

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
профессионального образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»

Факультет Автоматики и электроники

Кафедра электрофизических установок

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе на тему:

**«РАСЧЕТ ДИНАМИКИ ЗАРЯЖЕННЫХ ЧАСТИЦ В СТРУКТУРАХ
СЛОЖНОЙ ГЕОМЕТРИИ С ЭЛЕМЕНТАМИ ПАРАЛЛЕЛЬНОГО
ПРОГРАММИРОВАНИЯ»**

Студент дипломник Варнин А.С.

Руководитель проекта Ращиков В.И.

к.т.н., доцент

Рецензент Комаров Д.А.

к.ф.-м.н, начальник отдела ФГУП НПП "Торий"

Зав. кафедрой Диденко А.Н.

профессор, Член - корр. РАН

Москва - 2014 г.

Содержание

Введение.....	3
Программа СУМА.....	10
Математическая модель.....	11
Раздача заряда и токов на сетку.....	13
Решение уравнений Максвелла.....	15
Решение уравнения Пуассона совместно с уравнением Максвелла.....	22
Интегрирование уравнения движения.....	28
Описание блоков программы СУМА.....	33
Библиотека СУМА.....	33
Описание параметров программы.....	36
Списки инициализации (namelist).....	37
Модернизация интерфейса программы.....	39
Описание элементов интерфейса SUMA.....	40
Запуск и процесс расчета.....	40
Переключение шагов.....	40
Мгновенные распределения.....	41
Распределения на границах.....	48
Траектории отдельных частиц.....	50
Настройки вывода графиков.....	51
Прочие элементы интерфейса.....	53
Дополнительная программа для описания геометрии.....	53
Оптимизация использования ресурсов машины.....	57
Вводные сведения о параллельных вычислениях.....	57
Автоматизация внедрения параллельных расчетов при помощи компилятора....	60
Используемые настройки компилятора.....	63
Анализ проведенных оптимизаций программы СУМА.....	63
Моделирование динамики дейтронного пучка в нейтронной трубке.....	68
Заключение.....	84
Литература.....	85

Введение

На сегодняшний день благодаря стремительно растущим возможностям вычислительной техники численное моделирование становится неотъемлемой частью практически любого научного исследования. Не обходится без него и исследование динамики заряженных частиц. Значение вычислительного эксперимента особенно велико там, где имеется большой разрыв между возможностями теории и эксперимента. При разработке теорий обычно опираются на линейность или возможность разложения в степенные ряды, высокую степень симметрии, небольшое количество параметров и упрощающие приближения. Эксперименты в лаборатории сталкиваются со всей сложностью природы: условия могут с трудом поддаваться контролю, измерения не всегда легко осуществлять, и, как следствие, результаты зачастую трудно интерпретировать однозначно. Вычислительный эксперимент благодаря его универсальности становится связующим звеном между теорией и экспериментом.

Задача расчета динамики заряженных частиц в электромагнитных полях в различных сложных ВЧ структурах возникает при моделировании широкого круга процессов в ускорителях заряженных частиц, приборах СВЧ, полупроводниковых приборах и т. д. В большинстве практически интересных случаях поля могут быть найдены лишь с помощью численного моделирования.

Так как в подавляющем большинстве сложных моделей структура поля скоростей электронов в пучке имеет многозначное распределение, то совершенно очевидно, что коды, основанные на эйлеровой схеме, здесь не приемлемы. Поэтому при решении данного спектра задач используют коды на основе лагранжевой схемы, моделирующие динамику электронного потока так называемым “методом крупных частиц”.

Для пучков с отраженными частицами одним из первых стал применять этот метод О. Бунеман[1], в качестве крупных частиц он использовал заряженные плоские листы. Такой подход получил свое широкое применение в одномерных электростатических кодах. Основной недостаток кодов с плоскими листами заключается в их неограниченности в поперечнике, что не позволяет рассчитывать приборы с реальным значением тока. Этого недостатка нет в электростатическом коде с крупными частицами в виде жестких заряженных колец или дисков конечных размеров, что позволяет оценить некоторые численные характеристики пучка при реальных параметрах задачи.

Дальнейшее совершенствование компьютерной техники позволило перейти к многомерному моделированию на основе более развитых схем расчета. Наиболее удобным оказался particle-in-cell – метод (PIC метод) для расчета релятивистской динамики крупных частиц на пространственной сетке[2], в узлах которой разностным методом решаются уравнения электромагнитного поля. Иногда используются “cloud-in-cell”- метод (CIC-метод[3], в котором крупные частицы представлены в виде размазанных по пространству облачков заряда).

К настоящему моменту в различных научных лабораториях созданы комплексы вычислительных программ, позволяющие проводить самосогласованное моделирование динамики частиц в различных приборах, которые используют разностный метод решения уравнений Максвелла на сетке практически в любой трехмерной геометрии и релятивистский **particle in cell (PIC)** вариант метода крупных частиц. Среди них можно выделить “КАРАТ”[4], “MAGIC”[5], “QUICKSILVER”[6] и др.

Наиболее известные коды, позволяющие моделировать динамику электронных пучков, представлены в таблице.

Таблица 1: Характеристики наиболее известных кодов для моделирования динамики сильноточных пучков.

Код	Размерность и тип координатной сетки	Метод расчета динамики частиц	Метод расчета электромагнитного поля
Рубин[7]	2.5	РІС	уравнения Максвелла или уравнения Пуассона.
КАРАТ	$i(x-z), (r-z), (r-\theta)$ $(x-y-z), (r-z-\theta)$	РІС(СІС)	уравнения Максвелла или уравнения Пуассона.
ССУВЕ[8]	$2.5 (r-z)$	РІС	уравнения Максвелла
QUICKSILVER	3	РІС	уравнения Максвелла
MAGIC	$2.5 (r-z)$	РІС	уравнения Максвелла.
PARMELA	$2 (r-z)$ и 3	Particle-push method	Решение уравнений Максвелла
Microwave studio	3	FIT метод конечных интегралов	Решение уравнений Максвелла

Код	Размерность и тип координатной сетки	Метод расчета динамики частиц	Метод расчета электромагнитного поля
SUPERFISH[9]	2-3	PIC	уравнения Максвелла или уравнения Пуассона

Для расчета динамики частиц используют классический PIC (particle in cell) метод или по-другому - частица сетка. В методе частица-сетка используется формализм близкодействия и уравнение поля для потенциала. В результате сила вычисляется гораздо быстрее, но обычно менее точно, чем при использовании метода PP(метод частица – частица). Всю иерархию вычислительных методов динамики сильноточных пучков можно разбить на два типа: singleparticle code, multiparticle code (PARMELA, ASTRA, GPT, Impact-T, elegant, KARAT, MAGIC). В singleparticle code, пучок рассматривается как цельная система, основными характеристиками которой являются энергия пучка, время пролета пучка, фаза и изменение энергии пучка. Преимущество такого моделирование является скорость вычислений. Multiparticle code, напротив дает более детальное описание динамики поведения пучка, но затрачивает большее количество времени и вычислительных ресурсов. Математическая модель такого кода сильно подвержена неустойчивостям численного решения, восприимчива к шумам, чувствительна к начальным условиям инжектируемого пучка. Рассмотрим несколько из них.

Код Parmela [10] был разработан в конце 80-х годов прошлого столетия и с тех пор успешно и широко используется для решения электродинамических задач (преимущественно для расчета динамики пучков в линейных ускорителях). Расчет динамики сильноточного пучка может быть выполнен для различных 2х и 3х мерных геометрий: группирователи,

виглеры, соленойды, накопители, ВЧ резонаторы, секступоли, квадруполь и т. д.

В основе алгоритма лежит PIC-метод. В ходе расчета учитывается влияние паразитных полей и пространственного заряда. В данном методе рассматривается так называемая модель пучка спокойного старта “quite start”, инжектируемый пучок берется с минимальным энергетическим разбросом (как правило по распределению Гаусса), чтобы исключить малый шум (корреляцию). Основным недостатком построения математических моделей, такого вида, как уже было упомянуто, является сложность задание состояния инжектируемого пучка, что оказывает большое влияние на изменение эммитенса и яркости пучка. Электромагнитные поля, полученные в ходе решения уравнений Максвелла и Пуассона, импортируются из следующих программ Magic или Superfish.

Код КАРАТ (Москва, ИТЭС ОИВТ РАН) написан в конце 80х годов прошлого века и с тех пор успешно и интенсивно используется для решения электродинамических задач.

Задачи могут быть поставлены для различных 2х и 3х мерных геометрий с учетом запуска, распространения и выпуска различных типов электромагнитных волн и пучков частиц, моделирования плазменных процессов, учета конечной проводимости и диэлектрических свойств сред, вторичной эмиссии частиц, ионизации газа, моделирования нагрузок и источников в виде сосредоточенных параметров, с учетом фольг, внешних электрических и магнитных полей.

В основе алгоритма лежит полная система уравнений Максвелла и PIC-метод. Для решения системы уравнений Максвелла применена разностная схема на сдвинутых сетках.

Программа CST MWS [11] использует метод конечных интегралов (FIT) - достаточно общий подход, который сначала описывает уравнения Максвелла на пространственной сетке, с учетом закона сохранения энергии, а

затем по ним формирует систему специфических дифференциальных уравнений, таких как волновое уравнение или уравнение Пуассона. Метод может быть реализован как во временной, так и в частотной области. Кроме того, не накладывает никаких ограничений на тип используемой сетки разбиения, наряду со структурированной сеткой в декартовой системе координат поддерживаются неортогональные сетки, например, тетраэдральная. Таким образом, программа CST MWS первый на настоящий момент пакет объемного электромагнитного моделирования позволяющий выбирать оптимальные для данной задачи метод решения и способ разбиения.

В основе программы лежит разработанный компанией CST метод аппроксимации для идеальных граничных условий (РВА), удачно дополняющий хорошо зарекомендовавший себя метод определенных интегралов (FI), работающим во временной области. В любом методе, связанном с моделированием конечных элементов, все поверхности разбиваются на небольшие элементы. Если модель СВЧ устройства задана только прямыми плоскостями, то число анализируемых элементов разбиения невелико, и расчет проводится относительно быстро. При использовании в СВЧ устройствах криволинейных поверхностей, для их аппроксимации требуется намного большее количество элементов разбиения, что приводит к значительным временным затратам при анализе. Комбинация методов РВА и FI, предложенная компанией CST, позволила быстро решать задачи моделирования сложных СВЧ устройств с криволинейными поверхностями на широко распространенных компьютерах с процессорами Pentium. Типичными устройствами, моделируемыми с помощью пакета CST Microwave Studio, являются: волноводные и микрополосковые направленные ответвители мощности, делители и сумматоры мощности; волноводные, микрополосковые и диэлектрические фильтры, одно- и многослойные микрополосковые структуры; различные линии передачи, коаксиальные и многовыводные соединители; коаксиально-волноводные и коаксиально-

полосковые переходы, оптические; волноводы и коммутаторы, различные типы антенн - рупорные, спиральные, планарные, электронные пушки, высоковольтные лампы, магнетроны и т.д. Доступная в настоящий момент версия продукта CST PARTICLE STUDIO (CST PS) включает следующие функции:

- Моделирование траекторий элементарных частиц в статических и динамических магнитном и электрическом полях с учетом эффекта объемного заряда.
- Различные модели эмиссии, например, Space-Charge Limited Emission.
- Мощные возможности построения 3D моделей.
- Параметризация и автоматическая оптимизация.
- Поддержка макроскриптов на языке VBA и технологии OLE (COM/DCOM).

Все перечисленные выше программы являются коммерческими продуктами.

Программа СУМА

Программа СУМА(Система Уравнений МАксвелла) была написана в середине 80-х годов на языке Фортран для ЭВМ серии ЕС. Первая публикация по готовому коду была осуществлена в 1990 [1]. Начиная с этого времени программа интенсивно использовалась для решения ряда интересных физических задач, требующих численного моделирования [2-10]. Код представляет собой 2.5 мерную нестационарную модель, позволяющую описывать самосогласованным образом динамику заряженных частиц в прямоугольной, цилиндрической и полярной системе координат. В программе моделируется вход, распространение и выход различных типов электромагнитных волн и заряженных частиц с любыми значениями заряда и массы, плазменные процессы, учитывается конечная проводимость и диэлектрические свойства среды. Реализована возможность моделирования вторичной эмиссии частиц, ионизации газа, нагрузок и антенн, задания внешних электрических и магнитных полей любого вида.

Математическая модель

Система уравнений математической модели состоит из уравнений Максвелла, уравнения среды и уравнения движения. На каждом шаге решения в текущий момент времени t сначала выполняется вычисление плотностей заряда и тока, входящие в уравнения Максвелла. Раздача заряда и токов в узлы пространственной сетки и их сглаживание выполняется путем взвешивания площадей частицы (облака) и ячейки сетки. Поступление новых частиц на шаге моделирования Δt в рассматриваемую область реализовано в виде механизма инжекции, эмиссии, либо вторичной эмиссии с соответствующими законами распределения. После этого численно решаются уравнения Максвелла, и осуществляется коррекция полученного решения на предмет соответствия уравнению Пуассона. Коррекция необходима, так как в процессе моделирования часть частиц могут покинуть исследуемую область, выходя через её границы, либо попадая на поверхности электродов.

Коррекция выполняется путем решения уравнения Пуассона для разности распределения плотностей заряда, полученных из дивергентного уравнения Максвелла, и реальным распределением зарядов ρ .

$$\Delta \Phi_c = \frac{-(\rho - \rho^*)}{\varepsilon_0}, \text{ где } \operatorname{div} E = \frac{\rho}{\varepsilon}.$$

Скорректированное выражение для поля имеет следующий вид

$$E_c = E - \operatorname{grad}(\Phi_c).$$

Уравнение Пуассона решается с использованием алгоритма быстрого преобразования Фурье по одной координате и прогонки (алгоритм Томаса) по второй. Для областей со сложной геометрией, а также при наличии электродов внутри области, используется метод матрицы ёмкости, связывающий потенциал и заряд в необходимых узлах.

На границе области могут задаваться следующие краевые условия.

- Металлическая поверхность.
- Условия периодичности.
- Условия симметрии.
- Условия прохождения волны.

В представленной вычислительной модели электроны и ионы объединялись в крупные частицы с соответствующим зарядом и массой, и рассчитывалось движение крупных частиц, число которых, на много порядков ниже реального (в большинстве рассматриваемых вариантов число крупных частиц составляло несколько миллионов). Полевые величины, заполняющие все пространство физической системы, приближенно представлялись значениями в регулярно расположенных узлах пространственной сетки. Для реализации данного алгоритма изучаемый ансамбль физических частиц воспроизводился с помощью модели частиц-облаков (крупных частиц). Заряды и массы представленных в модели крупных частиц могут быть любыми.

Поскольку в результате решения уравнений Максвелла поля находятся в узлах пространственной сетки, для численного интегрирования уравнений движения необходимо вычисление поля в промежуточных точках, где располагаются частицы. Для этого используется интерполирование и сглаживание сеточных функций.

Дифференциальные операторы заменяются конечно-разностными аппроксимациями на сетке. Потенциалы в месте положения частицы вычисляются посредством интерполяции по массиву сеточных значений. Сеточные плотности рассчитываются с помощью обратной процедуры раздачи характеристик частицы (например, заряда) в ближайшие узлы сетки для того, чтобы получить сеточные значения (например, плотность заряда с учетом наличия частиц разного знака).

Количество частиц в модели $N(t)$ с течением времени меняется вследствие поступления (эмиссии, инжекции, вторичной эмиссии,

ионизации) и ухода частиц через границы области. Таким образом, алгоритм вычисления шага по времени выглядит следующим образом:

- 1) Раздача заряда на сетку для дейтронного и электронного компонентов.
- 2) Решение на этой сетке уравнения Пуассона.
- 3) Вычисления поля.
- 4) Интегрирование уравнения движения для каждого сорта частиц.

Раздача заряда и токов на сетку.

Для максимального уменьшения уровня шумов или флуктуаций в системе раздача заряда на сетку было выполнено с помощью метода “облака в ячейках” CIC (clouds in cell). Пусть единичный заряд q нулевого размера, расположенный в точке r_a , создает в i -м узле плотность

$$\rho(r_i) = q \sum_a S(r_a, r_i); S(r_a, r_i) = \int S_q(r, r_a) w(r, r_i) dr', \quad (1)$$

где $S_q(r, r_a)$ - весовая функция, учитывающая, как форму облака, так и способ раздачи заряда, $w(r_a, r_i)$ - интерполяционная формула, описывающая способ (правило) “раздачи заряда” в узлы сетки, $S_q(r, r_a)$ - коэффициент формы. Для оптимальной точности было принято, что $h = a_q$, где a_q - размер частицы облака, а h - размер сетки.

При этом $S_q(x)$ совпадает с $w(x)$, т.е. облако имеет форму сеточной ячейки; интервал на котором два облака не взаимодействуют, стягивается в нуль; плоская вершина весовой функции исчезает:

$$S(x) = \begin{cases} (1 - |x|/h)/h, & |x| < h \\ 0, & |x| \geq h \end{cases},$$

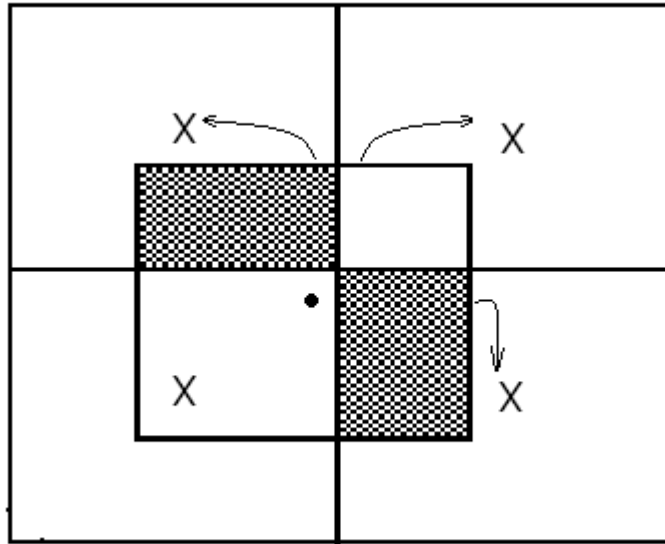
(2)

В данном методе прямоугольное облако движется по сетке. Использование интерполяционной функции означает, что доля объема облака, попадающая в ячейку сетки, которая окружает некоторый узел, составляет

вклад в число частиц в этом узле. Таким образом, вклад частицы с номером a в число частиц в i -м узле, т.е. значение $V_g S(x_i - x_a)$ (V_g — объем ячейки) представляет собой отношение заштрихованной площади к площади частицы облака (рис. 1). Поэтому такой способ называется еще взвешиванием по площадям (area weighting). Алгоритм распределения токов на разностной сетке осуществляется по аналогичной схеме:

$$j_a(r) = qv_a \sum_a S(r_i - r_a) \quad (3),$$

где v_a — скорость частицы с номером a .



На погрешность аппроксимации плотности заряда и тока влияет конечность шага моделирования Δt . Если за шаг $[t, t + \Delta t]$ центр облака переходит из i -й ячейки в ячейку j , то мгновенная картина расположения частиц в момент t даст завышенную плотность в ячейке i и заниженную в ячейке j . Вследствие чего возникает сеточный шум. Шум объясняется тем, что в описанном методе при сглаживании используется лишь фазовое положение частиц, наблюдаемое в текущий момент времени t . Вычисляемая

таким образом плотность дает поэтому мгновенную картину ансамбля, но не учитывает его динамику в течении шага Δt .

Огромный выигрыш в скорости метода достигается ценой потери разрешения в поле потенциала и силы. Ошибки, допускаемые бесстолкновительной моделью частиц в ячейке, обусловлены конечным размером шага, как по пространству, так и по времени. Конечность пространственного шага ведет прежде всего к тому, что не полностью описываются Фурье-гармоники потенциала, и поэтому нелинейные явления, в которых связано множество гармоник, не могут быть переданы адекватно. Эти ошибки все же можно уменьшить, выбрав малые временные и пространственные шаги. Однако, на отдельных моделях было доказано, что при выбранном способе вычисления сглаженной плотности объемного заряда и тока в узлах нельзя неограниченно снижать погрешность, уменьшая шаг пространственной сетки h , т.е. существует оптимальное соотношение между числом частиц и числом сеточных ячеек. Это объясняется тем, что чрезмерное увеличение сетки при том же числе частиц увеличивает скачки плотности объемного заряда и тока в узлах, т.е. ухудшает их аппроксимацию. В результате возрастает погрешность решения уравнений для собственного поля пучка и погрешность моделирования в целом. Аналогично оптимальный временной шаг Δt зависит от шага сетки h , поскольку от h зависит погрешность вычисления поля в точках нахождения частиц, а, следовательно, погрешность интегрирования уравнений движения.

Решение уравнений Максвелла.

Непосредственное решение уравнений системы Максвелла обеспечивает автоматический учет таких физических явлений, как излучение и запаздывание.

Предположим, что свойства среды, характеризуемые диэлектрической ϵ и магнитной μ проницаемостями, не зависят от времени.

Тогда электрические и магнитные поля выражаются, как

$$E = \tilde{E} + \bar{E} \quad (4)$$

$$B = \tilde{B} + \bar{B} \quad (5)$$

где \tilde{E} и \tilde{B} , поля переменные во времени генерируемые токами и частицами, \bar{E} и \bar{B} - квазистатические поля. Переменные во времени поля \tilde{E} и \tilde{B} определяются уравнениями Максвелла:

$$\nabla \times \tilde{B} = J + \frac{\partial \tilde{D}}{\partial t} \quad (6)$$

На отдельных участках границы Γ рассматриваемой области могут быть поставлены следующие краевые условия:

1. Условие Дирихле, т.е. заданные значения $E(r, t)$ или $H(r, t)$
2. Условие симметрии (однородное условие Неймана), т.е. обращение в нуль нормальной производной поля.
3. Условие периодичности
4. Условие идеальной проводимости $E_t = 0, H_n = 0$, где E_t тангенциальная (к границе) составляющая электрического поля. H_n нормальная составляющая магнитного поля.
5. Условие “прохождения волны”, т.е. отсутствие отражения.

Краевые условия обычно задаются на основании электродинамического рассмотрения и не учитывают перенос частиц через границу области. Поэтому моделирование процессов поступления и ухода частиц также создает трудности при решении уравнений Максвелла, так как нарушает краевые условия. Кроме того, изменение заряда в области должно

сопровождаться соответствующими изменениями электрического поля согласно уравнению Пуассона.

Уравнение движения заряженной частицы имеет следующий вид:

$$\frac{d\mathbf{p}}{dt} = q \left(\mathbf{E} + \left[\frac{\mathbf{v}}{c} \mathbf{B} \right] \right), \quad (7)$$

где \mathbf{p} и \mathbf{v} импульс и скорость частицы $\mathbf{p} = m \mathbf{v} \gamma$; $\gamma = 1 / \sqrt{1 - \frac{v^2}{c^2}}$; m – масса покоя,

q – заряд частицы. Приведем уравнения Максвелла к безразмерному виду, воспользовавшись следующими заменами:

$$\mathbf{r}^i = \mathbf{r} / R, \quad \mathbf{v}^i = \mathbf{v} / c = \beta, \quad t^i = t c / R, \quad \mathbf{p}^i = \mathbf{p} / m c = \beta \gamma, \quad I^i = I / m c^3, \quad E^i = E \frac{e R}{m c^2}, \quad B^i = B \frac{e R}{m c^2},$$

$$\mathbf{j}^i = \mathbf{j} \frac{e R^2}{m c^3}, \quad \rho^i = \rho \frac{e R^2}{m c^2}, \quad \sigma^i = \sigma \frac{R}{c}, \quad \text{где } R - \text{нормировочная константа.}$$

Таким образом, уравнения приобретут следующую форму:

$$\begin{aligned} \nabla \times \sim B^* &= \mathbf{j}^* + \frac{\partial \sim D^*}{\partial t^*} \\ \nabla \times \sim E^* &= -\frac{\partial \sim B^*}{\partial t^*} \end{aligned} \quad (8)$$

$$\frac{d\mathbf{p}^*}{dt} = \mathbf{E}^* + [\rightarrow \beta \mathbf{B}^*] \quad (9)$$

В дальнейшем будем использовать безразмерные величины без *.

В основе предложенной программы лежит 2.5 мерная модель, в нашем случае учитываются две пространственные координаты $(r, z; r, \theta; x, y)$ и три компоненты скорости $(V_\theta, V_r, V_z; V_x, V_y, V_z)$. Таким образом, исходя из сказанного, в рассматриваемой программе реализованы следующие геометрии полей, которые могут вычисляться вместе, либо по отдельности.

Таблица 2: Конфигурация поля

GEOMETRY	TRIAD1			TRIAD2		
CARTESIAN						

(X, Y)	D_x	D_y	H_z	D_z	H_x	H_y
CILINDRICAL						
(R, θ)	D_z	D_R	H_θ	D_θ	H_z	H_R
CYLINDRICAL						
(R, Z)	D_θ	D_R	H_z	D_z	H_θ	H_R

Декартово пространство

TRIAD1

$$\frac{\partial D_x}{\partial t} = \frac{\partial H_z}{\partial y} - J_x$$

(10)

$$\frac{\partial D_y}{\partial t} = \frac{\partial H_z}{\partial x} - J_y$$

(11)

$$i \frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left(\frac{\partial}{\partial y} \frac{\partial D_x}{\partial \varepsilon} - \frac{\partial D_y}{\partial x \varepsilon} \right) \quad (12)$$

$$\frac{\partial D_x}{\partial x} + \frac{\partial D_y}{\partial y} = \rho$$

(13)

TRIAD2

$$\frac{\partial D_z}{\partial t} = \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) - J_z$$

(14)

$$\frac{\partial H_x}{\partial t} = \frac{-1}{\mu} \frac{\partial D_z}{\partial y \varepsilon} \quad (15)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu} \frac{\partial D_z}{\partial x \varepsilon} \quad (16)$$

$$\frac{\partial \mu H_x}{\partial x} + \frac{\partial \mu H_y}{\partial y} = 0 \quad (17)$$

Цилиндрическая геометрия в общем виде

$$\frac{\partial D_R}{\partial t} = \frac{1}{R} \frac{\partial H_z}{\partial \theta} - \frac{\partial H_\theta}{\partial z} - J_R$$

(18)

$$\frac{\partial D_\theta}{\partial t} = \frac{\partial H_R}{\partial z} - \frac{\partial H_z}{\partial R} - J_\theta$$

(19)

$$\frac{\partial D_z}{\partial t} = \frac{1}{R} \frac{\partial R H_\theta}{\partial R} - \frac{1}{R} \frac{\partial H_R}{\partial \theta} - J_z$$

(20)

$$\frac{1}{R} \frac{\partial R D_R}{\partial R} + \frac{1}{R} \frac{\partial D_\theta}{\partial \theta} + \frac{\partial D_z}{\partial z} = \rho$$

(21)

$$\frac{\partial H_R}{\partial t} = \frac{1}{\mu} \left(\frac{\partial D_\theta}{\partial z \varepsilon} - \frac{1}{R} \frac{\partial D_z}{\partial \theta \varepsilon} \right)$$

(22)

$$\frac{\partial H_\theta}{\partial t} = \frac{1}{\mu} \left(\frac{\partial D_z}{\partial R \varepsilon} - \frac{\partial D_R}{\partial z \varepsilon} \right)$$

(23)

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left(\frac{1}{R} \frac{\partial D_R}{\partial \theta \varepsilon} - \frac{1}{R} \frac{\partial R D_\theta}{\partial R \varepsilon} \right)$$

(24)

$$\frac{1}{R} \frac{\partial R}{\partial R} \mu H_R + \frac{1}{R} \frac{\partial \mu H_\theta}{\partial \theta} + \frac{\partial \mu H_z}{\partial z} = 0$$

(25)

Цилиндрическая геометрия (R, z)

TRIAD1

$$\frac{\partial D_z}{\partial t} = \frac{1}{R} \frac{\partial R H_\theta}{\partial R} - J_z$$

(26)

$$\frac{\partial D_R}{\partial t} = -\frac{\partial H_\theta}{\partial z} - J_R$$

(27)

$$\frac{\partial H_\theta}{\partial t} = \frac{1}{\mu} \left(\frac{\partial D_z}{\partial r \varepsilon} - \frac{\partial D_R}{\partial z \varepsilon} \right)$$

(28)

$$\frac{1}{R} \frac{\partial D_R}{\partial R} + \frac{\partial D_z}{\partial z} = \rho$$

(29)

TRIAD2

$$\frac{\partial D_\theta}{\partial t} = \frac{\partial H_R}{\partial z} - \frac{\partial H_z}{\partial R} - J_\theta \quad (30)$$

$$\frac{\partial H_z}{\partial t} = \frac{-1}{\mu} \frac{\partial R D_\theta}{R \partial R \varepsilon} \quad (31)$$

$$\frac{\partial H_z}{\partial t} = \frac{-1}{\mu} \frac{\partial R D_\theta}{R \partial R \varepsilon} \quad (32)$$

$$\frac{1}{R} \frac{\partial \mu H_R}{\partial R} + \frac{\partial \mu H_z}{\partial z} = 0 \quad (33)$$

Цилиндрическая геометрия (R, θ)

TRIAD1

$$\frac{\partial D_\theta}{\partial t} = \frac{-\partial H_z}{\partial R} - J_\theta \quad (34)$$

$$\frac{\partial D_R}{\partial t} = \frac{1}{R} \frac{\partial H_z}{\partial \theta} - J_R \quad (35)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left(\frac{1}{R} \frac{\partial D_R}{\partial \theta \varepsilon} - \frac{1}{R} \frac{\partial R D_\theta}{\partial R \varepsilon} \right) \quad (36)$$

$$\frac{1}{R} \frac{\partial R D_R}{\partial R} + \frac{\partial D_z}{\partial z} = \rho \quad (37)$$

TRIAD2

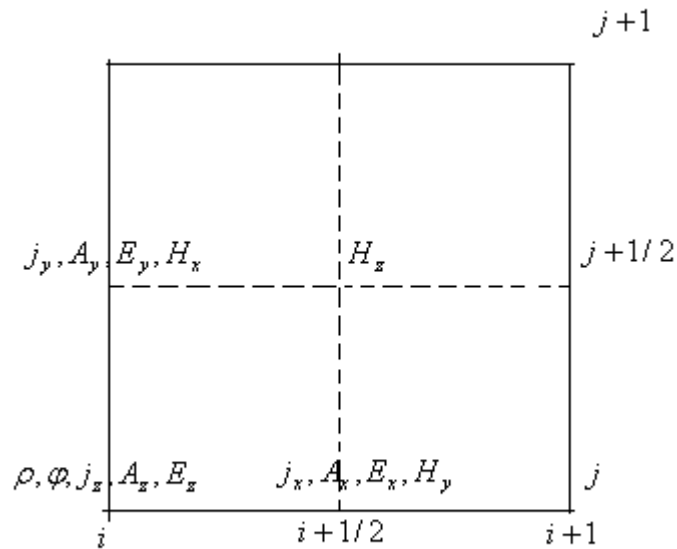
$$\frac{\partial D_z}{\partial t} = \frac{1}{R} \frac{\partial R H_\theta}{\partial R} - \frac{1}{R} \frac{\partial H_R}{\partial \theta} - J_z \quad (38)$$

$$\frac{\partial H_\theta}{\partial t} = \frac{1}{\mu} \frac{\partial D_z}{\partial R \varepsilon} \quad (39)$$

$$\frac{\partial H_R}{\partial t} = \frac{-1}{\mu R} \frac{\partial D_z}{\partial \theta \varepsilon} \quad (40)$$

$$\frac{1}{R} \frac{\partial R}{\partial R} \mu H_R + \frac{\partial \mu H_z}{\partial z} = 0 \quad (41)$$

Уравнения Максвелла решаются на двумерной сетке с помощью конечно-разностного двухшагового метода (рис. 2).



Так, например, для случая прямоугольной геометрии, трех компонент поля E_x, E_y и H_z и отсутствия внешних источников уравнения Максвелла примут вид ($\varepsilon = \mu = 1, \rho = 0, j = 0$) .

$$\frac{1}{c} \frac{dE_x}{dt} = \frac{dB_z}{dy}, \quad (42)$$

$$\frac{1}{c} \frac{dE_y}{dt} = -\frac{dB_z}{dx} \quad (43)$$

$$\frac{1}{c} \frac{dB_z}{dt} = \frac{dE_x}{dy} - \frac{dE_y}{dx}$$

(44)

Тогда в соответствии с явным методом и отсутствием внешних источников (см. рис. 2) на прямоугольной сетке $\bar{\omega}_n(i\Delta x, j\Delta y)$ получаем для внутренних узлов конечно-разностную схему:

$$\begin{aligned}
& \frac{1}{c\Delta T} \{E_x^{N+1}[(i+\frac{1}{2})\Delta x, j\Delta y] - E_x^N[(i+\frac{1}{2})\Delta x, j\Delta y]\} = \dot{c} \\
& \dot{c} \frac{1}{\Delta y} \{B_z^N[(i+\frac{1}{2})\Delta x, (j+\frac{1}{2})\Delta y] - B_z^N[(i+\frac{1}{2})\Delta x, (j-\frac{1}{2})\Delta y]\}, \\
& \frac{1}{c\Delta T} \{E_y^{N+1}[i\Delta x, (j+\frac{1}{2})\Delta y] - E_y^N[i\Delta x, (j+\frac{1}{2})\Delta y]\} = \dot{c} \\
& \dot{c} \frac{-1}{\Delta x} \{B_z^N[(i+\frac{1}{2})\Delta x, (j+\frac{1}{2})\Delta y] - B_z^N[(i-\frac{1}{2})\Delta x, (j+\frac{1}{2})\Delta y]\}, \\
& \frac{1}{c\Delta T} \{B_z^{N+1}[(i+\frac{1}{2})\Delta x, (j+\frac{1}{2})\Delta y] - B_z^N[(i+\frac{1}{2})\Delta x, (j+\frac{1}{2})\Delta y]\} = \dot{c} \\
& \frac{1}{\Delta y} \{E_x^N[(i+\frac{1}{2})\Delta x, (j+1)\Delta y] - E_x^N[(i+\frac{1}{2})\Delta x, j\Delta y]\} - \dot{c} \\
& \dot{c} \frac{-1}{\Delta x} \{E_y^N[(i+1)\Delta x, (j+\frac{1}{2})\Delta y] - E_y^N[i\Delta x, (j+\frac{1}{2})\Delta y]\},
\end{aligned}$$

которая осуществляет переход от временного слоя $N\Delta T$ к временному слою

$(N+1)\Delta T$. Данная схема устойчива при условии $\Delta T \leq \frac{\Delta x \Delta y}{c\sqrt{\Delta x^2 + \Delta y^2}}$ (45). Для

выполнения условия $\frac{dE}{dn}=0$ на границе вводятся дополнительные фиктивные

узлы с таким же, как и в граничных узлах значением поля.

Решение уравнения Пуассона совместно с уравнением Максвелла.

Рассмотрим уравнение непрерывности $\frac{d\rho}{dt} + \text{div}j = 0$ (46) т.е. закон

сохранения объемного заряда. Возьмем дивергенцию от вихревых уравнений Максвелла.

$$\text{rot } H = \frac{\partial D}{\partial t} + J, \text{rot } E = -\frac{\partial B}{\partial t} \quad (47)$$

С учетом уравнения непрерывности находим, что

$$\frac{\partial}{\partial t}(\text{div } D - \rho) = 0, \frac{\partial}{\partial t}(\text{div } B) = 0. \quad (48)$$

Следовательно, если уравнение Максвелла

$$\text{div } D = \rho, \text{div } B = 0 \quad (49)$$

удовлетворяются в области в начальный момент времени, то они будут удовлетворяться тождественно во времени при выполнении уравнения

непрерывности. В подавляющем большинстве интересных нам задач заряженные частицы в силу различных физических эффектов покидают исследуемую область. Поэтому на каждом временном шаге решения уравнений Максвелла необходимо производить коррекцию поля посредством решения уравнения Пуассона.

В данной модели помимо полей уравнений Максвелла реализован расчет статических полей (уравнения Пуассона), что делает возможным производить коррекцию поля пучка при нарушении закона сохранения заряда (потери частицы в пучке). Уравнение Максвелла решается совместно с уравнением Пуассона для собственного поля пучка. При решении уравнения Пуассона используется быстрое преобразование Фурье (БПФ).

Алгоритм БПФ хорошо известен и является классическим методом для решения уравнения Пуассона. На первом шаге алгебраическое уравнение представляется в конечно-разностной форме. На втором этапе рассчитываются коэффициенты разложения Фурье.

Решение уравнения Пуассона в декартовых координатах.

$$\begin{aligned}\frac{\partial D_x}{\partial x} + \frac{\partial D_y}{\partial y} &= \rho \\ D_x &= -\varepsilon_y \frac{\partial \Phi}{\partial x} \\ D_y &= -\varepsilon_x \frac{\partial \Phi}{\partial y}\end{aligned}$$

Представим уравнение в конечно-разностной форме:

$$\begin{aligned}\alpha_1^0(j) &= \frac{1}{\Delta x_1} \quad \alpha_2^-(j) = \frac{-1}{\Delta x_2} \quad \alpha_2^+(j) = \frac{1}{\Delta x_2} \\ (D_{x_{i+1/2,j}}^n - D_{x_{i-1/2,j}}^n) \alpha_1^0(j) &+ (\alpha_2^+(j) D_{y_{i,j+1/2}}^n + \alpha_2^-(j) D_{y_{i,j-1/2}}^n) = \rho_{i,j}^n \\ D_{x_{i+1/2,j}}^n &= \frac{-\varepsilon_x}{\Delta x_1} (\Phi_{i+1,j}^n - \Phi_{i,j}^n) \\ D_{x_{i,j+1/2}}^n &= \frac{-\varepsilon_x}{\Delta x_2} (\Phi_{i,j+1}^n - \Phi_{i,j}^n) \\ \xi^{+}(j) \Phi_{i,j+1}^n + \xi^0(j) \Phi_{i,j}^n + \xi^{-}(j) \Phi_{i,j-1}^n + \gamma^{+}(j) &[\Phi_{i+1,j}^n - \gamma^0 \Phi_{i,j}^n + \Phi_{i-1,j}^n] = -\rho_{i,j}^n\end{aligned}$$

Где $\xi^{+}(j) = \frac{\varepsilon_y}{\Delta x_2^2}$, $\gamma^{+}(j) = \frac{\varepsilon_x}{\Delta x_1^2}$, $\xi^0(j) = -2 \frac{\varepsilon_y}{\Delta x_2^2}$, $\gamma^0 = 2$, $\xi^{-}(j) = \frac{\varepsilon_y}{\Delta x_2^2}$

После выполнения преобразования Фурье получаем:

$$\xi^{+}(j) \sim \Phi_{k,j+1}^n + [\xi^0(j) - \gamma^{+}(j) 4 \sin^2(k\pi/N)] \sim \Phi_{k,j}^n + \xi^{-}(j) \sim \Phi_{k,j-1}^n = -\sim \rho_{k,j}^n, \quad (50)$$

Где N пространственная гармоника, а k номер моды.

Решение уравнения Пуассона для цилиндрических координат (R, Z) .

$$\frac{\partial D_z}{\partial Z} + \frac{1}{R} \frac{\partial R D_R}{\partial R} = \rho$$

$$D_z = -\varepsilon_z \frac{\partial \Phi}{\partial Z} \quad D_R = -\varepsilon_R \frac{\partial \Phi}{\partial R}$$

Представим уравнение в конечно-разностной форме:

$$\alpha_1^0(j) = \frac{1}{\Delta x_1} \alpha_2^{-}(j) = \frac{1}{2 x_2(j)} - \frac{1}{\Delta x_2} \alpha_2^{+}(j) = \frac{1}{2 \Delta x_2(j)} + \frac{1}{\Delta x_2},$$

г д е $x_2(j) = R_0 + \Delta x_2 j$

Частный случай:

$$j R_0 = 0, \quad j = 0, \quad \alpha_2^{+}(0) = \frac{1}{\Delta x_2}, \quad \alpha_2^{-}(0) = 0$$

$$(D_{x_{i+1/2}, j}^n - D_{x_{i-1/2}, j}^n) \alpha_1^0(j) + (\alpha_2^{+}(j) D_{R_{i, j+1/2}}^n + \alpha_2^{-}(j) D_{R_{i, j-1/2}}^n) = \rho_{i,j}^n,$$

Уравнение для потенциала имеет вид:

$$D_{z_{i+1/2}, j}^n = \frac{-\varepsilon_z}{\Delta x_1} (\Phi_{i+1, j}^n - \Phi_{i, j}^n)$$

$$D_{R_{i, j+1/2}}^n = \frac{-\varepsilon_R}{\Delta x_2} (\Phi_{i, j+1}^n - \Phi_{i, j}^n),$$

Тогда уравнения для потенциала будут иметь следующую форму

$$\xi^{+}(j) \Phi_{i, j+1}^n + \xi^0(j) \Phi_{i, j}^n + \xi^{-}(j) \Phi_{i, j-1}^n + \gamma^{+}(j) [\Phi_{i+1, j}^n - \gamma^0 \Phi_{i, j}^n + \Phi_{i-1, j}^n] = -\rho_{i, j}^n,$$

Где

$$\xi^{+}(j) = \frac{\varepsilon_R}{\Delta x_2} \left(\frac{1}{2 x_2(j)} + \frac{1}{\Delta x_2} \right), \quad \alpha^{+}(j) = \frac{\varepsilon_x}{\Delta x_1}, \quad \xi^0(j) = -2 \frac{\varepsilon_R}{\Delta x_2}, \quad \gamma^0 = 2, \quad \xi^{-}(j) = \frac{\varepsilon_R}{\Delta x_2} \left(\frac{1}{\Delta x_2} - 2 \frac{1}{2 x_2(j)} \right)$$

$$\gamma^{+}(j) = \frac{\varepsilon_z}{\Delta x_1^2}$$

$$x_2(j) = R_0 + \Delta x_2 j$$

Частный случай:

$$R_0 = 0, \quad j = 0,$$

Если преобразование Фурье выполнено, то получаем выражение аналогично (50).

Решение уравнения Пуассона в полярных координатах (R, θ) .

$$\frac{1}{R} \frac{\partial D_\theta}{\partial \theta} + \frac{1}{R} \frac{\partial R D_R}{\partial R} = \rho$$

$$D_\theta = \frac{-\varepsilon_\theta}{R} \frac{\partial \Phi}{\partial \theta}$$

$$D_R = -\varepsilon_R \frac{\partial \Phi}{\partial R}$$

Представим уравнение в конечно-разностной форме:

$$\alpha_1^0(j) = \frac{1}{\Delta x_2(j)} - \frac{1}{\Delta x_1} \alpha_2^-(j) = \frac{1}{2x_2(j)} - \frac{1}{\Delta x_2} \alpha_2^+(j) = \frac{1}{2x_2(j)} + \frac{1}{\Delta x_2},$$

где $x_2(j) = R_0 + \Delta x_2 j$

Частный случай:

$$R_0 = 0, j = 0, \alpha_2^+(j) = \frac{1}{\Delta x_2}, \alpha_2^-(0) = 0, \alpha_1^0(j) = \frac{2}{\Delta x_1 \Delta x_2}$$

$$(D_{x_{i+1/2}, j}^n - D_{x_{i-1/2}, j}^n) \alpha_1^0(j) + (\overset{i}{\alpha_2^+(j)} D_{R_{i, j+1/2}}^n + \overset{i}{\alpha_2^-(j)} D_{R_{i, j-1/2}}^n) = \rho_{i, j}^n$$

$$D_{\theta_{i+1/2}, j}^n = \frac{-\varepsilon_\theta}{x_2(j) \Delta x_1} (\Phi_{i+1, j}^n - \Phi_{i, j}^n)$$

$$D_{R_{i, j+1/2}}^n = \frac{-\varepsilon_R}{\Delta x_2} (\Phi_{i, j+1}^n - \Phi_{i, j}^n)$$

$$\xi^+(j) \Phi_{i, j+1}^n + \xi^0(j) \Phi_{i, j}^n + \xi^-(j) \Phi_{i, j-1}^n + \gamma^+(j) [\Phi_{i+1, j}^n - \gamma^0 \Phi_{i, j}^n + \Phi_{i-1, j}^n] = -\rho_{i, j}^n,$$

Где $\xi^+(j) = \frac{\varepsilon_R}{\Delta x_2} \left(\frac{1}{2x_2(j)} + \frac{1}{\Delta x_2} \right), \alpha^+(j) = \frac{\varepsilon_x}{\Delta x_1}, \xi^0(j) = -2 \frac{\varepsilon_R}{\Delta x_2}, \gamma^0 = 2, \xi^-(j) = \frac{\varepsilon_R}{\Delta x_2} \left(\frac{1}{\Delta x_2} - \frac{1}{x_2(j)} \right)$

$$\gamma^+(j) = \frac{\varepsilon_\theta}{\Delta x_2(j)^2 \Delta x_2}$$

$$x_2(j) = R_0 + \Delta x_2 j$$

Если преобразование Фурье выполнено, то получаем выражение аналогично (50).

После несложных преобразований матрица системы уравнений, связывающих коэффициенты Фурье для заряда и потенциала, приобретает трехдиагональный вид и решается на сетке методом матричной прогонки (метод Томаса). Наконец, решается задача синтеза Фурье для потенциала поля.

Для решения уравнения Пуассона со сложными граничными условиями реализован метод матрицы емкости. Основная идея метода матрицы емкости заключается в предварительном расчете матрицы емкости (C) , которая связывает потенциал и заряд в целом ряде узлов внутри области. Эти узлы могут быть электродами, поддерживаемыми при заданном потенциале, и цель состоит в вычислении поверхностного заряда, индуцированного на электродах окружающим пространственным зарядом. Решение получаем, дважды решая уравнение Пуассона, следующим образом. Сначала решаем уравнение Пуассона с нулевым зарядом на электродных узлах и записываем отклонение потенциала на электродных узлах от требуемых значений. Это отклонение после умножения на матрицу емкости и замены дает искомый поверхностный заряд в каждом электродном узле. Уравнение Пуассона решается снова с подсчитанным поверхностным зарядом, и теперь решение справедливо всюду в области, включая электродные узлы.

Способ непосредственного отыскания матрицы емкости состоит в помещении единичного заряда в каждый электродный узел по очереди (с нулевым зарядом на других электродных узлах) и решении относительно потенциала. Значения потенциала на электродных узлах образуют элементы одного столбца обратной матрицы емкости $A = C^{-1}$. Повторение этого процесса для каждого из J электродных узлов заполняет J столбцов матрицы A . Затем матрица емкости находится обращением матрицы A . В случае, если граничные условия области являются периодическими, то потенциал, обусловленный единичным зарядом, зависит только от вектора смещения между узлами потенциала и заряда. Таким образом, обратную матрицу емкости можно построить, исходя из единственного решения для потенциала от единичного заряда, помещенного в любую точку.

Коррекция поля с учетом решения уравнения Пуассона производится следующим образом. Зная расположение зарядов в фиксированный момент времени, можно найти создаваемое им статическое поле E_s решая уравнение:

$$\begin{aligned} \nabla^2 \Phi &= \frac{-\rho}{\epsilon_0} \\ E_s &= -\nabla \Phi \end{aligned} \quad (51)$$

С другой стороны, полное поле в уравнении Максвелла E_T связано с распределением заряда следующим уравнением:

$$\nabla E_T = \rho \hat{\epsilon}, \quad (52)$$

В этом случае мы можем сказать, что

$$E_T = -\nabla \Phi_L - \nabla \times A_T \quad (53)$$

Так что

$$\nabla^2 \Phi_L = \frac{-\rho^*}{\epsilon_0} \quad (54)$$

Таким образом, в случае выхода заряда из исследуемой области, ρ и ρ^* не совпадают. Правильное (скорректированное) выражение для поля задаётся формулой:

$$E_c = -\nabla \Phi - \nabla \times A_T \quad (55)$$

Выразим $\nabla \times A_T$ из (53) и, подставляя в (55) получим:

$$E_c = E_T - \nabla (\Phi - \Phi_L) \quad (56)$$

Решая уравнение Пуассона (57), найдем $(\Phi - \Phi_L)$ и скорректированное поле (58):

$$\nabla^2 (\Phi - \Phi_L) = \frac{-(\rho - \rho^*)}{\epsilon_0} \quad (57)$$

$$E_c = E_T - \nabla (\Phi - \Phi_L) \quad (58)$$

Интегрирование уравнения движения.

При решении уравнения движения заряженных частиц следует с помощью интерполяционной формулы вычислить напряженность поля $E(r)$ в произвольной точке r по значениям E_i в сеточных узлах r_i . Воспользуемся для этого той же формулой $w(r, r_i)$, которая использовалась для раздачи заряда в узлы сетки. Тогда

$$E(r) = V_g \sum_i E_i w(r, r_i). \quad (59)$$

Напряженность поля E_α , действующая на частицу с номером α , получается усреднением напряженности (59) по объему облака с выбранным коэффициентом формы $S_q(r, r)$:

$$E_\alpha = \int E(r) S_q(r, r_\alpha) dr = V_g \sum_i E_i S(r_i, r_\alpha), \quad (60)$$

Где $S(r, r_\alpha)$ та же весовая функция (функция усреднения), которая использовалась для вычисления сеточной плотности объемного заряда. Действие формулы (60) является обратным действием формул (1),(3). Последние преобразуют функцию с континуума точек на сетку, тогда как формула (60) распространяет функцию с сетки на континуум. Так же вычисляется магнитное поле B_α и действующая сила

$$F(r_\alpha) = F_\alpha = q(E_\alpha + v_\alpha \times B) \quad (61)$$

Выбор при расчете напряженности поля и плотности заряда одинаковых интерполяционных формул, а, следовательно, одинаковых весовых функций дает два преимущества. Во-первых, можно показать, что если при этом конечно-разностное уравнение Пуассона, связывающее напряженность поля с плотностью заряда, пространственно симметрично, т.е. записано в

центрально – разностных производных, т.е. исключается самодействие, изолированная частица в неограниченной или периодической области не испытывает действия сил. Во-вторых, использование различных весовых функций s_1 и s_2 может привести к гравитационной неустойчивости. Данная неустойчивость приводит к тому, что одинаковые частицы будут не отталкиваться, а притягиваться, как гравитационные массы.

Затем решается методом перешагивания непосредственно само уравнение движения облаков-частиц.

Уравнения движения в декартовых координатах.

$$\begin{aligned}\frac{dU_x}{dt} &= \frac{q}{m} E_x + \frac{q}{m\gamma} [V_y B_z - V_z B_y] \\ \frac{dU_y}{dt} &= \frac{q}{m} E_y + \frac{q}{m\gamma} [V_z B_x - V_x B_z] \\ \frac{dU_z}{dt} &= \frac{q}{m} E_z + \frac{q}{m\gamma} [V_x B_y - V_y B_x]\end{aligned}$$

$$\frac{dx}{dt} = \frac{V_x}{\gamma}$$

$$\frac{dy}{dt} = \frac{V_y}{\gamma}$$

$$\frac{dz}{dt} = \frac{V_z}{\gamma}$$

$$\begin{aligned} &\dot{\dot{}} \\ &\dot{\dot{}} \\ &\dot{\dot{}} \\ &\dot{\dot{}} \\ &\dot{\dot{}} \\ &\dot{\dot{}} \\ &\dot{\dot{}} \\ &\dot{\dot{}} \\ &\dot{\dot{}} \\ &\dot{\dot{}} \end{aligned}$$

(62)

Закон сохранения энергии.

$$\begin{aligned}\frac{1}{2} \frac{d}{dt} (V_x^2 + V_y^2 + V_z^2) &= \frac{q}{m} \{V_x E_x + V_y E_y + E_z V_z\} \\ \frac{dV_z}{dt} &= \frac{-q}{m} \left\{ \frac{\partial A_z}{\partial t} + \frac{\partial V_x}{\gamma} \frac{\partial A_z}{\partial x} + \frac{V_y}{\gamma} \frac{\partial A_z}{\partial y} + \frac{V_y}{\gamma} \frac{\partial A_z}{\partial y} \right\} = \frac{-q}{m} \frac{\partial A_z}{\partial t} \\ V_z + \frac{q}{m} A_z &= \text{const} = \frac{P_z}{m}\end{aligned}\tag{63}$$

Основные уравнения движения в цилиндрических координатах

$$\begin{aligned}
\frac{dV_R}{dt} &= \frac{V_\theta^2}{\gamma R} + \frac{q}{m} E_R + \frac{q}{m\gamma} [V_\theta B_z - V_z B_\theta] \\
\frac{dV_\theta}{dt} &= \frac{-V_\theta V_R}{\gamma R} + \frac{q}{m} E_\theta + \frac{q}{m\gamma} [V_z B_R - V_R B_z] \\
\frac{dV_z}{dt} &= \frac{q}{m} E_z + \frac{q}{m\gamma} [V_R B_\theta - V_\theta B_R] \\
\frac{dR}{dt} &= \frac{V_R}{\gamma} \\
\frac{dz}{dt} &= \frac{V_z}{\gamma} \\
\frac{d\theta}{dt} &= \frac{V_\theta}{\gamma R}
\end{aligned}$$

(64)

Закон сохранения энергии

$$\begin{aligned}
\frac{1}{2} \frac{d(V_R^2 + V_\theta^2 + V_z^2)}{dt} &= \frac{q}{m} \{V_z E_z + V_R E_R + V_\theta E_\theta\} \\
\frac{dRV_\theta}{dt} &= R \frac{q}{m} E_\theta + R \frac{q}{m\gamma} [V_z B_R - V_R B_z],
\end{aligned}$$

(65)

Где V_θ, B_R, B_z получаем следующим образом:

$$\begin{aligned}
E_\theta &= -\frac{\partial A_\theta}{\partial t} \\
B_z &= \frac{1}{R} \frac{\partial RA_\theta}{\partial R} \\
B_R &= -\frac{\partial A_\theta}{\partial z}
\end{aligned}$$

$$\frac{dRV_\theta}{dt} = \frac{q}{m} \left[-\frac{\partial RA_\theta}{\partial t} - \frac{V_z}{\gamma} \frac{\partial RA_\theta}{\partial z} - \frac{V_z}{\gamma} \frac{\partial RA_\theta}{\partial z} \right] = \frac{-q}{m} \frac{dRA_\theta}{dt}$$

Тогда, $RV_\theta + \frac{q}{m} RA_\theta = \text{const} = \frac{P_\theta}{m}$

При расчёте используется неявный метод с перешагиванием. Наличие релятивистского фактора сильно усложняет стандартный метод, вынуждая использовать дополнительные операции для получения необходимых в алгоритме промежуточных величин.

Для упрощения положим, $\{E_i, \beta_i\} = \frac{q}{m} \{\varepsilon_i, B_i\}$, где $\gamma = \frac{1}{\sqrt{1 - V^2/c^2}}$

Пусть в начале мы знаем $V_0^N, E_0, x_0^{N+1/2}$.

Тогда для импульса имеем:

$$\begin{aligned}
 V_0^{N+1/2} &= V_0^N + \rightarrow E_i^{N+1/2}(x_0^{N+1/2}) \frac{\Delta T}{2} \\
 \gamma^{N+1/2} &= \gamma(V_0^{N+1/2}) \\
 V_1^{N+1/2} &= V_0^{N+1/2} + \left[\frac{V_0^{N+1/2} + V_1^{N+1/2}}{2} \right] \times \beta(x_0^{N+1/2})^{N+1/2} \Delta T \\
 V_0^{N+1} &= V_1^{N+1/2} + e \rightarrow E_i^{N+1/2}(x_0^{N+1/2}) \frac{\Delta T}{2}
 \end{aligned}
 \tag{67}$$

Положение частицы на сетке

$$x_0^{N+3/2} \Rightarrow x_0^{N+1/2} + V_0^{N+1} \Delta T$$

На каждом временном шаге модель позволяет контролировать баланс энергии электромагнитного поля в исследуемой области в процессе расчетов

$P_{\sum_{\partial V} \frac{\partial w}{\partial t} + p=0}$. С этой целью на сетке вычисляются интегралы следующего вида:

$W = \frac{1}{2} \int_V dV (\rightarrow H \rightarrow B + \rightarrow E \rightarrow D)$ дающий величину запасенной в области энергии;

$P = \int_V \rightarrow j \rightarrow E dV$ позволяющий оценить преобразование мощности в системе;

$P_{\sum_i} \oint [\rightarrow E H] dS$ вычисляющий поток через заданную поверхность.

Описание блоков программы СУМА

Основные расчетные блоки в программе **СУМА** выделены в статическую библиотеку, поскольку это упрощает повторное использование кода. Программа, реализующая интерфейс с пользователем сама является пользователем этой библиотеки. За счет такой модульности, появляется возможность при необходимости создавать программы, использующие части модели, и решающие вспомогательные задачи. Другим примером применения библиотеки, могло бы быть использования ее для создания командного интерпретатора, который бы предоставлял возможность более гибко настраивать модель, чем это позволяет графический интерфейс, и управлять процессом расчета. В настоящее время библиотека подключается к пользовательскому интерфейсу на базе библиотеки **DISLIN** [16], а также к вспомогательной программе **SARMAT**. Программа **SARMAT** используется для расчета матрицы емкости, необходимой в качестве входного параметра для модели при расчёте поля пространственного заряда в структурах со сложной границей.

Рассмотрим более детальное описание отдельных блоков.

Библиотека СУМА

Модуль SYSTEM

Содержит наиболее общие параметры модели, такие как физические размеры, временные и пространственные шаги, и пр.

Модули VL DAT и FL DAT

Эти модули содержат структуры данных для частиц и для полей соответственно, а также процедуры для динамической инициализации памяти для массивов.

Модуль НАКОР

Содержит массивы для сохранения рассчитанных значений на каждой итерации, процедуры инициализации массивов, а также процедуры чтения/записи результатов работы программы на диск.

Управляющий драйвер - FLDVLS

Управляет итеративным процессом расчета, последовательно вызывая подпрограммы вычисляющие состояние системы на каждом шаге.

Блоки инициализации.

Блоки инициализации можно разделить на два типа: полевой блок и блок частиц. В полевом блоке производится инициализация полевых программ в соответствии с геометрией системы, наличием сложных областей и граничными условиями. В блоке частиц реализуется настройка программ ввода частиц.

FLDINI – инициализация полевого блока.

VLSINI – инициализация блока частиц.

Ввод частиц.

В данном блоке программы моделируется ввод частиц в исследуемую область.

VLSINJ - инжекция пучка

VLSEMI - эмиссии частиц

VLSGYR - гиротронная инжекция

VLSKNK - вторичная эмиссия частиц.

FLDANT – моделируются антенны и нагрузки.

Расчет заряда и токов.

В данном блоке производится раздача заряда и токов на сетку в соответствии с методом “облако в ячейке”.

VLSCUR - расчет заряда и токов.

Вычисление поля в системе с учетом нагрузки током и заданного внешнего поля.

В данном блоке решается полная система уравнений Максвелла на сетке.

FLDPRP – решение уравнения Максвелла,

FLDSL – решение уравнения Пуассона,

FLDEXF – задание внешних полей.

FLDTRS – реализация метода прогонки для решения СЛАУ.

Интегрирование уравнения движения.

В данном блоке производится интегрирование уравнения движения методом перешагивания.

VLSFLD – расчет полей, действующих на частицы

VLSADV – интегрирование уравнения движения

VLSLOC проверка соответствия динамики частиц с граничными условиями.

VLSORB – промежуточная подпрограмма, которая управляет интегрированием уравнения движения, и вызывает последовательно VLSFLD, VLSADV, VLSCUR и VLSLOC.

Баланс энергии в системе

На каждом временном шаге контролируется баланс энергии в исследуемой области.

FLDENR – расчет энергии запасенной в системе.

FLDJEN – расчет преобразованной мощности в системе.

FLDPYT – поток через границы.

Граничные условия

В данных блоках проверяются граничные условия.

FLDBNC – проверка граничных условий для пространственного заряда.

FLDBND – проверка граничных условий для полей и плотности тока.

Вывод графиков

Данный блок содержит набор подпрограмм для построения и вывода необходимой графической информации.

GRAPHIC – построение графиков эквипотенциалей, потенциала, полей, эмиттанса, заряда.

GRAPHIC1_7 – построение графиков зависимостей тока от времени, поля от времени.

SPACECHARGE – пространственные распределения разряда, картина траекторий частиц.

Описание параметров программы

Данные параметры задаются во входных файлах программа fldat.dat и vldat.dat, и настраивают, в основном, предельные значения тех или иных величин. Влияют на количество выделяемой памяти для внутренних структур данных.

NBP – максимальное количество частиц на каждом шаге.

NES – максимально количество эмитирующих поверхностей.

NETHT – количество идеально проводящих структур.

NGS – максимальное количество поверхностей гиротронной инжекции.

NMK – максимальное количество металлических блоков в структуре.

NIN — максимальное количество изолирующих блоков в структуре.

NIS – максимальное количество поверхностей инжекции.

NKS – максимальное количество поверхностей вторичной эмиссии.

NKN – максимальный размер массива для энергетического спектра.

NAN – максимальное размер массива для углового распределения частиц.

NSG – максимальное количество резистивных блоков.

NQP – количество компонент вектора описания частицы.

NSP – количество сортов частиц.

LL1,LL2 – размеры сетки.

KK1,KK2 – размеры массивов потенциала и заряда.

NN1,NN2 – размеры массивов полей и токов первой триады.

MM1,MM2 – размеры массивов полей и токов второй триады.

Списки инициализации (namelist)

Эти списки задаются в файле `viricator.dat` и описывают вариант расчета.

NLMAIN – основные параметры задачи: временной шаг, шаги по координатам, тип геометрии, и пр.

NFLVLS – настройки модели: количество шагов интегрирования, выбор включаемых подпрограмм.

NFLMSK – настройки маски, описывающей геометрию структуры.

NFLSIG – настройки проводящих элементов.

NFLCAP – настройка использования матрицы емкости.

NFLPYT – настройка диагностики вектора Пойнтинга.

NFLSPL – настройка полевых проб.

NFWAVE – настройка и диагностика волновых граничных условий.

NFLPRT – настройка вывода результатов полевой диагностики.

NFLEXF – настройка внешних полей и определение внешней электрической цепи.

NFLANT – настройки антенны и внешней нагрузки.

NFLSYS – сглаживание заряда и тока.

NVLBUF – настройки буферизации частиц.

NVLSYS – настройка использования полей.

NVLQMS – определение сортов частиц.

NVLRUN — настройки диагностики подпрограмм.

NVLDIA – диагностика частиц.

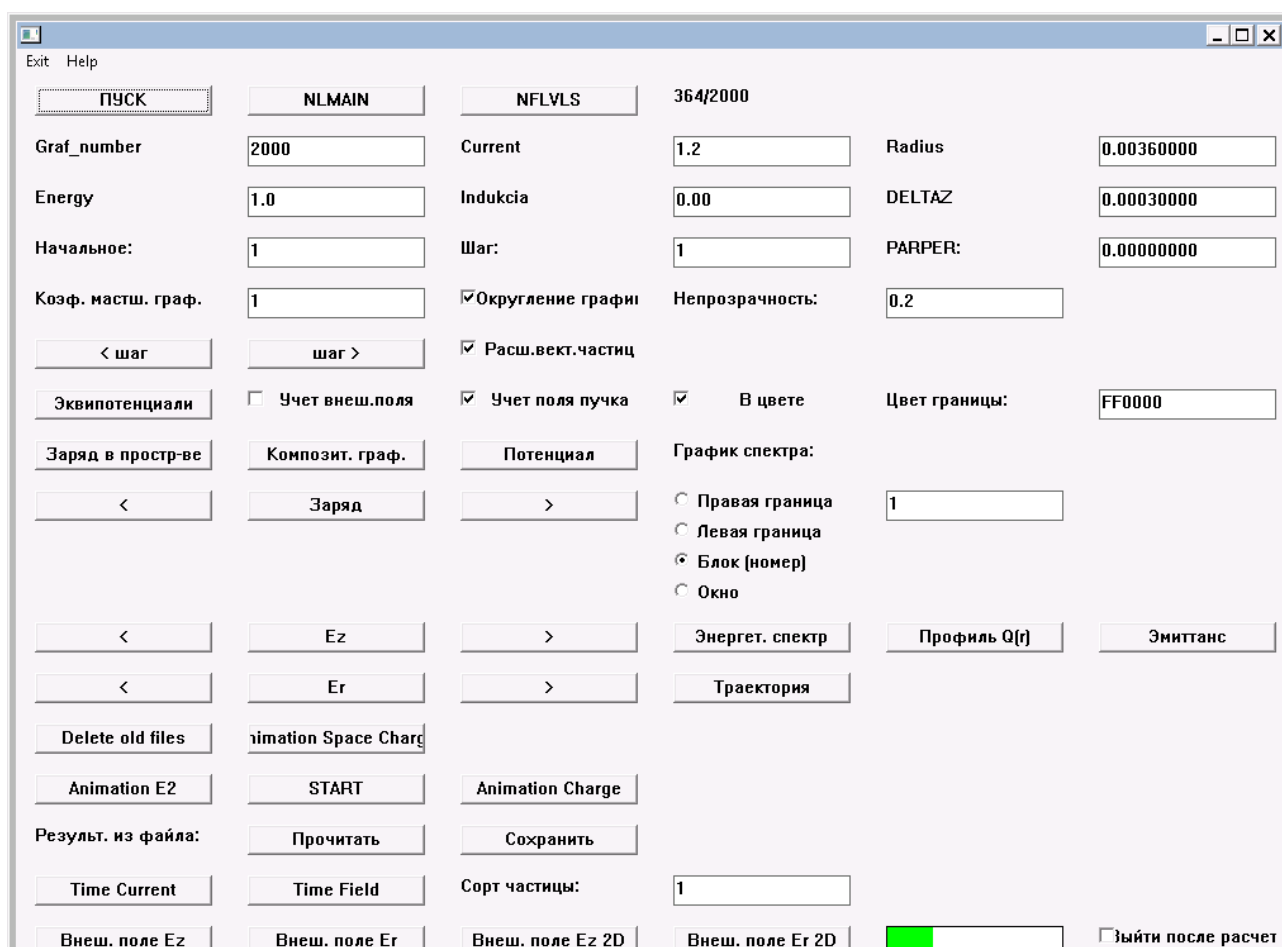
NVLINP – настройки начального распределения частиц.

NVLEQI – настройки вычисления изначального равновесного состояния частиц.

Модернизация интерфейса программы

Интерфейс программы представляет собой графический интерфейс на базе окон. Интерфейс и построение графиков осуществляется с помощью библиотеки DISLIN. Основное окно программы дает возможность запуска варианта расчета, заданного с помощью файлов входных данных программы, а также коррекции этих данных. В основном окне также сосредоточены элементы управления, предназначенные для настройки необходимой в данном варианте графической информации и ее вывода.

В рамках данной работы была проведена модернизация интерфейса программы, с целью повышения удобства использования и добавления новых возможностей. В этом разделе сделан обзор основных элементов интерфейса, а также дана справка по его использованию.



Описание элементов интерфейса SUMA

Запуск и процесс расчета

В левом верхнем углу расположена кнопка ПУСК, запускающая расчет заданного варианта. В основном окне присутствует так называемый «прогресс бар», что позволяет следить за временем выполнения данного варианта задания, а вверху в крайней справа позиции находится счетчик выполненных шагов. Этот счетчик устанавливается управляющим драйвером **FLDVLS** и показывает, на какой временной итерации находится программа.

В правом нижнем углу находится опция «выйти после расчета». При её активации (галка установлена), по окончании расчета приложение будет автоматически закрыто. Эта опция удобна тем, что может быть активирована с помощью аргумента командной строки '`--batch`', что позволяет запускать несколько вариантов расчетов в пакетном режиме. Когда расчеты занимают длительное время, это даёт возможность их последовательного выполнения без участия оператора.

В процессе расчета окно реагирует на действия пользователя, что позволяет менять некоторые настройки программы. Например, можно переключать описанную выше опцию «выйти после расчета».

Переключение шагов

Программа позволяет проводить детальный пошаговый анализ моделируемого процесса. Для этого в главном окне присутствует возможность задания вывода необходимых пространственных распределений тех или иных величин в фиксированный момент времени, либо их временных зависимостей в фиксированной координате. Для выбора момента, в котором мы выводим распределение, предусмотрено соответствующие интерактивные поля.

К выбору шага относится два текстовых поля: «Начальное» и «Шаг». В первом указывается значение шага, для которого выводится график при нажатии на кнопку вывода (напр. кнопка «Потенциал»). Во втором указывается количество шагов, на которое будет увеличиваться или уменьшаться шаг при нажатии на кнопки «< шаг» и «шаг >». При нажатии на эти кнопки для смещенного значения шага выводится график того типа, который был выведен последним.

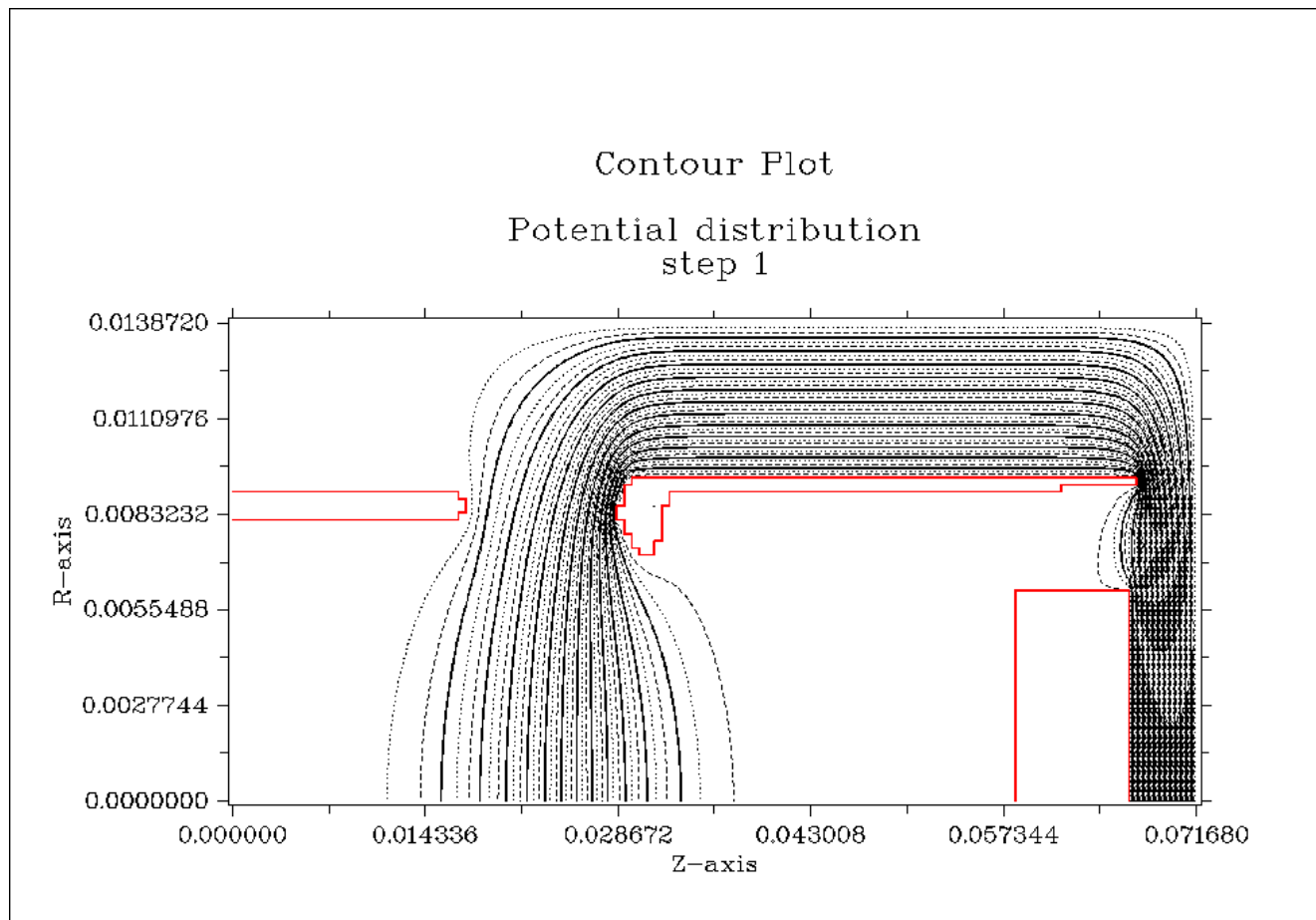
Мгновенные распределения

В программе можно вывести следующие мгновенные распределения:

- эквипотенциали (с учетом внешнего электростатического поля и без, в цвете и стандартный),
- распределение крупных частиц в пространстве,
- композитный график (эквипотенциали, наложенные на распределение частиц),
- распределение потенциала(трехмерный график),
- распределение заряда (трехмерный график),
- распределение компонент электрического поля(трехмерный график).

Ниже приведены примеры графиков.

График эквипотенциалей суперпозиции внешнего поля и собственного поля пучка. Поле пучка на много порядков меньше внешнего поля, поэтому его не видно на графике.



Тот же график, но в виде цветовой диаграммы.

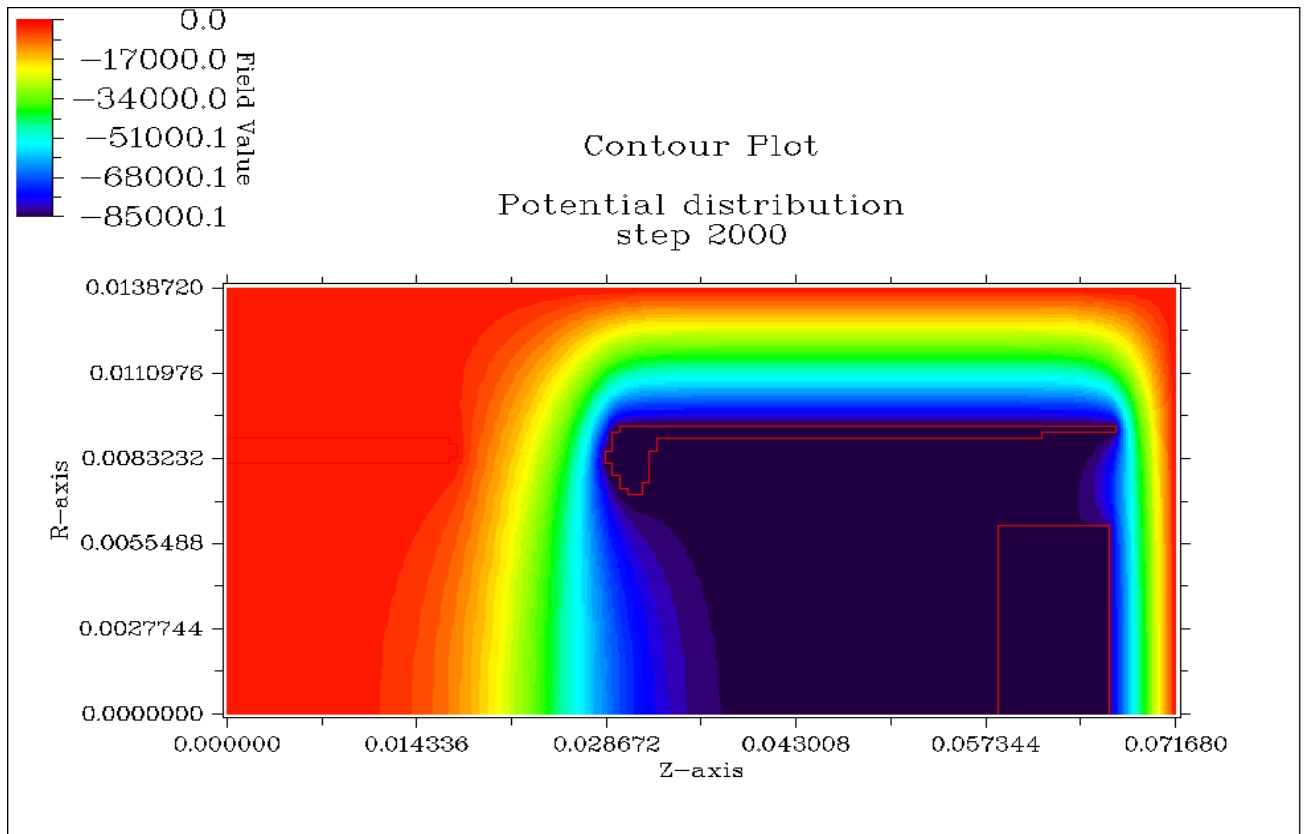
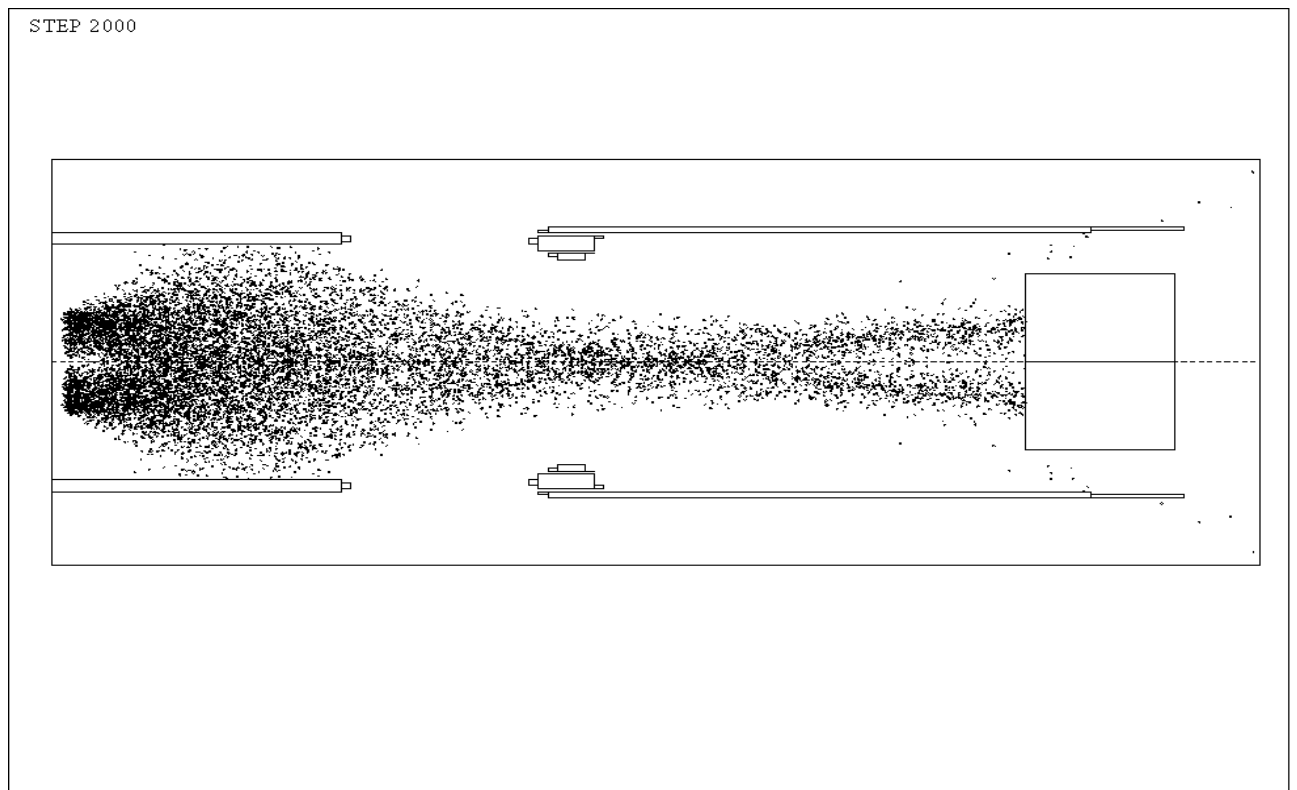
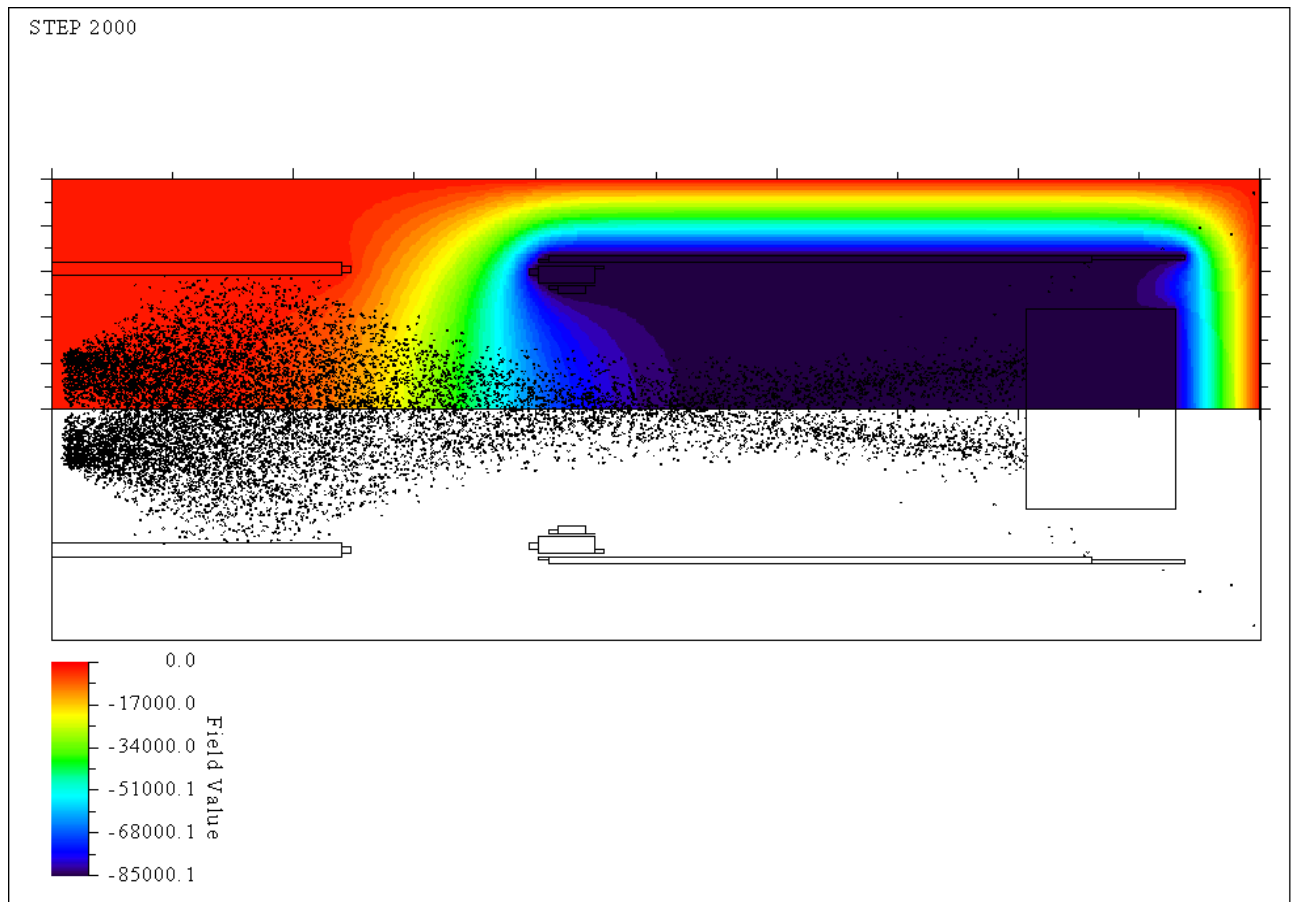


Рис. 5: Эквивалентная внешняя электростатическая поля и поля внутри (цветовая)

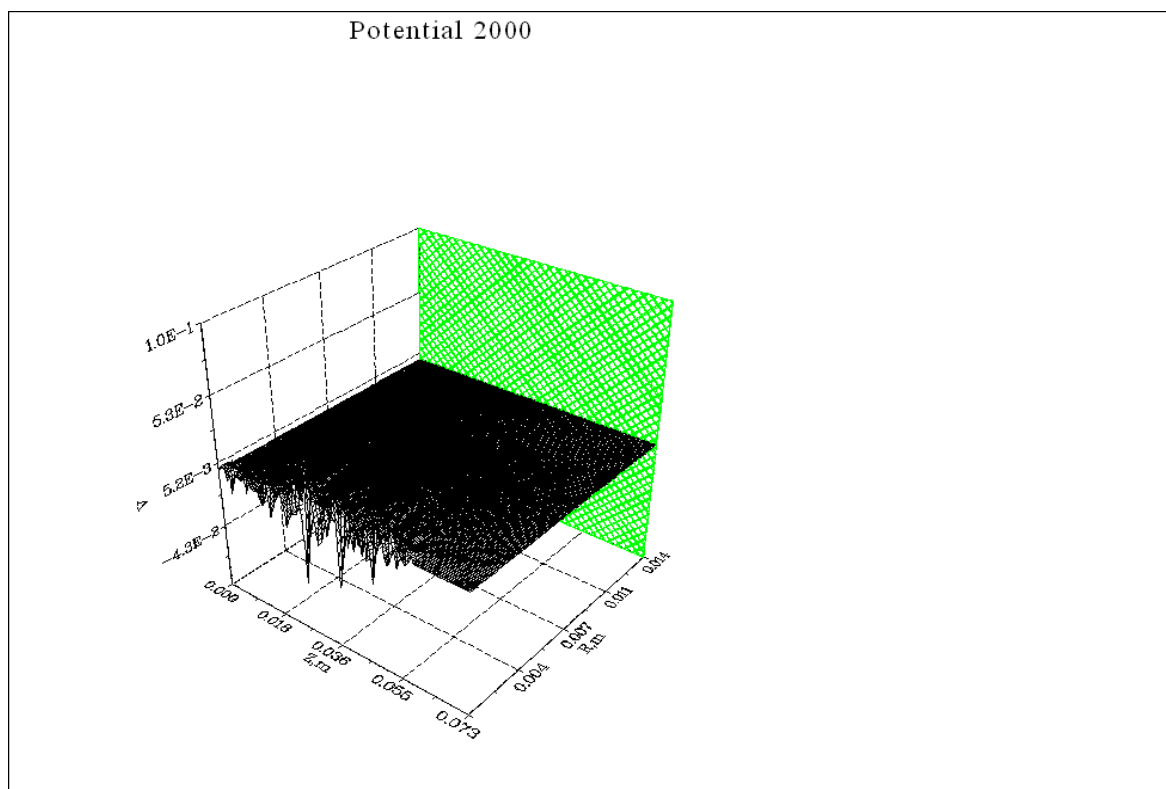


Распределение заряда в пространстве:

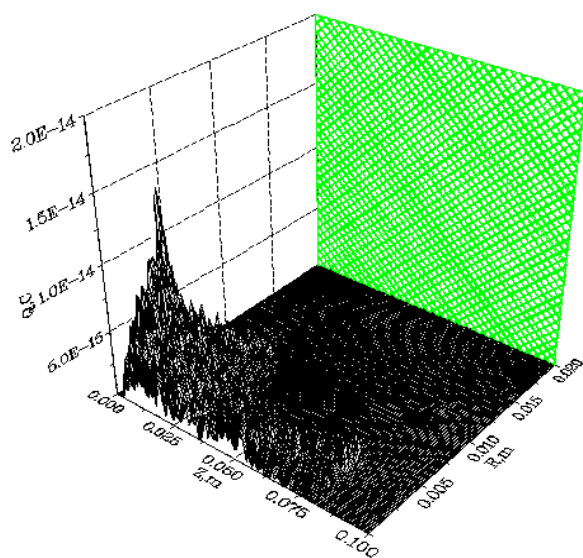
Композитный график - эквипотенциали, наложенный на пространственное распределение заряда.



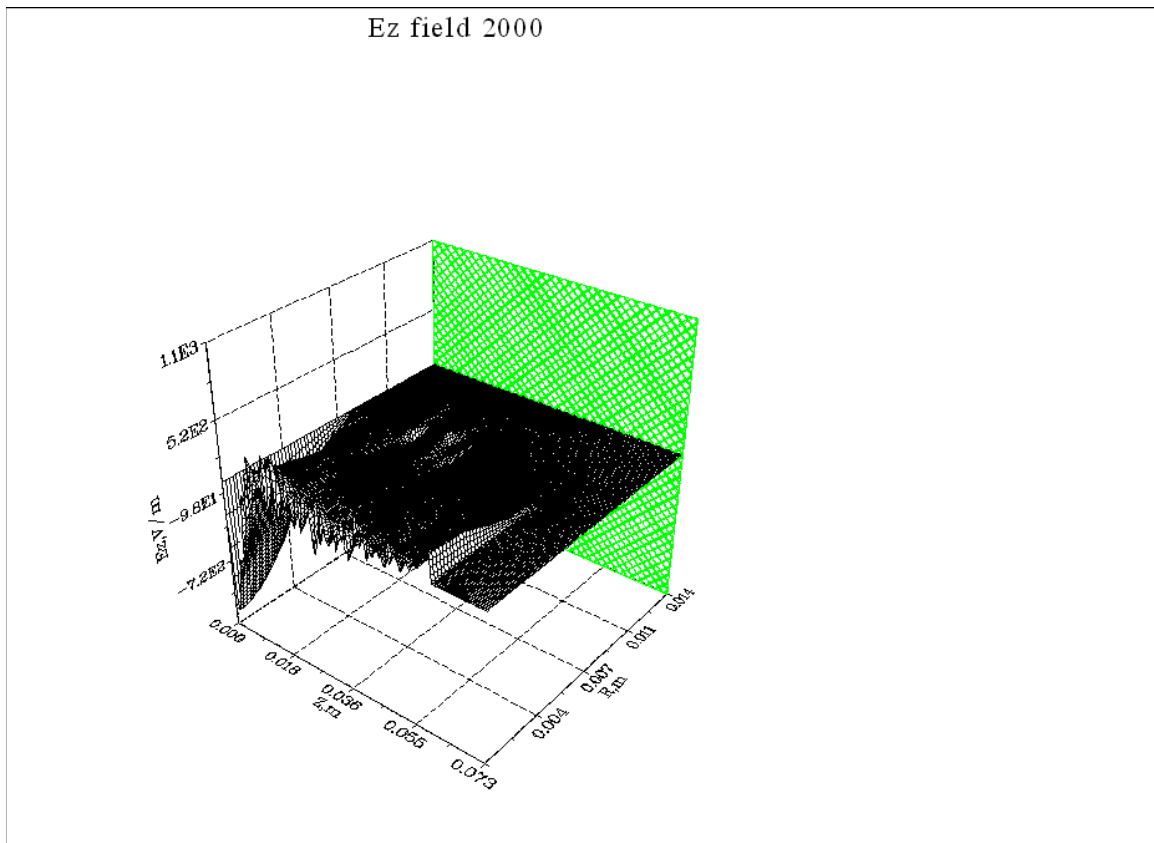
Трёхмерный график распределения потенциала.



Charge 2000



Трехмерный график распределения заряда.

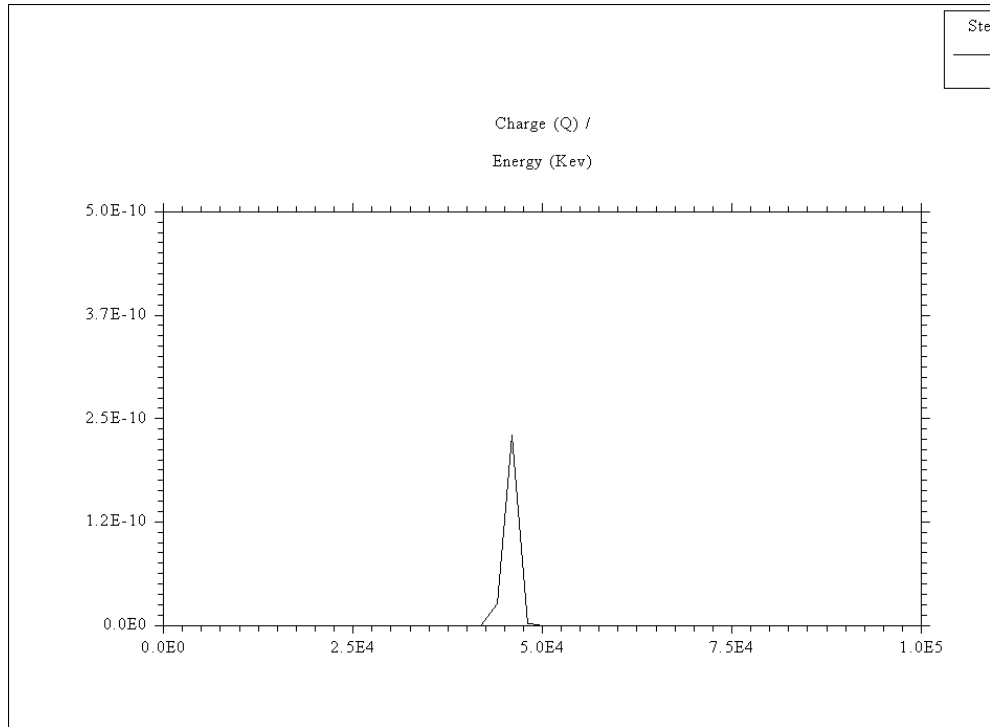


Трехмерный график распределения продольной компоненты электрического поля.

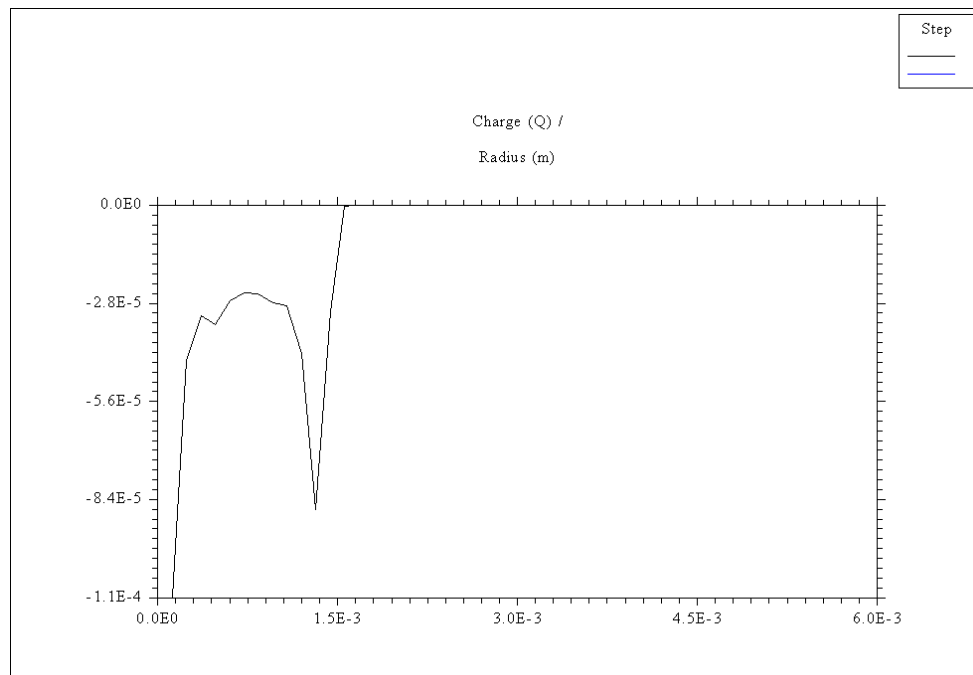
Распределения на границах

К этой категории можно отнести следующие графики: энергетический спектр пучка, профиль пучка, эмиттанс. Эти графики можно вывести на левой границе, правой границе, на внутреннем блоке структуры и в специально заданном окне. Координаты окна задаются в файле начальных данных до запуска расчета.

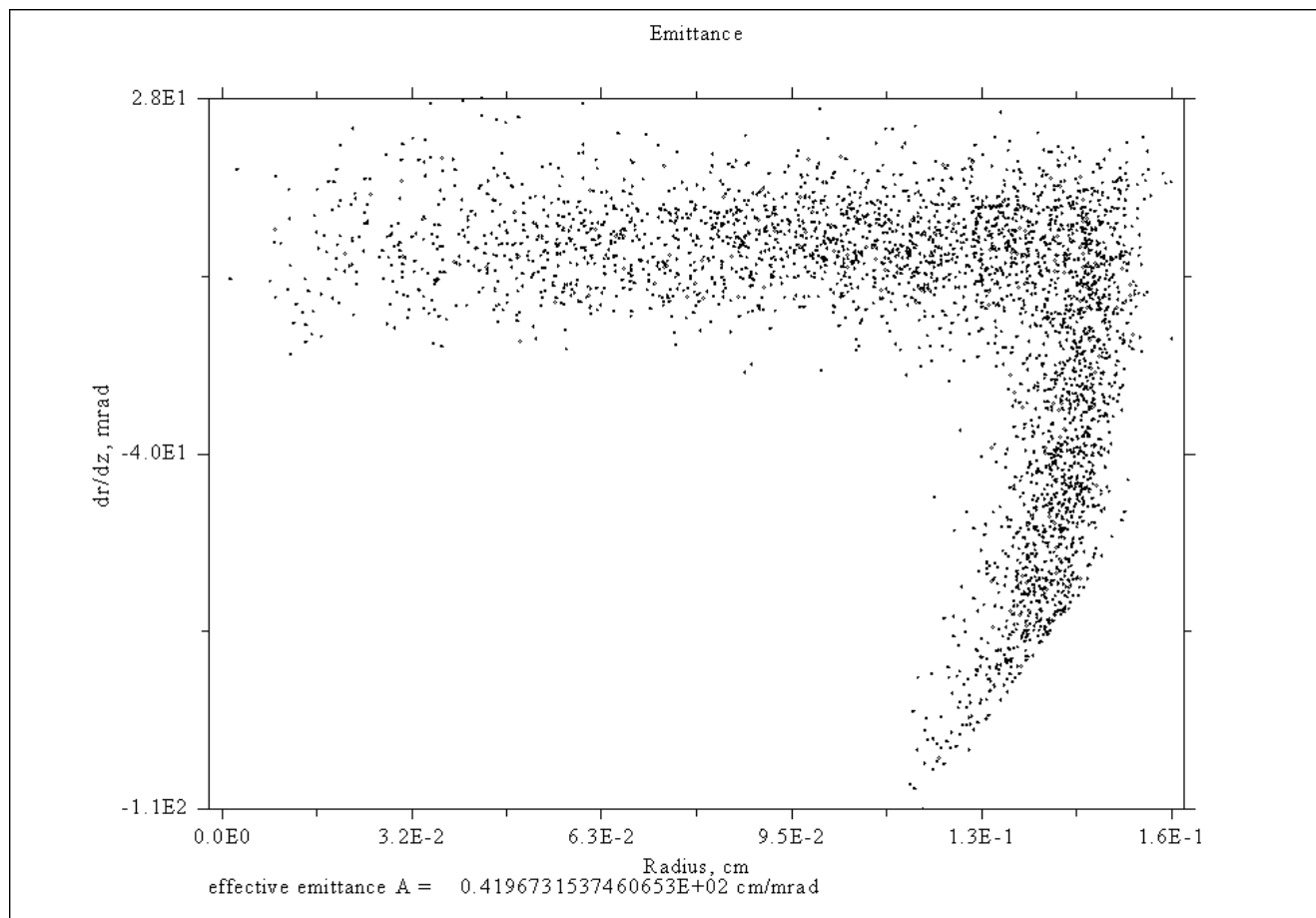
Энергетический спектр пучка.



Профиль пучка.



Эмиттанс. Под графиком выводится вычисленное значение эффективного эмиттанса.

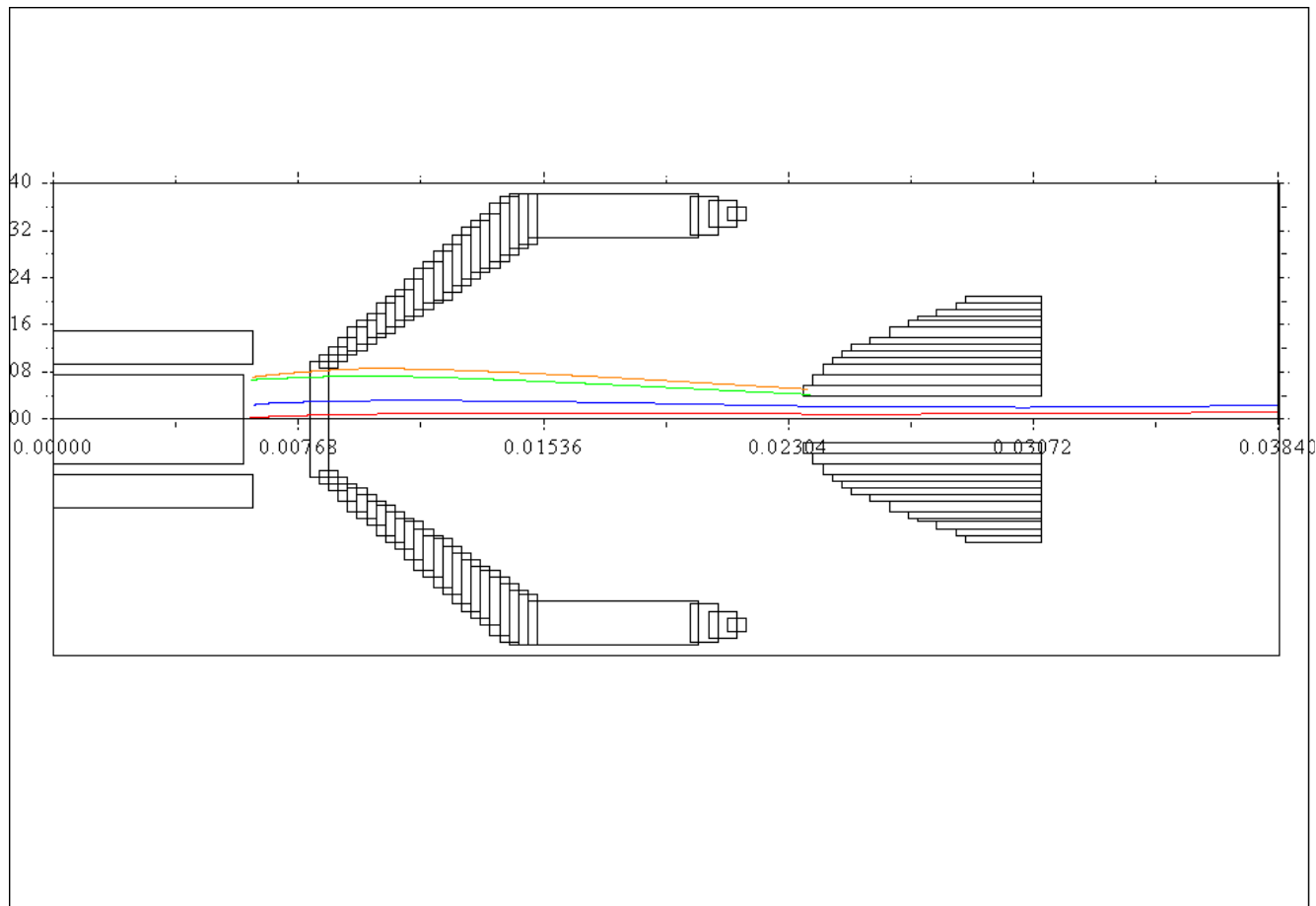


Траектории отдельных частиц

В случае, если расчет выполнен с сохранением расширенного вектора частиц (задается по умолчанию, см. ниже «Прочие элементы интерфейса») становится возможным вывести траектории отдельных частиц в системе. При нажатии на соответствующую кнопку в главном окне, программа выведет запрос идентификаторов частиц, для которых мы хотим увидеть траектории. В текстовое поле идентификаторы вводятся, разделяясь пробелами, запятыми, или точкой-запятой.

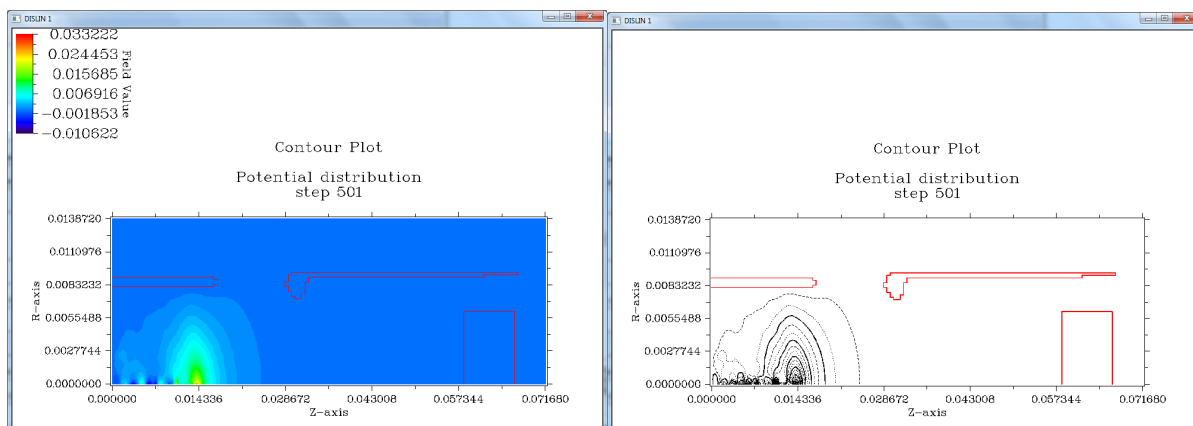
Рисунок 14: ввод номеров частиц

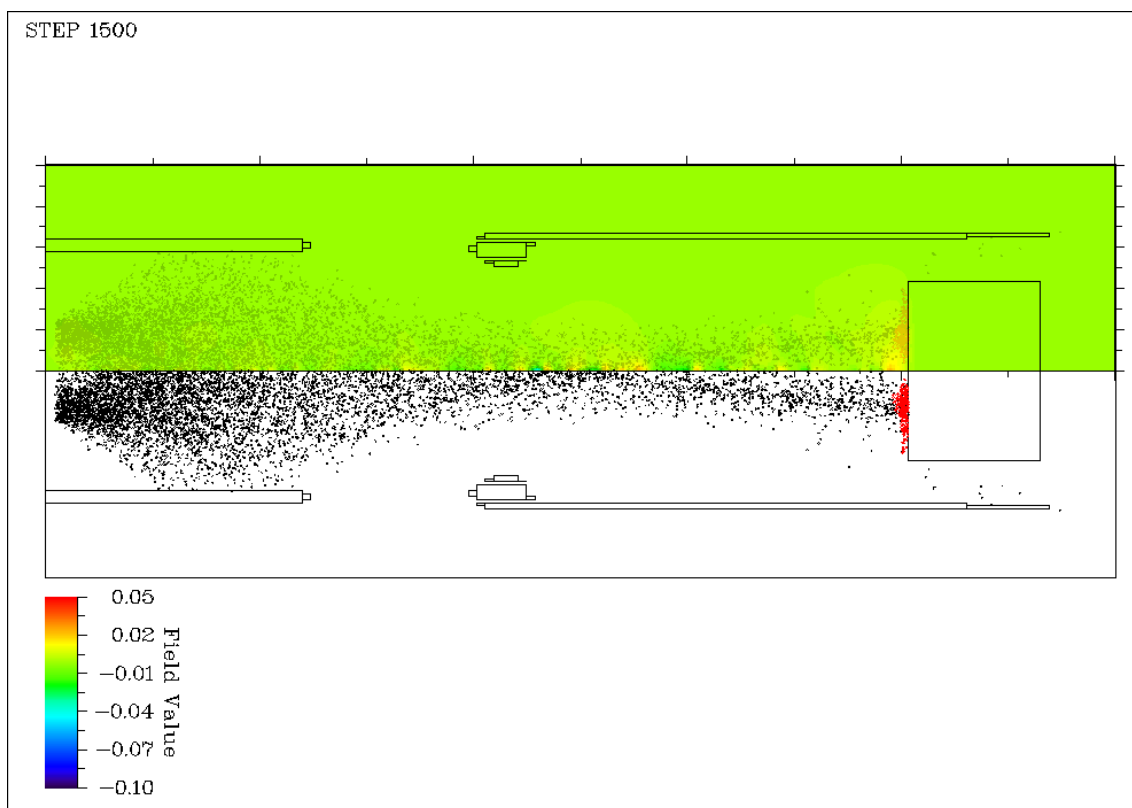
После нажатия на ОК программа выведет картину траекторий для всех частиц с заданными идентификаторами, если такие нашлись.



Настройки вывода графиков

В программу были добавлены новые элементы управления, позволяющие изменять настройки вывода графиков. Для графиков введена возможность округления предельных значений на осях, а также масштабирования этих графиков. Эквипотенциалы могут быть построены как в виде пунктирных линий, так и в виде цветовой диаграммы.





Для распределений связанных с зарядом введена возможность указания сорта частицы, для которой выполняется вывод распределения. Так, можно вывести отдельные графики распределения пространственного заряда по ионам и по электронам.

Отдельный интерес представляет добавленная возможность построения композитного графика, совмещающего в себе распределение пространственного заряда и картину эквипотенциалей в структуре. Для этого распределения имеется возможность указать прозрачность для распределения заряда в верхней половине. Таким образом можно подобрать наиболее удобный вид, как и с полностью отсутствующим зарядом на картине эквипотенциалей, так и вид, в котором наличие изображения заряда не будет мешать увидеть картину эквипотенциалей. Композитный график выводит все виды частиц, присутствующие в модели одновременно и разным цветом. Для композитного графика также действует настройка, позволяющая выбирать вид эквипотенциалей — цветовую диаграмму или пунктирную.

Прочие элементы интерфейса

В основной интерфейс было внесено построение графиков внешних электростатических полей, ранее доступные только из вспомогательной программы.

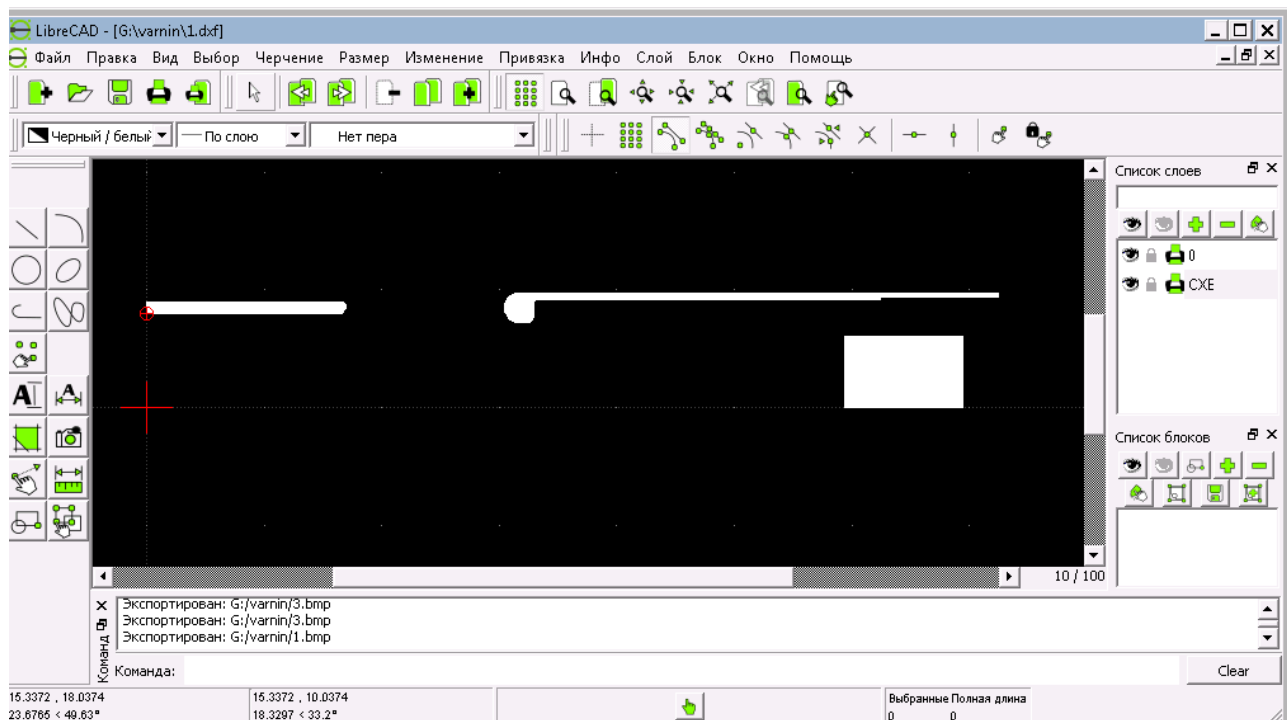
В главном окне присутствуют кнопки «Сохранить» и «Прочитать», используемые для сохранения и чтения результатов расчета на жесткий диск компьютера. Эти данные могут быть использованы для построения графиков после перезапуска программы.

Опция «Расш.вект.частиц» - в главном окне программы. По умолчанию — включено. Управляет количеством параметров частицы, которые необходимо запоминать в массив результатов расчета. При выключенной опции сохраняются только 3 координаты. При включенной сохраняются также 3 компоненты импульса и идентификатор частицы. Включение опции необходимо для вывода траекторий частиц. Выключение может понадобиться в случае, когда при расчете наблюдается дефицит оперативной памяти.

Дополнительная программа для описания геометрии

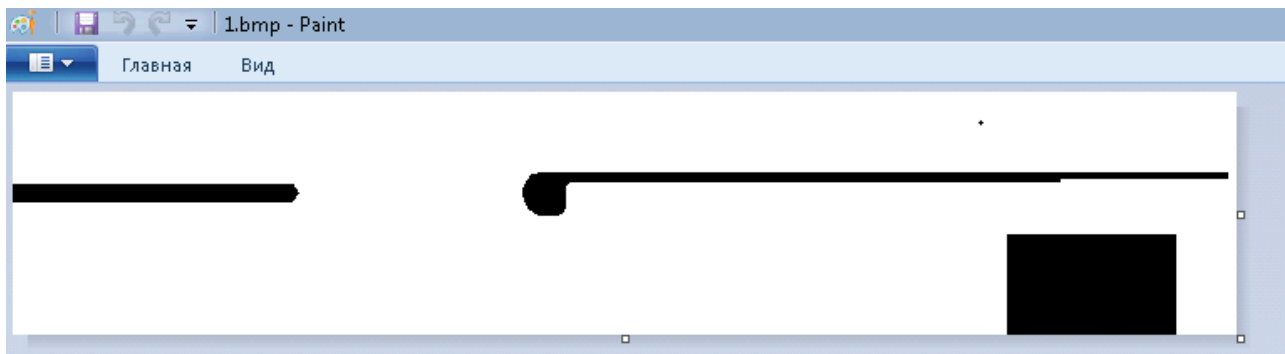
При задании начальных данных конкретного варианта наиболее трудоемкой задачей является задание границ. Входными данными для программ **САРМАТ** и **СУМА** среди прочего является текстовый блок, описывающий сложную геометрию внутренних блоков, например, электродов. Для упрощения и универсализации процесса ввода таких данных была написана специальная программа, позволяющая автоматизировать этот трудоемкий процесс.

Идея автоматизации заключается в том, что для задания геометрии используется двумерная CAD система. Такая система должна поддерживать возможность экспорта чертежа в растровый графический формат. В качестве примеров таких систем может выступать LibreCAD, QCAD, AutoCAD. В CAD системе чертеж задается в реальных размерах.

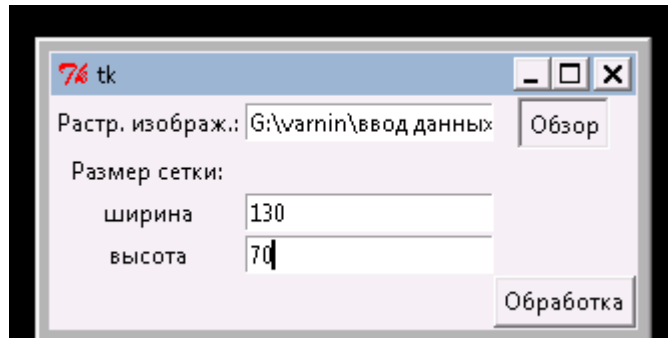


Блоки, закрашенные белым являются блоками, которые будут преобразованы во входной формат программы.

Чертеж из CAD системы экспортируется в растровую графику.



Затем растровое изображение передается специальной программе, написанной в рамках данной работы. Использованный язык — Python.



Эта программа, основываясь на размерах сетки генерирует входные данные для программы **СУМА**, которые можно использовать для описания геометрии. Сначала растровое изображение масштабируется к размеру сетки. Затем путем перебора программа ищет способ представить изображение минимальным количеством прямоугольных блоков.

Результат

MBND (1, 1) = 106, MBND (2, 1) = 123,
MBND (3, 1) = 0, MBND (4, 1) = 28,
MBND (1, 2) = 0, MBND (2, 2) = 29,
MBND (3, 2) = 38, MBND (4, 2) = 43,
MBND (1, 3) = 55, MBND (2, 3) = 111,
MBND (3, 3) = 44, MBND (4, 3) = 46,
MBND (1, 4) = 112, MBND (2, 4) = 129,
MBND (3, 4) = 45, MBND (4, 4) = 46,
MBND (1, 5) = 55, MBND (2, 5) = 58,
MBND (3, 5) = 36, MBND (4, 5) = 43,
MBND (1, 6) = 30, MBND (2, 6) = 30,
MBND (3, 6) = 39, MBND (4, 6) = 42,
MBND (1, 7) = 56, MBND (2, 7) = 58,
MBND (3, 7) = 35, MBND (4, 7) = 35,

Копировать

Оптимизация использования ресурсов машины

Вводные сведения о параллельных вычислениях

Одной из стоявших задач, являлось задействование технологий распараллеливания для сокращения времени расчета.

В задаче моделирования динамики пучков заряженных частиц часто производятся однотипные операции над множеством данных, например арифметические операции над значениями физических величин в каждом узле сетки, или операции над массивом координат или скоростей частиц. В данном случае наилучшим образом подходит концепция распараллеливания, называемая **SIMD** (англ. *single instruction, multiple data* — **оди́ночный поток команд, множественный поток данных, ОКМД**). Он заключается в том, что один и тот же машинный код выполняется одновременно для массива однотипных данных.

Рисунок 23: Иллюстрация модели SIMD

Если осуществлять вычисления данных на последовательном процессоре, один и тот же машинный код будет выполняться столько раз, сколько элементов нужно обработать. В виде программного кода это представляется оператором цикла (DO, WHILE). Модель SIMD обычно реализуется на аппаратном уровне. Ценой более сложного технологического устройства удастся получить возможность выполнять вычисления, используя

одну, более сложную машинную инструкцию, но выполняющую операцию сразу над множеством данных. Такое вычисление оказывается быстрее, чем аналогичные последовательные операции.

Рассмотрим наиболее распространенные реализации SIMD-модели.

- Параллельные расширения процессоров x86.

Сюда можно отнести такие технологии, как MMX, 3DNow, SSE, AVX. Все это дополнительные инструкции для процессоров, позволяющие внутри обычного x86 машинного кода задействовать SIMD операции.

- Технологии GPGPU.

GPGPU (англ. *General-purpose graphics processing units* — «GPU общего назначения») - это технология заключается в применении графического процессора (англ. *graphics processing unit, GPU*) для произвольных расчетов, не связанных с ускорением графических операций. Это стало возможным в результате эволюции графических процессоров, которые в определенный момент достигли возможности вычислений с точностью, не уступающей такой у процессора.

Самые известные примеры технологий GPGPU – CUDA и OpenCL. CUDA является вендор-специфичной технологией, принадлежащей компании NVIDIA. По этой причине она поддерживается только видеокартами от компании NVIDIA. Технология OpenCL не является вендор-специфичной, и развивается множеством компаний одновременно (в том числе, и NVIDIA). Поэтому с OpenCL можно использовать как видеокарты NVIDIA, так и видеокарты AMD. Однако по развитию возможностей технология OpenCL отстает, и машинный код, генерируемый на ее основе оказывается медленнее, чем у CUDA.

Таблица 3: Сравнительная характеристика подходов SIMD

Расширения x86	GPGPU
<p>Интегрируется в обычный x86 код, за счет чего может произвольным образом внедряться в программу, не требуя ее переработки.</p>	<p>Требует выделения параллельного кода в отдельные подпрограммы, которые затем загружаются на GPU. Такие программы называются «ядром». Эту особенность необходимо учитывать при написании программы.</p>
<p>Работает с данными в регистрах. Копирование из/в память происходит быстро.</p>	<p>Требует копирования из основной оперативной памяти в оперативную память видеокарты. Копирование занимает больше времени, чем в случае расширений x86.</p>
<p>Размеры векторов данных для операций это несколько фиксированных значений. Размеры векторов ограничены возможностями технологии. Иными словами, степень распараллеливания операций не растет.</p>	<p>Векторы могут быть произвольной длины. Технология подразумевает возможность использования разных видеокарт, содержащих произвольное количество параллельных процессоров. Операции над векторами данных будут автоматически адаптированы под возможности вычислительной системы. Степень распараллеливания может быть увеличена с помощью замены видеокарты.</p>
<p>Широко применяется в мультимедиа обработке — обработке звука, видео и других сигналов.</p>	<p>Широко применяется в научных расчетах и в других приложениях требующих специализированных математических расчетов.</p>

Автоматизация внедрения параллельных расчетов при помощи компилятора

Язык программирования FORTRAN (далее Фортран) является высокоуровневым компилируемым языком, предназначенный в основном для инженерных и научных вычислений. Синтаксис определения массивов и операций с массивами определен так, что массивы чисел оказываются достаточно абстрактными сущностями. Более абстрактными, чем в языке C, где массив является всего лишь синонимом указателя в памяти. В Фортране же указатели являются редко используемой возможностью, и по умолчанию, без указания ключевого слова *target*, массивы в Фортране нельзя использовать вместе с указателями.

За счет такой абстракции в Фортране удастся достичь очень прозрачных способов автоматической оптимизации работы с массивами, включая автоматическое внедрение параллельных вычислений, не требующее глубокого понимания деталей работы параллельных вычислений. Это соответствует философии Фортрана, который изначально был создан, чтобы научные работники, не являющиеся специалистами по вычислительной технике, могли писать программы и не вникать в детали реализации конкретной аппаратной платформы.

Большинство актуальных компиляторов языка Фортран поддерживают автоматическое распараллеливание с помощью расширений для x86. При этом технологию GPGPU поддерживает единственный компилятор — PGI Fortran от компании Portland Group (теперь ставшей частью NVIDIA). Этот компилятор поддерживает технологию CUDA. В данной работе применяется компилятор PGI Fortran 13.10. [19]

В то время, как для использования расширений x86 изменять программу не нужно, для распараллеливания CUDA перестройка программы необходима. При использовании технологии типа GPGPU данные должны находиться в памяти видеокарты. В случае, когда над одними и теми же данными нужно осуществлять последовательно и операции на видеокарте, и

операции на процессоре, накладные расходы на копирование значительно возрастают. В таком случае, накладные расходы могут превысить выигрыш от распараллеливания. По этой причине оказывается невозможным задействовать автоматизацию распараллеливания CUDA в компиляторе, без предварительной переработки кода так, чтобы исключить чрезмерное копирование данных между ОЗУ и видеокартой. В случае программы СУМА это означает необходимость переноса всех рабочих массивов, которые используются в пределах каждой итерации (массивы значений полей, характеристик частиц на данном шаге, и пр.) в память видеокарты. Копирование в ОЗУ должно осуществляться только в момент сохранения значений этих массивов уже по окончании каждой итерации (это происходит в подпрограмме FLDSMP).

Эти ограничения в ближайшем будущем может быть преодолены технологически. Компания AMD производит гибридные процессоры APU (англ. *Accelerated Processing Unit*, букв. «ускоренное обрабатывающее устройство»). Процессоры APU – это процессоры физически совмещенные с GPGPU модулем. AMD анонсировали выпуск в 2014 году процессоров нового поколения под кодовым названием Kaveri. В этих процессорах будет реализована поддержка адресации к единому адресному пространству, то есть копирование памяти требоваться не будет. С другой стороны, выпущенный APU все еще будет отставать по возможностям от топовых видеокарт, и производительность самих вычислений у него будет ниже.

В рамках данной работы рассматривалась возможность внедрения распараллеливания на базе CUDA в программу СУМА. Были сделаны некоторые шаги в этом направлении. Однако задача оказалась слишком объемной и не была до конца выполнена. Сложность состоит в том, что для эффективного задействования параллельности CUDA необходимо переработать весь код, исполняющийся в рамках отдельных итераций, переместив все данные на видеокарту. Все циклы обработки нужно как минимум обрмить директивами автоматического распараллеливания, а в

менее удачных случаях их нужно переписывать, поскольку они могут использовать неприемлемые по эффективности выполнения на GPU алгоритмы. Например, не рекомендуется помещать в ядра CUDA разветвления и непоследовательные, не выровненные по границе блоков в 32 элемента обращения к элементам массивов [17]. Так, например, поиск минимума или максимума с помощью последовательного прохода по массиву не используется в CUDA, поскольку такой код будет значительно менее эффективным, чем на CPU. Вместо этого используются параллельные алгоритмы, называемые редукциями.

К практическим результатам можно отнести включение в код программы интерфейсов к библиотеке cufft [18] – быстрое преобразование Фурье с помощью CUDA, и использования этой библиотеки для замены имеющегося последовательного преобразования Фурье. С такой заменой программа корректно отработала, однако показала падение производительности, чего и следовало ожидать, поскольку в такой ситуации выигрыш за счет распараллеливания не превысил расходов времени на копирование массивов на видеокарту и с нее.

В итоге для оптимизации была применена автоматизация распараллеливания на базе расширений процессоров x86. Это также потребовало модификации кода в некоторых критичных местах, чтобы «помочь» компилятору провести автоматическое распараллеливание. Например, важно было расставить порядок индексов массивов таким образом, чтобы числа, над которыми должны совершаться параллельные операции, располагались последовательно в памяти.

Метод оптимизации был следующий. Сначала с помощью профилировщика PGPROF (см. ниже) определялись наиболее длительные по времени выполнения подпрограммы. Затем проводился обзор алгоритма подпрограммы. Уже на этом этапе становилось заметно, какие места в программе можно оптимизировать. В отдельных случаях производилось разбиение крупных подпрограмм на более мелкие логические блоки. Это позволяло определить, в ка-

кой именно части крупной подпрограммы происходят наиболее интенсивные вычисления. Попутно сокращалось дублирование одного и того же кода, повторяющегося в программе, что сокращает необходимый труд, по внесению изменений в алгоритмы, и снижает вероятность появления ошибок. В случае изменения структуры программы, профилировка проводилась повторно. Затем использовался отчет компилятора, в котором он сообщает, какие участки программы удалось оптимизировать, в том числе распараллелить. В соответствии с отчетом производится попытка устранить проблемы, не позволившие компилятору произвести оптимизацию в каждом конкретном случае. Затем проводится новая профилировка, проверяются изменения и код корректируется снова уже в соответствии с новой информацией.

Используемые настройки компилятора

Следующие опции компилятора добавлены вручную к тем, которые используются по умолчанию.

Таблица 4: Опции компилятора

Опция	Пояснение [19]
-Mlarge_arrays	Позволяет передавать в подпрограммы массивы размером более 2 Гб в качестве аргументов.
-Minfo -Minfo=ipa	Включает отчеты компилятора по проведенным оптимизациям
-fast	Включает ряд автоматических оптимизаций, таких как использование инструкций SSE, замена идиом языка, и др.
-Mipa=inline	Включает возможность автоматического встраивания функций на место их вызова.

Анализ проведенных оптимизаций программы СУМА

В состав среды разработки PGI Fortran входит профилировщик PGPROF — средство анализа производительности программы и ее отдельных частей [20]. Профилировщик широко использовался в данной работе. Его основными применениями были измерение общего времени расчета, а также измерение времени работы отдельных подпрограмм. Профилировщик предоставляет информацию о количестве вызовов подпрограмм, что может быть также полезной информацией для оптимизации — например для решения о встраивании той или иной функции на место ее вызова. Профилировщик поддерживает исследование времени выполнения программы на GPU.

Характеристики машины, на которой проводились измерения приведены в таблице 5.

Таблица 5: Характеристики машины

Процессор	ОЗУ	ОС
Intel Core i5 750, 2.67 ГГц, 4 ядра	8 Гб, DDR3	Windows 7 Ultimate, 64 bit

Таблица 6: Временной профиль расчета

Вариант	Максимальное количество крупных частиц в системе (приблизительно)	Размеры сетки	Временной шаг, с	Время расчета без оптимизаций, с	Время расчета, после оптимизации, с
			Максимальное время модели, нс		
Электронная пушка для ЛУЭ на 10 МэВ	67200 ионов,	130x70	5e-13	4270,5	1522,7
	64600 электронов		1		
	8100	514x280	1.2e-13	2933,1	733,5

мощностью пучка до 20 кВт			0.12		
	9500	130x70	5e-13	428,0	138,7
			1		
Ионно- оптическая система нейтронной трубки ГНТ1- 32	6700	130x70	5e-11	187,6	50,7
			100		

Проведенные оптимизации позволили в среднем сократить время выполнения в 3 раза. Это достигнуто в первую очередь за счет оптимизации подпрограммы сохранения результатов FLDSMP. Ее переработка позволила векторизовать циклы в ней, а также более быстро работать с памятью. Также некоторые автоматические оптимизации были совершены компилятором с кодом, не потребовавшим никаких изменений.

Дальнейший анализ профиля программы показал, что программа имеет две наиболее интенсивно используемых подпрограммы. Одна относится к полевой части расчетов — `fldsmt_smooth_and_mask`, другая к определению попадания частиц на блоки структуры — `block_boundaries_corrections` из модуля VLSCUR. Соответственно процент времени выполнения этих подпрограмм от общего времени выполнения программы зависит от соотношения между количеством частиц и размером сетки. В вариантах с большим количеством частиц оптимизация будет наиболее эффективной для функции `block_boundaries_corrections`, а в вариантах с большой сеткой - `fldsmt_smooth_and_mask`.

Другое произведенное изменение, которое стоит отметить, заключается в переводе массива для хранения набора частиц в системе на всех шагах на динамическое выделение памяти. За счет этого заметно сократилось

использование памяти, в котором ранее статически выделяемая память в значительной части просто не использовалась.

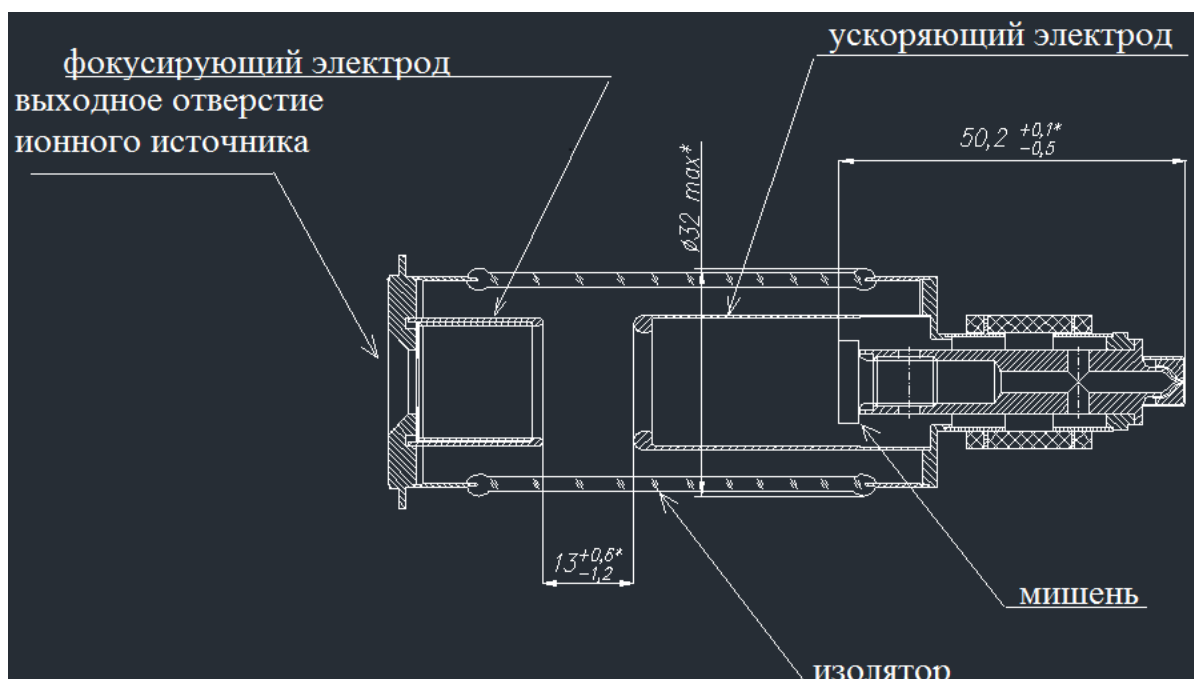
Таблица 7: Требования к оперативной памяти

Вариант	Используемая память до оптимизации, КБ	Используемая память после оптимизации, КБ	Соотношение до к после
Электронная пушка для ЛУЭ на 10 МэВ и мощность пучка до 20 кВт	8 422 460	4 774 128	1,76
	8 548 692	5 270 176	1,62
	2 299 436	772 012	2,98
Ионно-оптическая система нейтронной трубки ГНТ1-32	4 173 016	692 160	6,02

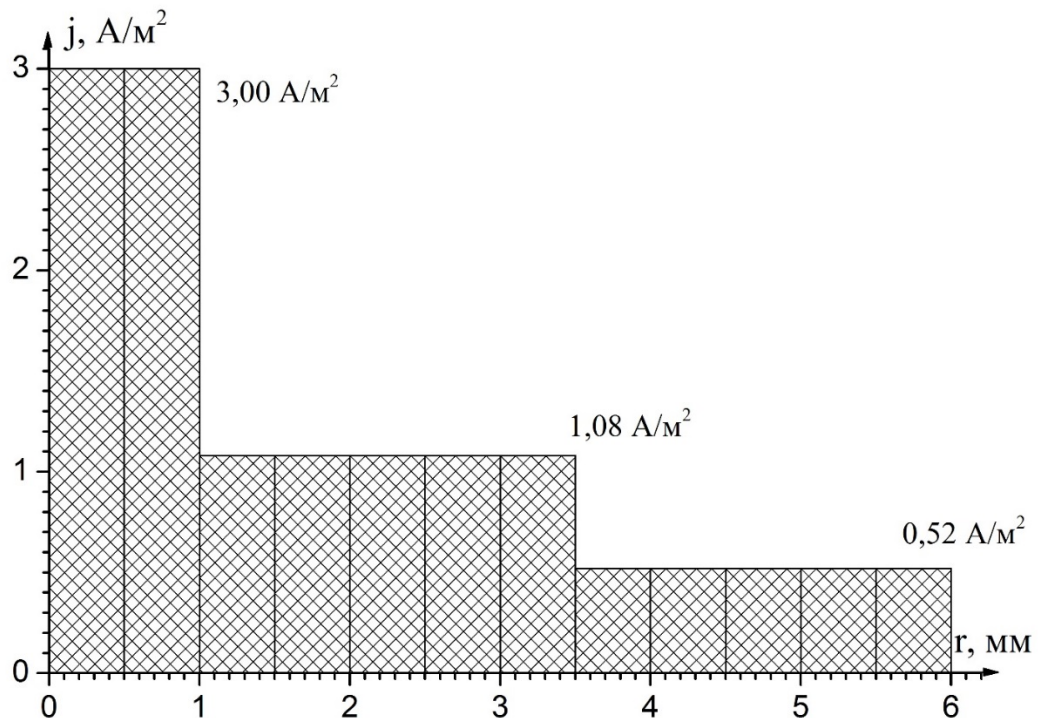
Скачок для последнего варианта объясняется тем, что память выделялась еще и для второго сорта частиц — электронов, хотя в этом конкретном варианте их в системе не было. Поэтому выделенная память была вдвое больше, чем у примерно аналогичного по количеству частиц и по размеру сетки 3-го варианта из таблицы. Перевод на динамическое выделение памяти решил эту проблему автоматически. В целом же видно, что чем меньше число частиц в системе по отношению к размеру сетки, тем больше был выигрыш.

Моделирование динамики дейтронного пучка в нейтронной трубке

В качестве примера использования кода СУМА выполнено исследование динамики дейтронного пучка в ионно-оптической системе нейтронной трубки ГНТ1-32, представленная на рис. 24. Работы проводились в рамках договора с ВНИИА им. Духова.



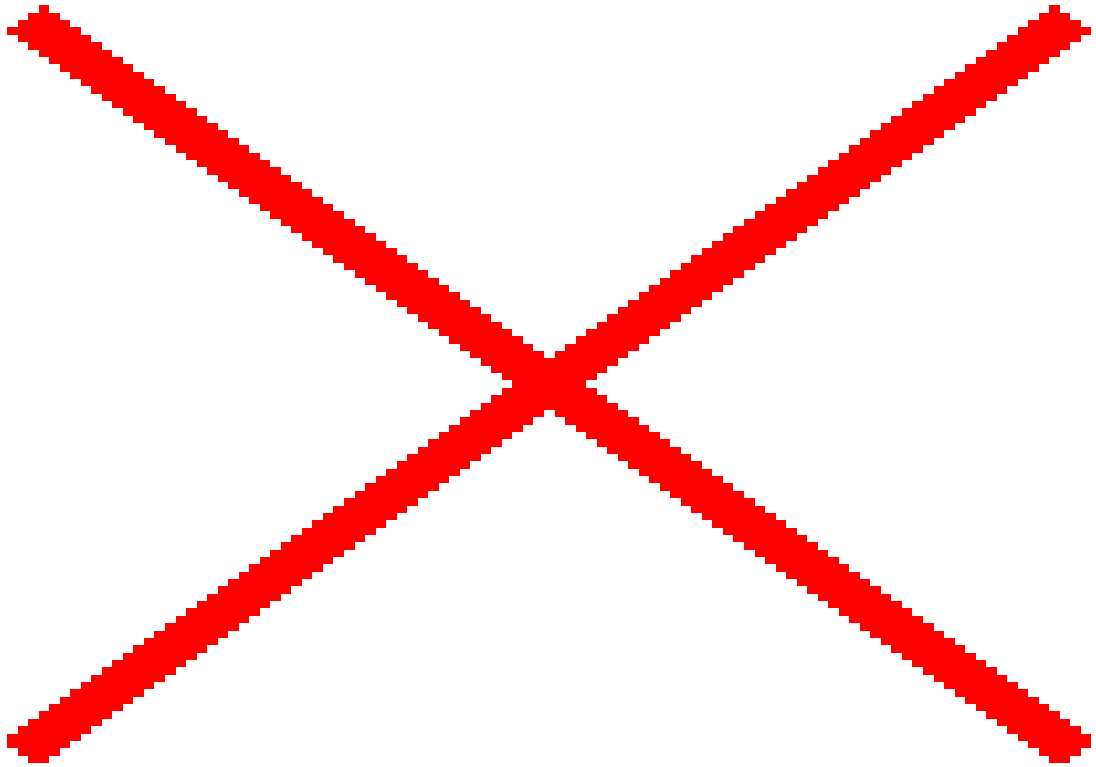
Исследовалось влияние параметров дейтронного и вторичных электронных пучков (начальная энергия, распределение скоростей по величине и углу вылета и т.д.) на выходные характеристики дейтронного пучка (спектр, профиль, угловая расходимость). Предпринималась попытка подобрать начальное распределение таким образом, чтобы профиль дейтронного пучка на мишени был максимально близок к полученному экспериментально (рис. 25).



Проведенное моделирование показало, что при следующих параметрах инжекции дейтронного пучка, выходное распределение наиболее близко к экспериментальному:

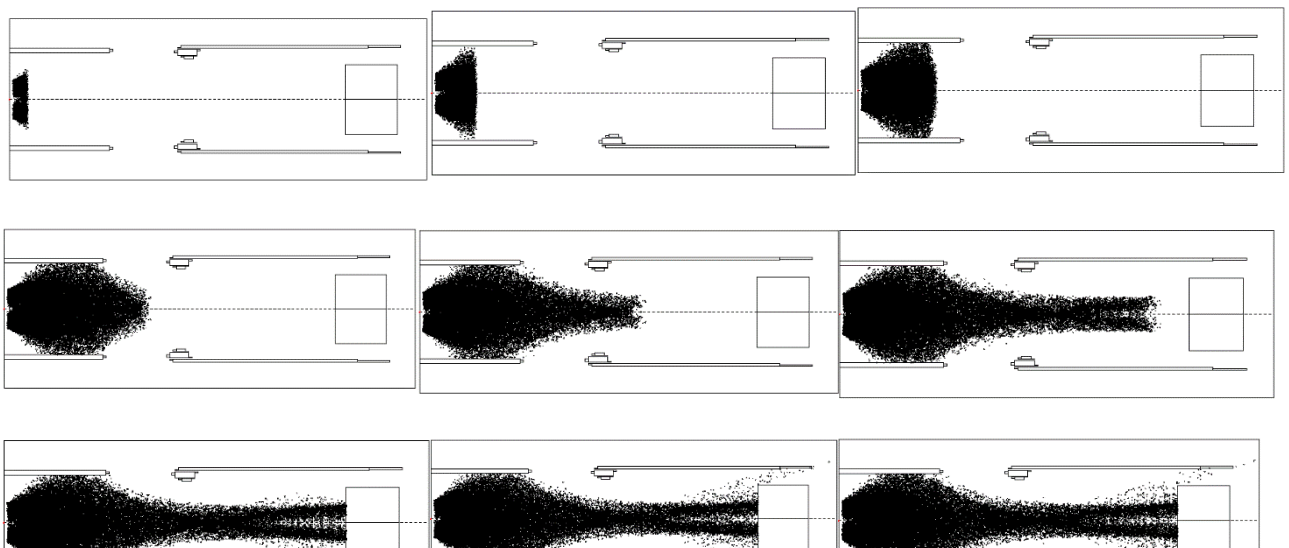
- Энергия дрейфа пучка в продольном направлении 1900 эВ
- Разброс по энергии в продольном направлении 100 эВ
- Разброс по энергии в поперечном направлении 120 эВ

Выполненные позднее эксперименты по измерению эмиттанса используемого в данной нейтронной трубке ионного источника показали, что при энергии ионов порядка 2кэВ, большая часть инжектируемых дейтронов (в соответствии с определением эмиттанса как площади проекции фазового объема пучка, в которой находится 90% полного числа изображающих точек) имеет угловой разброс dr/dz в диапазоне от 0.15 до 0.1 (рис. 26).



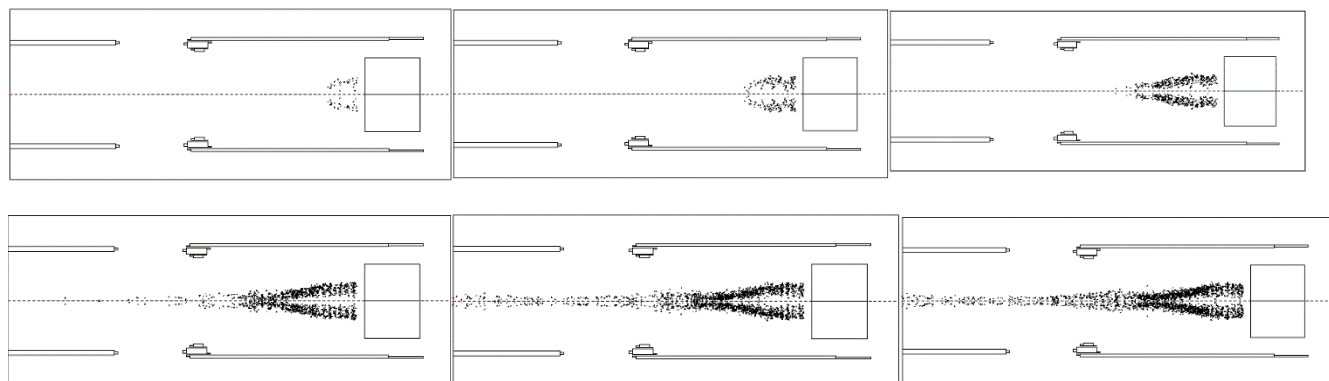
Такой разброс скоростей соответствует разбросу по энергии в поперечном направлении равному 125 эВ при энергии дрейфа в продольном направлении 2кэВ и, соответственно, 119 эВ при энергии дрейфа 1900 эВ, что хорошо согласуется с результатами, полученными в ходе моделирования.

Следующий рисунок иллюстрирует процесс заполнения пространства трубки дейтонным пучком с приведенными выше параметрами инъекции.

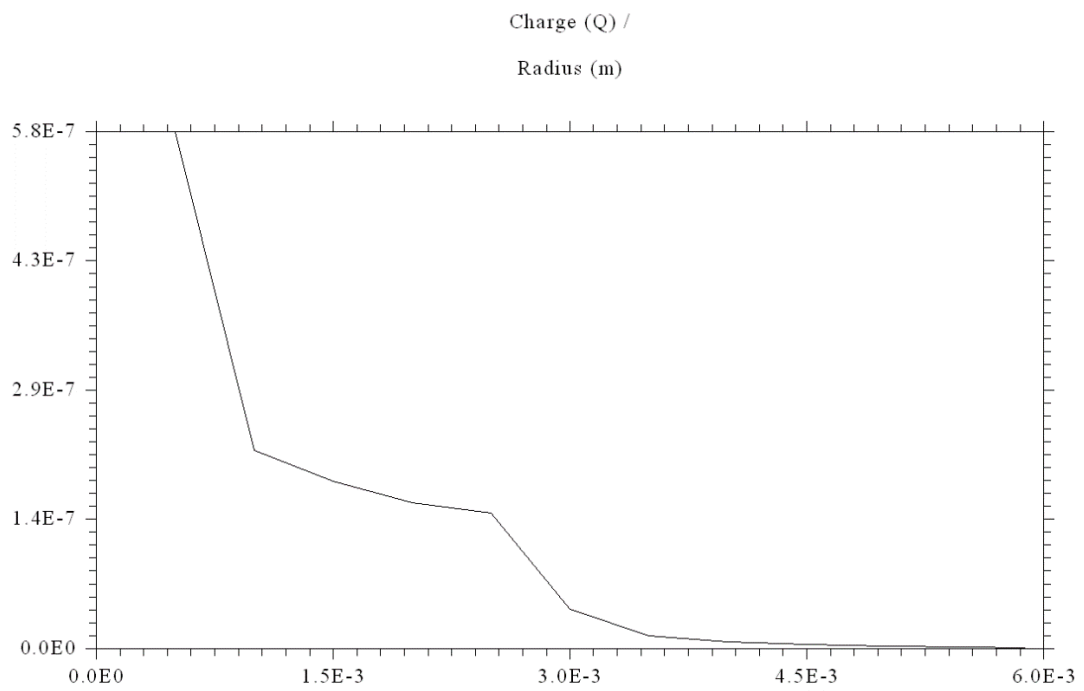
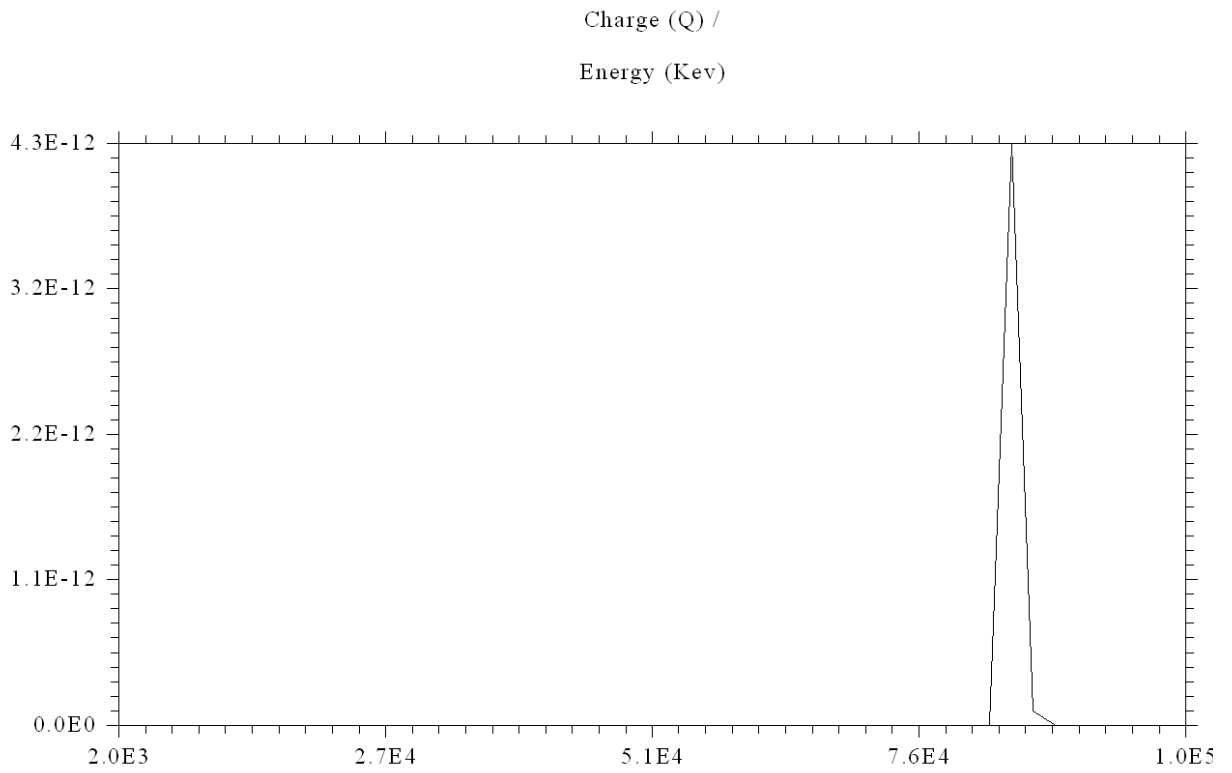


Каждой точке на рисунке соответствует одна крупная частица. После 60нс картинка практически не меняется.

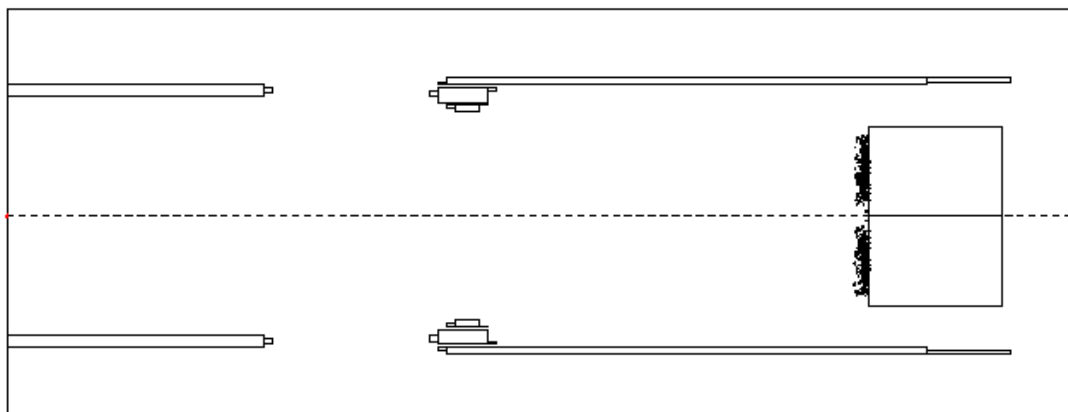
Когда дейтроны достигают мишени, то вследствие ион – электронной эмиссии на ней появляются вторичные электроны. Если разброс продольной энергии вторичных электронов превышает 2кВ (разность потенциалов между мишенью и ускоряющим электродом), то они быстро заполняют объём нейтронной трубки.



Видно, что не всем вторичным электронам удается преодолеть существующий потенциальный барьер ускоряющий электрод – мишень, часть из них разворачивается полем в направлении мишени. Поскольку скорость электронов значительно больше скорости ионов, то их количество в системе оказывается существенно меньше чем ионов, и, как следствие, влияние их объёмного заряда оказывается меньшим. Так энергетический спектр дейтонов и их плотность тока, приведенные ниже, практически не меняются при наличии и отсутствии вторичной эмиссии.

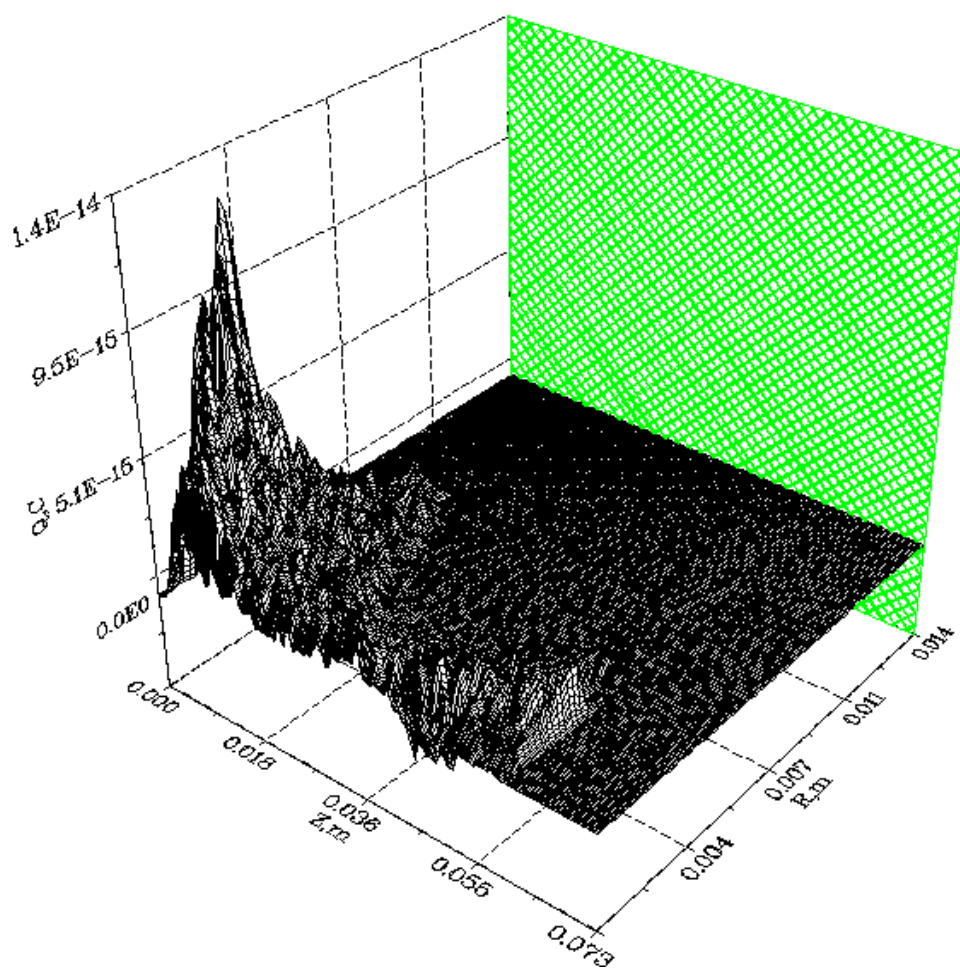


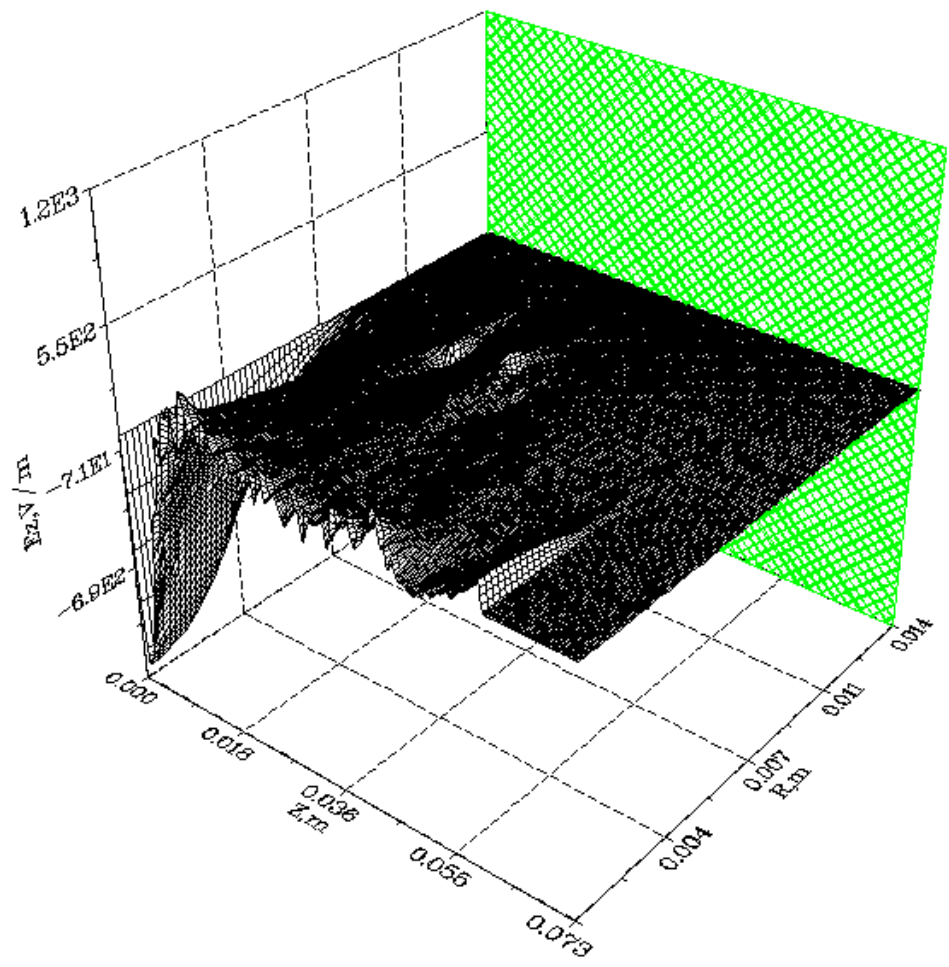
Если же начальная энергия вторичных электронов близка к нулю, то они запираются существующим потенциальным барьером.

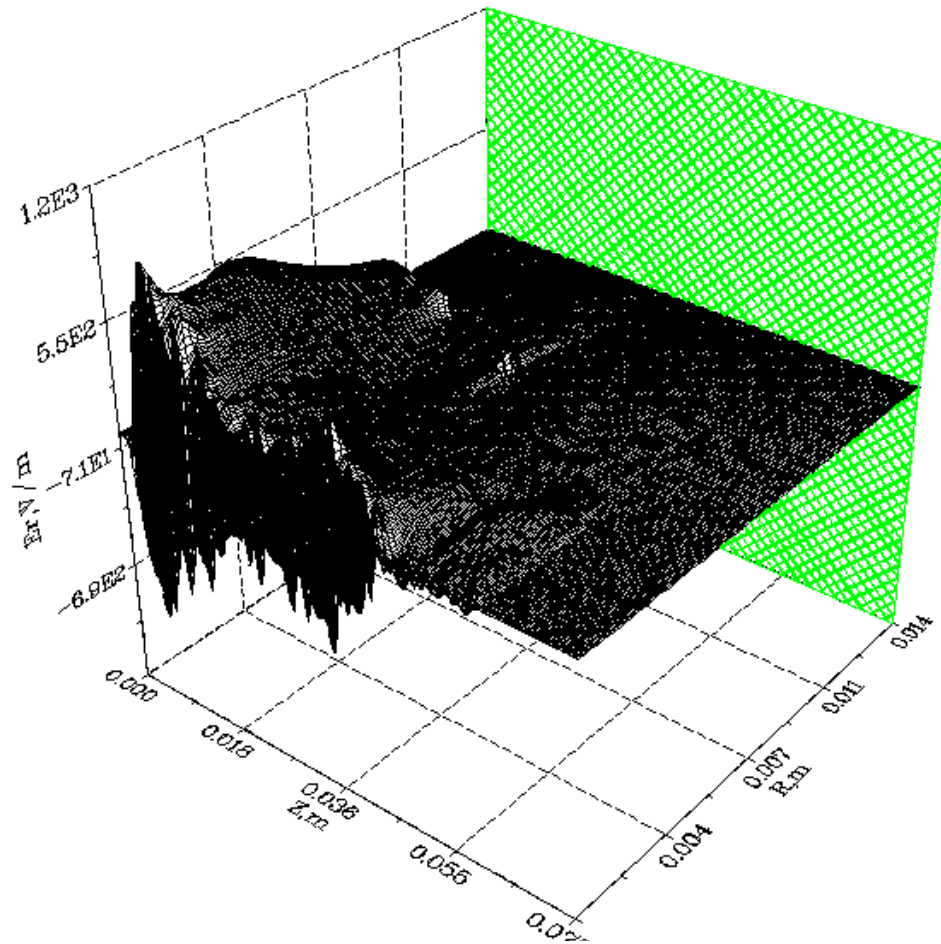


Приведенная картинка практически не меняется во времени.

Отсутствие влияния вторичных электронов на процессы в нейтронной трубке при данных параметрах задачи объясняется тем, что их собственные поля оказываются существенно меньше внешних. На следующем рисунке приведено распределение заряда и собственного продольного и поперечного поля в системе.





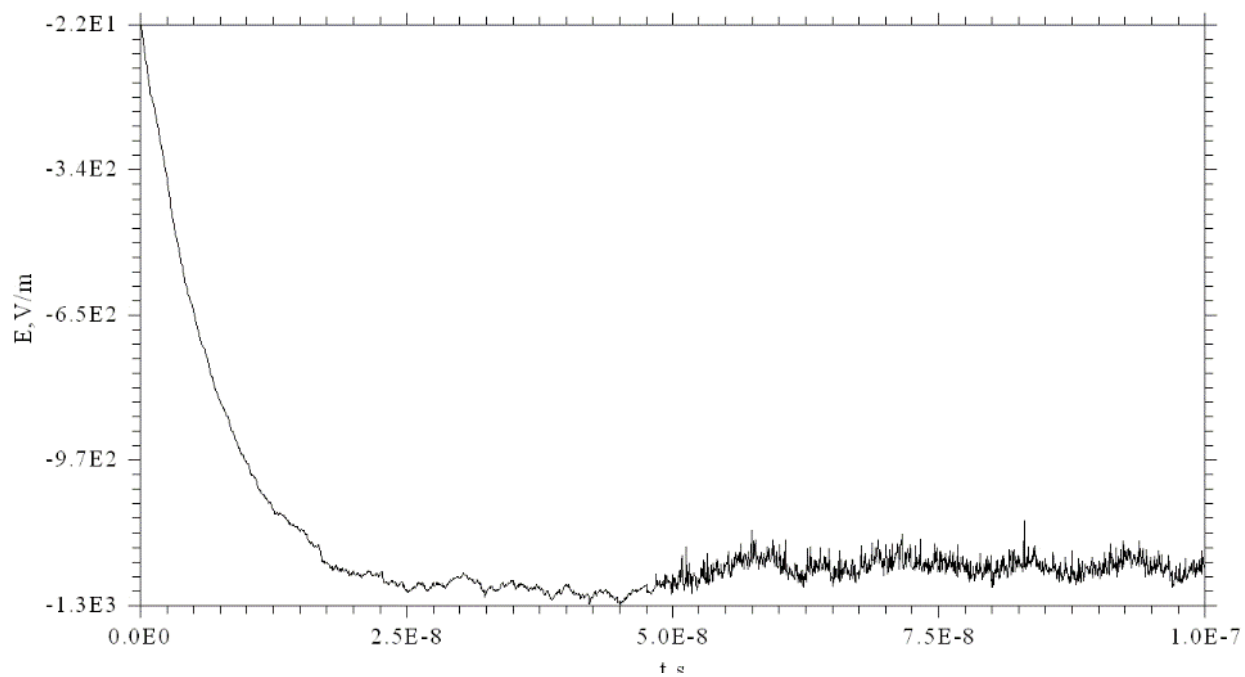


Видно, что по мере заполнения дейтронами системы, создаваемое ими электрическое поле растёт, однако его величина оказывается существенно меньше внешнего поля.

На следующих трёх рисунках представлена временная зависимость продольного поля пучка на оси системы в трёх точках: в начале трубки, в центре трубки и у мишени.

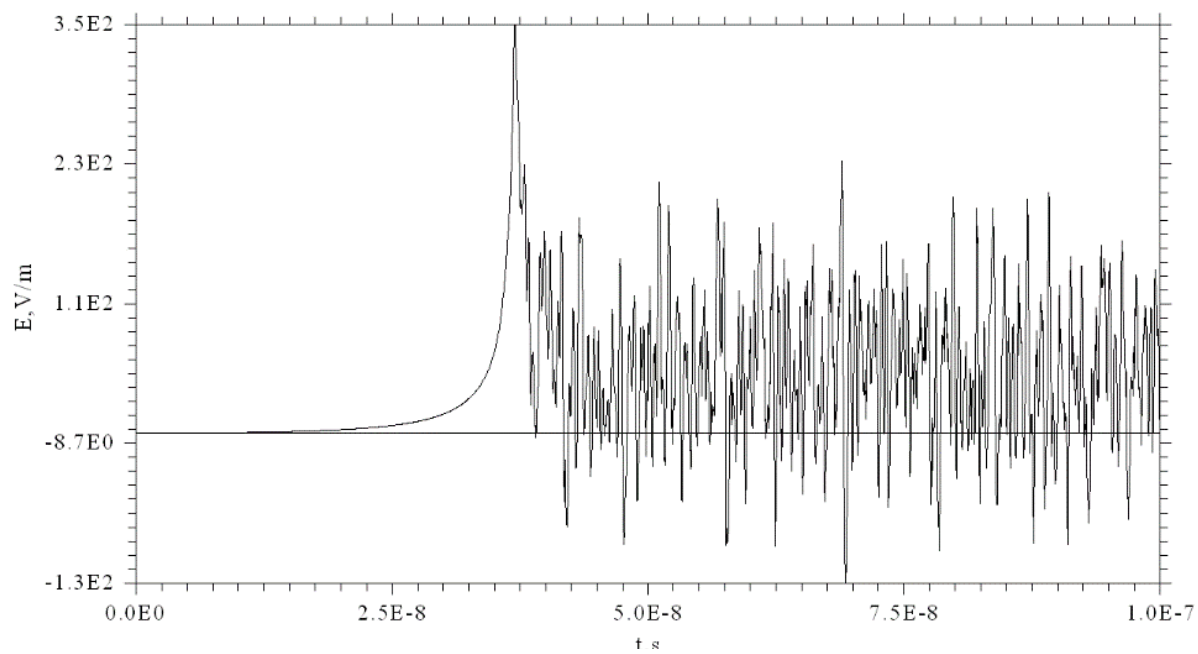
FIELD E1(Volt/m) /

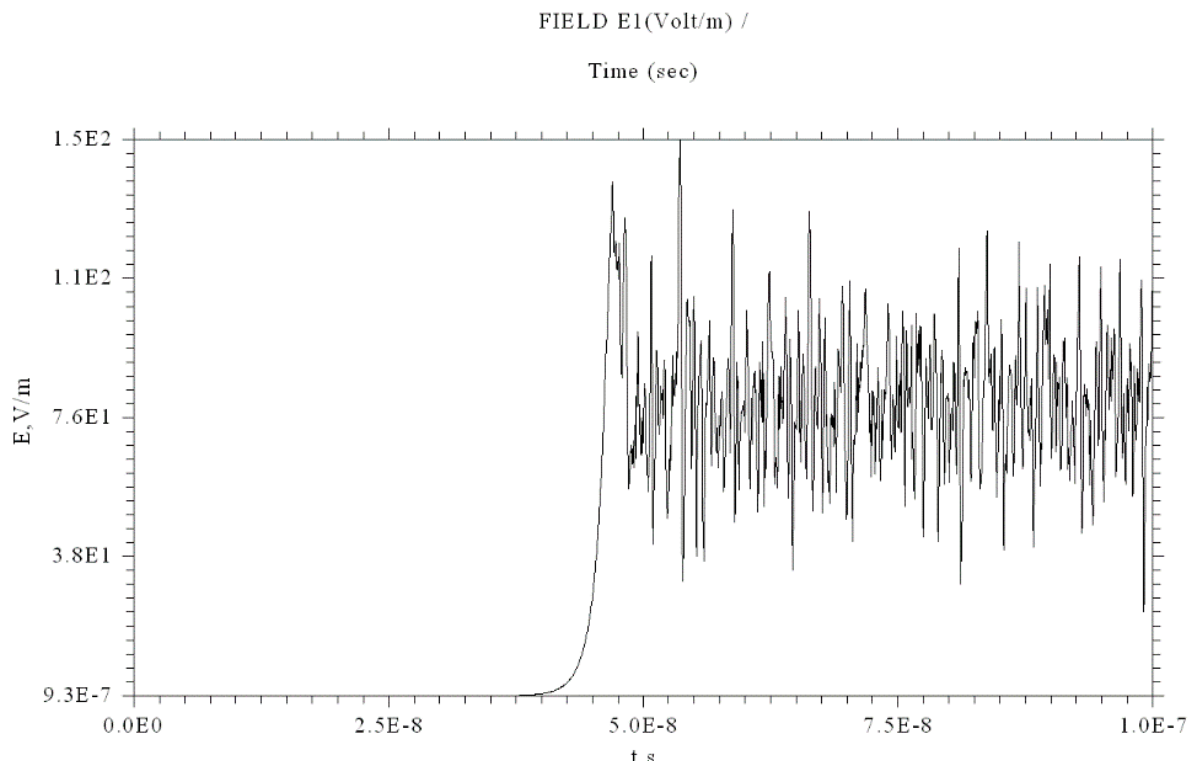
Time (sec)



FIELD E1(Volt/m) /

Time (sec)

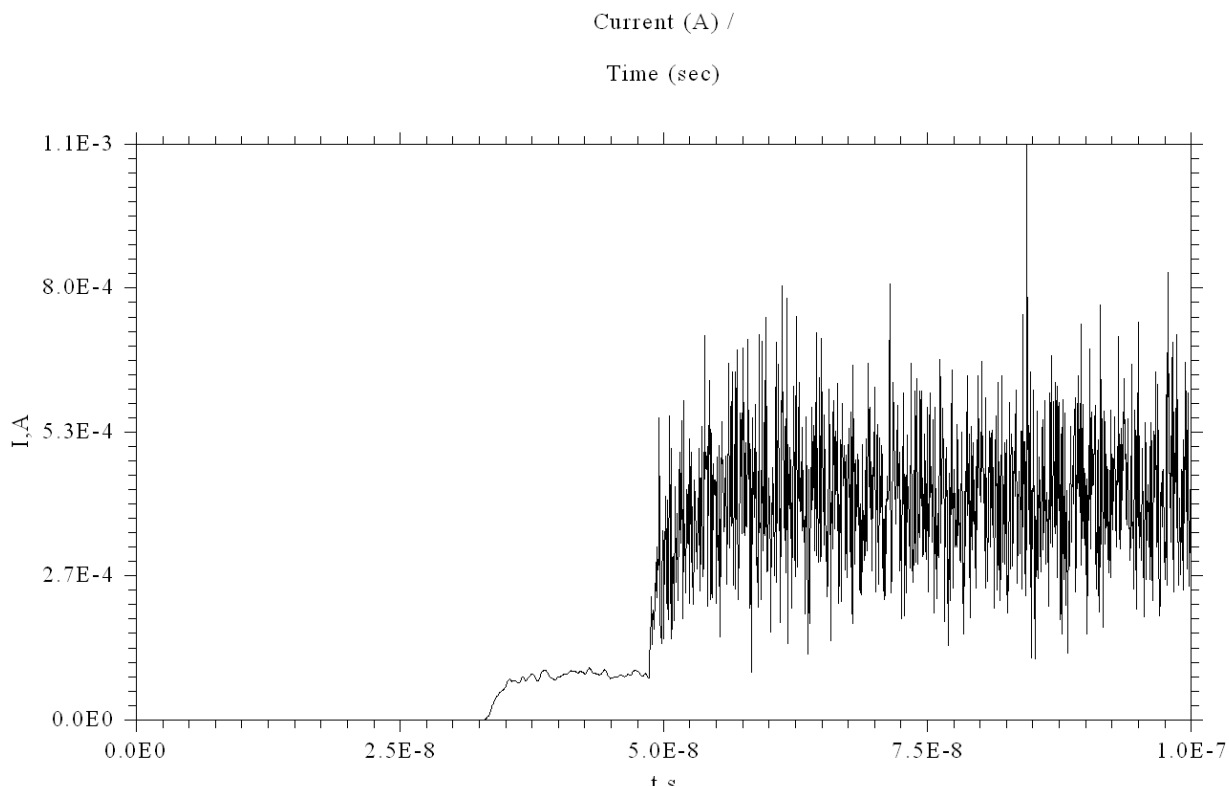




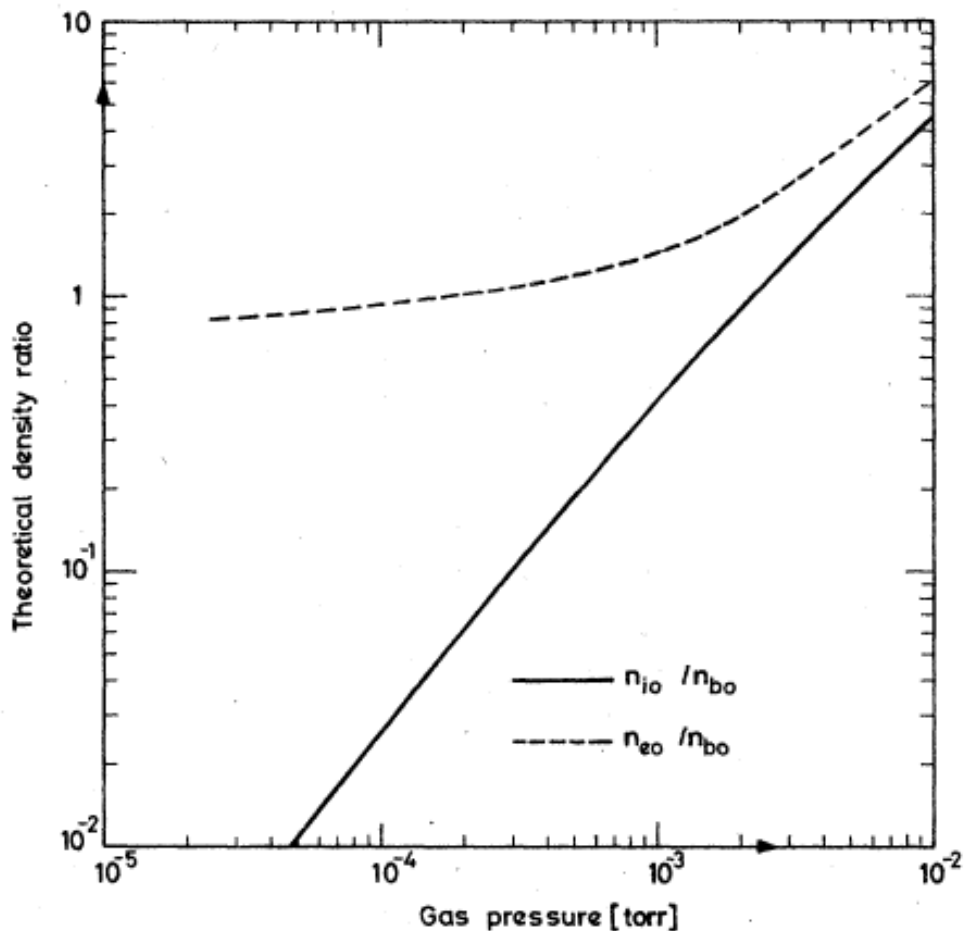
Появляющийся на рисунках «дребезг» вызван вторичными электронами.

Как видно из рисунков, максимальное собственное поле оказывается порядка 1кВ/м, что на два – три порядка меньше внешнего.

На следующем рисунке приведена временная диаграмма тока на оси системы в середине трубки. Видно, что ток равен нулю, пока ионы не достигли середины трубки. Затем он вырастает до уровня 80мкА, пока не появляется ток вторичных электронов. В рассматриваемом варианте предполагалось, что один дейтрон выбивает из мишени два электрона со случайным распределением по энергии от нуля до 2 кэВ. С учётом их частичного захвата потенциальной ямой ускоряющий электрод-мишень, их ток в центре системы оказывается порядка 500мкА.

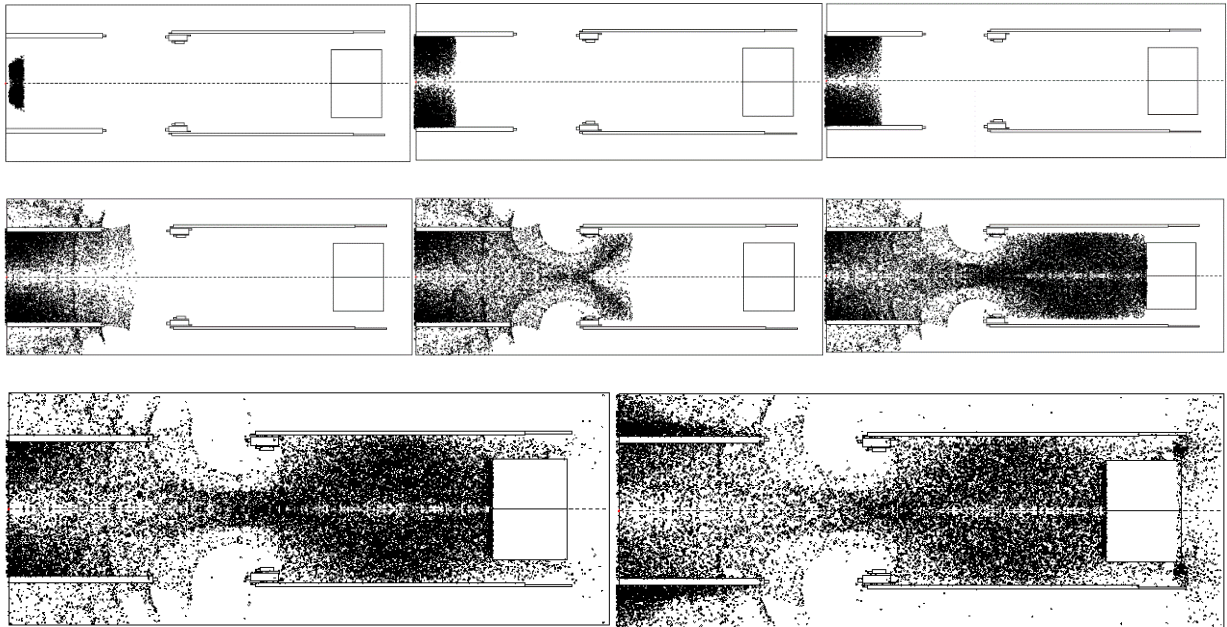


В выполненных выше расчётах учитывалась лишь вторичная эмиссия с мишени. Однако, как показывают проведенные ранее исследования [11-15], при прохождении дейтонного пучка через остаточный газ, образуется плазменно-пучковое образование, получившее название *ионно-пучковая плазма*. При достаточно высокой энергии ($E_{и} > 10$ кэВ), преобладают процессы ионизации остаточного газа и нагрева электронов непосредственно пучком ионов. При столкновении положительных, ионов пучка с атомами остаточного газа образуются электроны, и относительно медленные положительные ионы, концентрация которых существенно меньше электронной (смотри рис. 39 [11]). Последние выталкиваются из пучка электрическим полем, а электроны накапливаются в нём, несмотря на то, что этому препятствуют кулоновские столкновения их с первичными ионами. Так достигается частичная компенсация объёмного заряда в пучке положительных ионов.



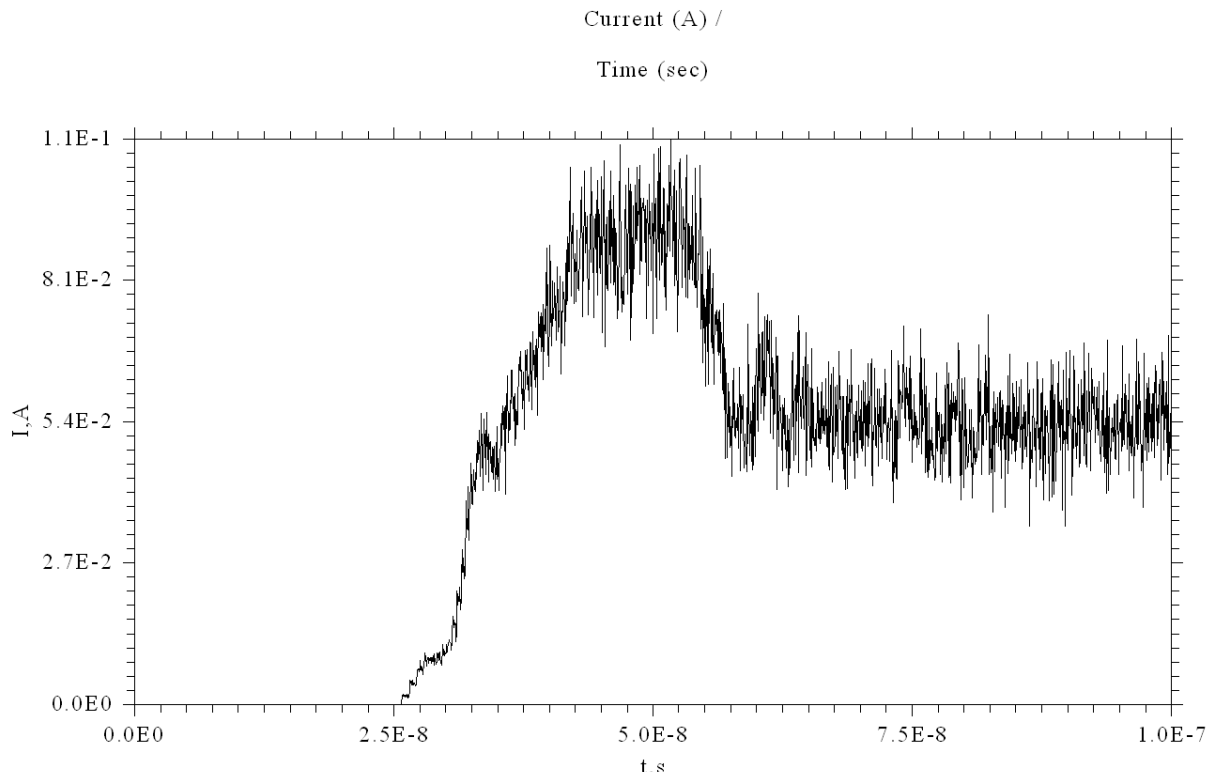
Следующая серия расчётов выполнена в предположении равенства плотностей электронного пучка, полученного в результате ионизации дейтронным пучком, и исходного дейтронного пучка.

На рис. 40 представлена картина заполнения нейтронной трубки вторичными электронами, образующимися при ионизации остаточного газа проходящим дейтронным пучком.

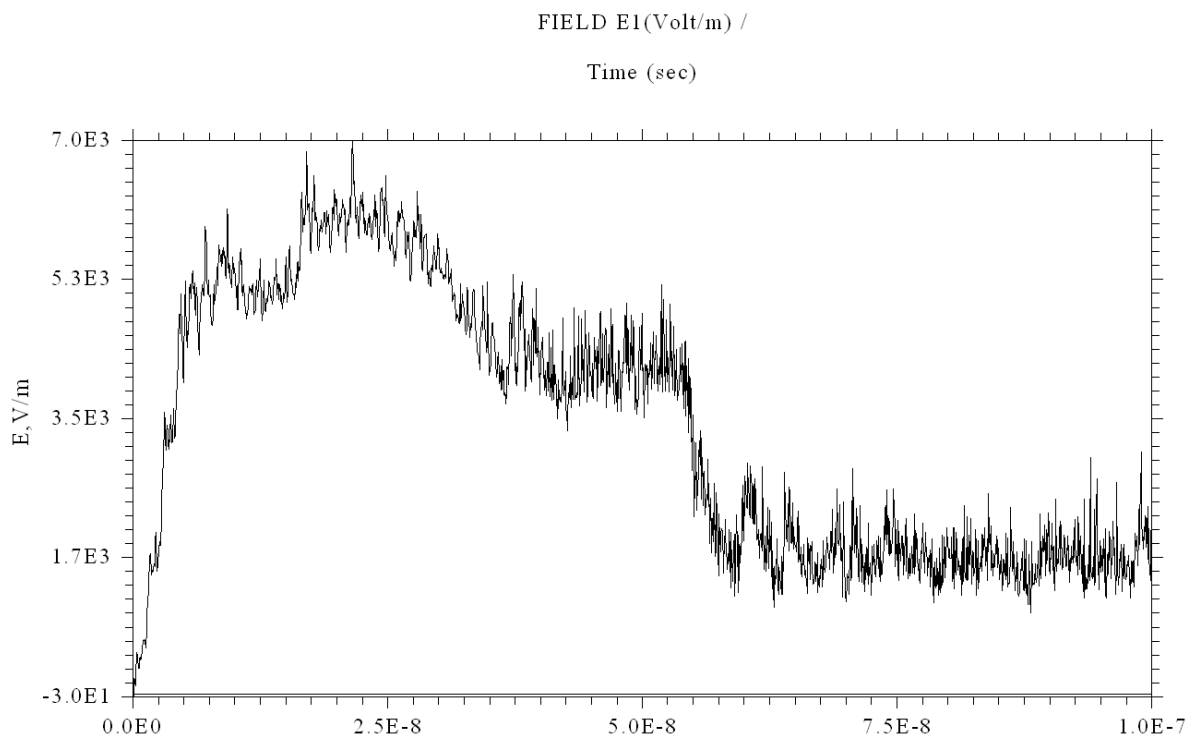


Каждой точке на рисунке соответствует одна крупная частица. После 60нс картинка практически не меняется.

На следующем рисунке приведена временная диаграмма тока на оси системы в середине трубки.



Видно, что по при наличии ионизации остаточного газа ток вырастает почти на три порядка. Однако и в этом случае собственное поле заряженных частиц в системе оказывается существенно меньше внешнего, что и иллюстрирует следующий рисунок, на котором приведено распределение собственного продольного поля зарядов на входе в систему.



Таким образом можно сделать вывод о том, что при данных параметрах системы и при принятых ранее допущениях о равенстве плотности дейтронного и полученного при ионизации вторичного электронного пучков, влиянием вторичных электронов можно пренебречь. Более детальный анализ возможен при наличии точных данных о сечениях реакции ионизации заполняющих систему газов при заданном давлении.

Заключение

В ходе выполнения работы получены следующие результаты:

- Проведена глубокая модернизация кода программы СУМА, позволившая значительно улучшить структуру программы, что в свою очередь существенно упростило её дальнейшее использование и модернизацию.
- Существенно улучшен интерфейс программы, что позволило, с одной стороны, значительно облегчить задание начальных данных для конкретных вариантов расчёта, а с другой стороны дать более детальную картину моделируемого процесса за счёт модернизации выходной графической информации.
- Использование элементов параллельного программирования дало возможность заметно ускорить выполнение кода, что позволяет за приемлемые времена счёта увеличить число моделируемых крупных частиц и размер покрываемой область исследования сетки, повышая тем самым точность модели и расширяя круг решаемых задач.
- Проведенная модернизация кода позволила наметить пути его дальнейшей оптимизации для более глубокого использования параллельных вычислений. Потенциал для дальнейшей оптимизации остается очень большим, и в дальнейшем можно получить еще более значительный прирост производительности.
- Проведенные в рамках договора с ВНИИА им. Духова расчёты, продемонстрировали возможности модернизированной программы и позволили провести детальный анализ физических процессов в нейтронной трубке.

Литература

1-11. Вставить ссылки и поправить номера в тексте

1. Рашиков В.И. Расчет электромагнитных полей в структурах сложной геометрии. В кн. : Вопросы атомной науки и техники Серия: Ядерно-физические исследования вып. 10(18), М. , 1990, с. 50- 53
2. А.Н.Диденко, В.И.Рашиков. Механизм генерации мощных СВЧ колебаний в системах с виртуальным катодом. ДАН СССР, т. 313, N3,1990, с. 597
3. A.N.Didenko, V.I.Rashchikov. High- power microwave generation in virtual cathode systems. IEEE 1991 Particle Acceleration conference San- Francisco , US 1991, v. 5, p. 3111
4. Рашиков В.И. Поле пространственного заряда пучка в структуре с трубками дрейфа. В кн. : Линейные ускорители. Теория и эксперимент. М. , Энергоатомиздат, 1991, с. 48-55
5. А.Н.Диденко, В.И.Рашиков. Генерация мощных СВЧ колебаний в системах с виртуальным катодом. Физика плазмы, т. 18, N9, 1992, с. 1182-1190
6. А.Н.Диденко, В.И.Рашиков, В.Е.Фортов. О возможности генерации мощного излучения терагерцового диапазона частот при воздействии мощных лазерных импульсов на мишень. Письма в ЖТФ, 2011, т.37, вып.6 , с.27-32.
7. А.Н.Диденко, В.И.Рашиков, В.Е.Фортов. Механизм генерации мощного излучения терагерцового диапазона частот при воздействии мощных лазерных импульсов на мишень. ЖТФ, 2011, т.81, вып.10, с.155-158.
8. А.Н.Диденко, В.И.Рашиков, В.Е.Рыжков, А.Е.Шиканов. Генерация наносекундных нейтронных импульсов в диодных ускорительных трубках с вакуумно-дуговыми и лазерными источниками дейтронов . ВАНТ, 2012. №4(80) стр. 129-132
9. А.Н.Диденко, В.И.Рашиков, В.Е.Рыжков, А.С.Цыбин, А.Е.Шиканов. Компьютерное моделирование генерации наносекундных нейтронных импульсов в вакуумных ускорительных трубках. Атомная энергия, 2012, т. 112, вып. 3, с. 182 -184
10. А.Н.Диденко, В.И.Рашиков, В.Е.Фортов. Генерация терагерцового излучения при воздействии мощных лазерных импульсов на мишень. ВАНТ, 2012. №3(79)с.179-182
11. Holmes A.J.T. Theoretical and experimental study of space charge in intense ion beam// Phys.Rev. A. – 1979. – Vol. 19, № 1. – P. 389-407.

12. Габович М.Д. Ионно-пучковая плазма и распространение компенсированных ионных пучков//УФН. – 1977. – Т. 121, – № 2. –С. 259-284.
13. Габович М.Д. Физика и техника плазменныхисточников ионов. – М.: Атомиздат, 1972. –304 с.
14. Габович М.Д. Компенсированные ионные пучки// УФЖ. – 1979. – Т. 24, № 2. –С. 257-273.
15. Габович М.Д., Кацубо Л.П., Солошенко И.А.,Шуба Я.М. Динамическая декомпенсация пучка положительных ионов в отсутствие магнитного поля// Физика плазмы. – 1980. –Т. 6, Вып. 4. – С. 925-932.
16. Max Planck Institute for Solar System Research, DISLIN Home Page, <http://www.dislin.de>.
17. В. Фролов. Введение в технологию CUDA, <http://cgm.computergraphics.ru/issues/issue16/cuda>.
18. NVIDIA, CUFFT, <http://docs.nvidia.com/cuda/cufft/index.html>.
19. The Portland Group, PGI Visual Fortran® User's Guide, Release 2013.
20. The Portland Group, PGPROF® Profiler Guide, Release 2013.