.kz, https://orcid.org/0000-0002-0085-1708.

Аннотация. *Актуальность.* В статье рассматривается задача повышения вычислительной производительности при решении кинетических уравнений процессов горения на графических процессорах. Детальные химико-кинетические механизмы, применяемые для моделирования горения реальных топлив, могут включать сотни компонентов и тысячи элементарных реакций, что существенно увеличивает время расчета при использовании традиционных CPU-методов. В связи с этим актуальным является применение GPU-вычислений и оптимизация доступа к памяти графического процессора. *Цель.* Разработать и оценить CUDA-ориентированный подход к ускорению расчета кинетических уравнений процессов горения на GPU с использованием global memory и shared memory. *Методы.* Для ускорения вычислений предложен подход, основанный на распределении реакций между потоками GPU и оптимизации доступа к памяти графического процессора.

В работе сравниваются три варианта вычислений: последовательный расчет на CPU, параллельный расчет на GPU с использованием global memory и параллельный расчет на GPU с использованием shared memory. В качестве тестового механизма использован Aramco_Mech. Численные эксперименты проведены для механизмов различной размерности: от 33 компонентов и 118 реакций до 493 компонентов и 2716 реакций. Расчеты выполнялись на вычислительной системе с процессором AMD Ryzen 9 5950X 16-Core Processor, 128 ГБ оперативной памяти и графическим адаптером NVIDIA GeForce RTX 4080 Ti. *Результаты и выводы.* Полученные результаты показывают, что с увеличением числа реакций преимущество GPU-вычислений становится более выраженным. Для механизма, содержащего 493 компонента и 2716 реакций, время расчета на CPU составило 1 сутки 4 часа 2 минуты 4 секунды 796 миллисекунд, тогда как время расчета на GPU с использованием global memory составило 21 минуту 10 секунд 591 миллисекунду, а с использованием shared memory - 10 минут 51 секунду 585 миллисекунд. Ускорение составило 79,43 раза для global memory и 154,89 раза для shared memory. Результаты подтверждают эффективность GPU-архитектуры и важность оптимальной организации памяти при решении больших систем химической кинетики. Практическая значимость исследования заключается в возможности применения предложенного подхода для ускорения кинетических солверов, моделирования процессов горения и выполнения ресурсоемких вычислений в задачах химической кинетики.

Ключевые слова: химическая кинетика, горение, GPU, CUDA, shared memory, global memory, Рунге - Кутта, Aramco_Mech, параллельные вычисления, кинетический солвер

Introduction. Modeling combustion processes is one of the important problems of modern computational science, energy engineering, ecology, and industrial safety. An accurate description of combustion processes is necessary for the design of internal combustion engines, gas turbine units, thermal power plants, industrial combustion chambers, as well as for the study of alternative fuels and the assessment of harmful emissions. Modern software packages for chemical kinetics and reacting flows, such as Cantera, ReactionMechanismSimulator.jl, KinetiX, PeleC, and PeleMP, demonstrate that a detailed description of chemical processes requires advanced numerical methods and high-performance computing resources (Goodwin et al., 2023; Johnson et al., 2024; Danciu and Frouzakis, 2025; de Frahan et al., 2023; Owen et al., 2024).

The combustion process is a complex combination of physicochemical phenomena, including chemical reactions, heat and mass transfer, gas-dynamic effects, interaction of multicomponent mixtures, and formation of reaction products. Chemical kinetic mechanisms are used to describe such processes, and they may include tens, hundreds, or even thousands of elementary reactions. Each

reaction contributes to the change in the concentrations of chemical components, which leads to the formation of a large system of ordinary differential equations.

Traditional CPU-based methods for calculating chemical kinetics have limited performance when working with large mechanisms. As the number of components and reactions increases, the number of operations related to calculating forward and reverse reaction rates, concentration power terms, rate constants, and summing reaction contributions to the change in component concentrations also increases. As a result, the calculation of large mechanisms on the CPU may take hours or even days, which complicates serial computational experiments and the practical use of such models in engineering problems.

Literature reviw. One effective way to solve this problem is to use graphics processing units. A GPU has a large number of computing cores and allows many similar operations to be performed in parallel. In chemical kinetics problems, a significant part of the computations for individual reactions can be parallelized, since the contribution of each reaction can be calculated independently of other reactions. Modern studies on GPU acceleration of chemical kinetics and reacting flows confirm the promise of this approach for combustion CFD and high-performance scientific computing (de Frahan et al., 2023; Cai et al., 2026; Moure et al., 2025; Rao et al., 2024; Wang et al., 2026).

However, the efficiency of GPU computations depends not only on the number of parallel threads, but also on proper memory organization. CUDA architecture includes several memory levels, such as global memory, shared memory, and registers (Nickolls et al., 2008; Hwu et al., 2022). Global memory has a large capacity but relatively high access latency. Shared memory has a smaller capacity but provides fast access within a block of threads. Therefore, transferring frequently used data from global memory to shared memory can significantly improve computational performance.

The aim of this paper is to investigate the influence of GPU memory organization on the computation time of kinetic equations of combustion processes and to compare the performance of CPU computation, GPU computation using global memory, and GPU computation using shared memory.

Problem Statement.

A chemical kinetic mechanism consisting of a set of components and elementary reactions is considered. For each component, it is necessary to determine the change in concentration over time, taking into account the contributions of all reactions in which the component participates. In general form, the system of kinetic equations can be written as:

$$\frac{dC_k}{dt} = \sum_{i=1}^{N_r} \nu_{ik}\, q_i \qquad (1)$$

where C_k is the concentration of the *k*-th component, N_r is the number of reactions, v_{ik} is the stoichiometric coefficient of component *k* in reaction *i*, and q_i is the rate of the *i*-th reaction.

The reaction rate depends on the concentrations of reactants and products, as well as on the rate constants of the forward and reverse reactions. As the number of reactions and components increases, the number of computational operations also increases, because for each reaction it is necessary to calculate products of concentrations with corresponding powers, determine the reaction contribution, and then sum this contribution for the corresponding components.

The fifth-order Runge–Kutta method is used for numerical integration of the system of kinetic equations. At each time step, intermediate coefficients k_1, k_2, k_3, k_4, k_5 are computed, after which the concentration values are updated. This approach provides high accuracy of the numerical solution; however, it requires repeated evaluation of the right-hand side of the system of equations. Therefore, accelerating the computation of the right-hand side is a key task in the development of a high-performance kinetic solver.

In this work, the problem is reduced to accelerating the computation of the right-hand side of the kinetic equation system by parallel distribution of reactions among GPU threads and optimization of memory access.

Computational Methodology.

The Aramco_Mech mechanism was used as the chemical kinetic mechanism. To evaluate performance, mechanisms of different sizes were considered. The number of components varied from 33 to 493, and the number of reactions varied from 118 to 2716.

The computational experiments were carried out with the following parameters:

- mechanism: Aramco Mech;
- time step: $dt = 1.0 \times 10^{-19}$ s
- total calculation time: $1000 \cdot dt = 1.0 \times 10^{-16}$ s

numerical method: fifth-order Runge–Kutta method.

The technical specifications of the computing system were as follows:

- CPU: AMD Ryzen 9 5950X 16-Core Processor;
- RAM: 128 GB;
- GPU: NVIDIA GeForce RTX 4080 Ti.

To evaluate performance, three computational variants were implemented and compared:

- sequential computation on the CPU
- parallel computation on the GPU using global memory
- parallel computation on the GPU using shared memory.

In the CPU version, the computations are performed sequentially. In the GPU version, each reaction is assigned to CUDA threads. The reaction index is determined using the block index and thread index:

$$i = blockIdx.x \cdot blockDim.x + threadIdx.x. \quad (2)$$

If the index i is smaller than the total number of reactions, the corresponding thread computes the contribution of this reaction. At the same time, it is necessary to correctly sum the contributions of different reactions to the change in concentration of the same component. Atomic operations and reduction procedures are used for this purpose. The use of atomic operations, including atomicAdd, is a standard way to ensure correct data accumulation when multiple threads write simultaneously (Nickolls et al., 2008; Hwu et al., 2022).

In the variant using global memory, data are stored and read directly from the GPU global memory. This approach is simple in terms of implementation, but a large number of memory accesses can lead to significant delays.

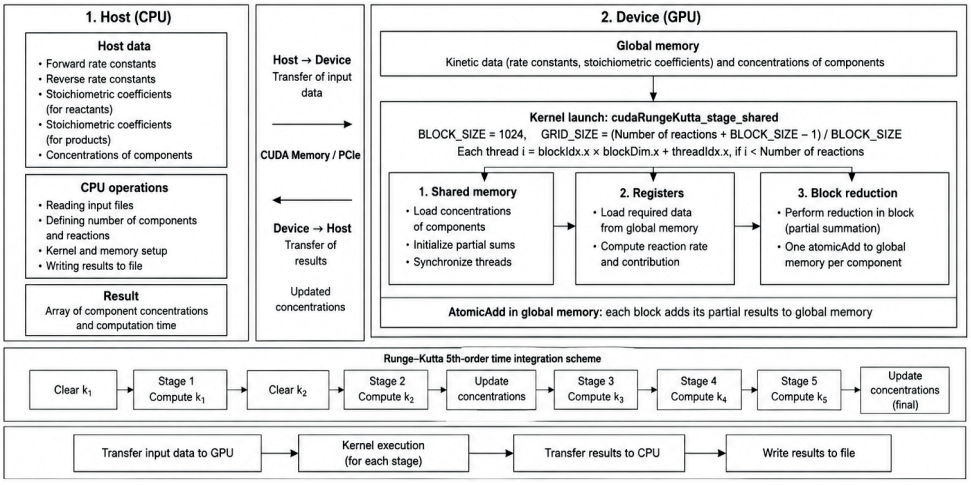

Figure 1 – Architecture of Shared Memory Usage in a CUDA-Based Kinetic Solver

Figure 1 illustrates the architecture of shared memory usage in a CUDA-based kinetic solver. The scheme shows the interaction between the host side, where input kinetic data and control operations are prepared, and the device side, where the main parallel computations are performed on the GPU. Input data, including reaction rate constants, stoichiometric coefficients, and component concentrations, are transferred from CPU memory to GPU global memory. During kernel execution, each CUDA thread processes the contribution of an individual reaction, while frequently used concentration data are loaded into shared memory to reduce global memory access latency.

The figure also shows the role of registers for storing temporary thread-local variables and the use of block reduction to accumulate partial reaction contributions inside a CUDA block. After block-level summation, the final contribution is added to global memory using atomic operations. This organization reduces the number of global memory atomic operations and improves the overall performance of the kinetic solver. The lower part of the diagram presents the fifth-order Runge–Kutta time integration sequence and the general workflow: transferring input data to

GPU memory, launching the kernel, transferring results back to CPU memory, and writing the results to a file.

In the variant using shared memory, frequently used data, in particular component concentrations, are first loaded into the fast memory of a thread block. Shared memory reduces the number of accesses to global memory and accelerates computations. In addition, partial summation of reaction contributions can be performed inside the block, after which the final value is added to global memory with fewer atomic operations.

Results. Table 1 presents the results of comparing computation time on CPU and GPU for mechanisms of different sizes.

Table 1 — Change in computation time depending on mechanism size

Number of components	Number of reactions	CPU time	GPU time, global memory	GPU time, shared memory	Global memory speedup	Shared memory speedup
33	118	24 s 426 ms	8 s 784 ms	7 s 027 ms	2.78	3.48
39	191	54 s 834 ms	12 s 617 ms	9 s 705 ms	4.35	5.65
52	245	2 min 4 s 418 ms	21 s 629 ms	16 s 021 ms	5.75	7.77
61	301	3 min 30 s 468 ms	29 s 373 ms	20 s 981 ms	7.17	10.03
66	376	5 min 7 s 382 ms	36 s 169 ms	24 s 944 ms	8.50	12.32
84	448	9 min 51 s 524 ms	54 s 495 ms	36 s 330 ms	10.85	16.28
103	565	18 min 42 s 217 ms	1 min 20 s 946 ms	50 s 591 ms	13.86	22.18
162	1006	1 h 11 min 10 s 032 ms	2 min 49 s 740 ms	1 min 36 s 994 ms	25.16	44.02
493	2716	1 day 4 h 2 min 4 s 796 ms	21 min 10 s 591 ms	10 min 51 s 585 ms	79.43	154.89

The results show that for a small number of reactions, the GPU speedup compared with the CPU is moderate. For example, for the mechanism with 33 components and 118 reactions, the speedup is 2.78 times when global memory is used and 3.48 times when shared memory is used. This can be explained by the fact that for a small computation volume, overhead costs related to data transfer between CPU and GPU, kernel function launches, and thread synchronization have a noticeable effect on total computation time.

As the mechanism size increases, GPU efficiency also increases. For 301 reactions, the speedup is 7.17 times when using global memory and 10.03 times when using shared memory. For 565 reactions, the speedup reaches 13.86 and 22.18 times, respectively. This indicates that as the number of reactions increases, the GPU begins to be used more efficiently, since the volume of parallel operations increases.

A particularly significant increase in performance is observed for large mechanisms. For the mechanism with 162 components and 1006 reactions, the

CPU computation time was 1 hour 11 minutes 10 seconds 032 milliseconds. When global memory was used, the GPU computation time decreased to 2 minutes 49 seconds 740 milliseconds, and when shared memory was used, it decreased to 1 minute 36 seconds 994 milliseconds. The speedup was 25.16 times for global memory and 44.02 times for shared memory.

The most representative result was obtained for the mechanism containing 493 components and 2716 reactions. In this case, the CPU calculation took 1 day 4 hours 2 minutes 4 seconds 796 milliseconds. When using the GPU with global memory, the time decreased to 21 minutes 10 seconds 591 milliseconds, and when using shared memory, it decreased to 10 minutes 51 seconds 585 milliseconds. Thus, the speedup was 79.43 times and 154.89 times, respectively.

Discussion. The obtained results confirm that the use of GPU is an effective approach for solving large systems of chemical kinetics. As the number of reactions increases, the computational load grows, and the parallel GPU architecture makes it possible to distribute this load among a large number of threads. This trend is consistent with modern studies in the field of GPU acceleration of chemical kinetics and reacting flows (Barwey et al., 2021; Uranakara et al., 2023; Ghioldi and Piscaglia, 2024; Mao et al., 2024; Rao et al., 2024; Cai et al., 2026).

One of the important factors affecting performance is memory organization. In the global memory variant, each thread accesses data located in GPU global memory. Although this memory has a large capacity, access to it has relatively high latency. Therefore, when there are many threads and frequent accesses to the same data, additional time costs arise.

Using shared memory reduces these costs. Since shared memory is fast memory available to threads within a single block, its use is especially effective when the same data are used repeatedly. In chemical kinetics problems, component concentrations are used in the calculation of many reaction rates. Therefore, loading concentrations into shared memory accelerates data access and reduces the load on global memory.

In addition, when calculating concentration changes, it is necessary to sum the contributions of a large number of reactions. If each reaction directly performs atomic addition in global memory, this may lead to significant delays. The use of block reduction allows partial summation to be performed first inside a block of threads, followed by a smaller number of atomic operations. This approach also contributes to improved performance.

Comparison of the two GPU variants shows that shared memory consistently provides a shorter computation time than global memory for all considered mechanisms. Moreover, the difference between them increases as the mechanism size grows. For example, for 118 reactions, the shared memory speedup relative to the CPU is 3.48 times, while for 2716 reactions it reaches 154.89 times. This confirms that memory optimization is especially important for large chemical kinetic systems.

It should also be noted that the efficiency of the GPU algorithm depends on

the structure of the mechanism itself. The greater the number of reactions and the greater the number of independent operations that can be performed in parallel, the higher the potential speedup. In the case of Aramco_Mech, a significant part of the computations can be distributed among threads, which makes this mechanism suitable for GPU acceleration.

Conclusion. This paper considered an approach to accelerating the calculation of kinetic equations of combustion processes using GPU and CUDA technology, following modern principles of high-performance computing, CUDA memory hierarchy, and parallel implementation of chemical kinetics solvers (Dagum and Menon, 1998; Nickolls et al., 2008; Stone and Davis, 2013; Niemeyer et al., 2017; Zhang et al., 2019; Sarsembayev and Urmashev, 2019; Hwu et al., 2022). Based on the Aramco_Mech mechanism, computation times were compared for CPU, GPU using global memory, and GPU using shared memory.

The obtained results lead to the following conclusions.

First, the use of GPU significantly reduces the computation time of chemical kinetic mechanisms compared with CPU computation. At the same time, the advantage of GPU increases as the number of components and reactions grows.

Second, GPU memory organization plays a key role in improving performance. Shared memory provides faster access to frequently used data and reduces the number of accesses to global memory.

Third, the use of shared memory, atomicAdd, and block reduction makes it possible to efficiently organize the accumulation of reaction contributions to component concentration changes and to increase the overall performance of the algorithm.

Fourth, for the mechanism containing 493 components and 2716 reactions, the proposed GPU algorithm using shared memory achieved a 154.89-fold speedup compared with the CPU calculation. This confirms the feasibility of using GPU architecture to solve large systems of chemical kinetics.

Thus, the proposed approach can be used in modeling combustion processes, studying alternative fuels, optimizing energy systems, and performing high-performance scientific computations. Further development of this work may be related to expanding the set of tested mechanisms, applying more complex integration schemes, optimizing thread distribution, and adapting the algorithm for multi-GPU systems.

References

Balos C.J., Day M., Esclapez L., Felden A.M., Gardner D.J., Hassanaly M., Reynolds D.R., Rood J., Sexton J.M., Wimer N.T., & Woodward C.S. (2025) SUNDIALS time integrators for exascale applications with many independent systems of ordinary differential equations. *The International Journal of High Performance Computing Applications, 39*(1). — P. 123–146. https://doi.org/10.1177/10943420241280060 (In Eng.)

Barwey S., Raman V., & Im H.G. (2021) A neural network-inspired matrix formulation of chemical kinetics for GPU acceleration. *Energies, 14*(9), 2710. https://doi.org/10.3390/en14092710 (In Eng.)

Cai Y., Ban Y., Dang G., Gong T., Wang H., Zhang F., & Chen Z.X. (2026) A GPU-accelerated stiff ODE solver for combustion CFD with fully analytical chemical kinetics and heterogeneous CPU/GPU integration. *Fuel, 408*, 137627. https://doi.org/10.1016/j.fuel.2025.137627 (In Eng.)

Dagum L., & Menon R. (1998). OpenMP: An industry-standard API for shared-memory programming. *IEEE Computational Science & Engineering, 5*(1). — P. 46–55. https://doi.org/10.1109/99.660313 (In Eng.)

Danciu B.A., & Frouzakis C.E. (2025) KinetiX: A performance portable code generator for chemical kinetics and transport properties. *Computer Physics Communications*. (In Eng.)

de Frahan M.T.H., Rood J.S., Day M.S., et al. (2023) PeleC: An adaptive mesh refinement solver for compressible reacting flows. *The International Journal of High Performance Computing Applications, 37*(2). — P. 115–131. https://doi.org/10.1177/10943420221121151 (In Eng.)

Ghioldi F., & Piscaglia F. (2024) A hybrid CPU–GPU paradigm to accelerate reactive CFD simulations. *International Journal for Numerical Methods in Fluids, 96*(8). — P. 1461–1486. https://doi.org/10.1002/fld.5297 (In Eng.)

Goodwin D.G., Moffat H.K., Schoegl I., Speth R.L., & Weber B.W. (2023) *Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. Version 3.0.0*. Zenodo. https://doi.org/10.5281/zenodo.8137090 (In Eng.)

Hwu W.-m. W., Kirk D.B., & El Hajj I. (2022) *Programming Massively Parallel Processors: A Hands-on Approach* (4th ed.). Morgan Kaufmann. https://doi.org/10.1016/C2020-0-02969-5 (In Eng.)

Johnson M.S., Pang H.-W., Payne A.M., & Green W.H. (2024)ReactionMechanismSimulator.jl: A modern approach to chemical kinetic mechanism simulation and analysis. *International Journal of Chemical Kinetics*. https://doi.org/10.1002/kin.21753 (In Eng.)

Kee R.J., Rupley F.M., & Miller J.A. (1989) *CHEMKIN-II: A Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics*. Sandia National Laboratories. https://doi.org/10.2172/5681118 (In Eng.)

Mao R., et al. (2024) An integrated framework for accelerating reactive flow simulations using machine learning and GPU computing. *Computer Physics Communications*. (In Eng.)

Moure Á., Surapaneni A., & Mira D. (2025) Optimized workload distribution for GPU-accelerated combustion simulations in heterogeneous CPU–GPU architectures. *Computers & Fluids, 302*, 106846. https://doi.org/10.1016/j.compfluid.2025.106846 (In Eng.)

Nickolls J., Buck I., Garland M., & Skadron K. (2008) Scalable parallel programming with CUDA. *ACM Queue, 6*(2), 40–53. https://doi.org/10.1145/1365490.1365500 (In Eng.)

Niemeyer K.E., Curtis N.J., & Sung C.-J. (2017) pyJac: Analytical Jacobian generator for chemical kinetics. *Computer Physics Communications, 215*. — P. 188–203. https://doi.org/10.1016/j.cpc.2017.02.004 (In Eng.)

Owen L.D., Ge W., Rieth M., Arienti M., Esclapez L., Soriano B., Mueller M.E., Day M., Sankaran R., & Chen J.H. (2024) PeleMP: The multiphysics solver for the combustion Pele adaptive mesh refinement code suite. *Journal of Fluids Engineering, 146*(4), 041103. https://doi.org/10.1115/1.4064494 (In Eng.)

Rao S., Chen B., & Xu X. (2024) Heterogeneous CPU–GPU parallelization for modeling supersonic reacting flows with detailed chemical kinetics. *Computer Physics Communications, 300*, 109188. https://doi.org/10.1016/j.cpc.2024.109188 (In Eng.)

Sarsembayev M.S., & Urmashev B.A. (2019) Using CUDA technology to accelerate calculations in problems of chemical kinetics. *Cybernetics and Systems Analysis*. Springer New York. (In Eng.)

Stone C.P., & Davis R.L. (2013) Techniques for solving stiff chemical kinetics on graphical processing units. *Journal of Propulsion and Power, 29*(4). — P. 764–773. https://doi.org/10.2514/1.B34774 (In Eng.)

Uranakara H.A., Barwey S., Hernández Pérez F.E., Raman V., & Im H.G. (2023) Accelerating turbulent reacting flow simulations on many-core/GPU architectures. *Computers & Fluids*. (In Eng.)

Wang Y., Zeng Y., Deiterding R., & Liang J. (2026) An efficient GPU-accelerated adaptive mesh refinement framework for high-fidelity compressible reactive flows modeling. *Computer Physics Communications, 318*, 109870. https://doi.org/10.1016/j.cpc.2025.109870 (In Eng.)

Zhang W., Almgren A., Beckner V., Bell J., Blaschke J., Chan C., Day M., Friesen B., Gott K., Graves D., Katz M., Myers A., Nguyen T., Nonaka A., Rosso M., Williams S., & Zingale M. (2019) AMReX: A framework for block-structured adaptive mesh refinement. *Journal of Open Source Software, 4*(37), 1370. https://doi.org/10.21105/joss.01370 (In Eng.)

Publication Ethics and Publication Malpractice
in the journals of the Central Asian Academic Research Center LLP

For information on Ethics in publishing and Ethical guidelines for journal publication see http://www.elsevier.com/publishingethics and http://www.elsevier.com/journal-authors/ethics.

Submission of an article to the journals of the Central Asian Academic Research Center LLP implies that the described work has not been published previously (except in the form of an abstract or as part of a published lecture or academic thesis or as an electronic preprint, see http://www.elsevier.com/postingpolicy), that it is not under consideration for publication elsewhere, that its publication is approved by all authors and tacitly or explicitly by the responsible authorities where the work was carried out, and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder. In particular, translations into English of papers already published in another language are not accepted.

No other forms of scientific misconduct are allowed, such as plagiarism, falsification, fraudulent data, incorrect interpretation of other works, incorrect citations, etc. The Central Asian Academic Research Center LLP follows the Code of Conduct of the Committee on Publication Ethics (COPE), and follows the COPE Flowcharts for Resolving Cases of Suspected Misconduct (http://publicationethics.org/files/u2/New_Code.pdf). To verify originality, your article may be checked by the Cross Check originality detection service http://www.elsevier.com/editors/plagdetect.

The authors are obliged to participate in peer review process and be ready to provide corrections, clarifications, retractions and apologies when needed. All authors of a paper should have significantly contributed to the research.

The reviewers should provide objective judgments and should point out relevant published works which are not yet cited. Reviewed articles should be treated confidentially. The reviewers will be chosen in such a way that there is no conflict of interests with respect to the research, the authors and/or the research funders.

The editors have complete responsibility and authority to reject or accept a paper, and they will only accept a paper when reasonably certain. They will preserve anonymity of reviewers and promote publication of corrections, clarifications, retractions and apologies when needed. The acceptance of a paper automatically implies the copyright transfer to the Central Asian Academic Research Center LLP.

The Editorial Board of the Central Asian Academic Research Center LLP will monitor and safeguard publishing ethics.

Requirements for articles design for publication in the journal are available on the websites:
www:nauka-nanrk.kz
http://physics-mathematics.kz/index.php/en/archive
ISSN2518-1726 (Online),
ISSN 1991-346X (Print)

Managing Editor: *A.Shormakova*
Editors: *D.S. Alenov, T. Apendiev*
Computer layout: *G.D. Zhadyranova*

Signed for print: June 15, 2026
Format: 70×90 1/16. 26.5 printed sheets. Order No. 2.

«Central Asian Academic Research Center» LLP
Almaty, Shevchenko Street, 28