

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Нижегородский государственный университет им. Н. И. Лобачевского  
Национальный исследовательский университет

**А. В. Маслов**

# **МОДЕЛИРОВАНИЕ ВОЛНОВЫХ ПРОЦЕССОВ В ЭЛЕКТРОДИНАМИКЕ**

Учебное пособие

Рекомендовано научно-методическим советом исследовательской школы «Лазерная физика» для аспирантов ННГУ, обучающихся по направлению подготовки 03.06.01 «Физика и астрономия», и для магистрантов ННГУ, обучающихся по направлениям подготовки 03.04.03 «Радиофизика», 02.04.02 «Фундаментальная информатика и информационные технологии»

Нижний Новгород  
2019

УДК 537.87  
ББК 22.336  
М-31

М-31 **Маслов А.В.** МОДЕЛИРОВАНИЕ ВОЛНОВЫХ ПРОЦЕССОВ В ЭЛЕКТРОДИНАМИКЕ: Учебное пособие. – Нижний Новгород: Нижегородский госуниверситет, 2019. – 75 с.

Рецензенты: д.т.н., профессор **С. В. Оболенский**  
к.ф.-м.н., доцент **И. Ю. Демин**

В пособии изложены основы и приёмы численного подхода к решению электродинамических задач методом конечных разностей во временной области (КРВО) (finite-difference time-domain method, FDTD). Для практического усвоения материала рассмотрены решения конкретных задач возбуждения и распространения волн в различных средах, а также их рассеяния на объектах. Используются примеры моделирования сред с временной дисперсией, поглощением, анизотропией. Представлены коды программ численного моделирования на языке С. Описаны организация файлов и директорий для оптимизации процесса моделирования, а также процесс компиляции программ в системах Unix/Linux. Пособие способствует приобретению практических навыков решения электродинамических задач методом КРВО, который широко используется в фундаментальных и прикладных исследованиях. Метод КРВО лежит в основе многих коммерческих пакетов электродинамического моделирования, используемых в инженерно-конструкторских работах.

Пособие предназначено для магистрантов и аспирантов радиофизического факультета ННГУ, специализирующихся в области электродинамики, нелинейной оптики, лазерной физики.

Ответственный за выпуск:  
Председатель научно-методического совета  
исследовательской школы «Лазерная физика»  
д.ф.-м.н., профессор Бакунов М.И.

УДК 537.87  
ББК 22.336

# Содержание

<b>1</b>	<b>Введение</b>	<b>5</b>
	<i>Цели и задачи пособия. Пререквизиты дисциплины.</i>	
<b>2</b>	<b>Рассматриваемые электродинамические задачи</b>	<b>6</b>
<b>3</b>	<b>Основные принципы решения</b>	<b>8</b>
	<i>Уравнения Максвелла. Дискретизация уравнений. Структура пространственной сетки. Алгоритм изменения поля во времени. Начальные условия. Граничные условия.</i>	
<b>4</b>	<b>Пример 1: Распространение импульса в свободном пространстве</b>	<b>13</b>
4.1	Разработка программы моделирования . . . . .	13
	<i>Разбивка программы на функции. Организация директорий и файлов. Компиляция и выполнение программы.</i>	
4.2	Результаты моделирования и их обсуждение . . . . .	19
	<i>Распространение импульса. Отражение от границы области моделирования. Искажение импульса. Зависимость искажения от длительности импульса и временного шага.</i>	
<b>5</b>	<b>Выбор пространственного и временного шагов</b>	<b>21</b>
	<i>Устойчивость алгоритма. Численная дисперсия.</i>	
<b>6</b>	<b>Моделирование сред без дисперсии</b>	<b>23</b>
	<i>Материальные соотношения. Структура алгоритма. Критерий устойчивости. Выбор пространственного и временного шагов.</i>	
<b>7</b>	<b>Получение частотных характеристик</b>	<b>25</b>
	<i>Преобразование Фурье. Выбор частот. Дискретизация по времени. Выбор начального и конечного временных пределов.</i>	
<b>8</b>	<b>Пример 2: Падение импульса на слой диэлектрика</b>	<b>26</b>
8.1	Разработка программы моделирования . . . . .	26
	<i>Функция управления моделированием. Функции задания диэлектрической проницаемости и расчёта поля <math>E</math>. Функция взятия преобразования Фурье. Компиляция.</i>	
8.2	Результаты моделирования и их обсуждение . . . . .	32
	<i>Выбор точек для преобразования Фурье. Получение спектра падающего импульса. Сравнение результатов численного расчёта с аналитическими.</i>	

<b>9</b>	<b>Моделирование сред с поглощением</b>	<b>33</b>
	<i>Физическая модель поглощения. Алгоритм расчёта поля <math>E</math>. Введение добавочных величин.</i>	
<b>10</b>	<b>Безотражательные (поглощающие) граничные условия</b>	<b>35</b>
	<i>Причина отражения. Простейшее безотражательное граничное условие в 1D. Идеально согласованный слой (PML).</i>	
<b>11</b>	<b>Моделирование сред с дисперсией</b>	<b>37</b>
	<i>Использование диэлектрической проницаемости в частотной области. Использование дифференциальных уравнений.</i>	
<b>12</b>	<b>Разделение на области полного и рассеянного полей</b>	<b>39</b>
	<i>Цель разделения. Граничные условия. Формулы пересчёта полей на границе.</i>	
<b>13</b>	<b>Пример 3: Падение импульса на слой плазмы</b>	<b>41</b>
13.1	Разработка программы моделирования . . . . .	41
	<i>Функция управления моделированием. Функции поглощающего слоя. Функции учёта свободных носителей.</i>	
13.2	Результаты моделирования и их обсуждение . . . . .	45
	<i>Распределение полей в разные моменты времени. Поведение на границе областей полного и рассеянного полей. Сравнение результатов численного расчёта с аналитическими.</i>	
<b>14</b>	<b>Двумерное моделирование</b>	<b>46</b>
	<i>Уравнения Максвелла для TE волн. Выбор 2D сетки. Дискретизация уравнений и алгоритм расчёта изменения полей.</i>	
<b>15</b>	<b>Идеально согласованный слой в 2D</b>	<b>49</b>
	<i>Отражение плоской волны от границы при наклонном падении. Анизотропные среды и отражение от них. Условие согласования между изотропной и анизотропной средами. Поглощение в анизотропной среде. Алгоритм расчёта полей <math>E</math> из <math>D</math> и <math>H</math> из <math>B</math>.</i>	
<b>16</b>	<b>Пример 4: Возбуждение волн точечным источником и их поглощение идеально согласованным слоем</b>	<b>53</b>
	<i>Использование точечных источников. Цилиндрические волны. Сравнение поведения волн при наличии идеально согласованного слоя и без него.</i>	
<b>17</b>	<b>Разделение на области полного и рассеянного полей в 2D</b>	<b>55</b>

*Выбор границы областей. Формулы расчёта полей.*

<b>18 Пример 5: Рассеяние плоской волны на цилиндре</b>	<b>57</b>
<i>Создание импульса и его распространение при наличии областей полного и рассеянного полей. Использование аналитической формулы и вспомогательной 1D сетки при моделировании падающего импульса. Основные свойства рассеяния.</i>	
<b>19 Расчёт сечения рассеяния</b>	<b>60</b>
<i>Определение сечения рассеяния. Расчёт сечения рассеяния с использованием преобразования Фурье от полей.</i>	
<b>20 Расчёт диаграммы рассеяния и пересчёт полей из ближней зоны в дальнюю</b>	<b>61</b>
<i>Диаграмма рассеяния. Метод эффективных токов.</i>	
<b>21 Заключение</b>	<b>65</b>
<b>Список литературы</b>	<b>66</b>
<b>22 Приложение:</b>	
<b>Излучение точечных источников в 2D</b>	<b>68</b>
<i>Уравнение Гельмгольца и его решение. Асимптотические свойства решения. Функции Грина для электрических и магнитных токов.</i>	
<b>Задачи</b>	<b>72</b>
<b>Об авторе</b>	<b>74</b>

# 1 Введение

Пособие предназначено для приобретения практических навыков решения электродинамических задач методом конечных разностей во временной области (КРВО). Обозначение КРВО является переводом на русский термина FDTD method (finite-difference time-domain method), используемого в англоязычной литературе. Метод КРВО является одним из наиболее распространённых методов моделирования. В пособии изложены основы метода КРВО на примере решения одномерных (1D) и двумерных (2D) задач. Приобретённые навыки помогут при дальнейшем изучении метода и его применении для решения и трехмерных (3D) задач.

В пособии делается упор на следующих вопросах:

1. Теоретические основы метода КРВО и особенности вычислительных сеток.
2. Разработка программ с модульной структурой.
3. Организация файлов, директорий и процесса создания выполняемых файлов для оптимизации моделирования.
4. Проработка и усвоение теоретического материала с использованием полного текста программ моделирования.
5. Формирование базовых теоретических знаний и практических навыков для дальнейшего использования метода КРВО при решении 3D задач.

В пособие включены основные приемы метода КРВО, позволяющие использовать среды с дисперсией и поглощением, моделировать открытые системы с помощью идеально согласованных слоёв на границе вычислительной области, получать частотные характеристики (коэффициенты отражения и прохождения).

Для овладения материалом в пособии предполагается знание электромагнитной теории в рамках курса общей физики, основ математического анализа и наличие навыков программирования. В пособии представлены программы на языке C и даны практические советы по их компиляции и использованию для работы на системах Unix/Linux. Программы были написаны в текстовом редакторе XEmacs, который облегчает редактирование исходных кодов на множестве языков программирования, включая C. Для построения графиков была использована программа Gnuplot. Алгоритмы, представленные в пособии, могут быть реализованы читателем в конкретные программы и на других языках программирования, а программы могут выполняться и под

другими операционными системами. Тем не менее следует отметить, что выбор языка C и систем Unix/Linux обусловлен их широким использованием как в коммерческих компаниях, так и в различных научно-исследовательских лабораториях и университетах во всем мире.

## 2 Рассматриваемые электродинамические задачи

Поведение электромагнитных волн описывается уравнениями Максвелла. К сожалению, аналитическое решение этих уравнений возможно только в отдельных идеализированных случаях. Примерами таких случаев являются отражение от плоской границы или рассеяние света сферическими частицами. Поэтому для решения практических задач приходится использовать численные методы. Метод КРВО зарекомендовал себя как достаточно надежный и универсальный метод, который к тому же довольно прост на практике. Его широкое распространение обусловлено применимостью к объектам произвольной формы и к средам с произвольными свойствами, включая временную дисперсию и анизотропию.

Многие коммерческие пакеты электродинамического моделирования используют метод КРВО, например, FullWAVE компании Synopsys (США), CST Microwave Studio компании CST (США), OptiFDTD компании Optiwave Systems (Канада), FDTD Solutions компании Lumerical (Канада), OmniSim FDTD компании Photon Design (Великобритания), EM-Suite компании Panoramic Technology (США) и другие.

Метод КРВО применяется для дизайна различных приборов и моделирования процессов, например:

- Приборы (резонаторы) для спектрального уплотнения каналов (WDM).
- Дифракционные решётки.
- Светоизлучающие диоды и лазеры.
- Антенны.
- Процессы литографии, создание и оптимизация масок.
- Взаимодействие света с биологическими объектами (биофотоника).
- Рассеяние света.
- Изучение электромагнитных свойств метаматериалов.

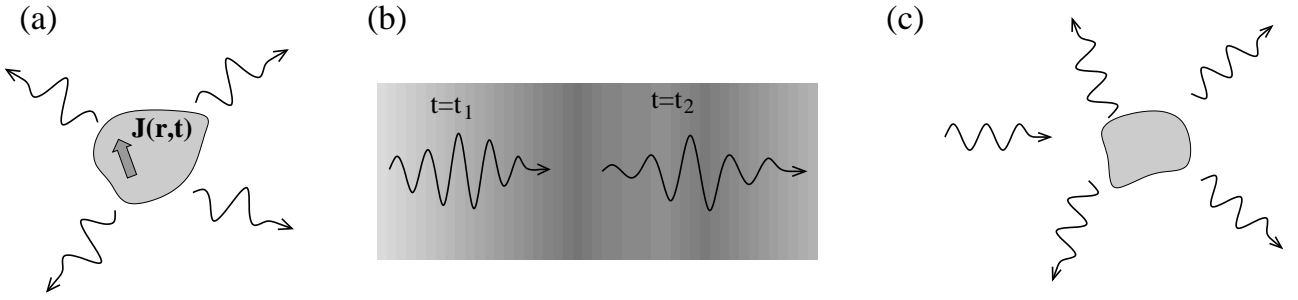


Рис. 1: Иллюстрация различных электродинамических задач. (a) Излучение волн источниками тока с произвольными пространственной и временной зависимостями. (b) Распространение волн в неоднородной среде. (c) Рассеяние волн на некотором объекте.

Многие из перечисленных выше приложений метода КРВО можно представить в виде упрощенных типичных задач, проиллюстрированных на рис. 1:

- Излучение заданных источников тока в среде.
- Распространение электромагнитных импульсов в различных средах.
- Рассеяние импульсов на различных объектах.

Можно здесь привести и пример приложения метода к решению конкретной задачи. Одним из наиболее распространенных методов получения терагерцового излучения является его генерация с использованием оптических импульсов в нелинейных кристаллах. Наводящаяся нелинейная поляризация является источником, который и генерирует волны при его движении. В двумерной модели, в которой поля не зависят от координаты  $z$  и импульс распространяется вдоль  $x$ , можно записать нелинейную поляризацию в виде

$$P_z(x, y, t) = p_0 F(\xi) G(x, y), \quad \text{где} \quad F(\xi) = e^{-\xi^2/\tau^2}, \quad G(y) = e^{-y^2/a_0^2}. \quad (1)$$

Переменная  $\xi = t - x/V_{\text{gr}}$  определяется групповой скоростью движения оптического импульса  $V_{\text{gr}} = c/n_{\text{gr}}$ . Параметр  $\tau$  определяет длительность импульса, а  $a_0$  – величину его фокусировки. Величина поляризации  $p_0$  определяется нелинейными свойствами кристалла и интенсивностью оптического поля. Если групповая скорость движения импульса  $V_{\text{gr}}$  превосходит фазовую скорость терагерцовых волн, определяемую индексом  $n_{\text{THz}}$ , то будет возникать черенковский конус, как и в случае движения заряженной частицы в среде. Пространственное распределение терагерцового излучения показано на рис. 2, где видно формирование черенковского конуса из терагерцовых волн. Половина угла раствора конуса  $\theta$  определяется соотношением  $\sin \theta = n_{\text{gr}}/n_{\text{THz}}$ . Помимо этого, от границы кристалла отходит цилиндрическая волна, которая является результатом переходных процессов. Теперь можно приступить к описанию процедуры решения таких задач.



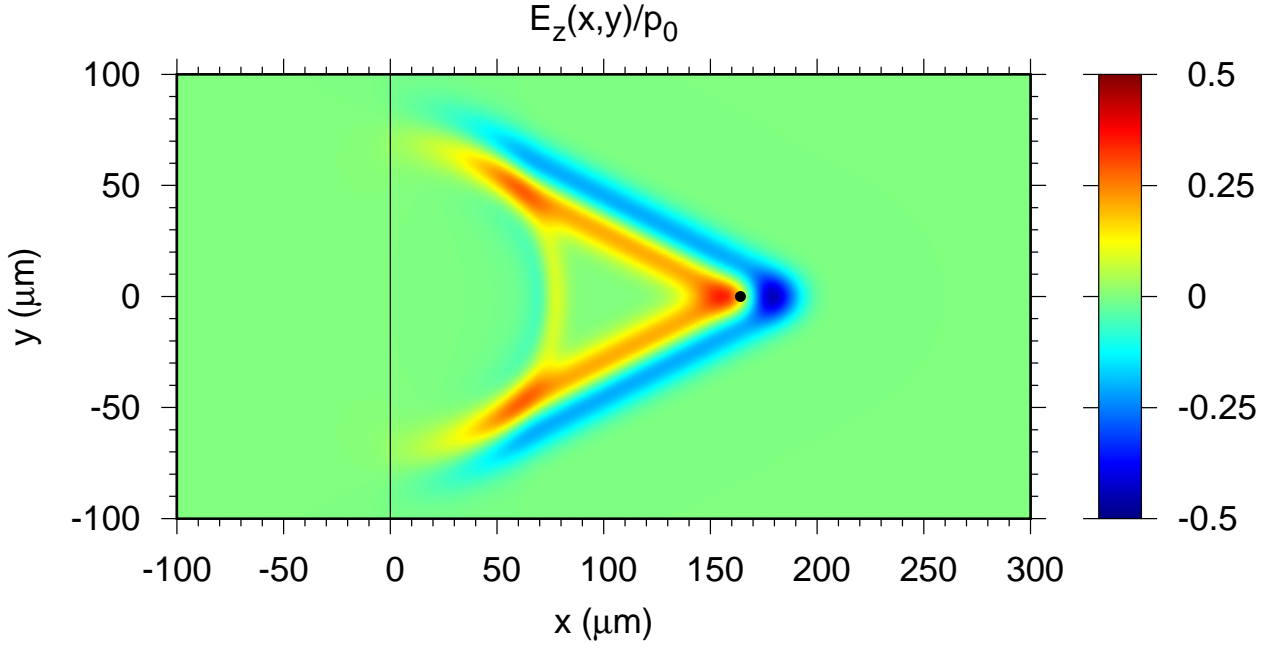


Рис. 2: Типичное распределение поля  $E_z(x, y)$ , формируемое в нелинейном кристалле  $\text{LiNbO}_3$  (при  $x > 0$ ) в некоторый момент времени после падения из вакуума ( $x < 0$ ) на его границу сфокусированного оптического импульса. Центр импульса нелинейной поляризации в момент снимка показан черной точкой. Параметры: длительность  $\tau = 0.1$  пс, ширина  $a_0 = 10$   $\mu\text{m}$ , коэффициент преломления для терагерцовых волн  $n_{\text{THz}} = 5.12$ , групповой индекс в оптическом диапазоне  $n_{\text{gr}} = 2.23$ . Угол раствора конуса равен  $2\theta = 2 \arcsin(n_{\text{gr}}/n_{\text{THz}}) = 51.64^\circ$ .

### 3 Основные принципы решения

Электромагнитное поле описывается уравнениями Максвелла:

$$\nabla \times \mathbf{E} = -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t}, \quad (2a)$$

$$\nabla \times \mathbf{H} = \frac{1}{c} \frac{\partial \mathbf{D}}{\partial t} + \frac{4\pi}{c} \mathbf{J}. \quad (2b)$$

В этих уравнениях:

$\mathbf{E}$  – напряжённость электрического поля,

$\mathbf{H}$  – напряжённость магнитного поля,

$\mathbf{D}$  – электрическая индукция,

$\mathbf{B}$  – магнитная индукция,

$\mathbf{J}$  – плотность электрического тока.

Заметим, что уравнения (2) записаны в гауссовой системе, в которой поля  $\mathbf{E}$ ,  $\mathbf{D}$ ,  $\mathbf{H}$ ,  $\mathbf{B}$  имеют одинаковые размерности. Такой выбор удобен не только для

теоретических выкладок, но и для численного расчёта, так как величины полей имеют сравнимые значения. Часто также записывают сначала уравнения Максвелла в системе СИ, а затем домножают поля на определенные коэффициенты, чтобы привести их к соизмеримым величинам. После численного расчёта полученные поля переводят обратно в единицы СИ. Мы не будем пользоваться системой СИ в данном пособии. Уравнения (2) следует дополнить материальными соотношениями, которые будут обсуждаться позднее.

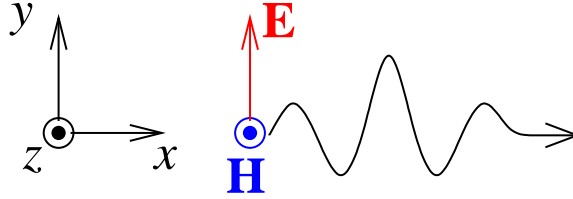


Рис. 3: Распространение импульса вдоль оси  $x$  в свободном пространстве в 1D случае.

Рассмотрим распространение электромагнитного импульса в свободном пространстве в 1D случае, см. рис. 3. Мы ограничимся только компонентами  $E_y$ ,  $D_y$ ,  $J_y$ ,  $H_z$ ,  $B_z$ . Тогда векторные уравнения Максвелла сводятся к

$$\frac{\partial B_z}{\partial t} = -c \frac{\partial E_y}{\partial x}, \quad (3a)$$

$$\frac{\partial D_y}{\partial t} = -c \frac{\partial H_z}{\partial x} - 4\pi J_y. \quad (3b)$$

Далее мы опустим индексы  $x$  и  $y$ , показывающие ориентацию полей, так как каждое поле имеет только одну векторную компоненту. Мы получим

$$\frac{\partial B}{\partial t} = -c \frac{\partial E}{\partial x}, \quad (4a)$$

$$\frac{\partial D}{\partial t} = -c \frac{\partial H}{\partial x} - 4\pi J. \quad (4b)$$

Уравнения (4) – основа построения численного алгоритма. В свободном пространстве  $D = E$  и  $B = H$ .

Для расчёта изменения поля во времени и в пространстве производные в (4) представляются через конечные разности. Производную в произвольной точке можно оценить путем расчёта значений функции в окрестности этой точки. Пусть, например, требуется оценить производную от некоторой функции  $f(x)$  при  $x = x_0$ , см. рис. 4. Мы знаем, что

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + O((x - x_0)^3). \quad (5)$$

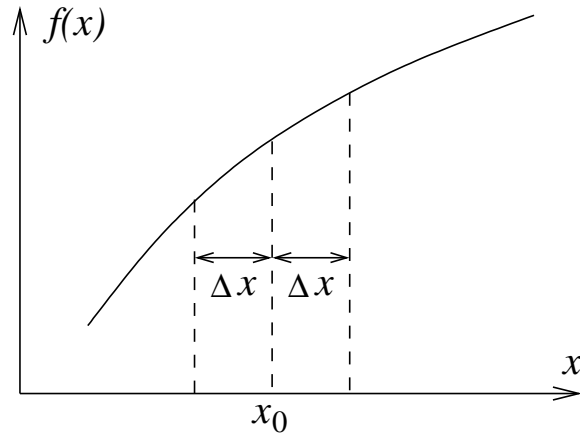


Рис. 4: Схема численного расчёта производной от функции  $f(x)$  при  $x = x_0$ .

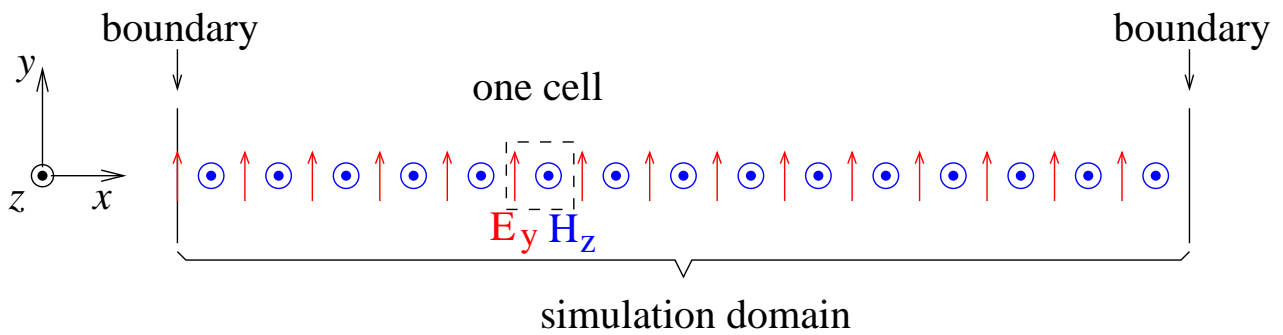


Рис. 5: Разбиение пространства моделирования (simulation domain) на ячейки (cells) и расположение точек (узлов), в которых заданы поля  $E_y$  и  $H_z$ .

Тогда мы можем оценить  $f'(x_0)$ , используя значения  $f(x_0)$  и  $f(x_0 + \Delta x)$ :

$$f'(x_0) = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} + O(\Delta x). \quad (6)$$

Точность такой оценки пропорциональна  $\Delta x$ . Более точный результат, с точностью порядка  $\Delta x^2$ , получается, если использовать центральную разность, то есть значения  $f(x_0 - \Delta x)$  и  $f(x_0 + \Delta x)$ :

$$f'(x_0) = \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} + O(\Delta x^2). \quad (7)$$

Такая аппроксимация производных и будет использоваться в алгоритмах.

Рассмотрим теперь внимательно одно из уравнений, например, (4а). Временное изменение поля  $B$  в произвольной точке определяется пространственной производной от поля  $E$ , которую можно выразить через значения  $E$  в двух соседних точках. С другой стороны, временное приращение поля  $B$  можно записать, используя его значения сдвинутые по времени. Такие рассуждения приводят к тому, что для применения центральных производных в

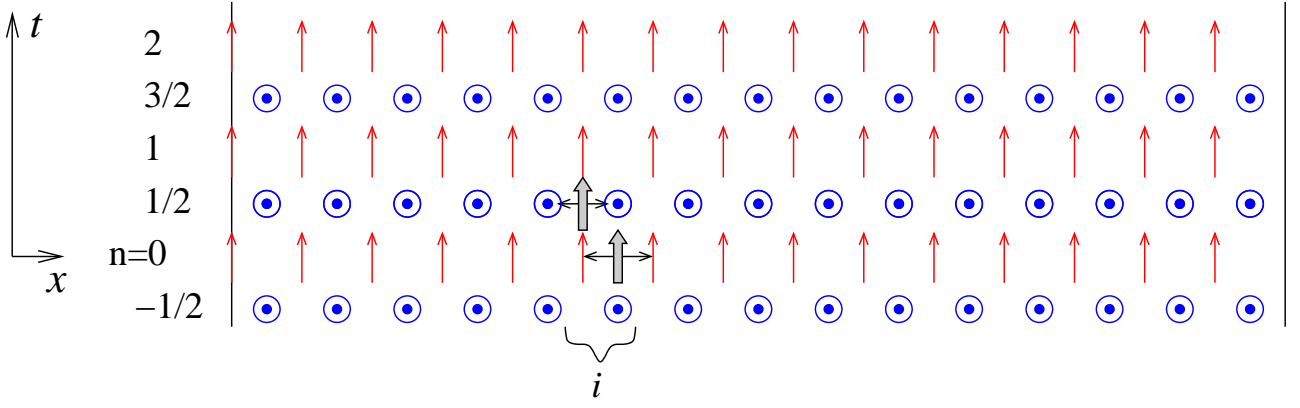


Рис. 6: Схема алгоритма пересчёта поля.

пространстве удобно задавать значения полей  $E$  и  $B$  на одномерных сетках, которые смещены по отношению друг к другу, см. рис. 5. Таким же образом надо выбирать и моменты времени. Поле  $D$  задается в тех же точках и в те же моменты времени, что и  $E$ , а поле  $H$  – где и  $B$ .

Обычно считается, что полная вычислительная сетка состоит из ячеек размера  $\Delta x$ , в каждой из которых заданы поля  $E$  и  $H$  в смещенных точках. Предположим, что  $\Delta t$  – шаг по времени. Тогда мы можем задать поле  $E$  в моменты времени  $t = n\Delta t$ , где  $n = 0, 1, 2, \dots$ , а поле  $H$  – в моменты  $t = (n + 1/2)\Delta t$ . Будем считать, что  $E_i^n$  – значение поля  $E$  в  $i$ -ой ячейке в момент времени  $t = n\Delta t$  и  $H_i^{n+1/2}$  – значение поля  $H$  в  $i$ -ой ячейке в момент времени  $t = (n + 1/2)\Delta t$ . Подчернем еще раз, что в такой записи поля  $E$  и  $H$  имеют одинаковый пространственный индекс и принадлежат одной ячейке  $i$ , но в действительности они задаются в точках, отстоящих на  $\Delta x/2$  друг от друга.

Мы можем записать частные производные с помощью конечных разностей с учётом введённых обозначений. Уравнения (4) становятся дискретными и выглядят (без учета тока) как

$$B_i^{n+1/2} - B_i^{n-1/2} = -\frac{c\Delta t}{\Delta x} (E_{i+1}^n - E_i^n), \quad (8a)$$

$$D_i^{n+1} - D_i^n = -\frac{c\Delta t}{\Delta x} (H_i^{n+1/2} - H_{i-1}^{n+1/2}). \quad (8b)$$

Далее, уравнения (8) можно переписать в следующем виде:

$$B_i^{n+1/2} = B_i^{n-1/2} - \xi (E_{i+1}^n - E_i^n), \quad (9a)$$

$$D_i^{n+1} = D_i^n - \xi (H_i^{n+1/2} - H_{i-1}^{n+1/2}), \quad (9b)$$

где был введён параметр  $\xi = c\Delta t/\Delta x$ .

С учетом  $D = E$  и  $B = H$  мы получаем из (9) уравнения для свободного

пространства:

$$H_i^{n+1/2} = H_i^{n-1/2} - \xi (E_{i+1}^n - E_i^n), \quad (10a)$$

$$E_i^{n+1} = E_i^n - \xi \left( H_i^{n+1/2} - H_{i-1}^{n+1/2} \right), \quad (10b)$$

Уравнения (9) являются основными дискретными уравнениями для моделирования распространения электромагнитного импульса в свободном пространстве. Алгоритм продвижения во времени можно представить в виде схемы

$$\dots \rightarrow H^{n+1/2} \rightarrow E^{n+1} \rightarrow H^{n+3/2} \rightarrow E^{n+2} \rightarrow \dots \quad (11)$$

Эта схема проиллюстрирована на рис. 6. Например, для нахождения  $H_i^{1/2}$  берётся его предыдущее значение  $H_i^{-1/2}$  и значения полей  $E_i^0, E_{i+1}^0$  в соседних точках. То же делается и для нахождения  $E_i^1$ .

Как видно из (9a), изменение поля  $B$  в определённой точке зависит от значений  $E$  в точках справа и слева от нее. То же справедливо и для полей  $D$  и  $H$  в (9b). Таким образом, возникает вопрос, как ограничить область моделирования. Для этого требуются некоторые граничные условия. Для простоты мы предположим, что область ограничена стенками, на который зануляется поле  $E$ , то есть имеется идеальный металлический проводник. Такое условие ведёт к отражению импульсов от границ области. Далее будут рассмотрены граничные условия, которые позволяют избежать такое отражение и получить выход импульсов из области моделирования, как это и происходит в неограниченном пространстве.

Для начала моделирования по схеме (11) требуется также задать начальные поля. Предположим, что имеется некоторый импульс, который распространяется вдоль  $x$ . Тогда мы знаем, что в свободном пространстве

$$E_y(x, t) = H_z(x, t) = f(t - x/v), \quad (12)$$

где  $f$  – произвольная функция и  $v = c$  – скорость распространения импульса. Функция (12) позволяет рассчитать поля  $E_y(x, t = 0), H_z(x, t = -\Delta t/2)$  и создать распределения  $E_i^0, H_i^{-1/2}$ , которые обеспечивают возможность продвижения во времени согласно (11). Далее в пособии мы будем использовать

$$f(t) = \cos(\omega_0 t) e^{-t^2/\tau^2} \quad (13)$$

где  $\omega_0$  – угловая частота импульса и  $\tau$  – его продолжительность.

Таким образом, с учётом начальных и граничных условий мы можем теперь применить разностную схему (10) для моделирования распространения импульса.

## 4 Пример 1: Распространение импульса в свободном пространстве

### 4.1 Разработка программы моделирования

Рассмотрим конкретный пример моделирования распространения импульса в свободном пространстве. Остановимся сначала подробно на том, как воплотить алгоритмы, разработанные в разделе 3, в программы на языке C. В общем случае программы на C состоят из отдельных функций, которые взаимодействуют друг с другом. Программы численного моделирования могут использовать десятки и сотни различных функций. Поэтому при написании программ особое внимание требуется уделять выделению операций в функции и разбивке функций по файлам. Оптимальная разбивка позволяет собирать различные программы из уже существующих функций и файлов, как будет видно по мере того, как мы будем совершенствовать алгоритмы моделирования в последующих разделах.

Алгоритм моделирования (11) подразумевает хранение значений  $E$  и  $H$  в виде массивов, содержимое которых изменяется последовательно во времени. При этом заранее неизвестно, сколько требуется осуществить шагов для того, чтобы импульс прошел определенное расстояние. Поэтому удобно предоставить пользователю возможность задания начального количества шагов, их добавления или остановки выполнения программы. Такое решение будет приниматься в зависимости от прохождения импульсом какого-либо расстояния. Удобно расположить функцию управления в одном файле, а функции, отвечающие за алгоритм – в другом. Функция управления будет отвечать за выделение памяти, вывод данных и взаимодействие с пользователем.

Разработанная программа состоит из следующих файлов:

**free\_space.c:** функция управления.

**fdtd\_1d\_maxwell.c** и **fdtd\_1d\_maxwell.h:** функции алгоритма и их описание.

**pulse.c** и **pulse.h:** функции создания импульса и их описание.

**output.c** и **output.h:** функции вывода в файлы и их описание.

**constants.c:** задание постоянных величин.

Давайте теперь изучим содержимое файлов.

Файл **free\_space.c** с функцией управления **free\_space()**:

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 #include<math.h>
5
6 #include"fDTD_1d_maxwell.h"
7 #include"pulse.h"
8 #include"output.h"
9
10 extern const double cspeed;
11
12 /*****
13 void free_space(void){
14
15     const char *tag="v1";    // label for output files
16
17     /** Optical pulse ***/
18     double tau = 1.0;    // fs, width of the pulse
19     double w0=0;    // no oscillating component
20
21     /** Computational parameters ***/
22     int    Nx = 4000;    // number of cells along x
23     double dx = 20.0;    // nm
24     double xi = 0.9;
25     int    ix0 = 1000;    // cell number of the pulse center at t=0
26     int    No = 1000;    // defines the output rate
27
28     /** start execution ***/
29     double dt = xi*dx/cspeed; // in fs
30     printf("dx=%.12e nm, dt=%.12e fs\n", dx, dt);
31
32     /** arrays for the fields ***/
33     double *fields = malloc(2*Nx*sizeof(double));
34     double *Hz = fields+0*Nx;
35     double *Ey = fields+1*Nx;
36
37     int T=0; // total steps
38
39     create_initial_dist(Nx, Ey, Hz, dx, dt, cspeed, ix0, tau, w0);
40     output_Ey_vs_x(Nx, Ey, 0, dx, tag);
41
42     for(;;){
43         printf("Number of steps to run (<=0 to exit) -> "); fflush(stdout);
44         int steps;
45         scanf("%d", &steps);
46         if(steps<=0) break;
47
48         printf("Making %d steps\n", steps);

```

```

49     for(int n=0; n<steps; n++, T++){
50         update_Bz(Nx, Hz, Ey, xi);    // find Bz at n+1/2
51         update_Dy(Nx, Ey, Hz, xi);    // find Ey at n+1
52
53         /* output of Hz */
54         if((T+1)%No == 0){
55             printf("Elapsed time -> %g fs (%d steps)\n", dt*(T+1), T+1);
56             output_Ey_vs_x(Nx, Ey, T+1, dx, tag);
57         }
58     }// end of n loop
59 }// end of global loop
60
61     free(fields);
62 }
63
64 /*****
65 int main(void){
66
67     free_space();
68
69     return 0;
70 }
71
72 /*** END OF FILE *****/

```

В функции `free_space()` задаются основные параметры:  $N_x$  – количество ячеек вдоль оси  $x$ ,  $dx$  – пространственный шаг,  $\xi$  – параметр  $\xi$ ,  $ix_0$  – номер ячейки, где расположен пик импульса в начальный момент времени,  $\tau$  и  $w_0$  – параметры импульса. Переменная  $T$  содержит полное количество шагов. Программа спрашивает пользователя, сколько шагов требуется осуществить и даёт возможность остановки после их выполнения. После каждых  $N_o$  шагов программа записывает состояние поля  $E$  в файл. Имя файла содержит значение  $T$ , что позволяет хранить информацию о динамике в различные моменты времени. Имя файла также имеет дополнительную метку, заданную переменной  $tag$ , что позволяет отличать файлы при запуске программы с изменёнными параметрами. Вверху файла включаются файлы с описанием функций, которые расположены в других файлах.

Файл `fdtd_1d_maxwell.c` с основными функциями алгоритма (9):

```

1 /*****
2 void update_Bz(int Nx, double *Bz, const double *Ey, double xi){
3
4     for(int i=0; i<Nx-1; i++){// except the last point
5         Bz[i] +=-xi*(Ey[i+1]-Ey[i]);
6     }
7     int i=Nx-1; // last point

```



```

8   Bz[i] +=-xi*(0.0 - Ey[i]);
9   }
10
11  /*****
12  void update_Dy(int Nx, double *Dy, const double *Hz, double xi){
13
14     // Ey[0]=0 always and not updated
15     for(int i=1; i<Nx; i++){
16         Dy[i] +=-xi*(Hz[i]-Hz[i-1]);
17     }
18 }
19 /*** END OF FILE *****/

```

Функции `update_Bz()` и `update_Dy()` изменяют поля во времени. Заметим, что мы предположили, что происходит изменение полей  $B$  и  $D$ , а не  $E$  и  $H$ , что удобнее для дальнейшего использования этих функций.

Файл `fdtd_1d_fspace_basic.h` с описанием функций из `fdtd_1d_maxwell.c`:

```

1 void update_Bz(int Nx, double *Bz, const double *Ey, double xi);
2 void update_Dy(int Nx, double *Dy, const double *Hz, double xi);

```

Файл `pulse.c` с функциями создания импульса:

```

1 #include<stdio.h>
2 #include<math.h>
3
4 /*****
5 double pulse(double x, double t, double speed, double tau, double w0){
6     double t1 = t - x/speed;
7     double a = t1/tau;
8     return exp(-a*a)*cos(w0*t1);
9 }
10
11 /*****
12 void create_initial_dist(int Nx, double *Ey, double *Hz, double dx,
13                         double dt, double speed, int ix0, double tau,
14                         double w0){
15     /*
16     * Creates Ey[T=0] and Hz[T=-0.5]
17     */
18
19     printf("Pulse parameters: w0=%g rad/fs, tau=%g fs\n", w0, tau);
20
21     for(int i=0; i<Nx; i++){
22         Ey[i] = pulse(dx*(i+0.0-ix0), 0*dt, speed, tau, w0);
23         Hz[i] = pulse(dx*(i+0.5-ix0), -0.5*dt, speed, tau, w0);

```

```

24 }
25 Ey[0]=0; // to satisfy b.c.
26 }
27
28 /*** END OF FILE *****/

```

Функция `create_initial_dist()` задаёт начальное распределение полей.  
 Файл `pulse.h` с описанием функций из `pulse.c`:

```

1 double pulse(double x, double t, double speed, double tau, double w0);
2 void create_initial_dist(int Nx, double *Ey, double *Hz, double dx,
3                          double dt, double speed, int ix0, double tau,
4                          double w0);

```

Файл `output.c` с функциями вывода в файлы:

```

1 #include<stdio.h>
2 #include<string.h>
3
4 /*** END OF FILE *****/
5 void output_Ey_vs_x(int Nx, const double *Ey, int T, double dx,
6                    const char *tag){
7
8     char fname[100];
9     sprintf(fname, "Ey_vs_x_T=%d_%s.dat", T, tag);
10    FILE *fp=fopen(fname, "w");
11    fprintf(fp, "# 1: x (micron) | 2 : Ey (a.u.) \n");
12    for(int i=0; i<Nx; i++){
13        double x=i*dx; // coordinate for Ey
14        fprintf(fp, "% .12e % .12e\n", 1e-3*x, Ey[i]);
15    }
16    fclose(fp);
17    printf("Completed writing to file \"%s\"\n", fname);
18 }
19 /*** END OF FILE *****/

```

Файл `output.h` с описанием функций из `output.c`:

```

1 void output_Ey_vs_x(int Nx, const double *Ey, int T, double dx,
2                    const char *tag);

```

Файл `constants.c` задает скорость света  $c$  и значение  $\pi$ :

```

1 const double cspeed=299.792458; // nm/fs
2 const double pi=3.141592653589793;

```

После разбивки функций по файлам, обсудим теперь структуру директорий (каталогов), которая удобна для хранения исходных файлов, компиляции программы и выполнения расчётов. Можно использовать следующую иерархическую структуру директорий:

```
1 free_space/
2 |-- bin
3 |   '-- fdttd_1d.exe
4 |-- data [137 entries exceeds filelimit, not opening dir]
5 |-- makefile
6 |-- obj
7 |   |-- constants.o
8 |   |-- fdttd_1d_maxwell.o
9 |   |-- free_space.o
10 |   |-- output.o
11 |   '-- pulse.o
12 '-- src
13     |-- constants.c
14     |-- fdttd_1d_maxwell.c
15     |-- fdttd_1d_maxwell.h
16     |-- free_space.c
17     |-- output.c
18     |-- output.h
19     |-- pulse.c
20     '-- pulse.h
21
22 4 directories, 15 files
```

В директории `src/` хранятся исходные файлы, в `obj/` – объектные файлы, в `bin/` – выполняемые файлы, а в `data/` – файлы с результатами расчётов.

Для создания выполняемого файла рекомендуется использовать стандартную команду `make`, которая выполняет инструкции из файла `makefile`. Преимущество использования команды `make` заключается в том, что не надо запоминать все опции компиляции файлов, а только один раз их записать в `makefile`. Создание выполняемого файла состоит из двух шагов. Сначала проходит компиляция файлов с исходным кодом и получение из них объектных файлов. Затем – соединение объектных файлов и получение из них выполняемого файла.

Содержание файла `makefile`:

```
1 CC=gcc
2 GCCFLAGS= -O3 -Wall -std=c99
3
4 all : free_space
```

```

5
6 OBJFS = obj/free_space.o  obj/fdtd_1d_maxwell.o  obj/pulse.o  \
7         obj/output.o      obj/constants.o
8 free_space: $(OBJFS)
9             $(CC) $(GCCFLAGS) -o bin/free_space.exe $(OBJFS) -lm
10            @echo -e 'Created executable "bin/free_space.exe"\n'
11
12 ### creation of object files
13 obj/%.o : src/%.c
14         $(CC) -c $(GCCFLAGS) $< -o $@
15
16 clean:
17         rm -f obj/*.o bin/*.exe

```

В файле используется следующая структура:

цель: зависимости  
[tab] команда

Обратите внимание, что строка с командой должна всегда начинаться с табуляции.

Выполняемый файл создается в директории `bin/` после выполнения `make` из директории `free_space/`. Выполнять программу удобно из директории `data/`, куда и будут записываться все файлы с данными. После запуска программы используя `../bin/fdtd_1d.exe` появляется строка, запрашивающая требуемое количество шагов. Затем программа выполнит введённое количество шагов и повторит запрос. Закончить выполнение можно введя число  $\leq 0$ . При выполнении программа осуществляет вывод данных в файлы, что позволяет сразу же анализировать результаты.

## 4.2 Результаты моделирования и их обсуждение

Возьмём сначала область моделирования с  $N_x=4000$  ячейками и запустим гауссов импульс. Типичные распределения полей в различные моменты времени показаны на рис. 7. Как и ожидалось, созданный в начальный момент импульс распространяется вправо. Достигнув границы, он отражается с изменением знака и начинает бежать влево.

Давайте проследим за изменением формы импульса более внимательно. Для этого мы увеличим область расчёта, чтобы дать импульсу больше места для распространения. Так как в моделируемой среде нет дисперсии, то импульс не должен меняться в процессе распространения. Для проверки этого результата удобно строить зависимость поля от величины  $x - (i0 + \xi T)\Delta x$ , то есть в системе координат, совпадающей с пиком импульса. Такие результаты представлены на рис. 8. Видно, что при большом количестве шагов импульс

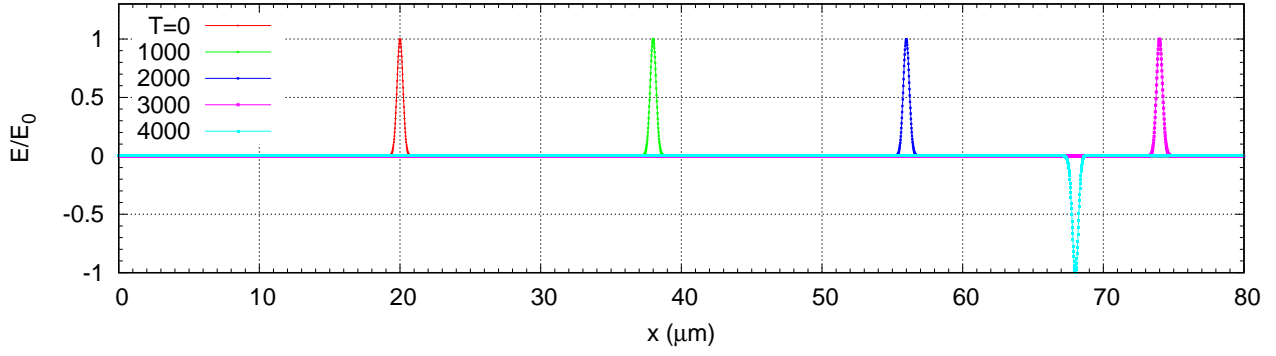


Рис. 7: Распространение импульса в свободном пространстве в одномерном случае при  $\tau = 1$  фс,  $\Delta x = 20$  нм,  $N_x = 4000$ ,  $\xi = 0.9$ .

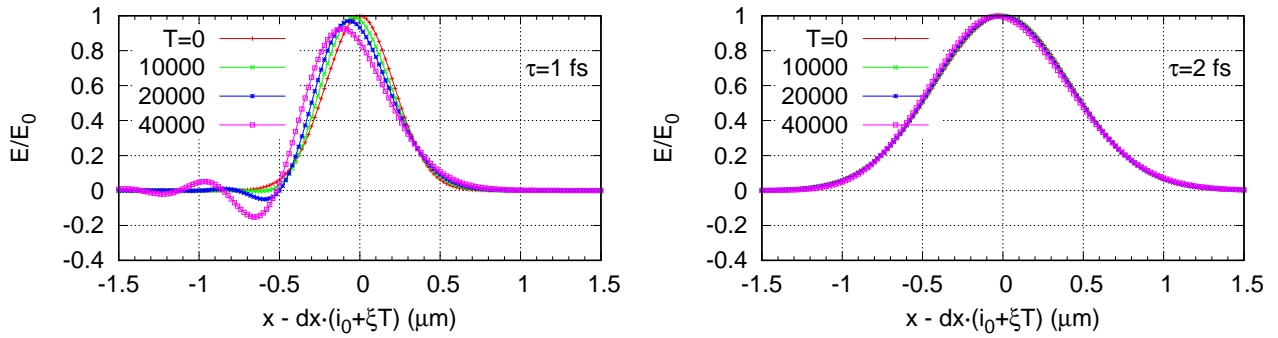


Рис. 8: Изменение формы импульса в процессе его распространения при  $\Delta x = 20$  нм,  $\xi = 0.9$  и  $\tau = 1, 2$  фс.

деформируется. Причём деформация становится более заметной с уменьшением ширины импульса  $\tau$ .

Давайте теперь попробуем изменить временной шаг и посмотреть, как это скажется на результате, см. рис. 9. Мы видим, что при меньшем значении параметра  $\xi$ , а значит и меньшем временном шаге  $\Delta t$ , импульс деформируется больше при прохождении одинакового расстояния. При меньшем  $\xi$  при этом требуется, естественно, выполнить и больше шагов.

Таким образом, мы разработали простейшую программу для моделирования распространения электромагнитного импульса и продемонстрировали ее функциональность. При достаточно малых значениях времени мы получили хорошее сохранение формы импульса при его распространении, что соответствует точному решению уравнений Максвелла. При большом количестве шагов импульс начинает искажаться. Степень его искажения при прохождении определённого расстояния возрастает как с уменьшением его ширины  $\tau$ , так и временного шага  $\Delta t$ . Причина такого поведения будет рассмотрена в следующем разделе.

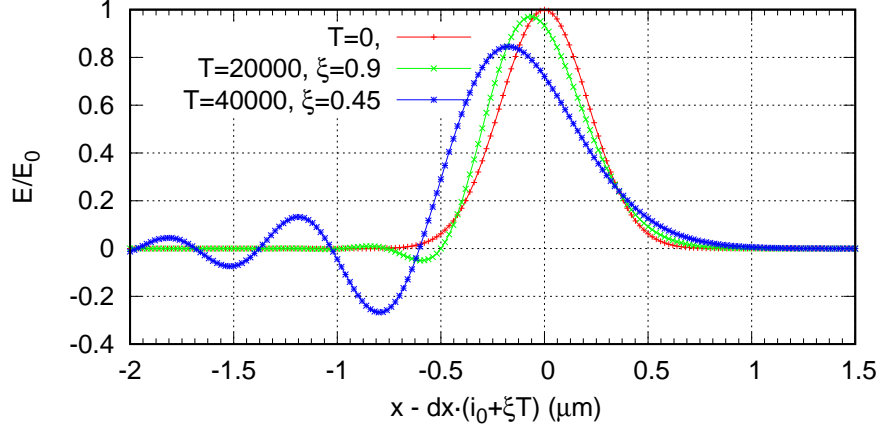


Рис. 9: Изменение формы импульса в процессе его распространения при  $\Delta x = 20$  нм,  $\xi = 0.9, 0.45$  и  $\tau = 1$  фс.

## 5 Выбор пространственного и временного шагов

Для выяснения причин искажения импульса рассмотрим систему дискретных уравнений (10). Она описывает распространение сигнала между заданными точками. Любой сигнал можно представить в виде собственных функций такой системы. Найдём свойства этих функции. Для этого подставим общую форму решения в комплексном виде

$$E(x, t) = E_0 e^{-i\omega t + ikx}, \quad (14a)$$

$$H(x, t) = H_0 e^{-i\omega t + ikx}, \quad (14b)$$

в систему (10). Мы получим

$$H_0 e^{ik\Delta x/2} \left( e^{-i\omega\Delta t/2} - e^{i\omega\Delta t/2} \right) = -\xi E_0 \left( e^{ik\Delta x} - 1 \right), \quad (15a)$$

$$E_0 \left( e^{-i\omega\Delta t} - 1 \right) = -\xi H_0 e^{-i\omega\Delta t/2} \left( e^{ik\Delta x/2} - e^{-ik\Delta x/2} \right), \quad (15b)$$

или

$$\begin{pmatrix} \xi \sin \frac{k\Delta x}{2} & -\sin \frac{\omega\Delta t}{2} \\ \sin \frac{\omega\Delta t}{2} & -\xi \sin \frac{k\Delta x}{2} \end{pmatrix} \begin{pmatrix} E_0 \\ H_0 \end{pmatrix} = 0. \quad (16)$$

Система (16) имеет решение, если детерминант равен нулю. Это условие приводит к

$$\xi^2 \sin^2 \frac{k\Delta x}{2} = \sin^2 \frac{\omega\Delta t}{2}. \quad (17)$$

Понятно, что два знака при решении уравнения (17) соответствуют волнам, распространяющимся в противоположные стороны. Для анализа искажения достаточно взять одну волну, бегущую вдоль  $+x$  и удовлетворяющую

$$\xi \sin \frac{k\Delta x}{2} = \sin \frac{\omega\Delta t}{2}. \quad (18)$$

Уравнение (18) – дисперсионное уравнение для волн в дискретной системе (10).

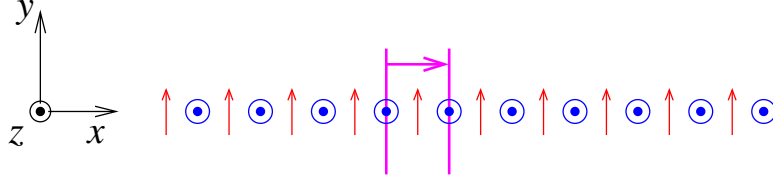


Рис. 10: Распространение сигнала по 1D сетке.

Если мы создадим волну с некоторым волновым числом  $k$ , то ее изменение во времени определяется частотой, получаемой из (18). Для волн с  $k\Delta x = \pi$  уравнение (18) становится  $\xi = \sin(\omega\Delta t/2)$  и при  $\xi > 1$  имеет решение только на мнимых частотах. Соответствующие волны будут нарастать со временем и, в конце концов, достигать бесконечно больших значений. Такое поведение не соответствует модели, описываемой непрерывными уравнениями (4). Таким образом, выбор  $\xi > 1$  приводит к расхождению алгоритма. Критерий устойчивости можно интерпретировать и на сетке, см. рис. 10. За один временной шаг  $\Delta t$  алгоритма информация о поле распространяется в соседнюю ячейку, то есть на расстояние  $\Delta x$ . Понятно, что правильное описание распространения волн возможно, если только волна бежит с равной или меньшей скоростью, то есть  $c \leq \Delta x/\Delta t$  или  $\xi \leq 1$ .

Теперь предположим, что  $\xi \leq 1$ . Для анализа распространения удобно ввести эффективный коэффициент (показатель) преломления  $n$ , то есть  $k = n\omega/c$ , который описывает численную дисперсию. Обозначая  $\delta = \Delta x\omega/c$ , мы получаем

$$n = \frac{2}{\delta} \arcsin \left( \frac{1}{\xi} \sin \left( \delta \frac{\xi}{2} \right) \right). \quad (19)$$

Зависимость  $n - 1$  от величины пространственного шага  $\Delta x$  при различных значениях параметра  $\xi$  показана на рис. 11. В непрерывной системе (4) все волны распространяются с  $n = 1$ . В дискретной системе, описываемой уравнениями (10), фазовые скорости волн будут зависеть от пространственной и временной дискретизаций. Как видно из рис. 11, для приближения к правильному результату необходимо выбирать  $\Delta x/\lambda \ll 1$ . То есть шаг пространственной дискретизации должен быть много меньше, чем минимальная

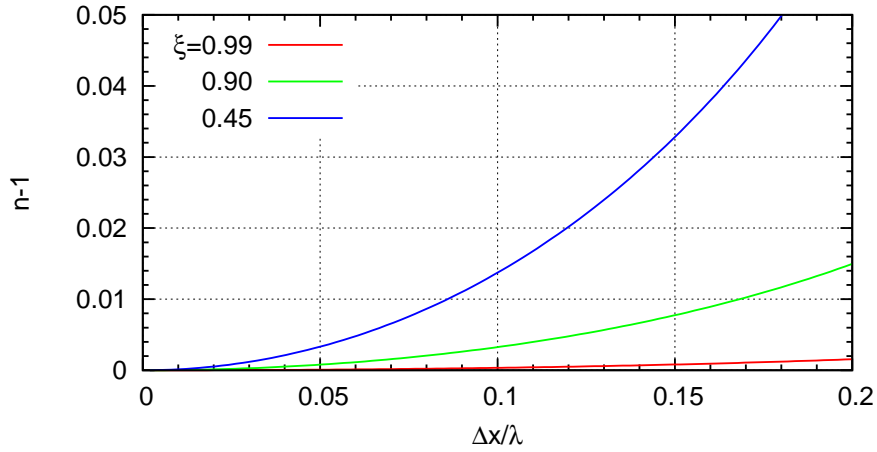


Рис. 11: Зависимость эффективного (численного) коэффициента преломления  $n$  от величины пространственного шага  $\Delta x$  при различных значениях параметра  $\xi$ .

длина волны в разложении импульса по пространственным гармоникам. При этом рис. 11 показывает, что дисперсия уменьшается при  $\xi \rightarrow 1$ , то есть уменьшение временного шага ведёт к увеличению численной дисперсии. Это согласуется и с результатами расчёта, показанными на рис. 9.

Таким образом, при выборе шагов  $\Delta x$  и  $\Delta t$  необходимо пользоваться параметрами моделируемого импульса. Его ширина позволяет получить максимальную длину волны в его спектре. Пространственный шаг  $\Delta x$  должен быть выбран значительно меньше, чем эта длина. На практике выбирают  $\Delta x/\lambda \lesssim 1/20$ . Временной шаг надо выбирать из условия  $\xi \leq 1$ . При этом надо иметь в виду проявление эффектов численной дисперсии при большом количестве шагов.

## 6 Моделирование сред без дисперсии

Изучив распространение импульса в свободном пространстве в разделах 3 и 4, мы можем теперь приступить к изучению его прохождения через различные неоднородности. Для этого надо ввести распределение диэлектрической проницаемости (или показателя преломления) среды, то есть дополнить уравнения Максвелла материальными соотношениями. Конкретная задача, которую мы будем рассматривать, показана на рис. 12, где импульс падает на слой диэлектрика с заданной проницаемостью. Помимо моделирования распространения импульса, попробуем найти зависимости коэффициентов прохождения и отражения от частоты.

Естественно, при наличии неоднородности уравнения Максвелла и их дискретная форма остаются прежними. Нам лишь надо дополнить их соот-



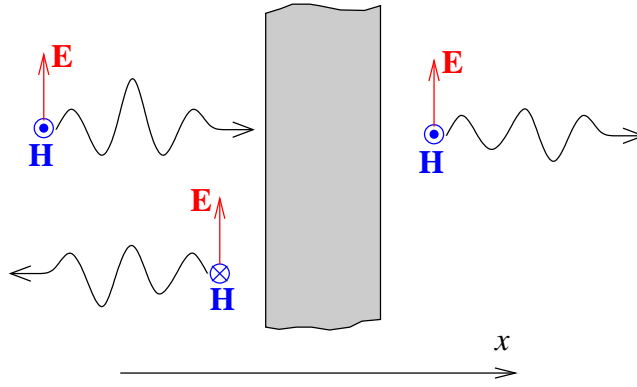


Рис. 12: Геометрия задачи о падении импульса на диэлектрический слой.

ношением для нахождения  $E$  через  $D$ . Предполая, что среда описывается материальным уравнением

$$D(x, t) = \varepsilon(x)E(x, t), \quad (20)$$

дискретная форма принимает вид

$$E_i^n = \frac{D_i^n}{\varepsilon_i}, \quad (21)$$

где  $\varepsilon_i$  – значение проницаемости в точке задания поля. Таким образом, наличие диэлектрической проницаемости приводит к незначительному усложнению алгоритма пересчета полей:

$$\dots \rightarrow H^{n+1/2} \rightarrow D^{n+1} \rightarrow E^{n+1} \rightarrow H^{n+3/2} \rightarrow D^{n+2} \rightarrow E^{n+2} \rightarrow \dots \quad (22)$$

Следует отметить, что мы могли бы полностью исключить поле  $D$  из уравнений Максвелла, используя (20), и работать только с полями  $E$  и  $H$ . В этом случае проницаемость вошла бы в уравнения Максвелла. Такой подход часто применяется, так как даёт возможность исключить материальное уравнение для сред без дисперсии из численного алгоритма. С другой стороны, алгоритм (22) является более общим и применим даже для сред с дисперсией, если заменить уравнение (21) на соответствующее материальное уравнение. Для сред без дисперсии использование уравнения (21) в алгоритме (22) практически не приводит к существенному увеличению времени расчёта, а поэтому имеет смысл пользоваться сразу более общим подходом.

Важно, что при наличии диэлектриков критерий сходимости алгоритма изменяется. Для диэлектрической среды без дисперсии с восприимчивостью  $\varepsilon$  критерий устойчивости становится

$$\frac{c\Delta t}{\sqrt{\varepsilon}\Delta x} = \frac{\xi}{\sqrt{\varepsilon}} < 1 \quad \text{или} \quad \frac{\Delta x}{\Delta t} > v_{ph}, \quad (23)$$

где  $v_{ph}$  – фазовая (и групповая) скорость волн. Таким образом, при наличии диэлектрика можно брать больший временной шаг  $\Delta t$ . Как правило, в области моделирования присутствуют диэлектрики с различными значениями  $\varepsilon$ . В этом случае временной шаг  $\Delta t$  определяется наименьшим значением  $\varepsilon$ . С другой стороны, пространственный шаг  $\Delta x$  определяется наименьшим масштабом поля, то есть длиной волны в диэлектрике с наибольшим значением  $\varepsilon$ .

## 7 Получение частотных характеристик

На практике часто требуется получить свойства среды при падении монохроматического сигнала, то есть частотные характеристики. К ним относятся, например, коэффициенты отражения и прохождения для задачи, показанной на рис. 12. Так как расчёт ведётся во временной области, то для этого можно применить преобразование Фурье. Преобразование Фурье задаёт соотношение между некоторой функцией во времени  $f(t)$  и ее комплексным спектром  $\tilde{f}(\omega)$ . Мы будем пользоваться следующими соотношениями для непрерывных функций:

$$f(t) = \int_{-\infty}^{+\infty} d\omega \tilde{f}(\omega) e^{-i\omega t}, \quad \tilde{f}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} dt f(t) e^{i\omega t}. \quad (24)$$

Для получения спектра сигнала, заданного через равные промежутки времени, мы должны выбрать набор требуемых частот  $\omega_j$  и просто просуммировать вклады от сигнала с экспоненциальными коэффициентами:

$$\tilde{f}(\omega_j) = \frac{\Delta t}{2\pi} \sum_{n=0}^T f(t_n) e^{i\omega_j t_n}. \quad (25)$$

Расчёт по формуле (25) осуществляется после выбора точки в пространстве, где требуется найти спектр электрического или магнитного поля. Для правильности расчёта временной интервал, ограниченный значениями  $n = 0$  и  $n = T$ , в сумме (25) должен быть таким, чтобы импульс полностью в него помещался, то есть значение импульса должно быть пренебрежимо мало как в начальный, так и в конечный моменты времени. При этом Фурье спектр достигает стационарного значения и не меняется с дальнейшим увеличением количества шагов  $T$ .

Следует отметить, что для расчёта спектра нет необходимости сначала сохранять все значения функции  $f(t_n)$ , а потом применять (25). Достаточно просто добавлять их вклад во все компоненты  $\tilde{f}(\omega_j)$  в каждый момент

времени. При сохранении значений во времени возможно также использование алгоритма быстрого преобразования Фурье. При этом набор частот будет определяться величиной временного шага и числом шагов. Надо также учитывать, что для нахождения, например, коэффициента отражения требуется знать не только спектр отражённого импульса, но и спектр падающего.

## 8 Пример 2: Падение импульса на слой диэлектрика

### 8.1 Разработка программы моделирования

Теперь применим методики, разработанные в разделах 6 и 7, для численного нахождения коэффициентов отражения и прохождения при падении волны на однородный слой с проницаемостью  $\varepsilon$ , находящийся в свободном пространстве. Для этого надо изменить файл управления, а также написать новые функции для материального уравнения и преобразования Фурье.

Файл `slab.c` с функцией управления:

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 #include<math.h>
5 #include<complex.h>
6
7 typedef double complex dcomplex;
8
9 #include"fdd_1d_maxwell.h"
10 #include"pulse.h"
11 #include"output.h"
12 #include"mater.h"
13 #include"set_mem.h"
14 #include"rfourier.h"
15
16 extern const double cspeed, pi;
17
18 /*****
19 void slab(void){
20
21     const char *tag="v1"; // label for output files
22     double eslab = 4.0; // permittivity of the slab
23
24     /** Optical pulse **/
25     double lambda0 = 1000; // nm
26     double tau = 8; // fs, width of the pulse
```

```

27
28  /*** Computational parameters ***/
29  double dx = 20.0;    // nm
30  int    Nx = 20000;   // number of cells along x
31  int    ix0 = 9000;   // center of the pulse at t=0
32
33  int    Nslab = 200;   // width of the slab
34  int    si1 = 10000;  // start of the slab
35  int    si2 = si1+Nslab-1; // end of the slab
36
37  int    fi1 = 7500;   // location of fourier transform
38  int    fi2 = si2+10; // location of fourier transform
39
40  double xi = 0.9;
41  int    No = 200;    // defines the output rate
42
43  /*** start execution ***/
44  double w0 = 2*pi*cspeed/lambda0; // rad/fs
45  double dt = xi*dx/cspeed;        // fs
46  printf("dx=%.12e nm, dt=%.12e fs\n", dx, dt);
47
48  /*** arrays for the fields ***/
49  double *fields = malloc(3*Nx*sizeof(double));
50  double *Hz = fields+0*Nx;
51  double *Ey = fields+1*Nx;
52  double *Dy = fields+2*Nx;
53
54  double *eps = malloc(Nx*sizeof(double));
55
56  create_slab(Nx, eps, si1, si2, eslab);
57  output_eps_x(Nx, eps, dx, tag);
58
59  create_initial_dist(Nx, Dy, Hz, dx, dt, cspeed, ix0, tau, w0);
60
61  update_Ey(Nx, Ey, Dy, eps);
62  output_Ey_vs_x(Nx, Ey, 0, dx, tag);
63
64  double wmin = 0.8*w0; // rad/fs
65  double wmax = 1.2*w0; // rad/fs
66  int Nw=200;
67
68  dcomplex *ftall=malloc(2*Nw*sizeof(dcomplex)); // keeps Fourier
69  dcomplex *ft1 = ftall + 0*Nw;
70  dcomplex *ft2 = ftall + 1*Nw;
71  zset_mem(2*Nw, ftall, 0.0+I*0.0); // both parts of complex
72
73  int T=0; // total steps
74  for(;;){
75      printf("Number of steps to run (<=0 to exit) -> "); fflush(stdout);

```

```

76     int steps;
77     scanf("%d", &steps);
78     if(steps<=0) break;
79
80     printf("Making %d steps\n", steps);
81     for(int n=0; n<steps; n++, T++){
82         update_Bz(Nx, Hz, Ey, xi);    // find Bz at n+1/2
83         update_Dy(Nx, Dy, Hz, xi);    // find Dy at n+1
84         update_Ey(Nx, Ey, Dy, eps);    // find Ey at n+1
85
86         /* output of Ey */
87         if((T+1)%No == 0){
88             printf("Elapsed time -> %g fs (%d steps)\n", dt*(T+1), T+1);
89             output_Ey_vs_x(Nx, Ey, T+1, dx, tag);
90         }
91
92         /** take running fourier ***/
93         double time=dt*(T+1); // for Ey
94         rfourier2(wmin, wmax, Nw, ft1, ft2, Ey[fi1], Ey[fi2], dt, time);
95
96     }// end of n loop
97
98     char fname1[100], fname2[100];
99     sprintf(fname1, "ft_fi=%d_%s.dat", fi1, tag);
100    sprintf(fname2, "ft_fi=%d_%s.dat", fi2, tag);
101    four_out(Nw, ft1, wmin, wmax, fname1);
102    four_out(Nw, ft2, wmin, wmax, fname2);
103
104 }// end of global loop
105
106 free(fields); free(eps); free(ftall);
107 }
108
109 /*****
110 int main(void){
111
112     slab();
113
114     return 0;
115 }
116
117 /*** END OF FILE *****/

```

В новой функции управления `slab()` мы задаём дополнительно границы слоя, его проницаемость, выбираем интервал частот и точки, где берётся преобразование Фурье. Также мы осуществляем вызов функций, отвечающих за материальные свойства в моделируемой области и за преобразование Фурье.

В файле `mater.c` мы поместим функции, задающие распределение про-

нищаемости во всей области и рассчитывающие поле  $E$ . Файл mater.c:

```
1 #include<stdio.h>
2
3 /*****
4 void output_eps_x(int Nx, const double *eps, double dx, const char *tag){
5
6     char fname[100];
7     sprintf(fname, "eps_vs_x_%s.dat", tag);
8     FILE *fp=fopen(fname, "w");
9     fprintf(fp, "# 1: x (micron) | 2 : eps \n");
10    for(int i=0; i<Nx; i++){
11        double x=i*dx;
12        fprintf(fp, "%.12e %.12e\n", 1e-3*x, eps[i]);
13    }
14    fclose(fp);
15    printf("Completed writing to file \"%s\"\n", fname);
16 }
17
18 /*****
19 void create_slab(int Nx, double *eps, int j1, int j2, double eslab){
20     /*
21     * Creates array of dielectric permittivity
22     */
23
24     for(int i=0; i<Nx; i++){
25         eps[i] = 1;
26     }
27     for(int i=j1; i<j2+1; i++){
28         eps[i] = eslab;
29     }
30
31 }
32
33 /*****
34 void update_Ey(int Nx, double *Ey, const double *Dy, const double *eps){
35     /*
36     * Calculates E from D and epsilon
37     */
38     for(int i=0; i<Nx; i++){
39         Ey[i] = Dy[i]/eps[i];
40     }
41 }
42
43 /*** END OF FILE *****/
```

Файл mater.h с описанием функций из mater.c:

```

1 void update_Ey(int Nx, double *Ey, const double *Dy, const double *ebkg);
2 void create_slab(int Nx, double *eps, int j1, int j2, double eslab);
3 void output_eps_x(int Nx, const double *eps, double dx, const char *tag);

```

Далее мы создаём функции для преобразования Фурье и вывода спектра в файл. Файл `rfourier.c`:

```

1 #include<stdio.h>
2 #include<math.h>
3 #include<complex.h>
4
5 typedef double complex dcomplex;
6 extern const double cspeed, pi;
7
8 /*****
9 void four_out(int Nw, const dcomplex *ft, double wmin, double wmax,
10               const char *fname){
11
12     FILE *fp=fopen(fname, "w");
13
14     printf("output to \"%s\"\n", fname);
15     double dw=(wmax-wmin)/(Nw-1.0);
16     for(int i=0; i<Nw; i++){
17         double w=wmin+dw*i;
18         fprintf(fp, "%.12e %.12e %.12e\n",
19               w, creal(ft[i]), cimag(ft[i]));
20     }
21     fclose(fp);
22 }
23
24 /*****
25 void rfourier2(double wmin, double wmax, int Nw, dcomplex *ft1,
26               dcomplex *ft2, double Ey1, double Ey2, double dt,
27               double time){
28
29     double dw = (wmax-wmin)/(Nw-1.0);
30     double mult=dt/(2*pi);
31     for(int n=0; n<Nw; n++){
32         double w=wmin+dw*n;
33         dcomplex ce=mult*cexp(I*w*time);
34         ft1[n] += ce*Ey1;
35         ft2[n] += ce*Ey2;
36     }
37 }
38 /*** END OF FILE *****/

```

Файл `rfourier.h` с описанием функций из `rfourier.c`:

```

1 void four_out(int Nw, const dcomplex *ft, double wmin, double wmax,
2               const char *fname);
3 void rfourier2(double wmin, double wmax, int Nw, dcomplex *ft1,
4               dcomplex *ft2, double Ey1, double Ey2, double dt,
5               double time);

```

Нам также понадобятся вспомогательные функции, которые осуществляют инициализацию массивов. Файл `set_mem.c`:

```

1 #include<complex.h>
2
3 typedef double complex dcomplex;
4
5 /*****
6 void zset_mem(int N, dcomplex *A, dcomplex var){
7     for(int i=0; i<N; i++){
8         A[i]=var;
9     }
10 }
11 *****/
12 void dset_mem(int N, double *A, double var){
13     for(int i=0; i<N; i++){
14         A[i]=var;
15     }
16 }
17 /*** END OF FILE *****/

```

Файл `set_mem.h` с описанием функций из `set_mem.c`:

```

1 void zset_mem(int N, dcomplex *A, dcomplex var);
2 void dset_mem(int N, double *A, double var);

```

После разработки всех новых функций мы дополним файл `makefile` возможностью создания выполняемого файла `slab.exe`. При этом компиляция будет также использовать файлы, созданные для моделирования импульса в свободном пространстве в разделе 4.

Файл `makefile`:

```

1 CC=gcc
2 GCCFLAGS= -O3 -Wall -std=c99
3
4 all : free_space slab
5

```



```

6 OBJSL = obj/slab.o    obj/fdtd_1d_maxwell.o  obj/mater.o \
7         obj/pulse.o  obj/output.o                obj/constants.o\
8         obj/set_mem.o obj/rfourier.o
9 slab:  $(OBJSL)
10        $(CC) $(GCCFLAGS) -o bin/slab.exe  $(OBJSL) -lm
11        @echo -e 'Created executable "bin/slab.exe"\n'
12
13 OBJFS = obj/free_space.o  obj/fdtd_1d_maxwell.o  obj/pulse.o \
14         obj/output.o      obj/constants.o
15 free_space:  $(OBJFS)
16        $(CC) $(GCCFLAGS) -o bin/free_space.exe  $(OBJFS) -lm
17        @echo -e 'Created executable "bin/free_space.exe"\n'
18
19 ### creation of object files
20 obj/%.o : src/%.c
21        $(CC) -c $(GCCFLAGS) $< -o $@
22
23 clean:
24        rm -f obj/*.o bin/*.exe

```

Приведённый выше `makefile` позволяет создавать как выполняемый файл для моделирования распространения импульса в свободном пространстве (при выполнении команды `make free_space`), так и выполняемый файл для моделирования прохождения через слой (`make slab`). Можно также создать оба файла одновременно при выполнении команды `make` без аргументов.

## 8.2 Результаты моделирования и их обсуждение

Следует обратить внимание, что одна из точек, где берётся преобразование Фурье, находится слева от начального импульса. Это позволяет наблюдать в этой точке только отражённый от слоя сигнал. Другая точка берётся справа от слоя, где распространяется прошедший сигнал. Для расчёта коэффициентов отражения и прохождения требуется также и спектр начального сигнала. Его проще всего получить, если выполнить программу еще раз и либо положить толщину слоя равной нулю, либо его проницаемость – единице. Полученные файлы с данными позволяют построить требуемые коэффициенты. Типичные спектры прошедшего и падающего импульсов показаны на рис. 13. Видно, что прошедший спектр сильно изменён по сравнению с падающим.

Интересно проверить, насколько правилен расчёт. Для этого можно получить аналитические формулы для коэффициентов. Они имеют следующий вид:

$$r = \frac{(n^2 - 1)(e^{2i\varphi} - 1)}{(n + 1)^2 - (n - 1)^2 e^{2i\varphi}}, \quad t = \frac{4ne^{i\varphi}}{(n + 1)^2 - (n - 1)^2 e^{2i\varphi}}, \quad (26)$$

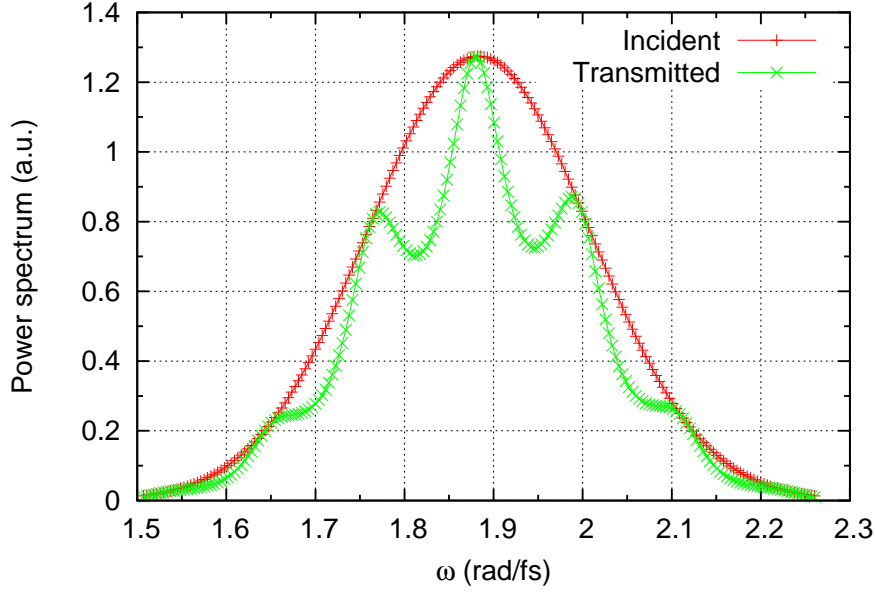


Рис. 13: Спектры  $|\tilde{E}(\omega)|^2$  падающего (incident) и прошедшего (transmitted) импульсов при  $\lambda_0 = 1000$  нм,  $\tau = 8$  фс,  $\varepsilon = 4$ ,  $L = 4000$  нм.

где  $\varphi = nkL$ ,  $n = \sqrt{\varepsilon}$  – индекс преломления для слоя,  $L$  – его толщина,  $k = \omega/c$  – волновое число в вакууме.

Рисунок 14 показывает коэффициенты отражения и прохождения, рассчитанные численно, и те же самые коэффициенты, построенные на основании формул (26). Видно, что результаты хорошо совпадают.

## 9 Моделирование сред с поглощением

В предыдущих разделах мы изучили распространение импульсов в свободном пространстве и в диэлектрике без дисперсии и поглощения. На практике же большинство сред имеет дисперсию и поглощение, а их моделирование требует дополнительной работы. Более того, использование временного подхода означает, что моделирование проводится в некоторой области частот, и учёт изменения свойств среды в этом диапазоне может быть просто необходим для получения правильного результата. Следует также отметить, что аналитическое решение электродинамических задач в таких средах становится, как правило, очень сложным, а поэтому разработка численных методов представляет особый практический интерес.

Численные подходы к моделированию сред с дисперсией и поглощением зависят от теоретической модели, используемой для описания частотной зависимости. При расчёте распространения импульса в диэлектрике в разделе 6 мы в некоторых моменты  $n\Delta t$  находили поле  $E$  из  $D$ . А поле  $D$  получалось из уравнения Максвелла. Давайте теперь обобщим этот подход на случай, когда

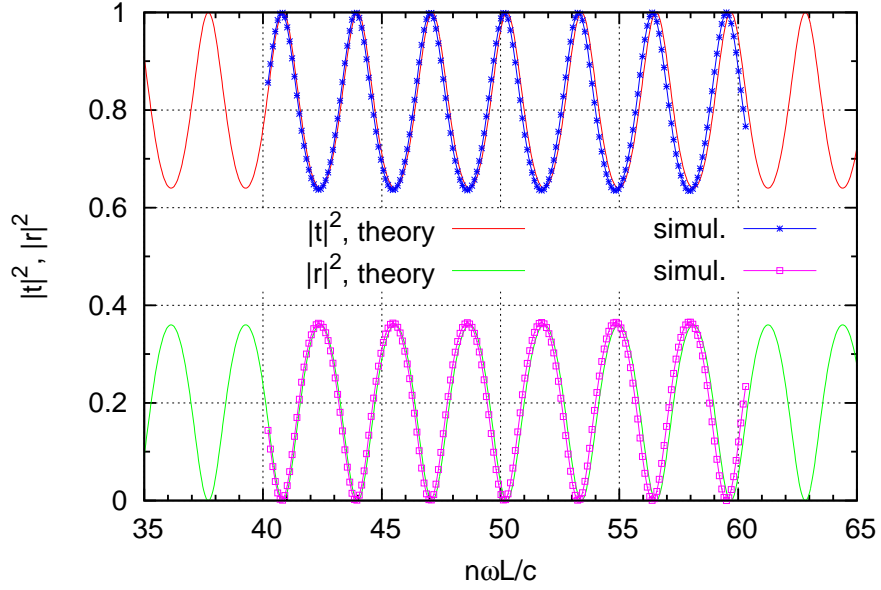


Рис. 14: Коэффициенты прохождения и отражения для слоя диэлектрика, рассчитанные численно (simul.) и аналитически (theory).

в диэлектрической среде с некоторым действительным  $\varepsilon$  имеется поглощение, которое описывается проводимостью  $\sigma$ :

$$J = \frac{\partial P}{\partial t} = \sigma E. \quad (27)$$

Будем далее считать, что ток, определяемый уравнением (27), входит в поле  $D$  и, таким образом, уравнение Максвелла

$$\frac{\partial D}{\partial t} = -c \frac{\partial H}{\partial x} \quad (28)$$

остается таким же, как и без поглощения. Воспользуемся связью полей

$$D = \varepsilon E + 4\pi P \quad (29)$$

и, продифференцировав по времени это уравнение, мы получим

$$\frac{\partial D}{\partial t} = \varepsilon \frac{\partial E}{\partial t} + 4\pi \sigma E. \quad (30)$$

Используя конечные разности, мы перепишем (30) для каждой ячейки  $i$  в виде

$$D_i^{n+1} - D_i^n = \varepsilon (E_i^{n+1} - E_i^n) + 2\pi\sigma\Delta t (E_i^{n+1} + E_i^n). \quad (31)$$

Обозначая

$$\eta = 2\pi\sigma\Delta t/\varepsilon, \quad (32)$$

мы получим следующую формулу для нахождения поля  $E_i^{n+1}$ :

$$E_i^{n+1} = \frac{1 - \eta}{1 + \eta} E_i^n + \frac{D_i^{n+1} - D_i^n}{\varepsilon(1 + \eta)} \quad (33)$$

Таким образом, для нахождения поля  $E_i^{n+1}$  при наличии проводимости  $\sigma$  необходимо знать не только  $D_i^{n+1}$ , но и  $D_i^n$ ,  $E_i^n$ , то есть значения полей в предыдущие моменты времени.

Можно поступить и по-другому. Интегрирование уравнения (29) даёт

$$D(t) = \varepsilon E(t) + 4\pi\sigma \int_{-\infty}^t E(t') dt'. \quad (34)$$

Используя метод трапеций для оценки интеграла в (34), мы получаем следующее соотношение для полей в момент времени  $(n + 1)\Delta t$ :

$$D_i^{n+1} = \varepsilon E_i^{n+1} + 2\varepsilon\eta \left( \sum_{m=0}^n E_i^m + \frac{1}{2} E_i^{n+1} \right). \quad (35)$$

Используя (35), поле  $E_i^{n+1}$  можно найти как

$$E_i^{n+1} = \frac{D_i^{n+1}}{\varepsilon(1 + \eta)} - \frac{2\eta S_i^n}{1 + \eta}, \quad \text{где} \quad S_i^n = \sum_{m=0}^n E_i^m. \quad (36)$$

То есть наличие поглощения привело к необходимости ввести величину  $S_i^n$  в уравнение (36), но нет необходимости теперь сохранять  $D_i^n$  и  $E_i^n$ .

Таким образом, мы получили два разных алгоритма, (33) и (36), для расчёта  $E_i^{n+1}$  для сред с поглощением. Оба они одинаково применимы и совместимы со схемой (22). Следует отметить, что полученные формулы работают только в рамках модели поглощения, задаваемой формулой (27). При использовании других моделей конкретные формулы алгоритма будут отличаться от (33) и (36), но могут быть получены методами, изложенными в этом разделе.

## 10 Безотражательные (поглощающие) граничные условия

Для нахождения коэффициентов прохождения и отражения при падении импульса на слой в разделе 8 нам приходилось брать довольно большую область моделирования, чтобы избежать попадания сигнала, отраженного от

границы области, в точки, где бралось преобразование Фурье. Было бы проще, если можно было бы избежать такого отражения. Мы рассмотрим здесь два способа достичь этого.

Сначала разберёмся, почему возникает отражение. Рассмотрим правую границу области моделирования на рис. 5. При нахождении поля  $B^{n+1/2}$  в крайней точке, используя уравнение (9а), мы не знаем поле  $E$  справа от нее. Поэтому мы использовали условие  $E = 0$ , как на идеальной металлической границе, что и приводило к отражению импульса. В свободном пространстве импульс распространяется со скоростью света вдоль оси  $x$ . Таким образом, если взять  $\xi = 1$ , то  $E^n$  справа от последней точки задания  $B$  будет равно значению поля  $E$  слева, но в предыдущий момент времени, то есть  $E^{n-1}$ . Это можно и использовать как граничное условие для того, чтобы избежать отражения. Таким образом, при  $\xi = 1$  уравнение для крайней правой точки становится

$$B_i^{n+1/2} = B_i^{n-1/2} - \xi (E_i^{n-1} - E_i^n). \quad (37)$$

Если выбрать  $\xi = 1/2$ , то надо брать значение  $E^{n-2}$  вместо  $E^{n-1}$ . Такое граничное условие работает хорошо в 1D случае. В 2D и 3D такой подход не работает, так как волны не будут плоскими и нельзя будет так просто экстраполировать их значения.

Теперь мы изучим подход, который можно обобщить на 2D и 3D задачи. Он основан на использовании около границ некоторого слоя, от границы которого нет отражения и внутри которого прошедшая волна быстро поглощается. Рассмотрим сначала падение плоской волны из пространства с диэлектрической проницаемостью  $\varepsilon_1$  и магнитной проницаемостью  $\mu_1$  на границу материала с  $\varepsilon_2$  и  $\mu_2$ . Будем считать, что этот материал обладает поглощением, а поэтому величины  $\varepsilon_2$  и  $\mu_2$  будут комплексными при нахождении решения в виде  $e^{-i\omega t}$ . Комплексный коэффициент отражения для  $E$  имеет следующий вид:

$$r(\omega) = \frac{\sqrt{\varepsilon_1/\mu_1} - \sqrt{\varepsilon_2/\mu_2}}{\sqrt{\varepsilon_1/\mu_1} + \sqrt{\varepsilon_2/\mu_2}}. \quad (38)$$

Видно, что при соблюдении условия согласования

$$\frac{\varepsilon_2}{\mu_2} = \frac{\varepsilon_1}{\mu_1} \quad (39)$$

отражения не будет. Теперь мы может выбрать параметры  $\varepsilon_2$  и  $\mu_2$  так, чтобы как удовлетворить (39), так и обеспечить поглощение:

$$\varepsilon_2 = \varepsilon_1 s, \quad \mu_2 = \mu_1 s, \quad \text{где} \quad s = 1 + i \frac{4\pi \bar{\sigma}}{\omega}, \quad \bar{\sigma} = \frac{\sigma^e}{\varepsilon_1} = \frac{\sigma^m}{\mu_1}. \quad (40)$$

Волновое число  $k_2$  прошедшей волны будет

$$\operatorname{Re} k_2 = k_1, \quad \operatorname{Im} k_2 = k_1 \frac{4\pi\bar{\sigma}}{\omega}. \quad (41)$$

Волна, падающая на такой материал, не будет отражаться из-за полного согласования и будет постепенно поглощаться. Слой достаточной толщины из такого материала, помещенный на правый и левый концы области моделирования, может полностью поглотить уходящие волны. Граничные условия в таком случае не будут играть никакой роли, так как волны до границ области моделирования доходить не будут. Особенностью такого искусственного материала является то, что поглощение связано не только с мнимой частью в  $\varepsilon_2$ , но и с мнимой частью в  $\mu_2$ . Такое магнитное поглощение  $\sigma^m$  можно ввести так же, как и электрическое  $\sigma^e$ , рассмотренное в разделе 9. Наличие магнитного поглощения обеспечивает и условие согласования, и добавочное поглощение. Слой такого материала обычно называют идеально согласованным (perfectly matched layer или PML). Конечно, можно брать и слой только с электрическим поглощением. Но тогда низкое отражение будет только при адиабатически слабом росте проводимости  $\sigma^e$ . Это потребует использование очень толстых, много больше длины волны, слоёв.

Хотя теоретически идеально согласованная граница должна обеспечивать полное прохождение волны через нее, на практике появляется слабое отражение, связанное с дискретизацией. Поэтому, как правило, параметр поглощения  $\eta(x) = 2\pi\bar{\sigma}\Delta t$ , определённый формулой (32), выбирают таким образом, чтобы его величина плавно нарастала от начала слоя, как показано на рис. 15. Удобно использовать аналитическую форму

$$\eta(x) = \eta_0 \left(\frac{x}{L}\right)^\alpha, \quad (42)$$

в которой параметры  $\eta_0$ ,  $\alpha$  и  $L$  подбирают с целью уменьшения отражения. Как правило, эти параметры можно брать в пределах

$$\eta_0 \sim 0.2 \div 0.5, \quad \alpha \sim 2 \div 3, \quad L \sim (10 \div 20)\Delta x \sim (0.5 \div 1)\lambda_0. \quad (43)$$

Как видим, достаточно выбрать толщину слоя порядка и даже меньше длины волны. Заметим, что так как электрическое и магнитное поля заданы в разных точках, то надо выбирать значения  $\eta(x)$  с учётом этого.

## 11 Моделирование сред с дисперсией

Как мы видели в разделе 6, в средах без дисперсии нахождение поля  $E$  из  $D$  не вызывает сложностей. Если же есть дисперсия, то, как правило, простое

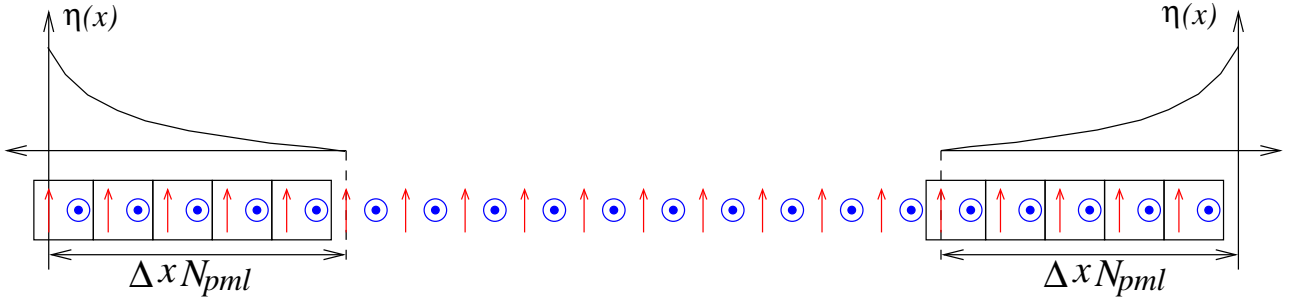


Рис. 15: Применение идеально согласованного слоя (PML) с плавно меняющимся поглощением на концах области моделирования.

соотношение между  $E$  и  $D$  существует только в частотной области:

$$\tilde{D}(\omega) = \varepsilon(\omega)\tilde{E}(\omega). \quad (44)$$

Это уравнение можно перевести во временную область и полученный интеграл использовать для нахождения  $E^{n+1}$ . Это наиболее общий подход.

На практике, однако, часто встречаются ситуации, в которых можно довольно просто учесть дисперсионные свойства. Мы рассмотрим типичный случай среды, где имеются свободные носители, то есть плазму. Такая среда характеризуется частотно-зависимой диэлектрической проницаемостью

$$\varepsilon(\omega) = \varepsilon_b \left( 1 - \frac{\omega_p^2}{\omega^2} \right), \quad \omega_p = \sqrt{\frac{4\pi e^2 N}{\varepsilon_b m_e}}, \quad (45)$$

где  $\varepsilon_b$  – диэлектрическая проницаемость материала без носителей,  $\omega_p$  – плазменная частота. Но мы не будем использовать это соотношение. Вместо этого мы возьмём уравнение для динамики свободных носителей

$$m_e \frac{\partial V}{\partial t} = eE. \quad (46)$$

Учитывая, что плотность тока  $J_{fc} = eNV$ , мы получаем

$$\frac{\partial J_{fc}}{\partial t} = \frac{\varepsilon_b \omega_p^2}{4\pi} E. \quad (47)$$

Удобно задавать  $J$  в тех же точках, что и  $E$ . При этом брать  $J$  в моменты времени, такие же как  $H$ . Тогда мы получаем

$$J_i^{n+1/2} = J_i^{n-1/2} + \frac{\varepsilon_b \omega_p^2}{4\pi} \Delta t E_i^n \quad (48)$$

и

$$D_i^{n+1} = D_i^n - \xi \left( H_i^{n+1/2} - H_{i-1}^{n+1/2} \right) - 4\pi \Delta t J_i^{n+1/2}. \quad (49)$$

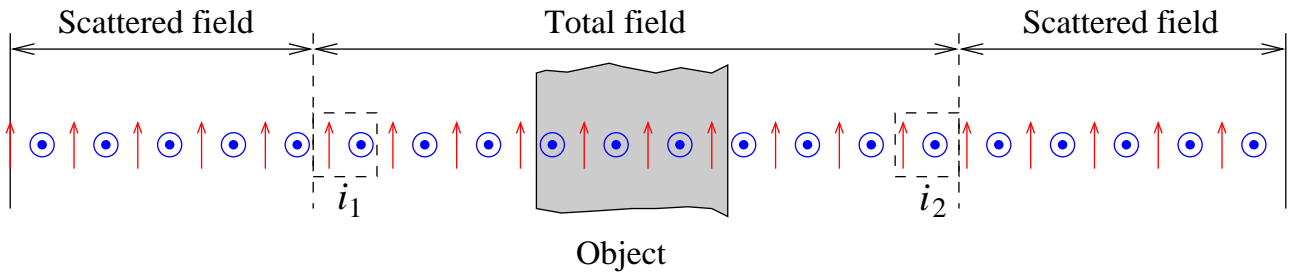


Рис. 16: Разбиение области моделирования на область полного (total) и рассеянного (scattered) полей.

На практике бывает удобно избежать различных постоянных множителей и ввести, например,  $\bar{J} = 4\pi\Delta tJ$ .

Таким образом, для учёта дисперсии, вызываемой свободными носителями, в основной алгоритм (22) добавляется уравнение для расчёта тока (48), а в уравнение Максвелла (49) – дополнительный член. При этом структура алгоритма и все разработанные до этого функции можно использовать практически без изменений.

## 12 Разделение на области полного и рассеянного полей

В разделах 4 и 8 моделирование распространения требовало задания начальных полей  $E(x, t = 0)$  и  $H(x, t = -\Delta t/2)$ . Для этого надо, чтобы область моделирования полностью вмещала начальный импульс. На практике это может приводить к необходимости увеличивать размер области, что особенно критично при обобщении подхода на 2D и 3D случаи. Чтобы избежать этого, можно воспользоваться подходом, который описывается в этом разделе.

Представим, что у нас имеется импульс, который падает на некоторый объект. Тогда полное поле всегда можно представить как сумму падающего поля и рассеянного поля, возбуждаемого наведённой поляризацией в объекте. До сих пор мы всегда рассчитывали полное поле, то есть не разделяли его на падающее и рассеянное.

Мы теперь можем разделить область моделирования на две области, см. рис. 16. В первой области, области полного поля, в которую входит и рассеивающий объект, мы будем рассчитывать полное поле, как и раньше. Во второй области, области рассеянного поля, мы будем учитывать только рассеянное поле, то есть без вклада падающего поля. В 1D случае область рассеянного поля состоит из двух несвязанных частей – справа и слева от области полного поля.

Возникает вопрос, как нам правильно стыковать поля на границе. Да-



вайте предположим, что границы области полного поля заданы ячейками  $i_1$  и  $i_2$ , см. рис. 16. Тогда нам надо изменить формулы пересчёта полей по обе стороны от границы, то есть  $B_{i_1-1}^{n+1/2}$ ,  $D_{i_1}^{n+1}$ ,  $B_{i_2}^{n+1/2}$ ,  $D_{i_2+1}^{n+1}$ . Предположим, что падающее поле задано функциями  $E_0(x, t)$  и  $H_0(x, t)$ . Тогда на левой границе мы получаем

$$B_{i_1-1}^{n+1/2} = B_{i_1-1}^{n-1/2} - \xi [(E_{i_1}^n - E_0(i_1\Delta x, n\Delta t)) - E_{i_1-1}^n], \quad (50a)$$

$$D_{i_1}^{n+1} = D_{i_1}^n - \xi \left[ H_{i_1}^{n+1/2} - \left( H_{i_1-1}^{n+1/2} + H_0((i_1 - 1/2)\Delta x, (n + 1/2)\Delta t) \right) \right] \quad (50b)$$

или

$$B_{i_1-1}^{n+1/2} = B_{i_1-1}^{n-1/2} - \xi [E_{i_1}^n - E_{i_1-1}^n] + \xi E_0(i_1\Delta x, n\Delta t), \quad (51a)$$

$$D_{i_1}^{n+1} = D_{i_1}^n - \xi [H_{i_1}^{n+1/2} - H_{i_1-1}^{n+1/2}] + \xi H_0((i_1 - 1/2)\Delta x, (n + 1/2)\Delta t). \quad (51b)$$

Мы видим из (51), что после вычисления полей по формулам без учёта существования границы между областями к полям по обе стороны от неё надо добавить члены, зависящие только от падающего поля. На правой границе мы получаем похожие формулы:

$$B_{i_2}^{n+1/2} = B_{i_2}^{n-1/2} - \xi [(E_{i_2+1}^n + E_0((i_2 + 1)\Delta x, n\Delta t)) - E_{i_2}^n], \quad (52a)$$

$$D_{i_2+1}^{n+1} = D_{i_2+1}^n - \xi \left[ H_{i_2+1}^{n+1/2} - \left( H_{i_2}^{n+1/2} - H_0((i_2 + 1/2)\Delta x, (n + 1/2)\Delta t) \right) \right] \quad (52b)$$

или

$$B_{i_2}^{n+1/2} = B_{i_2}^{n-1/2} - \xi [E_{i_2+1}^n - E_{i_2}^n] - \xi E_0((i_2 + 1)\Delta x, n\Delta t), \quad (53a)$$

$$D_{i_2+1}^{n+1} = D_{i_2+1}^n - \xi [H_{i_2+1}^{n+1/2} - H_{i_2}^{n+1/2}] - \xi H_0((i_2 + 1/2)\Delta x, (n + 1/2)\Delta t). \quad (53b)$$

При разделении областей следует иметь в виду, что в начальный момент времени падающий импульс должен находиться полностью за пределами области полного поля. В противном случае надо также задавать поле начального импульса в области полного поля, чего удобнее избежать. Отличительной особенностью 1D случая по сравнению с 2D и 3D является то, что область рассеянного поля разделена на две несвязанные части. Бывает удобно в 1D использовать разделение на области полное/рассеянное поле слева, где находится падающий импульс, а справа такое разделение не использовать. При этом прошедшее поле будет полным.

# 13 Пример 3: Падение импульса на слой плазмы

## 13.1 Разработка программы моделирования

Рассмотрение падения импульса на слой плазмы позволяет нам применить методики моделирования, изученные в разделах 9, 10, 11, 12: разделение на области полного и рассеянного полей, создание поглощающих слоев на границе области моделирования и учёт временной дисперсии. Ниже приведены разработанные функции моделирования.

Файл `slab2.c` с новой функцией управления `slab2()`:

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 #include<math.h>
5 #include<complex.h>
6
7 typedef double complex dcomplex;
8 extern const double cspeed, pi;
9
10 #include"fddt_1d_maxwell.h"
11 #include"pulse.h"
12 #include"output.h"
13 #include"mater.h"
14 #include"pml.h"
15 #include"mater_fc.h"
16 #include"rfourier.h"
17 #include"set_mem.h"
18
19 /*****
20 void slab2(void){
21
22     char *tag="v1";    // used to label output files
23
24     /** Plasma slab ***/
25     double wp = 3.0; // rad/fs
26     int Nslab = 30; // number of cells
27
28     /** Optical pulse ***/
29     double lambda0 = 2*pi*cspeed/wp; // nm
30     double tau = 5; // fs, width of the pulse
31
32     /** Computational parameters ***/
33     double dx = 20.0; // nm
34     int Nx = 2000; // number of cells along x
```

```

35 | int    ix0 = 0;        // center of the pulse at t=0
36 | int    i1 = 500;      // scattered/total boundary
37 |
38 | int    si1 = 1000;    // start of the slab
39 | int    si2 = si1+Nslab-1; // end of the slab
40 |
41 | int    fi1 = si2+5;   // location of fourier transform
42 | int    fi2 = i1-5;   // location of fourier transform
43 |
44 | double xi = 0.9;
45 | int    No = 200;     // defines the output rate
46 |
47 | /*** start execution ***/
48 | double w0 = 2*pi*cspeed/lambda0; // rad/fs
49 | double dt = xi*dx/cspeed; // in fs
50 | double twp2 = wp*wp*dt*dt;
51 | printf("dx=%.12e nm, dt=%.12e fs\n", dx, dt);
52 | printf("w_p*L/speed=%g\n", wp*dx*Nslab/cspeed);
53 |
54 | /*** Implement pml ***/
55 | int    Npml = 20;    // number of PML cell
56 | double eta0 = 0.5;  // absorption parameter
57 | double *pml = malloc(4*Npml*sizeof(double)); // to keep sums of fields
58 | double *pmlSE1 = pml + 0*Npml, *pmlSE2 = pml + 1*Npml;
59 | double *pmlSH1 = pml + 2*Npml, *pmlSH2 = pml + 3*Npml;
60 | dset_mem(4*Npml, pml, 0.0); // initialize PML data
61 |
62 | /*** arrays for the fields ***/
63 | double *fields = malloc(5*Nx*sizeof(double));
64 | double *Hz = fields+0*Nx;
65 | double *Ey = fields+1*Nx;
66 | double *Dy = fields+2*Nx;
67 | double *Bz = fields+3*Nx;
68 | double *nJ = fields+4*Nx;
69 | dset_mem(5*Nx, fields, 0.0); // initialize fields
70 |
71 | double *eps = malloc(Nx*sizeof(double));
72 | dset_mem(Nx, eps, 1.0); // sets permittivity
73 |
74 | double wmin = 0.7*w0; // rad/fs
75 | double wmax = 1.3*w0; // rad/fs
76 | int    Nw = 500;
77 |
78 | dcomplex *ftall=malloc(2*Nw*sizeof(dcomplex));
79 | dcomplex *ft1 = ftall + 0*Nw;
80 | dcomplex *ft2 = ftall + 1*Nw;
81 | zset_mem(2*Nw, ftall, 0.0+I*0.0); // initialize Fourier data
82 |
83 | int    T=0; // total steps

```

```

84 for(;;){
85     printf("Number of steps to run (<=0 to exit) -> ");  fflush(stdout);
86     int steps;
87     scanf("%d", &steps);
88     if(steps<=0) break;
89
90     printf("Making %d steps\n", steps);
91     for(int n=0; n<steps; n++, T++){
92         update_Bz(Nx, Bz, Ey, xi);                // find Bz at n+1/2
93         Bz[i1-1] += xi*pulse((i1-ix0)*dx,T*dt,cspeed,tau,w0); // t/s
94         memcpy(Hz, Bz, Nx*sizeof(double));        // find Hz at n+1/2
95         update_H_pml(Nx, Hz, Bz, pmlSH1, pmlSH2, Npml, eta0); // Hz in PML
96         update_nJ(Nslab, nJ+si1, twp2, Ey+si1);   // find nJ at n+1/2
97         update_Dy(Nx, Dy, Hz, xi);                // find Dy at n+1
98         Dy[i1] += xi*pulse((i1-0.5-ix0)*dx,(T+0.5)*dt,cspeed,tau,w0); // t/s
99         add_J_to_D(Nslab, Dy+si1, nJ+si1);        // free carrier current
100        update_Ey(Nx, Ey, Dy, eps);                // find Ey from Dy at n+1
101        update_E_pml(Nx, Ey, Dy, pmlSE1, pmlSE2, Npml, eta0);
102
103        /* output of Ey */
104        if((T+1)%No == 0){
105            printf("Elapsed time -> %g fs (%d steps)\n", dt*(T+1), T+1);
106            output_Ey_vs_x(Nx, Ey, T+1, dx, tag);
107        }
108
109        /** take running fourier ***/
110        double time=dt*(T+1); // for Ey
111        rfourier2(wmin, wmax, Nw, ft1, ft2, Ey[fi1], Ey[fi2], dt, time);
112
113    }// end of n loop
114
115    char fname1[100], fname2[100];
116    sprintf(fname1, "ft_fi=%d_%s.dat", fi1, tag);
117    sprintf(fname2, "ft_fi=%d_%s.dat", fi2, tag);
118    four_out(Nw, ft1, wmin, wmax, fname1);
119    four_out(Nw, ft2, wmin, wmax, fname2);
120
121 }// end of global loop
122
123 free(fields);  free(eps);  free(pml);
124 }
125
126 /*****
127 int main(void){
128
129     slab2();
130
131     return 0;
132 }

```

133

134 /\*\*\* END OF FILE \*\*\*\*\*/

Файл pml.c с функциями, отвечающими за поглощающие слои:

```

1  /*****/
2  static void bupdate(const double *D, double *E, double *SE,
3                      double shift, int i, int j, int Npml, double eta0){
4      /*
5       * j - index for the field arrays E, D; i - index for the SE array
6       */
7
8      double tmp = (i+shift)/Npml;
9      double eta = tmp*tmp*eta0;
10     double bot = 1.0+eta;
11
12     E[j]  = (D[j] - 2*eta*SE[i])/bot;
13     SE[i] += E[j];
14 }
15
16 /*****/
17 void update_H_pml(int Nx, double *H, const double *B, double *SH1,
18                  double *SH2, int Npml, double eta0){
19     /*
20      * Updates H inside pml
21      */
22
23     for(int i=0; i<Npml; i++){
24         bupdate(B, H, SH1, 0.5, i, Npml-1-i, Npml, eta0); // left
25         bupdate(B, H, SH2, 0.5, i, Nx-Npml+i, Npml, eta0); // right
26     }
27 }
28
29 /*****/
30 void update_E_pml(int Nx, double *E, const double *D, double *SE1,
31                  double *SE2, int Npml, double eta0){
32     /*
33      * Updates E inside pml
34      */
35
36     for(int i=0; i<Npml; i++){
37         bupdate(D, E, SE1, 1.0, i, Npml-1-i, Npml, eta0); // left
38         bupdate(D, E, SE2, 0.0, i, Nx-Npml+i, Npml, eta0); // right
39     }
40 }
41
42 /*** END OF FILE *****/

```

Файл `pml.h` с описанием функций из `pml.c`:

```
1 void update_H_pml(int Nx, double *H, const double *B, double *SH1,
2                   double *SH2, int Npml, double eta0);
3 void update_E_pml(int Nx, double *E, const double *D, double *SE1,
4                   double *SE2, int Npml, double eta0);
```

Файл `mater_fc.c` с функциями, позволяющими моделировать наличие свободных носителей, то есть дисперсии:

```
1 /*
2  * Contains functions to model free carriers
3  */
4
5 /*****
6 void add_J_to_D(int Nx, double *D, double *nJ){
7     for(int i=0; i<Nx; i++)
8         D[i] -= nJ[i];
9 }
10
11 /*****
12 void update_nJ(int Nx, double *nJ, double twp2, double *E){
13     for(int i=0; i<Nx; i++)
14         nJ[i] += twp2*E[i];
15 }
16 /*** END OF FILE *****/
```

Файл `mater_fc.h` с описанием функций из `mater_fc.c`:

```
1 void add_J_to_D(int Nx, double *D, double *nJ);
2 void update_nJ(int Nx, double *nJ, double twp2, double *E);
```

Для создания выполняемого файла мы должны добавить соответствующие команды в `makefile`. Это делается так же, как и в предыдущих примерах.

## 13.2 Результаты моделирования и их обсуждение

Рисунок 17 показывает распределение поля  $E_y(x)$  в разные моменты времени. Сначала ( $T = 600$  на рис. 17) импульс входит постепенно в область полного поля. При этом значение  $E_y$  меняется скачком на границе областей полного и рассеянного полей. Рассеянное поле еще не образовалось в этот момент. Затем ( $T = 1200$  на рис. 17) импульс начинает взаимодействовать со

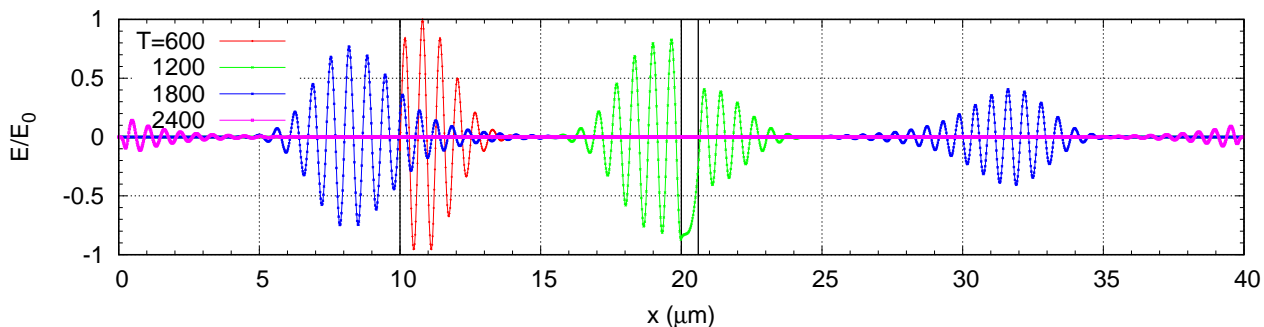


Рис. 17: Распределение  $E_y(x)$  в разные моменты времени. Черные сплошные линии показывают границы слоя и границу между областями полного и рассеянного полей.

слоем и формируются прошедшее и отраженное поля ( $T = 1800$  на рис. 17). В этот момент нет скачка поля слева от слоя на границе областей полного и рассеянного полей, так как существует только рассеянное поле. Достигнув концов области моделирования ( $T = 2400$  на рис. 17), прошедшее и отраженное поля поглощаются. На самом деле присутствует очень слабое отражение от идеально согласованного слоя. Коэффициент отражения  $|r| \sim 10^{-5}$ , что пренебрежимо мало в данном случае. Такие маленькие значения свидетельствуют о правильном выборе алгоритма и параметров поглощения.

Рассмотрим теперь полученные коэффициенты отражения и прохождения, см. рис. 18. При расчёте отраженного сигнала точка для преобразования Фурье берётся в области рассеянного поля, что позволяет исключить вклад падающего импульса. Прошедшее поле не разделяется на падающее и рассеянное. Совпадение численных результатов с аналитическими на рис. 18 свидетельствует о правильности расчета. При низких частотах,  $\omega/\omega_p < 1$ , проницаемость  $\varepsilon < 0$  и поля спадают в слое. Прошедший сигнал получается в результате туннелирования падающего поля. При  $\omega/\omega_p > 1$  проницаемость становится  $\varepsilon > 0$  и волны в слое могут распространяться. При этом начинают проявляться интерференционные эффекты, которые затем исчезают при  $\omega/\omega_p \gg 1$ , когда слой становится прозрачным из-за  $\varepsilon \approx 1$ .

## 14 Двумерное моделирование

Изучив принципы моделирования в одномерном (1D) случае, мы теперь можем приступить к двумерному (2D). Двумерное моделирование можно разделить на два независимых случая: ТЕ (transverse electric) и ТМ (transverse magnetic). Мы рассмотрим случай ТЕ поляризации, в котором имеются поля  $E_z, H_x, H_y$ . Все поля не зависят от координаты  $z$ . Мы тогда получаем три

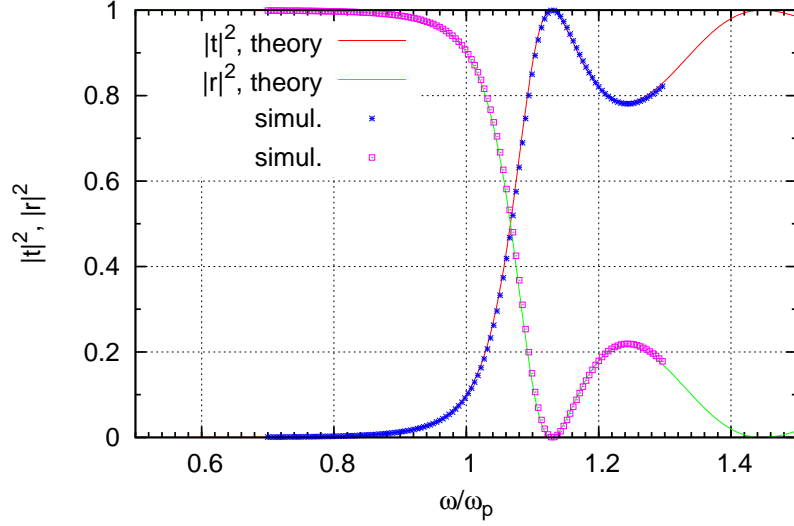


Рис. 18: Коэффициенты прохождения и отражения для слоя плазмы, рассчитанные численно (simul.) и аналитически (theory) при  $\omega_p = 3$  рад/фс,  $L = 600$  нм.

уравнения для компонент полей:

$$\frac{\partial B_x}{\partial t} = -c \frac{\partial E_z}{\partial y}, \quad (54a)$$

$$\frac{\partial B_y}{\partial t} = c \frac{\partial E_z}{\partial x}, \quad (54b)$$

$$\frac{\partial D_z}{\partial t} = c \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) - 4\pi J_z. \quad (54c)$$

Следующий шаг заключается в дискретизации этих дифференциальных уравнений. Для использования центральных разностей можно выбрать сетку, показанную на рис. 19. В каждой ячейке содержится по одной компоненте поля. Компоненты смещены в пространстве по отношению друг к другу. Также они будут смещены по времени. Будем обозначать индексы ячеек как  $[i, j]$ . Тогда получаем следующие дискретные уравнения (без учета тока  $J_z$ ):

$$B_x^{n+1/2}[i, j] = B_x^{n-1/2}[i, j] - \xi (E_z^n[i, j+1] - E_z^n[i, j]), \quad (55a)$$

$$B_y^{n+1/2}[i, j] = B_y^{n-1/2}[i, j] + \xi (E_z^n[i+1, j] - E_z^n[i, j]), \quad (55b)$$

$$D_z^{n+1}[i, j] = D_z^n[i, j] + \xi \left( H_y^{n+1/2}[i, j] - H_y^{n+1/2}[i-1, j] \right) - \xi \left( H_x^{n+1/2}[i, j] - H_x^{n+1/2}[i, j-1] \right). \quad (55c)$$

Порядок пересчёта полей будет таким же, как и в одномерном случае:

$$\dots \rightarrow B_{x,y}^{n+1/2} \rightarrow H_{x,y}^{n+1/2} \rightarrow D_z^{n+1} \rightarrow E_z^{n+1} \rightarrow B_{x,y}^{n+3/2} \rightarrow \dots \quad (56)$$



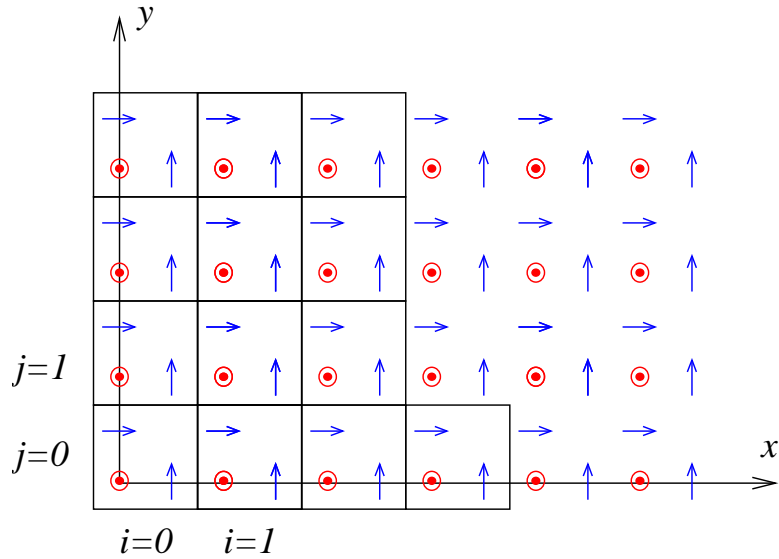


Рис. 19: Разбиение двумерного пространства  $x, y$  на ячейки и расположение точек (узлов), в которых заданы поля  $E_z, H_x, H_y$ .

Если среда немагнитная, то  $B_{x,y} = H_{x,y}$ .

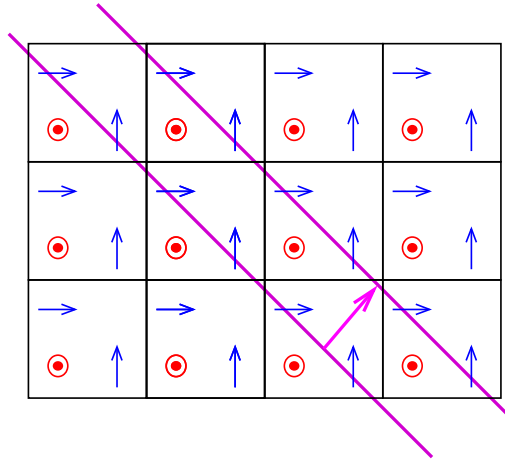


Рис. 20: Распространение сигнала по 2D сетке.

Давайте теперь обсудим сходимость приведенного 2D алгоритма (56). Для 1D случая мы получили, что алгоритм для диэлектрической среды без дисперсии устойчив при

$$\frac{\xi}{\sqrt{\varepsilon}} = v_{ph} \frac{\Delta t}{\Delta x} < 1. \quad (57)$$

В 2D случае волны могут распространяться под произвольными углами на сетке и это меняет критерий устойчивости (57). Если волна бежит под углом в  $45^\circ$ , см. рис. 20, то тогда за один шаг информация может распространиться

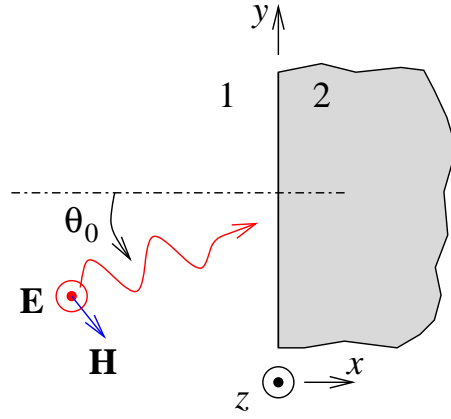


Рис. 21: Падение плоской ТЕ волны на границу раздела двух сред.

на  $\Delta x/\sqrt{2}$ . Тогда критерий устойчивости становится

$$\frac{\Delta x}{\sqrt{2}\Delta t} > v_{ph} \quad \text{или} \quad \frac{\xi}{\sqrt{\varepsilon}} < \frac{1}{\sqrt{2}} \approx 0.71. \quad (58)$$

Если волна бежит под другими углами, то это ослабляет критерий (58). Таким образом, временной шаг  $\Delta t$  в 2D случае должен быть меньше, чем в 1D.

## 15 Идеально согласованный слой в 2D

Давайте теперь рассмотрим, как можно моделировать идеально согласованный слой, который обеспечивает отсутствие отражения волн на границе области моделирования в 2D случае. Для начала рассмотрим задачу о падении плоской волны на границу между двумя средами с  $\varepsilon_1, \mu_1$  и  $\varepsilon_2, \mu_2$ , см. рис. 21. Коэффициент отражения для электрического поля плоской волны падающей под углом  $\theta_0$  будет

$$r = \frac{\sqrt{\varepsilon_1/\mu_1} \cos \theta_0 - \sqrt{\varepsilon_2/\mu_2} \cos \theta_t}{\sqrt{\varepsilon_1/\mu_1} \cos \theta_0 + \sqrt{\varepsilon_2/\mu_2} \cos \theta_t}, \quad (59)$$

где  $\theta_t$  – угол преломления. Как видно из (59), мы не можем подобрать параметры  $\varepsilon_2$  и  $\mu_2$  таким образом, чтобы обеспечить отсутствие отражения, то есть  $r = 0$ , для всех углов падения  $\theta_0$ .

Оказывается, отсутствие отражения можно получить при использовании более сложных сред. Давайте рассмотрим анизотропную среду, которая характеризуется следующими тензорами

$$\hat{\varepsilon} = \begin{pmatrix} \varepsilon_x & 0 & 0 \\ 0 & \varepsilon_y & 0 \\ 0 & 0 & \varepsilon_z \end{pmatrix} \quad \text{и} \quad \hat{\mu} = \begin{pmatrix} \mu_x & 0 & 0 \\ 0 & \mu_y & 0 \\ 0 & 0 & \mu_z \end{pmatrix}. \quad (60)$$

Материальные параметры (60) можно подставить в уравнения Максвелла и получить

$$\frac{\partial E_z}{\partial y} = \frac{i\omega}{c} B_x = \frac{i\omega}{c} \mu_x H_x \quad (61a)$$

$$-\frac{\partial E_z}{\partial x} = \frac{i\omega}{c} B_y = \frac{i\omega}{c} \mu_y H_y \quad (61b)$$

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = -\frac{i\omega}{c} D_z = -\frac{i\omega}{c} \varepsilon_z E_z. \quad (61c)$$

Для понимания того, как распространяются волны в такой среде, подставим искомый вид решения

$$\sim e^{-i\omega t + ik_x x + ik_y y} \quad (62)$$

в уравнения (61). Это даёт дисперсионное соотношение

$$\frac{k_x^2}{\mu_y \varepsilon_z} + \frac{k_y^2}{\mu_x \varepsilon_z} = \frac{\omega^2}{c^2}. \quad (63)$$

Теперь рассмотрим отражение при падении плоской волны из обычной изотропной среды с  $\varepsilon_1$ ,  $\mu_1$  на границу такой анизотропной среды. Волновые числа для падающей волны задаются ее углом падения:

$$k_{1x} = k_1 \cos \theta_0, \quad k_{1y} = k_1 \sin \theta_0. \quad (64)$$

Используя непрерывность тангенциальных компонент, мы получим коэффициент отражения

$$r = \frac{k_{1x}/\mu_1 - k_{2x}/\mu_{2y}}{k_{1x}/\mu_1 + k_{2x}/\mu_{2y}}. \quad (65)$$

Следовательно, условие отсутствия отражения  $r = 0$  будет

$$\frac{k_{1x}}{\mu_1} = \frac{k_{2x}}{\mu_{2y}}. \quad (66)$$

Мы можем подставить  $k_{1x}$  из (64) в (66) и выразить  $k_{2x}$ . С другой стороны,  $k_{2y} = k_{1y}$  из условия непрерывности фазы поля вдоль границы. Полученные  $k_{2x}$ ,  $k_{2y}$  можно подставить в дисперсионное уравнение (63). Тогда получим

$$\frac{\varepsilon_1 \mu_{2y}}{\varepsilon_{2z} \mu_1} \cos^2 \theta_0 + \frac{\varepsilon_1 \mu_1}{\varepsilon_{2z} \mu_{2x}} \sin^2 \theta_0 = 1. \quad (67)$$

Уравнение (67) будет выполняться, если

$$\varepsilon_{2z} = \varepsilon_1 s_x, \quad \mu_{2y} = \mu_1 s_x, \quad \mu_{2x} = \mu_1 s_x^{-1}, \quad (68)$$

где  $s_x$  – произвольный комплексный множитель. Как и в 1D случае, мы можем выбрать этот множитель так, чтобы волна затухала:

$$s_x = 1 + i \frac{4\pi\bar{\sigma}_x}{\omega}, \quad (69)$$

где  $\bar{\sigma}_x = \sigma_x^e/\varepsilon_1 = \sigma_x^m/\mu_1$  – коэффициент (проводимость), который обеспечивает поглощение. Давайте посмотрим, как будет распространяться волна в анизотропной среде при падении плоской волны. Волновые числа будут

$$k_{2y} = k_1 \sin \theta_0, \quad k_{2x} = s_x k_1 \sin \theta_0. \quad (70)$$

Тогда

$$\operatorname{Re} k_{2x} = k_1 \cos \theta_0 = k_{1x}, \quad \operatorname{Im} k_{2x} = \frac{4\pi\bar{\sigma}_x}{\omega} k_1 \cos \theta_0. \quad (71)$$

Наличие  $\operatorname{Im} k_{2x} > 0$  обеспечивает затухание волны. Коэффициент затухания зависит от угла падения. Наиболее сильно затухают волны, падающие по нормали  $\theta_0 = 0$ . Материальные параметры для слоя перпендикулярного оси  $x$  (правая и левая границы области моделирования) принимают следующий вид:

$$\hat{\varepsilon}_2 = \varepsilon_1 \begin{pmatrix} s_x^{-1} & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & s_x \end{pmatrix} \quad \text{и} \quad \hat{\mu}_2 = \mu_1 \begin{pmatrix} s_x^{-1} & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & s_x \end{pmatrix}, \quad (72)$$

где параметр  $\bar{\sigma}_x$  задан уравнением (69). Таким образом, мы получили параметры, при которых падающие волны не будут отражаться от слоя и будут поглощаться по мере проникновения внутрь.

Аналогичным образом получаются уравнения для слоя, перпендикулярного оси  $y$  (верхняя и нижняя границы). На углах области моделирования используют комбинацию материальных уравнений для двух направлений, то есть

$$\hat{\varepsilon}_2 = \varepsilon_1 \begin{pmatrix} s_x^{-1} & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & s_x \end{pmatrix} \cdot \begin{pmatrix} s_y & 0 & 0 \\ 0 & s_y^{-1} & 0 \\ 0 & 0 & s_y \end{pmatrix} \quad (73)$$

Таким образом, на углах материальные уравнения задаются как

$$\hat{\varepsilon}_2 = \varepsilon_1 \begin{pmatrix} s_x^{-1}s_y & 0 & 0 \\ 0 & s_x s_y^{-1} & 0 \\ 0 & 0 & s_x s_y \end{pmatrix} \quad \text{и} \quad \hat{\mu}_2 = \mu_1 \begin{pmatrix} s_x^{-1}s_y & 0 & 0 \\ 0 & s_x s_y^{-1} & 0 \\ 0 & 0 & s_x s_y \end{pmatrix}. \quad (74)$$

Схематическое распределение материальных параметров показано на рис. 22.

Давайте теперь обсудим как использовать полученные материальные уравнения для пересчета полей в таких анизотропных слоях. В нашем случае ТЕ поляризации мы используем

$$\varepsilon_z = \varepsilon_1 s_x s_y, \quad \mu_x = \mu_1 s_x^{-1} s_y, \quad \mu_y = \mu_1 s_x s_y^{-1}. \quad (75)$$

$\sigma_x \neq 0$ $\sigma_y \neq 0$	$\sigma_x = 0$ $\sigma_y \neq 0$	$\sigma_x \neq 0$ $\sigma_y \neq 0$
$\sigma_x \neq 0$ $\sigma_y = 0$		$\sigma_x \neq 0$ $\sigma_y = 0$
$\sigma_x \neq 0$ $\sigma_y \neq 0$	$\sigma_x = 0$ $\sigma_y \neq 0$	$\sigma_x \neq 0$ $\sigma_y \neq 0$

Рис. 22: Параметры идеально согласованных слоев (PMLs) около границы области моделирования.

Рассмотрим сначала уравнение

$$D_z(\omega) = \varepsilon_z(\omega)E_z(\omega) = \varepsilon_1 s_x(\omega)s_y(\omega)E_z(\omega). \quad (76)$$

Это уравнение можно преобразовать к виду

$$-i\omega D_z = \varepsilon_1 \left( -i\omega + 4\pi(\bar{\sigma}_x + \bar{\sigma}_y) + \frac{4\pi\bar{\sigma}_x 4\pi\bar{\sigma}_y}{-i\omega} \right) E_z. \quad (77)$$

Для перехода из частотной области во временную мы заменяем

$$-i\omega \rightarrow \frac{\partial}{\partial t} \quad \text{и} \quad \frac{1}{-i\omega} \rightarrow \int dt \quad (78)$$

и получаем

$$\frac{\partial D_z}{\partial t} = \varepsilon_1 \frac{\partial E_z}{\partial t} + \varepsilon_1 4\pi(\bar{\sigma}_x + \bar{\sigma}_y) + \varepsilon_1 16\pi^2 \bar{\sigma}_x \bar{\sigma}_y \int_0^t dt E_z. \quad (79)$$

Проводя дискретизацию соотношения (79) в момент  $n + 1/2$  и вводя обозначение  $\eta_{x,y} = 2\pi\bar{\sigma}_{x,y}\Delta t$ , получаем

$$D_z^{n+1} - D_z^n = \varepsilon_1(E_z^{n+1} - E_z^n) + \varepsilon_1(\eta_x + \eta_y)(E_z^{n+1} + E_z^n) + \varepsilon_1 4\eta_x \eta_y \sum_{m=0}^n E_z^m. \quad (80)$$

Это ведет к формуле для нахождения  $E_z^{n+1}$ :

$$E_z^{n+1} = \frac{D_z^{n+1} - D_z^n}{\varepsilon_1(1 + \eta_x + \eta_y)} + \frac{(1 - \eta_x - \eta_y)E_z^n}{1 + \eta_x + \eta_y} - \frac{4\eta_x \eta_y S^n}{1 + \eta_x + \eta_y}, \quad S^n = \sum_{m=0}^n E_z^m. \quad (81)$$

Теперь перейдем к уравнению

$$B_x(\omega) = \mu_x(\omega)H_x(\omega) = \mu_1 \left(1 + \frac{4\pi\bar{\sigma}_x}{-i\omega}\right)^{-1} \left(1 + \frac{4\pi\bar{\sigma}_y}{-i\omega}\right) H_x(\omega). \quad (82)$$

Это уравнение можно переписать как

$$\left(1 + \frac{4\pi\bar{\sigma}_x}{-i\omega}\right) B_x = \mu_1 \left(1 + \frac{4\pi\bar{\sigma}_y}{-i\omega}\right) H_x \quad (83)$$

или во временной области

$$\left(\frac{\partial}{\partial t} + 4\pi\bar{\sigma}_x\right) B_x = \mu_1 \left(\frac{\partial}{\partial t} + 4\pi\bar{\sigma}_y\right) H_x. \quad (84)$$

Проводя дискретизацию в момент  $n + 1$  и выражая  $H_z^{n+3/2}$ , получаем

$$H_x^{n+3/2} = \frac{1 - \eta_y}{1 + \eta_y} H_x^{n+1/2} + \frac{1 + \eta_x}{\mu_1(1 + \eta_y)} B_x^{n+3/2} + \frac{1 - \eta_x}{\mu_1(1 + \eta_y)} B_x^{n+1/2}. \quad (85)$$

Аналогичным образом записывается и выражение для  $H_y^{n+3/2}$ . Таким образом, мы получили все формулы для расчета полей в идеально согласованном слое. Выбор параметров  $\eta_{x,y}$  проходит так же, как и в 1D случае. То есть,  $\eta_x$  растет монотонно с глубиной проникновения в слой, расположенные перпендикулярно оси  $x$ , а  $\eta_y$  растет аналогичным образом с глубиной проникновения в слой, расположенные перпендикулярно оси  $y$ .

## 16 Пример 4: Возбуждение волн точечным источником и их поглощение идеально согласованным слоем

Рассмотрим, как ведут себя волны в 2D случае при использовании поглощающего слоя, изученного в разделе 15. Параметры моделирования:  $N_x = 600$ ,  $N_y = 400$ ,  $\Delta x = 20$  нм,  $\varepsilon_1 = \mu_1 = 1$ ,  $\xi = 0.5$ . Будем возбуждать волны точечным источником тока

$$J_z(t) = A e^{-(t-t_0)^2/\tau^2} \cos((t-t_0)\omega_0), \quad (86)$$

где  $\lambda_0 = 600$  нм,  $\tau = 5$  фс,  $t_0 = 4\tau$ . Источник тока расположен в ячейке  $i_0 = 250$ ,  $j_0 = 120$ . Параметры поглощающего слоя:  $N_{pml} = 12$ ,  $\eta_0 = 0.4$ ,  $\alpha = 3$ . Рисунок 23 показывает распределения полей в разные моменты времени при наличии самосогласованного слоя и без него. Видно, что пока поле

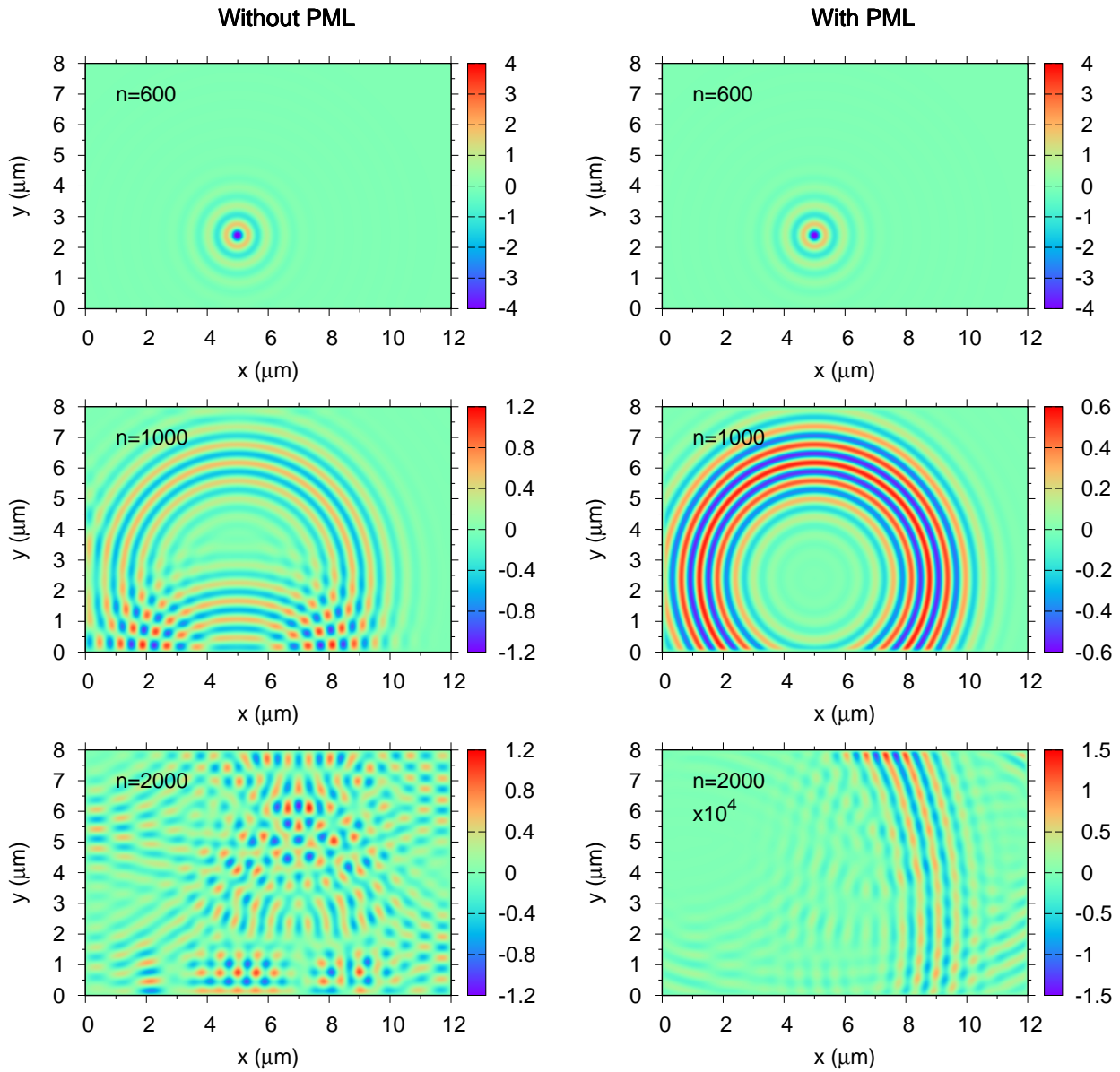


Рис. 23: Излучение точечного источника при наличии самосогласованного слоя на границе (with PML) и без него (without PML). Количество шагов во времени  $n$  указано на каждом рисунке. Поле при  $n = 2000$  при наличии PML было умножено на указанное значение ( $10^4$ ).

не достигло слоя,  $n = 600$ , слой роли не играет. Без слоя происходит полное отражение волн от границы моделирования. Со слоем волны поглощаются, что приводит к очень низкому,  $< 0.5 \times 10^{-3}$ , значению коэффициента отражения. Таким образом моделируется излучение источников в свободном пространстве.

## 17 Разделение на области полного и рассеяного полей в 2D

Как правило, в задачах рассеяния размеры падающей волны значительно превосходят размеры рассеивающего объекта. Поэтому для получения численного решения прибегают к разделению области моделирования на область полного и рассеяного полей. Такое разделение показано схематично на рис. 24.

Как и в 1D случае, см. раздел 12, мы должны изменить формулы для пересчета полей около такой границы. Давайте предположим, что область полного поля начинается с ячейки  $i = i_1$  и заканчивается ячейкой  $i = i_2$  вдоль оси  $x$  и, соответственно,  $j_1, j_2$  вдоль  $y$ . Предположим, что мы пересчитали все поля, используя уравнения (55), то есть без учета разделения на области.

Рассмотрим теперь верхнюю границу. При пересчёте из уравнений Максвелла поля  $B_x$  в ячейках  $j = j_2$  (то есть в области полного поля) мы брали поля  $E_z$  в ячейках  $j_2 + 1$  (то есть в области рассеяного поля). Таким образом, для правильного расчёта поля  $B_x$  мы должны добавить к использованному значению  $E_z$  в ячейках  $j_2 + 1$  значение, которое имеет там падающее поле  $E_z$ . Аналогично, при расчёте  $D_z$  в ячейках  $j = j_2 + 1$  (в области рассеяного поля) мы брали поля  $H_x$  в ячейках  $j = j_2$  (в области полного поля). Поэтому из использованных значений  $H_x$  мы должны вычесть значения падающего поля. Те же самые правила работают и на других границах. В результате мы получаем следующие изменения в формулах для  $B_x, B_y, D_z$ :

$$j = j_2, i_1 \leq i \leq i_2 : \quad B_x^{n+1/2}[i, j] += -\xi E_{iz}^n[i, j + 1], \quad (87a)$$

$$j = j_1 - 1, i_1 \leq i \leq i_2 : \quad B_x^{n+1/2}[i, j] += \xi E_{iz}^n[i, j + 1], \quad (87b)$$

$$i = i_2, j_1 \leq j \leq j_2 : \quad B_y^{n+1/2}[i, j] += \xi E_{iz}^n[i + 1, j], \quad (87c)$$

$$i = i_1 - 1, j_1 \leq j \leq j_2 : \quad B_y^{n+1/2}[i, j] += -\xi E_{iz}^n[i + 1, j], \quad (87d)$$

$$j = j_2 + 1, i_1 \leq i \leq i_2 : \quad D_z^{n+1}[i, j] += -\xi H_{ix}^{n+1/2}[i, j - 1], \quad (87e)$$

$$i = i_2 + 1, j_1 \leq j \leq j_2 : \quad D_z^{n+1}[i, j] += \xi H_{iy}^{n+1/2}[i - 1, j], \quad (87f)$$

$$j = j_1, i_1 \leq i \leq i_2 : \quad D_z^{n+1}[i, j] += \xi H_{ix}^{n+1/2}[i, j - 1], \quad (87g)$$

$$i = i_1, j_1 \leq j \leq j_2 : \quad D_z^{n+1}[i, j] += -\xi H_{iy}^{n+1/2}[i - 1, j], \quad (87h)$$

где оператор  $+=$  подразумевает добавление к переменной в левой части значения в правой части,  $E_{iz}, H_{ix}, H_{iy}$  – компоненты падающего поля, которое считается заданным. Значения падающего поля используются только вблизи границы разделения областей.



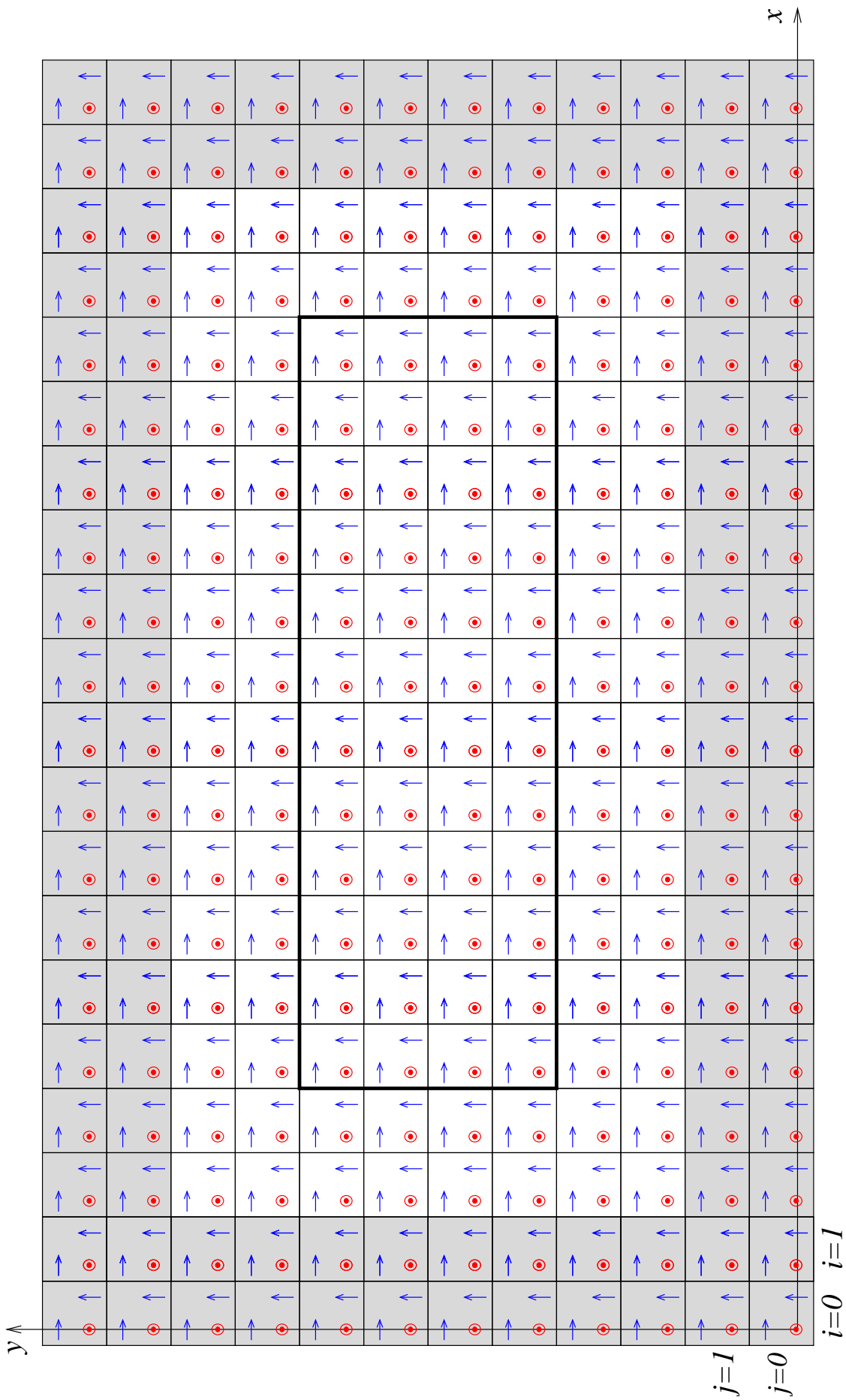


Рис. 24: 2D сетка. Схематически показаны поглощающие слои (серый цвет) и граница между областями полного и рассеяного полей (толстая линия).

## 18 Пример 5: Рассеяние плоской волны на цилиндре

Возьмём импульс

$$E_z(x, t) = Ae^{-X^2/V^2\tau^2} \cos(\omega_0 X/V), \quad H_y(x, t) = -E_z(x, t), \quad (88)$$

где  $X = x - V(t - t_0)$ , распространяющийся вдоль оси  $x$ , и рассмотрим, как он взаимодействует с рассеивающим объектом (цилиндром). Давайте сначала рассмотрим распространение волны без цилиндра, см. рис. 25, чтобы проверить правильность задания импульса. Область моделирования ( $N_x = 1200$ ,  $N_y = 400$ ) разделена на область полного поля и область рассеянного поля. Границы области полного поля:  $i_1 = 100$ ,  $i_2 = 500$ ,  $j_1 = 80$ ,  $j_2 = 320$ . Параметры:  $\Delta x = 20$  нм,  $\xi = 0.5$ ,  $\lambda_0 = 600$  нм,  $\tau = 5$  фс,  $t_0 = 4\tau$ ,  $N_{pml} = 12$ ,  $\eta_0 = 0.4$ ,  $\alpha = 3$ .

Падающий импульс (88) должен входить в область моделирования слева и исчезать справа, если внутри области нет рассеивающего объекта. Следует заметить, что такое выключение поля справа будет работать хорошо, если численно рассчитанный импульс совпадает с аналитической зависимостью (88). Как мы убедились в разделе (4), численное распространение приводит к искажению импульса со временем, а поэтому выключение на правой границе будет неполным, причём ошибка будет расти с увеличением области полного поля по оси  $x$ . Поэтому для достижения полноты выключения часто используют не аналитическую зависимость, а численно рассчитанную на отдельной (вспомогательной) 1D сетке, которая соответствует 2D сетке при фиксированной координате  $y$ . Аналитическая зависимость типа (88) используется для запуска импульсов на 2D и 1D сетках синхронно. На 1D сетке распространение импульса рассчитывается согласно 1D модели. Полученные значения такого импульса в каждый момент времени используются в качестве падающего поля для расчета полей как верхней, нижней и правой границах в 2D сетке в соответствии с (87). При таком синхронном моделировании происходит компенсация дисперсии, а значит и практически полное выключение импульса на 2D сетке справа.

Рисунок 25 сравнивает результат моделирования свободно распространяющегося импульса на 2D сетке при использовании аналитической зависимости падающего поля на границе областей и соответствующих значений, полученных на вспомогательной 1D сетке. Разница между результатами хорошо видна при  $n = 2200$ . При использовании аналитической зависимости остаётся довольно значительный импульс справа в области рассеянного поля. В случае использования вспомогательной 1D сетки видны два импульса с пренебрежимо малыми амплитудами. Один из них связан с дискретизацией

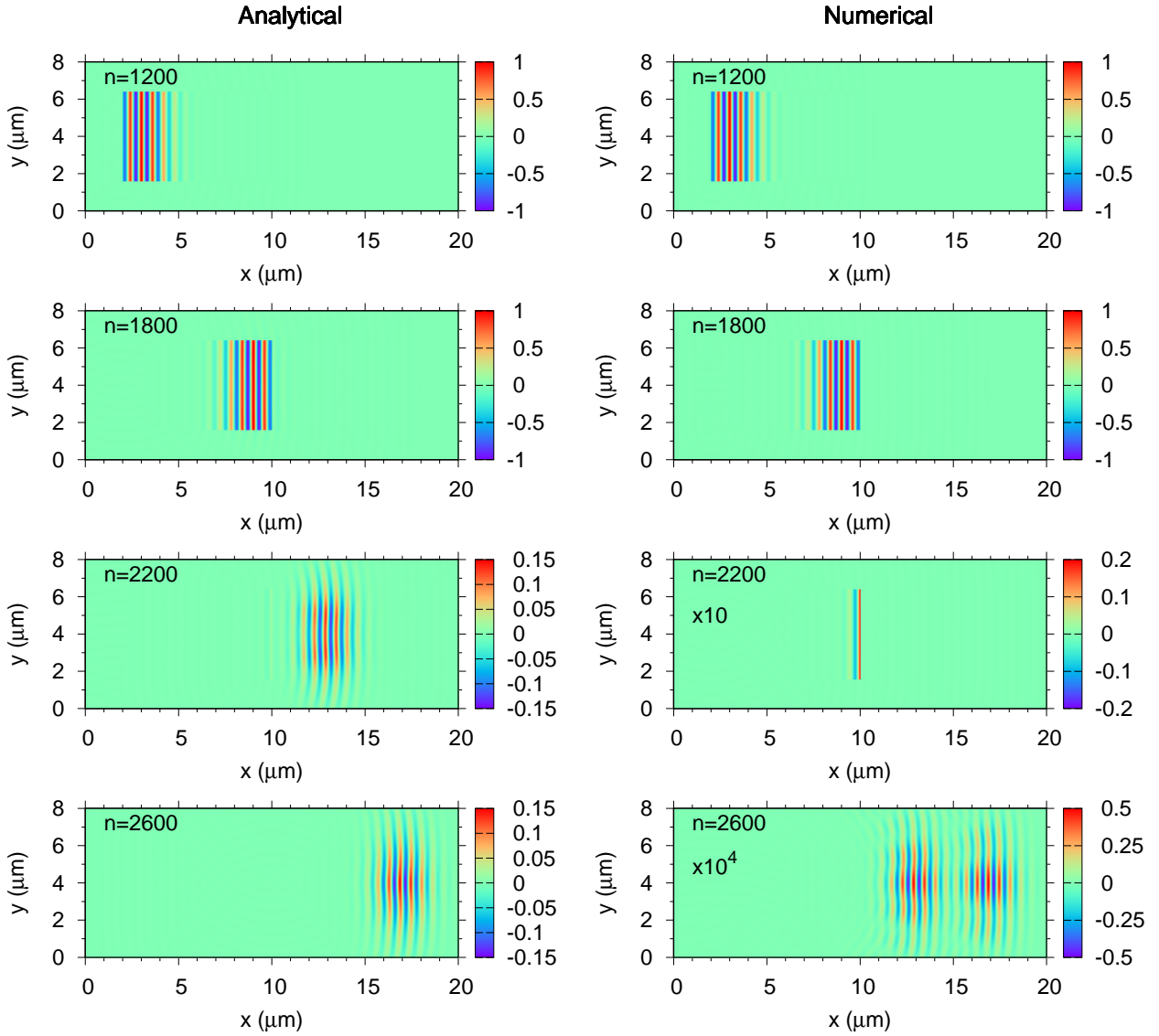


Рис. 25: Распространение импульса с плоским фронтом при разделении области моделирования на области полного и рассеянного полей. Левые картинки показывают распределения  $E_z(x, y)$ , когда падающее поле в (87) рассчитывается аналитически. Правые картинки – когда поле рассчитывается численно на вспомогательной 1D сетке. На рисунках справа при  $n = 2200$  и  $n = 2600$  поля были умножены на указанные значения.

процесса выключения полей на границе. Второй импульс появляется из-за того, что при вводе поля на 1D сетке возникает очень слабый импульс, распространяющийся влево и затем отражающийся от левой границы, где не было поглощающего слоя.

Убедившись в правильности ввода падающей волны в область моделирования, мы теперь помещаем в нее цилиндр с  $\varepsilon = 2$ ,  $R = 1.2 \mu\text{m}$ . Полная область моделирования имеет размеры  $N_x = N_y = 1000$ . Область полного

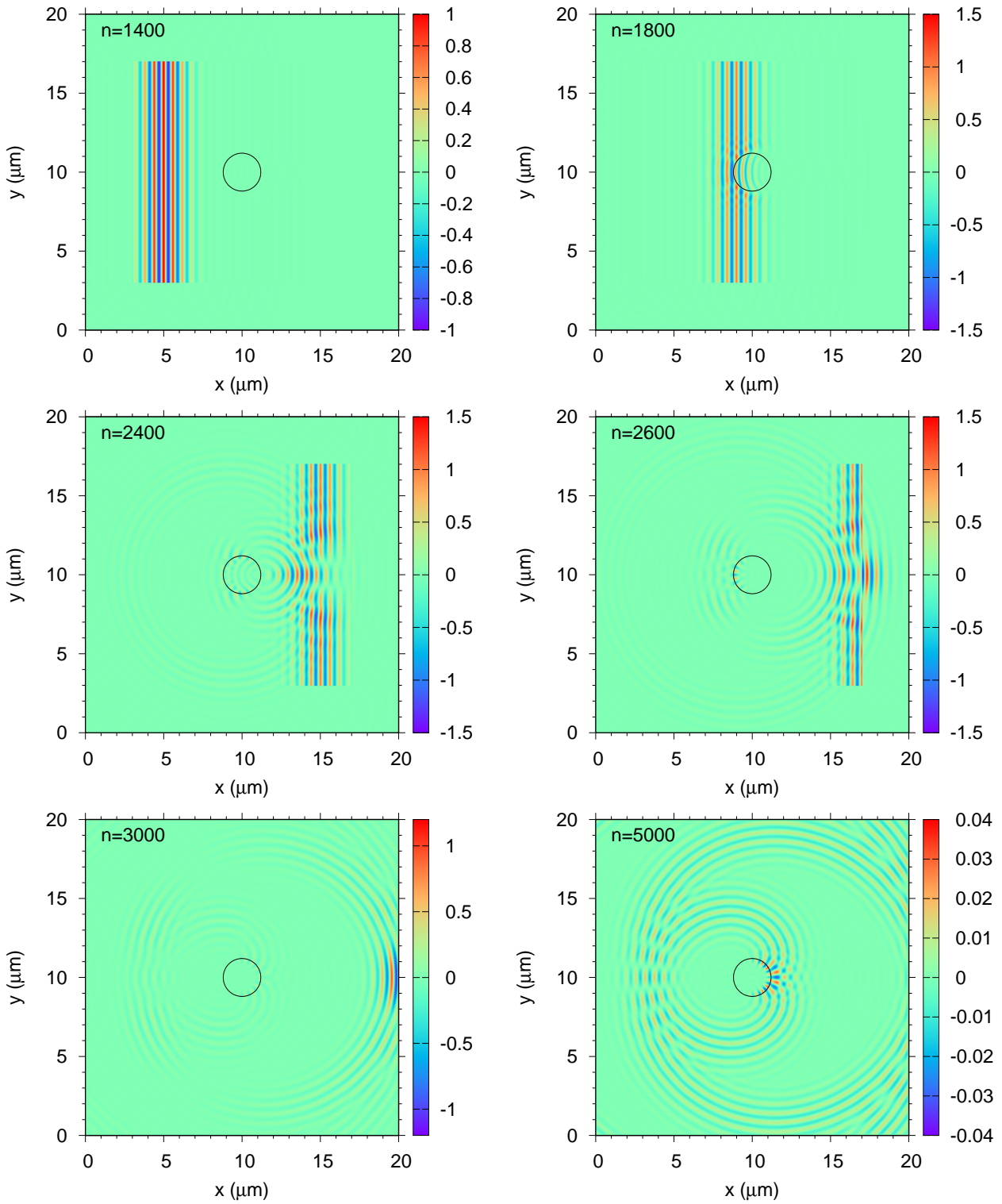


Рис. 26: Рассеяние импульса с плоским фронтом на цилиндре. Картинки показывают распределения  $E_z(x, y)$  в разные моменты времени, соответствующие количеству шагов во времени  $n$ .

поля ограничена ячейками  $i_1 = 150$ ,  $i_2 = N_x - 150$ ,  $j_1 = 150$ ,  $j_2 = N_y - 150$ . Результаты моделирования показаны на рис. 26. При  $n = 1800$  видно, что по-

ле внутри цилиндра запаздывает по отношению к полю вне его. Далее, при  $n = 2400$ , цилиндр забирает эту часть волны и рассеивает в другие направления. При  $n = 2600$  хорошо видно отделение рассеянного поля на правой границе раздела областей полного и рассеянного полей. Энергия внутри цилиндра может находиться в течение довольно долгого времени и даже при  $n = 5000$  цилиндр продолжает переизлучать волны. Это связано с наличием в цилиндре резонансов (мод шепчущей галереи), которые возбуждаются падающим импульсом.

## 19 Расчёт сечения рассеяния

Одной из характеристик рассеяния плоской волны на объекте является сечение рассеяния  $\sigma$ , которое определяется через соотношение

$$P_{sc} = \sigma S_0, \quad (89)$$

где  $P_{sc}$  – рассеянная мощность,  $S_0$  – плотность потока энергии падающей волны. Часто удобно нормировать сечение рассеяния на геометрическое сечение  $D$  объекта, то есть его размер в направлении, которое перпендикулярно волновому вектору падающей волны:

$$\frac{\sigma}{D} = \frac{P_{sc}}{S_0 D} = \frac{P_{sc}}{P_{inc}}, \quad (90)$$

где  $P_{inc} = DS_0$  – мощность, падающая на объект.

Так как сечение рассеяния определено для монохроматической волны, то для его нахождения из временных зависимостей полей надо воспользоваться преобразованием Фурье. Спектральная плотность потока энергии рассчитывается на произвольном контуре в области рассеяния, а затем интегрируется по контуру. Для области моделирования как на рис. 24 удобно выбрать прямоугольных контур со сторонами, параллельными границам области моделирования.

Давайте рассмотрим, как находится спектральная плотность потока энергии для части контура на правой границе (параллельно оси  $y$ ). Используя вектор Пойнтинга

$$S_x(t) = -\frac{c}{4\pi} E_z(t) H_y(t), \quad (91)$$

мы находим плотность энергии в каждой точке контура

$$W = \int_{-\infty}^{\infty} dt S_x(t) = 2\pi \int_0^{\infty} d\omega S_x(\omega). \quad (92)$$

Спектральная плотность потока энергии  $S_x(\omega)$  в (92) выражается как

$$S_x(\omega) = -\frac{c}{2\pi} \operatorname{Re} \tilde{E}_z(\omega) \tilde{H}_y(\omega), \quad (93)$$

где  $\tilde{E}_z(\omega)$  и  $\tilde{H}_y(\omega)$  – Фурье спектры, рассчитанные по (24). Заметим, что при расчёте полей на контуре следует учитывать, что поля  $E_z(t)$  и  $H_y(t)$  заданы в смещенных точках. Аналогичным образом рассчитывается плотность потока энергии и на других участках контура. Интегрируя полученные величины по контуру, мы найдем рассеянную мощность в частотной области. Нормируя её на падающую, получаем сечение рассеяния для данного объекта в заданном интервале частот.

## 20 Расчёт диаграммы рассеяния и пересчёт полей из ближней зоны в дальнюю

Диаграмма рассеяния – это распределение рассеянной мощности по углу в дальней зоне при падении монохроматической волны. Рассматривая конкретный 2D случай ТЕ поляризации, мы можем представить поля в дальней зоне в виде

$$E_z(\rho, \varphi) \approx \frac{f(\varphi)}{\sqrt{k_1 \rho}} e^{-i\pi/4 + ik_1 \rho}, \quad H_\varphi(\rho, \varphi) \approx -\sqrt{\varepsilon_1} E_z(\rho, \varphi), \quad (94)$$

где  $k_1 = \sqrt{\varepsilon_1} \omega / c$  и  $\varepsilon_1$  – диэлектрическая проницаемость среды. Функция  $f(\varphi)$  описывает угловое распределение поля. Используя вектор Пойнтинга и интегрирую по окружности с большим радиусом, мы получаем рассеянную мощность

$$P_{sc} = \frac{c^2}{2\pi\omega} \int_0^{2\pi} d\omega |f(\varphi)|^2, \quad (95)$$

где функция  $|f(\varphi)|^2$  даёт нам угловую плотность рассеянной мощности и является диаграммой направленности. Для ее нахождения нам надо представить поле  $E_z$  асимптотически в дальней зоне. Как правило, задача рассеяния решается в очень ограниченном пространстве около объекта и поля за пределами этой области не рассчитываются. Однако эти поля можно найти, зная их распределение около объекта, то есть воспользоваться методикой пересчёта полей из ближней зоны в дальнюю.

Так как область, в которой надо рассчитать поля, считается однородной, то поле в ней можно представить как результат излучения неких эффективных токов на границе, см. рис. 27. Эффективные токи определяются полями

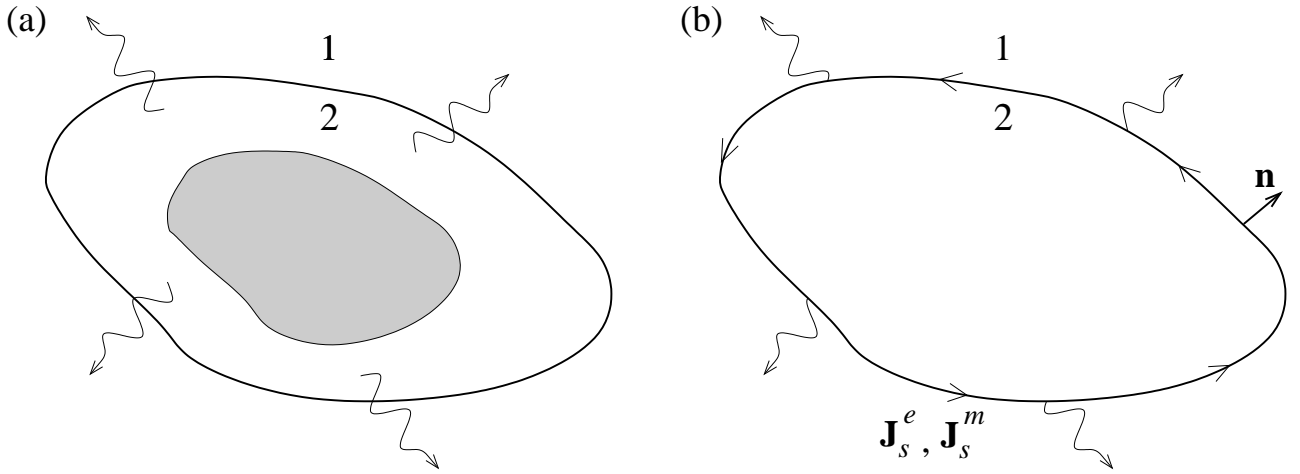


Рис. 27: Сведение расчёта рассеянных полей в дальней зоне к решению задачи об излучении полей эффективными токами. (а) Задача рассеяния с произвольно выбранным контуром, внутри которого находится рассеивающий объект. (б) Задача излучения эффективными поверхностными токами  $\mathbf{J}_s^e$ ,  $\mathbf{J}_s^m$ , которые определяются из значений рассеянных полей на выбранном контуре.

на контуре, внутри которого находится рассеивающий объект. Для нахождения токов воспользуемся уравнениями Максвелла, которые включают как электрический ток  $\mathbf{J}^e$ , так и магнитный  $\mathbf{J}^m$ :

$$\nabla \times \mathbf{E} = -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} - \frac{4\pi}{c} \mathbf{J}^m \quad (96a)$$

$$\nabla \times \mathbf{H} = \frac{1}{c} \frac{\partial \mathbf{D}}{\partial t} + \frac{4\pi}{c} \mathbf{J}^e \quad (96b)$$

Используя (96), получим граничные условия на поверхности с токами:

$$\mathbf{n} \times (\mathbf{E}_1 - \mathbf{E}_2) = -\frac{4\pi}{c} \mathbf{J}_s^m$$

$$\mathbf{n} \times (\mathbf{H}_1 - \mathbf{H}_2) = \frac{4\pi}{c} \mathbf{J}_s^e$$

Токи можно найти из условия, что на внешней стороне контура (область 1 на рис. 27) токи создают такое же поле, как и рассеивающий объект. А поле внутри контура  $E_2 = H_2 = 0$ . Тогда токи выражаются через известные поля  $\mathbf{E}_1$  и  $\mathbf{H}_1$  как

$$\mathbf{J}_s^m = -\frac{c}{4\pi} \mathbf{n} \times \mathbf{E}_1,$$

$$\mathbf{J}_s^e = \frac{c}{4\pi} \mathbf{n} \times \mathbf{H}_1.$$

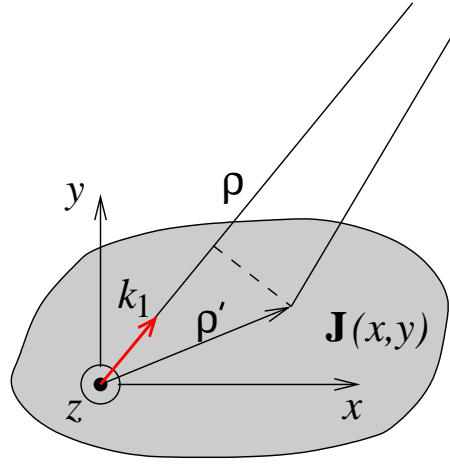


Рис. 28: Нахождение поля, созданного заданным распределением тока  $\mathbf{J}(x, y)$ , в дальней зоне.

В общем случае поле  $E_z(x, y)$ , генерируемое токами  $J_z^e(x, y)$ ,  $J_x^m(x, y)$ ,  $J_y^m(x, y)$  записывается в виде

$$\begin{aligned}
 E_z(x, y) = & \iint dx' dy' G_z^e(x, y; x', y') J_z^e(x', y') \\
 & + \iint dx' dy' G_x^m(x, y; x', y') J_x^m(x', y') \\
 & + \iint dx' dy' G_y^m(x, y; x', y') J_y^m(x', y'), \quad (99)
 \end{aligned}$$

где  $G_z^e(x, y; x', y')$ ,  $G_x^m(x, y; x', y')$ ,  $G_y^m(x, y; x', y')$  – функции Грина, то есть поля, созданные точечными токами. Функции Грина рассчитываются в разделе 22. В нашем случае все токи – поверхностные и интегралы сводятся к контурным.

Заметим также, что в некоторых случаях рассеянное поле в дальней зоне можно найти и с помощью объёмных токов. Действительно, при отсутствии дисперсии рассеянное поле можно интерпретировать, как созданное током

$$\mathbf{J}(x, y) = -i\omega \frac{\varepsilon_2 - \varepsilon_1}{4\pi} \mathbf{E}(x, y), \quad (100)$$

где  $\mathbf{E}(x, y)$  – поле внутри рассеивающего объекта,  $\varepsilon_2$  – диэлектрическая проницаемость объекта,  $\varepsilon_1$  – диэлектрическая проницаемость окружающей среды. Поле в дальней зоне рассчитывается с использованием функции Грина и объёмного распределения тока без магнитных токов в (99).

Для расчёта поля по (99) в дальней зоне можно сразу брать асимптотические выражения для функций Грина (125). Возьмём, например,

$$G_z^e(x, y; x', y') = -\frac{\pi\omega\mu_1}{c^2} H_0^{(1)}(k_1 |\boldsymbol{\rho} - \boldsymbol{\rho}'|), \quad (101)$$



где  $|\boldsymbol{\rho} - \boldsymbol{\rho}'| = \sqrt{(x - x')^2 + (y - y')^2}$ . Используя рис. 28, получим  $k_1|\boldsymbol{\rho} - \boldsymbol{\rho}'| \approx k_1\rho - \mathbf{k}_1\boldsymbol{\rho}'$ . Тогда с помощью асимптотического поведения (109) функции  $H^{(1)}(z)$  при  $z \rightarrow \infty$  получаем поле в виде (94) с

$$f(\varphi) = -\frac{\omega\mu_1\sqrt{2\pi}}{c^2} \iint dx' dy' J_z(x', y') e^{-i\mathbf{k}_1\boldsymbol{\rho}'}, \quad (102)$$

где  $\varphi$  – это угол вектора  $\mathbf{k}_1$ , то есть направление на точку наблюдения. Аналогичным образом мы можем учесть и вклады  $J_{x,y}^m(x, y)$  в угловое распределение.

## 21 Заключение

В пособии были представлены основы численного моделирования электромагнитных волн в различных средах в 1D и 2D случаях. Также обсуждались такие темы, как написание программ, организация директорий, получение выполняемых файлов. Теоретические и практические вопросы моделирования иллюстрировались примерами с конкретными программами и с обсуждением результатов.

Особое внимание следует уделять численным ошибкам, которые всегда возникают при расчёте. Правильный выбор сетки и временного шага позволяет их уменьшить. Как правило, на практике достоверность полученного результата проверяется путём варьирования значений пространственных и временных шагов и достижения совпадения получаемых решений. Появление численной дисперсии и необходимость дискретизации всего моделируемого пространства вносят ограничения на область применения метода. В частности, он хорошо работает для моделирования областей с размерами, которые сравнимы с длиной волны или превосходят ее на два-три порядка. Тем не менее, часто выделение малого объема со сложной геометрией бывает достаточно для того, чтобы найти поля и на больших масштабах с использованием приема пересчёта ближнего поля в дальнее, который также часто используется совместно с методом КРВО.

Представленные подходы, как теоретические, так и практические, могут быть использованы не только для решения электродинамических задач, но и, например, для моделирования акустических волн, описываемых системой уравнений для давления и скорости. Также подходы применимы для моделирования квантовой динамики с помощью функции распределения, описываемой уравнением Шрёдингера. Подходы, использованные для моделирования идеально согласованных слоев могут быть применены и к магнитным материалам, и к анизотропным средам. Моделирование в 1D и 2D случаях может быть далее обобщено и для 3D задач при использовании соответствующей сетки. Полученные алгоритмы в декартовых координатах позволяют их адаптировать для цилиндрических и сферических координат.

Метод КРВО применим к решению различных электродинамических задач в силу своей простоты как в теоретическом понимании, так и в практическом воплощении. При этом следует отметить, что другие методы (например, метод конечных элементов во временной и частотной областях) также могут быть успешно использованы и, в зависимости от проблемы, давать более точные и быстрые результаты. На практике такие методы, как правило, требуют больших затрат на получение формул для расчёта и их практическое воплощение. Для выбора оптимального способа решения конкретной задачи требуется хорошо представлять все возможные методы.

В силу популярности метода КРВО существует большое количество литературы на эту тему: статьи в научных журналах, монографии, учебники и различные пособия. Считается, что основные идеи алгоритма КРВО представлены в 1966 г. в статье [1]. Сильное развитие метод получил в период с конца 1980-х, что связано с прогрессом в вычислительной технике. Пожалуй, одно из наиболее полных описаний метода представлено в книге [2]. Более короткое описание можно найти, например, в учебнике [3].

## Список литературы

- [1] K. Yee, “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media”, IEEE Transactions on Antennas and Propagation, vol. 14, no. 3, pp. 302–307, 1966.
- [2] A. Taflove and S. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, 3rd edn. Artech House, 2005.
- [3] D. Sullivan, Electromagnetic Simulation Using the FDTD Method, 2nd Edition, Wiley-IEEE Press, 2013.

## 22 Приложение: Излучение точечных источников в 2D

Для того, чтобы найти поле, создаваемое заданным распределением токов, в разделе 17 нам нужно знать поля от точечных монохроматических источников различных типов (электрического и магнитного) и с разной ориентацией, то есть функции Грина. Для их нахождения в среде с  $\varepsilon_1$  и  $\mu_1$  воспользуемся уравнениями Максвелла для полей  $\{E_z, H_\rho, H_\varphi\}$  с временной зависимостью  $e^{-i\omega t}$  в цилиндрических координатах:

$$\frac{1}{\rho} \frac{\partial E_z}{\partial \varphi} = \frac{i\omega}{c} B_\rho - \frac{4\pi}{c} J_\rho^m, \quad (103a)$$

$$-\frac{\partial E_z}{\partial \rho} = \frac{i\omega}{c} B_\varphi - \frac{4\pi}{c} J_\varphi^m, \quad (103b)$$

$$\frac{1}{\rho} \frac{\partial}{\partial \rho} (\rho H_\varphi) - \frac{1}{\rho} \frac{\partial H_\rho}{\partial \varphi} = -\frac{i\omega}{c} D_z + \frac{4\pi}{c} J_z^e. \quad (103c)$$

Выразив магнитные поля из (103a) и (103b)

$$B_\rho = \frac{-ic}{\omega \rho} \frac{\partial E_z}{\partial \varphi} + \frac{4\pi}{i\omega} J_\rho^m, \quad (104a)$$

$$B_\varphi = \frac{ic}{\omega} \frac{\partial E_z}{\partial \rho} + \frac{4\pi}{i\omega} J_\varphi^m, \quad (104b)$$

мы получаем неоднородное уравнение Гельмгольца для  $E_z$ :

$$\left( \frac{1}{\rho} \frac{\partial}{\partial \rho} \rho \frac{\partial}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2}{\partial \varphi^2} + \frac{\omega^2}{c^2} \varepsilon_1 \mu_1 \right) E_z = F \quad (105)$$

с правой частью

$$F = \frac{4\pi}{c} \left( -i \frac{\omega \mu_1}{c} J_z^e + \frac{1}{\rho} \frac{\partial}{\partial \rho} \rho J_\varphi^m - \frac{1}{\rho} \frac{\partial}{\partial \varphi} J_\rho^m \right). \quad (106)$$

В областях без тока, то есть при  $F = 0$ , уравнение становится однородным и для решений  $\sim e^{in\varphi}$  принимает вид

$$\left( \rho^2 \frac{\partial^2}{\partial \rho^2} + \rho \frac{\partial}{\partial \rho} - n^2 + \rho^2 k_1^2 \right) E_z = 0, \quad k_1^2 = \frac{\omega^2}{c^2} \varepsilon_1 \mu_1. \quad (107)$$

Решение уравнения (107) можно записать в виде

$$E_z(\rho) = AJ_n(k_1\rho) + BY_n(k_1\rho) \quad \text{или} \quad E_z(\rho) = AH_n^{(1)}(k_1\rho) + BH_n^{(2)}(k_1\rho), \quad (108)$$

где  $J_n(z)$  – функции Бесселя первого рода,  $Y_n(z)$  – функции Бесселя второго рода (функции Неймана),  $H_n^{(1,2)}(z) = J_n(z) \pm iY_n(z)$  – функции Ганкеля первого и второго рода,  $A, B$  – произвольные постоянные коэффициенты. При выборе записи решения надо руководствоваться поведением соответствующих функций и граничными условиями (при  $\rho \rightarrow \infty$  или  $\rho \rightarrow 0$ ). Например, при больших значениях аргумента функция Ганкеля  $H_n^{(1)}$  описывает волну, бегущую в бесконечность, а функция  $H_n^{(2)}$  наоборот:

$$H_n^{(1)}(z) \sim \sqrt{\frac{2}{\pi z}} e^{i(z - n\pi/2 - \pi/4)}, \quad H_n^{(2)}(z) \sim \sqrt{\frac{2}{\pi z}} e^{-i(z - n\pi/2 - \pi/4)}. \quad (109)$$

Также надо учитывать, что  $Y_n(z) \rightarrow \infty$  при  $z \rightarrow 0$ , в частности:

$$Y_0(z) \approx \frac{2}{\pi} \left( \ln \frac{z}{2} + \gamma \right), \quad Y_1(z) \approx -\frac{2}{\pi z}. \quad (110)$$

Давайте начнём с нахождения  $G_z^e(x, y; 0, 0) = G_z^e(x, y)$ , то есть поля, возбуждаемого точечным током в начале координат  $\mathbf{J}^e(x, y) = \hat{\mathbf{z}} j_0 \delta(x) \delta(y)$ . Из соображений симметрии и общего вида решения (108) в однородной области понятно, что такой ток будет возбуждать поле вида

$$E_z(\rho, \varphi) = AH_0^{(1)}(k_1\rho), \quad (111)$$

где  $A$  – некоторая постоянная, которую мы найдём, интегрируя уравнение (96b) по контуру вблизи тока. Магнитное поле  $H_\varphi$  вблизи источника тока имеет вид

$$H_\varphi = \frac{ic}{\omega\mu_1} \frac{\partial E_z}{\partial \rho} = \frac{ic}{\omega\mu_1} \frac{\partial}{\partial \rho} AH_0^{(1)}(k_1\rho) \approx -A \frac{c}{\omega\mu_1} \frac{\partial}{\partial \rho} Y_0(k_1\rho) \approx -A \frac{2c}{\pi\omega\mu_1\rho}, \quad (112)$$

где мы использовали асимптотическое представление (110) функции  $Y_0(z)$  при  $z \rightarrow 0$ . С другой стороны, интегрируя (96b) мы получим поле

$$H_\varphi \approx \frac{2j_0}{c\rho}. \quad (113)$$

Сравнивая эти результаты, получим значение коэффициента  $A$ :

$$A = -j_0 \frac{\pi\omega\mu_1}{c^2}. \quad (114)$$

Это значение полностью определяет функцию Грина, то есть отклик  $E_z(x, y)$  при  $j_0 = 1$ :

$$G_z^e(x, y) = -\frac{\pi\omega\mu_1}{c^2} H_0^{(1)}(k_1\rho). \quad (115)$$

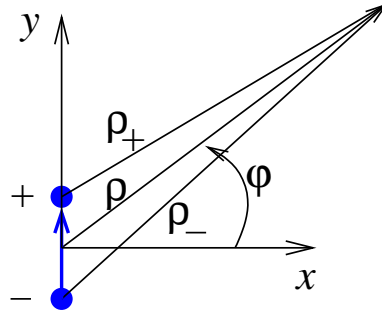


Рис. 29: Излучающий диполь в 2D случае.

Получим теперь поле  $G_y^m(x, y; 0, 0) = G_y^m(x, y)$ , которое возбуждается токами  $\mathbf{J}^m(x, y) = \hat{\mathbf{y}}j_0\delta(x)\delta(y)$ . Такой источник – это диполь и поэтому решение можно записать в следующем виде

$$E_z(\rho, \varphi) = AH_1^{(1)}(k_1\rho) \cos \varphi, \quad (116)$$

где  $A$  – некоторая постоянная, которую можно найти, сравнивая поля вблизи источника тока с полями магнитного диполя. Магнитные поля при  $\rho \rightarrow 0$  можно найти используя уравнения (104):

$$E_z(\rho, \varphi) \approx -A \frac{2i}{\pi k_1 \rho} \cos \varphi, \quad (117a)$$

$$B_\rho(\rho, \varphi) \approx A \frac{2c}{\pi \rho^2 k_1 \omega} \sin \varphi, \quad (117b)$$

$$B_\varphi(\rho, \varphi) \approx -A \frac{2c}{\pi \rho^2 k_1 \omega} \cos \varphi. \quad (117c)$$

Для нахождения  $A$  получим формулы для магнитного диполя и сравним их с выражениями (117). Используя

$$\operatorname{div} \mathbf{B} = 4\pi \rho^m, \quad (118)$$

где  $\rho^m$  – магнитный заряд, получаем, что точечный магнитный заряд  $q$  создает поле

$$B_\rho = \frac{2q}{\rho}. \quad (119)$$

Используя рис. 29, запишем

$$\rho_+ = \rho - d/2, \quad \rho_- = \rho + d/2, \quad (120)$$

а значит поле магнитного диполя имеет вид

$$\mathbf{B} = 2q \left( \frac{\rho_+}{\rho_+^2} - \frac{\rho_-}{\rho_-^2} \right) = \frac{2}{\rho^2} (2p\hat{\rho} \sin \varphi - \mathbf{p}). \quad (121)$$

Теперь можно найти компоненты этого поля

$$B_\rho(\rho, \varphi) = \frac{2p}{\rho^2} \sin \varphi, \quad B_\varphi(\rho, \varphi) = -\frac{2p}{\rho^2} \cos \varphi. \quad (122)$$

Сравнивая (122) со (117b), (117c) с учётом  $j_0 = -i\omega p$ , находим

$$A = \frac{i\pi k_1}{c} j_0, \quad (123)$$

что даёт

$$G_y^m(x, y) = \frac{i\pi k_1}{c} H_1^{(1)}(k_1 \rho) \cos \varphi. \quad (124)$$

Используя поворот координат, можно найти и  $G_x^m(x, y, 0, 0)$  из (124). Таким образом, мы получили функции Грина для источников в начале координат:

$$G_z^e(x, y) = -\frac{\pi\omega\mu_1}{c^2} H_0^{(1)}(k_1 \rho), \quad (125a)$$

$$G_x^m(x, y) = -\frac{i\pi k_1}{c} H_1^{(1)}(k_1 \rho) \sin \varphi, \quad (125b)$$

$$G_y^m(x, y) = \frac{i\pi k_1}{c} H_1^{(1)}(k_1 \rho) \cos \varphi. \quad (125c)$$



# Задачи

## 1. Основы 1D моделирования

- 1.1 Промоделировать распространение импульса, если в качестве начального условия задается его электрическое поле в момент  $t = 0$ , но используется нулевое значение магнитного поля при  $t = -\Delta t/2$ . Объяснить результат.
- 1.2 Рассчитать, как будет отражаться импульс с полями  $E_y$ ,  $H_z$  от границы области моделирования, если использовать на ней граничное условие  $H_z = 0$ . Объяснить результат.
- 1.3 Запустить два импульса (ширина  $\tau = 5$  фс, расстояние в начальный момент  $50 \mu\text{m}$ ), распространяющиеся навстречу друг другу. Что происходит при их прохождении друг через друга?
- 1.4 Создать монохроматическую волну в резонаторе с граничными условиями  $E_y = 0$ . Достаточно взять несколько пространственных периодов. Убедиться, что происходят периодические колебания во времени при выборе временного шага с  $\xi < 1$ . Найти, как будет расти поле при  $\xi = 1.01$ , и сравнить коэффициент роста во времени с аналитическим значением, которое получается из дисперсионного анализа для дискретной среды.
- 1.5 Создать начальное распределение в момент  $t = 0$  для поля  $E_y$  такое, чтобы поле было равно 1 в одном узле сетки и 0 во всех других. Для поля  $H_z$  в момент  $t = -\Delta t/2$  взять 1 в узле, слева от узла, где задано ненулевое поле  $E_y$  и 0 во всех других. Как распространяется такой импульс при  $\xi = c\Delta t/\Delta x = 1$ ? Что получается при  $\xi = 0.9$ ? Объяснить результат.
- 1.6 Создать в вакууме начальный импульс с длиной волны  $\lambda_0 = 600$  нм и длительностью  $\tau = 5$  фс. Найти численно и построить коэффициент отражения от полупространства с  $\varepsilon = 2$  в области частот, задаваемой начальным импульсом. Сравнить с аналитической зависимостью.
- 1.7 Выбрать параметры импульса и промоделировать его распространение, чтобы получить значительное искажение на больших расстояниях. Используя полученные значения полей, рассчитать с помощью преобразования Фурье численную дисперсию в области частот, задаваемой импульсом. Проверить, зависит ли дисперсия от пройденного расстояния. Сравнить полученную дисперсию с ожидаемым аналитическим результатом при выбранных величинах  $\Delta x$  и  $\Delta t$ .

1.8 Промоделировать распространение импульса (с длиной волны  $\lambda_0 = 600$  нм и длительностью  $\tau = 5$  фс) в свободном пространстве, используя методику разделения области моделирования на область полного поля и область рассеянного поля. Продемонстрировать правильность работы алгоритма, построив распределения полей в различные моменты времени: при входе в область полного поля и при выходе из нее.

## 2. Поглощающие слои в 1D

2.1 Создать в вакууме начальный импульс с длиной волны  $\lambda_0 = 600$  нм и длительностью  $\tau = 5$  фс. Найти численно и построить коэффициент отражения от полупространства с  $\varepsilon = 2$  при наличии поглощения с  $\sigma = 2\omega_0$ , где  $\omega_0 = 2\pi c/\lambda_0$ . Сравнить с аналитической зависимостью.

2.2 Создать в вакууме начальный импульс с длиной волны  $\lambda_0 = 600$  нм и длительностью  $\tau = 5$  фс. Найти частотную зависимость коэффициента отражения и коэффициента прохождения, если имеется слой с проницаемостью  $\varepsilon = 4$  и толщиной  $L = 1500$  нм. Сравнить результаты с аналитическими.

2.3 Создать в вакууме начальный импульс с длиной волны  $\lambda_0 = 600$  нм и длительностью  $\tau = 5$  фс. На границе области моделирования создать идеально согласованный поглощающий слой, в котором создаются электрические и магнитные токи. Проводимости считать однородными во всем слое. Для нескольких значений параметра  $\eta = 2\pi\sigma\Delta t$  найти и построить зависимость коэффициента отражения от толщины слоя (количества ячеек). Отражение можно оценивать используя максимальное значение отраженного импульса. Сделать выводы о минимальной полученной величине отражения и оптимальном выборе параметра  $\eta$  и толщины слоя для ее достижения.

## 3. 2D моделирование

3.1 Написать программу 2D моделирования для ТЕ поляризации ( $E_z$ ,  $H_x$ ,  $H_y$ ) используя нулевые граничные условия для  $E_z$ . Продемонстрировать динамику поля  $E_z$  (распространение поля, отражение от границ) при возбуждении точечным источником тока в свободном пространстве.

3.2 В разработанную программу 2D моделирования добавить поглощающий слой (PML) (с монотонным ростом проводимости) около всех границ. Промоделировать отражение волн при нескольких значениях (2–3) параметров слоя. (Конкретные результаты можно не приводить, но указать, какие значения использовались.) Выбрать наиболее подходящий набор параметров и продемонстрировать эффективность поглощения в слое.

- 3.3 Оценить эффективность использования РМЛ для снижения отражения в 2D моделировании. Для этого сравнить отражение в двух случаях. Случай 1: используя РМЛ с наиболее подходящими параметрами из предыдущего задания (2D моделирование, ТЕ поляризация). Случай 2: используя обычный изотропный слой с электрической проводимостью с такой же пространственной зависимостью для  $\sigma$ , как и для РМЛ. Сделать выводы об эффективности РМЛ по сравнению с обычным изотропным слоем с электрической проводимостью.
- 3.4 Разработать программу 2D моделирования для ТЕ поляризации ( $E_z, H_x, H_y$ ) с использованием поглощающего слоя на границе и разделением области моделирования на области полного и рассеянного полей. Продемонстрировать правильность работы программы: ввод плоской волны в область полного поля через левую границу и ее вывод через правую, поглощение остаточного поля слоем.
- 3.5 Используя программу двумерного моделирования, продемонстрировать рассеяние импульса ( $\lambda_0 = 600$  нм,  $\tau = 5$  фс) с плоским фронтом на диэлектрическом цилиндре (ось вдоль  $z$ ) с  $\varepsilon = 2$ ,  $R = 1200$  нм.
- 3.6 Найти численно сечение рассеяния для цилиндра ( $\varepsilon = 2$ ,  $R = 1200$  нм) в области частот, задаваемой падающим импульсом ( $\lambda_0 = 600$  нм,  $\tau = 5$  фс). Построить сечение рассеяния, нормирую его на геометрическое сечение  $2R$ , как функцию параметра размера  $kR = \omega R/c$ . Сравнить с аналитическим решением, представленным на рис. 30.

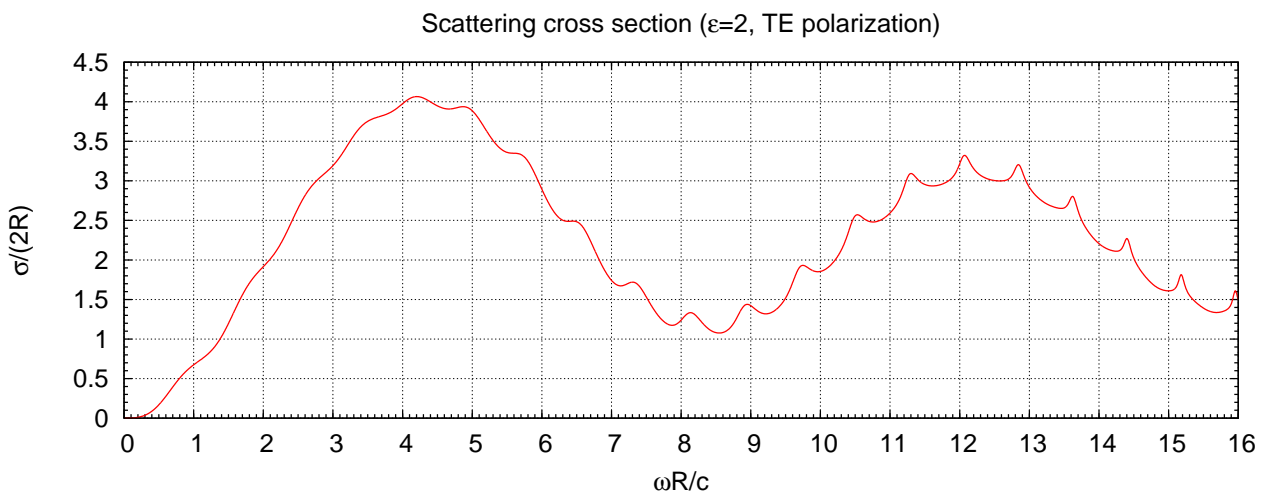


Рис. 30: Зависимость сечения рассеяния  $\sigma$  при падении ТЕ волны в вакууме на цилиндр с  $\varepsilon = 2$  от безразмерного параметра  $\omega R/c$ , где  $\omega$  – частота,  $R$  – радиус цилиндра,  $c$  – скорость света в вакууме.

## Об авторе

А. В. Маслов закончил радиофизический факультет ННГУ с получением степени бакалавра физики (с отличием) в 1995 г. В 1995–2001 гг. обучался в магистратуре и аспирантуре физического факультета Университета штата Вашингтон, г. Пульман, штат Вашингтон, США (Washington State University, Pullman, Washington, USA). Получил степени магистра физики в 1997 и доктора философии (Ph.D.) по физике в 2001. Тема диссертации “Оптические свойства квантовой ямы, находящейся под воздействием терагерцового электрического поля”. В 2001–2002 гг. работал научным сотрудником на факультете электронного и компьютерного инжиниринга в Технологическом университете штата Джорджия, г. Атланта (School of Electrical and Computer Engineering, Georgia Institute of Technology). В 2002–2006 гг. работал в отделе нанотехнологий научно-исследовательского центра НАСА Эймс (NASA Ames Research Center) в Кремниевой долине США, где разрабатывал теоретические модели и программы численного моделирования перспективных малоразмерных лазеров на основе полупроводниковых нанопроволок. Впоследствии работал старшим научным сотрудником в корпорации Кэнон США (Canon USA), где был ведущим разработчиком различных программ компьютерного моделирования: 3D моделирования полупроводниковых лазеров на квантовых ямах на основе GaN, оптоволоконных лазеров и параметрических усилителей, цифровой интерференционной микроскопии и других. С 2014 г. работает в ННГУ. Автор более 100 научных работ, из них более 80 статей в рецензируемых журналах, в области фотоники, физики полупроводниковых лазеров, низкоразмерных полупроводниковых наноструктур, электродинамики нестационарных сред, генерации терагерцового излучения, магнитных метаматериалов, оптического сверхразрешения, оптомеханики. Выступал в качестве приглашённого докладчика на ведущих международных научных конференциях, а также на семинарах в университетах США и Европы.

Алексей Владимирович Маслов

# МОДЕЛИРОВАНИЕ ВОЛНОВЫХ ПРОЦЕССОВ В ЭЛЕКТРОДИНАМИКЕ

Учебное пособие

Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский Нижегородский  
государственный университет им. Н.И. Лобачевского»  
603950, Нижний Новгород, пр. Гагарина, 23

Подписано в печать . . . 2019. Формат 60x84 1/16.

Бумага офсетная. Печать офсетная.

Усл. Печ. л. 4,8. Заказ № . Тираж 100 экз.

Отпечатано в типографии Национального исследовательского  
Нижегородского государственного университета им. Н. И. Лобачевского  
603000, г. Нижний Новгород, ул. Большая Покровская, 37