

# **Моделирование движения пучка в электромагнитном поле с помощью GPU**

Кадров Виктор Максимович

# Содержание

<b>1 Введение</b>	<b>4</b>
<b>2 Замена гравитационного взаимодействия на электрическое</b>	<b>6</b>
<b>3 Постановка задачи</b>	<b>7</b>
<b>4 Численное решение уравнений движения</b>	<b>8</b>
<b>5 Трудности в расчёте собственного поля</b>	<b>9</b>
<b>6 Метод крупных частиц</b>	<b>11</b>
<b>7 Метод Particle-In-Cell</b>	<b>12</b>
<b>8 Ускорение расчётов с использованием графических процессоров</b>	<b>13</b>
<b>9 Архитектура собственной реализации</b>	<b>14</b>
<b>10 Результаты моделирования</b>	<b>16</b>
<b>11 Дальнейшая работа</b>	<b>18</b>
<b>12 Заключение</b>	<b>20</b>
<b>Список литературы</b>	<b>21</b>

# Список иллюстраций

1.1	Пример моделирования гравитационной задачи (Source: Nvidia Developer Guide) . . . . .	5
5.1	Система из 256 частиц с разным зарядом . . . . .	10
7.1	PIC . . . . .	12
9.1	Схема программного решения . . . . .	15
10.1	Пучок 100 тыс. частиц на старте симуляции, вектор скорости направлен в катушку . . . . .	16
10.2	Фокусировка пучка . . . . .	17
10.3	Разлет на выходе из катушки . . . . .	17
11.1	Архитектура памяти GPU . . . . .	19

# 1 Введение

В докладе рассматривается подход к моделированию взаимодействия заряженных частиц в пучке. В качестве отправной точки анализируется задача с гравитационным взаимодействием (рис. 1.1), поскольку она во многом аналогична электрической, но проще для понимания на первом этапе.

$$\begin{cases} \dot{\vec{v}}_i = G \sum_{j \neq i} \frac{m_j}{||\vec{r}||^3} \vec{r}, \\ \dot{\vec{x}}_i = \vec{v}_i, \\ \vec{x}_i(0) = \vec{x}_0, \vec{v}_i(0) = \vec{v}_0, \end{cases}$$

где  $\vec{r} = \vec{r}_i - \vec{r}_j$  – радиус-вектор между телами

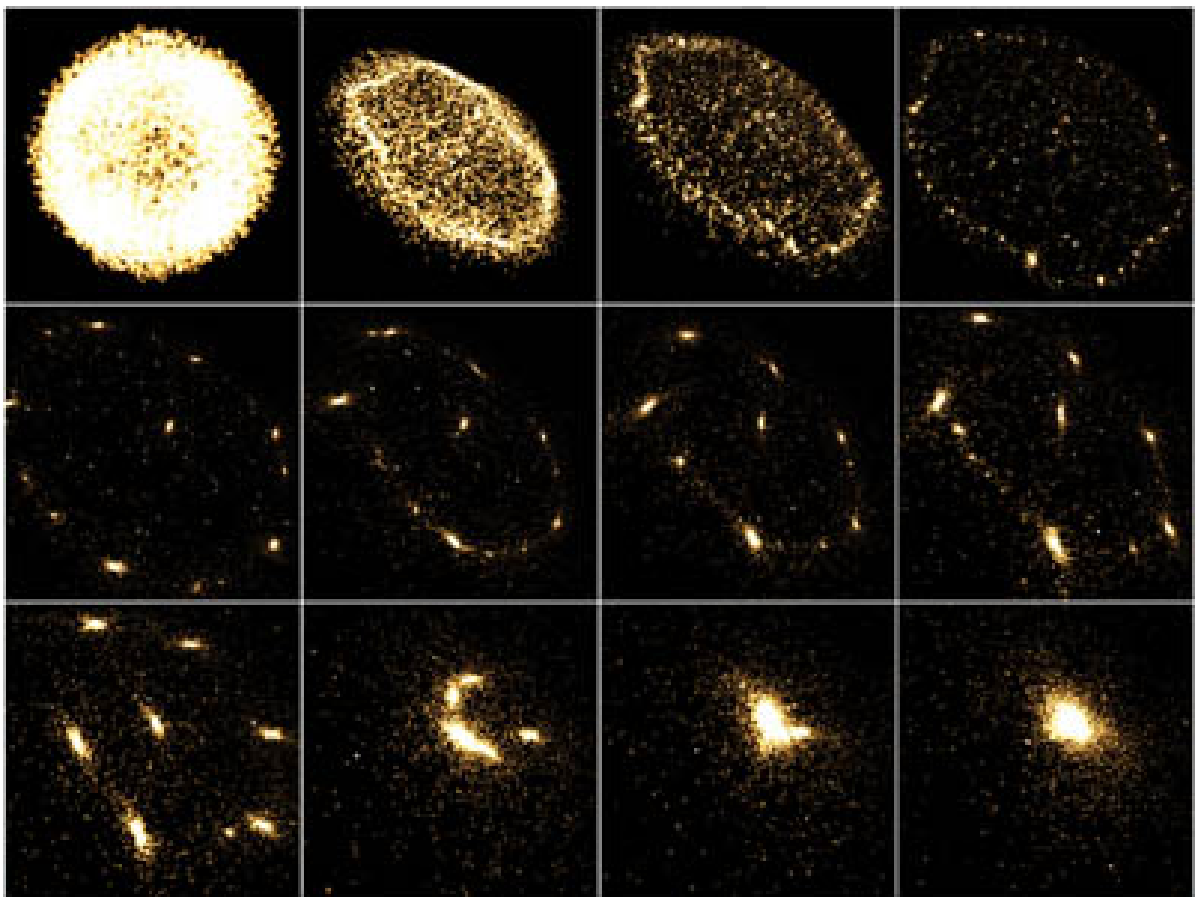


Рис. 1.1: Пример моделирования гравитационной задачи (Source: Nvidia Developer Guide)

Известно, что даже задача трёх тел в гравитационном взаимодействии не имеет общего аналитического решения, и при увеличении числа тел ( $N$ -тел) задача усложняется ещё больше. Однако уже на этом уровне можно выделить два важных аспекта:

1. Наличие члена, обратного к расстоянию в третьей степени, приводит к численным сложностям при моделировании близких взаимодействий.
2. С увеличением числа тел индивидуальные траектории теряют значение, уступая место коллективной динамике всей системы — в данном случае, пучка частиц.

## 2 Замена гравитационного взаимодействия на электрическое

Поскольку в задаче рассматриваются заряженные частицы, гравитационное взаимодействие заменяется на кулоновское.

$$\dot{\vec{v}}_i = \frac{1}{4\pi\epsilon_0 m_i} \sum_{j \neq i} \frac{q_i q_j}{||\vec{r}||^3} \vec{r}.$$

Обозначим теперь  $\frac{1}{4\pi\epsilon_0} \sum_{j \neq i} \frac{q_j}{||\vec{r}||^3} \vec{r}$  как  $\vec{E}_s$ . Добавим внешнее электрическое ( $\vec{E}_{ext}$ ) и магнитное ( $\vec{B}_{ext}$ ) поле и получим

$$\dot{\vec{v}}_i = \frac{q_i}{m_i} (\vec{E}_s + \vec{E}_{ext} + [\vec{v}_i, \vec{B}_{ext}]).$$

### 3 Постановка задачи

Таким образом, формулируется следующая задача Коши: заданы начальные положения и скорости частиц, а также известно внешнее по отношению к пучку электромагнитное поле

$$\begin{cases} \dot{\vec{v}}_i = \frac{q_i}{m_i}(\vec{E}_s + \vec{E}_{ext} + [\vec{v}_i, \vec{B}_{ext}]), \\ \dot{\vec{x}}_i = \vec{v}_i, \\ \vec{x}_i(0) = \vec{x}_0, \vec{v}_i(0) = \vec{v}_0, \end{cases}$$

Следует отметить, что рассматривается нерелятивистская постановка задачи, и в таком виде она далее и решается.

## 4 Численное решение уравнений движения

Уравнения движения решаются методом конечных разностей

$$\vec{x}_{t+1} = \vec{x}_t + F(\vec{x}_t, \dots) \Delta t$$

Особое внимание уделяется выбору численной схемы, поскольку на каждом шаге моделирования возникает множество взаимодействий, и даже незначительные численные ошибки могут существенно повлиять на результат.

В рамках тестирования применялась явная схема Эйлера, однако учитывались её ограничения.

Среди альтернативных схем рассматривается метод Бориса — численная схема второго порядка, обладающая временной обратимостью. Однако он вносит в систему дополнительные силы, отсутствующие в исходной физической постановке, и на релятивистских скоростях эти силы становятся заметными и искажают результат [1].



## 5 Трудности в расчёте собственного поля

Наибольшие сложности связаны не с решением уравнений движения, а с вычислением собственного поля пучка. Прямое моделирование взаимодействий по схеме “каждый с каждым” (Particle-to-Particle) требует вычислительных затрат порядка  $O(N^2)$ , что делает этот подход непрактичным даже при относительно малом числе частиц. Для иллюстрации на рис. 5.1 изображена система из 256 частиц, но в реальных задачах их число может достигать  $10^{15}$ .

Bodies: 256  
Time: 0.00600s

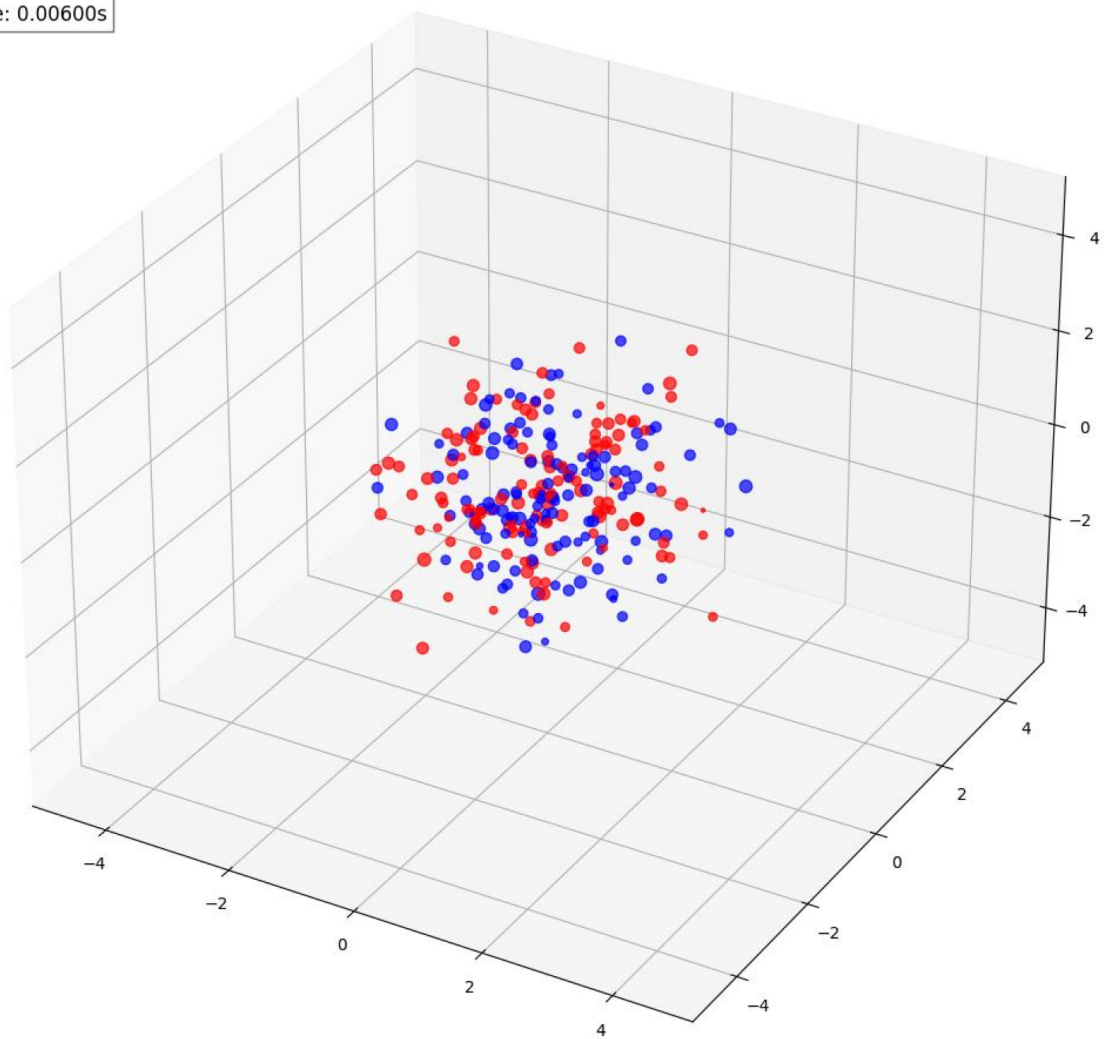


Рис. 5.1: Система из 256 частиц с разным зарядом

## 6 Метод крупных частиц

Одним из способов снижения вычислительной сложности является объединение множества частиц в макрочастицы (Метод крупных частиц).

Для каждой  $i$ -й макрочастицы пару

$$(\vec{r}_i, \vec{v}_i)$$

можно взять, например, как среднее по микрочастицам.

Тогда, взяв макрочастицу как облако с радиусом  $R$  поле вычисляется как

$$(\vec{E}_s(j \rightarrow i) = \frac{1}{4\pi\epsilon_0} \frac{q_j}{R^3} \vec{r}, \quad ||\vec{r}|| < R)$$

Кроме того, этот подход позволяет смягчить проблему моделирования близких взаимодействий, так как взаимодействие происходит только при пересечении облаков, то есть когда расстояние между ними меньше  $R$ . Это позволяет сократить число взаимодействий с  $10^{15}$  до, например,  $10^9$ , что всё ещё является вычислительно затратным. Поэтому на практике чаще применяется другой метод — Particle-In-Cell (PIC).

## 7 Метод Particle-In-Cell

Метод PIC исключает необходимость моделирования парных взаимодействий. Вместо этого рассчитывается распределение заряда на сетке (рис. 7.1), и далее решается краевая задача для уравнения Пуассона с целью нахождения электрического поля.

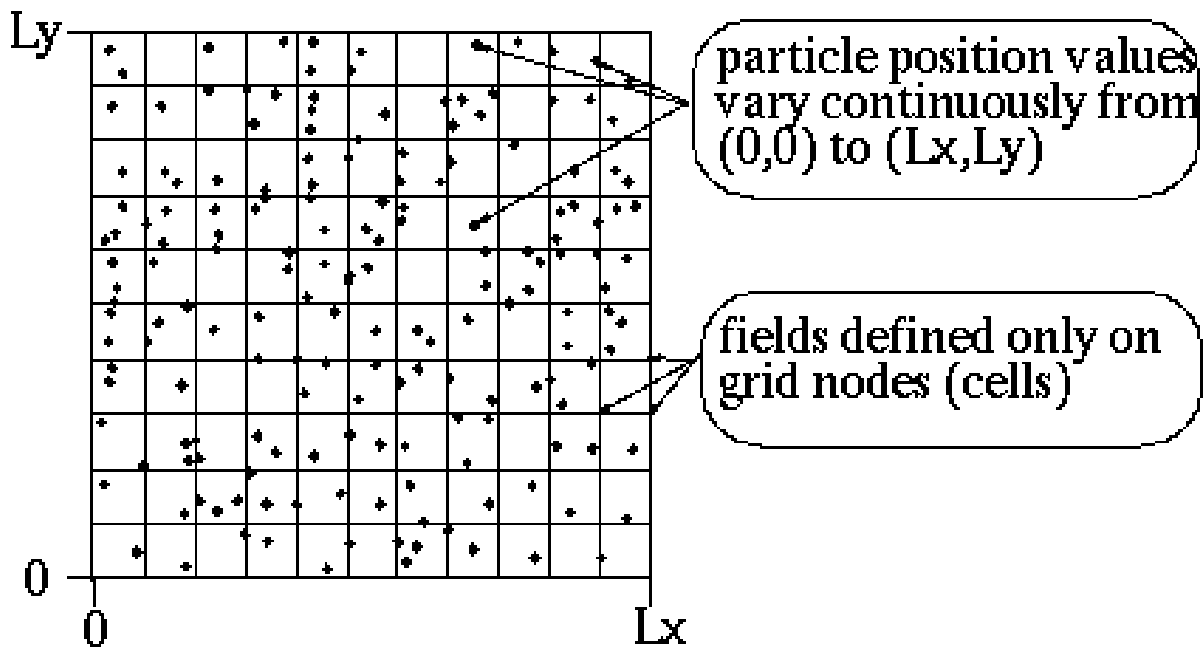


Рис. 7.1: PIC

Именно этот метод чаще всего используется в прикладных задачах.

## 8 Ускорение расчётов с использованием графических процессоров

Даже при использовании макрочастиц и метода PIC последовательное решение задачи остаётся неприемлемо медленным. Поэтому в современных вычислениях активно применяются графические ускорители (GPU). Они позволяют эффективно распараллелить как решение уравнений движения, так и вычисление собственного поля пучка.

Среди готовых решений:

1. <https://github.com/GabrielSCabrera/nBody> — предназначен для моделирования задачи N-тел с электрическим взаимодействием. Подходит скорее для учебных целей, но демонстрирует общий подход.
2. <https://developer.nvidia.com/gpugems/gpugems3/part-v-physics-simulation/chapter-31-fast-n-body-simulation-cuda> — ориентирован на гравитационные взаимодействия, но полезен с точки зрения изучения архитектуры GPU.
3. <https://developer.nvidia.com/gpugems/gpugems3/part-v-physics-simulation/chapter-31-fast-n-body-simulation-cuda> — разрабатывается в CERN для задач ускорительной физики. Хотя пакет слабо поддерживается, его структура может оказаться полезной для ряда задач.

## 9 Архитектура собственной реализации

В рамках практической части проекта была разработана архитектура (рис. 9.1), сочетающая удобство настройки с возможностью высокопроизводительных вычислений. Python-обвязка обеспечивает гибкость при задании начальных условий и параметров моделирования. Хотя подобные задачи можно реализовывать и средствами CUDA на C/C++, использование Python значительно упрощает отладку и запуск, особенно при работе на вычислительных кластерах.

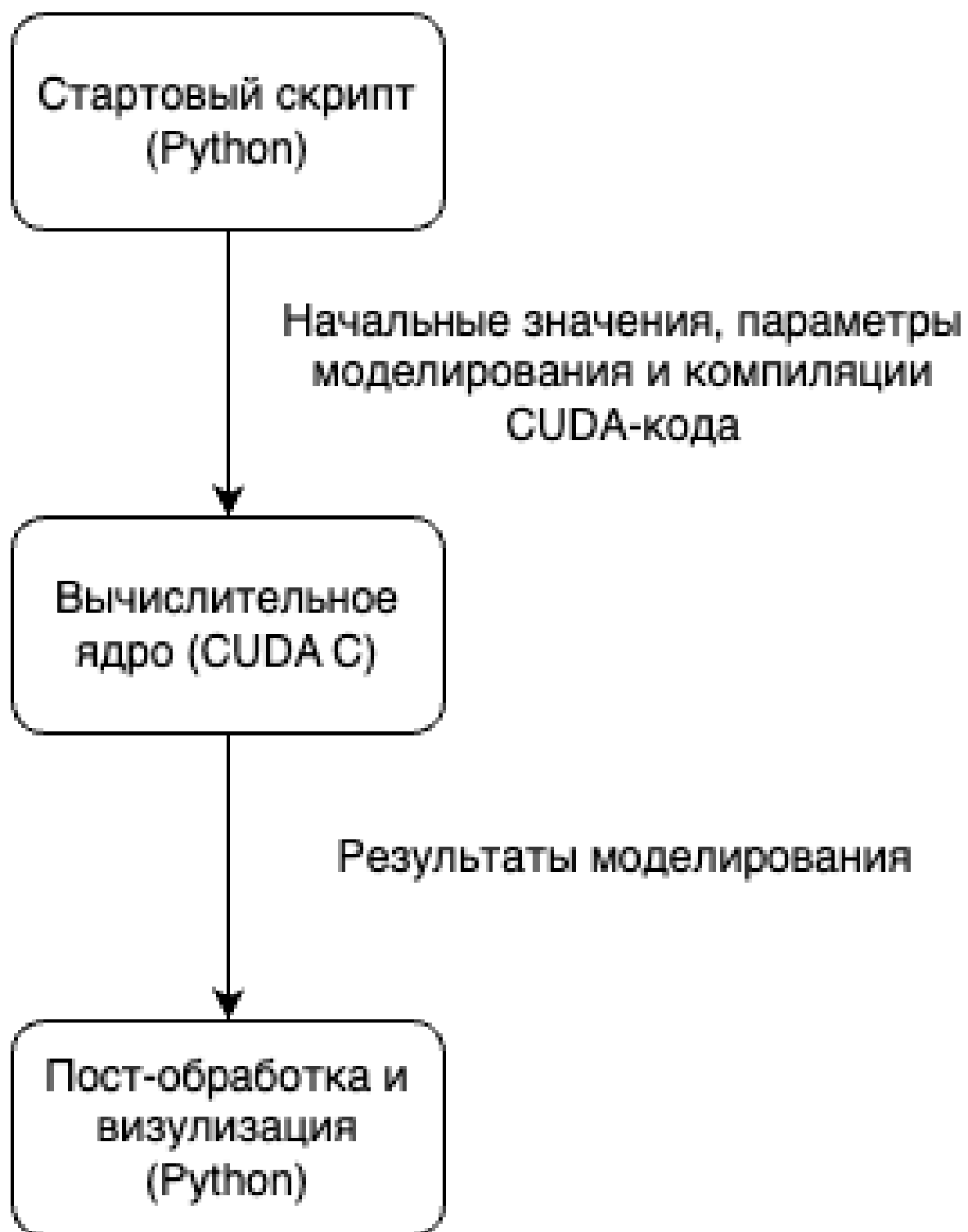


Рис. 9.1: Схема программного решения

## 10 Результаты моделирования

Реализация позволяет моделировать динамику пучка заряженных частиц в электромагнитном поле. Легко моделируются пучки до 200 тыс. частиц и приемлемо – до 500 тыс.

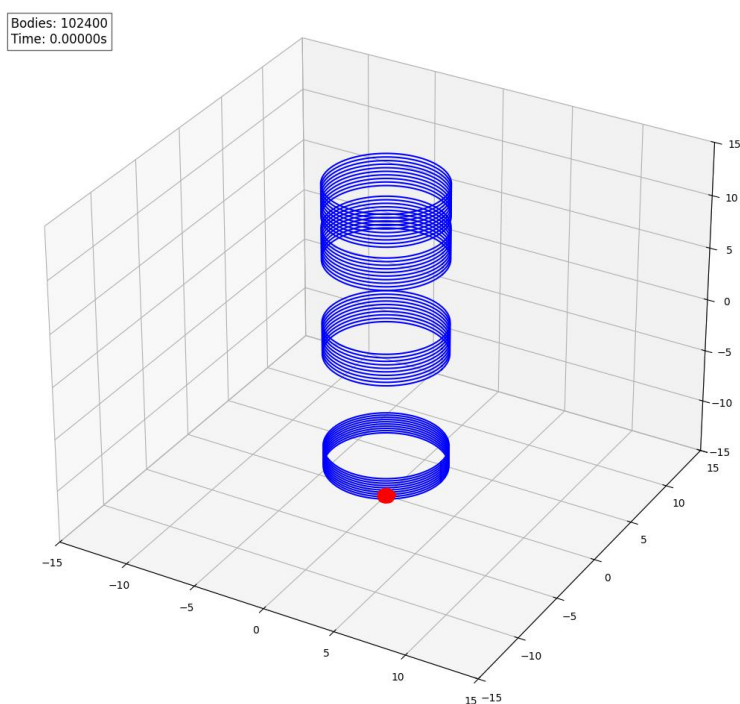


Рис. 10.1: Пучок 100 тыс. частиц на старте симуляции, вектор скорости направлен в катушку



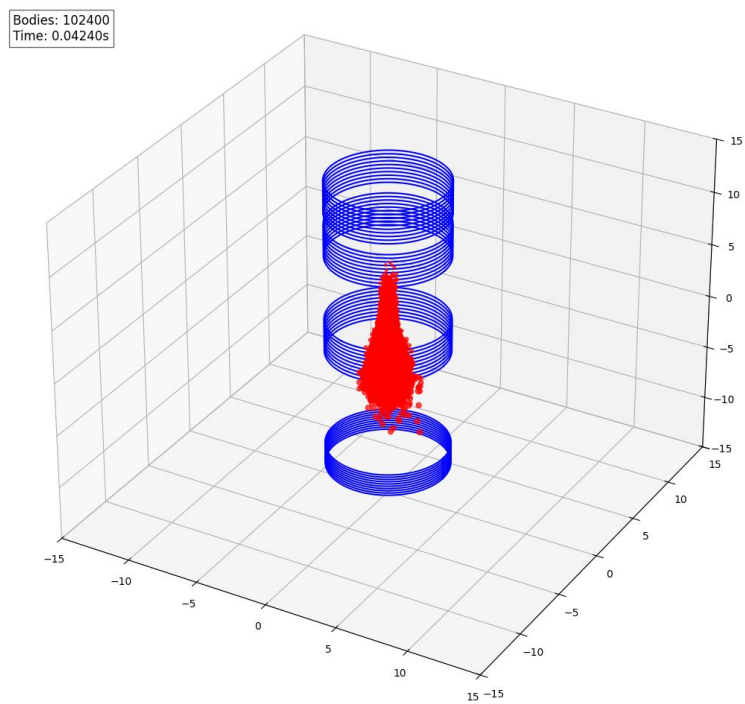


Рис. 10.2: Фокусировка пучка

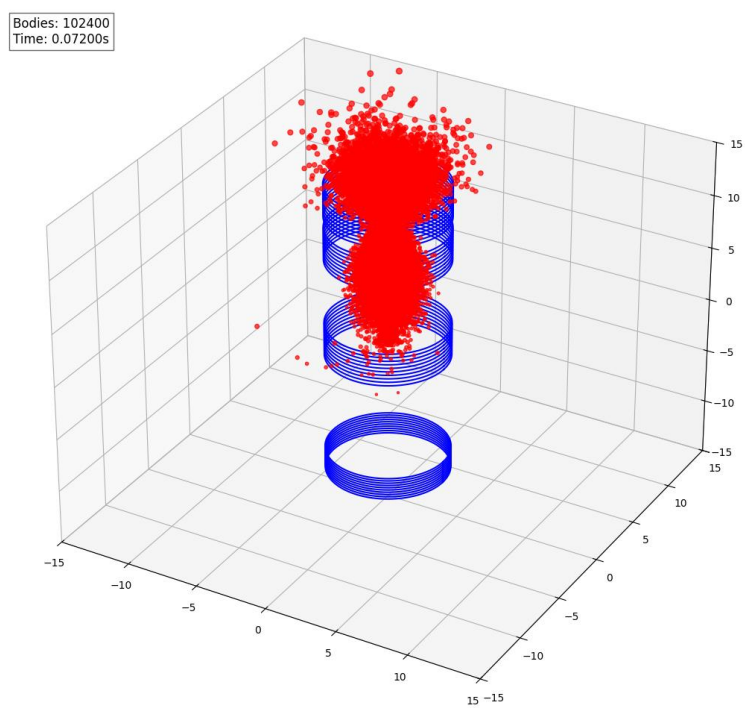


Рис. 10.3: Разлет на выходе из катушки

## 11 Дальнейшая работа

Отдельной задачей остаётся учёт геометрии экспериментальной установки. Даже при её упрощении и грубой дискретизации проблема столкновений частиц становится вычислительно трудоёмкой, особенно при прямом переборе всех возможных пар.

Наконец, важным аспектом является правильное использование различных уровней памяти в архитектуре видеокарт NVIDIA. На рис. 11.1 приведена иерархия памяти GPU [2], практически все типы которой доступны пользователю. От выбора типа памяти напрямую зависит производительность: при неправильной организации данных ускорение может быть полностью нивелировано.

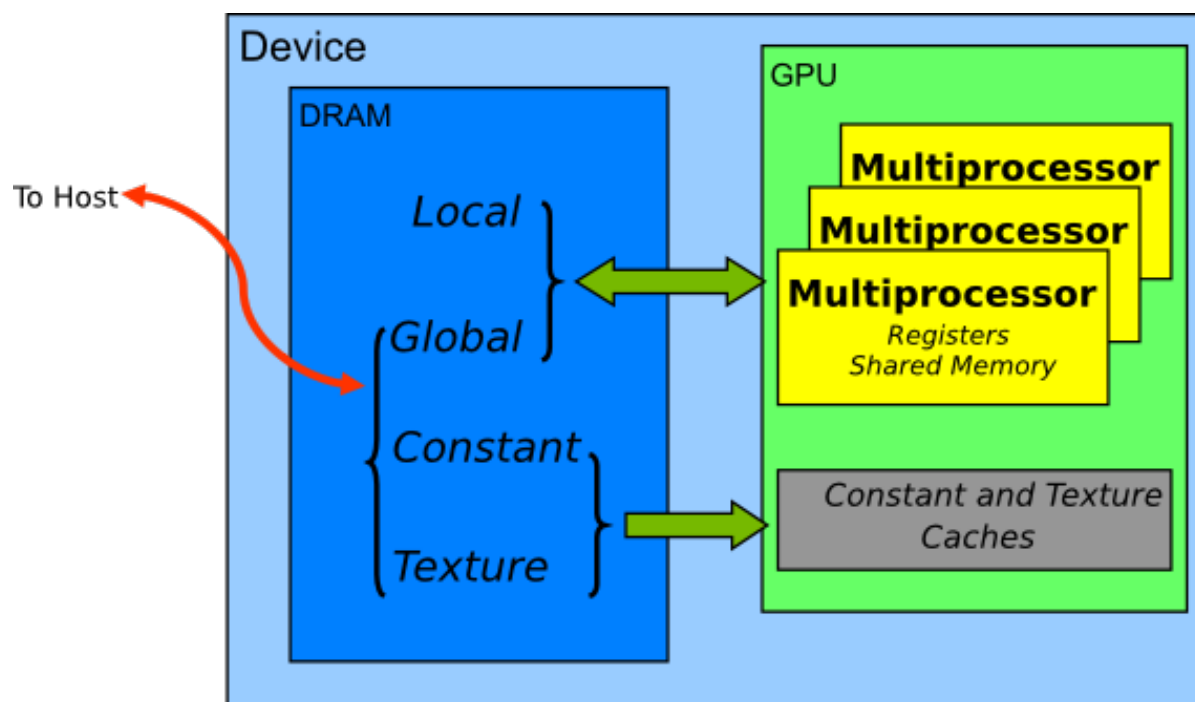


Рис. 11.1: Архитектура памяти GPU

## 12 Заключение

Рассмотренная задача моделирования пучков заряженных частиц в электромагнитных полях требует комплексного подхода, сочетающего точные численные методы, эффективную архитектуру программного обеспечения и использование современных вычислительных ресурсов. Несмотря на ряд нерешённых проблем, разработанная архитектура демонстрирует потенциал для дальнейшего развития и внедрения в более сложные модели, включая учёт геометрии установки и использование PIC-методов.

## Список литературы

1. Vay J.-L. Simulation of beams or plasmas crossing at relativistic velocity // Physics of Plasmas. 2008. T. 15, № 5. С. 056701.
2. 1. Introduction — CUDA C++ Programming Guide.  
<https://docs.nvidia.com/cuda/cuda-c-programming-guide/>.