

УДК 519.63

## Гибридный GPU-алгоритм для моделирования осесимметричных реагирующих течений с жесткой химической кинетикой\*

Логинов В.В.<sup>1</sup>, Пескова Е.Е.<sup>1</sup>

Национальный исследовательский Мордовский государственный университет<sup>1</sup>

*Аннотация:* Представлен гибридный вычислительный алгоритм для моделирования осесимметричных течений многокомпонентного реагирующего газа с учётом детальной химической кинетики. Газодинамическая часть решается на GPU с использованием PyCUDA, а интегрирование жёсткой системы обыкновенных дифференциальных уравнений химической кинетики выполняется с помощью библиотеки JAX (решатель Kvaerno5) на том же GPU. Для передачи данных между PyCUDA и JAX применён механизм zero-copy через DLPack, что позволяет избежать лишнего копирования через оперативную память. Проведено сравнение производительности предложенного подхода с реализацией на C++ с использованием решателя Radau5, а также оценено влияние zero-copy на скорость обмена данными. Результаты демонстрируют, что гибридный GPU-алгоритм обеспечивает высокую эффективность для задач с большим числом ячеек и сложной химией.

*Ключевые слова:* осесимметричные течения, реагирующие газы, химическая кинетика, GPU, PyCUDA, JAX, DLPack.

### 1. Введение

Численное моделирование реагирующих газодинамических течений находит широкое применение в аэротермохимии, энергетике и химической технологии. Особую сложность представляют задачи с детальной кинетикой, включающей десятки компонентов и сотни реакций, что приводит к необходимости интегрирования жёстких систем ОДУ в каждой ячейке расчётной сетки. Традиционные подходы, основанные на CPU, часто становятся узким местом по производительности. В последние годы активно развиваются гибридные алгоритмы, использующие графические процессоры для ускорения как газодинамических, так и химических вычислений [1], в том числе с применением подходов к JIT-компиляции, аналогичных используемым [2].

В данной работе предлагается гибридный GPU-алгоритм, в котором газодинамическая часть реализована на PyCUDA [3], а интегрирование химической кинетики — на JAX [4] с применением JIT-компиляции и векторизации. Такой подход позволяет эффективно использовать вычислительные ресурсы GPU для обоих этапов, а механизм DLPack обеспечивает обмен данными без копирования между библиотеками. В статье описываются математическая модель, численный метод, особенности реализации и результаты тестирования производительности.

---

\*Исследование выполнено за счёт гранта Российского научного фонда № 26-11-00095.

## 2. Математическая модель

Рассматривается осесимметричное течение в цилиндрических координатах  $(z, r)$  с компонентами скорости  $u_z$  и  $u_r$ . Система уравнений сохранения массы компонентов, импульса и энергии в консервативной форме имеет вид:

$$\frac{\partial U}{\partial t} + \frac{\partial(F^{(1)}(U) - H^{(1)}(U))}{\partial z} + \frac{1}{r} \frac{\partial(r(F^{(2)}(U) - H^{(2)}(U)))}{\partial r} = W, \quad (1)$$

где  $U$  — вектор консервативных переменных:

$$U = \begin{pmatrix} \rho Y_m \\ \rho u_z \\ \rho u_r \\ \rho h \end{pmatrix},$$

$\rho$  — плотность смеси,  $Y_m$  — массовая доля  $m$ -го компонента ( $m = \overline{1, N_s}$ ,  $N_s = 15$ ),  $h$  — удельная энтальпия смеси. Векторы конвективных потоков:

$$F^{(1)}(U) = \begin{pmatrix} \rho u_z Y_m \\ \rho u_z^2 + \pi \\ \rho u_z u_r \\ \rho h u_z \end{pmatrix}, \quad F^{(2)}(U) = \begin{pmatrix} \rho u_r Y_m \\ \rho u_z u_r \\ \rho u_r^2 + \pi \\ \rho h u_r \end{pmatrix},$$

где  $\pi$  — динамическое давление. Диффузионные потоки  $H^{(1)}$  и  $H^{(2)}$  в настоящей работе не учитываются (вместо них используется схема с искусственной вязкостью Лакса–Фридрихса). Источниковый член:

$$W = \begin{pmatrix} R_m \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

где  $R_m$  — скорость образования/расхода  $m$ -го компонента.

Используется детальный механизм, включающий 15 компонентов и 40 элементарных реакций. Список компонентов: 1.  $C_2H_4$ , 2.  $H_2$ , 3.  $CH_4$ , 4.  $CH_3$ , 5.  $C_2H_5$ , 6.  $H$ , 7.  $C_3H_7$ , 8.  $C_3H_6$ , 9.  $C_2H_3$ , 10.  $C_2H_2$ , 11.  $C_3H_4$ , 12.  $C_3H_3$ , 13.  $C_6H_6$ , 14.  $C_6H_5$ , 15.  $C_2H_6$ .

Скорости реакций задаются законом Аррениуса:

$$k_i = A_i \exp\left(-\frac{E_{a,i}}{RT}\right),$$

где  $A_i$  и  $E_{a,i}$  – предэкспоненциальный множитель и энергия активации соответственно,  $R$  – универсальная газовая постоянная. Для каждой ячейки решается система ОДУ относительно концентраций  $x_m$  (моль/м<sup>3</sup>):

$$\frac{dx_m}{dt} = f_m(x_1, \dots, x_{N_s}, T),$$

где  $f_m$  – сумма скоростей образования/расхода по всем реакциям. Система является жёсткой, что требует использования неявных методов.

### 3. Численный метод и особенности реализации

Для решения системы (1) применяется метод расщепления по физическим процессам: на каждом временном шаге  $\tau$  последовательно выполняются газодинамический и химический этапы.

Расчётная область покрывается структурированной сеткой размером  $N_x$  ячеек по  $z$  и  $N_y$  по  $r$ . Добавляются фиктивные слои ( $K = 1$ ) для граничных условий. Конвективные потоки вычисляются по схеме Лакса–Фридрихса:

$$F_{i-1/2} = \frac{1}{2} (F(U_{i-1}) + F(U_i) - \alpha(U_i - U_{i-1})),$$

где  $\alpha$  – максимальная скорость распространения возмущений. Для учёта давления используется проекционный метод: после вычисления промежуточных полей скорости формулируется уравнение Пуассона для поправки давления  $\pi'$ :

$$\nabla^2 \pi' = \frac{\rho_0}{\tau} \left( \frac{\partial u^*}{\partial z} + \frac{1}{r} \frac{\partial(rv^*)}{\partial r} \right),$$

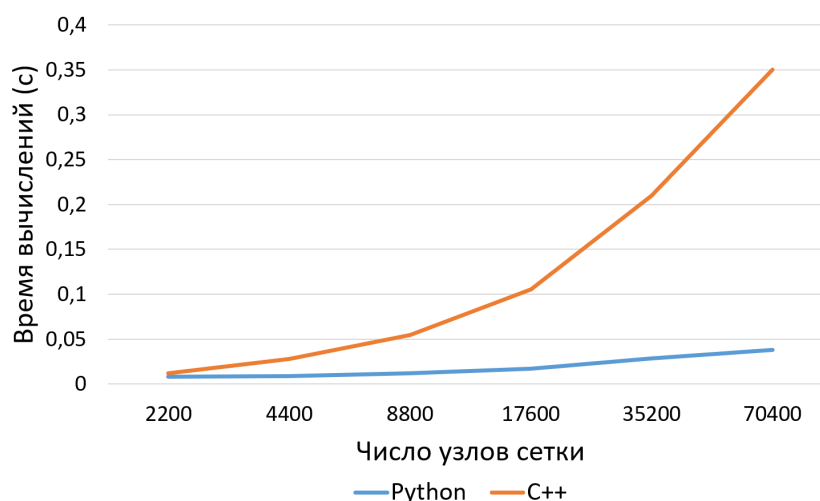
которое решается итерационным методом Якоби до сходимости  $10^{-5}$  [5]. Все газодинамические операции (расчёт потоков, обновление полей, итерации давления, коррекция скорости, граничные условия) реализованы в виде CUDA-ядер и вызываются через PyCUDA.

Интегрирование ОДУ выполняется с помощью библиотеки JAX и её расширения Diffax [6]. Используется адаптивный решатель Kvaerno5 (неявный метод Рунге–Кутты) с контролем шага (PIDController,  $\text{rtol}=\text{atol}=10^{-10}$ ). Для ускорения применяется векторизация по всем ячейкам, что позволяет одновременно интегрировать систему для всей сетки.

PyCUDA и JAX работают с разными абстракциями памяти GPU. Для обмена данными без лишнего копирования через CPU используется протокол DLPack, который позволяет передавать указатели на массивы в памяти GPU между различными фреймворками. В данном подходе массивы плотности, температуры и концентраций, хранящиеся в PyCUDA, преобразуются в объекты CuPy (которые поддерживают DLPack), затем передаются в JAX как дескрипторы DLPack, и наоборот. Это исключает этап копирования на CPU и значительно сокращает накладные расходы.

### 4. Результаты вычислительных экспериментов

Эффективность разработанного JAX-алгоритма оценивалась в сравнении с классическим решателем Radau5 на C++. Как видно из рис. 1, применение JAX на GPU позволяет сократить время вычислений примерно в 9 раз по сравнению с CPU-реализацией.

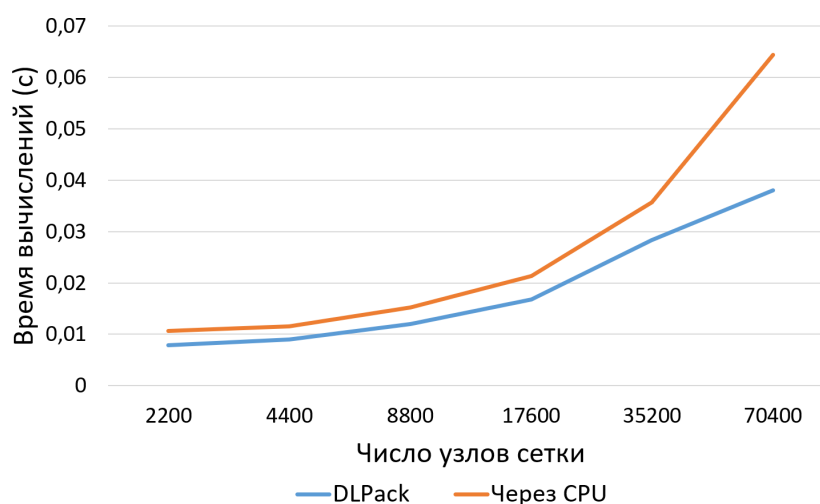


**Рис. 1.** Зависимость времени одного химического шага от числа ячеек.

Для передачи данных между PyCUDA и JAX сравнивались два подхода:

1. Zero-сору через DLPack.
2. Обычное копирование через промежуточные массивы NumPy (с чтением/записью через CPU).

Как показывают замеры времени (рис. 2), обмен данными через DLPack выигрывает в скорости на любых расчетных сетках. Уже на самых мелких сетках (2 200 узлов) стандартное копирование через процессор отстаёт примерно на 30%. По мере увеличения числа узлов этот разрыв только растет: на сетке размерностью 70 400 элементов DLPack сокращает время вычислений почти вдвое по сравнению с передачей через CPU.



**Рис. 2.** Сравнение времени передачи данных.

## 5. Заключение

Разработан гибридный GPU-алгоритм для моделирования осесимметричных реагирующих течений, сочетающий эффективную газодинамическую часть на PyCUDA и высокопроизводительное интегрирование химической кинетики на JAX. Использование метода расщепления по физическим процессам, схемы Лакса–Фридрихса, проекционного метода для давления и адаптивного решателя Kvaerno5 позволило достичь хорошей точности и устойчивости счёта. Применение DLPack для zero-copy обмена данными между библиотеками практически устраняет накладные расходы на передачу. Сравнение с C++ реализацией показало, что JAX обеспечивает лучшую производительность.

## Литература

1. Peskova E.E., Snytnikov V.N. The Influence of Laser Radiation on the Laminar Flow of a Chemically Active Gas-Dust Medium in a Narrow Circular Tube // Theoretical Foundations of Chemical Engineering. 2024. Vol. 58, No. 3. P. 517-525.
2. Numba. URL: <https://numba.pydata.org/>. Дата обращения: 10.07.2026.
3. PyCUDA. URL: <https://documen.tician.de/pycuda/>. Дата обращения: 10.07.2026.
4. Bradbury J. et al. JAX: composable transformations of Python+NumPy programs. 2018. URL: <http://github.com/google/jax>. Дата обращения: 10.07.2026.
5. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. Москва: Мир, 1991. 368 с.
6. Kidger P. Diffrax: a library for numerical integration of differential equations in JAX. 2021. URL: <https://github.com/patrick-kidger/diffrax>. Дата обращения: 10.07.2026.

MSC 65L20

## A hybrid GPU algorithm for modeling axisymmetric reacting flows with stiff chemical kinetics

V.V. Loginov<sup>1</sup>, E.E. Peskova<sup>1</sup>

National Research Ogarev Mordovia State University<sup>1</sup>

*Abstract:* A hybrid computational algorithm is presented for modeling axisymmetric flows of a multicomponent reacting gas with detailed chemical kinetics. The gas-dynamic part is solved on GPU using PyCUDA, while the integration of the stiff system of ordinary differential equations of chemical kinetics is performed with the JAX library (Kvaerno5 solver) on the same GPU. A zero-copy data exchange between PyCUDA and JAX is implemented via DLPack, avoiding unnecessary copying through host memory. The performance of the proposed approach is compared with a C++ implementation using the Radau5 solver, and the impact of zero-copy on data transfer speed is evaluated. The results demonstrate that the hybrid GPU algorithm provides high efficiency for problems with a large number of cells and complex chemistry.

*Keywords:* axisymmetric flows, reacting gases, chemical kinetics, GPU, PyCUDA, JAX, DLPack