

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С. П. КОРОЛЁВА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

С.Н. Хонина, С.Г. Волотовский

**МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ СВЕТОВЫХ ПОЛЕЙ
С ИСПОЛЬЗОВАНИЕМ ДИФРАКЦИОННЫХ ИНТЕГРАЛОВ**

*Электронное учебно-методическое пособие к выполнению
самостоятельных работ по курсу
«Современные методы и алгоритмы решения сложных задач
на суперкомпьютерах»*

САМАРА
2010

УДК 535.42
ББК 22.343

Авторы: **Хонина Светлана Николаевна,**
Волотовский Сергей Геннадьевич

Рецензенты: доктор физ.- мат. наук Д.Л. Головашкин,
кандидат физ.- мат. наук П.Г. Серафимович

Рассмотрено моделирование распространения световых полей с использованием дифракционных интегралов в рамках строгой теории дифракции, а также в различных приближениях и аппроксимациях, с использованием высокопроизводительных вычислительных средств.

Учебное пособие рекомендуется для магистрантов по курсу лекций «Современные методы и алгоритмы решения сложных задач на суперкомпьютерах» в рамках магистерской программы «Прикладная математика и информатика» по направлению 010400.68, а также для студентов других направлений, в программу которых входит исследование дифракционных процессов, и может быть полезно при выполнении курсовых работ, дипломных проектов и подготовке к экзаменам.

Подготовлено на кафедре технической кибернетики.

© Самарский государственный
аэрокосмический университет, 2010

Оглавление

Введение	4
1. Скалярные операторы распространения	5
2. Примеры расчета в рамках скалярной теории	7
2.1. Дифракция плоской волны на круглой апертуре	7
2.2. Дифракция вихревого пучка на круглой апертуре	10
2.3. Дифракция плоской волны на бинарном осесимметричном аксиконе	11
3. Векторные операторы распространения	13
3.1. Векторное интегральное преобразование Релея-Зоммерфельда	13
3.2. Векторное представление через плоские волны	14
4. Примеры расчета с использованием векторных интегралов	15
4.1. Дифракция плоской волны на круглой апертуре	15
4.2. Дифракция вихревого пучка на круглой апертуре	17
5. Векторное представление через плоские волны в модификации Мансурипура	18
6. Учёт в методе разложения по плоским волнам коэффициентов пропускания Френеля	21
6.1. Представление через ТЕ- и ТМ-компоненты	21
6.2. Расчет коэффициентов Френеля	22
6.3. Дифракция плоской волны на бинарном би-аксиконе	24
7. Описание программного обеспечения FieldPropagators	27
7.1. Структура языка	27
7.1.1. Пакеты параметров	28
7.1.2. Пакеты несовместимых параметров	29
7.2. Варианты запуска программы	29
7.2.1. Запуск на расчет под Win*	29
7.2.2. Запуск на расчет под Linux	29
7.2.3. Запуск на расчет в фоновом режиме на кластере	30
7.2.4. Выдача справочной информации	30
7.3. Описание языка параметров	32
7.3.1. Правила задания параметров	32
7.3.2. Типы параметров	32
8. Примеры использования программного обеспечения FieldPropagators	47
8.1 Пример с одним пакет параметров	47
8.2 Пример с несколькими входными полями	48
8.3 Пример с несколькими вариантами расчета	50
8.4 Пример сравнения двух скалярных пропагаторов	52
8.5 Пример сравнения скалярного и векторного пропагаторов	53
8.6. Пример вывода топологии продольных сечений	55
9. Задания к выполнению самостоятельных работ	56
Литература	59

Введение

В связи с уменьшением размеров оптических устройств большое внимание в последнее время уделяется описанию непараксиального распространения световых полей [1–9] и разработке алгоритмов моделирования такого распространения [10–21].

Дифракция на круглой апертуре плоской волны является тестовым примером для разрабатываемых алгоритмов как случай, имеющий аналитическое решение на оптической оси [22, 23], и исследовалась с различной степенью точности во многих работах [11, 24–29].

Непараксиальная скалярная модель, основанная на теории Рэлея-Зоммерфельда [30], позволяет получать согласующиеся с экспериментами результаты на очень близких расстояниях от апертуры [31–34]. В частности, в работе [33] на примере дифракции на цилиндрических тонких фазовых объектах (что практически соответствует скалярной задаче дифракции на щели) было показано, что в зависимости от фазового набега различные типы дифракционных интегралов – Кирхгофа, Рэлея-Зоммерфельда I или II типа – позволяют получить корректные результаты на расстоянии всего полдлины волны от объекта. На расстоянии более двух длин волн все типы интегралов дают одинаковый результат.

Однако при уменьшении поперечных размеров светового поля (или его характерных деталей) до размеров порядка длины волны также необходимо учитывать векторный характер светового поля. В этом случае применяют векторный вариант интегралов Рэлея-Зоммерфельда [35, 2, 3, 7, 12, 27–29] или метод разложения по плоским волнам [36, 37, 4, 20, 29].

При численной реализации интегралов Рэлея-Зоммерфельда сложно воспользоваться какой-либо симметрией световых полей, поэтому для упрощения расчетов или получения аналитических выражений приходится прибегать к различным аппроксимациям [12, 24, 37, 38]. Однако, как показано в работе [39], использование даже непараксиальных аппроксимаций в ближней зоне является некорректным, поэтому для расчетов в этой зоне необходимо использовать точные операторы распространения, а для ускорения расчетов применять высокопроизводительные технологии.

При использовании метода разложения по плоским волнам возможно применение алгоритма быстрого преобразования Фурье (БПФ) [10, 17], что значительно сокращает время расчета. Однако использование алгоритма БПФ имеет свои недостатки, связанные с фиксированной дискретностью сигналов на входе и выходе, а также возможностью вычислять только поперечные распределения. При необходимости получения продольных (вдоль оптической оси) распределений использовать алгоритм, основанный на БПФ,

нецелесообразно: зависимость метрических размеров от z требует пересчета (интерполяции) отсчетов в каждой плоскости. А при необходимости получения значений на некоторой кривой, не лежащей в поперечной плоскости, преимущество БПФ по скорости теряется.

В данном пособии предлагается к рассмотрению программное обеспечение **FieldPropagators**, предназначенное для расчета поля в заданных областях различными операторами распространения в пространстве с постоянным показателем преломления.

Для удобства использования на кластере программа выполнена как консольное приложение. Входные данные задаются в текстовом файле на языке описания задания. При запуске программы без параметров выдается краткое описание допустимых параметров.

В результате работы программы формируются:

- файл протокола (содержит входные параметры, имена выходных файлов и дополнительную информацию),
- выходные файлы изображений (в формате BMP и DAT),
- выходные файлы графиков (в текстовом формате),
- сообщения об ошибках, если таковые были.

Программное обеспечение **FieldPropagators** организовано так, что его удобно использовать для сравнения действия различных операторов распространения. Это позволит студентам оценить границы применимости различных приближений и аппроксимаций, используемых при моделировании дифракционных процессов.

1. Скалярные операторы распространения

В данном разделе рассматриваются скалярные операторы распространения, которые описывают распространение монохроматического поля с длиной волны λ , заданного в плоскости $z=0$ комплексным распределением $E_0(x, y)$ в пространстве с постоянным показателем преломления n .

Везде далее волновое число определяется по формуле:

$$k = \frac{\omega}{c} \sqrt{\mu\epsilon} = \frac{2\pi}{\lambda} n, \quad (1)$$

где ω - частота излучения, c – скорость света в вакууме, ϵ и μ диэлектрическая и магнитная проницаемости среды, соответственно.

Преобразование Фурье соответствует распределению в фокальной плоскости сферической линзы:

$$G(u, v) = \frac{k}{f} \iint_{\Sigma} E_0(x, y) \exp \left[-\frac{ik}{f} (xu + yv) \right] dx dy \quad (2)$$

где f – фокусное расстояние, Σ – область задания входного поля.

Преобразование Френеля описывает параксиальное (близкое к оптической оси) распространение на расстояние z .

$$G(u, v, z) = -\frac{ik}{2\pi z} \exp(ikz) \iint_{\Sigma} E_0(x, y) \exp \left[\frac{ik}{2z} ((x-u)^2 + (y-v)^2) \right] dx dy \quad (3)$$

где z – расстояние от входной плоскости.

Интегральное преобразование Релея-Зоммерфельда первого типа точно (без каких-либо аппроксимаций) описывает распространение на расстояние z [30, 40]:

$$E(u, v, z) = -\frac{z}{2\pi} \iint_{\Sigma} E_0(x, y) \frac{e^{ik\ell}}{\ell^2} \left(ik - \frac{1}{\ell} \right) dx dy, \quad (4)$$

где $\ell = \sqrt{(u-x)^2 + (v-y)^2 + z^2}$

Разложение по плоским волнам точно (без каких-либо аппроксимаций) описывает распространение на расстояние z [41]:

$$E(x, y, z) = \frac{1}{\lambda^2} \iint_{\Sigma_s} F(\xi, \eta) \exp \left[ikz \sqrt{1 - (\xi^2 + \eta^2)} \right] \exp [ik(\xi x + \eta y)] d\xi d\eta, \quad (5)$$

где $F(\xi, \eta) = \iint_{\Sigma} E_0(x, y) \exp [-ik(\xi x + \eta y)] dx dy \quad (6)$

где $F(\xi, \eta)$ – Фурье-спектр входного электрического поля $E_0(x, y)$, определенного в области Σ .

Спектр $F(\xi, \eta)$ рассматривается в области Σ_s : $\sigma_1 \leq \sqrt{\xi^2 + \eta^2} \leq \sigma_2$.

Распространяющимся волнам соответствуют пространственные частоты, расположенные в круге радиусом $\sigma_0 \leq 1$. Чтобы учесть также и затухающие волны, вносящие свой вклад на расстояниях меньше длины волны, необходимо увеличивать радиус учитываемых пространственных частот до некоторого значения $\sigma_z > 1$, зависящего от расстояния z от апертуры. При $\sigma_1 = 0, \sigma_2 = 1$ учитываются только распространяющиеся волны, при $\sigma_1 = 1, \sigma_2 > 1$ учитываются только затухающие волны.

Если значения пространственной переменной лежат в диапазоне $x \in [-R_x, R_x]$ и дискретизация в объектной плоскости: $h_x = \frac{2R_x}{N}$, то с учетом условия Найквиста в спектральной плоскости частоты лежат в диапазоне $\xi \in [-R_\xi, R_\xi]$, $R_\xi = \frac{\lambda}{2h_x}$.

Поэтому $\max|\xi| = R_\xi$ и $\max|\eta| = R_\eta$. Если этот диапазон частот оказывается меньше необходимого, т.е. $\sqrt{R_\xi^2 + R_\eta^2} < \sigma_2$, то необходимо уменьшать шаг дискретизации входного поля.

Выражения (4) и (5)-(6) можно свести друг к другу, однако в последнем случае возможно применение различных быстрых алгоритмов, в том числе БПФ [10, 17], что значительно сокращает время расчета, несмотря на удвоенное по сравнению (4) интегрирование.

Применение алгоритма БПФ имеет свои недостатки, связанные с фиксированной дискретностью сигналов на входе и выходе, а также возможностью вычислять только поперечные распределения. Кроме того, в работе [10] было указано на проблемы, возникающие при наличии наклона или смещения падающего пучка, для решения которых требуются дополнительные преобразования.

2. Примеры расчета в рамках скалярной теории

2.1. Дифракция плоской волны на круглой апертуре

Дифракция на круглой апертуре плоской волны является тестовым примером для разрабатываемых алгоритмов как случай, имеющий аналитическое решение на оптической оси [22, 23], и исследовалась с различной степенью точности во многих работах [11, 24-29]. Таким образом, данный пример может использоваться для начальной оценки точности рассматриваемых в данном пособии алгоритмов.

При дифракции плоской волны на круглой апертуре в скалярном случае известно аналитическое решение для значений интенсивности вдоль оптической оси [11]:

$$I(0, 0, z) = 1 + \frac{z^2}{r_0^2 + z^2} - \frac{2z}{\sqrt{r_0^2 + z^2}} \cos \left\{ k \left(\sqrt{r_0^2 + z^2} - z \right) \right\}, \quad (7)$$

где r_0 – радиус ограничивающей апертуры.

Для сравнения численных результатов с аналитическим выражением (7) были выполнены расчеты осевой интенсивности плоской волны, прошедшей через круглое отверстие радиусом $r_0 = 10\lambda$, на различных расстояниях с помощью интегрального преобразование Рэлея-Зоммерфельда (4) и разложения по плоским волнам (5), (6) при $m=0$.

Для получения значений на оптической оси или в продольном сечении применение алгоритма БПФ неэффективно, т.к. для этого потребуется вычислить столько поперечных сечений, сколько желательно получить значений вдоль оси распространения пучка.

На рис. 1 приведены сравнительные результаты вычисления осевой интенсивности в ближней зоне дифракции, из которых видно, что результаты обоих методов практически полностью совпадают для расстояний больших длины волны. Однако на расстояниях меньших длины волны метод разложения по плоским волнам несколько отличается от аналитического выражения, особенно при $z < 0,1\lambda$. По формулам (5), (6) для получения корректных результатов при $z = 0,1\lambda$ необходимо учитывать пространственные частоты вплоть до $\sigma_0 = 8$, а при $z = 0,01\lambda$ д.б. $\sigma_0 > 79$, что резко увеличивает временные затраты, т.к. в этом случае необходимо также увеличивать дискретизацию входного поля.

Для получения результатов, приведенных на рис. 1 в верхний предел учитываемых спектральных частот был принят $\sigma_0 = 20$, что в совокупности с уменьшением шага дискретизации привело к увеличению расчетного времени и превысило время прямого расчета (4) по алгоритму [21]. Заметим, что для получения корректных результатов в выражении (4) при $z < 0,1\lambda$ понадобилась на порядок меньшая степень дискретизации, чем при использовании формул (5), (6). Таким образом, в данном случае алгоритм, разработанный в [21], оказывается предпочтительным.

На рис. 1г показано внеосевое продольное распределение интенсивности при дифракции плоской волны на круглой апертуре в области $z \in [0,1\lambda, 10\lambda]$, $x \in [-10\lambda, 10\lambda]$, рассчитанное по формулам (5), (6). На расстояниях, больших одной десятой длины волны, алгоритм разложения по плоским волнам, учитывающий радиальную симметрию задачи [39], полностью совпадает с аналитическим решением и значительно опережает по скорости алгоритм [21].

Рассмотренный в [39] быстрый алгоритм удобнее БПФ. Удобство заключается в том, что он позволяет с произвольной дискретизацией (не связанной с бинарной системой) получать распределения на любых поверхностях за малое время. Недостатком является требование наличия радиальной симметрии или вихревой угловой зависимости входного пучка.

Более детально внеосевое распределение интенсивности в поперечных направлении распространения плоскостях на различных расстояниях от апертуры показано в Табл. 1. Среднеквадратичное отклонение в расчетах, получаемых двумя реализованными методами составляет менее 5%.

Таким образом, каждый из методов и их реализаций обладает достоинствами и недостатками. Относительно медленный алгоритм прямого расчета интеграла РЗ играет

важную роль на очень близких расстояниях от апертуры, а также как универсальный по отношению к типу падающей волны и форме апертуры тестовый инструмент. Метод разложения ПВ, реализованный через БПФ является самым быстрым, хотя и требующим много памяти [10], и также универсален. Радиальная реализация метода разложения ПВ [39] характеризуется высокой скоростью расчета, очень экономична в требованиях к свободной памяти и удобна для расчетов распределений на любых поверхностях, но круг решаемых задач ограничен.

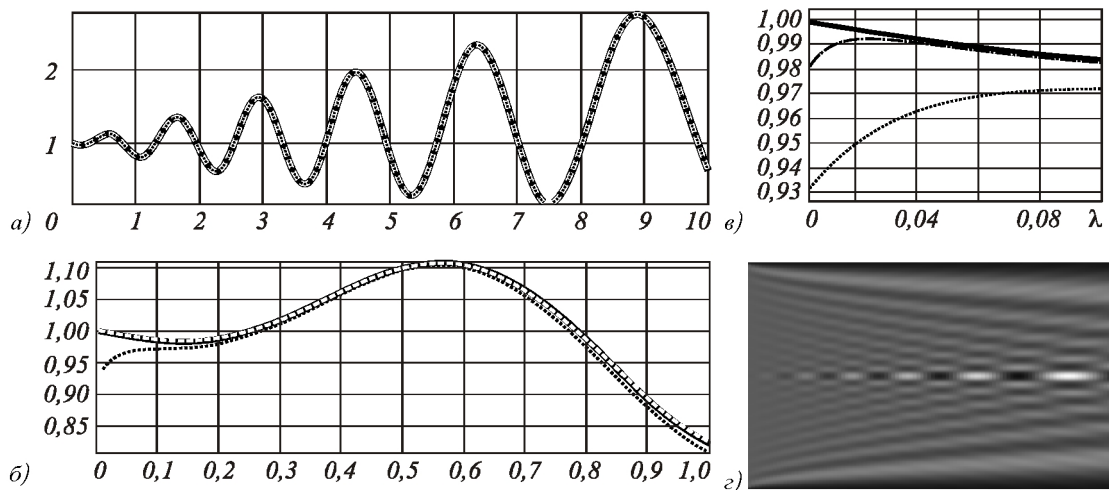
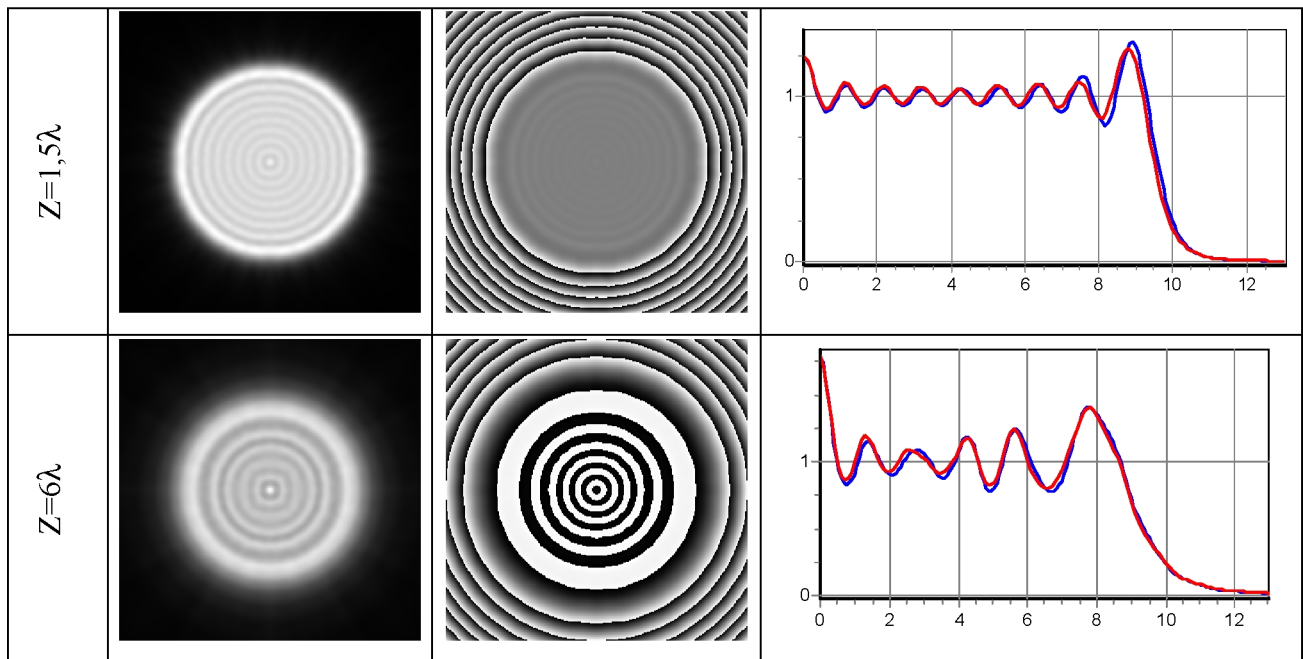


Рис. 1. Сравнение распределения осевой интенсивности, рассчитанной с помощью (I.1) (пунктирная линия) и (I.6) (точечная линия), с аналитическим выражением (I.3) (черный цвет сплошная линия): (а) на отрезке $z \in [0, 1\lambda, 10\lambda]$ при $\sigma_0 = 3$, (б) на отрезке $z \in [0, 0.1\lambda, 1\lambda]$ при $\sigma_0 = 5$ и (в) на отрезке $z \in [0, 0.1\lambda, 0.1\lambda]$ при $\sigma_0 = 5$ (точечная линия) и $\sigma_0 = 20$ (штрихпунктирная линия); (г) фокусирующее поведение круглой апертуры

Таблица 1. Распределение интенсивности в поперечных направлении распространения плоскостях на различных расстояниях от апертуры (поперечная область расчета $30 \times 30 \lambda^2$)

	Амплитуда	Фаза	Сечение интенсивности для РЗ (красный) и ПВ (синий) методов
$Z=0,3\lambda$			



2.2. Дифракция вихревого пучка на круглой апертуре.

Рассмотрим дифракцию вихревого пучка на круглой диафрагме радиуса r_0 :

$$E_0(r, \varphi) = E_0(r) \exp(im\varphi) \text{circ}(r/r_0), \quad (8)$$

где (r, φ) – полярные координаты во входной плоскости, $\text{circ}(r/r_0) = \begin{cases} 1, & r \leq r_0, \\ 0, & r > r_0. \end{cases}$

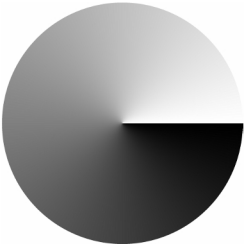
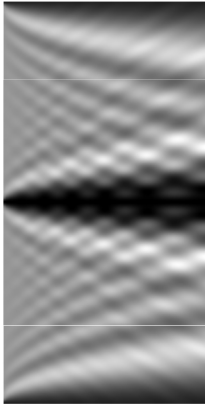
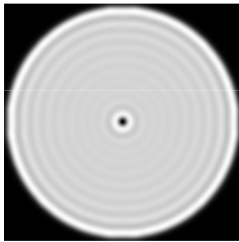
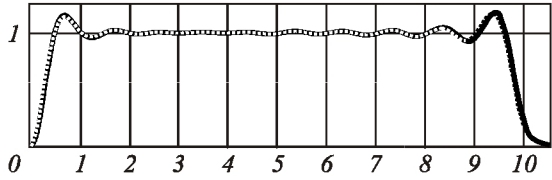
Как правило, дифракция на спиральной фазовой пластинке рассматривается в приближении тонкого оптического элемента, что фактически эквивалентно дифракции пучка с вихревой фазовой составляющей на апертуре того же радиуса соответствующего дифракционного оптического элемента.

При дифракции вихревого пучка с постоянной амплитудой на микроапертуре получить удобные аналитические выражения не удаётся.

В скалярной модели на оптической оси в этом случае будет нулевое значение интенсивности.

В табл. 2 показаны результаты моделирования в рамках скалярной непараксиальной модели дифракции вихревого пучка, имеющего порядок вихря $m=1$, с постоянной амплитудой на круглой апертуре радиусом 10λ в области $z \in [0, 1\lambda, 10\lambda]$, $x \in [-10\lambda, 10\lambda]$.

Таблица 2. Распределение в ближней зоне дифракции на круглой апертуре вихревого пучка ($m=1$) с постоянной амплитудой

Фаза в области апертуры радиусом 10λ ($z=0$)	Распределение интенсивности в области $z \in [0, 1\lambda, 10\lambda]$, $x \in [-10\lambda, 10\lambda]$	Распределение в плоскости $z = 0,3\lambda$, $r \in [0, 10\lambda]$	
		Интенсивность  Фаза 	

На графике в табл.2 приведены сечения интенсивности в плоскости $z = 0,3\lambda$, полученные при использовании формулы (4) (сплошная линия) и (5), (6) (точечная линия).

2.3. Дифракция плоской волны на бинарном осесимметричном аксиконе.

Фазовый дифракционный аксикон имеет комплексную функцию пропускания следующего вида:

$$\tau(r) = \exp(ik\alpha_0 r), \quad (9)$$

где $k = 2\pi/\lambda$ – волновое число, λ – длина волны, α_0 – параметр аксикона, определяющий числовую апертуру аксикона $\alpha_0 = NA$.

Рассмотрим высокоапертурный ($\alpha_0 = 1$) бинарный микроаксикон вида:

$$\tau_c(r) = \exp\{i \arg[\cos(kr)]\}, \quad (10)$$

который будет использоваться для преобразования излучения в дальнем инфракрасном диапазоне $\lambda = 10,6$ мкм, радиусом $R = 2,25\lambda = 23,85$ мкм (см. рис. 3а).

Картина дифракции ограниченной радиусом R плоской волны в приближении тонкого оптического элемента в рамках в скалярной непараксиальной теории описывается выражением:

$$E(\rho, z) = k^2 \int_0^{\sigma_0} P(\sigma) \exp(ikz\sqrt{1-\sigma^2}) J_0(k\sigma\rho) \sigma d\sigma, \quad (11)$$

где

$$P(\sigma) = \int_0^R \tau(r) J_0(k\sigma r) r dr \quad (12)$$

– пространственный спектр, σ_0 – радиус учитываемых пространственных частот.

На рис. 3б показано радиальное сечение амплитуды в спектральной плоскости (12), соответствующее расстоянию от плоскости аксикона (10) $z = 0,5$ мкм (точечная линия) и $z = 7$ мкм (сплошная линия). Как видно из графиков, в первом случае необходимо учитывать более высокие пространственные частоты.

На рис. 3в и 3д показаны поперечные распределения интенсивности в плоскостях $z = 0,5$ мкм и $z = 7$ мкм, соответственно. Видно, что в непосредственной близости от оптического элемента (рис. 3в) распределение интенсивности повторяет структуру элемента.

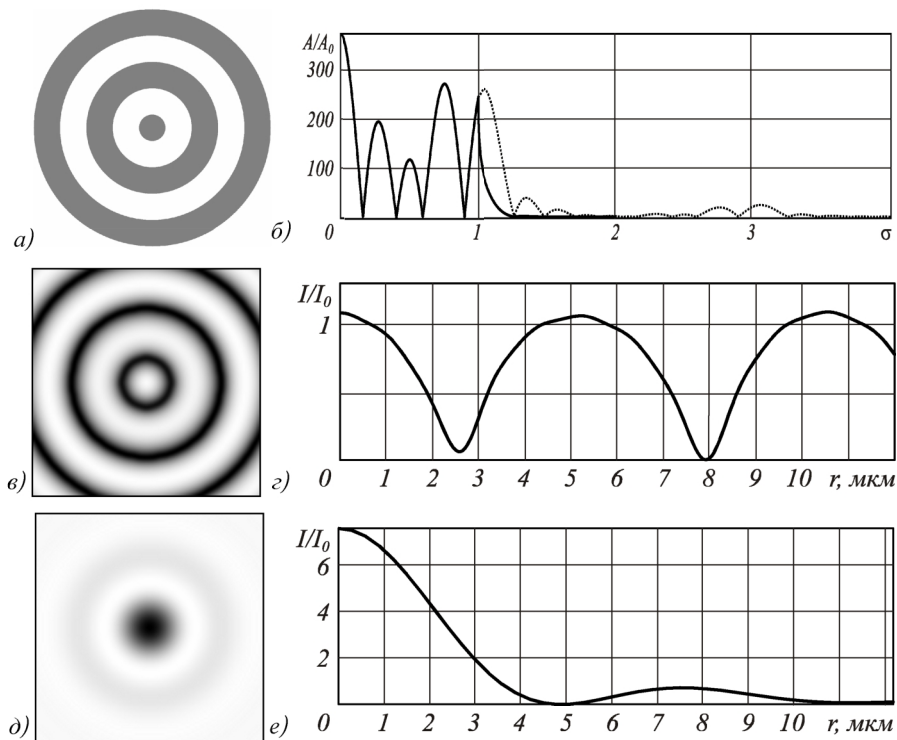


Рис. 3. Моделирование для аксикона (а) с помощью выражений (11)-(12): радиальное сечение амплитуды в спектральной плоскости, соответствующее распространению входного поля на расстояние $z = 0,5$ мкм (точечная линия) и $z = 7$ мкм (сплошная линия) (б), поперечное распределение интенсивности в плоскости $z = 0,5$ мкм (в), радиальное сечение интенсивности в плоскости $z = 0,5$ мкм (г), поперечное распределение интенсивности в плоскости $z = 7$ мкм (д), радиальное сечение интенсивности в плоскости $z = 7$ мкм (е)

Таким образом, в скалярном случае полная ширина центрального светового пятна по полуспаду интенсивности (FWHM), формируемого высокоапертурным бинарным микроаксиконом (10), на расстоянии $z = 0,5$ мкм ($\approx 0,05\lambda$) составляет 3,63 мкм ($\approx 0,34\lambda$), а

площадь по полуспаду интенсивности (НМА) равна $10,35 \text{ мкм}^2 (\approx 0,092\lambda^2)$. На расстоянии $z = 7 \text{ мкм} (\approx 0,66\lambda)$ $\text{FWHM} = 4,44 \text{ мкм} (\approx 0,42\lambda)$, а $\text{НМА} = 15,48 \text{ мкм}^2 (\approx 0,138\lambda^2)$.

Хотя приведенные выше результаты говорят о преодолении дифракционного предела, который соответствует $\text{FWHM} = 0,51\lambda$, минимум для бесселевых пучков ($\text{FWHM} = 0,36\lambda$) удается достичь лишь в непосредственной близости к поверхности элемента – в зоне затухающих волн.

3. Векторные операторы распространения.

В данном разделе рассматриваются векторные операторы распространения, которые необходимо использовать при уменьшении поперечных размеров светового поля (или его характерных деталей) до размеров порядка длины волны.

3.1. Векторное интегральное преобразование Рэлея-Зоммерфельда

Интегральная теорема Рэлея-Зоммерфельда первого типа в векторной форме записывается следующим образом [35, 2]:

$$E_x(u, v, z) = -\frac{z}{2\pi} \iint_{\Sigma} E_{0x}(x, y) \frac{e^{ik\ell}}{\ell^2} \left(ik - \frac{1}{\ell} \right) dx dy, \quad (13a)$$

$$E_y(u, v, z) = -\frac{z}{2\pi} \iint_{\Sigma} E_{0y}(x, y) \frac{e^{ik\ell}}{\ell^2} \left(ik - \frac{1}{\ell} \right) dx dy, \quad (13б)$$

$$E_z(u, v, z) = \frac{1}{2\pi} \iint_{\Sigma} [E_{0x}(x, y)(u-x) + E_{0y}(x, y)(v-y)] \frac{e^{ik\ell}}{\ell^2} \left(ik - \frac{1}{\ell} \right) dx dy, \quad (13в)$$

где $\ell = \sqrt{(u-x)^2 + (v-y)^2 + z^2}$, $E_{0x}(x, y)$, $E_{0y}(x, y)$ – тангенциальные компоненты входного электрического поля, которые также могут быть заданы через коэффициенты поляризации:

$$\begin{pmatrix} E_{0x}(x, y) \\ E_{0y}(x, y) \end{pmatrix} = E_0(x, y) \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad (14)$$

Во всех этих интегралах присутствует выражение для сферической волны $\exp(ik\ell)/\ell$. Как правило, это не позволяет аналитически вычислить интегралы (13). В [43] есть разложение сферической волны в ряд по функциям Бесселя (выражение 8.533), но замена подынтегрального выражения на бесконечный ряд, включающий спецфункции, не облегчает численный расчёт. Таким образом, для прямых вычислений по формулам (13) либо нужно использовать алгоритмы быстрого расчета, например, [21, 39], либо привлекать высокопроизводительные вычислительные средства.

3.2. Векторное представление через плоские волны

Вычисление с помощью выражений (13) является очень ресурсозатратным, а какие-либо аппроксимации расстояния ℓ приводят к существенным погрешностям в ближней зоне дифракции [39].

Векторный вариант оператора распространения, использующего разложение по плоским волнам позволяет учесть радиальную симметрию решаемых задач, а также использовать алгоритм быстрого преобразования Фурье.

В этом случае все компоненты электромагнитного поля на расстоянии z вычисляются с помощью следующих выражений:

$$\mathbf{E}(x, y, z) = \begin{bmatrix} E_x(x, y, z) \\ E_y(x, y, z) \\ E_z(x, y, z) \end{bmatrix} = \frac{1}{\lambda^2} \iint_{\Sigma_s} \mathbf{S}_E(\xi, \eta) \begin{bmatrix} F_x(\xi, \eta) \\ F_y(\xi, \eta) \end{bmatrix} \exp \left[ikz\sqrt{1-(\xi^2 + \eta^2)} \right] \exp [ik(\xi x + \eta y)] d\xi d\eta, \quad (14)$$

$$\mathbf{H}(x, y, z) = \begin{bmatrix} H_x(x, y, z) \\ H_y(x, y, z) \\ H_z(x, y, z) \end{bmatrix} = \frac{1}{\lambda^2} \iint_{\Sigma_s} \mathbf{S}_H(\xi, \eta) \begin{bmatrix} F_x(\xi, \eta) \\ F_y(\xi, \eta) \end{bmatrix} \exp \left[ikz\sqrt{1-(\xi^2 + \eta^2)} \right] \exp [ik(\xi x + \eta y)] d\xi d\eta, \quad (15)$$

$$\begin{bmatrix} F_x(\xi, \eta) \\ F_y(\xi, \eta) \end{bmatrix} = \iint_{\Sigma} \begin{bmatrix} E_x(x, y, 0) \\ E_y(x, y, 0) \end{bmatrix} \exp [-ik(\xi x + \eta y)] dx dy, \quad (16)$$

где $F_x(\xi, \eta)$ и $F_y(\xi, \eta)$ - спектры тангенциальных компонент входного электрического поля $E_x(x, y, 0)$, $E_y(x, y, 0)$ определенного в области Σ .

Матрицы поляризационного преобразования для электрических и магнитных компонент имеют следующий вид:

$$\mathbf{S}_E(\xi, \eta) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{\xi}{\sqrt{1-(\xi^2 + \eta^2)}} & -\frac{\eta}{\sqrt{1-(\xi^2 + \eta^2)}} \end{bmatrix}, \quad (17)$$

$$\mathbf{S}_H(\xi, \eta) = -\sqrt{\frac{\varepsilon}{\mu}} \begin{bmatrix} \frac{\xi\eta}{\sqrt{1-(\xi^2 + \eta^2)}} & \frac{1-\xi^2}{\sqrt{1-(\xi^2 + \eta^2)}} \\ -\frac{1-\eta^2}{\sqrt{1-(\xi^2 + \eta^2)}} & -\frac{\xi\eta}{\sqrt{1-(\xi^2 + \eta^2)}} \\ \xi & -\eta \end{bmatrix}. \quad (18)$$

4. Примеры расчета с использованием векторных интегралов

4.1. Дифракция плоской волны на круглой апертуре

В разделе 2 было показано, что при радиусе апертуры 10λ в скалярном случае рассмотренные алгоритмы РЗ и ПВ практически полностью совпадают. При уменьшении размеров апертуры до сравнимых с длиной волны необходимо учитывать векторный характер светового поля.

Как следует из раздела 3, при линейной x -поляризации падающей на микроапертуру плоской волны дополнительный вклад в суммарную интенсивность, кроме x -компоненты, будет вносить продольная компонента. При вычислении этой компоненты каждый из рассматриваемых алгоритмов также имеет свои особенности. Для оценки точности получаемых результатов в данном разделе используется алгоритм FDTD.

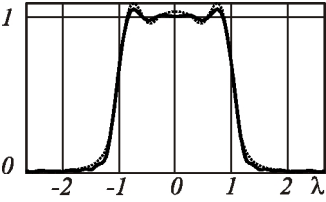
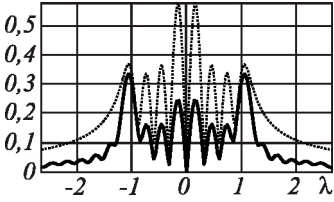
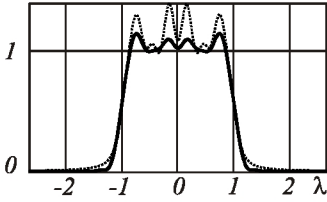
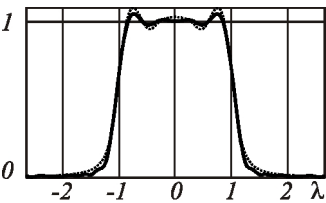
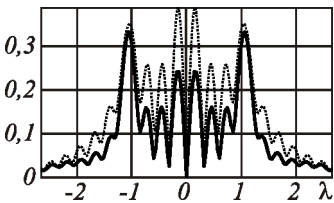
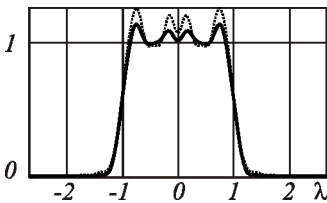
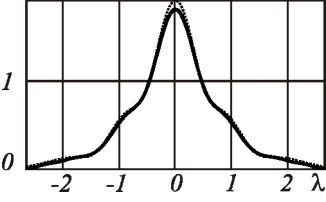
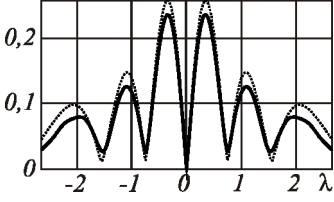
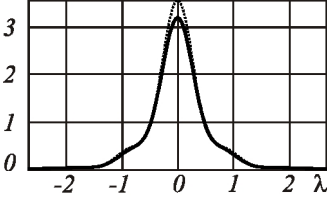
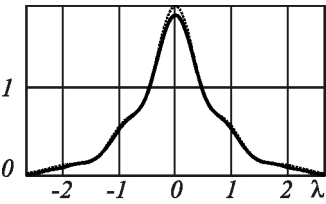
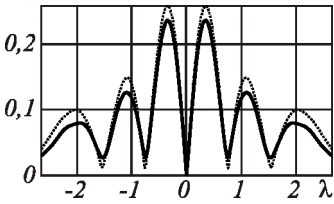
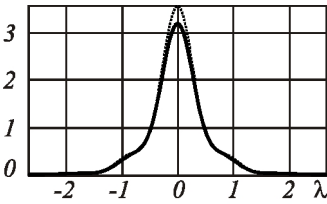
Чтобы согласовать результаты, полученные с помощью интегральных методов, не учитывающих материал и толщину экрана с отверстием, при использовании конечно-разностного временного метода рассматривалось распространение радиально-ограниченной плоской волны в свободном пространстве.

Параметры расчёта при использовании алгоритма FDTD, реализованного в программном продукте R-Soft: длина волны $\lambda = 532$ нм, радиус, ограничивающий плоскую волну на входе $r_0 = 2\lambda$, размер расчётной области $x \in [-5\lambda, 5\lambda]$, $y \in [-5\lambda, 5\lambda]$, $z \in [0, 5\lambda]$. Толщина PML – λ , шаг дискретизации по пространству: $\lambda/15$, шаг дискретизации по времени: $\lambda/(30c)$ (c – скорость света в вакууме), положение плоскости регистрации: $z = 0,3\lambda$ (≈ 160 нм) и $z = 4\lambda$ (≈ 2 мкм).

В табл. 3 приведены сравнительные результаты расчёта дифракции плоской волны при линейной x -поляризации с помощью метода FDTD, дифракционных интегралов РЗ (13) и разложения по плоским волнам (14), (16), (17).

Для алгоритма ПВ на расстоянии $z = 0,3\lambda$ от апертуры учитывались спектральные частоты в радиусе $\sigma_0 = 3$ и применялся обход особенности на основе формулы Ньютона-Лейбница, а на расстоянии $z = 4\lambda$ от апертуры полагалось $\sigma_0 = 1$. В табл. 3 приведены сравнительные графики и среднеквадратичные погрешности δ сечений полученной интенсивности для каждого метода: точечной линией показан график, полученный с помощью дифракционных интегралов, а сплошной – полученный с помощью метода FDTD.

Таблица 3. Сравнительные результаты расчёта дифракции плоской волны при линейной x -поляризации с помощью метода FDTD и дифракционных интегралов на расстоянии $z = 0,3\lambda$ и $z = 4\lambda$

Плоскость	Метод	Сечение амплитуды x -компоненты, $ E_x $	Сечение амплитуды z -компоненты, $ E_z $	Сечение суммарной интенсивности, $ E ^2$
$Z = 0,3\lambda$	P3 (13)	 $\delta=3,4\%, \delta_0=2,9\%$	 $\delta=52,3\%, \delta_0=26,5\%$	 $\delta=11,4\%, \delta_0=7,5\%$
	ПВ (14), (16), (17)	 $\delta=3,4\%, \delta_0=2,9\%$	 $\delta=33,1\%, \delta_0=17,3\%$	 $\delta=6,9\%, \delta_0=4,9\%$
$Z = 4\lambda$	P3 (13)	 $\delta=4,4\%, \delta_0=2,5\%$	 $\delta=13,6\%, \delta_0=7,4\%$	 $\delta=9,1\%, \delta_0=1,8\%$
	ПВ (14), (16), (17)	 $\delta=4,4\%, \delta_0=2,5\%$	 $\delta=13,1\%, \delta_0=6,5\%$	 $\delta=8,9\%, \delta_0=1,8\%$

Как следует из приведённых в табл. 3 результатов, основную погрешность в расчётах вносит именно продольная компонента. Результаты для x -компоненты (что соответствует скалярному случаю) совпадают для обоих алгоритмов и незначительно отличаются от предсказанных FDTD.

Погрешность в вычислении z -компоненты растёт с уменьшением расстояния до апертуры. При этом алгоритм ПВ (14), (16), (17) демонстрирует лучшие результаты, чем алгоритм РЗ [39], на расстояниях, меньших длины волны.

При удалении от апертуры всего на несколько длин волн результаты этих двух алгоритмов фактически совпадают (табл. 3 для $z = 4\lambda$) и отличаются от полученных с помощью FDTD только масштабно, т.е. рассматриваемые в данной работе алгоритмы позволяют получать правильное распределение, но несколько завышенное по амплитуде. Среднеквадратичное отклонение приведённых к одному масштабу распределений δ_0 составляет в обоих случаях менее 2% для суммарной интенсивности.

Такая амплитудная «завышенность» может быть связана с тем, что рассмотренные методы РЗ и ПВ не подразумевают наличия y -компоненты в дифракционной картине изначально x -поляризованного поля. Метод же FDTD показывает, что эта компонента присутствует, хотя её доля энергии в суммарном электрическом поле невелика.

4.2. Дифракция вихревого пучка на круглой апертуре.

При дифракции линейно-поляризованной вдоль оси x плоской волны с вихревой сингулярностью $m = 1$ на круглой апертуре выражения (14), (16), (17) приводят к следующим формулам:

$$E_x(\rho, \theta, z) = -\frac{i^2 z k \exp(ikR)}{R^2} \exp(i\theta) \int_0^{\eta_0} \exp\left(ik \frac{r^2}{2R}\right) J_1\left(\frac{kr\rho}{R}\right) r dr, \quad (19a)$$

$$E_y(\rho, \theta, z) = 0, \quad (19б)$$

$$E_z(\rho, \theta, z) = -\frac{k \exp(ikR)}{R^2} \exp(i\theta) \left\{ \rho \cos \theta \int_0^{\eta_0} \exp\left(ik \frac{r^2}{2R}\right) J_1\left(\frac{kr\rho}{R}\right) r dr - \right. \\ \left. - i \int_0^{\eta_0} \exp\left(ik \frac{r^2}{2R}\right) \left\{ e^{i\theta} J_2\left(\frac{kr\rho}{R}\right) - \left\{ -e^{-i\theta} J_0\left(\frac{kr\rho}{R}\right) \right\} r^2 dr \right\} \right\}. \quad (19в)$$

Таким образом, при $|m|=1$ для линейной x -поляризации легко видеть, что суммарная интенсивность на оптической оси не будет равна нулю, как это предсказывает скалярная теория. Осевые значения поперечной компоненты действительно обнуляются, но продольная компонента остаётся ненулевой:

$$E_z(0, 0, z) = -\frac{ik \exp(ikz)}{2z^2} \int_0^{\eta_0} \exp\left(ik \frac{r^2}{2z}\right) r^2 dr. \quad (20)$$

Из табл. 4 видно, что при расстояниях от апертуры более пяти длин волн оба метода полностью совпадают, в том числе по внеосевым распределениям. В таблице 2 сплошной

линией показаны графики для алгоритма РЗ (13), а точечной – для алгоритма ПВ (14), (16), (17).

Таблица 4. Сравнительные результаты вычислений дифракционных интегралов для дифракции плоской волны с вихревой фазовой сингулярностью $m = 1$ при линейной x -поляризации

Плос- кость	Сечение амплитуды x -компоненты, $ E_x $	Сечение амплитуды z -компоненты, $ E_z $	Сечение суммарной интенсивности, $ E ^2$
$Z = 0,3\lambda$			
$Z = 6\lambda$			

Также различие двух рассмотренных методов может усугубляться наличием в (17) особенности при вычислении z -компоненты, возникающей при $\sigma \rightarrow 1$, т.е. очень близко к плоскости апертуры. Таким образом, проблемы с расчётом продольной компоненты, обнаруженные для плоской волны, также проявляются в случае вихревых пучков.

5. Векторное представление через плоские волны в модификации Мансурипура

В [48] Мансурипуром было рассмотрено спектральное разложение падающей линейно-поляризованной волны как прохождение через виртуальные призмы, которые меняют направление вектора распространения волны. На основе такого подхода получается иная, чем в (17), матрица преобразования спектральных координат:

$$\mathbf{M}_E(\xi, \eta) = \begin{bmatrix} 1 - \frac{\xi^2}{1 + \sqrt{1 - (\xi^2 + \eta^2)}} & -\frac{\xi\eta}{1 + \sqrt{1 - (\xi^2 + \eta^2)}} \\ -\frac{\xi\eta}{1 + \sqrt{1 - (\xi^2 + \eta^2)}} & 1 - \frac{\eta^2}{1 + \sqrt{1 - (\xi^2 + \eta^2)}} \\ -\xi & -\eta \end{bmatrix}. \quad (21)$$

Интересно, что матрица Мансурипура предсказывает появление y -компоненты при распространении, даже если изначально волна была полностью x -поляризованной. Достоинство матрицы (21) также в отсутствии каких-либо особенностей.

Чтобы оценить корректность всех рассмотренных выше алгоритмов в ближней зоне дифракции, было проведено сравнение с результатами алгоритма FDTD.

Параметры расчёта при использовании алгоритма FDTD, реализованного в программном продукте R-Soft: длина волны $\lambda = 532$ нм, радиус круглой апертуры $r_0 = 2\lambda$, порядок вихря $m = 1$, размер расчётной области $x \in [-5\lambda, 5\lambda]$, $y \in [-5\lambda, 5\lambda]$, $z \in [0, 5\lambda]$. Толщина PML – λ , шаг дискретизации по пространству: $\lambda/15$, шаг дискретизации по времени: $\lambda/(30c)$, c – скорость света, положение плоскости экрана для измерений: $z = 0,3\lambda$ (≈ 160 нм) и $z = 4\lambda$ (≈ 2 мкм).

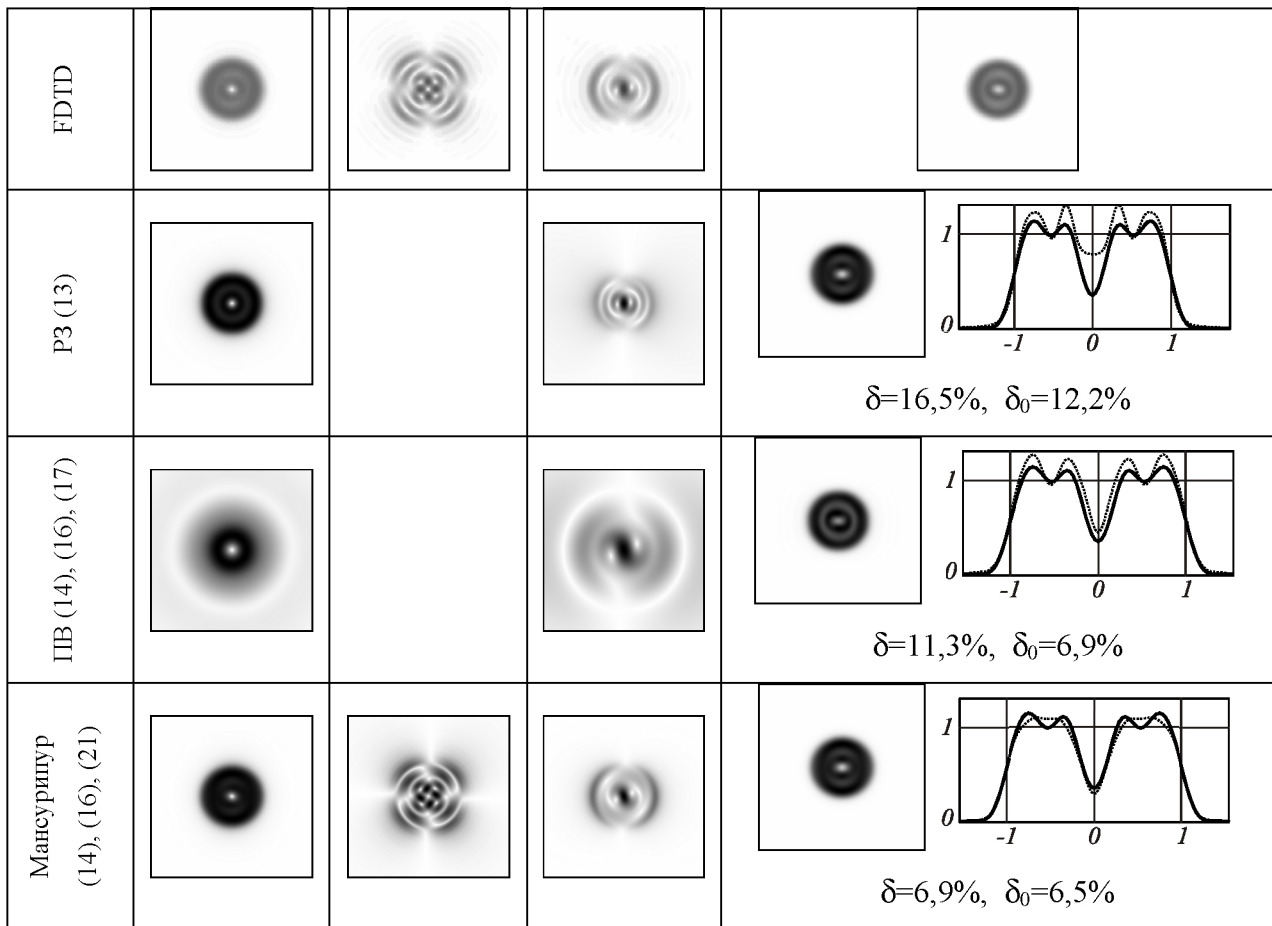
Сравнительные результаты расчёта дифракции плоской волны с вихревой фазовой сингулярностью $m = 1$ при линейной x -поляризации на расстоянии $z = 4\lambda$ от апертуры показали, что наименьшее отличие от результатов, предсказываемых FDTD, демонстрирует алгоритм ПВ с модификацией Мансурипура. В этом случае наблюдается не только наименьшее среднеквадратичное отклонение (менее 1%), но и полное качественное согласование – присутствует y -компонента.

Метод Мансурипура, являясь очень удобным для алгоритмизации, позволяет получать результаты, практически совпадающие с расчётами FDTD, кроме области очень близкой к апертуре. В этой области, как видно из табл. 5, имеется определённое отличие во внеосевом распределении, хотя погрешность остаётся минимальной из всех исследуемых интегральных методов.

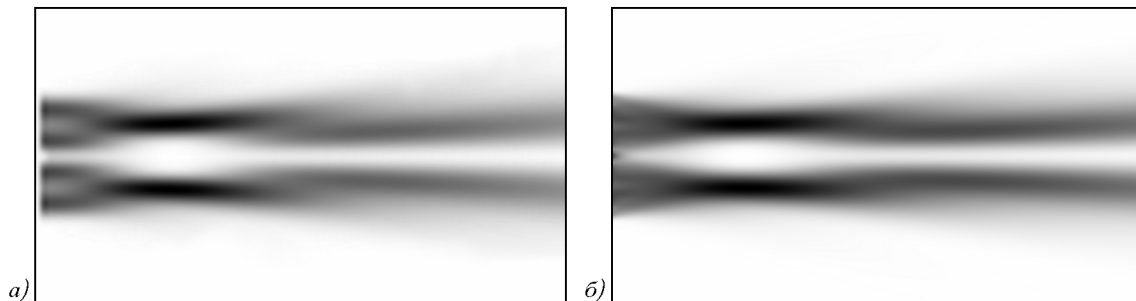
В табл. 5 показаны результаты работы рассматриваемых алгоритмов для области, очень близкой к апертуре, а именно, на расстоянии $z = 0,3\lambda$ от неё: с помощью метода FDTD (первая строка), дифракционных интегралов Релея-Зоммерфельда (13) (вторая строка), разложения по плоским волнам (14), (16), (17) (третья строка) и в модификации Мансурипура (14), (16), (21) (третья строка). Несмотря на рост погрешности в ближней к элементу области (до десятка процентов), имеется хорошее качественное согласование результатов, особенно для метода ПВ. На графиках также точечной линией показаны результаты, полученные с помощью дифракционных интегралов, а сплошной – полученные с помощью метода FDTD.

Таблица 5. Сравнительные результаты расчёта дифракции плоской волны с вихревой фазовой сингулярностью $m = 1$ при линейной x -поляризации с помощью метода FDTD и дифракционных интегралов на расстоянии $Z = 0,3\lambda$ (160 нм)

Метод, время расчёта	$ E_x $	$ E_y $	$ E_z $	$ E ^2$
----------------------------	---------	---------	---------	---------



На рис. 4 показано распределение интенсивности вихревого пучка в плоскости XZ при $y=0$ в области $x \in [-5\lambda, 5\lambda]$, $z \in [0, 5\lambda]$, рассчитанное методом FDTD и ПВ с матрицей Мансурипура. Как видно, полученные картины вполне согласованы друг с другом. При этом хорошо видна структура теневой области вихревого пучка: размер световой воронки то увеличивается, то уменьшается при распространении пучка, и он может быть достаточно мал (диаметр между максимумами $1,7\lambda$ на расстоянии $z = 3\lambda$).



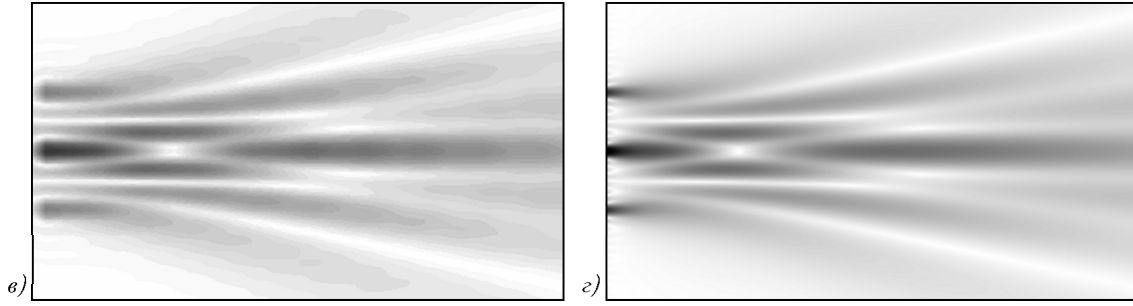


Рис.4. Распределение общей интенсивности (а), (б) и амплитуды продольной компоненты (в), (г) вихревого пучка в плоскости yz при $y=0$ в области $x \in [-5\lambda, 5\lambda]$, $z \in [0, 5\lambda]$, рассчитанное методом FDTD (а), (в) и ПВ с матрицей Мансуритур (14), (16), (21) (б), (г)

6. Учёт в методе разложения по плоским волнам коэффициентов пропускания Френеля

6.1. Представление через TE- и TM-компоненты

Запишем метод разложения по плоским волнам в общем виде в декартовых координатах, причём представим электрический вектор поля через сумму TE- и TM-поляризованных полей [5]:

$$\mathbf{E}(u, v, z) = \mathbf{E}_{\text{TE}}(u, v, z) + \mathbf{E}_{\text{TM}}(u, v, z), \quad (22)$$

$$\mathbf{E}_{\text{TE}}(u, v, z) = \iint_{\Sigma_s} \frac{1}{\xi^2 + \eta^2} \begin{bmatrix} \eta^2 & -\xi\eta \\ -\xi\eta & \xi^2 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} F_x(\xi, \eta) \\ F_y(\xi, \eta) \end{pmatrix} \exp\left[ikz\sqrt{1-(\xi^2 + \eta^2)}\right] \exp[ik(\xi u + \eta v)] d\xi d\eta, \quad (23)$$

$$\begin{aligned} \mathbf{E}_{\text{TM}}(u, v, z) = \\ = \iint_{\Sigma_s} \frac{1}{\xi^2 + \eta^2} \begin{bmatrix} \xi^2 & \xi\eta \\ \xi\eta & \eta^2 \\ -\frac{(\xi^2 + \eta^2)\xi}{\sqrt{1-\xi^2-\eta^2}} & -\frac{(\xi^2 + \eta^2)\eta}{\sqrt{1-\xi^2-\eta^2}} \end{bmatrix} \begin{pmatrix} F_x(\xi, \eta) \\ F_y(\xi, \eta) \end{pmatrix} \exp\left[ikz\sqrt{1-(\xi^2 + \eta^2)}\right] \exp[ik(\xi u + \eta v)] d\xi d\eta, \end{aligned} \quad (24)$$

Очевидно, что в свободном пространстве, выражения (22)-(24) записываются в классическом виде (14), (16), (17).

Однако при прохождении через границу двух сред вектора напряженности электрического поля волны, поляризованной перпендикулярно и параллельно плоскости падения, имеют различные коэффициенты пропускания Френеля. С учётом этого факта матрица (17) будет выглядеть следующим образом:

$$\mathbf{S}_E(\xi, \eta) = \frac{1}{\xi^2 + \eta^2} \begin{bmatrix} t_p \xi^2 + t_s \eta^2 & \xi \eta (t_p - t_s) \\ \xi \eta (t_p - t_s) & t_s \xi^2 + t_p \eta^2 \\ -\frac{t_p \xi (\xi^2 + \eta^2)}{\sqrt{1 - (\xi^2 + \eta^2)}} & -\frac{t_p \eta (\xi^2 + \eta^2)}{\sqrt{1 - (\xi^2 + \eta^2)}} \end{bmatrix}, \quad (25)$$

где t_s, t_p – коэффициенты пропускания Френеля для ТЕ- и ТМ-компонент, соответственно.

В модификации Мансурипура вид матрицы преобразования будет следующий:

$$\mathbf{M}_E(\xi, \eta) = \frac{1}{(\xi^2 + \eta^2)} \begin{bmatrix} t_p \gamma \xi^2 + t_s \eta^2 & \xi \eta (t_p \gamma - t_s) \\ \xi \eta (t_p \gamma - t_s) & t_s \xi^2 + t_p \gamma \eta^2 \\ -t_p \xi (\xi^2 + \eta^2) & -t_p \eta (\xi^2 + \eta^2) \end{bmatrix}, \quad \gamma = \sqrt{1 - (\xi^2 + \eta^2)}. \quad (26)$$

Заметим, что при $\gamma=1$ (соответствует нулевой пространственной частоте) (25) сводится к (17) только для поперечных компонент. Причём для продольной компоненты имеет место особенность при $\gamma \rightarrow 0$ (для высоких значений пространственных частот), которая формально приводит к неограниченному росту продольной компоненты. В модификации Мансурипура такая особенность отсутствует.

6.2. Расчет коэффициентов Френеля

Коэффициенты Френеля необходимо учитывать в случае, когда во входной плоскости имеется оптический элемент. Обычно коэффициенты рассчитываются исходя из формы элемента. Однако при использовании разложения по плоским волнам знание формы не требуется (она может быть определена с точностью до длины волны по создаваемому элементом распределению фазы). Достаточно только знать показатели преломления элемента и среды.

Коэффициенты Френеля вычисляются по формулам:

$$t_s = \frac{2n_1 \cos \beta_i}{n_1 \cos \beta_i + n_2 \cos \beta_t}, \quad t_p = \frac{2n_1 \cos \beta_i}{n_2 \cos \beta_i + n_1 \cos \beta_t}, \quad (27)$$

где n_1, n_2 – показатели преломления соответственно элемента и среды, β_i – угол падения, β_t – угол преломления. Они связаны законом преломления:

$$n_1 \cos \beta_i = n_2 \cos \beta_t \quad (28)$$

и соотношением, позволяющим обойтись без знания формы элемента [44]:

$$\cos(\beta_t - \beta_i) = \gamma \quad (29)$$

Из равенств (28) и (29) получаем значения квадратов косинусов углов

$$\cos^2(\beta_i) = \frac{(n_1 - \gamma n_2)^2}{n_1^2 + n_2^2 - 2\gamma n_1 n_2}, \quad \cos^2(\beta_t) = \frac{(n_2 - \gamma n_1)^2}{n_1^2 + n_2^2 - 2\gamma n_1 n_2}, \quad (30)$$

Если подставить выражения (30) в (27), то получим следующие результаты:

$$t_s = \frac{2}{1 + \frac{n_2 |n_2 - \gamma n_1|}{n_1 |n_1 - \gamma n_2|}}, \quad t_p = \frac{2}{\frac{n_2}{n_1} + \frac{|n_2 - \gamma n_1|}{|n_1 - \gamma n_2|}} \quad (31)$$

В таком виде коэффициенты пропускания Френеля удобно использовать только для распространяющихся волн, когда γ является вещественным. Для затухающих волн γ является мнимым и действие операции взятия модуля приводит к неверным результатам. Поэтому избавимся от знака модуля.

Пусть γ является вещественным, $0 \leq \gamma \leq 1$, примем также, что $n_1 > n_2$. В этом случае $n_1 - \gamma n_2 > 0$, поэтому модуль этой величины совпадает с ней самой, а

$$|n_2 - \gamma n_1| = \begin{cases} (n_2 - \gamma n_1), & \gamma < \frac{n_2}{n_1}, \\ -(n_2 - \gamma n_1), & \text{иначе.} \end{cases} \quad (32)$$

Таким образом, вместо (31) можно записать:

$$t_s = \begin{cases} \frac{2n_1(n_1 - \gamma n_2)}{n_1^2 + n_2^2 - 2\gamma n_1 n_2}, & \gamma < \frac{n_2}{n_1}, \\ \frac{2n_1(n_1 - \gamma n_2)}{n_1^2 - n_2^2}, & \text{иначе.} \end{cases} \quad t_p = \begin{cases} \frac{2n_1(n_1 - \gamma n_2)}{2n_1 n_2 - \gamma(n_1^2 + n_2^2)}, & \gamma < \frac{n_2}{n_1}, \\ \frac{2n_1(n_1 - \gamma n_2)}{\gamma(n_1^2 - n_2^2)}, & \text{иначе.} \end{cases} \quad (33)$$

Выражения (33) также можно использовать для учёта затухающих волн. Причём при $\gamma < n_2 / n_1$ имеет место непрерывность перехода от вещественной области к мнимой.

Выражения (33) соответствуют прохождению только через рельефную сторону оптического элемента, т.е. из среды оптического элемента в окружающую среду. Учтем далее также вход падающего излучения в оптический элемент.

В [49] приведён следующий вид соотношения между коэффициентами пропускания и отражения:

$$r^2 + t^2 \frac{N_2 \cos \beta_2}{N_1 \cos \beta_1} = 1, \quad (34)$$

где t – коэффициент пропускания, а r – коэффициент отражения, N_1 и N_2 – показатели преломления соответственно первой (откуда идут лучи) и второй среды, а β_1 и β_2 – соответственно углы падения в первой среде и выхода во вторую.

Соотношение (34) означает, что для вычисляемых по (33) коэффициентов пропускания Френеля необходимо выполнять коррекцию:

$$t^c = t \sqrt{\frac{N_2 \cos \beta_2}{N_1 \cos \beta_1}} = t A_c. \quad (35)$$

Рассмотрим теперь с учётом (35) прохождение излучения через обе стороны оптического элемента с показателем преломления n_1 , размещённого в среде с показателем преломления n_2 . Будем считать, что излучение падает на плоскую сторону оптического элемента.

Используя выражения (33), в которых меняем местами n_1 и n_2 , и, учитывая равенство нулю углов падения и преломления на плоской стороне элемента, получаем

$$t_s^1 = t_p^1 = \frac{2n_2}{n_1 + n_2} \quad (36)$$

Коррекция по формуле (41), в которой $N_1 = n_2$, $N_2 = n_1$, $\beta_1 = \beta_2 = 0$, $A_c = \sqrt{n_1/n_2}$ приводит к следующему виду корректных коэффициентов пропускания:

$$t_s^{1c} = t_p^{1c} = \frac{2\sqrt{n_1 n_2}}{n_1 + n_2}. \quad (37)$$

При прохождении через вторую (рельефную сторону) оптического элемента в (41) $N_1 = n_1$, $N_2 = n_2$, $\beta_1 = \beta_t$, $\beta_2 = \beta_i$ и коррекционный коэффициент равен

$$A_c = \sqrt{\frac{n_2 \cos \beta_t}{n_1 \cos \beta_i}}. \quad (38)$$

С учетом выше приведенных рассуждений для косинусов преломленного и падающего углов, выражение (38) можно записать следующим образом:

$$A_c = \begin{cases} \sqrt{\frac{n_2 |n_2 - \gamma n_1|}{n_1 (n_1 - \gamma n_2)}}, & \gamma - \text{вещественная}, \\ \sqrt{\frac{n_2 (n_2 - \gamma n_1)}{n_1 (n_1 - \gamma n_2)}}, & \gamma - \text{мнимая}. \end{cases} \quad (39)$$

Итоговый коэффициент пропускания равен произведению коэффициентов пропускания для плоской и рельефной сторон.

6.3. Дифракция плоской волны на бинарном би-аксиконе

В работе [50] было показано, что аподизация функции пропускания высокоапертурной фокусирующей системы фазовыми линейными скачками позволяет уменьшить размер фокального пятна при линейной поляризации. Рассмотрим аналогичный подход для дифракционного аксикона (9).

В этом случае входное поле будет иметь следующий вид (рис. 5а):

$$E_0(x, y) = E_0(r, \varphi) = \begin{cases} \exp\{i \arg[\cos(kr) \cos \varphi]\}, & r \leq R, \\ 0, & r > R, \end{cases} \quad (40)$$

не обладающий осевой симметрией, поэтому для расчётов на основе разложения по плоским волнам необходимо воспользоваться выражениями (14)-(16) с соответствующими матрицами. Для ускорения вычислений в поперечной плоскости можно применить быстрое преобразование Фурье (БПФ), но для продольных распределений этот алгоритм затруднительно применять из-за необходимости менять в каждой плоскости шаг дискретизации.

На рис. 5 и в Таблице 6 приведены сравнительные результаты моделирования с помощью метода разложения по плоским волнам (14), (16) с учётом коэффициентов Френеля для стандартной матрицы (25) и матрицы Мансурипура (26), а также FDTD. В последнем случае был использован симметричный аксикон в условиях падения на него пучка с равномерной амплитудой и фазой, имеющий линейный скачок перпендикулярно оси поляризации.

В данном случае можно говорить только о качественном совпадении результатов, полученных с помощью интегральных методов и разностного. Причем метод разложения по плоским волнам в модификации Мансурипура обеспечивает корректные результаты на расстоянии более длины волны от оптического элемента (см. рис 5б), использование классической матрицы (25) позволяет достаточно близкие к FDTD результаты в непосредственной близости к элементу.

На расстоянии $z = 7 \text{ мкм}$ ($\approx 0,66\lambda$) поперечный размер центрального светового пятна вдоль оси x составляет по полуспаду интенсивности $\text{FWHM}(-) = 0,46\lambda$ для метода FDTD и $\text{FWHM}(-) = 0,45\lambda$ для метода разложения по плоским волнам в модификации Мансурипура, что достаточно близко к результатам, предсказанным скалярной теорией ($\text{FWHM} = 0,42\lambda$) и почти в два раза меньше, чем при использовании радиально-симметричного аксикона. Такой эффект был достигнут за счет формирования на оптической оси распределения, в основном состоящего из продольной компоненты [50].

В непосредственной близости к оптическому элементу (в плоскости $z = 0,5 \text{ мкм} \approx 0,05\lambda$), как показывают результаты применения метода FDTD, формируется очень компактное круглое световое пятно с $\text{FWHM} = 0,32\lambda$ и $\text{HMA} = 0,08\lambda^2$. Такой же горизонтальный размер получается при использовании классической матрицы (25), хотя световое пятно в этом случае вытянутое.

Из рис. 6 видно, что протяженный фокус с компактным световым пятном сохраняется на расстоянии примерно две длины волны от оптического элемента, что соответствует радиусу биаксикона.

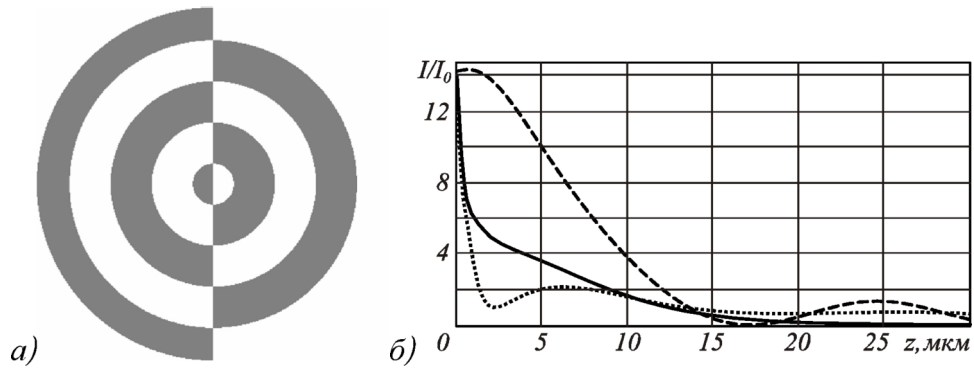


Рис. 5. Результаты моделирования для фазового биаксикона (а): осевое распределение интенсивности (б), полученное с помощью выражений (14), (16), (25) (пунктирная линия), (14), (16), (26) (точечная линия) и метода FDTD (сплошная линия)

Таблица 6. Результаты моделирования дифракции на бинарном биаксиконе ограниченной плоской волны с линейной х-поляризацией на расстоянии $z = 7 \text{ мкм}$ от элемента, размер изображений $50 \text{ мкм} \times 50 \text{ мкм}$

	Выражения (19), (20), (22)	Выражения (19), (20), (28)	Метод FDTD
Амплитуда $ E_x $			
Амплитуда $ E_y $			
Амплитуда $ E_z $			
Интенсивность $ E ^2$	 FWHM(–) = $0,33\lambda$ FWHM() = $0,52\lambda$ HMA = $0,136\lambda^2$	 FWHM(–) = $0,45\lambda$ FWHM() = $0,52\lambda$ HMA = $0,175\lambda^2$	 FWHM(–) = $0,46\lambda$ FWHM() = $0,45\lambda$ HMA = $0,179\lambda^2$

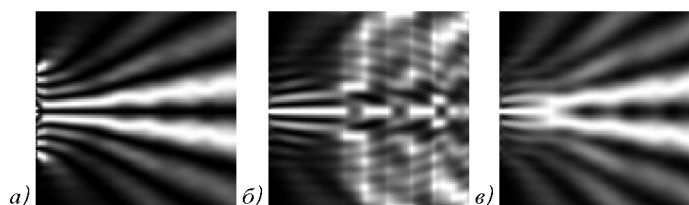
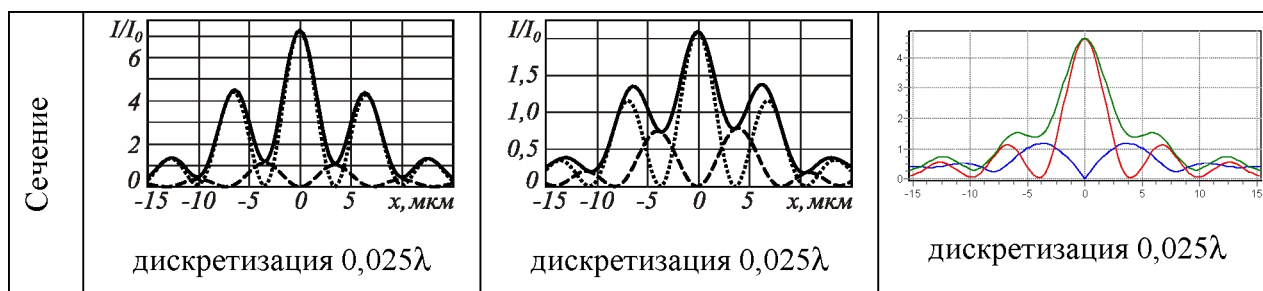


Рис. 6. Картина дифракции на биаксиконе (топология), рассчитанная с использованием (14), (16), (25) в области $z \in [0, 1 \text{ мкм}, 100 \text{ мкм}]$, $x \in [-50 \text{ мкм}, 50 \text{ мкм}]$: интенсивность для x-компоненты (а), z-компоненты (б), суммарная (в)

7. Описание программного обеспечения FieldPropagators.

7.1. Структура языка

Можно выделить следующие типы параметров

- параметры задания одного или нескольких вариантов расчета,
- параметры задания одного или нескольких пропагаторов (операторов распространения),
- параметры задания одного или нескольких входных полей,
- параметры общие для всех пакетов.

Строка задания вариант расчета имеет следующий формат:

target = <наименованиеВариантаРасчета> <тпараметры> [: <суффикс>]

В файле параметров должен быть задан один или несколько вариантов расчета. Если задано более одного варианта расчета, то они должны иметь различный суффикс. В файле параметров может быть не более одного варианта расчета без суффикса.

Строка задания пропагатора имеет следующий формат:

propagator = <имяПропагатора> <параметры> [: <суффикс>]

В файле параметров может быть задан один или несколько пропагаторов. Если задано более одного пропагатора, то они должны иметь различный суффикс (<postfix>). В файле параметров может быть не более одного пропагатора без суффикса. Для вывода графика или

изображений входного поля пропагатор не нужен. В этом случае можно не задавать пропагатор.

Пропагаторы могут быть векторными или скалярными.

Строка задания входного поля имеет следующий формат:

objectField = <имяТипаПоля> <параметры> [: <суффикс>]

В файле параметров должно быть задано одно или несколько входных полей. Если задано более одного входного поля, то они должны иметь различный суффикс (<postfix>). В файле параметров может быть не более одного входного поля без суффикса.

Входные поля могут быть одного из трех типов:

- поля для скалярных пропагаторов (имеют одну компоненту),
- поля для векторных пропагаторов, заданные одной компонентой и коэффициентами поляризации,
- поля для векторных пропагаторов, заданные двумя компонентами.

Возможность множественного задания пропагаторов, входных полей и вариантов расчета позволяет проще и быстрее решать многие задачи. Например, для сравнения работы разных пропагаторов достаточно задать более одного пропагатора. Или для одного поля и одного пропагатора задать несколько вариантов расчета. Или сравнить результат от нескольких входных полей.

7.1.1. Пакеты параметров

В случае задания нескольких пропагаторов, входных полей и вариантов расчета выполняется формирование пакетов параметров. В каждом пакете параметров лишь одно исходное поле, один вариант расчета и не более одного пропагатора (если нужен). Таким образом, при обработке задания на расчет формируются и обрабатываются пакеты параметров.

Существуют параметры общие для всех пакетов. К ним относятся параметры задания волны, показателя преломления элемента, задания префикса (приставки) всех выходных файлов и другие.

Для того чтобы различать файлы, формируемые при выполнении задания, каждому пакету приписывается имя. Имя пакета формируются из:

- приставки для всех файлов (задается в **result.filePrefix**),
- суффикса поля (может отсутствовать),
- суффикса пропагатора (может отсутствовать),

- суффикса варианта расчета (может отсутствовать).

Суффиксы дополняются последовательно, отделяясь от предыдущей части подчеркиванием.

При выполнении для каждого пакета в файл протокола выводятся:

- строка имени пакета,
- параметры пакета,
- характеристики входного поля (энергия, минимальное и максимальное значения интенсивности),
- список имен сформированных файлов.

В зависимости от варианта расчета для каждого пакета формируется один или более файлов. Имена всех файлов пакета начинаются с имени пакета.

7.1.2. Пакеты несовместимых параметров

В некоторых случаях могут сформироваться пакеты несовместимых параметров.

Можно выделить два типа таких пакетов:

- несоответствие входного поля и пропагатора (векторное поле для скалярного пропагатора или наоборот),
- вывод графиков или изображений спектра для пропагатора, в котором нет спектра.

Это не считается ошибкой. В этом случае просто выдается сообщение о несоответствии и на этом обработка пакета параметров заканчивается.

7.2. Варианты запуска программы.

Существует различные варианты запуска программы:

- запуск на расчет на персональном компьютере под Win*,
- запуск на расчет на персональном компьютере под Linux,
- запуск на расчет в фоновом режиме на кластере.
- для выдачи краткой информации,

7.2.1. Запуск на расчет под Win*.

Для запуска на расчет надо запустить программу, указав имя файла параметров:

Например,

FieldPropagators pars._pars

7.2.2. Запуск на расчет под Linux.

Для запуска на расчет надо запустить программу, указав имя файла параметров:

Например,

.\FieldPropagators pars._pars

7.2.3. Запуск на расчет в фоновом режиме на кластере.

Для запуска на расчет в фоновом режиме необходимо сформировать командный файл для диспетчера задач и запустить процесс командой

qsub comFile

При этом в командном файле в строке запуска программы надо указать имя файла параметров.

Командный файл comFile :

```
#PBS -l nodes=14:ppn=8
# Script for running MPI sample program on Any processors on cluster
cd SPBS_O_WORKDIR
echo #PBS -o $DIR/stdout.log
echo "Current working directory is `pwd`"
echo "Node file: SPBS_NODEFILE :"
echo "-----"
cat SPBS_NODEFILE
echo "-----"
NUM_PROCS=`bin/awk 'END {print NR}' SPBS_NODEFILE`
echo "Running on $NUM_PROCS processors."

echo "Starting run at: `date`"
# /usr/mpi/intel/openmpi-1.2.8/bin/mpirun -np 16 -hostfile /home/sv/cluster_ns ./FieldPropagators p._pars
/usr/mpi/intel/openmpi-1.2.8/bin/mpirun -np $NUM_PROCS -hostfile SPBS_NODEFILE ./FieldPropagators
p._pars
echo "Job finished at: `date`"
```

При этом в первой строке указывается число узлов (14 в данном примере) и число процессоров на каждом узле (8). Максимальные значения этих параметров зависят от конфигурации конкретного кластера. Не обязательно использовать все узлы кластера, т.е. если на кластере есть 15 узлов можно задавать для программы от 1 до 15 узлов.

7.2.4. Выдача справочной информации

Если запустить программу без параметров, то на консоль будет выдано краткое описание языка задания параметров:

```
main properties
waveLength = <realValue>
mediaRefractiveIndex = <complexValue>
result.filePrefix = <stringValue>
result.isTrace = <booleanValue={ false | true }>

additional output ( optional )
result.additionalOutput = amplitude
result.additionalOutput = intensitySquare
result.additionalOutput = topologyForPlanImage

one or more of
target = objectRadialGraph <angleInGrad> <r1> <r2> <pointN> [ : <postfix> ]
target = objectImage [ : <postfix> ]
target = objectField [ : <postfix> ]
target = spectrumRadialGraph <angleInGrad> <r1> <r2> <pointN> [ : <postfix> ]
target = spectrumImage [ : <postfix> ]
target = weightedSpectrumRadialGraph <angleInGrad> <r1> <r2> <pointN> <z> [ : <postfix> ]
target = weightedSpectrumImage <z> [ : <postfix> ]
target = radialGrapAccrossOz <angleInGrad> <r1> <r2> <pointN> <z> [ : <postfix> ]
```

target = graphAcrossOz <x1> <y1> <x2> <y2> <pointN> <z> [: <postfix>]
 target = imageAcrossOz <xSize> <xN> <ySize> <yN> <z> [: <postfix>]
 target = fieldAcrossOz <xSize> <xN> <ySize> <yN> <z> [: <postfix>]
 target = graphAlongOz <x> <y> <z1> <z2> <zN> [: <postfix>]
 target = imageAlongOzByXY <x1> <y1> <x2> <y2> <xyN> <z1> <z2> <zN> [: <postfix>]
 target = imageAlongOzByR <angleInGrad> <r1> <r2> <rN> <z1> <z2> <zN> [: <postfix>]
 target = timeGraphOfCrossSections <xSize> <xN> <xN_step> <ySize> <yN> <yN_step> <z> <graphPointN> [: <postfix>]

one or more of

propagator = scalarFourier [: postfix]
 propagator = scalarFresnel [: postfix]
 propagator = scalarByPlaneWaveExpansion <yN> <yHS> <xN> <xHS> <oat> [: postfix]
 propagator = scalarRayleighSommerfeld [: postfix]
 propagator = vectorRayleighSommerfeld [: postfix]
 propagator = vectorByPlaneWaveExpansionMansuripur <yN> <yHS> <xN> <xHS> <oat> [: postfix]
 propagator = vectorByPlaneWaveExpansionMansuripur+FresnelCoeff <yN> <yHS> <xN> <xHS> <oat> <n>
 <correctoinType> [: postfix]
 propagator = vectorByPlaneWaveExpansionStratton <yN> <yHS> <xN> <xHS> <oat> <delta> [: postfix]
 propagator = vectorByPlaneWaveExpansionStratton+FresnelCoeff <yN> <yHS> <xN> <xHS> <oat> <n>
 <correctoinType> <delta> [: postfix]
 <yHS> <xHS> - halfwidthes
 <oat> = { pointwise | cellwise } - objectValueApproximationType
 <n> - elementRefractiveIndex (complex)
 <correctoinType> = { no reliefSide planeSide bothSides }

one ore more objectField from

objectField = fieldForScalarPropagator <xR> <xN> <yR> <yN> <fName> [: <postfix>]
 objectField = fieldForVectorPropagatorWithPolarizationCoef <xR> <xN> <yR> <yN> <fName> <cx> <cy> [: <postfix>]
 objectField = fieldForVectorPropagatorAsTwoPolarizationField <xR> <xN> <yR> <yN> <fNameX> <fNameY> [: <postfix>]
 <xR> <yR> - metricalHalfSideSizes
 <xN> <yN> - pixelSizes
 <fName> <fNameX> <fNameY> - fieldName

simpleElementField list (with parNames and comments after '#')

<fieldName> = ConstantVortical <angularOrder>
 <fieldName> = SuperGaussVortical <s> <p> <m> # $\exp(-(r/s)^p) * \exp(i*m*fi)$
 <fieldName> = GaussLaguerre <gaussianRadius> <radialOrder> <angularOrder>
 <fieldName> = Bessel <alfa> <angularOrder>
 <fieldName> = Zernike <radialOrder> <angularOrder> <maxR>
 <fieldName> = GaussHermite <gaussianRadius> <horizontalOrder> <verticalOrder>
 <fieldName> = RingSet { : <complexCoef> <ringExternalRadius> }
 <fieldName> = FieldInPolarCoordsFromFile <fileName>
 <fieldName> = CartesianFieldFromFile <fileName>
 <fieldName> = FieldByTwoCrossedGraphs <horGraphFileName> <verGraphFileName>
 <fieldName> = RandomPhase
 <fieldName> = RadialPower <a> # $a*r^b$, a-complex, b-real
 <fieldName> = ExponentOfRadialPower <a> # $\exp(a*r^b)$, a-complex, b-real
 <fieldName> = ExponentOfRadialLogarithm <a> # $\exp(a*\ln(|b*r|))$, a-complex, b-real
 <fieldName> = DecartPower <a> <c> <d> # $a*x^b + c*y^d$, a,c-complex, b,d-real
 <fieldName> = ExponentOfDecartPower <a> <c> # $\exp(a*x^b*y^c)$, a-complex, b,c-real
 <fieldName> = ExponentOfDecartLogarithm <a> <c> <d> # $\exp(a*\ln(|b*x|) + c*\ln(|d*y|))$, a,c-complex, b,d-real

ElementFieldFunction list (with parNames and comments after '#')

<fieldName> = SumOf() { : <name> <complexCoef> }
 <fieldName> = ProductOf() { <name> }
 <fieldName> = ExponentOf() <c> <name> # $\exp(c*f(..))$ c-complex
 <fieldName> = PowOf() <a> <name> # $a*(f(..))^b$ a-complex, b-real
 <fieldName> = KinoformOf() <name>

```

<fieldName> = MultiLevelKinoformOf() <levelNumber> <name>
<fieldName> = KinoformBinarizationByLevelOf() <level> <name>
    <level> - realValue from [0, 1]
<fieldName> = PhaseOf() <name>
<fieldName> = AmplitudeOf() <name>
<fieldName> = DecartTransformOf <a> <n> <b> <c> <m> <d> <g> <h> <name> # F( a*x^n+b, c*y^m+d )*g+h,
    a,b,c,d-real, g,h-complex, n,m-int
<fieldName> = RadialTransformOf <a> <b> <c> <g> <h> <name> # F( a*r^b+c )*g+h, a,b,c-real, g,h-complex
<fieldName> = RingCompositeOf { : <ci> <name> <externalR> } # ci*Fi( r ), ci-complex
    if <name> = unused, value is equal to <ci>
<fieldName> = RotationOf <angleInGrad> <name>
<fieldName> = UpDownCompositeOf <UpName> <DownName>
<fieldName> = QuadrantCompositeOf <UpRightName> <UpLeftName> <DownLeftName> <DownRightName>
<fieldName> = KinoformWithBinaryCodingByGratings <period> <fillFactor> <name>

```

7.3. Описание языка параметров.

7.3.1. Правила задания параметров.

1. Любая информация после знака # воспринимается как комментарий.
2. Каждый параметр задается в отдельной строке.
3. Все параметры задаются в формате

<имяПараметра> = <строкаЗначения>

При этом формат и семантика **<строкиЗначения>** определяется семантикой для **<имяПараметра>**.

4. Строки параметров в файле параметров могут располагаться в любом порядке.
5. Действительное число может задаваться в экспоненциальном формате (например, 1.5e-6).
6. Комплексные числа задаются без пробелов в одном из форматов:
 - <действительнаяЧасть><знак><мнимаяЧасть>*i (например, 2+1*i 5.5-3.3*i),
 - <действительнаяЧасть> (например, -2 5.5),
 - <мнимаяЧасть>*i (например, 1*i -3.3*i).
7. Лишние параметры игнорируются.

7.3.2. Типы параметров.

Можно выделить следующие типы параметров:

- общие для всех пакетов,
- параметры задания одного или нескольких вариантов расчета,
- параметры задания одного или нескольких пропагаторов,
- параметры задания одного или нескольких входных полей.

7.3.2.1 Общие для всех пакетов параметры

Общие параметры едины для всех пакетов параметров (см. пакеты параметров).

К ним относятся:

waveLength = <realValue>

mediaRefractiveIndex = <complexValue>

result.filePrefix = <stringValue>

result.isTrace = <booleanValue={ false | true }>

и параметры дополнительного вывода

result.additionalOutput = **amplitude**

result.additionalOutput = **intensitySquare**

result.additionalOutput = **topologyForPlanImage**

mpi.isParallelByPoint = **true**

Первые два параметра не требуют пояснений.

Все выходные файлы будут начинаться со строки, заданной в **result.filePrefix**.

Так если <stringValue> задано как **adb**, то файл протокола будет иметь имя '**adb._info**'

и имена всех выдаваемых файлов будут начинаться со строки **adb_** (например, **adb_int.bmp**).

Параметр **result.isTrace** определяет необходимость выдачи на консоль строки выполнения. При запуске на кластере не имеет смысла, т.к. вывод на консоль буферизуется. По умолчанию имеет значение **false**.

Примеры.

waveLength = 2

waveLength = 5.7

waveLength = 1e-6

mediaRefractiveIndex = 1.5

mediaRefractiveIndex = 2+1*i

result.filePrefix = a

result.filePrefix = **results_for**

result.filePrefix = **adb**

result.isTrace = **false**

result.isTrace = **true**

Каждый из этих параметров (кроме **result.isTrace**) обязан быть задан один раз в файле параметров. Т.е. наличие двух строк задания параметра или отсутствие задания (например, для **waveLength**) вызовет ошибку. Диагностика о ней будет выдана в файл протокола.

При выводе изображений выводятся изображения интенсивности и фазы. Если требуется выводить изображение амплитуды, то в файл параметров надо добавить строку

result.additionalOutput = **amplitude**

Если требуется выводить изображение квадрата интенсивности, то в файл параметров надо добавить строку

result.additionalOutput = intensitySquare

Если в файл параметров добавить строку

result.additionalOutput = topologyForPlanImage

то для всех продольных изображений (кроме изображений фазы) будут сформированы и изображения с поиском экстремума поперек оси OZ.

Параметр **mpi.isParallelByPoint** имеет смысл применять только при исследованиях времени расчета поперечных или продольных сечений поля. По умолчанию имеет значение **true**. В этом случае в данные рассчитанные в не главном процессе передаются по одной точке. Если этому параметру задать значение **false**, то данные передаются по строкам для поперечных сечений и по столбцам для продольных (чтобы минимизировать время расчета).

7.3.2.2 Параметры задания вариантов расчета.

В файле параметров должен быть один или несколько вариантов расчета. Если задано более одного варианта расчета, то они должны иметь различные суффиксы (**<postfix>**). В файле параметров может быть не более одного варианта расчета без суффикса. Далее идут описания параметров для каждого вариантов расчета.

ЗАМЕЧАНИЕ: при задании параметров радиальных графиков и изображений, если задать значение радиуса отрицательным, то будет взята точка с положительным радиусом и противоположным углом (смещенным на π)

ЗАМЕЧАНИЕ: при выводе изображений выводятся изображения интенсивности, фазы и других, заданных в параметрах дополнительного вывода; при этом имя файла интенсивности имеет вид *_int.bmp, фазы - *_pha.bmp, амплитуды - *_amp.bmp, топологии интенсивности - *_int_topo.bmp, топологии амплитуды - *_amp_topo.bmp

ЗАМЕЧАНИЕ: структура входного поля зависит от типа пропагатора; для скалярных пропагаторов поле имеет одну компоненту, для векторных две поперечные компоненты поляризации (вдоль OX и вдоль OY); поэтому при формировании графиков или изображений входного поля для скалярного поля будет сформирован один график или изображения одного поля, а для векторного – графики или изображения полей компонент, при этом для X-ой компоненты к имени файла добавляется суффикс ‘_x’, для Y-вой – ‘_y’; со спектрами – аналогично.

ЗАМЕЧАНИЕ: для скалярных пропагаторов поле после элемента имеет одну компоненту, для векторных три компоненты электрического поля (X, Y и Z); поэтому при формировании графиков или изображений поля для скалярного поля будет сформирован один график или изображения одной компоненты поля, а для векторного – графики или изображения полей компонент, при этом для X-ой компоненты к имени файла добавляется суффикс ‘_x’, для Y-вой – ‘_y’, для Z-вой – ‘_z’, для интенсивности поперечного поля – ‘_t’, для интенсивности поля – ‘_’.

target = objectRadialGraph <angleInGrad> <r1> <r2> <pointN> [: <postfix>]

радиальный график исходного поля, где

<angleInGrad> - угол в градусах,

<r1> - начальный радиус,

<r2> - конечный радиус

<pointN> - число точек графика

target = objectImage [: <postfix>]

изображения входного поля

нет параметров

target = fieldImage [: <postfix>]

входное поле в файл (файлы)

нет параметров

target = spectrumRadialGraph <angleInGrad> <r1> <r2> <pointN> [: <postfix>]

радиальный график спектра (только для пропагаторов, имеющих спектр)

<angleInGrad> - угол в градусах, где

<r1> - начальный радиус,

<r2> - конечный радиус

<pointN> - число точек графика

target = spectrumImage [: <postfix>]

изображения спектра (только для пропагаторов, имеющих спектр),

нет параметров

target = weightedSpectrumRadialGraph <angleInGrad> <r1> <r2> <pointN> <z> [: <postfix>]

радиальный график взвешенного спектра (только для пропагаторов, имеющих спектр) , где

<angleInGrad> - угол в градусах,

<r1> - начальный радиус,

<r2> - конечный радиус

<pointN> - число точек графика

<z> - значение координаты Z

target = weightedSpectrumImage <z> [: <postfix>]

изображения взвешенного спектра (только для пропагаторов, имеющих спектр) , где

<z> - значение координаты Z

target = radialGrapAccrossOz <angleInGrad> <r1> <r2> <pointN> <z> [: <postfix>]

радиальный график в поперечном сечении, где

<angleInGrad> - угол в градусах,

<r1> - начальный радиус,

<r2> - конечный радиус

<pointN> - число точек графика

<z> - значение координаты Z

target = grapAccrossOz <x1> <y1> <x2> <y2> <pointN> <z> [: <postfix>]

график в поперечном сечении, где

<x1> <y1> - координаты начальной точки

<x2> <y2> - координаты конечной точки

<pointN> - число точек графика

<z> - значение координаты Z

target = imageAccrossOz <xSize> <xN> <ySize> <yN> <z> [: <postfix>]

изображения поперечного сечения (центрального) , где

<xSize> <xN> - метрический и пиксельный размер вдоль оси OX

<ySize> <yN> - метрический и пиксельный размер вдоль оси OY

<z> - значение координаты Z

target = fieldAcrossOz <xSize> <xN> <ySize> <yN> <z> [: <postfix>]

поперечное сечение поля в файл, где

<xSize> <xN> - метрический и пиксельный размер вдоль оси OX

<ySize> <yN> - метрический и пиксельный размер вдоль оси OY

<z> - значение координаты Z

target = graphAlongOz <x> <y> <z1> <z2> <zN> [: <postfix>]

продольный график (на отрезке параллельном оси OZ), где

<x> <y> - координаты точки в поперечном сечении

<z1> <z2> - координаты начальной и конечной точек

<zN> - число точек графика

target = imageAlongOzByXY <x1> <y1> <x2> <y2> <xyN> <z1> <z2> <zN> [: postfix]

изображения продольного сечения, где

<x1> <y1> <x2> <y2> - координаты концов отрезка в поперечном сечении

<xyN> - число пикселей в поперечном сечении (по вертикали)

<z1> <z2> - координаты концов отрезка в продольном сечении

<zN> - число пикселей в продольном сечении (по горизонтали)

target = imageAlongOzByR <angleInGrad> <r1> <r2> <rN> <z1> <z2> <zN> [: <postfix>]

изображения продольного радиального сечения, где

<angleInGrad> - угол в градусах

<r1> <r2> - координаты концов отрезка в поперечном сечении

<rN> - число пикселей в поперечном сечении (по вертикали)

<z1> <z2> - координаты концов отрезка в продольном сечении

<zN> - число пикселей в продольном сечении (по горизонтали)

**target = timeGraphOfCrossSections <xSize> <xN> <xN_step> <ySize> <yN> <yN_step> <z>
<graphPointN> [: <postfix>]**

график времени вычисления поперечных сечений изменяемых размеров, где

<xSize> <xN> <xN_step> - метрический и пиксельный размер вдоль оси OX и приращение пиксельного размера,

<ySize> <yN> <yN_step> - метрический и пиксельный размер вдоль оси OY и приращение пиксельного размера,

<z> - значение координаты Z

<graphPointN> - число точек графика

этот вариант используется совместно с параметром **mpi.isParallelByPoint**, который и определяет вариант расчета и передачи данных поля (по точкам или по строкам/столбцам).

7.3.2.3 Параметры задания пропагатора (пропагаторов).

В файле параметров может быть задан один или несколько пропагаторов. Если задано более одного пропагатора, то они должны иметь различные суффиксы (<postfix>). В файле параметров может быть не более одного пропагатора без суффикса. Если в файле параметров заданы только варианты вывода входного поля (нет вариантов расчета поля или вывода спектра), то задание пропагатора не требуется. Далее идут описания параметров для каждого пропагатора.

ЗАМЕЧАНИЕ: для пропагаторов, имеющих спектр, задаются следующие параметры:

<yN> <yHS> - пиксельный и метрический размер по вертикали

<xN> <xHS> - пиксельный и метрический размер по горизонтали

<oat> - вариант аппроксимации значений исходного поля,
может принимать одно из значений pointwise или cellwise,
поточечной или ступенчатой аппроксимации соответственно

propagator = scalarFourier [: postfix]

скалярный пропагатор Фурье

propagator = scalarFresnel [: postfix]

скалярный пропагатор Френеля

propagator = scalarByPlaneWaveExpansion <yN> <yHS> <xN> <xHS> <oat> [: postfix]

скалярный пропагатор разложения по плоским волнам

<yN> <yHS> <xN> <xHS> <oat> - параметры спектра

propagator = scalarRayleighSommerfeld [: postfix]

скалярный пропагатор Рэллея-Зоммерфельда

propagator = vectorRayleighSommerfeld [: postfix]

векторный пропагатор Рэллея-Зоммерфельда

propagator = vectorByPlaneWaveExpansionMansuripur <yN> <yHS> <xN> <xHS> <oat> [: postfix]

векторный пропагатор Мансирипура (разложения по плоским волнам)

<yN> <yHS> <xN> <xHS> <oat> - параметры спектра

propagator = vectorByPlaneWaveExpansionMansuripur+FresnelCoeff <yN> <yHS> <xN> <xHS> <oat> <n> <correctoinType> [: postfix]

векторный пропагатор Мансирипура с коэффициентами Френеля

<yN> <yHS> <xN> <xHS> <oat> - параметры спектра

<n> - комплекснозначный показатель преломления элемента

<correctoinType> - тип коррекции, одно из значений { no reliefSide planeSide bothSides },

которые соответствуют вариантам: без учета коррекции коэффициентов Френеля, учет прохождения только через рельефную сторону оптического элемента, учет прохождения только через плоскую сторону оптического элемента и учет прохождения через обе стороны оптического элемента.

propagator = vectorByPlaneWaveExpansionStratton <yN> <yHS> <xN> <xHS> <oat> <delta> [: postfix]

векторный пропагатор Стрэттона (разложения по плоским волнам)

<yN> <yHS> <xN> <xHS> <oat> - параметры спектра

<delta> - параметр, определяющий размер игнорируемой области в спектральной плоскости в районе частот, близких к единице (область наличия особенности).

propagator = vectorByPlaneWaveExpansionStratton+FresnelCoeff <yN> <yHS> <xN> <xHS> <oat> <n> <correctoinType> <delta> [: postfix]

7.3.2.4 Параметры задания одного или нескольких входных полей.

В файле параметров должно быть задано одно или несколько входных полей. Если задано более одного входного поля, то они должны иметь различные суффиксы (<postfix>). В файле параметров может быть не более одного входного поля без суффикса. Далее идут описания параметров для каждого входного поля.

Можно задавать три варианта входных полей:

- поле для скалярных пропагаторов (задается одной компонентой),

- поле для векторных пропагаторов (задается одной компонентой и коэффициентами поляризации),

- поле для векторных пропагаторов (задается двумя компонентами поля).

Если будет нарушено соответствие входного поля и пропагатора (например, скалярный пропагатор и векторное поле), то выдается сообщение, но программа не прерывается по ошибке.

Можно задать только центрированные поля (ось OZ проходит через центр поля).

Строки задания входных полей имеют следующую структуру:

objectField = fieldForScalarPropagator <xR> <xN> <yR> <yN> <fName> [: <postfix>]

objectField = fieldForVectorPropagatorWithPolarizationCoef <xR> <xN> <yR> <yN> <fName> <cx> <cy> [: <postfix>]

objectField = fieldForVectorPropagatorAsTwoPolarizationField <xR> <xN> <yR> <yN> <fNameX> <fNameY> [: <postfix>]

где

<xR> <yR> - метрические полуширины области определения поля

<xN> <yN> - пиксельные размеры поля

<fName> <fNameX> <fNameY> - имена полей

Строка задания поля имеет следующую структуру:

<имяПоля> = <типПоля> [<параметры>]

Поля формируются из простых полей и полей, заданных как функция или суперпозиция других полей.

Например.

```
objectField = fieldForScalarPropagator 1 100 1 100 a1
objectField = fieldForScalarPropagator 1 100 1 100 a2 : a2
objectField = fieldForScalarPropagator 1 100 1 100 a3 : aa3
# <fieldName> = RandomPhase
a1 = RandomPhase
# <fieldName> = RadialComplexPower <a> <b> # a*r^b, a-complex, b-real
a2 = RadialComplexPower 3 2
# <fieldName> = UpDownCompositeOf <UpName> <DownName>
a3 = UpDownCompositeOf a1 a2
```

Строки комментариев выделены серым цветом. Они приведены для пояснения семантики параметров.

В данном примере первое поле (без суффикса) имеет случайную фазу при постоянной амплитуде. Второе поле имеет суффикс a2 и определяется как радиальная функция $3 \cdot r^2$.

Третье поле (суффикс aa3) составное: в верхней половине случайная фаза, а в нижней - радиальная функция. Задание второго поля демонстрирует также, что суффикс поля и имя функции могут совпадать.

7.3.2.5 Структура файла графиков (*.gr).

Данный формат предназначен для сохранения одного или нескольких графиков в одном файле. Каждый график сопровождается строкой комментария.

Можно сохранять комплекснозначный график.

<строкаКомментария>

<строкаОписания>

<строкаОднойТочки>

...

<строкаОднойТочки>

<строкаОписания> := <числоТочек> [<индексЦвета>]

<строкаОднойТочки> := <xValue> <yValue>

<xValue> := <действительноеЧисло>

<yValue> := <realValue> [<imagePartOfComplexValue>]

<realValue> := <действительноеЧисло>

Примеры.

График содержит один отрезок, x1=0, y1=1, x2=2, y2=4

```
-----
example: line
2
0 1
2 4
-----
```

График содержит один отрезок, x1=0, y1=-1e+27, x2=2, y2=4.66

```
-----
example: line
2
-0.1 -1e+27
2.2 4.66
-----
```

График содержит 3 точки: x1=5, y1=1, x2=2, y2=4, x1=0, y1=5

```
-----
example: line
2
5 1
2 4
0 5
```

Два графика в файле

Первый график содержит один отрезок, $x1=0, y1=1, x2=2, y2=4$.

Второй график содержит один отрезок, $x1=10, y1=1, x2=12, y2=4$.

line 1

2

0 1

2 4

line 1

2

10 1

12 4

Комплекснозначный график содержит 3 точки:

$x1=0, y1=\text{Complex}(1, 2)$ $x2=2, y2=\text{Complex}(4, 3)$, $x1=5, y1=\text{Complex}(5, 4)$

example: line

2

0 1 2

2 4 3

5 5 4

Комплекснозначный график содержит 3 точки:

$x1=0, y1=\text{Complex}(1, 0)$ $x2=2, y2=\text{Complex}(4, 3)$, $x1=5, y1=\text{Complex}(5, 0)$

example: line

2

0 1

2 4 3

5 5

7.3.2.6 Список базовых функций.

<fieldName> = ConstantVortical <angularOrder>

Вихревое поле с постоянной амплитудой

$$\Psi_m(\varphi) = \exp(im\varphi)$$

m – <angularOrder> - порядок вихря (действительное)

<fieldName> = SuperGaussVortical <s> <p> <m> # $\exp(-(r/s)^p) * \exp(i*m*\varphi)$

Вихревое поле с супергауссовой амплитудой

$$\Psi_m(r, \varphi) = \exp\left(-\left(\frac{r}{\sigma}\right)^p\right) \exp(im\varphi)$$

σ – <s> - гауссовый радиус (действительное)

p – <p> - экспоненциальная степень (действительное)

$m - \langle m \rangle$ порядок вихря (действительное)

<fieldName> = GaussLaguerre <gaussianRadius> <radialOrder> <angularOrder>

Моды Гаусса-Лагерра:

$$\Psi_{nm}(r, \varphi) = \frac{1}{\sigma} \sqrt{\frac{n!}{\pi(n+|m|)!}} \cdot \exp\left(-\frac{r^2}{2\sigma^2}\right) \left(\frac{r}{\sigma}\right)^{|m|} L_n^{|m|}\left(\frac{r^2}{\sigma^2}\right) \exp(im\varphi), \quad n \geq 0$$

$\sigma - \langle \text{gaussianRadius} \rangle$ - гауссовый радиус (действительное)

$n - \langle \text{radialOrder} \rangle$ - радиальный индекс (целое, положительное)

$m - \langle \text{angularOrder} \rangle$ - угловой индекс (целое)

<fieldName> = Bessel <alfa> <angularOrder>

Моды Бесселя:

$$\Psi_{\alpha m}(r, \phi) = \frac{(-i)^m}{r_0 \sqrt{\pi}} J_m(\alpha r) \exp(im\phi)$$

$\alpha - \langle \text{alfa} \rangle$ - масштабный радиальный индекс (действительное)

$m - \langle \text{angularOrder} \rangle$ - угловой индекс (целое)

<fieldName> = Zernike <radialOrder> <angularOrder> <maxR>

Функции Цернике:

$$\Psi_{nm}(r, \varphi) = \frac{1}{r_0} \sqrt{\frac{n+1}{\pi}} R_n^{|m|}(r/\sigma) \exp(im\varphi), \quad n \geq 0, \quad |m| \leq n, \quad (n-m) - \text{четное},$$

$n - \langle \text{radialOrder} \rangle$ - радиальный индекс (целое, положительное)

$m - \langle \text{angularOrder} \rangle$ - угловой индекс (целое)

$\sigma - \langle \text{maxR} \rangle$ (действительное)

<fieldName> = GaussHermite <gaussianRadius> <horizontalOrder> <verticalOrder>

Моды Гаусса-Эрмита:

$$\Psi_{nm}(x, y) = \frac{1}{\sigma} \sqrt{\frac{(-i)^{n+m}}{2^{n+m} n! m!}} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) H_n\left(\frac{x}{\sigma}\right) H_m\left(\frac{y}{\sigma}\right), \quad n, m \geq 0$$

$\sigma - \langle \text{gaussianRadius} \rangle$ - гауссовый радиус (действительное)

$n - \langle \text{horizontalOrder} \rangle$ - индекс по горизонтали (целое, положительное)

$m - \langle \text{verticalOrder} \rangle$ - индекс по вертикали (целое, положительное)

<fieldName> = RingSet { : <complexCoef> <ringExternalRadius> }

Кусочно-постоянная радиальная функция:

$$g(r) = \begin{cases} a_1, & r \in [0, r_1] \\ a_2, & r \in [r_1, r_2] \\ \dots & \end{cases}$$

<fieldName> = FieldInPolarCoordsFromFile <fileName>

Поле в полярных координатах, считанное из файла специального формата.

<fieldName> = CartesianFieldFromFile <fileName>

Поле в декартовых координатах, считанное из файла специального формата.

<fieldName> = FieldByTwoCrossedGraphs <horGraphFileName> <verGraphFileName>

Поле в декартовых координатах, созданное как произведение одномерных распределений по координатам x и y , считанных из соответствующих файлов.

<fieldName> = RandomPhase

Случайная фаза

<fieldName> = RadialPower <a> # $a \cdot r^b$, a -complex, b -real

$$g(r) = ar^b$$

a – <complexCoef> (комплексное)

b – <power> (действительное)

<fieldName> = ExponentOfRadialPower <a> # $\exp(a \cdot r^b)$, a -complex, b -real

$$g(r) = \exp(ar^b),$$

a – <complexCoef> (комплексное)

b – <Power> (действительное)

<fieldName> = ExponentOfRadialLogarithm <a> # $\exp(a \cdot \ln(|b \cdot r|))$, a -complex, b -real

$$g(r) = \exp(a \ln|br|),$$

a – <a> (комплексное)

b – (действительное)

<fieldName> = DecartPower <a> <c> <d> # $a*x^b + c*y^d$, a,c-complex, b,d-real

$$g(x, y) = ax^b + cy^d$$

a – <a> (комплексное)

b – (действительное)

c – <c> (комплексное)

d – <d> (действительное)

<fieldName> = ExponentOfDecartPower <a> <c> # $\exp(a*x^b*y^c)$, a-complex, b,c-real

$$g(x, y) = \exp(ax^b y^c),$$

a – <a> (комплексное)

b – (действительное)

c – <c> (действительное)

<fieldName> = ExponentOfDecartLogarithm <a> <c> <d>

$\exp(a*\ln(|b*x|) + c*\ln(|d*y|))$, a,c-complex, b,d-real

$$g(x, y) = \exp(a \ln|bx| + c \ln|dy|),$$

a – <a> (комплексное)

b – (действительное)

c – <c> (комплексное)

d – <d> (действительное)

7.3.2.7 Список операций над функциями.

<fieldName> = SumOf() { : <name> <complexCoef> }

- сумма с комплексными коэффициентами;

<fieldName> = ProductOf() { <name> }

- произведение;

<fieldName> = ExponentOf() <c> <name> # $\exp(c*f(..))$ c-complex

- экспонента с комплексным коэффициентом

<fieldName> = PowOf() <a> <name> # $a*(f(..)^b)$ a-complex, b-real

- степень с комплексным коэффициентом

<fieldName> = KinoformOf() <name>

- создание киноформа: амплитуда заменяется на единичную;

<fieldName> = MultiLevelKinoformOf() <levelNumber> <name>

- квантование фазы для нескольких уровней:

$\varphi \in [\pi/2, 3\pi/2]$ заменяется на π , остальные на 0;

<fieldName> = KinoformBinarizationByLevelOf() <level> <name>

<level> - realValue from [0, 1]

- создание бинарного киноформа: амплитуда заменяется на единичную, а фаза обинаривается по заданному уровню в долях от 2π ;

<fieldName> = PhaseOf() <name>

- вычисление фазы поля $g(x) = \arg[f(x)]$

<fieldName> = AmplitudeOf() <name>

- создание амплитудной функции, фаза обнуляется

<fieldName> = DecartTransformOf <a> <n> <c> <m> <d> <g> <h> <name>

$F(a*x^n + b, c*y^m + d)*g + h$, a,b,c,d-real, g,h-complex, n,m-int

Декартовое преобразование вида $g(x) = f(ax^n + b, cy^m + d)g + h$,

<fieldName> = RadialTransformOf <a> <c> <g> <h> <name>

$F(a*r^b + c)*g + h$, a,b,c-real, g,h-complex

Радиальное преобразование вида $g(r) = f(ar^b + c)g + h$,

<fieldName> = RingCompositeOf { : <ci> <name> <externalR> }

$ci*Fi(r)$, ci-complex, if <name> = unused, value is equal to <ci>

Набор функций, расположенных в последовательных вложенных кольцах со соответствующими комплексными множителями

<fieldName> = RotationOf <angleInGrad> <name>

Поворот двумерного распределения на заданный угол

<fieldName> = UpDownCompositeOf <UpName> <DownName>

Поле, составленное из двух половинок, в каждой из которых своя функция

**<fieldName> = QuadrantCompositeOf <UpRightName> <UpLeftName>
<DownLeftName> <DownRightName>**

Поле, составленное из четырех квадрантов, в каждом из которых своя функция

8. Примеры использования программного обеспечения FieldPropagators

В каждом примере дается его описание, приводятся тексты файла параметров и файла протокола.

В файлы параметров добавлены строки комментариев из справочной информации, которые выделены серым цветом.

8.1 Пример с одним пакет параметров

Демонстрирует вариант параметров, содержащих один пакет без использования суффиксов.

Файл входных параметров (строки комментариев выделены серым цветом):

```

waveLength = 1
mediaRefractiveIndex = 1.5
result.filePrefix = a
result.isTrace = false

# propagator = scalarFresnel [ : postfix ]
propagator = scalarFresnel

# objectField = fieldForScalarPropagator <xS> <xN> <yS> <yN> <fName> [ :<postfix>]
objectField = fieldForScalarPropagator 2 101 2 101 a

# <fieldName> = ConstantVortical <angularOrder>
a = ConstantVortical 2

# target = fieldImageAccrossOz <xSize> <xN> <ySize> <yN> <z> [ : <postfix> ]
target = fieldImageAccrossOz 2 101 2 101 5

```

Файл протокола имеет имя a._info и содержит:

```

D:\work\cpp\field\try\FieldPropagators.exe
propertyFileName = p._pars

```

строка имени программы и строка, содержащая имя файла параметров,
далее идут строки введенных параметров, лишние параметры в этом списке отсутствуют,
в начале каждой строки выводится индекс этой строки в файле параметров

```
----- Input parameters: <index> <propName> = <string>
1 waveLength = 1
2 mediaRefractiveIndex = 1.5
3 result.filePrefix = a
4 result.isTrace = false
7 propagator = scalarFresnel
10 objectField = fieldForScalarPropagator 2 101 2 101 a
13 a = ConstantVortical 2
16 target = fieldImageAccrossOz 2 101 2 101 5
```

далее идут параметры пакета (общие параметры не выводятся повторно)
в начале каждой строки выводится индекс этой строки в файле параметров

```
-- prefix = a
10 objectField = fieldForScalarPropagator 2 101 2 101 a
7 propagator = scalarFresnel
16 target = fieldImageAccrossOz 2 101 2 101 5
13 a = ConstantVortical 2
```

далее идут параметры входного поля: энергия, минимальное и максимальное значения

```
<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
a [ConstantVortical] e = 16. min = 1. max = 1.
```

далее перечислены имена сформированных файлов и время выполнения пакета, для
изображения интенсивности выдаются характеристики (максимальное значение, энергия,
отношение энергии в изображении к энергии входного поля и отношение энергии к
площади изображения)

```
a_int.bmp formed maxV = 0.879317 e = 1.521399 e/e0 = 0.095087 e/s = 0.38035
a_pha.bmp formed
-- Computation time: 25.516000 sec
```

далее идет время работы программы

```
Computation time: 25.531000 sec
```

8.2 Пример с несколькими входными полями

Демонстрирует построение сложного поля с выводом всех промежуточных полей.

Файл входных параметров (строки комментариев выделены серым цветом):

```
waveLength = 1
mediaRefractiveIndex = 1.5
result.filePrefix = a
result.isTrace = false

# propagator = scalarFresnel [ : postfix ]
propagator = scalarFresnel
```

```

# objectField = fieldForScalarPropagator <xS> <xN> <yS> <yN> <fName> [ : <postfix> ]
objectField = fieldForScalarPropagator 1 100 1 100 a1 : a1
objectField = fieldForScalarPropagator 1 100 1 100 a2 : a2
objectField = fieldForScalarPropagator 1 100 1 100 a3 : a3
# <fieldName> = RandomPhase
a1 = RandomPhase
# <fieldName> = RadialComplexPower <a> <b> # a*r^b, a-complex, b-real
a2 = RadialComplexPower 3 2
# <fieldName> = UpDownCompositeOf <UpName> <DownName>
a3 = UpDownCompositeOf a1 a2

# target = objectImage
target = objectImage

```

В данном примере также демонстрируется, что имена полей и строки суффиксов могут совпадать (a2). В этом примере задание пропагатора не требуется, т.к. задан только вывод входного поля. Поэтому строка задания пропагатора игнорируется (не выводится в списке введенных параметров).

Файл протокола имеет имя a._info и содержит:

```

D:\work\cpp\field\try\FieldPropagators.exe
propertyFileName = p._pars
----- Input parameters: <index> <propName> = <string>
1 wavelength = 1
2 mediaRefractiveIndex = 1.5
3 result.filePrefix = a
4 result.isTrace = false
10 objectField = fieldForScalarPropagator 1 100 1 100 a1 : a1
11 objectField = fieldForScalarPropagator 1 100 1 100 a2 : a2
12 objectField = fieldForScalarPropagator 1 100 1 100 a3 : a3
14 a1 = RandomPhase
16 a2 = RadialComplexPower 3 2
18 a3 = UpDownCompositeOf a1 a2
21 target = objectImage

```

далее идет строка, содержащая имя первого пакета, параметры пакета, характеристики входного поля, имена сформированных файлов и время выполнения пакета параметров

```

-- prefix = a_a1
10 objectField = fieldForScalarPropagator 1 100 1 100 a1 : a1
7 propagator = scalarFresnel
21 target = objectImage
14 a1 = RandomPhase

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
a1 [RandomPhase] e = 4. min = 1. max = 1.

a_a1_int.bmp formed maxV = 1. e = 4. e/e0 = 1. e/s = 1.
a_a1 pha.bmp formed
-- Computation time: 0.016000 sec

```

далее идет информация для второго пакета параметров

```

-- prefix = a_a2
11 objectField = fieldForScalarPropagator 1 100 1 100 a2 : a2
7 propagator = scalarFresnel
21 target = objectImage

```

16 a2 = RadialComplexPower 3 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
a2 [RadialComplexPower] e = 22.41306 min = 3.747673e-07 max = 36.

a_a2_int.bmp formed maxV = 36. e = 22.41306 e/e0 = 1. e/s = 5.603265

a_a2 pha.bmp formed

-- Computation time: 0.015000 sec

далее идет информация для третьего пакета параметров

-- prefix = a_a3

12 objectField = fieldForScalarPropagator 1 100 1 100 a3 : a3

7 propagator = scalarFresnel

21 target = objectImage

18 a3 = UpDownCompositeOf a1 a2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>

a3 [UpDownCompositeOf] e = 13.20653 min = 3.747673e-07 max = 36.

a1 [RandomPhase] e = 2. min = 1. max = 1.

a2 [RadialComplexPower] e = 11.20653 min = 3.747673e-07 max = 36.

т.к. поле составное, то выдается информация, как для всего поля, так и для полей, из которых оно состоит

a_a3_int.bmp formed maxV = 36. e = 13.20653 e/e0 = 1. e/s = 3.301632

a_a3 pha.bmp formed

-- Computation time: 0.016000 sec

Computation time: 0.063000 sec

8.3 Пример с несколькими вариантами расчета

Данный пример показывает наиболее часто используемую возможность – задание нескольких вариантов расчета.

Файл входных параметров (строки комментариев выделены серым цветом):

waveLength = 10.6

mediaRefractiveIndex = 1.5

result.filePrefix = a

result.isTrace = true

result.additionalOutput = amplitude

target = objectImage [: <postfix>]

target = objectImage : obj

target = radialGrapAccrossOz <angleInGrad> <r1> <r2> <pointN> <z> [: <postfix>]

target = radialGrapAccrossOz 0 -2 2 101 5 : z5

target = imageAccrossOz <xSize> <xN> <ySize> <yN> <z> [: <postfix>]

target = imageAccrossOz 2 101 2 101 5 : z05

target = imageAccrossOz 2 101 2 101 10 : z10

propagator = scalarRayleighSommerfeld [: postfix]

propagator = scalarRayleighSommerfeld

objectField = fieldForScalarPropagator <xR> <xN> <yR> <yN> <fName> [: <postfix>]

objectField = fieldForScalarPropagator 5 31 5 31 fun

```
# <fieldName> = ConstantVortical <angularOrder>
fun = ConstantVortical 2
```

В данном примере заданы :

- вывод изображений входного поля (target с суффиксом obj),
- вывод графика поля на расстоянии $z = 5$ (суффикс z5),
- вывод изображений поля при $z = 5$ (суффикс z05),
- вывод изображений поля при $z = 10$ (суффикс z10).

Кроме этого задан дополнительный вывод изображений амплитуды (**result.additionalOutput = amplitude**).

Файл протокола имеет имя a._info и содержит:

```
D:\work\cpp\try\FieldPropagators.exe
propertyFileName = p._pars
----- Input parameters: <index> <propName> = <string>
1 wavelength = 10.6
2 mediaRefractiveIndex = 1.5
3 result.filePrefix = a
4 result.isTrace = true
5 result.additionalOutput = amplitude
9 target = objectImage : obj
11 target = radialGrapAccrossOz 0 -2 2 101 5 : z5
13 target = imageAccrossOz 2 101 2 101 5 : z05
14 target = imageAccrossOz 2 101 2 101 10 : z10
17 propagator = scalarRayleighSommerfeld
21 objectField = fieldForScalarPropagator 5 31 5 31 fun
24 fun = ConstantVortical 2

-- prefix = a_obj
21 objectField = fieldForScalarPropagator 5 31 5 31 fun
9 target = objectImage : obj
24 fun = ConstantVortical 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.

a_obj_int.bmp formed maxV = 1. e = 100. e/e0 = 1. e/s = 1.
a_obj_amp.bmp formed
a_obj_pha.bmp formed
-- Computation time: 0.000000 sec

-- prefix = a_z5
21 objectField = fieldForScalarPropagator 5 31 5 31 fun
17 propagator = scalarRayleighSommerfeld
11 target = radialGrapAccrossOz 0 -2 2 101 5 : z5
24 fun = ConstantVortical 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.

a_z5.gr formed
-- Computation time: 0.078000 sec

-- prefix = a_z05
```

```

21 objectField = fieldForScalarPropagator 5 31 5 31 fun
17 propagator = scalarRayleighSommerfeld
13 target = imageAccrossOz 2 101 2 101 5 : z05
24 fun = ConstantVortical 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.

a_z05_int.bmp formed maxV = 0.013752 e = 0.007853 e/e0 = 0.000079 e/s = 0.001963
a_z05_amp.bmp formed
a_z05 pha.bmp formed
-- Computation time: 7.953000 sec

-- prefix = a_z10
21 objectField = fieldForScalarPropagator 5 31 5 31 fun
17 propagator = scalarRayleighSommerfeld
14 target = imageAccrossOz 2 101 2 101 10 : z10
24 fun = ConstantVortical 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.

a_z10_int.bmp formed maxV = 0.001359 e = 0.000682 e/e0 = 6.815931e-06 e/s = 0.00017
a_z10_amp.bmp formed
a_z10 pha.bmp formed
-- Computation time: 7.937000 sec

Computation time: 15.984000 sec

```

8.4 Пример сравнения двух скалярных пропагаторов

В данном примере рассчитываются графики поля двумя пропагаторами.

Файл входных параметров (строки комментариев выделены серым цветом):

```

waveLength = 10.6
mediaRefractiveIndex = 1.5
result.filePrefix = a
result.isTrace = true

# target = radialGrapAccrossOz <angleInGrad> <r1> <r2> <pointN> <z> [ : <postfix> ]
target = radialGrapAccrossOz 0 -2 2 101 5

# propagator = scalarRayleighSommerfeld [ : postfix ]
propagator = scalarRayleighSommerfeld : rs

# propagator = scalarFresnel [ : postfix ]
propagator = scalarFresnel : fr

# objectField = fieldForScalarPropagator <xR> <xN> <yR> <yN> <fName> [ : <postfix> ]
objectField = fieldForScalarPropagator 5 101 5 101 fun

# <fieldName> = ConstantVortical <angularOrder>
fun = ConstantVortical 2

```

Т.к. заданы два пропагатора, то при их задании выгодно задавать суффиксы.

Файл протокола имеет имя a._info и содержит:

```

D:\work\cpp\try\FieldPropagators.exe
propertyFileName = p._pars
----- Input parameters: <index> <propName> = <string>

```



```

1 waveLength = 10.6
2 mediaRefractiveIndex = 1.5
3 result.filePrefix = a
4 result.isTrace = true
8 target = radialGrapAccrossOz 0 -2 2 101 5
11 propagator = scalarRayleighSommerfeld : rs
14 propagator = scalarFresnel : fr
17 objectField = fieldForScalarPropagator 5 101 5 101 fun
20 fun = ConstantVortical 2

-- prefix = a_rs
17 objectField = fieldForScalarPropagator 5 101 5 101 fun
11 propagator = scalarRayleighSommerfeld : rs
8 target = radialGrapAccrossOz 0 -2 2 101 5
20 fun = ConstantVortical 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.

a_rs.gr formed
-- Computation time: 0.750000 sec

-- prefix = a_fr
17 objectField = fieldForScalarPropagator 5 101 5 101 fun
14 propagator = scalarFresnel : fr
8 target = radialGrapAccrossOz 0 -2 2 101 5
20 fun = ConstantVortical 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.

a_fr.gr formed
-- Computation time: 0.281000 sec

Computation time: 1.047000 sec

```

8.5 Пример сравнения скалярного и векторного пропагаторов.

Файл входных параметров (строки комментариев выделены серым цветом):

```

waveLength = 10.6
mediaRefractiveIndex = 1.5
result.filePrefix = a
result.isTrace = true

# target = radialGrapAccrossOz <angleInGrad> <r1> <r2> <pointN> <z> [ : <postfix> ]
target = radialGrapAccrossOz 0 -2 2 101 5

# propagator = scalarRayleighSommerfeld [ : postfix ]
propagator = scalarRayleighSommerfeld : sRS

# propagator = vectorRayleighSommerfeld [ : postfix ]
propagator = vectorRayleighSommerfeld : vRS

# objectField = fieldForScalarPropagator <xR> <xN> <yR> <yN> <fName> [ : <postfix> ]
objectField = fieldForScalarPropagator 5 101 5 101 fun : sf

# objectField = fieldForVectorPropagatorWithPolarizationCoef <xR> <xN> <yR> <yN> <fName> <cx> <cy>
# [ : <postfix> ]
objectField = fieldForVectorPropagatorWithPolarizationCoef 5 101 5 101 fun 1. 0. : vf

# <fieldName> = ConstantVortical <angularOrder>

```

fun = ConstantVortical 2

Т.к. для скалярного и векторного пропагаторов необходимо задавать разные входные поля, поэтому заданы два варианта поля. В данном примере видно, что для пакетов с несовместимыми параметрами просто выдается сообщение об этом и никаких файлов не формируется.

Следует обратить внимание, что для скалярного пропагатора выдается один график, а для векторного – комплект графиков: по компонентам (a_vf_vRS__x.gr, a_vf_vRS__y.gr, a_vf_vRS__z.gr), интенсивности поперечного поля (a_vf_vRS__t.gr) и интенсивности всего поля (a_vf_vRS____.gr).

Файл протокола имеет имя a._info и содержит:

```
D:\work\cpp\try\FieldPropagators.exe
propertyFileName = p._pars
----- Input parameters: <index> <propName> = <string>
1 wavelength = 10.6
2 mediaRefractiveIndex = 1.5
3 result.filePrefix = a
4 result.isTrace = true
7 target = radialGrapAccrossOz 0 -2 2 101 5
10 propagator = scalarRayleighSommerfeld : sRS
13 propagator = vectorRayleighSommerfeld : vRS
16 objectField = fieldForScalarPropagator 5 101 5 101 fun : sf
19 objectField = fieldForVectorPropagatorWithPolarizationCoef 5 101 5 101 fun 1. 0. : vf
22 fun = ConstantVortical 2

-- prefix = a_sf_sRS
16 objectField = fieldForScalarPropagator 5 101 5 101 fun : sf
10 propagator = scalarRayleighSommerfeld : sRS
7 target = radialGrapAccrossOz 0 -2 2 101 5
22 fun = ConstantVortical 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.

a_sf_sRS.gr formed
-- Computation time: 0.765000 sec

-- prefix = a_sf_vRS
16 objectField = fieldForScalarPropagator 5 101 5 101 fun : sf
13 propagator = vectorRayleighSommerfeld : vRS
7 target = radialGrapAccrossOz 0 -2 2 101 5
22 fun = ConstantVortical 2

--- incompatible type of field and propagator
-- Computation time: 0.000000 sec

-- prefix = a_vf_sRS
19 objectField = fieldForVectorPropagatorWithPolarizationCoef 5 101 5 101 fun 1. 0. : vf
10 propagator = scalarRayleighSommerfeld : sRS
7 target = radialGrapAccrossOz 0 -2 2 101 5
22 fun = ConstantVortical 2

--- incompatible type of field and propagator
-- Computation time: 0.000000 sec
```

```
-- prefix = a_vf_vRS
19 objectField = fieldForVectorPropagatorWithPolarizationCoef 5 101 5 101 fun 1. 0. : vf
13 propagator = vectorRayleighSommerfeld : vRS
7 target = radialGrapAccrossOz 0 -2 2 101 5
22 fun = ConstantVortical 2
```

```
<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.
```

```
a_vf_vRS__x.gr formed
a_vf_vRS__y.gr formed
a_vf_vRS__z.gr formed
a_vf_vRS__t.gr formed
a_vf_vRS___.gr formed
-- Computation time: 1.313000 sec
```

Computation time: 2.078000 sec

8.6. Пример вывода топологии продольных сечений.

Файл входных параметров (строки комментариев выделены серым цветом):

```
waveLength = 10.6
mediaRefractiveIndex = 1.5
result.filePrefix = a
result.additionalOutput = topologyForPlanImage
result.isTrace = true

# target = imageAlongOzByR <angleInGrad> <r1> <r2> <rN> <z1> <z2> <zN> [ : <postfix> ]
target = imageAlongOzByR 0 -10 10 101 2 7 101

# propagator = scalarRayleighSommerfeld [ : postfix ]
propagator = scalarRayleighSommerfeld

# objectField = fieldForScalarPropagator <xR> <xN> <yR> <yN> <fName> [ : <postfix> ]
objectField = fieldForScalarPropagator 5 101 5 101 fun

# <fieldName> = ConstantVortical <angularOrder>
fun = ConstantVortical 2
```

В данном примере, что для файла a_int.bmp файл топологии имеет имя a_int_topo.bmp.

Для изображения фазы изображение топологии не формируется.

Файл протокола имеет имя a._info и содержит:

```
D:\work\cpp\try\FieldPropagators.exe
propertyFileName = p._pars
----- Input parameters: <index> <propName> = <string>
1 waveLength = 10.6
2 mediaRefractiveIndex = 1.5
3 result.filePrefix = a
4 result.additionalOutput = topologyForPlanImage
5 result.isTrace = true
8 target = imageAlongOzByR 0 -10 10 101 2 7 101
11 propagator = scalarRayleighSommerfeld
14 objectField = fieldForScalarPropagator 5 101 5 101 fun
17 fun = ConstantVortical 2

-- prefix = a
14 objectField = fieldForScalarPropagator 5 101 5 101 fun
```

```

11 propagator = scalarRayleighSommerfeld
8 target = imageAlongOzByR 0 -10 10 101 2 7 101
17 fun = ConstantVortical 2

<name> [<type>] e = <energy> min = <intensityMin> max = <intensityMax>
fun [ConstantVortical] e = 100. min = 1. max = 1.

a_int.bmp formed maxV = 0.53096
a_int_topo.bmp formed
a pha.bmp formed
-- Computation time: 76.797000 sec

```

Computation time: 76.813000 sec

Ниже приведены изображения для файла a_int.bmp (слева) и файла a_int_topo.bmp (справа). Видно, что на втором изображении лучше видна топология распространения света.

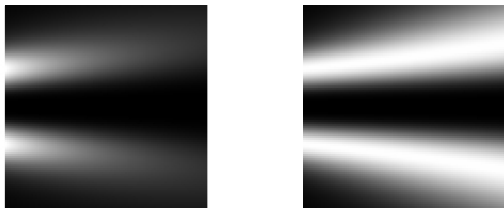


Рис. 7. Иллюстрация топологии.

9. Задания к выполнению самостоятельных работ.

1. Сформировать входное поле, соответствующее:

- а) суперпозиции двух вихревых пучков с порядками $m_1=3$ и $m_2=-4$;
- б) моде Гаусса-Лагерра $(3,-2)$;
- в) моде Бесселя нулевого порядка;
- г) моде Гаусса-Эрмита $(1,4)$;
- д) функции Цернике $(3,1)$;

2. Сформировать поле, соответствующее падению гауссового пучка на:

- а) цилиндрическую линзу - $\exp\left(\frac{ikx^2}{2f}\right)$, f – фокусное расстояние линзы;
- б) сферическую линзу - $\exp\left(\frac{ikr^2}{2f}\right)$, f – фокусное расстояние линзы;
- в) линейный аксикон - $\exp(ik\alpha r)$, α – числовая апертура аксикона;
- г) логарифмический аксикон - $\exp(ik\alpha \ln(r/b))$;

3. Выполнить моделирование действия сферической линзы с фокусным расстоянием $f = 100$ мм для излучения с длиной волны $\lambda = 0,000633$ мм в рамках скалярного приближения с помощью преобразование Фурье при условии, что на линзу падает:

- а) плоская волна, ограниченная радиусом $R = 1$ мм;
- б) гауссовый пучок радиусом $\sigma = 0,5$ мм;
- в) вихревой гауссовый пучок радиусом $\sigma = 2$ мм, порядок вихря $m = 5$;
- г) конический пучок $\exp(-i0,01kr)$ с постоянной амплитудой;

4. Сравнить моделирование распространения световых полей на расстояние 100 мкм с помощью преобразования Френеля и интеграла Рэлея-Зоммерфельда в рамках скалярного приближения для излучения с длиной волны $\lambda = 0,532$ мкм при условии, что во входной плоскости поля заданы в виде:

- а) суперпозиции вихревых пучков $m: 1, -2, 5$, ограниченная радиусом $R = 10$ мкм;
- б) моды Гаусса-Лагерра $(1,1)$ с радиусом $\sigma = 5$ мкм;
- в) моды Гаусса-Эрмита $(3,2)$ с радиусом $\sigma = 3$ мкм;
- г) поля с фазовой функцией $\exp(-i0,5k\sqrt{r})$ с постоянной амплитудой, ограниченный радиусом $R = 10$ мкм;

5. Сравнить моделирование распространения световых полей на расстояние 50 мкм с помощью скалярного и векторного интеграла Рэлея-Зоммерфельда для излучения с длиной волны $\lambda = 1$ мкм в виде плоской волны, ограниченной радиусом $R = 1$ мкм при условии, входной пучок имеет:

- а) линейную x -поляризацию;
- б) линейную y -поляризацию;
- в) круговую поляризацию;
- г) эллиптическую поляризацию;

6. Сравнить моделирование распространения световых полей на расстояние 10 мкм с помощью векторного интеграла Рэлея-Зоммерфельда и векторного разложения по плоским волнам для излучения с линейной x -поляризацией и длиной волны $\lambda = 1$ мкм в виде $\cos(x)$, ограниченной по радиусу 2 мкм:

а) построить и сохранить в bmp-формате картины распределения интенсивности для всех компонент электрического поля в плоскости, поперечной оптической оси на расстоянии 10 мкм от входной плоскости;

б) построить графики распределения интенсивности для всех компонент электрического поля в плоскости, поперечной оптической оси на расстоянии 10 мкм от входной плоскости; определить ширину центрального светового пятна по уровню полуспада от максимума интенсивности;

в) построить и сохранить в bmp-формате картины распределения интенсивности для всех компонент электрического поля в плоскости, параллельной оптической оси в области от $z_1=0,1$ мкм до $z_2=10$ мкм;

7. Сравнить моделирование фокусировки плоской волны ограниченной по радиусу 2 мкм с линейной x -поляризацией и длиной волны $\lambda=1$ мкм при использовании входного фазового распределения $\exp(-i0,5kr^2)$ с помощью векторного разложения по плоским волнам в классическом варианте и в модификации Мансурипура:

а) построить графики распределения интенсивности на оптической оси в диапазоне от $z_1=0,1$ мкм до $z_2=10$ мкм; определить положение глобального фокуса;

б) построить и сохранить в bmp-формате картины распределения интенсивности для всех компонент электрического поля в плоскости, соответствующей положению глобального фокуса;

в) построить графики распределения интенсивности для всех компонент электрического поля в плоскости, соответствующей положению глобального фокуса; определить ширину центрального светового пятна по уровню полуспада от максимума интенсивности;

8. Сравнить моделирование фокусировки вихревой волны порядка $m=-1$, ограниченной по радиусу 5 мкм с линейной x -поляризацией и длиной волны $\lambda=1$ мкм при использовании бинарной дифракционной линзы, полученной бинаризацией фазового распределения $\exp(-ikr^2)$ с помощью векторного разложения по плоским волнам в модификации Мансурипура с учетом и без учета коэффициентов Френеля:

а) построить графики распределения интенсивности на оптической оси в диапазоне от $z_1=0,1$ мкм до $z_2=5$ мкм; определить положение глобального фокуса;

б) построить и сохранить в bmp-формате картины распределения интенсивности для всех компонент электрического поля в плоскости, соответствующей положению глобального фокуса;

в) построить графики распределения интенсивности для всех компонент электрического поля в плоскости, соответствующей положению глобального фокуса; определить ширину центрального светового пятна по уровню полуспада от максимума интенсивности;

Литература

1. R. Martinez-Herrero, P.M. Mejias, S. Bosch, A. Carnicer, Vectorial structure of nonparaxial electromagnetic beams, *J. Opt. Soc. Am. A* 18, 1678-1680 (2001).
2. A. Ciattoni, B. Crosignani, and P. D. Porto, Vectorial analytical description of propagation of a highly nonparaxial beam, *Opt. Commun.* 202, 17–20 (2002)
3. Shekhar Guha, Glen D. Gillen, Description of light propagation through a circular aperture using nonparaxial vector diffraction theory, *OPTICS EXPRESS* Vol. 13, No. 5, 1424-1447 (2005)
4. Hanming Guo, Jiabi Chen, Songlin Zhuang, Vector plane wave spectrum of an arbitrary polarized electromagnetic wave, *OPTICS EXPRESS* Vol. 14, No. 6, 2095-2100 (2006)
5. Dongmei Deng and Qi Guo, Analytical vectorial structure of radially polarized light beams, *OPTICS LETTERS* Vol.32, No.18, 2711-2713 (2007)
6. Sergey P. Anokhov, Plane wave diffraction by a perfectly transparent half-plane, *J. Opt. Soc. Am. A* Vol.24, No.9, 2493-2498 (2007)
7. А.А. Ковалев, В.В. Котляр, Непараксиальная векторная дифракция гауссового пучка на спиральной фазовой пластинке, *Компьютерная оптика*, том 31, №4, 19-22 (2007)
8. Guohua Wu, Qihong Lou, Jun Zhou, Analytical vectorial structure of hollow Gaussian beams in the far field, *OPTICS EXPRESS* Vol. 16, No. 9, 6417-6424 (2008)
9. Guoquan Zhou, The analytical vectorial structure of a nonparaxial Gaussian beam close to the source, *OPTICS EXPRESS* Vol. 16, No. 6, 3504-3514 (2008)
10. Nuri Delen and Brian Hooker, Verification and comparison of a fast Fourier transform-based full diffraction method for tilted and offset planes, *APPLIED OPTICS* Vol. 40, No. 21, 3525-3531 (2001)
11. I.J. Cooper, C.J.R. Sheppard, M. Sharma, Numerical integration of diffraction integrals for a circular aperture, *Optik* vol. 113, No.7, 293–298 (2002)
12. Kailiang Duan, Baida Lu, A comparison of the vectorial nonparaxial approach with Fresnel and Fraunhofer approximations, *Optik* vol. 115, No.5, 218–222 (2004)

13. I. J. Cooper, C. J. R. Sheppard, and M. Roy, The numerical integration of fundamental diffraction integrals for converging polarized spherical waves using a two-dimensional form of Simpson's 1/3 Rule, *Journal of Modern Optics*, Vol. 52, No. 8, 1123–1134 (2005).
14. Jan A. C. Veerman, Jurgen J. Rusch, and H. Paul Urbach, Calculation of the Rayleigh–Sommerfeld diffraction integral by exact integration of the fast oscillating factor, *J. Opt. Soc. Am. A* Vol. 22, No. 4, 636–646 (2005)
15. Zhiguo Zhao, Kailiang Duan, Baida Lu, Focusing and diffraction by an optical lens and a small circular aperture, *Optik* v. 117, 253–258 (2006)
16. Xueen Wang, Zhaozhong Fan and Tiantong Tang, Numerical calculation of a converging vector electromagnetic wave diffracted by an aperture by using Borgnis potentials. I. General theory, *J. Opt. Soc. Am. A* Vol.23, No.4, 872–877 (2006)
17. Fabin Shen and Anbo Wang, Fast-Fourier-transform based numerical integration method for the Rayleigh–Sommerfeld diffraction formula, *APPLIED OPTICS* Vol. 45, No. 6, 1102–1110 (2006)
18. Kotlyar, V.V. Sharp focus area of radially-polarized Gaussian beam by propagation through an axicon / V.V. Kotlyar, A.A. Kovalev, S.S. Stafeev // *Prog. In Electr. Res. C*. -2008. – Vol.5. – P.35–43.
19. Victor Nascov and Petre Cătălin Logofătu, Fast computation algorithm for the Rayleigh–Sommerfeld diffraction formula using a type of scaled convolution, *APPLIED OPTICS* Vol. 48, No. 22, 4310–4319 (2009)
20. Kyoji Matsushima, Tomoyoshi Shimobaba, Band-Limited Angular Spectrum Method for Numerical Simulation of Free-Space Propagation in Far and Near Fields, *OPTICS EXPRESS* Vol. 17, No. 22, 19662–19673 (2009)
21. Устинов А.В. Быстрый способ вычисления интеграла Рэлея-Зоммерфельда первого типа, // *Компьютерная оптика*. – 2009. – Т. 33, № 4. – С. 412–419
22. H. Osterberg and L.W. Smith, *J. Opt. Soc. Am.* 51, 1050 (1961)
23. E. WOLF, E. W. MARCHAND, Comparison of the Kirchhoff and the Rayleigh–Sommerfeld Theories of Diffraction at an Aperture, *J. Opt. Soc. Am.* 54(5), 587–594 (1964).
24. Tore Gravelseter and Jakob J. Stamnes, Diffraction by circular apertures. 1: Method of linear phase and amplitude approximation, *APPLIED OPTICS*, Vol. 21, No. 20, 3644–3651 (1982)
25. C. J. R. Sheppard, M. Hrynevych, Diffraction by a circular aperture: a generalization of Fresnel diffraction theory, *J. Opt. Soc. Am. A* Vol. 9, No. 2, 274–281 (1992)

26. Klaus D. Mielenz, Optical Diffraction in Close Proximity to Plane Apertures. I. Boundary-Value Solutions for Circular Apertures and Slits, *J. Res. Natl. Inst. Stand. Technol.* 107, 355–362 (2002)
27. Julio A. Romero and Luis Hernández, Vectorial approach to Huygens's principle for plane waves: circular aperture and zone plates, *J. Opt. Soc. Am. A* Vol. 23, No. 5, 1141-1145 (2006)
28. J. A. Romero and L. Hernández, Diffraction by a circular aperture: an application of the vectorial theory of Huygens's principle in the near field, *J. Opt. Soc. Am. A* Vol. 25, No. 8, 2040-2043 (2008)
29. Jianlong Li, Shifu Zhu, Baida Lu, The rigorous electromagnetic theory of the diffraction of vector beams by a circular aperture, *Opt. Commun.* 282, 4475–4480 (2009)
30. M. Born and E. Wolf, *Principles of Optics*, 6th ed. (Pergamon, Oxford, 1980), Chap. 8.3.
31. C. L. Andrews, Diffraction pattern in a circular aperture measured in the microwave region, *J. Appl. Phys.* 21, 761-767 (1950)
32. S. Silver, Microwave aperture antennas and diffraction theory, *J. Opt. Soc. Am.* 52, 131-139 (1962)
33. Totzeck, M. Validity of the scalar Kirchhoff and Rayleigh-Sommerfeld diffraction theories in the near field of small phase objects / M. Totzeck // *J. Opt. Soc. Am. A* – 1991. V. 8, No. 1. – P. 27-32.
34. Tsoy, V.I. The use of Kirchhoff approach for the calculation of the near field amplitudes of electromagnetic field / V.I. Tsoy, L.A. Melnikov // *Optics Communications* – 2005. V. 256. – P. 1–9
35. R.K. Luneburg, *Mathematical Theory of Optics*, University of California Press, Berkeley, California, 1966;
36. W.H. Carter, Electromagnetic Field of a Gaussian Beam with an Elliptical Cross Section, *J. Opt. Soc. Am. A* vol. 62, No.10, 1195-1201 (1972)
37. G.P. Agrawal, D.N. Pattanayak, Gaussian beam propagation beyond the paraxial approximation *J. Opt. Soc. Am. A* vol.69, No.4, 575-578 (1979).
38. A.S. Marathay, J.F. McCalmont, On the usual approximation used in the Rayleigh–Sommerfeld diffraction theory, *J. Opt. Soc. Am. A* 21 (2004) 510–516
39. Хонина, С.Н. Распространение радиально-ограниченных вихревых пучков в ближней зоне: I. Алгоритмы расчёта / С.Н. Хонина, А.В. Устинов, А.А. Ковалев, С.Г. Волотовский // *Компьютерная оптика*. – 2010. – Т. 34, № 3. – С. 317-332.
40. J. W. Goodman, *Introduction to Fourier Optics* (McGraw-Hill, 1968), Chap. 3

41. Виноградова М.Б., Руденко О.В., Сухоруков А.П. Теория волн. - М.: Наука. Главная редакция физико-математической литературы, 1979. - 384 с.
42. С.А. Балалаев, С.Н. Хонина, Реализация быстрого алгоритма преобразования Кирхгофа на примере бесселевых пучков, Компьютерная оптика №30, 69-73 (2006)
43. S. Gradshteyn and I. M. Ryzhik, Table of Integrals, Series, and Products (Elsevier, 2007).
44. Yaoju Zhang, Ling Wang, Chongwei Zheng, Vector propagation of radially polarized Gaussian beams diffracted by an axicon, J. Opt. Soc. Am. A Vol. 22, No. 11, 2542-2542 (2005).
45. Z. Mei and D. Zhao, "Nonparaxial analysis of vectorial Laguerre-Bessel-Gaussian beams," Opt. Express 15, 11942–11951 (2007).
46. Helseth, L.E. Optical vortices in focal regions // Opt. Commun. – 2004. – Vol. 229. – P. 85–91
47. Хонина, С.Н. Распространение радиально-ограниченных вихревых пучков в ближней зоне: II. Результаты моделирования / С.Н. Хонина, А.А. Ковалев, А.В. Устинов, С.Г. Волотовский // Компьютерная оптика. – 2010. – Т. 34, № 3. – С. 332-343.
48. Mansuripur, M. Certain computational aspects of vector diffraction problems // J. Opt. Soc. Am. A. – 1989. – Vol. 6, No. 5. – P. 786-805.
49. Jenkins, F.A. Fundamentals of Optics, 4th edition / F.A. Jenkins, H.E. White – NY: McGraw-Hill Book Company, 1976. – Chapter 25.
50. Хонина, С.Н. Управление вкладом компонент вектор-ного электрического поля в фокусе высокоапертурной линзы с помощью бинарных фазовых структур / С.Н. Хонина, С.Г. Волотовский // Компьютерная оптика. – 2010. – Т. 34, № 1. – С. 58-68.