

04

## Технические решения при кинетическом моделировании высокочастотного разряда: код KITE

© Л.А. Варшавчик, Д.Д. Галицын, Е.А. Старовойтов, В.А. Бочарников, С.А. Никитенко, Е.Е. Мухин

Физико-технический институт им. А.Ф. Иоффе РАН, Санкт-Петербург, Россия  
E-mail: lidia.varsh@mail.ioffe.ru

Поступило в Редакцию 5 июня 2024 г.

В окончательной редакции 27 сентября 2024 г.

Принято к публикации 1 октября 2024 г.

Разработан код KITE, предназначенный для кинетического 3D-моделирования транспорта атомов и ионов примеси в газе, в том числе во внешнем магнитном поле, и для моделирования емкостного высокочастотного разряда. Приведено описание основных технических решений: организация параллельной работы кода, управление группами частиц в расчете, беспроигрышный выбор параметров частиц из имеющихся ограничения распределений, применение иерархии ограничивающих объемов (BVH) для обработки столкновений частиц со стенками и использование параллелизма на уровне данных (инструкции AVX). Такой подход позволяет проводить моделирование в среднем за одни-двое суток расчетного времени.

**Ключевые слова:** метод частиц в ячейках, кинетическое моделирование, высокочастотный разряд, KITE.

DOI: 10.61011/PJTF.2025.03.59816.20011

Газовый высокочастотный (ВЧ) емкостной разряд применяется в широком спектре задач, таких как напыление пленок, травление поверхностей, модификация материалов, существует также ряд космических приложений. Большинство разработанных кодов по моделированию ВЧ-разряда работает в гидродинамическом приближении при высоких давлениях. Среди кинетических кодов, применимых при низких давлениях, распространено двумерное приближение, использующее симметрию модели. Нашей группой разработан код KITE [1], позволяющий проводить полностью трехмерное моделирование емкостного ВЧ-разряда низкого давления (до 10 Па), результаты которого могут помочь в разработке и оптимизации соответствующих технологий. Высокая ресурсоемкость и техническая сложность реализации данной задачи приводят к тому, что в мире существует лишь несколько подобных программ. Поэтому представляется важным дать описание технических решений и оптимизаций, обеспечивающих работоспособность и быстродействие разработанного нами кода KITE.

Для моделирования используется классический метод частиц в ячейках (PIC), в котором плазма рассматривается как набор тестовых частиц с определенным статистическим весом. Временная шкала разбивается на малые интервалы  $\Delta t$ , на каждом из которых пересчитываются скорости и координаты частиц. В начале расчета в объеме или на 2D-поверхностях создаются потоки идентичных тестовых частиц, для которых задаются распределения энергии, плотности потока частиц и трехканальное распределение для вектора скорости, где каждый канал отвечает за распределение на оси  $x$ ,  $y$  или  $z$ . Для ускорения генерации частиц применяется

алгоритм построения распределения таким образом, чтобы создаваемая частица гарантированно удовлетворяла требуемым условиям. Используется кэш распределений: если распределение уже в каком-то виде используется в KITE, то оно не строится повторно.

Для работы с тестовыми частицами был разработан менеджер частиц, призванный в первую очередь уменьшить объем потребляемой памяти при расчете. Частицы разделяются на несколько групп. Это позволяет унифицировать методы работы с большим количеством частиц: общие для набора частиц данные не дублируются в каждом объекте частицы. Частицы распределяются на четыре группы: отслеживаемые и неотслеживаемые частицы, созданные в потоке и имеющие его характеристики; отслеживаемые и неотслеживаемые частицы определенного элемента, не связанные с потоком (созданные в результате столкновения или распыления стенки). При расчете могут быть задействованы не все группы. Присутствует механизм сборщика мусора для слабо используемых групп. Структура группы включает в себя параметры всех частиц, количество элементов (частиц) в этой группе, размер в памяти объекта частицы, размер блока в памяти для хранения заряда, принадлежность частиц к потоку и смещение при записи в память. Добавление новых частиц и их перебор осуществляются параллельно через набор предварительно инициализированных потоков (*thread pool*), при создании частицы временно блокируется запись в ее группу. В зависимости от группы различается алгоритм извлечения частицы из бинарных данных. Старые частицы уничтожаются не сразу: запись только что созданных частиц происходит на место уничтоженных.

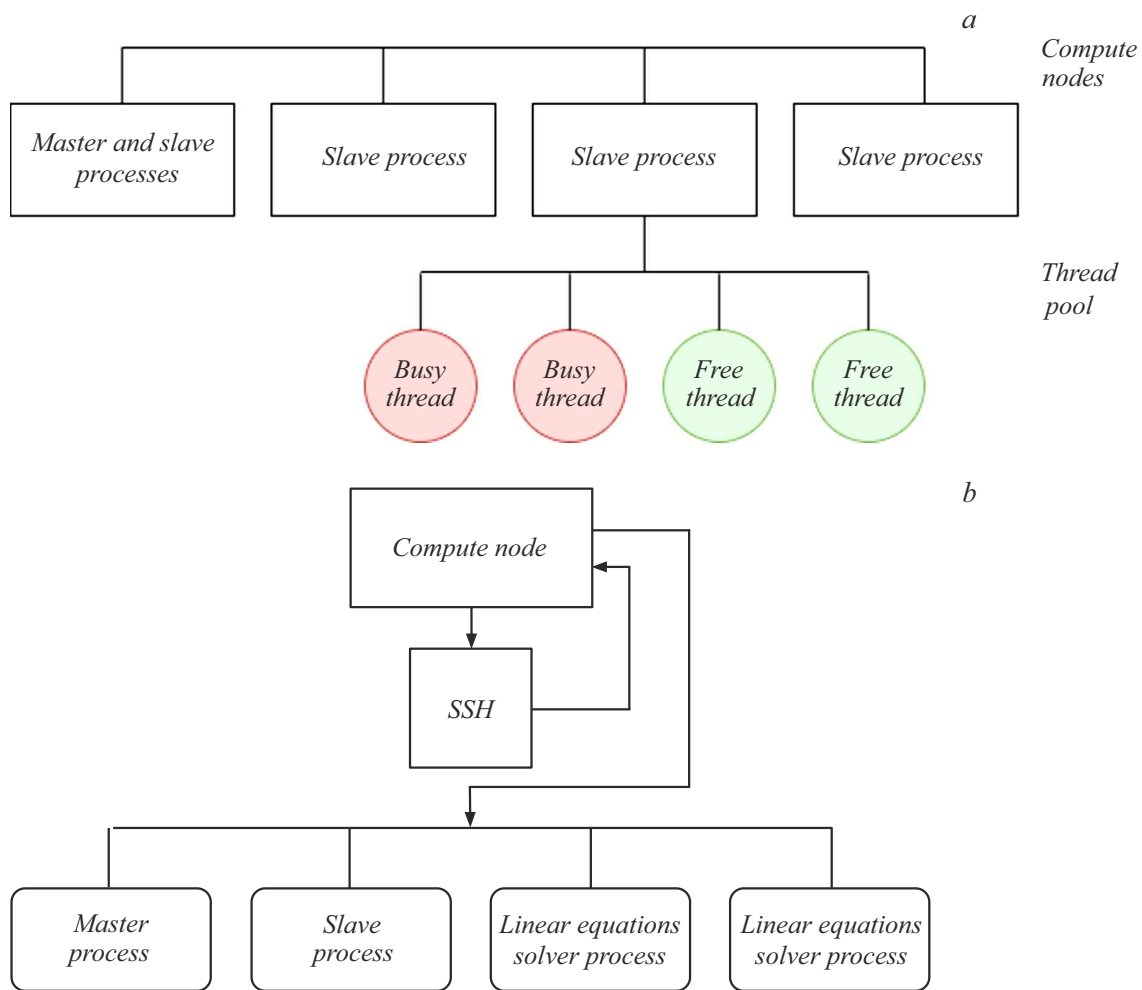
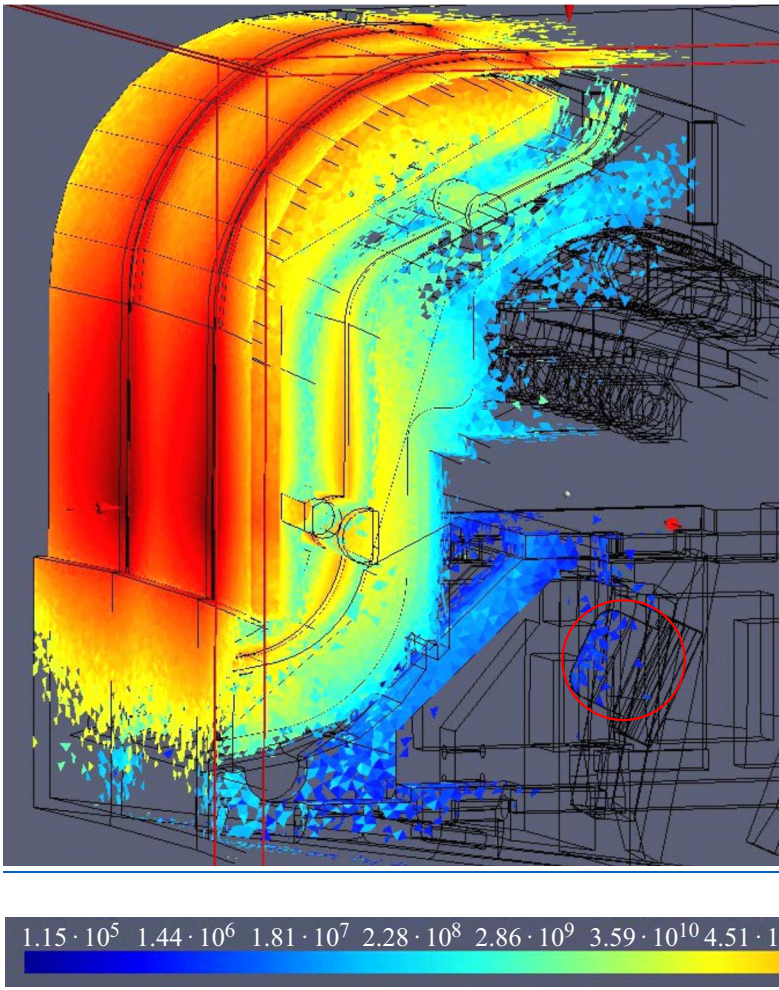


Рис. 1. Схема параллельной работы KITE при моделировании транспорта частиц (a) и при расчете электрических полей (b).

При моделировании столкновения частицы со стенкой используется ускоряющая структура данных, которая позволяет снизить сложность проверки на пересечение между объектами. Ускоряющие структуры — геометрические базы данных, предназначенные для эффективного хранения и быстрого поиска данных, — можно разделить на две категории: структуры с разбиением пространства и структуры с разбиением объектов. При разбиении пространства объем геометрии делится на несколько областей, и происходит запись того, какие объекты попали в ту или иную область. Во время проверки на пересечение сначала вычисляется последовательность областей, через которые проходит объект, а затем идет проверка на пересечение с объектами в заданных областях. Примером такой структуры является  $k$ -мерное дерево. Разбиение объектов предполагает, что все объекты геометрии разделяются на более мелкие составляющие этой геометрии. Одной из таких структур является иерархия ограничивающих объемов (BVH) [2], она была выбрана для решения задач, поставленных перед KITE.

На данный момент работа с BVH в KITE направлена на уменьшение количества условных переходов, что привело к уменьшению нагрузки на блок предсказаний (который угадывает, будет ли выполнен условный переход или нет) и вероятности сброса вычислительного конвейера (если предсказание выполнено неверно, то все предварительные вычисления, заложенные в конвейер, сбрасываются), а также на повышение вероятности попадания следующего сегмента BVH в кэш и улучшение возможности работы с SIMD (см. далее). Применение этой структуры позволяет существенно ускорить сбор статистики об адсорбировавшихся на стенках частицах, построение и оптимизацию системы линейных алгебраических уравнений (СЛАУ) для расчета электрических полей, а также дает возможность работы с динамической стенкой, в которой учитываются последовательно напыленные слои вещества. В дальнейшем использование BVH позволит легко адаптировать код KITE под вычислительный кластер с графическим процессором (GPU) в случае работы на таком.

При организации параллельной работы кода моделирование делится на два этапа: трассировка частиц и рас-



**Рис. 2.** Расчет в КИТе плотности  $[\text{cm}^{-2}]$  атомов бериллия, осажденных на стенки диагностики диверторного томсоновского рассеяния ИТЭР. Красным кружком обозначено первое зеркало. Цветной вариант рисунка представлен в электронной версии статьи.

чет локальных электрических полей. При трассировке параллельная работа организуется „по частицам“ (другой подход — параллельность „по пространству“). При расчете полей параллельность реализуется в решении СЛАУ, к которой сводится уравнение Пуассона.

Расчеты кодом КИТе запускаются на вычислительном кластере на CPU, поэтому используется MPI (Message Passing Interface) — интерфейс передачи сообщений между процессами, выполняющими одну задачу. Эта технология позволяет запускать код КИТе на нескольких узлах вычислительного кластера одновременно, разделяя выполнение кода между ними и, как следствие, уменьшая время расчета. Такой принцип реализован при запуске расчета транспорта частиц (рис. 1, *a*): на одном узле создаются два процесса, ведущий и ведомый (*master* и *slave*), а на всех остальных узлах создается по одному ведомому процессу. Каждый из ведомых процессов (т.е. каждый вычислительный узел) использует свой „стационарный“ пул потоков (*thread pool*), существующий до завершения работы всего кода, и обладает

своим собственным автономным менеджером частиц. На каждой итерации работы программы выполняется синхронизация между потоками.

В то же время MPI используется и при реализации решателей СЛАУ по причине того, что их реализация в библиотеках рассчитана на параллельность по процессам, а не по потокам, причем в большинстве случаев именно с использованием MPI. Для создания дополнительной группы процессов, принадлежащих решению СЛАУ, должна использоваться функция `MPI_Comm_spawn`, создающая максимально возможное количество копий одного процесса с MPI. Таким образом, кроме процессов, которые были созданы изначально для ведения расчета, создаются дополнительные процессы для решения СЛАУ. На время решения СЛАУ основная группа процессов приостанавливается, и наоборот, когда управление возвращается основной группе, дополнительные процессы приостанавливаются. Очередность остановки и повторного запуска процессов производится при помощи именованных каналов FIFO.

На вычислительном кластере ФТИ им. А.Ф. Иоффе используется система управления заданиями Slurm, которая не поддерживает функцию MPI\_Comm\_spawn. В данный момент используется обход этого ограничения (рис. 1, *b*), позволяющий запускать код на кластере, но только на одном вычислительном узле: KTe „арендует“ один узел кластера и, находясь в этом узле, подключается к нему же через SSH, и через SSH-подключение запускается MPI. При таком подходе не используются полностью возможности вычислительного кластера, хотя расчет происходит быстрее, чем на ПК или сервере. Чтобы решить эту проблему, требуется переработка архитектуры кода таким образом, чтобы изначально выделялось необходимое количество процессов, а после этого создавались вспомогательные коммутаторы, чтобы поделить группы расчета транспорта и решателей СЛАУ и организовать между ними общение.

Дополнительное ускорение вычислений можно получить, объединив выполнение однотипных операций линейной математики и поиска пересечений в BVH. Применяется принцип компьютерных вычислений SIMD [3], позволяющий осуществить параллелизм на уровне работы с данными, выполняя одну инструкцию одновременно для набора элементов. На текущий момент большинство процессоров, включая применяющиеся на кластере ФТИ им. А.Ф. Иоффе процессоры Intel, обладает этой инструкцией или расширением. В данном случае технология SIMD реализована в виде набора инструкций Advanced Vector Extensions (AVX). За счет AVX сокращается время на декодирование команд, самую сложную стадию внутри процессора. Улучшается работа процессора с оперативной памятью (она на порядок медленнее, чем операции внутри процессора), так как на вход поступают линейные участки памяти и сами команды представлены более компактно. Обеспечивается более эффективная загрузка вычислительных портов, поскольку пропадает конкуренция между ними.

Описание используемых методов для моделирования ВЧ-разряда, верификация кода KTe, сравнение результатов расчета транспорта бора и бериллия готовятся к публикации. Иллюстрация работы кода KTe на примере расчета скорости загрязнения первого зеркала диагностики диверторного томсоновского рассеяния ИТЭР атомами бериллия, распыленными с первой стенки, представлена на рис. 2. Моделирование проводилось на кластере ФТИ им. А.Ф. Иоффе на одном узле (96 потоков) и заняло 32.3 часа расчетного времени. В качестве входных данных использовались плотности потоков бериллия и дейтерия на входе в диагностический канал и их распределения по энергии, рассчитанные в программе SOLPS-ITER. При моделировании в KTe в расчет было запущено  $2.79 \cdot 10^8$  тестовых частиц (Be, D и D<sub>2</sub>), из них  $5.52 \cdot 10^5$  атомов Be. При движении от входа в канал к первому зеркалу частицы трижды клонируются 50, 50 и 30 раз с соответствующим уменьшением веса. При этом на первое зеркало долетело всего 18 тестовых частиц

бериллия, однако по ним уже можно сделать оценку скорости загрязнения зеркала.

Внедрение в код KTe изложенных выше принципов существенно ускорило проведение кинетического 3D-моделирования транспорта и ВЧ-разряда, что позволяет оперативно получать результаты моделирования с достаточной статистикой для численных оценок.

### Финансирование работы

Работа выполнена при частичной поддержке гранта Фонда развития теоретической физики и математики „БАЗИС“.

### Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

### Список литературы

- [1] L.A. Varshavchik, N.A. Babinov, P.A. Zatytkin, A.A. Chironova, Z.G. Lyullin, A.I.P. Chernakov, A.M. Dmitriev, I.M. Bukreev, E.E. Mukhin, A.G. Raazdobarin, D.S. Samsonov, V.A. Senitchenkov, S.Yu. Tolstyakov, I.T. Serenkov, V.I. Sakharov, Plasma Phys. Control. Fusion, **63** (2), 025005 (2021). DOI: 10.1088/1361-6587/abca7e
- [2] M. Pharr, W. Jakob, G. Humphreys, *Physically based rendering: from theory to implementation* (Elsevier/Morgan Kaufmann, Amsterdam–Boston, 2004), ch. 4.3.
- [3] <https://arstechnica.com/features/2000/03/simd/>