

# ОПТИМИЗАЦИЯ И ТЕСТИРОВАНИЕ ЧИСЛЕННОГО КОДА ДЛЯ РАСЧЁТА ХАРАКТЕРИСТИК ДИФРАКЦИОННОГО ИЗЛУЧЕНИЯ ОТ ПЛОСКИХ МИШЕНЕЙ

А.А. Дышеков

Научный руководитель: к.ф.-м.н., млад. науч. сотрудник Шкитов Д.А.

Томский политехнический университет

aad52@tpu.ru

## Введение

В настоящее время ускорители заряженных частиц широко используются как в прикладных, так и в фундаментальных исследованиях. Известно, что без качественного мониторинга характеристик пучков заряженных частиц, эксплуатация ускорителей практически не возможна. Для этого разрабатываются различные средства диагностики таких параметров пучков, как поперечные и продольные размеры, расходимость, эмиттанс и др. Одним из способов диагностики размеров сгустков, из которых состоит пучок, является применение дифракционного излучения (ДИ).

Дифракционное излучение [1] это излучение, которое возникает при пролёте заряженной частицы (например, электрон) вблизи некоторой неоднородности (например, мишень представляющей собой плоскую прямоугольную металлическую пластину). Источников излучения являются поверхностные токи, возникающие в результате динамической поляризации атомов мишени посредством кулоновского поля зарядов.

Для разработки способов диагностики на основе ДИ необходимо проводить расчёты различных характеристик этого излучения. Однако сейчас у экспериментаторов существует возможность использовать либо аналитические формулы (с многочисленными допущениями), либо коммерческие пакеты моделирования (требующие времени на освоение и опыта применения). Поэтому есть необходимость создать простое и понятное в использовании программное обеспечение для расчёта характеристик ДИ [2].

Целью настоящей работы является оптимизация, отладка и тестирование ранее написанного численного кода [2], который предоставил бы пользователям возможность рассчитать характеристики ДИ для реальных параметров эксперимента. Расчёты ведутся на основе обобщённого метода поверхностных токов [3].

## Описание численного кода и результаты

Первая версия компьютерного кода, написанного на языке *Wolfram Language* [4], включает в себя следующие файлы:

- «Geometry.wl» – содержит функции задающие геометрию мишени с необходимыми параметрами (размеры, угол наклона и т.д.).
- «Functions.wl» – содержит функции, непосредственно вычисляющие напряженность поля или

интенсивность излучения, возникающей от мишени, созданной с помощью функций из пакета «Geometry.wl», в заданной точке на детекторе.

Для работы с кодом необходимо подгрузить функции из описанных выше файлов, и в дальнейшем работать с ними как со встроенными функциями системы *Mathematica*. Подробную информацию о работе с данными функциями можно узнать, запросив справки о них.

Геометрия мишени задаётся формой, размерами и ее положением в пространстве. Для удобства подбора необходимых параметров геометрии к коду прилагается *CDF* файл, который содержит визуализированную мишень (см. рис. 1), задающуюся регулируемыми параметрами.

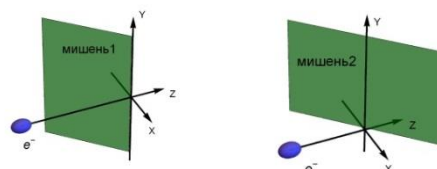


Рис. 1. Примеры геометрий мишеней ДИ от одной пластины (№1 – слева и №2 – справа), электрон движется вдоль оси Z, пластины наклонены под углом 45° к оси Z

На рис. 2 приведены первые результаты расчётов спектрально-углового (a1 и a2) и спектрального (b1 и b2) распределений в одночастичном приближении (т.е. от одного электрона). Параметры расчётов были следующие: размеры пластины 30×20 мм<sup>2</sup>, длина волны излучения 1 мм, энергия электрона 10 МэВ, расстояние по оси X от нуля системы отсчёта до детектора 0.5 м, прицельный параметр 0.5 мм.

Для расчёта данных «a1» и «a2» было затрачено 9,9 ч и 4,7 мин соответственно (81 точка). Для расчёта данных «b1» и «b2» было затрачено 8,6 ч и 6 мин соответственно (79 точек). Расчёты проводились на компьютере с процессором Intel Core Processor (Haswell) с частотой 2.59 ГГц на одном ядре.

На данный момент, при вычислениях ДИ от мишени №1 *Mathematica* выводит сообщения, которые указывают, что либо одна компонента поля равна нулю, либо подынтегральная функция сильно осциллирует, либо недостаточно точности внутренних расчётов. Исходя из симметрии задачи, действительно, имеем, что Y компонента поля равна нулю вдоль оси Z на детекторе. Поэтому расчёты для мишени №1 длились значительно дольше,

чем для №2 (внутренние алгоритмы *Mathematica* автоматически пытались повысить точность).

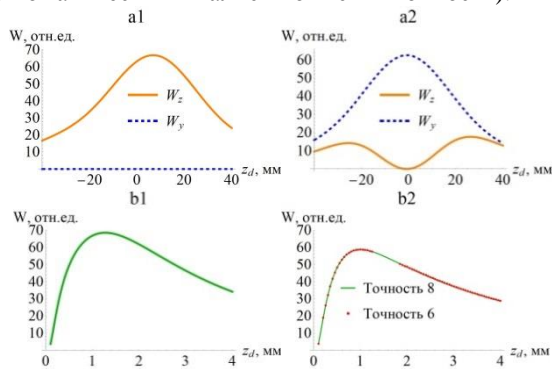


Рис. 2. Результаты расчёта ДИ, а – спектрально-угловые распределения для Y и Z компоненты поля излучения, б – спектры излучения (для  $\{z_d, y_d\} = \{7.2, 0\}$  и  $\{-1, 0\}$  соответственно), №1 и №2 – модификации мишеней, как на рис. 1

### Оптимизация и отладка кода

Сначала были учтены рекомендации по оптимизации кода, изложенные в обзоре [5].

В код была добавлена функция, отвечающая за очистку памяти от вычисленных данных. Это позволило оптимизировать использование памяти вычислительного устройства, что поможет предотвратить уменьшение производительности в дальнейшем.

Изю всех вычислений также было убрана абсолютная точность, ограничившись машинной точностью. Таким образом, были исключены возможные более громоздкие аналитические расчёты.

Основное вычисление характеристик ДИ в коде сводится к многократному интегрированию по поверхности мишени. Поэтому для оптимизации кода необходимо в первую очередь оптимизировать процесс интегрирования. Поэтому для встроенной функции *NIntegrate* [4] была выбрана опция, исключающая аналитические методы нахождения интеграла.

### Тестирование кода

Для оценки влияния входных параметров на время расчёта было проведено тестирование кода (результаты в таблице 1 ниже).

Таблица 1. Сравнение времени расчёта одной точки

Кол-во ядер	$\lambda=1$ мм	$\lambda=8$ мм	$\lambda=1$ мм, $S*2$	$\lambda=8$ мм, $S*2$
1	5,0 с.	3,4 с.	5,5 с.	3,9 с.

Из таблицы видно, что для больших значений длин волн расчёт происходит быстрее, а также что, при увеличении площади в два раза, время расчёта увеличилось не значительно.

### Рекомендации по дальнейшей разработке и отладке кода

Далее приведём несколько рекомендаций по дальнейшему совершенствованию кода:

1. Проверить возможность подключения библиотек языка “C” (или “python”) для ускорения расчёта многократных интегралов;
2. Изменить код для возможности исключения времязатратного расчёта компонент поля, которые равны нулю из соображений симметрии;
3. Модифицировать математическую модель для возможности расчёта ДИ от кривых поверхностей [6];
4. Получить отзывы о работе с программой от сторонних пользователей для дальнейшего её совершенствования.

Получение текущей версии разработанного ПО возможно по временной ссылке [7], либо по запросу по почте.

### Заключение

В результате проведения тестирования можно сделать вывод о том, что в целом код оптимизирован и показывает результаты соответствующие физическим ожиданиям. Однако выявлена необходимость оптимизации расчётов (или модификации функций) при особых случаях геометрии мишеней.

### Список использованных источников

1. Потылицын А.П. и др. Дифракционное излучение релятивистских частиц: учебное пособие. – Томск: Изд-во ТПУ, 2008. – 347 с.
2. Shkitov D.A. Code for simulation of diffraction radiation from flat finite surfaces // Proceedings of RuPAC2018, Protvino, 2018. – С. 510-513.
3. Karlovets D.V., Potylitsyn A.P. Generalized surface current method in the macroscopic theory of diffraction radiation // Physics Letters A – 2009. – V. 373. – P. 1988-1998.
4. Wolfram Language & System Documentation Center [Электронный ресурс] – URL: <https://reference.wolfram.com/language/> (дата обращения 19.11.2018).
5. Осипов Р. Оптимизация кода в Wolfram Mathematica [Электронный ресурс] – URL: <https://www.youtube.com/watch?v=wysmw-nzjfPM> (дата обращения 16.11.2018).
6. Potylitsyn A.P., Sukhikh L.G. Transition radiation from a cylindrical target and transverse beam size diagnostics // Proceedings of DIPAC2011, Hamburg, 2011. – P. 410-412.
7. Текущая версия кода [Электронный ресурс] – URL: <https://filecloud.tpu.ru/index.php/s/zDXLI-MFOGRfFyq4>