МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени М.В. ЛОМОНОСОВА

ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

Курсовая работа по дисциплине
“параллельное программирование”

Моделирование процессов в высокотемпературной лазерной плазме и анализ распределения прошедших через заданное сечение частиц

Выполнил:
студент 2 курса группы 216
Прокудин Владислав
Валерьевич “19” мая 2016

Научный руководитель:
Савельев-Трофимов Андрей
Борисович “19” мая 2016

Москва, 2016 г.

-

Аннотация

В ходе выполнения курсовой работы было получено базовое представление о плазме и процессах, происходящих в ней. Использование этих представлений позволило изучить метод моделирования, известный как Метод Крупных Частиц в ячейках. Был изучен программный код «Мандор», реализующий данный метод на практике. Собранные о коде программы знания позволили написать дополнение для неё, выдающее распределение по энергии частиц, прошедших некоторое заданное сечение.

Содержание

1 Аннотация……2
2 Содержание……..3
3 Плазма……...……4
3.1 Основные понятия…….4
3.2 Параметры плазмы……5
4 Метод крупных частиц в ячейках……...7
4.1 Суть метода……....7
4.2 Программная реализация «mandor»……11

5 Обработка данных……...14
5.1 Подход……...14
5.2 Параллелизация……....15
5.3 Результаты……16

6 Выводы……18

Литература……19

-

1 Плазма

1.1 Основные понятия

Плазма (греч. «вылепленное», «оформленное») — частично или полностью ионизированный газ, образованный из нейтральных атомов (или молекул) и заряженных частиц (ионов и электронов). Отличительной особенностью плазмы является её квазинейтральность.

С ионизованным газом приходится встречаться практически всюду. Он присутствует в верхних слоях атмосферы Земли (ионосфере). Число заряженных на высотах 300-500 км достигает максимума. Именно этот слой ионосферы, называемый Е-слоем, обеспечивает распространение электромагнитных волн вокруг Земли и устойчивую радиосвязь на Земле.

Другой распространенный пример — это плазма звездных атмосфер. Вещество в большинстве космических объектов находится в ионизованном состоянии, т. е. в состоянии плазмы. В плазме звезд, в частности Солнца, происходят реакции синтеза легких элементов, так называемые термоядерные реакции, обеспечивающие огромное выделение энергии и нагрев плазмы. В наше время ученые многих стран мира изучают возможности создания подобной высокотемпературной плазмы в земных условиях, поставив перед собой задачу осуществления управляемого термоядерного синтеза и обеспечения человечества неисчерпаемым запасом энергии. Именно эти исследования стимулируют развитие физики плазмы.

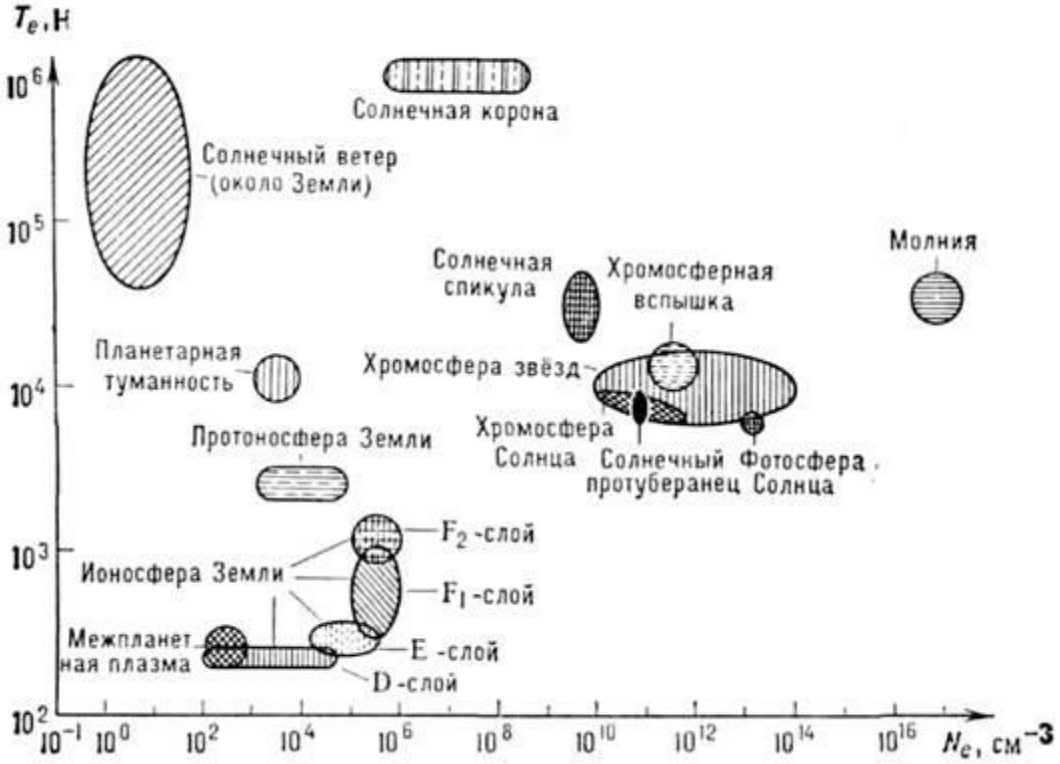


Рис 1. Низкотемпературная плазма в природе

Примером распространенной в природе плазмы является также плазма газового разряда. Интенсивные исследования плазмы газового разряда связаны с потребностями развития классической и квантовой электроники, для которых газоразрядные приборы играют большую роль.

Особый интерес представляет твердотельная плазма — электронная плазма металлов и электронно-дырочная плазма полупроводников.

Перечисленный ряд можно продолжить практически неограниченно, однако приведенных примеров достаточно для того, чтобы убедиться в чрезвычайно широком распространении плазмы в природе и важности изучения ее свойств.

1.2 Параметры плазмы

Для описания плазмы прежде всего нужно знать концентрацию частиц разного сорта N_α, где индекс α означает сорт частиц. Все величины, относящиеся к электронам плазмы, обычно обозначают индексом e, к ионам (дыркам) — индексом i, а к нейтральным частицам — индексом n. Отдельно задают концентрацию ионов каждого сорта, если плазма состоит из разных сортов. Нейтральные частицы (атомы) могут быть и в возбужденном состоянии. Этот аспект довольно сложно учесть, и обычно он мало на что влияет.

Другой удобный параметр — отношение концентрации электронов к концентрации нейтральных частиц, или степень ионизации $r = \frac{N_e}{N_n}$. По степени ионизации плазму обычно подразделяют на слабоионизованную ($r < 10^{-2} \div 10^{-3}$) и полностью ионизованную ($r \to \infty$), т. е плазму, состоящую только из заряженных частиц.

Важно помнить основные значения параметров: $|e| = 4{,}8 \cdot 10^{-10}$ СГСЭ, $m_e = 9{,}1 \cdot 10^{-27}$ г, $e_i = Ze$, $m_i = A \cdot 1{,}66 \cdot 10^{-24}$ г. Эти значения применимы к газовой плазме. В металлах же $m_e^{\text{металл}} \approx m_e$, а в полупроводниках, как правило, $m_e^{\text{полупр}} = (0{,}01 \div 0.1) \cdot m_e$.

Частицы плазмы находятся в состоянии хаотического теплового движения. Для характеристики этого движения вводят понятие температуры плазмы в целом T или отдельных ее компонент T_α. Температура плазмы вводится в предположении, что плазма в целом находится в состоянии термодинамического равновесия, а функции распределения частиц всех сортов по импульсам (скоростям) являются максвелловскими с одной и той же температурой T (изотермическая плазма). Гораздо чаще в плазме имеется частичное термодинамическое равновесие, когда отдельные ее компоненты имеют максвелловские распределения по скоростям с различными температурами:

$$f_{M\alpha} = \frac{N_\alpha}{(2\pi m_\alpha T_\alpha)^{3/2}} \exp\left(-\frac{p_a^2}{2m_\alpha T_\alpha}\right) \quad (1.2.1)$$

Такая плазма является неизотермической. В случае максвелловской функции распределения частиц температура T_α характеризует среднюю кинетическую энергию теплового движения частиц данного сорта:

$$\frac{3}{2} x T_a = \frac{\overline{m_\alpha v_\alpha^2}}{2} \quad (1.2.2)$$

где $x = 1{,}38 \cdot 10^{-16} \frac{\text{эрг}}{\text{град}}$ —постоянная Больцмана; T_α —температура частиц сорта α, К; v_α — скорость хаотического теплового движения частиц сорта α, см/с (черта сверху означает усреднение по всем частицам данного сорта α). Часто пользуются понятием температуры плазмы и тогда, когда функция распределения частиц отличается от максвелловской, понимая под температурой T_α величину, определенную соотношением (1.2.2).

-

О максвелловском распределении частиц можно говорить лишь при достаточно высоких температурах, когда отсутствует фермиевское вырождение, обусловленное принципом Паули. Явление фермиевского вырождения возможно для частиц с полуцелым спином (электроны, дырки и ионы атома водорода) и становится существенным, когда энергия Ферми превышает тепловую.

Различные примеры значений обозначенных параметров на известных примерах: в F-слое ионосферы $N_e \approx N_i \approx 10^6$ см$^{-3}, N_n \lesssim 10^{10}$ см$^{-3}, r \gtrsim 10^{-4}$, а температура плазмы оказывается довольно высокой, порядка $T \approx (3 \div 5) \cdot 10^3$ К. На больших высотах, превышающих радиус Земли, в межпланетной плазме концентрация заряженных частиц колеблется в пределах $10^{-2} \approx N_e \approx N_i \lesssim 10^{-3}$ см$^{-3}, N_n \ll N_e$, т. е. плазма практически полностью ионизована, а температура $T \approx 10^4$ К. Концентрация N и температура T плазмы звезд меняются в очень широких пределах: концентрация от $10^2 \div 10^3$ см$^{-3}$ до $10^{22} \div 10^{26}$ см$^{-3}$ и выше, температура от $10^4 \div 10^5$ К до $10^9 \div 10^{10}$ К, при этом плазма полностью ионизована. Так, например, в солнечной короне$10^4 \div 10^8 \approx N_e \approx N_i$см$^{-3}$, а $T \approx 10^6 \div 10^8$ К. В установках для получения термоядерного синтеза, поскольку реакции слияния ядер пороговые, температура плазмы должна превышать $T \gtrsim 10^8$ К, концентрация же заряженных частиц в зависимости от методов нагрева и удержания плазмы составляет либо $N_e \approx N_i \approx 10^{14} \div 10^{15}$ см$^{-3}$, либо $N_e \approx N_i \approx 10^{22} \div 10^{23}$ см$^{-3}$.

Плазму газового разряда в противоположность термоядерной часто называют низкотемпературной. Ее температура обычно не превышает $10^4 \div 10^5$ К, а концентрация заряженных частиц $N_e \approx N_i \approx 10^8 \div 10^{15}$ см$^{-3}$,, причем такая плазма практически всегда слабоионизована, так как $N_n \approx 10^{12} \div 10^{17}$ см$^{-3}$.

2 Метод крупных частиц в ячейках

2.1 Суть метода

Метод частиц в ячейках (**particle-in-cell -** PIC) относится к методике, используемой для решения определенного класса дифференциальных уравнений в частных производных. В этом способе отдельные частицы в лагранжевой рамке отслеживаются в непрерывном фазовом пространстве, в то время как параметры распределения, такие как плотность и токи вычисляются одновременно на эйлеровских (стационарных) точках сетки.
Методы PIC уже использовались в 1955 году еще до появления первых fortran-компиляторов. Метод приобрел популярность для моделирования плазмы в конце 1950-х годов и в начале 1960-ых. В физике плазмы метод сводится к следующему: траектории заряженных частиц в самосогласованных электромагнитных (или электростатических полях), вычисляются на фиксированной сетке.

Основы методики моделирования PIC плазмы
Внутри плазмы исследуются системы различных видов (электронов, ионов, нейтральных молекул, частиц пыли и т.д.). Система уравнений, связанных с PIC кодом, в том числе сила Лоренца как уравнение движения, решается в так называемом pusher-e (particle mover) (далее: «pm») и уравнения Максвелла, определяющие электрические и магнитные поля, рассчитываются в (поле) solver-e (далее: «solver»).

Крупные частицы
Реальные изучаемые системы часто чрезвычайно большие с точки зрения числа частиц, которые они содержат. Для того чтобы сделать моделирование эффективным или вообще возможным, используются так называемые супер-частицы или крупные частицы. Супер-частица представляет собой вычислительную частицу, являющуюся множеством реальных частиц; она может содержать миллионы электронов или ионов в случае моделирования плазмы. Допускается отмасштабировать число частиц, так как сила Лоренца зависит только от заряда к массе, так что супер-частица будет следовать той же траектории, что и реальная частица.
Число реальных частиц, соответствующих супер-частице должны быть выбраны таким образом, чтобы достаточное количество статистических данных могло быть собрано о движении частиц. Если существует значительная разница между плотностью различных видов частиц в системе (между ионами и нейтральными частицами, например), тогда может быть использован отдельный род супер-частиц для каждого вида.

The particle mover (pm)
Даже с использованием супер-частиц, количество моделируемых частиц, как правило, очень велико ($> 10^5$), и часто pm является наиболее трудоемкой частью PIC, так как он должен быть сделан для каждой частицы в отдельности. Таким образом, pm должен иметь высокую точность и скорость. Поэтому много усилий тратится на оптимизацию различных схем.
Схемы, используемые для движителя частиц можно разделить на два типа: скрытые и явные solver-ы. В то время как неявный solver-ы (например, неявная схема Эйлера) вычисляет скорость частиц от уже обновленных полей, явный solver использует только старую силу из предыдущего шага времени, и, следовательно, этот метод проще и быстрее, но требует

-

меньшего временного шага. Две наиболее часто используемые схемы — это метод «чехарда» и «схема Бориса», которые являются явными solver-ами.

Для моделирования плазмы, метод «чехарда» принимает следующий вид:

$$\frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\Delta t} = \boldsymbol{v}_{k+\frac{1}{2}}$$

$$\frac{\boldsymbol{v}_{k+\frac{1}{2}} - \boldsymbol{v}_{k-\frac{1}{2}}}{\Delta t} = \frac{q}{m}(\mathbf{E}_k + \frac{\boldsymbol{v}_{k+\frac{1}{2}} + \boldsymbol{v}_{k-\frac{1}{2}}}{2} \times \mathbf{B}_k)$$

где индекс k относится к «старым» величинам из предыдущего шага по времени, к + 1 уточненным величинам от следующего временного шага (т.е. $\mathrm{t}_{k+1} = t_k + \Delta t$, и скорости вычисляются промежуточные обычное время шаги t_k .

Для сравнения, уравнения «схемы Бориса». являются:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x_k} + \Delta t \boldsymbol{v}_{k+\frac{1}{2}}$$

$$\boldsymbol{v}_{k+\frac{1}{2}} = \boldsymbol{u}' + q'\boldsymbol{E}_k$$

Где:

$$\boldsymbol{u}' = \boldsymbol{u} + (\boldsymbol{u} + (\boldsymbol{u} \times \boldsymbol{h})) \times \boldsymbol{s}$$

$$\boldsymbol{u} = \boldsymbol{v}_{k-\frac{1}{2}} + q'\boldsymbol{E}_k$$

$$\boldsymbol{h} = q'\boldsymbol{B}_k$$

$$\boldsymbol{s} = 2\boldsymbol{h}/(1 + h^2)$$

$$q' = \Delta t \times (\frac{q}{2m})$$

Из-за своей точности для большого количества шагов по времени, «алгоритм Бориса» является стандартом де-факто для заряженной частицы. Недавно было показано, что такая точность «алгоритма Бориса» связана с тем, что этот алгоритм сохраняет объем фазового пространства, практически во всех случаях. Для этого алгоритма также имеет место глобальная оценка ошибок по энергии, что делает его эффективным для динамики плазмы на большом числе масштабов.

Solver

Наиболее часто используемые методы решения уравнений Максвелла (или в более общем плане, дифференциальные уравнения в частных производных УРЧП) принадлежат к одному из следующих трех видов:

- Методы конечных разностей (МКР)
- Методы конечных элементов (МКЭ)
- Спектральные методы

В МКР, непрерывный домен заменяется дискретной сеткой точек, на которых рассчитываются электрические и магнитные поля. Производные затем аппроксимируется с приращениями между соседними значениями в узлах сетки и, таким образом, УРЧП превращается в систему алгебраических уравнений.

С помощью метода конечных элементов, непрерывный домен делится на дискретную сетку элементов. В МКЭ рассматриваются задача на собственные значения и (первоначально) решение рассчитывается с использованием базисных функций, которые локализованы в

каждом элементе. Конечное решение затем получают путем оптимизации, пока требуемая точность не будет достигнута.

Аналогично, спектральные методы, такие как быстрое преобразование Фурье (FFT), трансформируют УРЧП в задачу на собственные значения, но на в этом случае базисные функции высокого порядка, и они определены глобально на всей области. Домен сам не дискретизируется, он остается непрерывным. Опять же, решение найдено путем введения базисных функций в уравнения на собственные значения, а затем оптимизированы для определения наилучших значений исходных параметров испытаний.

Частицы и поля взвешивания

Название "частицы в ячейке" следует из того, что в плазме макровеличины (плотность, плотность тока и т.д.) присваиваются частицам моделирования (т.е. происходит «взвешивание частиц»). Частицы могут быть расположены в любом месте на непрерывной области, но макровеличины вычисляются только по точкам сетки, так же, как поля. При получении макровеличин, предполагается, что частицы имеют заданную «форму».

$$S(\boldsymbol{x} - \boldsymbol{X}),$$

Где **x** является координатой частицы и **X** - точки наблюдения. Пожалуй, самый простой и наиболее часто используемый выбор для функции формы является так называемое облако в ячейке (CIC) схема взвешивания первого порядка (линейного). Какой бы ни была схема, функция формы должна удовлетворять следующим условиям: изотропии пространства, сохранению заряда и повышению точности (сходимости) для членов высшего порядка.

Поля, полученные из поля solver-ом, определяются только по точкам сетки, и не могут быть использованы непосредственно в pm для вычисления силы, действующей на частицы, но должны быть интерполированы через поле взвешивания:

$$\boldsymbol{E}(\boldsymbol{x}) = \sum_i \boldsymbol{E}_i S(\boldsymbol{x}_i - \boldsymbol{x}),$$

где индекс i маркирует точку сетки. Для того, чтобы гарантировать, что силы, действующие на частицы, являются самосогласованными, используют способ вычисления макровеличин с учётом позиций частиц по точкам сетки и интерполяцией полей из точек сетки на позиции частиц. Прежде всего, схема интерполяции поля должна сохранить импульс. Это может быть достигнуто путем выбора той же самой схемы взвешивания для частиц и полей, а также обеспечением надлежащего пространственной симметрии (т.е. без самопересечений силы и выполняя закона реакции) поля solver-ом.

Столкновения

Поскольку внутри клетки поле, создаваемое частицей, должно уменьшаться при удалении от частицы, то силы между частицами внутри клетки не будут учтены надлежащим образом. Это может быть скомпенсировано кулоновскими столкновениями между заряженными частицами. Моделирование взаимодействий для каждой пары большой системы будет вычислительно слишком дорого, поэтому вместо этого были разработаны несколько методов Монте-Карло. Широко используемый метод «модель двойных столкновений», в которой частицы группируются в соответствии с их клетками, то эти частицы образуют пары случайным образом, и, наконец, эти пары сталкиваются.

-

В реальной плазме, многие другие реакции могут играть определенную роль, начиная от упругих столкновений, таких как столкновения между заряженными и нейтральными частицами, более неупругих столкновений, такие как электрон-нейтральных столкновений ионизации, химических реакций; каждая из них требует отдельного рассмотрения. Большую часть обработки моделей столкновений заряженных нейтральных столкновений можно использовать либо прямой схемой Монте-Карло, в котором все частицы несут информацию об их вероятности столкновения или схемы нуль-столкновений, которая не анализирует все частицы, но вместо этого использует максимальную вероятность столкновений для каждой заряженной частицы.

Точность и стабильность условий

Как и в каждом методе моделирования, так и в PIC, шаг по времени, и размер сетки должны быть хорошо подобраны, так что время и масштаб длины представляющего интерес явления были надлежащим образом разрешены в задаче. Кроме того, шаг по времени и размер сетки оказывают влияние на скорость и точность программы.

Для электростатического моделирования плазмы с использованием явной временной схемы интеграции (например, «чехарда», которая чаще всего используется), есть два важных условия, касающиеся размера сетки Δxи шага по времени Δt, которые должны быть выполнены для того, чтобы обеспечить стабильность работы решение:

$$\Delta x < 3.4\lambda_D$$
$$\Delta t \leq 2\omega_{pe}^{-1}$$

которые могут быть получены с учетом гармонических колебаний одномерной не намагниченной плазмы. Последние условия требуется строго, но практические соображения, связанные с энергосбережением, предполагают использование чуть менее строгих ограничений, где коэффициент 2 заменяется числом на один порядок меньше. Использование условия $\Delta t \leq 0{,}1\omega_{pe}^{-1}$ является общепринятым. Не удивительно, что естественный масштаб времени в плазме задается обратной частотой ω_{pe}^{-1}и масштаб длины по дебаевской длины λ_D.

Для явного электромагнитного моделирования плазмы, шаг по времени должен также удовлетворять условию:

$\Delta t < \frac{\Delta x}{c}$, где $\Delta x \sim \lambda_D$, и c - это скорость света.

2.2 Программная реализация «mandor»

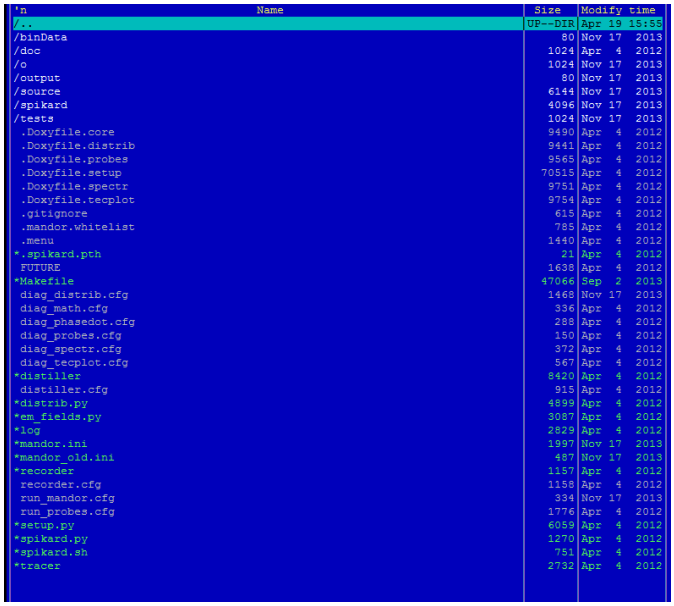


Рис 2.

Перейдем теперь к описанию программы, реализующей вышеописанный метод PIC-кода для моделирования процессов, происходящих при прохождении лазерного излучения высоких частот через среду.

В программной реализации «mandor» находятся следующие основные исполняемые элементы:

1 Исполняемое ядро: core.out

1.1 Численный код, запускающий процесс моделирования и сохранения двоичных данных для анализа.

1.2 Описание алгоритма, запускаемого в ядре:

1.2.1 Вход в параллельный режим

1.2.2 Инициализация профилировщика

1.2.3 Настройка всех модулей

1.2.4 Общее время этапа работ вычисляется для цикла

1.2.5 для (INT т = 1; т <= totalSteps && stopFlag;! T ++)

1.2.5.1 запуск кода, работающего с плазмой

1.2.5.2 Считывание данных из внешнего файла (работает как почтовый ящик)

1.2.5.3 $H^{n-\frac{1}{2}} \to H^n$

1.2.5.4 Плотность энергии поля при $T = N \cdot \tau$

1.2.5.5 Испытание закона Гаусса начало (вызов профилировщика находится внутри)

1.2.5.6 Плотность энергии: получает энергию плазмы Termal

1.2.5.7 Tecplot: диагностическая расстановка check-point-ов

1.2.5.8 Спектральная плотность энергии диагностики

1.2.5.9 Сохраняет локальные энергии в буфер

1.2.5.10 $H^n \to H^{n+\frac{1}{2}}$

1.2.5.11 $E^n \to E^{n+1}$

1.2.5.12 Получение полной картины токов

1.2.5.13Испытание закона Гаусса завершение

1.2.5.14Tecplot: диагностическая расстановка check-point-ов

1.2.5.15Обновления время после успешного шага по времени

1.2.5.16Получение данных, отправленных ранее

1.2.5.17Удаление файла TMP, чтобы сигнализировать конец выполнения программы

-

1.3 Для инициализации программы предварительно необходимо настроить все начальные условия, а также параметры, в файле mandor.ini. С помощью команды вида “sbatch -n 64 -p gpu impi ./o/setup.out mandor_old.ini create” эти значения «закрепляются». Сам файл содержит следующие поля:

1.3.1 Индексы (размеры) сетки: $i_{min}, j_{min}, k_{min}, i_{max}, j_{max}, k_{max}$.

1.3.2 Размеры домена L_x (= 5.000000000000000e+01 μm), L_y, L_z.

1.3.3 Шаг по времени: τ (= 3.000000000000000e-03 fs)

1.3.4 Граничные условия: bound_xmin, bound_xmax, bound_ymin, bound_ymax, bound_zmin bound_zmax

1.3.4.1 Они могут принимать следующие значения: BC_PERIODIC, BC_MIRROR BC_OPEN (first order Mur's condition).

1.3.5 Основные параметры лазера (с приемлемыми значениями):

1.3.6 r0 = 1.000000000000000e+00 micron $r_0 = \lambda$ = длина волны

1.3.7 ne_crit = 1.11485839911650e+21 cm^{-3} критическое значение ne для частоты лазера

1.3.8 v_0 = 1.000000000000000e+00 c (!) Скорость света

1.3.9 t_0 = 3.335640951981521e-15 sec $t_0 = \lambda/c$

1.3.10 ω_0 = 1.883651567308853e+15 sec^-1 $\omega_0 = 2\pi/t_0$

1.3.11 A_0 = 3.210709989316420e+10 volt/cm $A_0 = mc\omega_0/e$

1.3.12 E_0 = 5.110003656321110e+09 volt/cm $E_0 = A_0/2\pi$

1.3.13 n_0 = 2.823969314801762e+19 cm^-3 $n_0 = m\omega_0^2/(16\pi^3e^2)$

1.3.14 ρ_0 = 2.823969314801762e+19 e*(cm^-3) $\rho_0 = en_0$

1.3.15 ω_pe(n_0) = 2.997924579999803e+14 sec^-1 ω_pe(n_0) = $\sqrt{4\pi n_0 e^2/m}$

1.3.16 Цель для плазмы:

Форма, пример задания:

shape = (half_space(["20.000000000000000e+00 micron", "1.000000000000000e+00 micron", "0.000000000000000e+00 micron"], [-1, 0, 0])

* half_space(["30.000000000000000e+00 micron", "1.000000000000000e+00 micron", "0.000000000000000e+00 micron"], [1, 0, 0]))

1.3.17 (sampling = (3, 3, 3)*30)

1.3.18 Функция плотности, пример задания:

```
rho_func = Func(Param(rho_max='1.114858399116498e+21 e*(cm^-3)'),
          Return('0.25*rho_max'))
```

1.3.19 Отношение заряда к массе для Крупных Частиц, пример задания:

```
qDivM    = -1.0
```

1.3.20 support_code = (дополнительные параметры)

1.3.21 support_code_reserved_names = [] (дополнительные имена)

1.4 Для запуска кода на выполнение нужно использовать (когда решение уже собрано при помощи Makefile) команду вида: sbatch -n 64 -p test impi ./core.out

2 Диагностика: distribution

2.1 Диагностика функции распределения для пакета "Мандор".

Этот инструмент строит функцию распределения с учетом размерности: 1, 2 или 3. Могут быть любые комбинации различных параметров 'X', 'Y', 'Z', " Ух", 'Уу', 'VZ', 'e', 'thetaZ', 'phiZ'. Диагностика не предполагает удаление частиц. В программе может быть использован пространственный фильтр (срез плоскости) . Параметры, их диапазон, количество ячеек и некоторые другие настройки (см. далее) можно установить в конфигурационном файле- 'diag_distrib.cfg'.

2.2 Программа параллельно считывает данные из файлов BinData, полученных в результате работы core.out. После этого осуществляется анализ и обработка считанных данных по заданным параметрам

2.3 Параметры задаются в файле 'diag_distrib.cfg, который содержащет следующие настройки:

2.3.1 Начальная запись (откуда начать обработку).

2.3.2 Конечная запись (если отрицательное значение, то до последнего файла)

(например: @ -1)

2.3.3 Шаг (>0).

(например: @ 1)

2.3.4 Система единиц (1 - micron/fs, 0 - lambda/t0).(Например: @ 1)

2.3.5 q/M(например:-1.0)

2.3.6 in [0, 50], 100 bins (например:@ 'x')

2.3.7 Отрезающая плоскость (все частицы за ней удаляются из диагностики):

X [micron], Y [micron], Z [micron], nx, ny, r0 = (1, 1, 1) micron, n = (1, -1, 0)

2.3.8 *примеры и аббревиатуры:

```
          x  y  z                  | micron microns um cm r0 lambda
   vx vy vz       | c cm/sec
   e  energy              | erg eV KeV MeV GeV TeV
   thetaZ phiZ        | rad radians degree degrees
   f                      | 1 N q E M
- number of parameters(axises) can be 1, 2 and 3, for example:
  o 1D energy distribution:
    @ 'e'  in [0.0e+0,   8.0e+3], 60 bins
  o 2D phase plane (y, vy):
    @ 'y'  in [0,  6], 60 bins
    > 'vy'  in [-3.0e+0, 3.0e+0], 60 bins
  o 3D phase plane + energy (x, vx, e):
    @ 'x'  in [0,  6], 60 bins
    > 'e'  in [0,  1e7], 60 bins
    > 'vx'  in [-3.0e+0, 3.0e+0], 60 bins
- thetaZ and phiZ are spherical coordinate angles with respect to Z-axis, in degrees.
```

2.4 Для запуска на выполнение необходимо использовать команду следующего вида:

sbatch -n 64 -p test impi ./o/distrib.out diag_c,

где diag_distrib.cfg – тот самый файл с конфигурацией, лежащий в корневой папке.

-

3 Обработка данных

3.1 Подход

Задание состоит в следующем: необходимо вывести распределение по энергии частиц, прошедших через данное сечение в ходе выполнения численного моделирования на ядре, анализируя сохраненные чекпоинты.

Данные:

Исходные данные хранятся в бинарных файлах, записанных программой core.out. В интересующих нас файлах находится информация о частицах плазмы в данный момент времени. Файл имеет название следующего вида: plasma_[checkpoint]_[cpu].dat. Где checkpoint – номер момента времени, для которого ядро произвело запись данных в файл; cpu – номер процессора, который произвел эту запись. В каждом файле в начале записано количество частиц, которых он содержит, и информация про каждую частицу. В информацию входят: координаты, скорость, id. Примечательно, что принадлежность id к процессору не меняется со временем, что позволяет написать хорошо распараллеливаемую программу.

Сопутствующие сложности:

а) Каждая частица снабжена своим id, но в массиве данных они не отсортированы, следовательно, на каждой итерации необходимо сортировать частицы по id (Решение: создать статический массив, отсортированный по id только в первой итерации. Далее, путем копирования и подгружения новых данных в этот массив производить считывание данных за один проход. Решение не оправдало себя, поскольку всего в массиве (самом простом) от 10 миллионов частиц. Хранение такого объёма информации в оперативной памяти даже такого мощного компьютера как Ломоносов приводит к ошибке 11, недостаточности памяти. Работающее решение изложено в пункте «Конечный алгоритм выполнения»)

б) Код будет зависеть от большого числа (>10) уже имеющихся файлов проекта. Поэтому имеет смысл добавить инструкции к компиляции написанной программы в общий Makefile проекта. (что и было сделано)

в) Параллелизация зависит от выполнения core.out, поскольку ядро выводит чекпоинты в стольких частях на скольких процессорах был запущен код ядра. (решение1: синхронизовать количество процессоров ядра и обработчика, решение 2: добавить код подгружающий файлы распределения с учётом разности процессоров (были применены оба метода, но второй метод выдавал только часть необходимых данных (порядка0,4%), поскольку строго необходимо, чтобы все выданные core.out данные для одной контрольной точки обрабатывались одновременно)

г) Вывод в файл будет производиться параллельно, т.е. нужна программа, собирающая постфактум все файлы вывода в один. (решено использованием программы «Origin», открывающей все файлы одновременно)

Конечный алгоритм выполнения: запускаем ядро на N процессорах, оно выдаёт бинарные данные: контрольные точки с некоторым шагом по времени. Причем каждая контрольная точка записывается в N файлах. Запускаем разработанную в ходе курсовой работы программу на тех же N процессорах:

```
sbatch -n 64 -p gpu impi ./o/setup.out mandor_old.ini create
sbatch -n 64 -p gpu impi ./core.out
sbatch –n 64 –p gpu impi ./o/plane_count.out
```

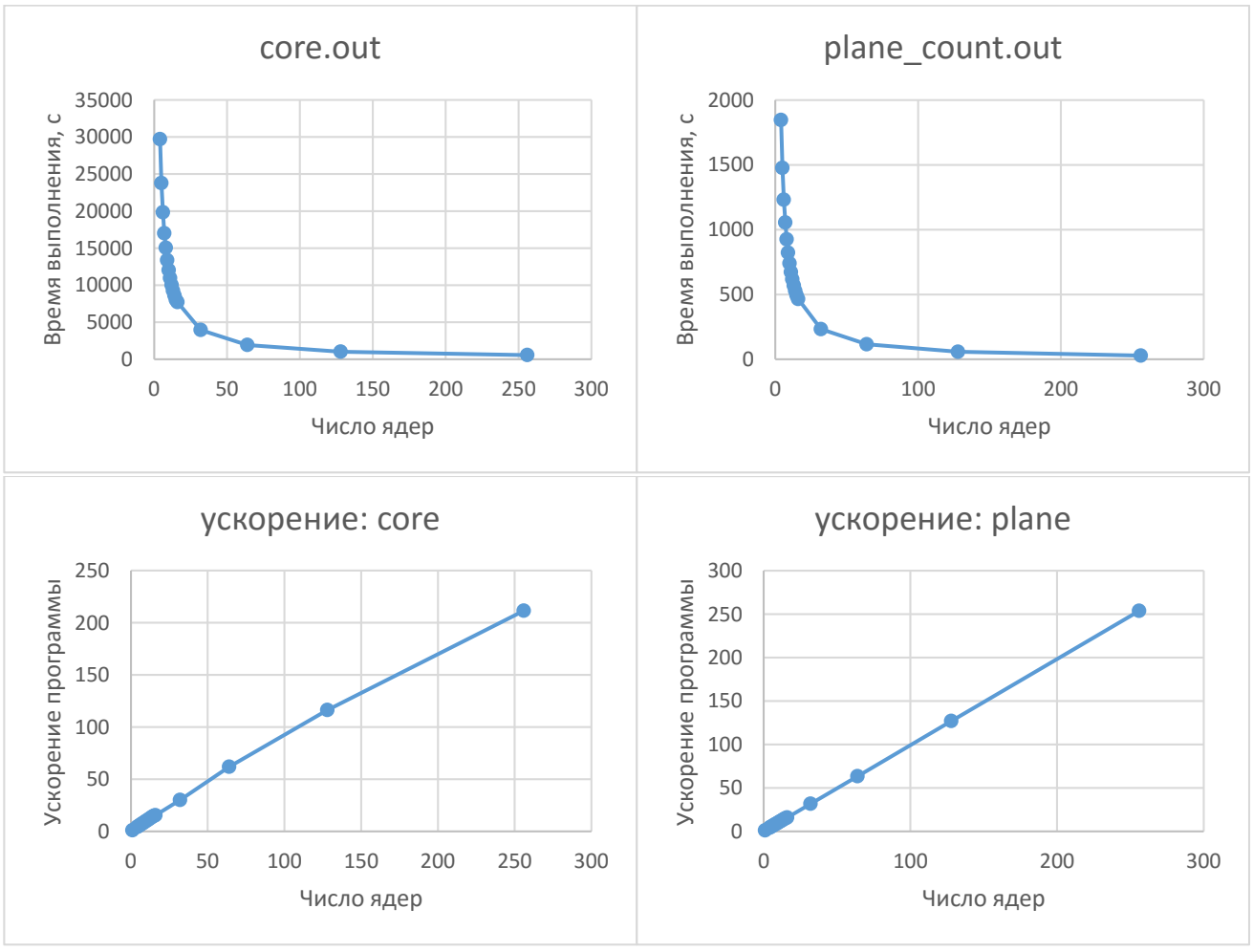
Эта программа работает следующим образом: Открываются контрольные точки, следующие друг за другом в диапазоне, представляющем интерес для получения распределения

[240;340], каждая из них просматривается обработчиком, причём так, что один процессор открывает соответствующий ему кусок плазмы. Если частица из данного куска находится вблизи заданного программе сечения (рамки), то происходит проверка направления движения частицы условие прохода её через рамку. Если частица проходит через рамку, то в файл выводится значение её энергии. Поскольку интерес представляет только высокоэнергетическая часть спектра, то проверка на id излишня и частицы, прошедшие через рамку с вероятностью p>99% уже не пройдут рамку обратно.

Но эта проверка всё же была осуществлена, что снизило скорость выполнения программы в $(10{,}2 \pm 0{,}2)$ раз не изменив при этом получаемое распределение (график остался таким же, только низкоэнергетическая часть стала примерно вдвое ниже в максимуме). Проверка id осуществлялась простым алгоритмом использующем массив short int (bool) для запоминания id частиц, прошедших рамку. После проверки, изложенной выше, вместо записи в файл id, записывалось в массив. При проходе алгоритма через следующую контрольную точку для частиц с id из массива с номерами, записанными на предыдущем шаге, производилась проверка прохода через сечение. Если частица проходила, то id оставалось в массиве и происходил вывод энергии в файл, иначе id удалялось из массива. При этом на каждом шаге происходит проверка наличия частицы, которая может пройти через сечение в массиве.

Если она уже проходила, то она игнорируется.

3.2 Параллелизация



-

Из графика для ускорения написанной программы (plane_count.out) видно, что благодаря синхронизации с ядром (core.out) и его выводом программа является практически идеально параллелизуемой.

3.3 Результаты

В результате выполнения программы для двумерного случая, показанного на рисунке 3:

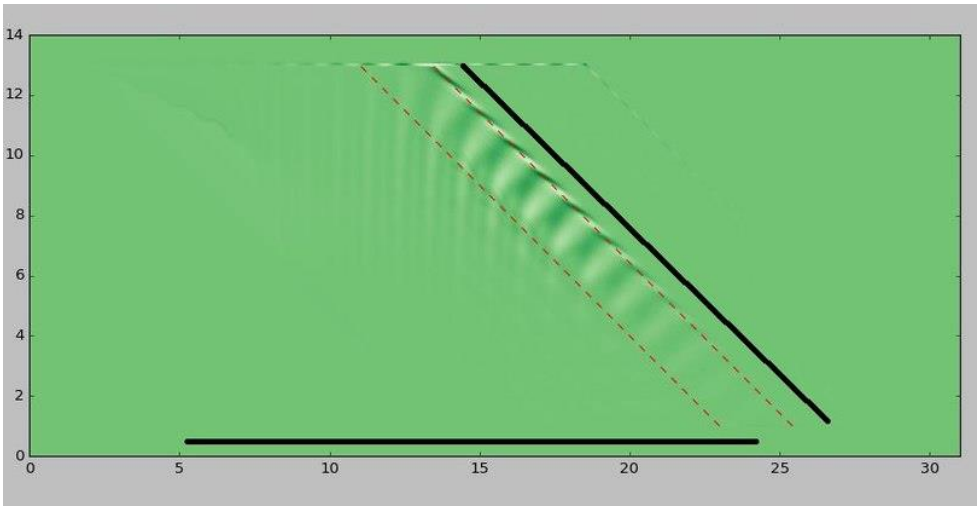
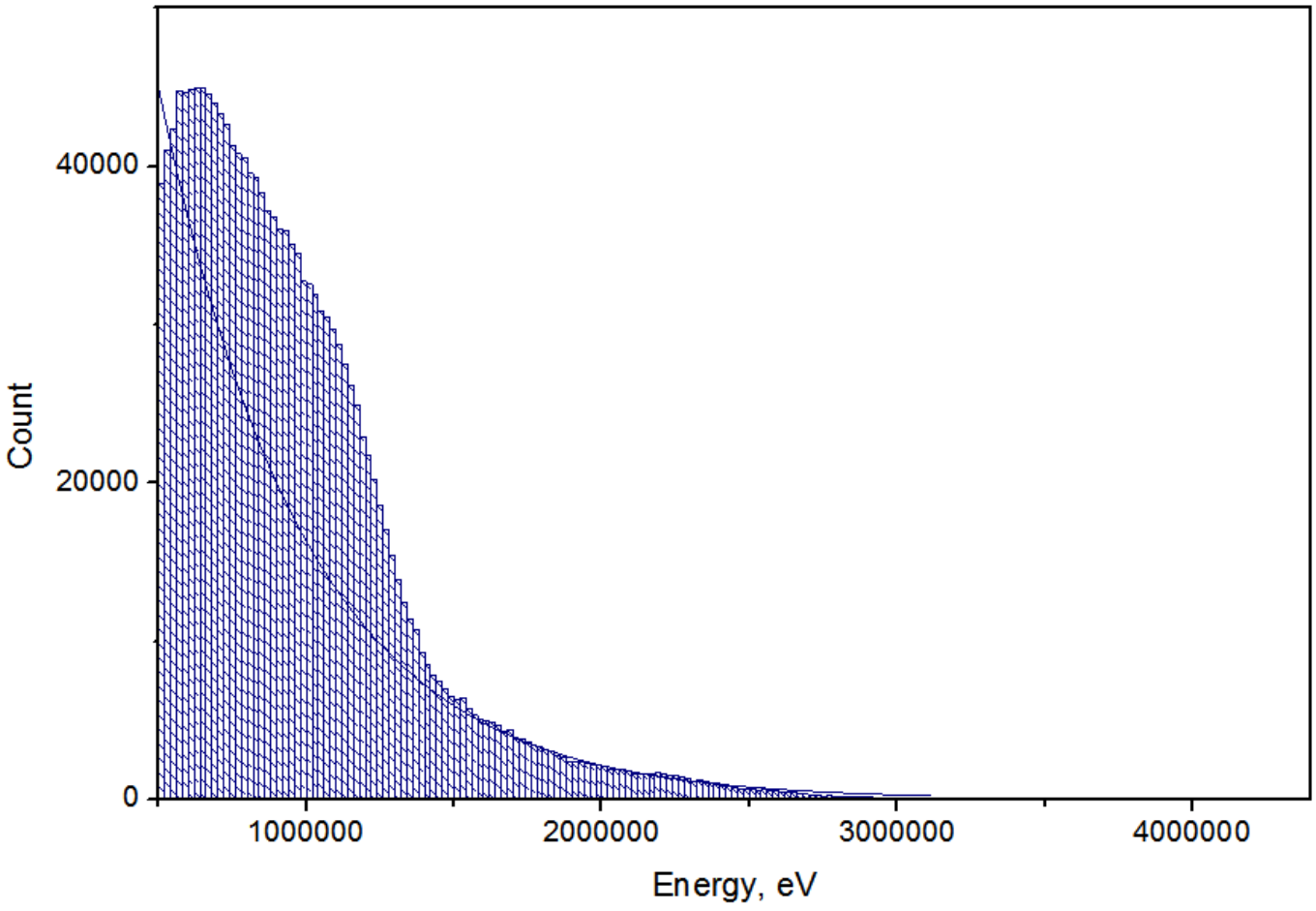


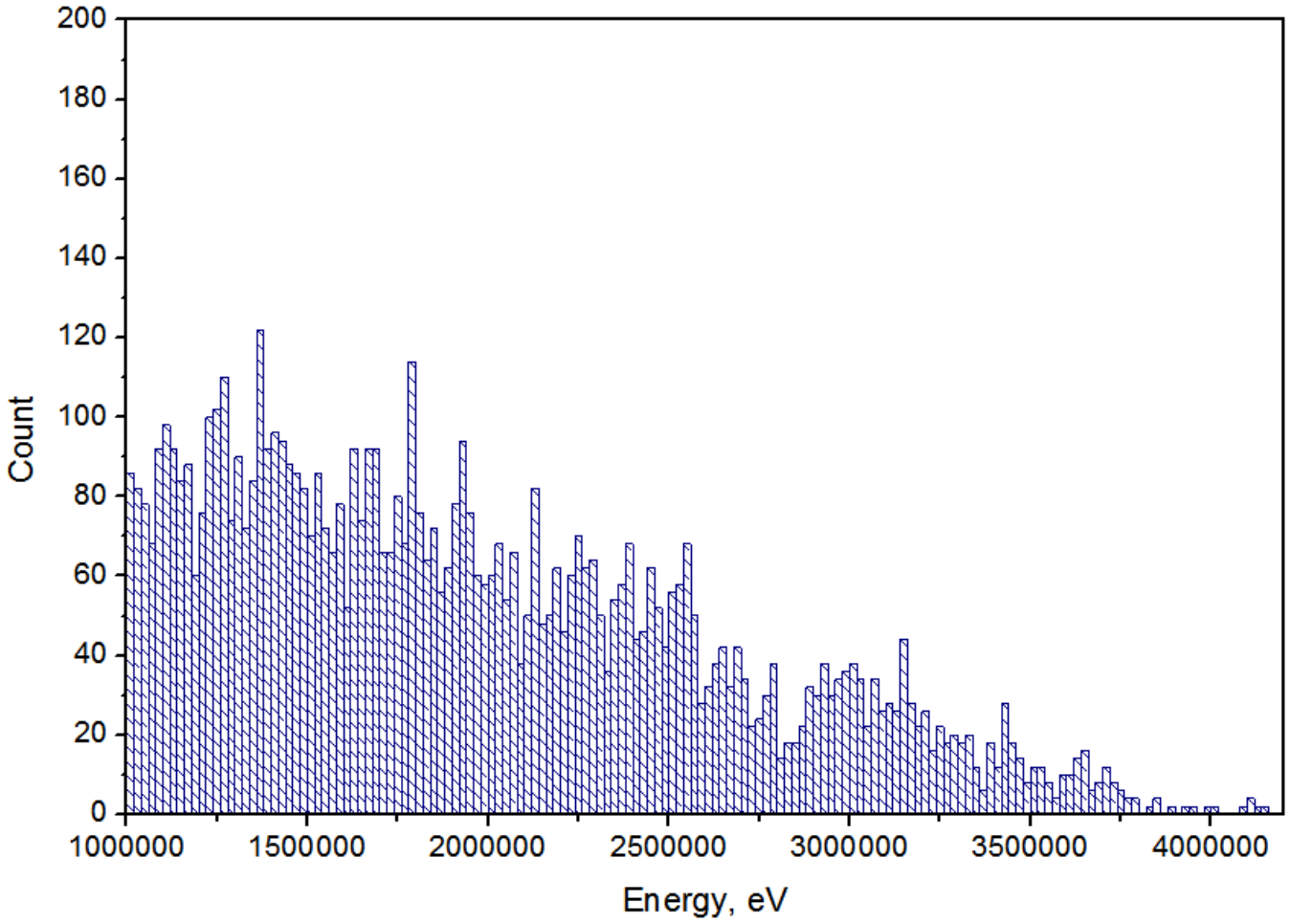
Рис 3

удалось получить следующее распределение частиц:

Сечение - прямая, указанная под углом на рис 3:



Сечение - прямая, указанная снизу на рис 3:



Форма распределения для частиц, прошедших с энергией, большей, чем 0,5 MeV через заданную под углом рамку напоминает экспоненциальное распределение. В то время как для нижней рамки можно оценить распределение как линейно убывающее.

-

4 Вывод

В результате работы были получены базовые представления о плазме, о процессе моделирования лазерной плазмы посредством PIC-кода. Произведено знакомство с таким кодом, используемым на практике – «Mandor». В ходе работы написано расширение для кода этой программы, позволяющее получить распределение по энергиям частиц, проходящих в процессе моделирования заданное сечение. Представлены графики этих распределений и произведена экспоненциальная аппроксимация одного из графиков.

Литература

[1] Александров А.Ф., Богданкевич Л.С., Рухадзе А.А. Основы электродинамики плазмы (Высшая школа, 1978)

[2] Бедсель Ч., Ленгдон А. Физика плазмы и численное моделирование (Энергоатомиздат 1989)

[3] Воеводин В.В., Воеводин В.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002. - 608 с.

[4] Григорьев Ю. Н., Вшивков В.А., Федорук М.П. Численное моделирование методами частиц-в-ячейках