

Численное моделирование процесса фокусировки ионов в пылеударном времяпролетном масс-спектрометре

И.В. Пияков¹, Д.В. Родин¹, М.А. Родина¹, А.М. Телегин¹

¹Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

Аннотация. Приведено описание конструкции пылеударного времяпролетного масс-спектрометра. Рассмотрен способ расчета траектории заряженной частицы на треугольной сетке. Дано описание реализаций алгоритма расчета для персонального компьютера, GPU ускорителя и суперкомпьютера. Приведены основные различия реализаций, сравнение производительности и точности расчета. Рассмотрены преимущества и недостатки каждой из реализаций.

1. Введение

Исследование свойств микрометеороидов и частиц космического мусора *in-situ* невозможно без использования соответствующей регистрирующей аппаратуры. Так, для анализа химического состава частиц используются пылеударные времяпролетные масс-спектрометры. Преимуществом данного типа анализаторов являются высокое быстродействие, возможность регистрации однократных событий, малый размер и энергопотребление. Особенностью пылеударных времяпролетных масс-спектрометров является ионизация высокоскоростным соударением частицы-ударника и мишени, которая происходит случайным образом в произвольной точке мишени. Повышение вероятности ионизации при таком способе достигается за счет увеличения активной площади мишени, что приводит к необходимости дополнительной пространственной фокусировки полученных ионов с использованием параболических отражателей или электростатических отражателей со специальной формой электрического поля. Плазма, образовавшаяся в результате высокоскоростного соударения, обладает высоким энергетическим разбросом, что обуславливает необходимость применения электростатических ионных зеркал, обеспечивающих временную фокусировку ионных пакетов.

Принцип работы приборов времяпролетного типа основан на измерении времени пролета частиц с известной энергией от источника ионов до приемника. В таком, идеальном, случае время пролета зависит только от массы иона. Разработка же реальных приборов основана на определении конструктивных параметров, обеспечивающих максимальную независимость от начальной энергии ионов их времени пролета от места соударения до приемника. С этой целью рассчитывают одномерную структуру поля [1], обеспечивающую требуемую временную фокусировку и, используя численное моделирование, вырабатывают требования к элементам конструкции, формирующим близкую к расчетной структуру поля в максимальном объеме конструкции. Проверка каждого из вариантов конструкции осуществляется путем моделирования траекторий модельных ионных пакетов с последующим расчетом разрешающей способности. В данной работе рассмотрено программное обеспечение, реализующее массивно-

параллельный алгоритм расчета траекторий частиц на треугольной сетке, обеспечивающей возможность моделирования криволинейных элементов конструкции.

2. Постановка задачи

Структурная схема рассматриваемого прибора представлена на рисунке 1. Подробное описание прибора приведено в [2].

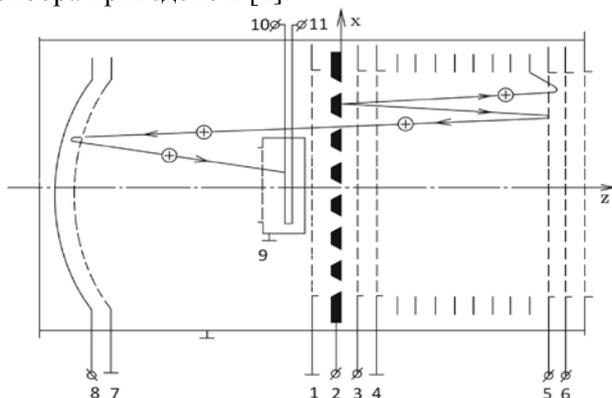


Рисунок 1. Пылеударный масс-спектрометр с параболическим отражателем (2 – мишень; 3 – ускоряющая сетка; 4 – тормозящая сетка, заземлена; 5, 6 – сетки ионного зеркала; 1 – экранирующая сетка, заземлена; 7, 8 – параболический отражатель; 10, 11 – приёмник ионов; 9 – кожух приемника, заземлен).

Устройство работает следующим образом. При ударе частицы о мишень 2 вещество частицы ионизируется, ускоряется электрическим полем и попадает в ионное зеркало, ограниченное сетками 4, 5, 6, где отражается и, проходя назад сквозь мишень (к этому моменту ускоряющий потенциал отключается), пролетает через бесполевого пространство между сетками 1 и 8, отражается в параболическом отражателе и попадает на вход 10 приемника, на его выходе 11 образуется ионный импульс.

Особенностью отражателя является то, что он имеет форму параболоида с фокусом в приемнике ионов. Это способствует дополнительной пространственной фокусировке ионных пучков. Таким образом уменьшаются потери ионов, что способствует увеличению чувствительности устройства. Фокусировка ионов по энергиям производится в ионном зеркале.

Основными параметрами времяпролетного масс-спектрометра является его разрешающая способность и чувствительность. При этом их величины зависят от многих факторов:

- конструктивные параметры времяпролетного масс-спектрометра;
- собственная энергия ионов, образовавшихся в результате высокоскоростного соударения;
- направление начальных векторов скоростей ионов;
- координата удара частицы о мишень;
- химический состав частиц

Для определения влияния всех этих факторов на интересующие нас характеристики требуется рассчитать траектории ионов в тракте прибора. С этой целью для схемы масс-спектрометра задаются: расстояния между сетками и ускоряющий потенциал, габариты мишени и колец, кривизна параболического отражателя, диаметр центрального отверстия мишени, загружаются параметры ползающих колец и заранее вычисляется сетка с распределением электростатической напряженности внутри тракта прибора, которая используется на этапе моделирования работы прибора.

3. Алгоритм расчета

Численное моделирование движения заряженных частиц может быть реализовано как с использованием сетки значений поля, полученной методом конечных разностей, так и с использованием узловых значений, полученных методом конечных элементов. Преимуществом первого метода являются наглядность, простота реализации функций поиска ячейки и кубической интерполяции, легкость стыковки сеток при расчете в нескольких областях. Однако существенным недостатком этого типа сеток является невозможность расположения узлов на криволинейных поверхностях. Метод конечных элементов позволяет разбивать произвольные области на отдельные элементы (в частном случае – треугольники), что обеспечивает

возможность точного восстановления формы интересующих областей. Вместе с тем, использование сеток, полученных методом конечных элементов, требует реализации дополнительных функций, например, поиска треугольника, которому принадлежит точка, а также более сложных, чем в методе конечных разностей, структур данных для хранения информации о сетке, представляющей собой граф с описанием узлов и треугольников.

На основе данных о координате взаимодействия и скорости формируется ионный пакет с известной величиной энергетического разброса [3]. Для каждого иона в ионном пакете с заданным шагом по времени вычисляется приращение координат. После каждой итерации происходит проверка на соударение с элементом конструкции, вылет и детектирование. Времена пролета протектированных ионов сохраняются. Для оценки характеристик конструкции для каждой массы требуется вычислить количество ионов, оценить коэффициент сбора, среднее время пролета, среднеквадратичное отклонение и разрешение прибора по двум сигма.

Пошагово алгоритм работы программы на треугольной сетке можно описать следующим образом.

1. Загрузка данных о расчетной сетке с полем из файла.
2. Разбор файла: увязывание узлов, треугольников, областей и значений поля.
3. Создание объекта Mesh с описанием взаимосвязей узлов и треугольников.
4. Формирование модельного ионного пакета, исходя из заданных координат и скорости взаимодействия.
5. Нахождение текущего треугольника для каждого иона.
6. Интерполяция поля в точке нахождения каждого иона.
7. Расчет смещения каждого иона за время dt .
8. Проверка нахождения иона в текущем треугольнике.
9. Рекурсивный поиск нового треугольника.
10. Проверка на вылет/детектирование – возврат на пункт 6.

В целом структура реализации схожа с описанной в статье [4]. Так, в пункте 4 для формирования пакета с Максвелловским распределением по скоростям аналогично используется алгоритм Бокса-Мюллера, расчет смещения иона также может быть произведен как методом трапеций, так и методом Рунге-Кутты. Отличие данной реализации заключается в использовании метода билинейной интерполяции для треугольной сетки и необходимости проверки на принадлежность координат частицы текущему треугольнику. Обе задачи решены с использованием известных подходов с небольшими модификациями. Для ускорения операции интерполяции каждый экземпляр класса треугольник хранит предварительно рассчитанные коэффициенты интерполяции $s_1, t_1, s_2, t_2, s_3, t_3, A, \text{coeff}$, а для ускорения поиска нового треугольника, которому принадлежат координаты частицы, этот же экземпляр хранит ссылки на соседние треугольники, что позволяет заменить поиск по всем треугольникам сетки на рекурсивный поиск от текущего треугольника.

4. Особенности реализации алгоритма расчета траекторий для CPU, GPU, HPC

Реализация для CPU написана на языке C# и содержит четыре основных класса для осуществления расчета траекторий. Класс Point используется для сохранения узлов графа, содержит номер узла, информацию о координатах, значениях поля, а также номера треугольников, которым принадлежит этот узел. Класс Triangle содержит номер треугольника, три ссылки на узлы-вершины этого треугольника, информацию о домене, которому принадлежит треугольник, три ссылки на соседние треугольники, предварительно рассчитанные коэффициенты для интерполяции, а также методы для билинейной интерполяции поля и поиска нового треугольника при выходе точки за пределы текущего треугольника. Класс Mesh содержит массивы узлов и треугольников, а также методы для поиска ближайшей точки, треугольника, чтения и записи файлов. Класс Particle содержит информацию о массе и заряде частицы, ее скорости, координатах, текущем треугольнике, а также метод, реализующий итеративный расчет траектории на треугольной сетке.

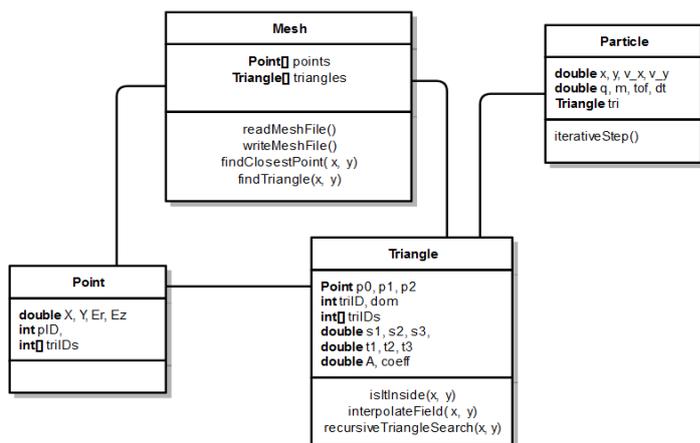


Рисунок 2. Структура классов для хранения информации о расчетной сетке, PC реализация на языке C#.

Использование графического ускорителя накладывает жесткие ограничения на реализацию программного обеспечения, приводящие к необходимости, по возможности, всю информацию представлять в виде массивов. Таким образом, реализация для GPU содержит следующие массивы: triangleNeighbours, trianglePoints – массивы векторного формата uint4 для хранения информации о соседних треугольниках и вершинах треугольников. Массивы s1_t1, s2_t2, s3_t3, A_coeff векторного формата double2 для хранения информации о коэффициентах для интерполяции поля. Массивы pointsCoordinates и pointsFields векторного формата double2 для хранения информации об узлах сетки. Массивы particlesCoordinates и particlesSpeeds векторного формата double2 для хранения информации о координатах и скоростях частиц. Массив particlesTimes формата double1 для хранения текущего времени частицы. Массив particlesTriangles формата uint1 для хранения номера текущего треугольника частицы. Во всех массивах в явном виде отсутствуют номера узлов, треугольников и частиц, вместо них используется индекс в массиве. Данные массивы создаются в host-памяти и заполняются данными из реализации для CPU, после чего создаются массивы в GPU памяти и данные из host-массивов копируются на GPU. Затем к GPU массивам проводится привязка текстурных и surface ссылок, что позволяет использовать операции текстурных выборок, реализующих аппаратное кэширование. Необходимо отметить, что привязка текстурных ссылок невозможна к массивам векторных типов double, поэтому при копировании данных с CPU на GPU происходит битовая конвертация одного числа типа double в пару чисел типа int, после текстурной выборки происходит обратная битовая конвертация.

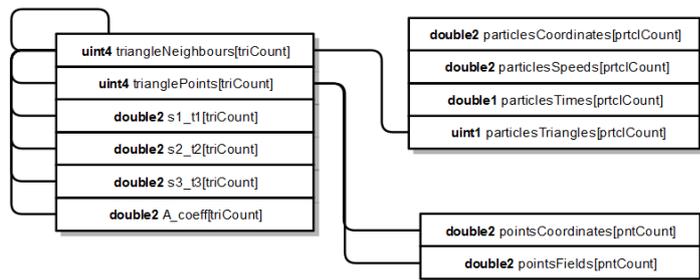


Рисунок 3. Структура массивов для хранения информации о расчетной сетке, GPU реализация на языке CUDA.

Реализация для суперкомпьютера Сергей Королев отличается как от версии для ПК, так и от версии для GPU. Поскольку эта реализация написана для компилятора MPICC, и используется вариант языка C, не поддерживающий ООП, данные о треугольниках и их связях между собой и с узлами хранятся в массивах triangleNeighbours, trianglePoints формата int, данные о коэффициентах для интерполяции – в массиве triangleCoeff, формата double, данные о координатах узлов и значениях поля в них – в массиве pointsCoordinatesAndFields формата double, данные о частицах – в массиве particles формата double. Данная реализация не требует дополнительных преобразований типов при обращении к памяти, во всех массивах также в

явном виде отсутствуют номера узлов, треугольников и частиц, вместо них используется индекс в массиве.

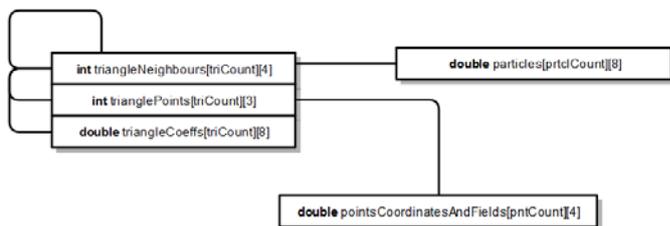


Рисунок 4. Структура массивов для хранения информации о расчетной сетке, НРС реализация на языке С.

5. Результаты расчета и сравнение производительности

Для наглядной визуализации процесса моделирования и оценки работоспособности конструкции был написан модуль визуализации, отображающий движение ионных пакетов в объеме конструкции. На рисунке 5 совмещены несколько кадров визуализации, хорошо видна фокусировка ионных пакетов по времени: две массы хорошо разделены у плоскости приемника.

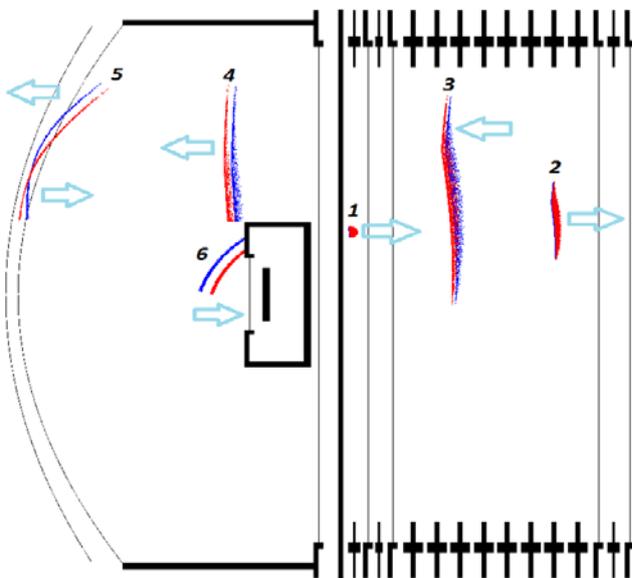


Рисунок 5. Визуализация трассировки ионных пакетов. Пакет из двух масс образуется в точке 1, ускоряется и вылетает в ионное зеркало 2, отражается и летит обратно к мишени 3, пролетает сквозь мишень, часть ионов оседает на кожухе приемника 4, отражается в параболическом отражателе 5 и летит в приемник 6.

Результаты моделирования конструкции времяпролетного масс-спектрометра с линейным ионным зеркалом и параболическим отражателем представлены на рисунках 6 и 7. Расчеты произведены для 16382 частиц без учета коэффициентов прозрачности сеток и мишени.

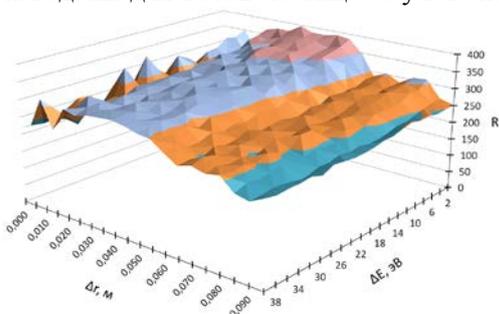


Рисунок 6. Зависимость разрешающей способности прибора от координаты иона соударения и энергетического разброса.

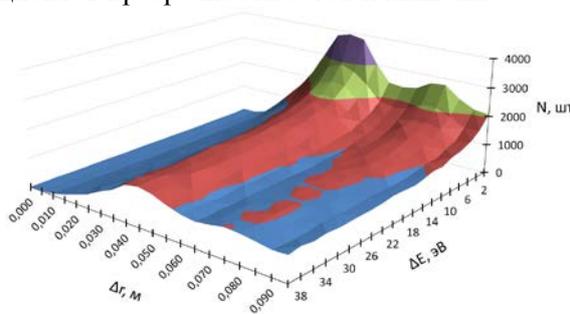


Рисунок 7. Зависимость количества принятых ионов от координаты соударения и энергетического разброса.

Разрешающая способность имеет локальный максимум на нулевом радиальном смещении. Коэффициент сбора ионов имеет максимумы на 35 мм и 75 мм радиального смещения при минимальном энергетическом разбросе.

Сравнение реализаций для персонального компьютера, GPU ускорителя и суперкомпьютера производилось для различного количества частиц. Сравнение не выявило различий в точности расчетов, что в целом объясняется тем, что все расчетные модули реализованы с использованием переменных типа double. HPC реализация запускалась на четырех процессорах.

Результаты измерений времени выполнения 10000 итераций приведены в таблице 1.

Таблица 1. Результаты измерения времени выполнения 10000 итераций расчета траектории для реализации на PC, GPU, HPC.

	PC parallel. (4CPU, 8 threads)	PC Single thread	GPU Host calls	GPU Device calls	HPC Sergey Korolev
16384	7с	24с	6с	6с	6с
65536	29с	1м 34с	23с	23с	18с
262144	1м 53с	6м 19с	1м 29с	1м 29с	1м 19с

6. Заключение

Используя представленное выше программное обеспечение, авторы промоделировали различные варианты конструкций пылеударного масс-спектрометра. Моделирование конструкции с линейным зеркалом и параболическим отражателем показало относительно невысокую разрешающую способность порядка 250 – 300 единиц независимо от энергетического разброса и координаты соударения. Однако по совокупности факторов, таких как достигаемая разрешающая способность, коэффициент сбора ионов, простота настройки, эта конструкция оказывается более предпочтительной, чем другие. Так, в конструкциях с плоским отражателем резко падает коэффициент сбора ионов, а в конструкциях с нелинейным ионным зеркалом не удастся достигнуть независимости разрешающей способности от координаты соударения.

Представленное программное обеспечение позволяет осуществить моделирование конструкций с осевой симметрией, содержащих криволинейные электроды и неоднородные электрические поля. В отличие от реализаций, использующих прямоугольные сетки [4, 5], реализация для GPU ускорителя (2880 ядер по 1020 МГц) не показала существенного роста производительности, что объясняется более высоким количеством ветвлений во время выполнения алгоритма, реализующего расчет на треугольной сетке. Однако даже такой скромный прирост скорости выполнения может быть полезен при решении задач оптимизации конструкций, требующих многократной трассировки ионных пакетов, при общем количестве расчетных проходов, достигающем нескольких десятков тысяч, и общем времени выполнения порядка нескольких суток или недель.

Реализация для HPC оказалась более производительной при том же количестве процессоров, что может быть объяснено лучшей работой с памятью бинарного кода, полученного MPICC компилятором. Благодаря тому, что данная задача является параллельной по исходным данным, повысить производительность можно путем увеличения количества вычислительных узлов.

7. Литература

- [1] Semkin, N.D. Analytical Method for Computing the Electrostatic Field Distribution in the Reflector of the Time-of-Flight Mass Spectrometer / N.D. Semkin, I.V. Piyakov, D.V. Rodin, R. A. Pomel'nikov // Technical Physics. – 2012. – Vol. 57(10). – P. 1400-1405.

- [2] Семкин, Н.Д. Бортовой пылеударный масс-спектрометр для исследования элементного состава микрометеороидов / Н.Д. Семкин, Д.В. Родин, И.В. Пияков, Р.А. Помельников // Научное приборостроение. – 2012. – Т. 22, № 3. – С. 13-20.
- [3] Hornung, K. Impact Vaporization and Ionization of Cosmic Dust Particles / K. Hornung, Y.G. Malama, K.S. Kestenboim // Astrophysics and Space Science. – 2000. – Vol. 274(1-2). – P. 355-363.
- [4] Piyakov, A.V. Numerical simulation of motion of dust particles in an accelerator path / A.V. Piyakov, D.V. Rodin, M.A. Rodina, A.M. Telegin // CEUR Workshop Proceedings. – 2017. – Vol. 1902. – P. 55-61.
- [5] Семкин, Н.Д. Метод расчета потенциалов ползающих элементов нелинейного ионного зеркала / Н.Д. Семкин, Д.В. Родин, И.В. Пияков // Научное приборостроение. – 2013. – Т. 23, № 3. – С. 69-75.

Numerical simulation of the ion focusing process in a dust impact time of flight mass spectrometer

I.V. Piyakov¹, D.V. Rodin¹, M.A. Rodina¹, A.M. Telegin¹

¹Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

Abstract. A description of a design of dust-impact time of flight mass-spectrometer is given. A method for calculating the trajectory of a charged particle on a triangular grid is considered. The implementation of the calculation algorithm for the personal computer, the GPU accelerator and the supercomputer is described. The main differences in implementations, comparison of performance and accuracy of calculation are given. The advantages and disadvantages of each of the implementations are considered.

Keywords: mass spectrometer, time of flight, CUDA, supercomputer, focusing, ion packet.