



Skolkovo Institute of Science and Technology

MASTER'S THESIS

**Evolutionary Structural Optimization Algorithm based on
FFT-GFIEM Solver for Inverse Design of Wave Devices**

Master's Educational Program: Computational Science and Engineering

Student _____

A handwritten signature in blue ink, appearing to read "Iurii Minin".

Iurii Minin

June 17, 2019

Research Advisor: _____

A handwritten signature in blue ink, appearing to read "Sergey G. Rykovannov".

Sergey G. Rykovannov
Associate Professor

Moscow 2019

All rights reserved. ©

The author hereby grants to Skoltech permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole and in part in any medium now known or hereafter created.



Skolkovo Institute of Science and Technology

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

**Алгоритм эволюционной структурной оптимизации
на основе FFT-GFEM метода
для проектирования волновых устройств**

Магистерская образовательная программа:
Вычислительные системы в науке и технике

Студент

Юрий Минин
17 июня, 2019

Научный руководитель:

Сергей Г. Рыкованов
Профессор

Москва 2019

Все права защищены. ©

Автор настоящим дает Сколковскому институту науки и технологий разрешение на воспроизведение
и свободное распространение бумажных и электронных копий настоящей диссертации в целом или
частично на любом ныне существующем или созданном в будущем носителе.

Evolutionary Structural Optimization Algorithm based on FFT-GFIEM

Solver for Inverse Design of Wave Devices

Iurii Minin

Submitted to the Skolkovo Institute of Science and Technology
on June 17, 2019

Abstract

I propose an Bi-directional Evolutionary Structural Optimization (BESO) methods with a target functions based on FFT-GFIEM (Green's Function Integral Equation Method based on fast matvec (implemented using Fast Fourier Transforms)) solver. It allows to find an optimal structure in a 3D space for homogeneous objects (lenses and metallic reflectors). The derived formula of target function gradient row expedited the iterative optimization processes to predict better solution. Such examples as antennas, glass lenses, parabolic reflectors, splitters are considered. Numerical experiments show that discrete gradient algorithm converges much faster than greedy algorithm for all considered cases. The developed BESO numerical methods were implemented using CUDA (C++) parallelization. In addition, the efficiency of parallelization in CUDA (C++) was analyzed in comparison with the computation time consumptions on Numpy (Python).

Research Advisor:

Name: Sergey G. Rykovanov

Degree: Associate Professor, PhD

Title: Associate Professor

Алгоритм эволюционной структурной оптимизации на основе FFT-GFIEM метода для проектирования волновых устройств

Юрий Минин

Представлено в Сколковский институт науки и технологий
17 июня, 2019

Реферат

Я предлагаю методы двунаправленной эволюционной структурной оптимизации с целевыми функциями на основе алгоритма FFT-GFIEM (метода интегральных уравнений с применением функций Грина на основе быстрого матричного умножения, реализованного с помощью быстрых преобразований Фурье). Предлагаемые методы позволяет найти оптимальную структуру в трехмерном пространстве для однородных объектов (линз и металлических отражателей). Полученная формула строки градиента целевой функции ускорила итерационные процессы оптимизации для прогнозирования лучшего решения. Рассмотрены такие примеры, как антенны, стеклянные линзы, параболические отражатели, разветвители. Численные эксперименты показывают, что дискретный градиентный алгоритм сходится намного быстрее, чем жадный алгоритм для всех рассмотренных случаев. Разработанные численные методы BESO были реализованы с использованием распараллеливания CUDA (C++). Кроме того, был проведен анализ эффективности распараллеливания на CUDA (C++) по сравнению с времязатратам на вычисления Numpy (Python).

Научный руководитель:

Имя: Сергей Г. Рыкованов

Ученое звание, степень: Доцент, Кандидат физико-математических наук

Должность: Профессор

Acknowledgments

I would first like to thank my thesis advisor Associate Professor Sergey Rykovnov. His high-level combination of professional skills in HPC programming, physics and computational science helped me whenever I ran into a trouble spot or had a question about my research or writing.

I would also like to thank the experts who were involved in the validation survey for this research project: Associate Professor Ivan Oseledets, PhD student Alexey Boyko, MSc student Egor Nuzhin, Research Intern Daniil Stefonishin, Junior Research Scientist Sergey Matveev, Research Scientist Igor Ostanin, Research Scientist Mikhail Litsarev.

Finally, I must express my very profound gratitude to MSc student Aleksandr Safin for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without him. Thank you.

Contents

1	Introduction	9
2	Problem statement and methods	15
2.1	Significance	16
2.2	Aims of the research work	16
2.3	Methodology	17
2.4	Suggested structural optimization	17
2.4.1	Antennas	19
2.4.2	Splitters	19
2.4.3	Superlenses	19
2.4.4	Subwavelength imaging devices	20
3	Numerical formulation	21
4	Methods to solve the problem	23
4.1	Development of BESO numerical methods	23
4.1.1	Greedy	23
4.1.2	Discrete gradient	23
4.1.3	Steepest descent	23
4.2	CUDA (C++) parallelization	23
5	Results	25
5.1	CUDA (C++) parallelization	25
5.2	FFT-GFIEM validation	30
5.3	Python BESO	35
5.4	The developed ESO and BESO numerical methods and CUDA implementations . .	39
5.4.1	Greedy	39
5.4.2	Discrete gradient	41
5.4.3	Steepest descent	45
6	Conclusion	48

A Formulas	57
A.1 FFT-GFIEM discretization	57
A.2 Fast (FFT-based) GMRES method to solve linear equation systems based on Toeplitz matrix	58
A.2.1 4-dimensional cubic Green's matrix tensor G^{4D}	58
A.2.2 Fast (FFT) matvec linear operator for G matrix by vector multiplication . .	58
A.2.3 Linear operator $\mathbf{A}(\mathbf{m})$ for $A(m)$ -matrix	59
A.3 BESO generalization development	59
A.3.1 Suggested generalized algorithm for BESO	61
A.4 Derivation of formulas for fast computations of Jacobian matrix row $J(x_{i^*})$	61

List of Figures

2.1	Wave propagation	17
2.2	Examples of design domains	18
5.1	Parallelization results (a)	26
5.2	Parallelization results (b)	27
5.3	Parallelization results (c)	28
5.4	GMRES (for FFT-GFIEM) residuals on CUDA and Numpy for $n = 1024$	28
5.5	Comparison of CUDA FFT-GFIEM solution to cylinder scattering problem with analytical one	29
5.6	FFT-GFIEM relative errors versus number of GMRES iterations for $n = 1024$	31
5.7	FFT-GFIEM relative errors on Numpy versus number of GMRES iterations	31
5.8	FFT-GFIEM relative errors on CUDA versus number of GMRES iterations	32
5.9	Optimization results (a)	35
5.10	Optimization results (b)	36
5.11	Optimization results (c)	36
5.12	Optimization results (d)	37
5.13	Optimization results (e)	37
5.14	Optimization results (f)	38
5.15	Initial design domain Ω that contains centered 4×4 device domain ω ($n = 1024$; $N = 6$ wavelengths per design domain Ω)	39
5.16	CUDA greedy ESO for lens (fixed focus)	40
5.17	CUDA discrete gradient ESO for superlens of one not fixed focus	41
5.18	CUDA discrete gradient ESO for superlens of internal focus surrounded by cutting exclusion zone	41
5.19	Attempt to post-process intuitively results of CUDA discrete gradient ESO for superlens of one fixed focus surrounded by cutting exclusion zone expressed by half-plane	42
5.20	CUDA discrete gradient BESO for superlens of three fixed focuses surrounded by cutting exclusion zone expressed by half-plane	44
5.21	CUDA discrete gradient ESO for device of required normal intensity distribution .	44

5.22 CUDA steepest descent BESO for superlens of one fixed point surrounded by exclusion zone expressed by half-plane	46
5.23 CUDA steepest descent BESO (initial mask is determined by biconvex lens) for superlens of one fixed point surrounded by exclusion zone expressed by half-plane	47

Chapter 1

Introduction

The main instruments of laser surgery and modern laser magnetic resonance imaging (MRI) [1, 2] are fiber devices, such as a surgical two-wavelength laser device FiberLase U1 (Urolaz) that is made by «IRE-Polus» and may be used in urological clinical practice to treat cancer [3]. With increasing complexity of tasks of the medical laser-based devices, human needs to develop the device components (for instance, to improve the quality of surgical procedure) are increasing. For example, device components such as polarization and wavelength splitters are necessary for these purposes [4].

Moreover, decrease in volume and increase in optical power of lens used in the computing devices design are challenging problems, which are directly related to optimization of its geometrical parameters [5]. It can be done by means of topology optimization approach, which allows to find optimal design parameters in the sense of some mathematical criteria corresponding to technological and mechanical constraints [6, 7].

Topology optimization (TO) methods allow to determine density distribution, shape and structure of material in a 3D domain under various conditions (waves and mechanical stresses influence, object interactions, heat transfer, etc.). In other words, it finds the optimal material mask for the homogeneous object or structure being studied, so that the target function of the parameter set has an extremum under certain constraints (formulated according to geometric, energetic, stiffness or any other criterion).

Furthermore, these devices should be cost-effective and should meet safety constraints. For example, consideration of reducing monetary expenditures is one of the main good reasons to develop and to improve the design processes, which minimizes the consumption of resources by improving the use of each material bit. For that, many engineering disciplines use various mathematical tools to describe the topology optimization.

The usage of topology optimization methods in computing devices design is a relatively new component of manufacturing procedure. Being a mathematical approach it allows to find and predict the considered system properties, which are not feasible in classical subtractive manufacturing methods [8].

The history of topology optimization begins with Maxwell's research. The research helped to connect strut stress distribution with design domain and to combine target functions of struts and

tie-bars [9]. However, Michell's research marked the beginning of calculus for topology optimization that decreased device domain ω area while remaining the same material flexibility by using principles of virtual work and force equilibrium [10].

Appearance of enough available (enough powerful) computers in the 1980s and 1990s provided an opportunity to solve time-consuming complex computational problems such as topology optimization. Main problems of topology optimization are tailoring the given homogeneous substance distribution in design domain Ω (for example, starting with design domains Ω of available decisions such as aircraft, bridge, house, etc.) [11, 12, 13, 14]. All these problems use target functions that should be maximized or minimized. One of the branches of modern topology optimization focuses on the development of lightweight nanophotonic devices. There topology optimization method may start, for example, with design domains Ω of available decisions such as Bragg reflector, photonic crystal, lens, dielectric mirror, cube, in nanophotonics, etc. For example, useful target function of nanophotonic splitters is the maximizing intensity of preferable signal and minimizing unexpected signals in each output junctions.

Soon after, topology optimization was integrated into software modules of most commercial CAD / CAE systems, such as Altair OptiStruct [14, 15, 16], SIMULIA Tosca module [17], Ansys [18], MSC Nastran [19, 20], etc. Since then, the refinement of theories and the development of new numerical methods are one of the most active areas in engineering materials for various purposes, such as reducing light beam intensity losses and the cost of polarization and wavelength splitters; increasing the strength of the bridges.

In the last two decades topology optimization has become an active research field. This led to the multidisciplinarity of modern topology optimization methods and their utilization in solid mechanics, hydrodynamics, thermodynamics, biology and other [21]. One of the particular interests focused on designing new photonic devices.

Appearance of new technologies and pieces of apparatus based on semiconductor lasers and optical fibers gave the birth to photonics research in the 1960s and 1970s. Photonic devices are step-by-step taking on more functions of telecommunication channels. For example, underwater optical fibers transmit gigabytes of data every second around the world by using light signals. Their electronic analogues are gradually becoming things of the past.

Despite the fact that telecommunications have become the main photonics focus area, the appearance of other photonics applications, such as fiber sensors, nonlinear optics and biophotonics are the result of deeper photonics research in interdisciplinary areas. For example, wave-controlled devices started to attract the attention of academia and industry because of their lower losses and higher bandwidth characteristics. Moreover, in a planar waveguide, the considered properties of light are limited by the total internal reflections in a small area inside sufficiently optically dense semiconductor materials. Such considered light does not have the properties of being directed by

discrete lenses or mirrors, as in volumetric optics. Planar waveguides convert photonic devices into compact sets of microstructures that have greater stability and lower energy consumption and are more portable in comparison with traditional optical components used in radiations, transmissions, amplifications, detections and signal modulations.

Researchers studied various semiconductor materials to find a good junction platform for implementing various photonic functional devices. III-V semiconductor compounds and other crystals, such as lithium niobate LiNbO_3 , were the main candidates for these devices in the early years, either due to their direct bandgaps (for radiations and light detections) or due to the Pockels effect (that has crucial significance, for example, to create an optically modulated signals). On the other hand, silicon has been established a long time ago as the most common material in the electronics industry. It is a cheap durable crystal. In addition, its complementary metal-oxide-semiconductor structure is well studied and has the capability to be mass-produced. In addition, it would be more cost-effective if the silicon photon component design software tool was readily available and took into account the current requirements and constraints of electrical circuits. In the 1980s, silicon photonics was acknowledged to be used for code-independent data transmission using light beams with wavelengths ranging from $1.3 \mu\text{m}$ to $1.55 \mu\text{m}$. In addition, the waveguides built on the silicon-on-insulator platform can direct a light beam with sufficiently low light beam intensity losses due to the large difference in refractive indices between the waveguide core and its quartz shell. Little by little, silicon becomes an important material for the design of wave photonic devices. Various efforts were made for designing and manufacturing of silicon-based photonic devices compatible with standard electronics technologies based on complementary metal-oxide-semiconductors. Although bulk crystalline silicon is not a direct bandgap material (for simpler light emissions) and does not have the Pockels effect (in order to apply silicon as a basis to construct switches), alternative properties of silicon for the development of silicon-based photonic components (including switches, light modulators and detectors) are being studied. On the other hand, heterogeneous integration between active and silicon-on-insulator-based materials [22, 23] also allows to group functional blocks of different materials on one photon chip.

To date, silicon has become the fundamental photonic material to construct optoelectronic integrated circuits and photonic integrated circuits, and progress in silicon photonics demonstrates the potential for better integration of electronics and photonics advantages.

Due to the ever-increasing Internet traffic during the current multimedia epoch, a large fiber bandwidth is required. Therefore, researchers strive to achieve not only a higher data rate per wavelength, but also a greater number of infocommunication channels per the optical fiber length. These high-speed systems are now actively developing. Their speed limits are higher than the speed limits of similar electronic components, which have originally started to use earlier. Devices such as photon switches (which can work extremely fast) have become a novel area for new methods

research. New methods are needed to increase indirect conversion speed of light into electricity and vice versa. However, with the modernization trend towards further integration of electronics and photonics, the need for additional reduction in the size of photonic components increases. Despite the fact that the cross section of the silicon waveguides have significantly decreased due to improvements of surface roughness during the manufacturing process, the traditional structure of the waveguides is still faced with problems. For example, total internal reflections, which allow light to be propagated along a waveguide, require large angles of incidence of the light. The path of the light beam looks like zigzags inside the waveguide section. This essentially puts the curvature limit value of the longitudinal waveguide bends, and, therefore, prevents further miniaturization of photonic devices.

Photonic crystals date back to 1987 [24, 25] and gradually became the most active research field after 2000s. Photonic crystal is a solid-state structure with periodically changing dielectric constant or inhomogeneity, the period of which is comparable to the light wavelength. By modeling crystal structures at a more macroscopic level, we can determine the bandgap as well as the bandgap of semiconductors while the electronic devices development. A photonic crystal bandgap is an area of energy values that a photon cannot have in a crystal. Consequently, in crystals with linear defects, light will be absorbed inside the defect line, forming an effective waveguide. Due to the fact that the propagation of light is strongly limited by the geometry of the defect, waveguides can have much sharper bends. In such waveguides, the light beam intensity loss is much less. This invention gave new impetus to designing a compact photonic circuits in a small microchip area. It effectively solves the size problem of the optoelectronic integrated circuits.

Areas for further improvements to photonic crystal components are increasing the bandwidth, reducing bending losses and enhancing other important physical and functional quantities. The rich opportunities for wave device structure component permutations is a resource for fine-tuning the characteristics of devices. In addition, various attempts have been made to locally adjust the lattice structure to improve device performance based on some basic physical characteristics. For example, the theory of coupled modes is used in designing effective Y-compounds based on photonic crystals [26] and waveguide bend curvatures with enhanced infocommunicational opportunities [27, 28, 29]. For instance, small uniform holes are drilled along the line defect or near the bends to convert the signal. Known frequency shifts (between different periods of crystal lattices) and different modes (of light beams propagation) lead to controlling band diagrams of crystal structures by moving parts of the lattice [30] or by inserting small holes in the middle of the waveguide to prohibit multimode light beam formation [31]. For that reason, for more efficient transmission of the light signal from the input waveguide to the output waveguides, resonant cavities are formed at the junctions with splitters [32] and at the waveguide bends [33]. In addition, intuitive geometric assumptions are also used to tune functional features in the structure. For example, to reduce the

curvature of the light beam inside the photonic crystal waveguide bend, a substance may be added or subtracted in particular design domain cells. Moreover, initially substance may be located at the lattice points [34, 35]. For most of the above applications, the details of geometric structure optimization, i.e. the size of the new holes, the lattice sparsity, and the exact cavity geometry are chosen empirically and are mainly determined by trial and error. Furthermore, the analytic physical quantities used for designing better geometry optimization prediction do not guarantee real performance improvement, although they are useful sometimes. For example, the transmission channel capacity, bandwidth, the reflected signals intensity and the intensity loss are very sensitive to small changes in geometry, which requires a more rigorous methodology of design process. The procedure also has no generality, which prohibits its further extension to more complex functional blocks. Moreover, there are more systematic measures. For example, instead of variation in the distribution of matter based on rough estimates of the physical quantities, sensitivity analysis can be used when small changes are applied to the positions, sizes or composition of the material of a particular lattice section, and the necessary characteristics of the device are evaluated accordingly [36]. This method quantifies the most influential geometric device parameters to which the target device characteristics are most susceptible. On the other hand, stochastic optimization (simulated annealing, evolutionary algorithms, etc.) used in order to find the optimal size or position of insertion or extraction of substances while optimizing structures of bends and waveguide connections splitters to enhance the transmission of photon waveguides (based on photonic crystal) [37] and frequency divider (based on photonic crystal) [38]. Although the search for a better structure with improved characteristics works efficiently, the number of target physical quantities (that are reference variables in stochastic optimization methods) is usually very limited. For the one-dimensional lattice design problems, these optimization methods are suitable if relatively few lattice layers are required [39]. For two-dimensional design problems (where geometry is more complex than their one-dimensional counterparts), optimization processes are often reduced to simplified structural optimization. Design variables are, f.i., the dimensions or material distributions of the lattice sites, but not the complete design area (where neither the boundaries of the objects nor their connectivity are *a priori* known). In addition, information about the complete topology prediction and its optimization is obtained by information accumulation of several of these design variables in the entire domain. Intuitively, we can assume that there is a better solution with a topology containing more irregular shapes than circular or rectangular holes. In 2004, Sigmund and Jensen proposed using topology optimization to optimize the curves of the photonic crystal waveguides [40]. By using the system algorithm, more optimal solutions were found. Such solutions contain topologies that are not limited to given shapes, such as a circle, rectangle, triangle, etc. Soon topology optimization was used to develop more photonic crystal waveguide devices [41, 42, 43, 44, 45]. Also, there are more application areas, including designing the geometry of photonic crystal cells to create opti-

mal bandgap (and bandpass) planar structures [46] as well as photonic crystal micro-resistors with high Q-factor [47]. An effective tool has been proven to optimize the design domain as part of the entire photonic crystal component in order to improve the performance of the instrument without degrading worsening the band properties of the original instrument.

Chapter 2

Problem statement and methods

With increased requests for modern topology optimization of photonic devices (such as power splitters and Fiber Bragg grating[48]) number of topology optimization numerical methods increases faster than number of strict theoretical approvements of them. However, experimental approvements of topology optimization plug up the the absences of theoretical approvements, for example, for Fiber Bragg grating[49]. In addition, topology optimization methods may be very easy-to-use [50]. The conventional optimization methods are ineffective in terms of computing time [51].

Mainly topology optimization methods such as homogenization method [11], solid isotropic material with penalization (SIMP) [52, 53, 54, 55, 56, 57], Level-Set method [58, 59, 60, 61, 62], evolutionary structural optimization (ESO) [63, 64, 65], bidirectional evolutionary structural optimization (BESO) [66, 67, 68, 69] and their various combinations [70] were designed to find the strongest structure under given constraints.

For example subtractive methods of topology optimization may use, for example, finite element method to obtain the target function values and trackable distributions that may be related to design domain Ω [63, 64]. The subtractive numerical methods of topology optimization are one-directional Evolutionary Structural Optimization methods [66, 67] that are aimed at reducing the device volume. In addition, BESO may be soft [71, 72] and hard [68]. In addition, BESO has improvements [73, 74, 75, 76, 77, 75, 71].

This research is aimed at, mainly, the development cost-effective numerical methods of BESO for devices that have constrains driven by market demand. In addition, the developed numerical methods are implemented by using the state-of-the-art programming on CUDA (C++) in order to speed up the computing procedure. The methodology will then be more generalized and extended to other functional material design needs. Offering innovative material design methodologies, in particular, the theoretical derivation of an improved gradient method to enhance the computational efficiency of the BESO numerical algorithmes to predict the optimal solution. To that end, new procedures for designing photonic materials with multifunctional properties will be introduced.

Furthermore, the considered nanometric scales ($> 1 \text{ nm}$) is much bigger than angstrom scale (0.1 nm). The angstrom scale is characteristic size of molecular structure of considered devices. Considered devices may be made, f.i., of glass. In addition, glass atoms may also be kept together

by physico-chemical bonds without any definable molecule. Therefore, the substance is assumed to be uniformly distributed and the internal atomic structure is negligible (more details about internal atomic physico-chemical processes is described in [78]).

2.1 Significance

During manufacturing new devices, we are facing the environmental challenges. For example, big devices consume a lot of energy and resources (such as oil, glass, quartz, etc.) and cause the carbon dioxide emissions. In addition, world resources are limited. Moreover, such pollution is the reason for global warming. That is why manufacturing smaller devices is necessary. Furthermore, one of the purposes of this research is to decrease the device domains ω by achieving the required characteristics of devices.

In addition, it is implemented by using the state-of-the-art topology optimization method that is Bi-directional Evolutionary Structural Optimization (BESO) method. They include minimizing structural volume with a displacement constraints [79, 72], optimization of the stiffness of the structure made of multiple materials [71], design of periodic structures [76], optimization of the structure frequencies [69, 72, 77], topology optimization for energy absorption structures [80, 68] and geometric and material nonlinearity problems [68, 80, 81, 76, 82].

Furthermore, the proposed developed BESO numerical methods are necessary to solve the problems of improving the accuracy of measurements with less time. In this work we treat photonic and nano-optical devices. These systems are described by Helmholtz equation. In topology optimization method it is extremely important (for fast computation of target function) to use as fast solver as possible. For this purpose we utilized ultra fast converging stable Green's function integral equation method (GFIEM) invented by Søndergaard [83]. Among other things, it produces more exact solutions for high values of complex relative permittivity, uses less memory (in comparison with grid methods) and has lowest cost complexity (because it is FFT-based).

2.2 Aims of the research work

- Development of a greedy numerical method for superlens using BESO.
- Development of gradient numerical method for superlens using BESO and its variations.
- Development of a BESO numerical gradient method for more generalized constrained problem to achieve the target distribution of the intensities under the restrictions on the cell locations of target distribution domain and design domain.
- Prototyping of the developed BESO numerical methods in Python.



Figure 2.1: Wave propagation

- Implementation of parallelization of the developed BESO numerical methods in CUDA C++

2.3 Methodology

We treat optimization problem on $L \mu\text{m} \times L \mu\text{m}$ square plane domain Ω while homogeneous device topology domain is $\omega \subset \Omega$. The source S_∞ is infinitely distant. An initial scattered field of incident plain waves F is given for each point ρ of vacuum space Ω ($\omega = \emptyset$). The number of wave periods placed in design domain Ω is $N = \frac{L}{\lambda}$, where λ is the wavelength of plane waves. $E(\rho) \equiv e^{i(k \cdot \rho)}$ is the propagated wave with wave vector k and a radius-vector ρ . It may be resolved by using Helmholtz equation [84]:

$$E(\rho) - k^2 \int \gamma(\rho - q)[\varepsilon(q) - 1]E(q) dq = F(\rho), \quad (2.1)$$

$$\gamma(\rho - q) = \frac{i}{4} H_0^{(1)}(|k||\rho - q|), \quad (2.2)$$

where γ is the Green's function, ε is the permittivity, F is the given electric-field vector distribution in a vacuum, $H_0^{(1)}$ is Hankel function of the first kind of zero order. Required scattered field E for device-filled Ω ($\omega \neq \emptyset$) is the solution to the Helmholtz equation (Fig. 2.1).

2.4 Suggested structural optimization

I suggest to conduct structural optimization of shapes for antennas, splitters, superlenses and devices of subwavelength imaging. Furthermore, the location of substance is called device domain ω . Moreover, the location of required intensity distribution is called target domain Φ . Device domain ω and target domain Φ are defined to be inside design domain Ω . In addition, the physical applications require specified positions of substance and required intensity distributions. That is why the structural optimization should be based on the constraints of locations of domains such as device domain ω and target domain Φ . In addition, numbers of device domains ω and target do-

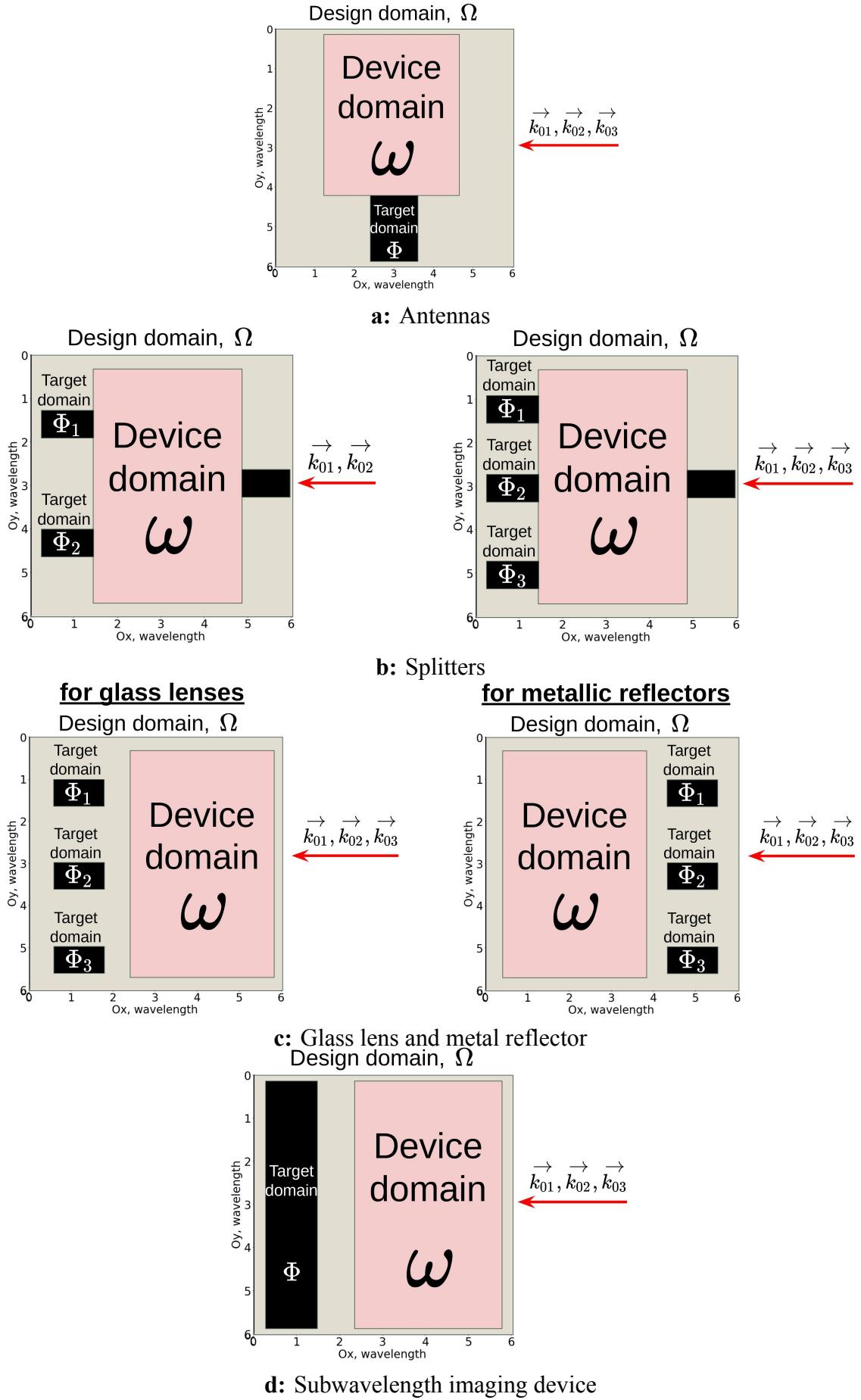


Figure 2.2: Examples of design domains

mains Φ may be more than one. However, device domains ω and target domains Φ are defined to be N_c -connected two-dimensional spaces, where $N_c \in [1, 3]$. Moreover, target domains Φ may be pre-filled with the designed device substance. However, the material distribution is not changeable in target domains Φ unlike in device domains ω .

The most frequent directions of waves are depicted as red arrows with captionings such as $\vec{k}_{01}, \vec{k}_{02}, \vec{k}_{03}$. The considered waves are the superpositions of N_m monochromatic plain waves which are specified by their vectors of initial (zero) waves $\vec{k}_{0i}, i = \overline{1, N_m}$. Initial wave propagation problem is the problem to find the electric field distribution E_0 in design domain Ω when device is absent. Thereafter, the electric field distribution E is found by using FFT-GFIEM (in A.1, A.2).

2.4.1 Antennas

Applications of antennas, for example, are 4G, 5G, Wi-Fi, Bluetooth, TV transmitters, radio-engineering, etc. According to application specifications of antennas, waves often are going to the device domain antennas perpendicularly to the device symmetry line. Device symmetry line is the line connecting centers of device domain and target domain Φ (Fig. 2.2a). Target domain Φ is filled with the device material. In addition, target domain Φ is the domain of distributed electric field intensity maximum. Moreover, the chosen material is metal with permittivity $\varepsilon = -10$. Thus, the refractive index n_r of a metal is imaginary unit.

2.4.2 Splitters

Splitters may be applied, for example, to obtain the light signals of separated polarizations TE/TM, wavelengths λ_i . According to application specifications of splitters, numbers of input and output junctions may be greater than one (Fig. 2.2b). All junctions (black domains in the picture) are filled with the device material. It is concerned with the device target to split combined light into separated lights of specific characteristics, for example, TE/TM-polarizations (the left picture of Fig. 2.2b), three monochromatic plain waves into separated monochromatic waves (the right picture of Fig. 2.2b), etc by maximizing intensity of specific expected lights in each target domain Φ_i and by minimizing intensity of rest (unexpected) lights in each target domain Φ_i . In addition, sets of expected and unexpected lights are different for different target domains Φ_i . Moreover, the chosen material is glass with permittivity $\varepsilon = 2.25$.

2.4.3 Superlenses

Superlenses may be applied, for example, in welding. According to their application specifications, the target domains Φ_i (Fig. 2.2c) should be located after device domain w.r.t. light direction propagation. The chosen material is, for example, glass. In addition, to create metal reflector, the

target domains should be before the design domain. In addition, target domains of superlenses and metal reflectors are not filled with device material. Moreover, target domains are locations of maximizing light intensities.

2.4.4 Subwavelength imaging devices

Subwavelength imaging devices may be applied, for example, for light painting such as portraits projected on walls. That is why their target domains (Fig. 2.2d) should be located after device domain w.r.t. light direction propagation. In addition, such devices are made of glass. Their target domain Φ is not filled with device material.

Chapter 3

Numerical formulation

The TO problem is to maximize the maximal intensity in 2D space by varying the mask of the 2D space m . According to (3.5), there are two optimization problem statements: for discrete ($T \equiv D$) and for continuous ($T \equiv C$) mask.

General formulation of this problem is non-convex because a lot of local optimums may be achieved in this statement because of big freedom degree of 2D space, but the convex problem may be formulated by using decreasing the freedom degree via heuristic constraints required for practical implementations to be devices. One of them is almost connected set. It means that result of this research should be mask to be one, two or three connected sets. In this research it was achieved by using neighbors bypassing idea. This idea is based on process of adding of neighbor mask pixels that corresponds to the best increasing of electromagnetic power in the pixel of maximal power electromagnetic power. More details in A.1, A.2, A.3.

For the numerical reasons it is more convenient to discretize the problem (2.1), (2.2). We apply homogenous discretization, namely we use:

$$r_{ij} = \rho(i, j), i, j = \overline{1, n}; \quad (3.1)$$

$$\omega = \{r_{ij}/M_{ij} = 1, i, j = \overline{1, n}\}; \quad (3.2)$$

$$\Omega = \{r_{ij}, i, j = \overline{1, n}\}; \quad (3.3)$$

2D distribution of material as a binary mask M for design domain Ω . M determines the presence of the substance in each pixel of the discretized design domain Ω .

We use a cost function to optimize the device shape determined by material distribution M . The cost function is the functional of the device quality, depending on the material distribution M . In case of wave devices such as lenses, parabolic reflectors or optical systems the cost function will be their optical power. The optical power is increasing with radiation intensity $|E|^2$ of required field distribution E at certain point r_{i^*, j^*} beyond device ($r_{i^*, j^*} \in \Omega/\omega$). i^*, j^* -th pixel corresponds to the maximum of radiation intensity distribution $|E|^2$. The solution to the problem is optimal binary mask M^* from the space of all possible masks T . Thus, we solve TO problem for target

function

$$\phi = \max_{M \in T} |E|^2(r_{i^*j^*}), \quad (3.4)$$

In order to resolve (2.1), we use stable FFT-GFIEM solver suggested by Søndergaard [83] which requires the same discretization approach. The FFT-GFIEM solver formulation has the following form for current discretization (more details in A.1):

$$\begin{cases} A(m) \cdot x(m) = f, \\ A(m) = I - k^2 \cdot (\varepsilon - 1) \cdot G \cdot \text{diag}(m), \\ x(m) = \text{vec}(E(r)), \\ f = \text{vec}(F(r)), \\ m = \text{vec}(M), \quad \tau = \text{vec}(T) \end{cases} \quad (3.5)$$

where vec is a linear transformation which converts the matrix into a column vector such that, for example, $\text{vec}(M) = [\mu_{1,1}, \dots, \mu_{s_1,1}, \mu_{1,2}, \dots, \mu_{s_1,2}, \dots, \mu_{1,s_2}, \dots, \mu_{s_1,s_2}]$ for $s_1 \times s_2$ matrix M ($s_1 \equiv s_2 \equiv n$); G is $p \times p$ matrix that corresponds to 4-dimensional cubic $n \times n \times n \times n$ Green's matrix tensor G^{4D} ($p \equiv n^2$), m is the p -dimensional vector to be vectorized 2D $n \times n$ mask M that corresponds to the presence of the substance in the 2D space that in this research case is square grid with sizes: $n \times n$: $M_{ij} = 1$ means the presence of the substance in the i, j -th pixel of grid, but $M_{ij} = 0$ means the lack of the substance in the i, j -th pixel. $m \in \tau$ is considered, where T in the LP problem with the respect to m is represented by continuous segments in p -dimensional space limited by zeros in the left and by ones in the right in each axis: $C \equiv [0, 1]^p$ is the p -fold Cartesian product, but in the ILP problem with the respect to m , T is represented by discrete values in p -dimensional space with zeros and ones in each axis: $D \equiv \{0, 1\}^p$.

The design problem is to determine the boundary of the design domain ω by maximizing objective function $|E(r_{ij})|$, $r_{ij} \in \Omega/\omega$. The TO method is based on the introduction of a fixed design domain Ω that includes the topology domain ω and on the utilization of the following characteristic function:

$$I(r_{ij}) = \begin{cases} 1, & \text{if } r_{ij} \in \omega \\ 0, & \text{if } r_{ij} \in \Omega \setminus \omega, \end{cases} \quad (3.6)$$

We distributed ε throw ω , ε is fixed. This function appeared in one of the first papers on TO [85, 86, 87, 88].

Chapter 4

Methods to solve the problem

4.1 Development of BESO numerical methods

4.1.1 Greedy

The main idea of greedy numerical BESO in this research is exhaustive search. For example, for additive method we should implement exhaustive search of the most neighbour bit (that is not filled with material) of current device material which contribution to the increase in intensity is the most maximal after addition of substance in that location location. For subtractive algorithm, similar exhaustive search should be implemented but it should consider device material bits which contribution to the increase in intensity is the most maximal after subtraction of substance from that location.

4.1.2 Discrete gradient

The main idea of discrete gradient is to use gradient derived in A.4. The idea is similar to greedy BESO. However, it would be definitely the same if we will consider intensity gradient values instead of intensities and exhaustive search. Thus, we will subtract the material bits where intensity gradient is negative. In addition, we will add the material bits in the neighbours to material where intensity gradient is positive.

4.1.3 Steepest descent

The idea of steepest descent is almost the same as well as the discrete gradient BESO. However, the addition and subtraction is where the maximum and minimum of the gradient. In addition, operation is feasible only one by one adding or subtracting device material bit.

4.2 CUDA (C++) parallelization

One of the main problem of the BESO is slow computations with big matrices. In addition, cuFFT (Nvidia) suggest speedup bigger than any another parallelization architectures for matrices bigger

that 512×512 . That is often used range of the actual problem. That is why we should implement parallelization on GPU.

Chapter 5

Results

Current software version is published on <https://github.com/archilless/projectwave>. Outcome is accepted scientific article [89].

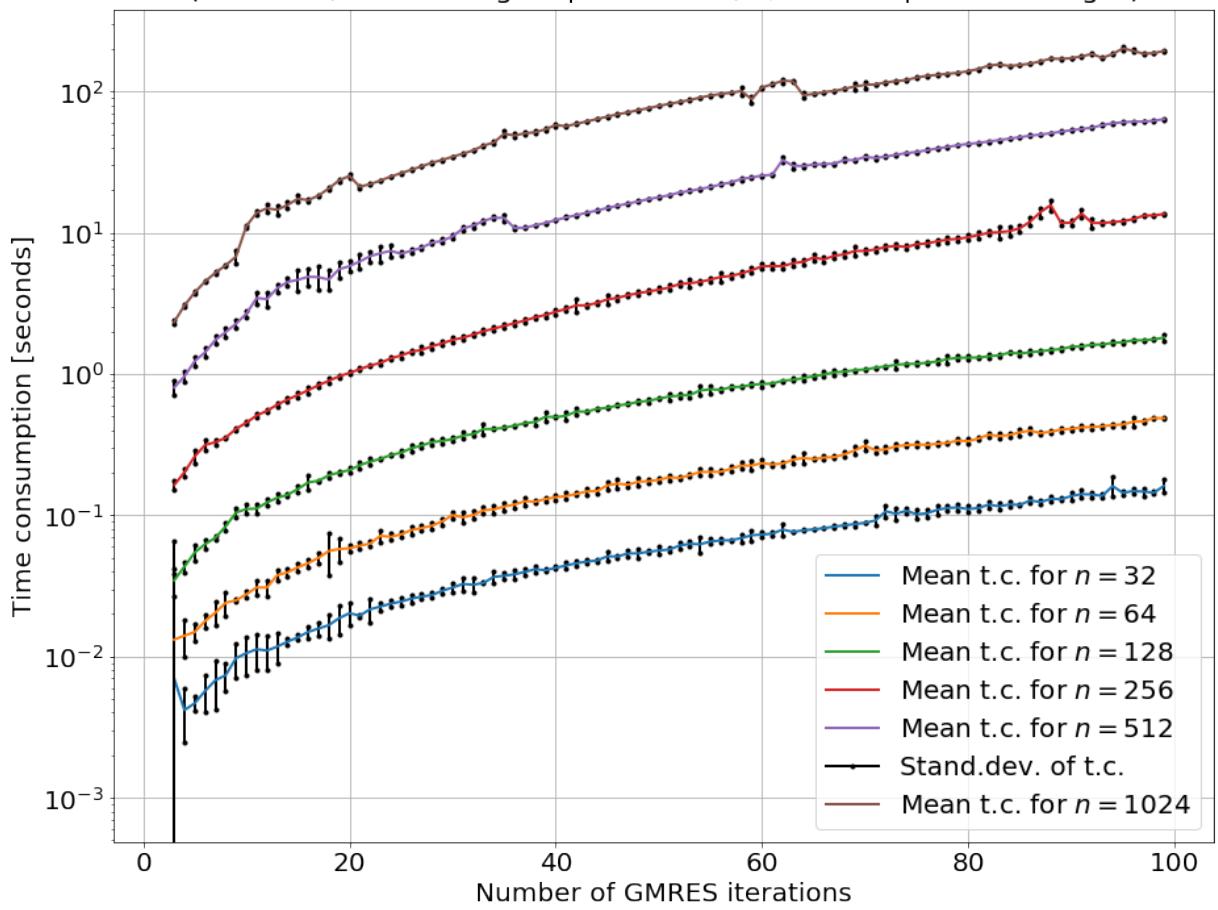
5.1 CUDA (C++) parallelization

The parallelization results are represented in the figures 5.1, 5.2, 5.3. Input binary mask of all parallelization experiments is the cylinder shown in the upper left graph 5.5. Numerical experiments are conducted with 6 resolutions $n \times n$, where $n = 32; 64; 128; 256; 512; 1024$. In addition, time consumption is the time that is spent to compute electric field distribution for cylinder by using GMRES to solve linear equation system built by FFT-GFIEM (more details in A.1, A.2). The numerical experiment is timing full GMRES of the given iterations. The figures describe computing time consumptions of GMRES on CUDA (C++) and Numpy (Python) and relative time consumption (t.c.) expressed by speedup calculated by dividing serial program t.c. (on Numpy) by parallel program t.c. (on CUDA). Measurements of each point of figure 5.1 are results of 100 program execution repetitions per each point of figures 5.1. That is why the graphs represents standard deviations (stand.dev.) and means according to refined (without points of big deviations) statistics based on 100 program execution repetitions.

Figure 5.1 on the top is Numpy t.c. versus number of GMRES iterations. With doubled n Numpy GMRES t.c. increases by approximately half-order for all considered n . Moreover, Numpy GMRES t.c. trend is increasing. Furthermore, standard deviations are on average less than 1% of mean values of t.c.. In the figure t.c. curve $n = 32$ is increasing more than by one order during all considered numbers of GMRES iterations. However, such increase grows with doubled n . For example, GMRES t.c. curve $n = 1024$ is increasing by almost two orders during all considered numbers of GMRES iterations. In addition, during all considered numerical experiments represented on the figure GMRES t.c. are changing approximately between 10^{-3} and 10^2 seconds. Such deviation accounts for approximately 5 orders.

Figure 5.1 on the bottom is CUDA (C++) t.c. versus number of GMRES iterations. Main distinctive feature is that CUDA (C++) GMRES t.c. curves are the same in range of number of GMRES iterations between 40 and 99 for $n = 32; 64; 128; 256$. Moreover, the curves $n = 32; 64; 128$ are the

Time consumption of GMRES on Numpy (Python)
 $(n^2$ nodes, 6 wavelengths per domain, $n/6$ nodes per wavelength)



Time consumption of GMRES on CUDA (C++)
 $(n^2$ nodes, 6 wavelengths per domain, $n/6$ nodes per wavelength)

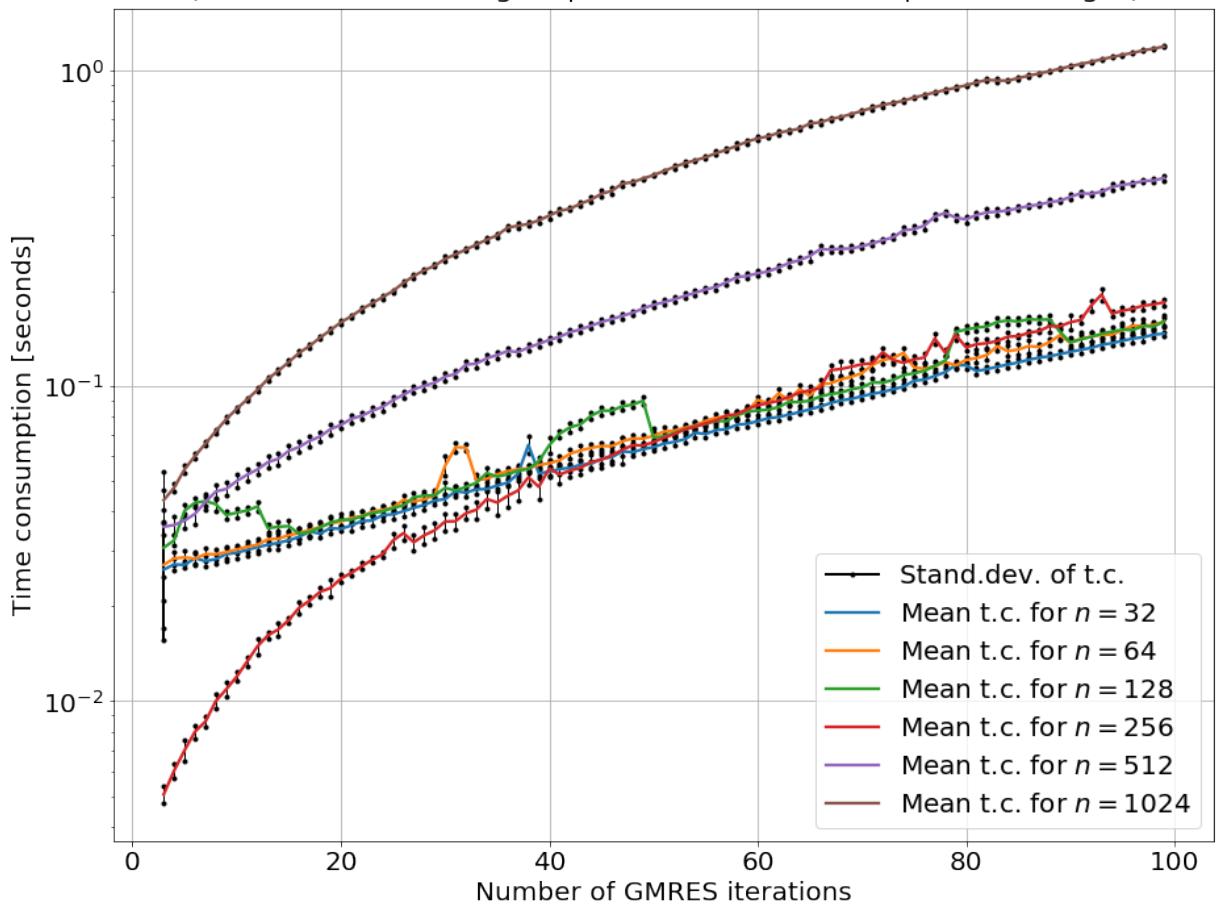


Figure 5.1: Parallelization results (a)

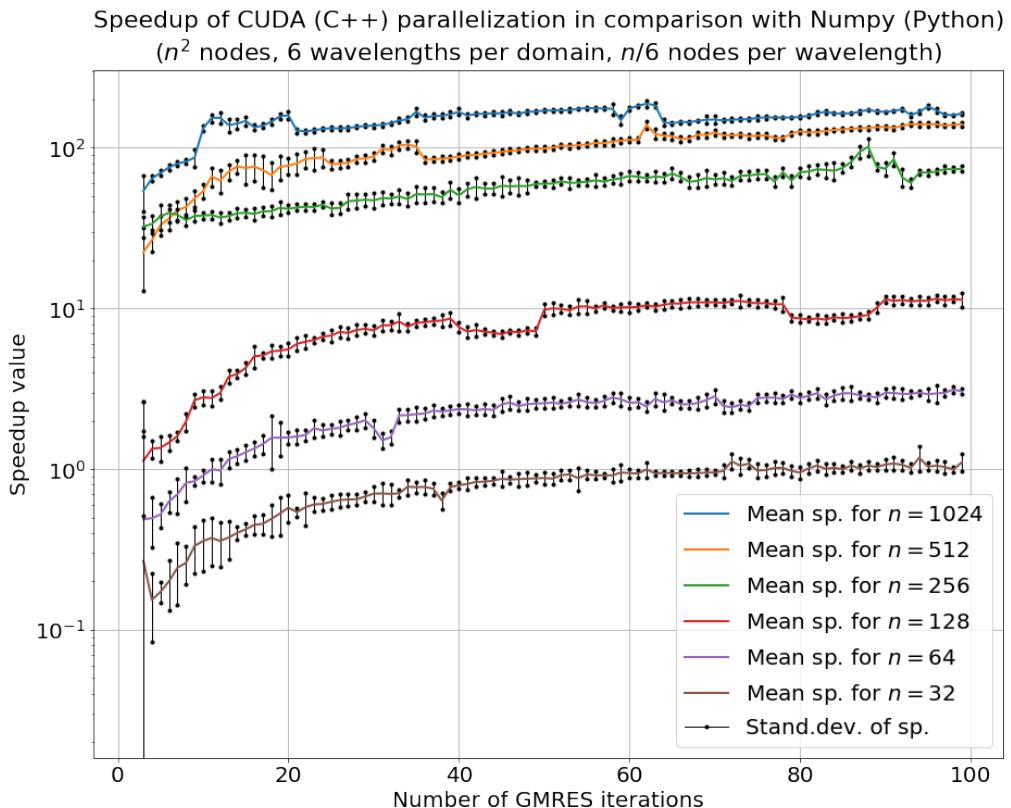
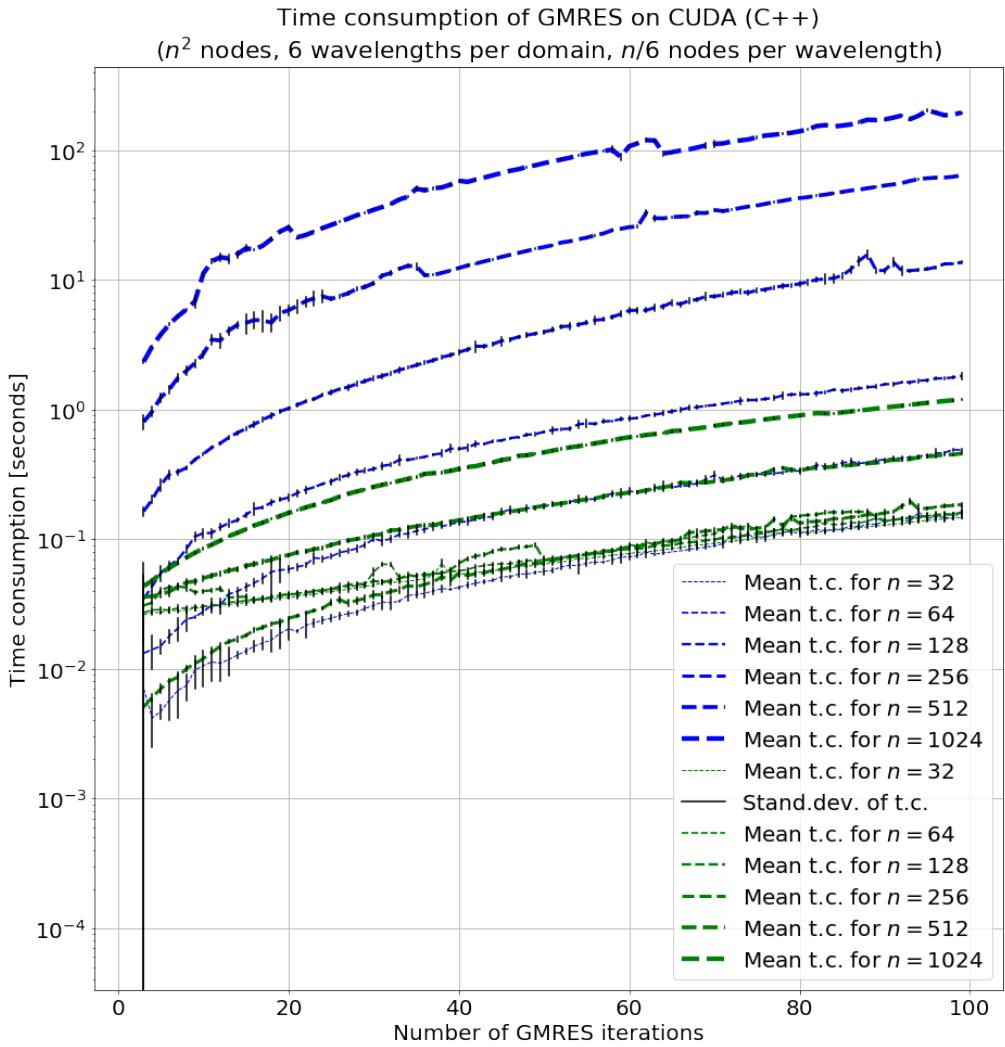


Figure 5.2: Parallelization results (b)

Speedup of CUDA (C++) parallelization in comparison with Numpy (Python)
 $(n^2$ nodes, 6 wavelengths per domain, $n/6$ nodes per wavelength)

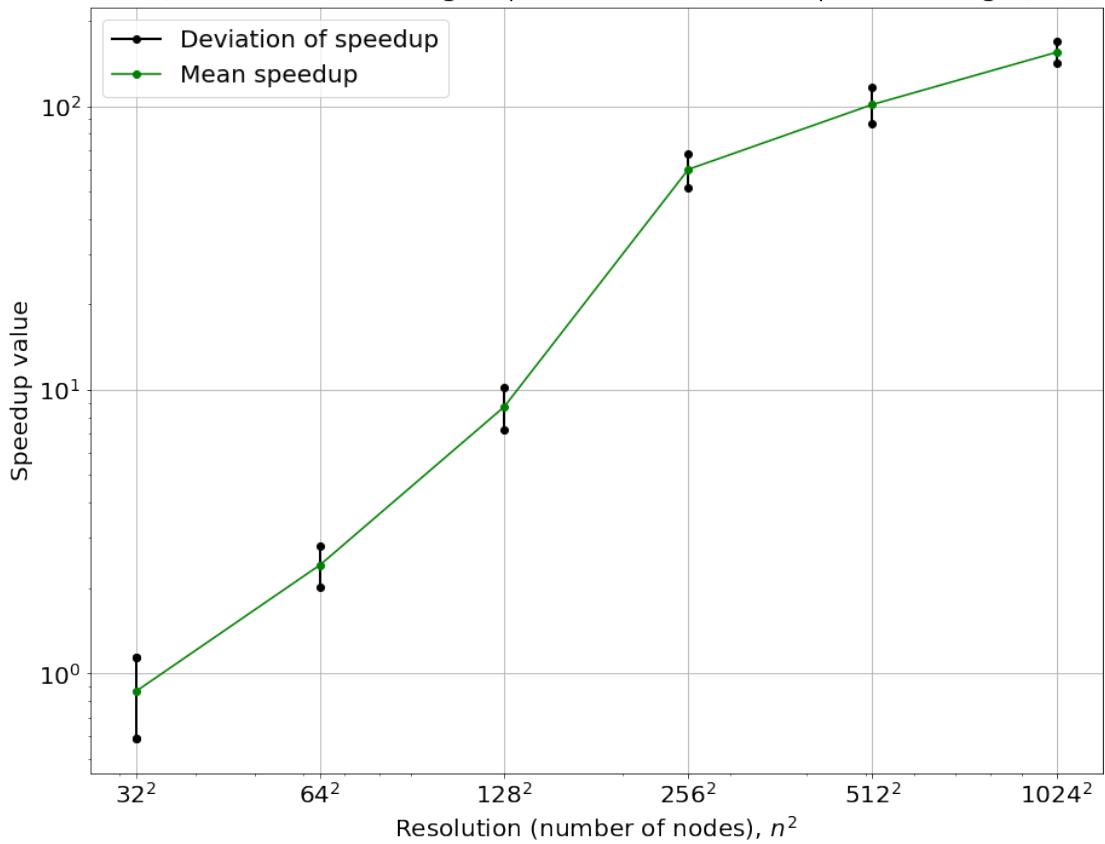


Figure 5.3: Parallelization results (c)

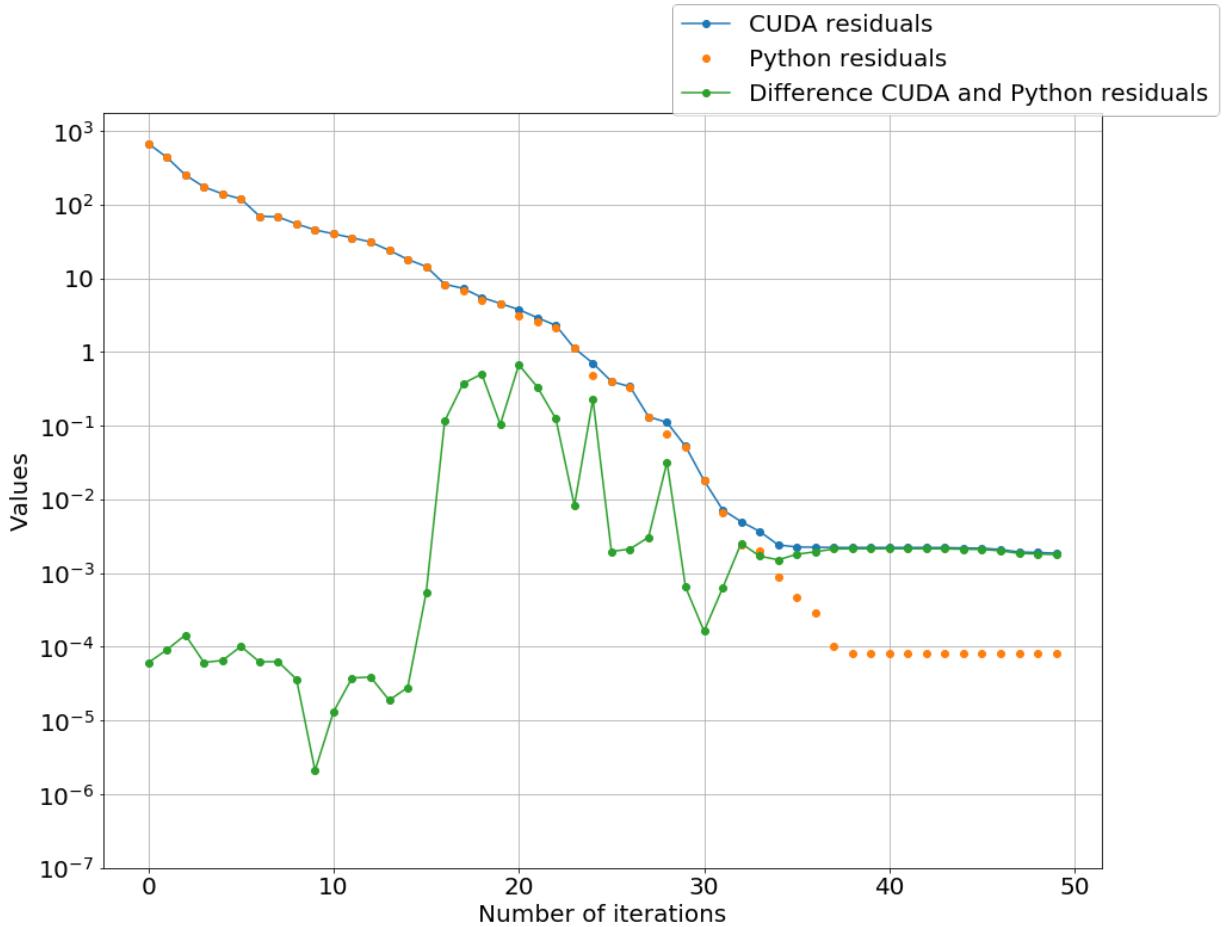


Figure 5.4: GMRES (for FFT-GFEM) residuals on CUDA and Numpy for $n = 1024$

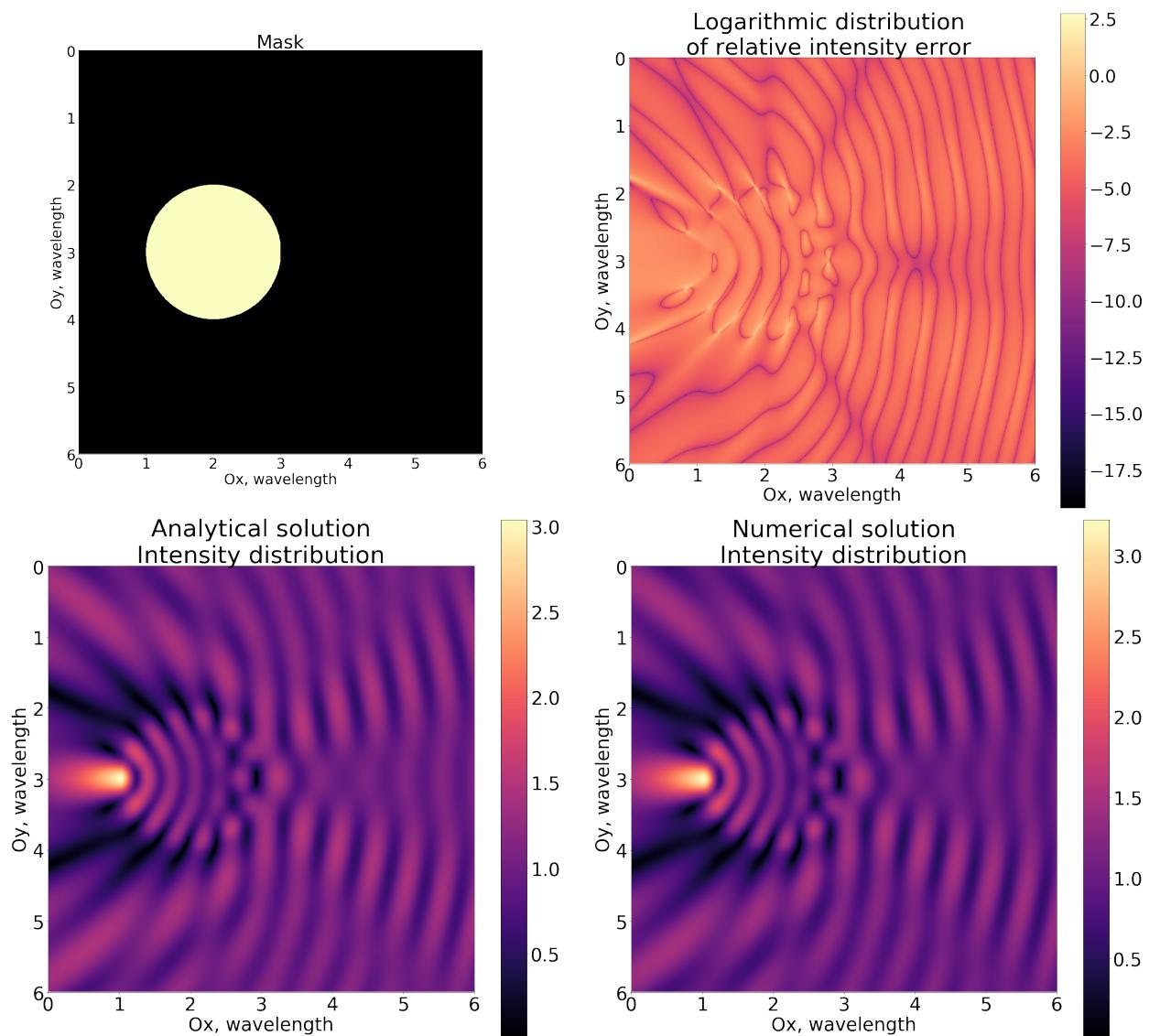


Figure 5.5: Comparison of CUDA FFT-GFIE solution to cylinder scattering problem with analytical one

same during all considered range of number of GMRES iterations (subject to neglecting the modulation of curve $n = 128$). The modulation of curve $n = 128$ is the result of algorithmic adjustment of program that has 5 switches (to restart CUDA by using “*cudaDeviceReset()*” and “*cudaSetDevice()*”) during 100 repetition to measure each point. However, the difference in values of curves $n = 256; 512; 1024$ is less than half-order. In addition, difference of values in their starting points is the order of 10^{-2} . Furthermore, the standard deviation is approximately 1% exactly the same as for Numpy. In addition, all considered CUDA GMRES t.c. curves are approximately between 10^{-2} and 1 seconds. Such deviation accounts for approximately 2 orders.

Upper figure 5.2 represents the combination of upper graph (in blue) and lower graphs (in green) of figure 5.1. According to the figure 5.2, mean Numpy GMRES t.c. curve $n = 64$ is above all considered CUDA GMRES t.c. curves. It means that, for example, CUDA GMRES to compute the field distribution of 1024×1024 resolution is faster than Numpy GMRES to compute the field distribution of 64×64 resolution.

More details on computational speedup (sp.) is represented in the lower figure 5.2. It is graph of the parallelization speedup curves dependent on characteristic data size n versus number of GMRES iterations. This graph is the calculated figure based on data obtained in graphs 5.1. According to this figure, with doubled n speedup curves goes upper. In addition, the curve $n = 32$ has speedup equal not bigger than 1. It means represented CUDA parallelization for $n = 32$ is inefficient. In addition, curve $n = 64$ has speedup values below 1 when number of GMRES iterations is less than 10. However, curve $n = 64$ has speedup values definitely above 1 when number of GMRES iterations that are more than 20. That means that CUDA parallelization (for curve $n = 20$) is successful only for more than 20 GMRES iterations. However, the remaining curves such as $n = 128; 256; 512; 1024$ have speedup values that are on average definitely bigger than 1. Deviations of speedup values of all considered curves account for approximately 1 order. Standard deviations of all curves is on average approximately less than 1%.

Details on total computational speedup dependant on characteristic size n of resolution is represented in the figure 5.3. According to the figure, speedup value growth decreases with an increase in resolution starting with $n \times n = 128 \times 128$. The speedup of represented figure varies from approximately 1 to a little more than 100.

5.2 FFT-GFIEM validation

The FFT-GFIEM validation is implemented by using comparison of FFT-GFIEM solution to the cylinder (upper left figure 5.5) with analytical one (lower left figure 5.5). The analytical solution MieField (<https://github.com/RI-imaging/MieField>) is implemented on Python. The represented numerical solution (lower right figure 5.5) is the solution of 1024^2 nodes, $N = 6$ wavelengths

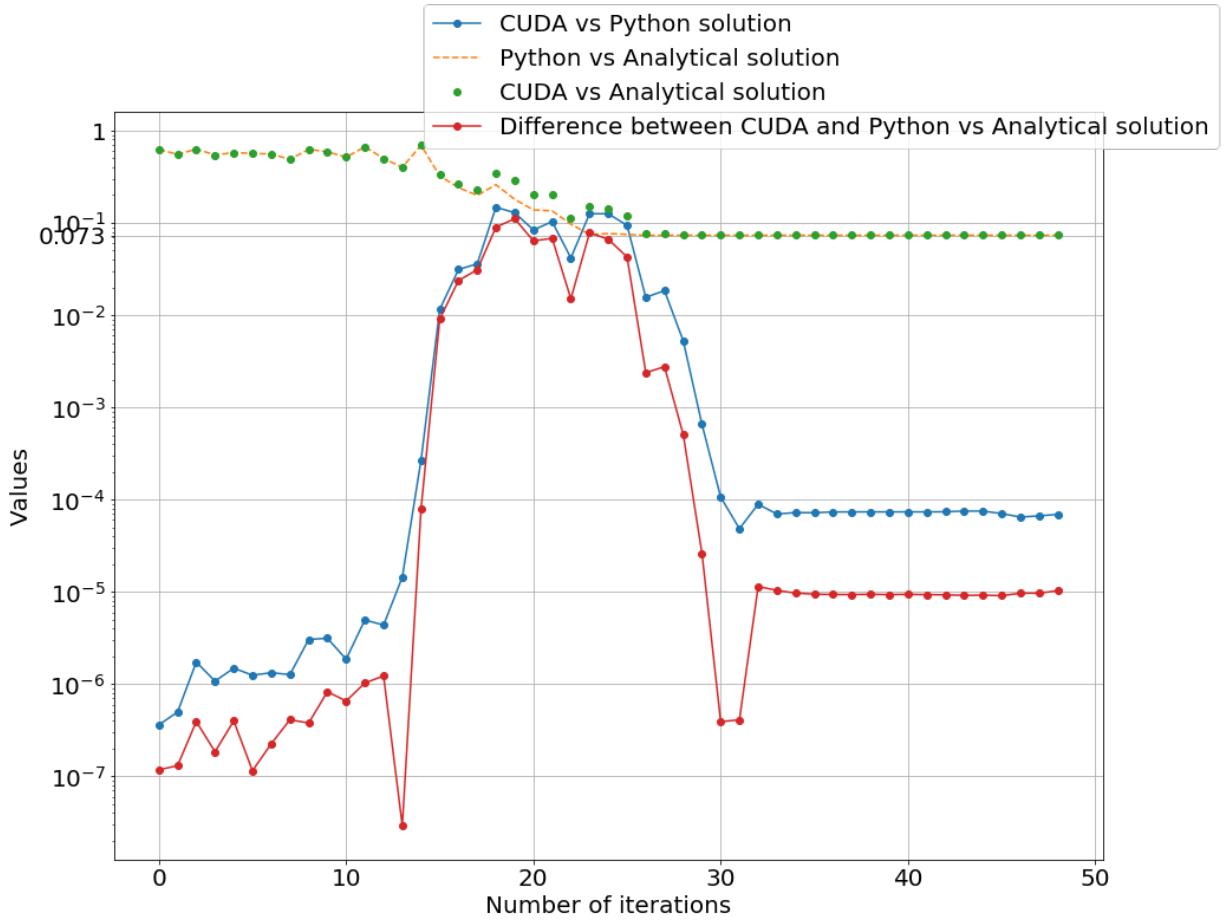


Figure 5.6: FFT-GFIEM relative errors versus number of GMRES iterations for $n = 1024$
 Numpy (Python) GMRES solution errors relative to analytical solution (MieField)
 $(n^2$ nodes, 6 wavelengths per domain, $n/6$ nodes per wavelength)

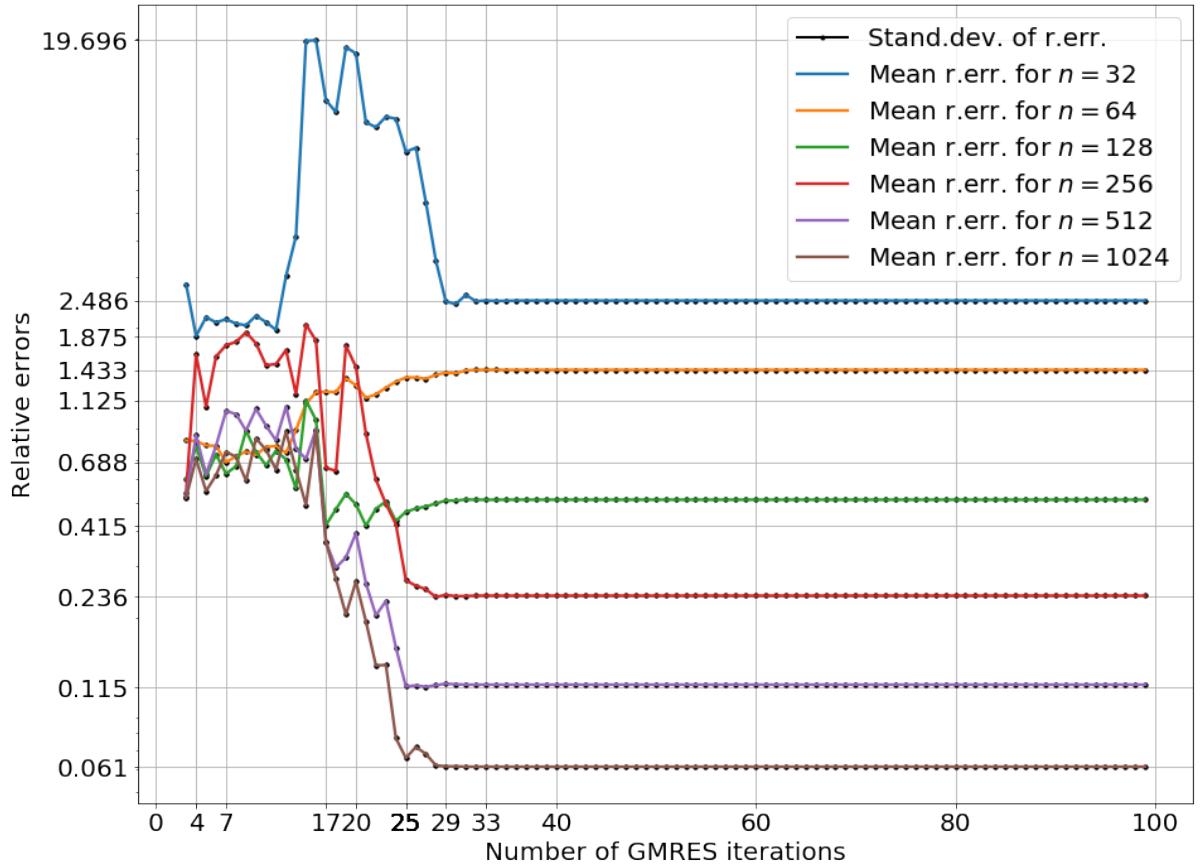


Figure 5.7: FFT-GFIEM relative errors on Numpy versus number of GMRES iterations

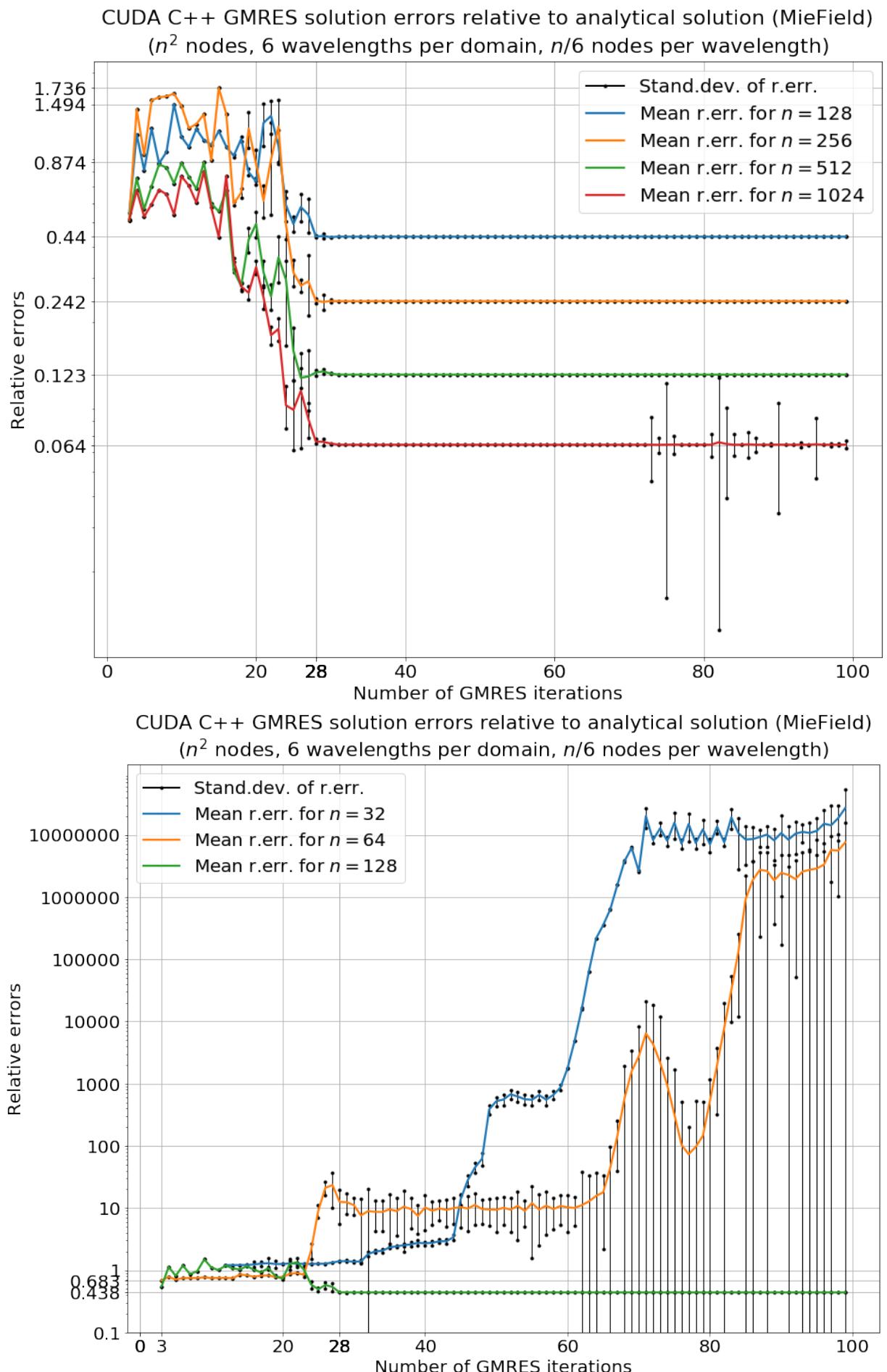


Figure 5.8: FFT-GFIEM relative errors on CUDA versus number of GMRES iterations

per design domain Ω (≈ 170.7 nodes per wavelength). Input GMRES tolerance is equal to 0.01, input maximum of GMRES iterations equals to 50. At first sight, the analytical solution coincides with numerical one. However, distribution of logarithm of relative errors of intensity (upper right 5.5) shows that relative errors of intensity is between $e^{-17.5} \approx -47.6$ and $e^{2.5} \approx 5.4$. The biggest error are located on the matches between edge of material and interference maximum. However, according to figure 5.8 the achieved FFT-GFIEM solution relative to analytical solution accounts for 6.4% that is much less than 100% and is good intensity measurement precision in order to use such computational apparatus to compute target function expressed by the electric field intensity distribution in design domain Ω .

The figure 5.6 represents the FFT-GFIEM solution errors relative to the analytical solution versus number of GMRES iterations for 1024×1024 resolution, when the number of program execution repetition to measure each point of the actual figure is equal to 1. That is why standard deviations are absent in the figure. The real points of number of GMRES iterations are shifted by “+3” w.r.t. the actual axis “Number of iterations”. Thus, number of GMRES iterations varies from 3 to 50. In addition, “Python vs Analytical solution” and “CUDA vs Analytical solution” curves represent the Numpy and CUDA FFT-GFIEM solution errors relative to analytical solution respectively. The CUDA solution relative errors has decreasing trend from 3 GMRES iterations to approximately 29 GMRES iterations. The relative error values of the curve are changing between ≈ 0.073 and ≈ 0.8 . Moreover, the curve is flat from approximately 29 GMRES iterations to end. The value of the flat part of the curve is 0.073. That means the solution has error equal to 7.3%. Such error is much less than 100%. Furthermore, curve of Numpy FFT-GFIEM errors has decreasing trend and flat part in a very similar manner to the CUDA FFT-GFIEM curve of relative errors. However, flat part of the Numpy curve starts with approximately 25 GMRES iterations. That means implemented program on CUDA my have approximately 10% more GMRES iterations in order for the solution to be converged to the same required precision relative to analytical solution. Moreover, the absolute difference between CUDA and Numpy relative errors is represented as “Difference between CUDA and Python vs Analytical solution” curve. The difference is approximately within 10^{-7} and 10^{-1} . It means that CUDA and Numpy solutions are always $\geq 90\%$ the same in terms of closeness to analytical solution. Moreover, the difference curve trend is increasing between 26 and 33 GMRES iterations. Furthermore, the difference values are observed to be equal to < 0.01 starting with 29 GMRES iterations. It means that CUDA and Numpy solutions are $> 99\%$ the same in terms of closeness to analytical solution when they have number of GMRES iterations equal > 29 . In addition, it is flat between 35 and 50 GMRES iterations. The value of flat part of the difference is 10^{-5} . Furthermore, “CUDA vs Python solution” curve represents CUDA FFT-GFIEM solution errors relative to Numpy FFT-GFIEM solution. The relative error curve trend is decreasing starting with 26 GMRES iterations in a very similar manner to the difference curve.

However, the order of the value (equal to 10^{-4}) of flat part of the relative error curve is one order higher than the flat part of the difference curve. Moreover, all the relative error curve is higher than the difference curve. It means that CUDA and Numpy solutions are closer one to another in terms of closeness to analytical solution.

The figure 5.7 represents the Numpy FFT-GFIEM solution errors relative to the analytical solution versus number of GMRES iterations for $n = 32; 64; 128; 256; 512; 1024$, when the number of program execution repetition to measure each point of the actual figure is equal to 100. That is why means relative errors (r.err.) and standard deviations are presented in the figure. Standard deviations of relative errors are significantly smaller than 10^{-5} . Values of standard deviations of relative errors divided by means of relative errors are significantly smaller than 10^{-3} . It means that standard deviations of relative errors are negligible. All curves represented in the figure are flat starting with 29 GMRES iterations. In addition, the values of flat parts approximately doubles with doubled n . Moreover, curve $n = 32$ has all values equal > 1 . It means that solution to cylinder scattering of 32×32 resolution ($N \approx 5.3$ nodes per wavelength λ) is $> 100\%$ incorrect. In addition, a trend of the curve $n = 512$ is increasing. In addition, its value of flat part is equal to 1.433. Moreover, its values are within 0.688 and 1.433. Consequently, $N \approx 10.7$ nodes per wavelength gives the solution that is $> 68.8\%$ incorrect. Trends of curves $n = 128; 256; 512; 1024$ are decreasing. Values of their flat parts are $n = 0.43; 0.236; 0.115; 0.061$ respectively. It means that after 29 GMRES iterations solutions of $N \approx 21.3; 42.7; 85.3; 170.7$ nodes per wavelengths are 57%; 76.4%; 88.5%; 93.9% correct respectively.

The figures 5.8 represent the CUDA FFT-GFIEM solution errors relative to the analytical solution versus number of GMRES iterations. Trend of curves $n = 32; 64$ is increasing. Their values are bigger than 0.683. It means that solutions $N \approx 5.3; 10.7$ nodes per wavelengths are $> 68.3\%$ incorrect in a similar manner to curves $n = 32; 64$ of the figure 5.7. For curves $n = 128, 256, 512, 1024$ in the figure 5.8 values of standard deviations relative errors divided by means of relative errors are on average of the order $\approx 1\%$. It means that standard deviations of relative errors are negligible. Trends of curves $n = 128; 256; 512; 1024$ are decreasing. Values of their flat parts are $n = 0.44; 0.242; 0.123; 0.064$ respectively. It means that after 28 GMRES iterations solutions of $N \approx 21.3; 42.7; 85.3; 170.7$ nodes per wavelengths are 56%; 75.8%; 87.7%; 93.6% correct respectively. The difference in precisions of respective curves illustrated in the figures 5.7 and 5.8 is less than 2%.

The figure 5.1 represents GMRES residuals of solution to linear equation system given by FFT-GFIEM. Three curves of figure represent the residuals of GMRES implemented on CUDA (“CUDA residuals”), Numpy (“Python residuals”) and difference between these residuals (“Difference CUDA and Python residuals”). The numerical experiments were conducted by using 1 program execution repetition. That is why standard deviation is absent in the figure. This figure is

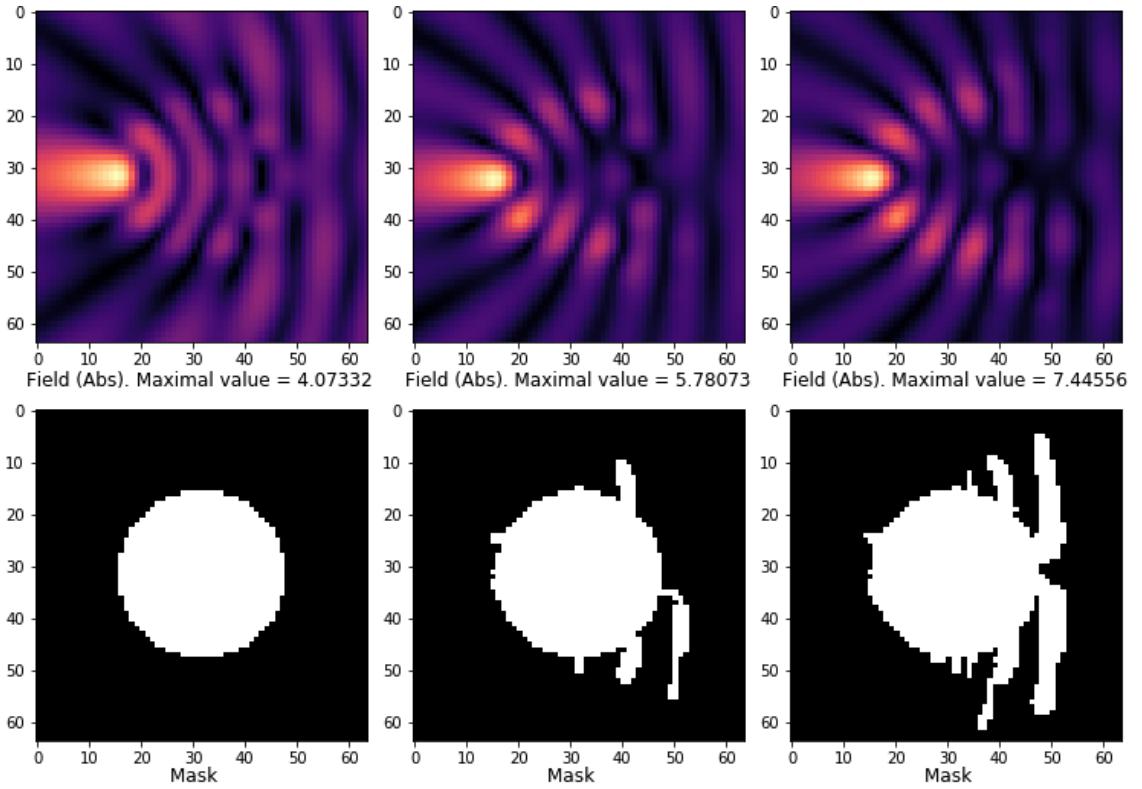


Figure 5.9: Optimization results (a)

directly connected to the figure 5.6. The real points of number of GMRES iterations are shifted by “+3” w.r.t. the actual axis “Number of iterations”. Thus, number of GMRES iterations varies from 3 to 50. The Numpy and CUDA curves in the figure 5.1 coincide one with another in the range of GMRES iterations between 3 and 34. Trend of both curves is decreasing. Flat parts of CUDA and Numpy curves starts with 37 and 42 respectively. Order of values of their flat parts are -3 and -4 respectively. The difference curve illustrate the absolute error of CUDA GMRES relative to Numpy GMRES. Values of difference curve vary between 10^{-6} and 1. The difference curve trend is decreasing. A flat part of the difference curve starts with 35 GMRES iterations. Values of flat parts of difference and CUDA curves coincide one with another because CUDA GMRES residuals significantly differ from Numpy GMRES residuals. That means that relative error of CUDA and GMRES residuals is $\approx 100\%$ in flat part of CUDA curve. However, relative errors in the flat part in terms of analytical solution and relative errors of solutions (represented in 5.6) are orders of 10^{-5} and 10^{-4} that is negligible. That shows GMRES residuals may give incorrect information about the relative errors between CUDA and Numpy solutions.

5.3 Python BESO

The optimization results are represented in figures *a – f* (Fig. 5.9, 5.10, 5.11, 5.12, 5.13, 5.14). We used $N = 2$ (*c, d, f*), 3 (*e*), 4 (*a, b*) wavelength per design domain Ω and characteristic grid

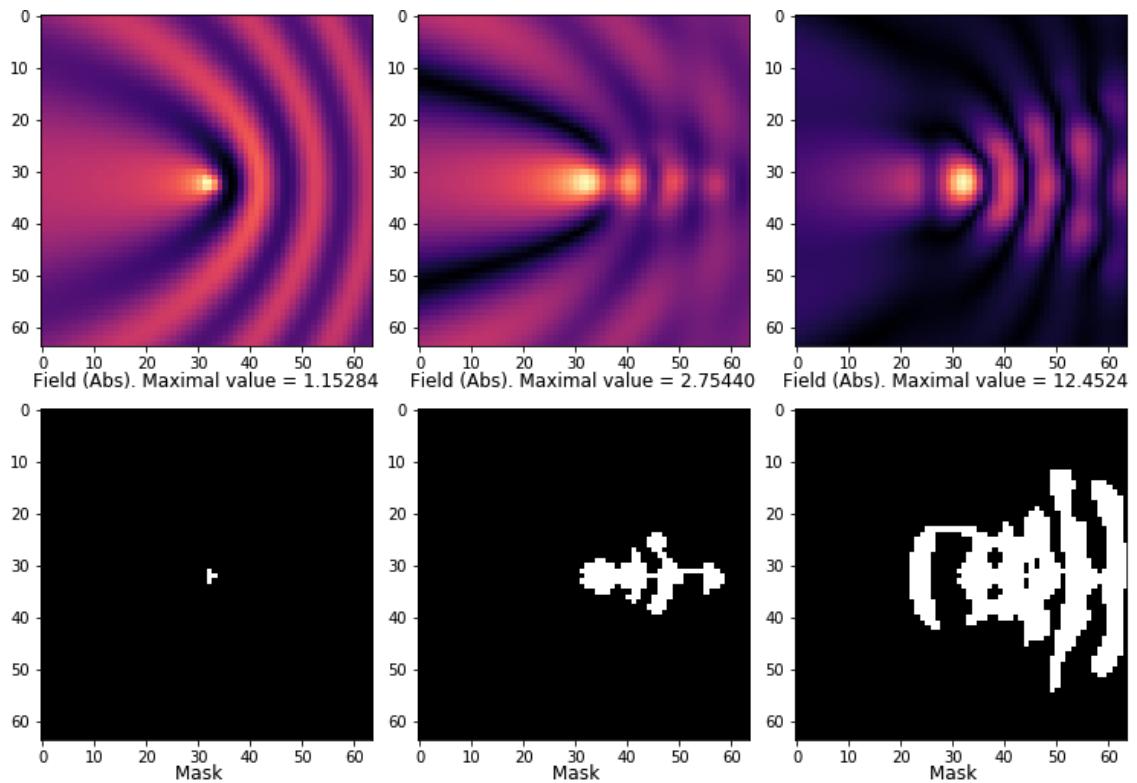


Figure 5.10: Optimization results (b)

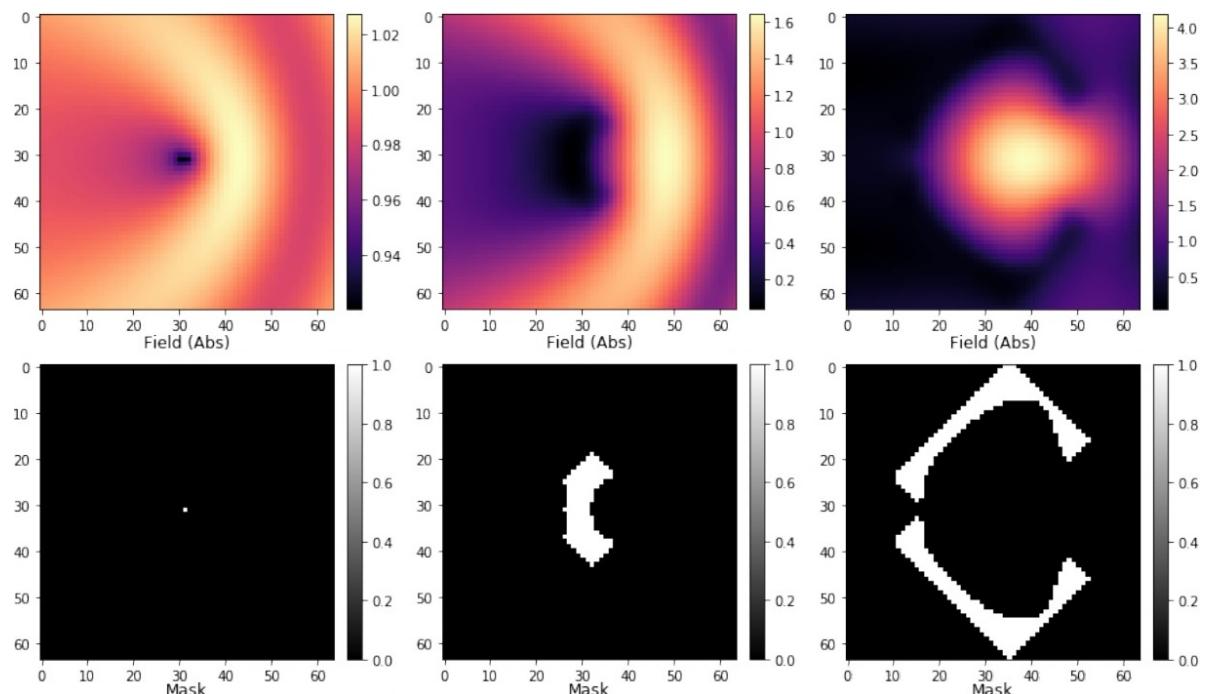


Figure 5.11: Optimization results (c)

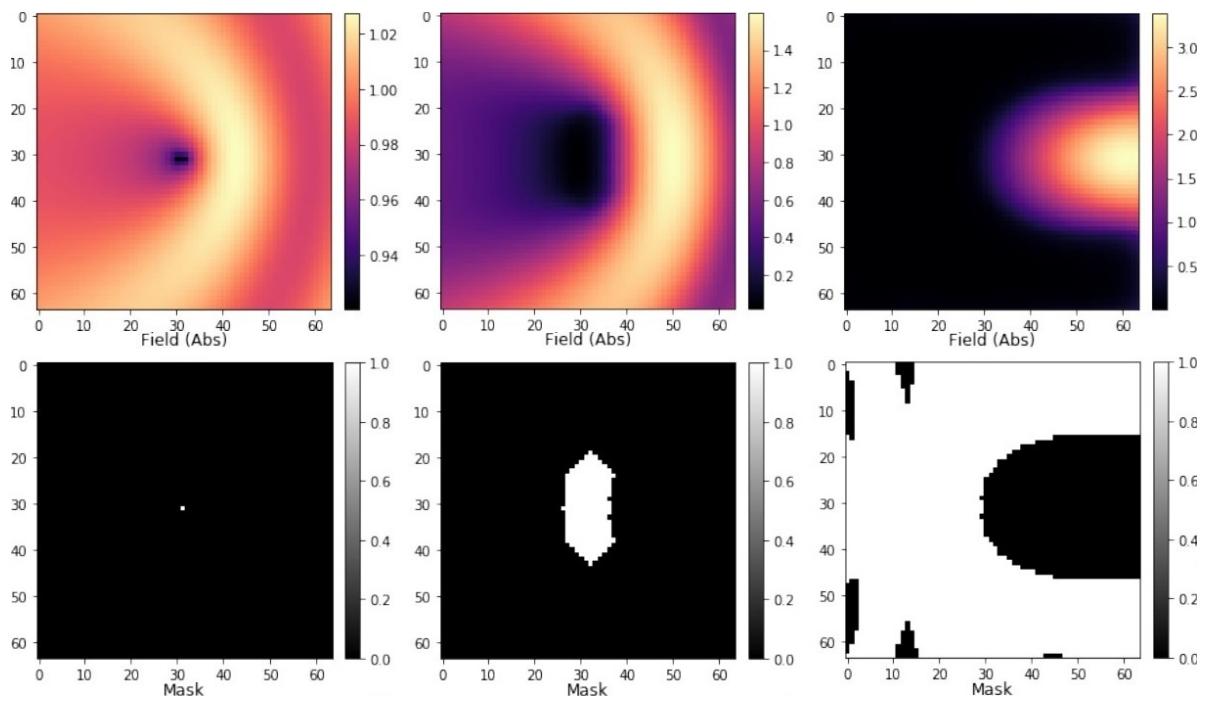


Figure 5.12: Optimization results (d)

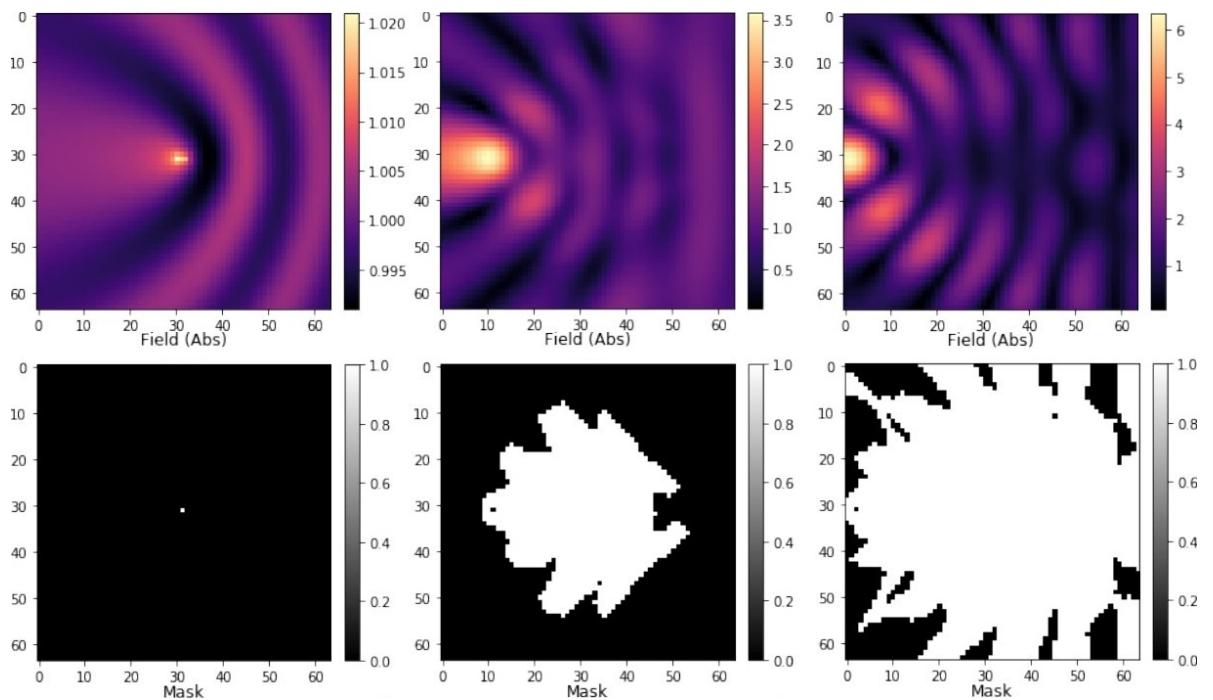


Figure 5.13: Optimization results (e)

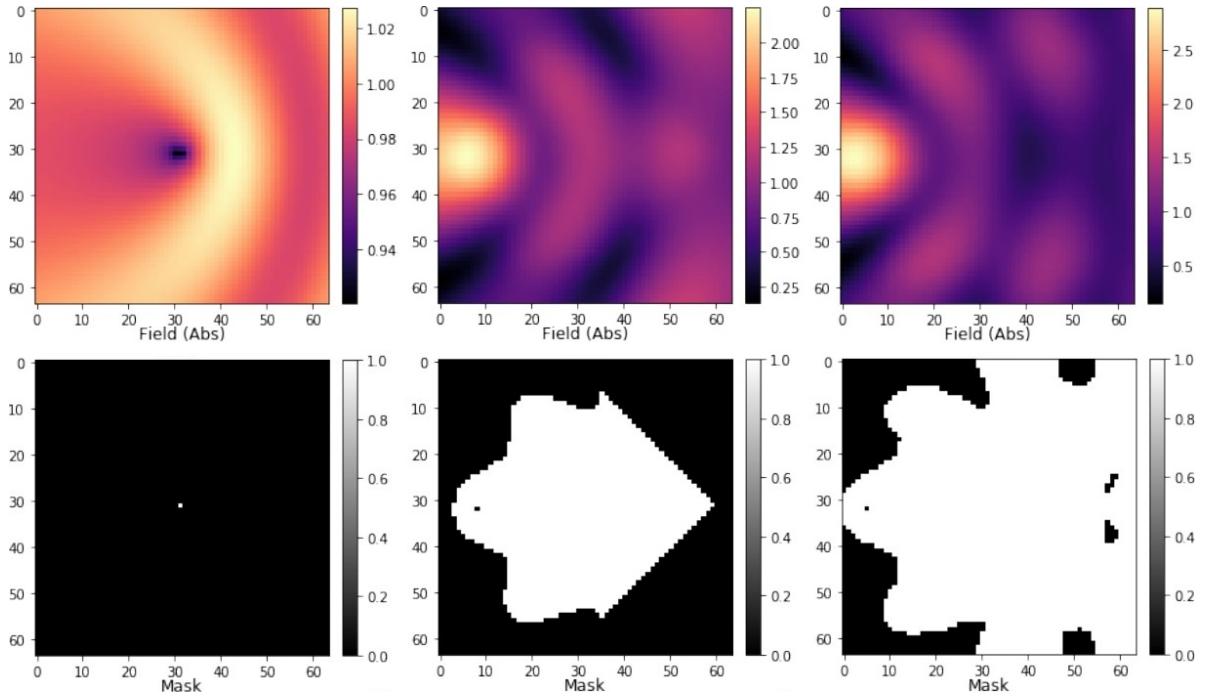


Figure 5.14: Optimization results (f)

size $n = 16$. We considered discrete case $T = D$. Magma colormapped figures shows the field distribution corresponding to the black-white figures of material masks below in neighborhood. White pixels matches the substance presence while black pixels matches the lack of the substance.

We implemented greedy method for glass lenses (*a, b*) and discrete gradient method for steel reflectors (*c, d*) and glass lenses (*e, f*).

In all field distributions figures the wave front is approximately convex right that is conditioned by phase delays when waves pass through substances. The delays are increasing with the substance width.

Material masks obtained by greedy method are less symmetrical than material masks obtained by discrete gradient method. It is associated with the target function symmetry distortion induced by pixel-by-pixel addings in greedy method. However, the distortion is less while several pixels are added that is used in the discrete gradient method.

The main characteristic curve of the result for evolutionary structural optimization in steel device (*d*) is parabola that corresponds to the widely known parabolic reflectors. However, this BESO did not use the option to subtract pixels. The BESO with this option was used in (*c*). Thus, more compact device was generated with target function value increased by $\approx 25\%$.

Some resulted masks contains small inclusions of substance lacks. In addition number of them is increasing with N (*e, f*). The biggest problem of discrete gradient method was erroneous surrounding i^*, j^* -th pixel by substance that limited the growth rate of target function. That is why the most values of target functions were obtained by using greedy algorithm. However, computing time taken for greedy method was from 6 hours to 27 hours while discrete gradient method took

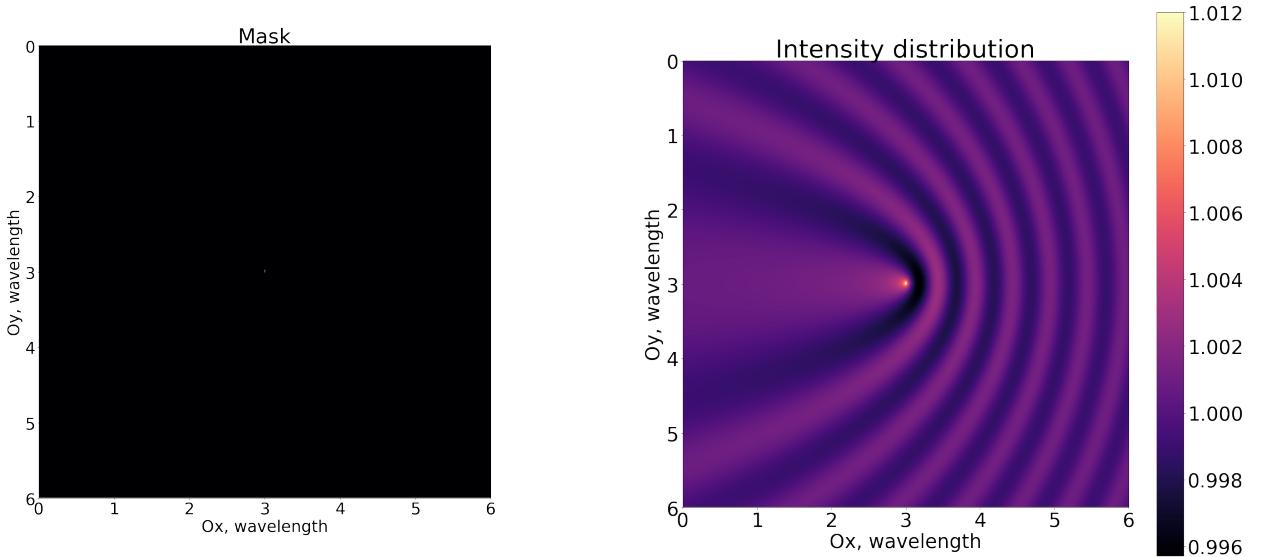


Figure 5.15: Initial design domain Ω that contains centered 4×4 device domain ω ($n = 1024$; $N = 6$ wavelengths per design domain Ω)

from 20 minutes to 50 minutes.

5.4 The developed ESO and BESO numerical methods and CUDA implementations

Figures 5.15 represent possible initial mask and intensity distribution for the device given by this mask. However, there are possibilities to choose another initial mask to start ESO and BESO processes, for example, circular mask (lower left figure 5.9), mask of biconvex lens (upper left figure 5.23), etc.

5.4.1 Greedy

Figure 5.16 represents result of greedy numerical method for ESO of glass lens shape. The resolution is 1024×1024 nodes². $N = 6$ wavelength per domain. ≈ 170.7 nodes per wavelength λ . Initial mask is in the figure 5.15. The focus is located at node point [5, 512]. Target function is the intensity at the focus point. Initial focus intensity equals to 1.00. The ESO progress is described by masks (in the left) and respective intensity distributions (in the right). The greedy iteration number increases (159; 319; 478) from the top down. The focus intensities in these iteration numbers (159; 319; 478) are 1.216; 1.327; 1.434. Respective device domain's areas are $0.006 \lambda^2$; $0.011 \lambda^2$; $0.017 \lambda^2$ (0.015% ; 0.030% ; 0.046% of design domain's area). In addition, the focus intensity increased by $\approx 40\%$.

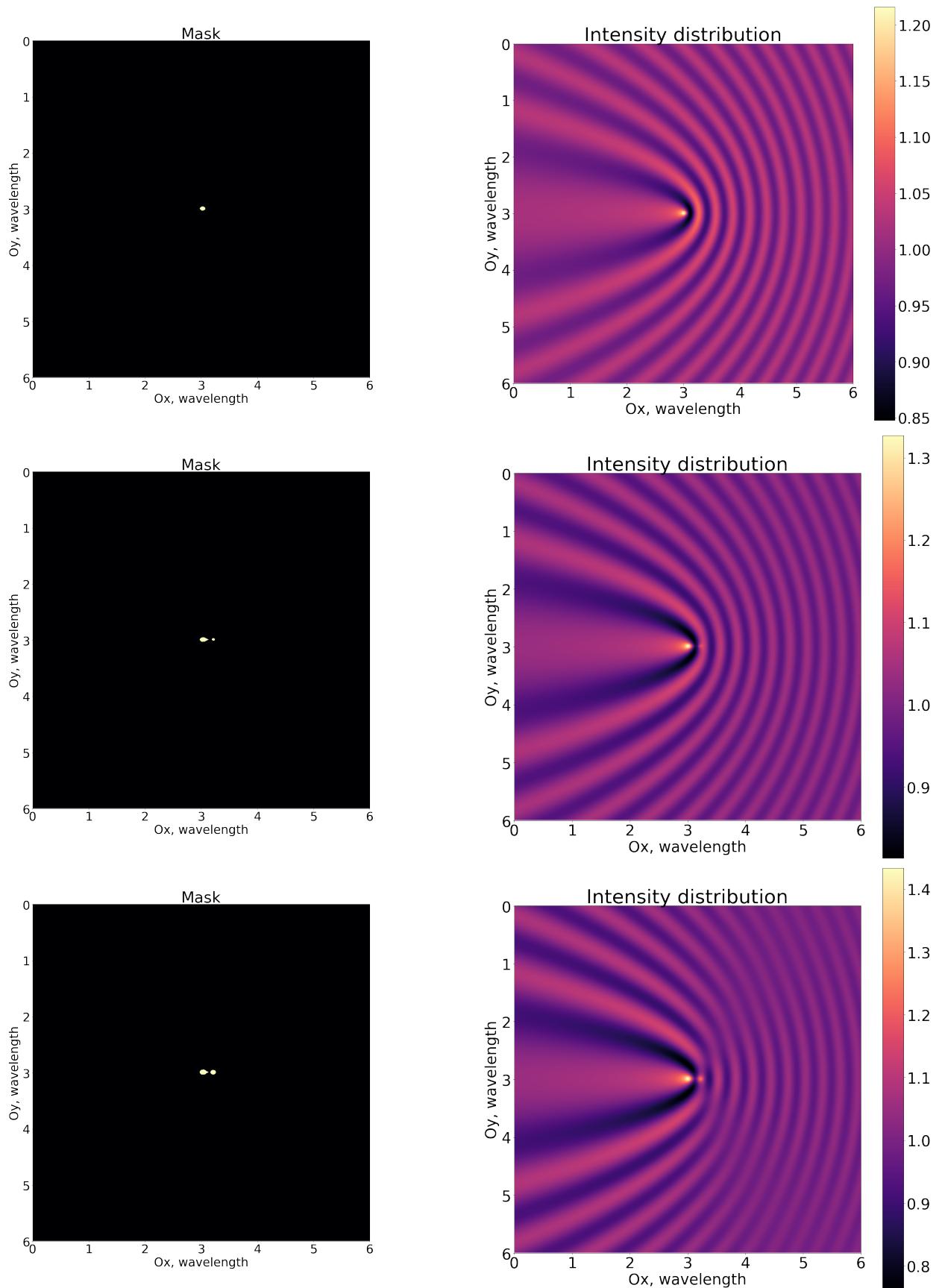


Figure 5.16: CUDA greedy ESO for lens (fixed focus)

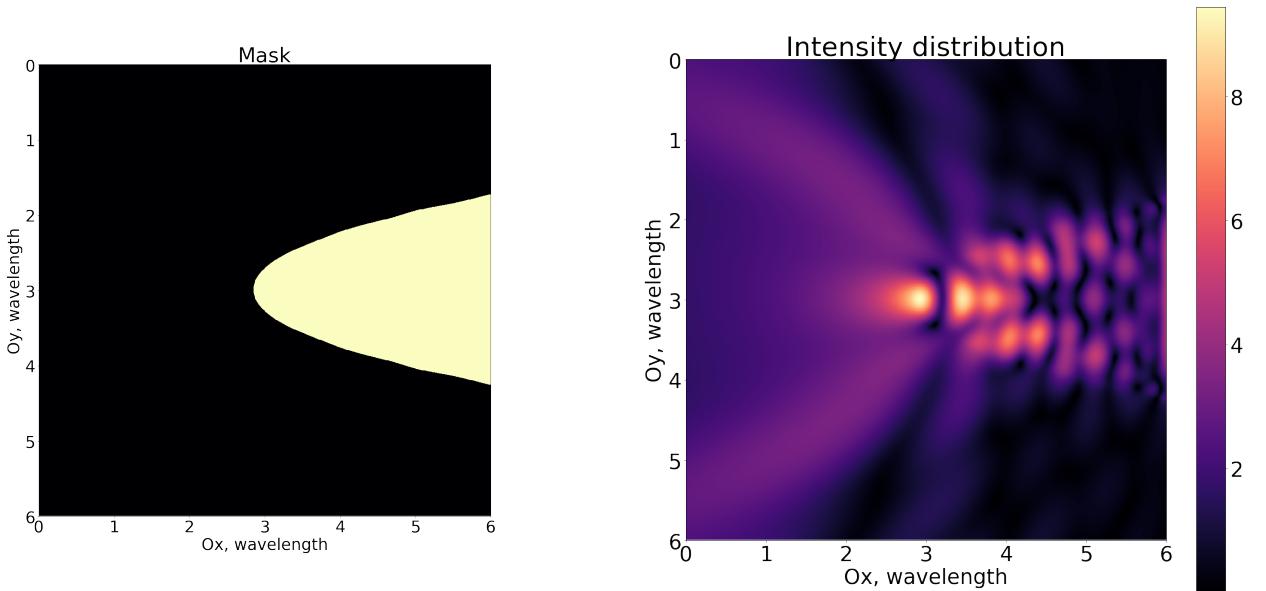


Figure 5.17: CUDA discrete gradient ESO for superlens of one not fixed focus

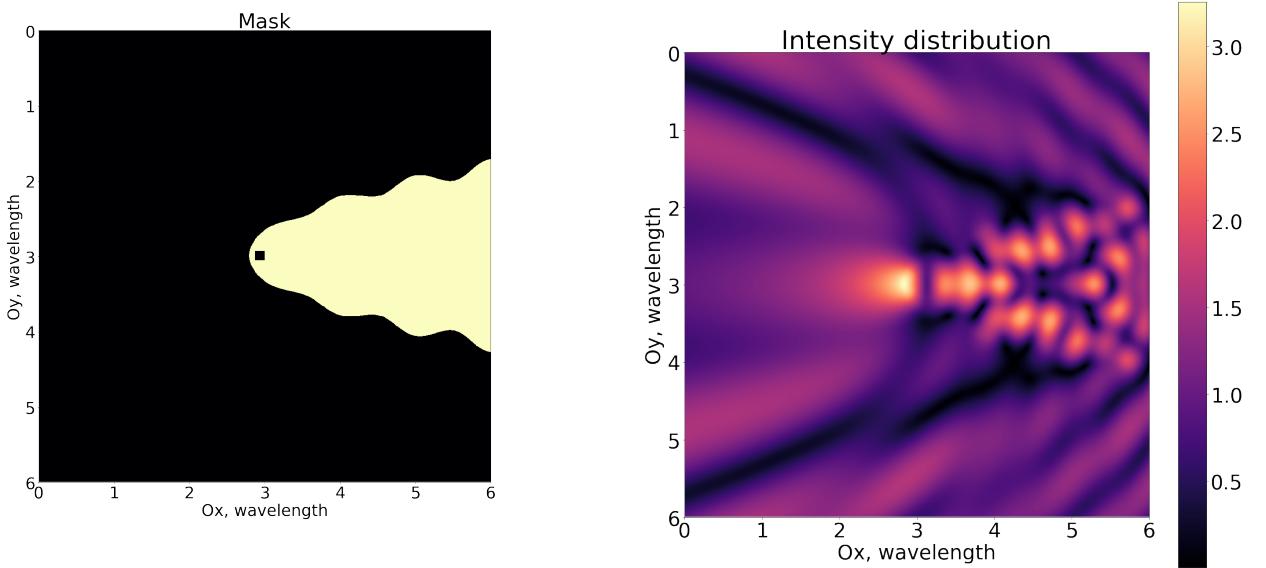


Figure 5.18: CUDA discrete gradient ESO for superlens of internal focus surrounded by cutting exclusion zone

5.4.2 Discrete gradient

Figure 5.17 represents result of gradient numerical method for ESO of glass lens shape. The resolution is 1024×1024 nodes². $N = 6$ wavelength per domain. ≈ 170.7 nodes per wavelength λ . The not fixed focus is located at node point [512, 512]. Target function is the intensity at the focus point. Initial mask is in the figure 5.15. Initial focus intensity equals to 1.00. The ESO progress is described by mask (in the left) and respective intensity distribution (in the right). We represented first discrete gradient iteration. The focus intensity equals to 8.663 after the first discrete gradient iteration. However, focus was ended up inside device domain. Due to that feature, the application of focus, for example, in welding is impossible. That is why next versions of suggested software were corrected so that the maximum is beyond device domain. However, the considered focus

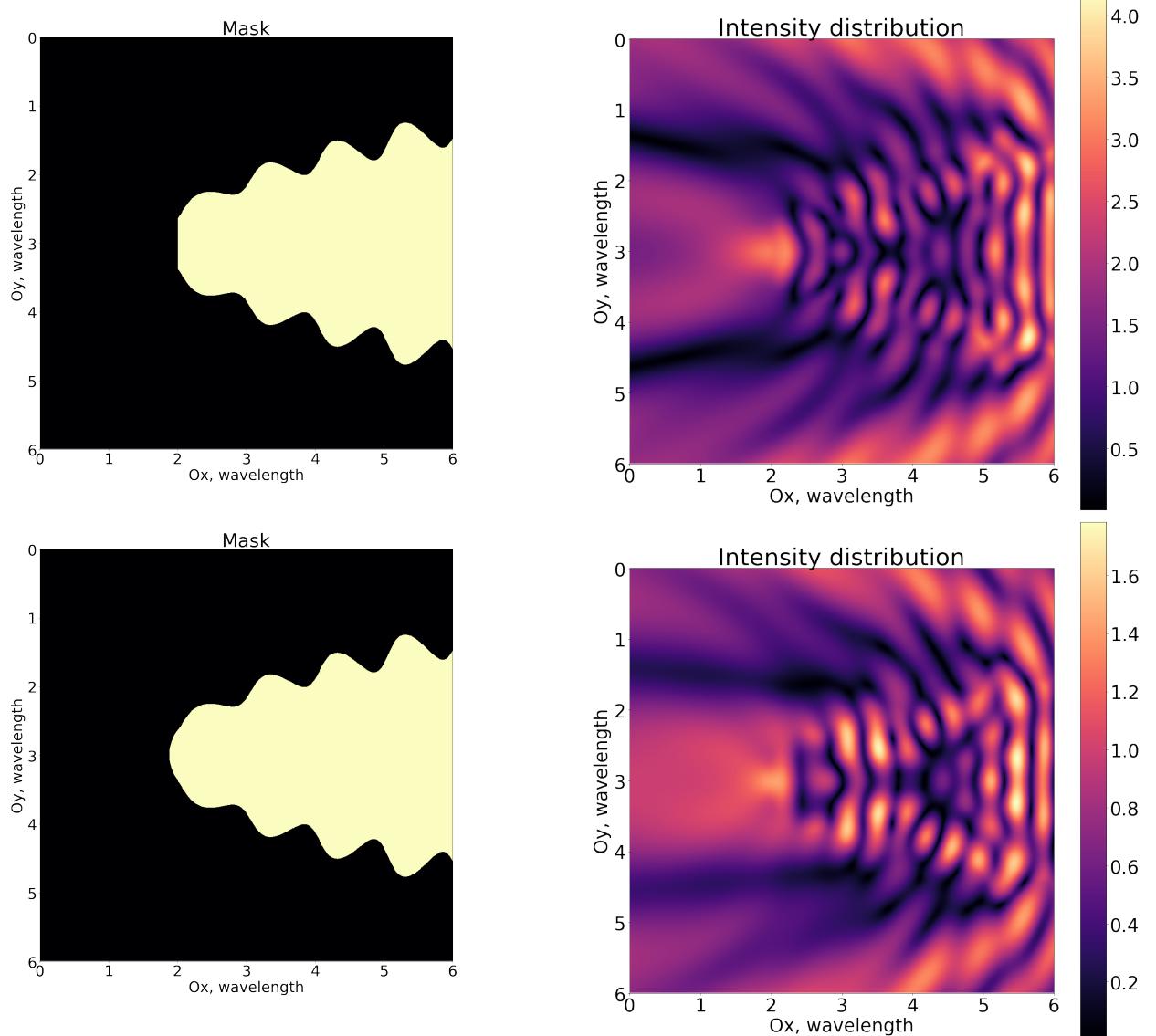


Figure 5.19: Attempt to post-process intuitively results of CUDA discrete gradient ESO for super-lens of one fixed focus surrounded by cutting exclusion zone expressed by half-plane

intensity increased by $\approx 740\%$. The device domain's area accounts for $5.3 \lambda^2$ (14.7% of design domain's area).

Figure 5.18 represents result of gradient numerical method for ESO of glass lens shape by using exclusion zone. This exclusion zone is the $10 \text{ nodes} \times 10 \text{ nodes}$ domain around focus where mask is defined to be empty. The resolution is $1024 \times 1024 \text{ nodes}^2$. $N = 6$ wavelength per domain. ≈ 170.7 nodes per wavelength λ . The fixed focus is located at node point [512, 512]. Target function is the intensity at the focus point. Initial mask is in the figure 5.15. Initial focus intensity equals to 1.029. The ESO progress is described by mask (in the left) and respective intensity distribution (in the right). We represented first discrete gradient iteration. The focus intensity equals to 2.020 after the first discrete gradient iteration. In addition, the focus intensity increased by $\approx 100\%$. The device domain's area accounts for $5.1 \lambda^2$ (14.1% of design domain's area).

Figure 5.19 represents result of gradient numerical method for ESO of glass lens shape by using exclusion zone. This exclusion zone is the Ox half-plane inside $[-\infty; 330]$ nodes where mask is defined to be empty. The resolution is $1024 \times 1024 \text{ nodes}^2$. $N = 6$ wavelength per domain. ≈ 170.7 nodes per wavelength λ . Initial mask is in the figure 5.15. The fixed focus is located at node point [300, 512]. Target function is the intensity at the focus point. Initial focus intensity equals to 1.01. The ESO progress is described by masks (in the left) and respective intensity distributions (in the right). The focus intensity equals to 1.85 after the first discrete gradient iteration. We tried to post-process lens shape by addition circular convex left shape to the left edge of the device (the result of attempt is mask represented in lower left figure 5.19). However, after such intuitive post-processing the intensity decreased to 1.00 that almost coincides with intensity of initial device. The device domain's areas (represented in the figure 5.19) account for $9.38 \lambda^2$; $9.44 \lambda^2$ (26.0% ; 26.2% ; of design domain's area). Figure 5.20 represents result of gradient numerical method for BESO of glass lens shape by using exclusion zone. This exclusion zone is the Ox half-plane inside $[-\infty; 330]$ nodes where mask is defined to be empty. The resolution is $1024 \times 1024 \text{ nodes}^2$. $N = 6$ wavelength per domain. ≈ 170.7 nodes per wavelength λ . The fixed focuses are located at node points [5, 128]; [5, 512]; [5, 896]. Target function is sum of intensities at the focus points. Initial mask is in the figure 5.15. Initial focus intensities equals to 1.00; 1.00; 1.00. The BESO progress is described by masks (in the left) and respective intensity distributions (in the right). We represented first two discrete gradient iterations. The focus intensities equal to 1.11; 1.18; 1.12 after the first discrete gradient iteration and equal to 1.30; 1.19; 1.29 after the second discrete gradient iteration. In addition, the sum of focus intensities increased by $\approx 14\%$ to 3.41, then by 10% to 3.78. Total intensity increase is equal to 26% . The device domain's areas account for $0.464 \lambda^2$; $0.582 \lambda^2$ (1.29% ; 1.62% of design domain's area) after the mentioned iterations respectively.

Figure 5.21 represents result of gradient numerical method for ESO of glass subwavelength

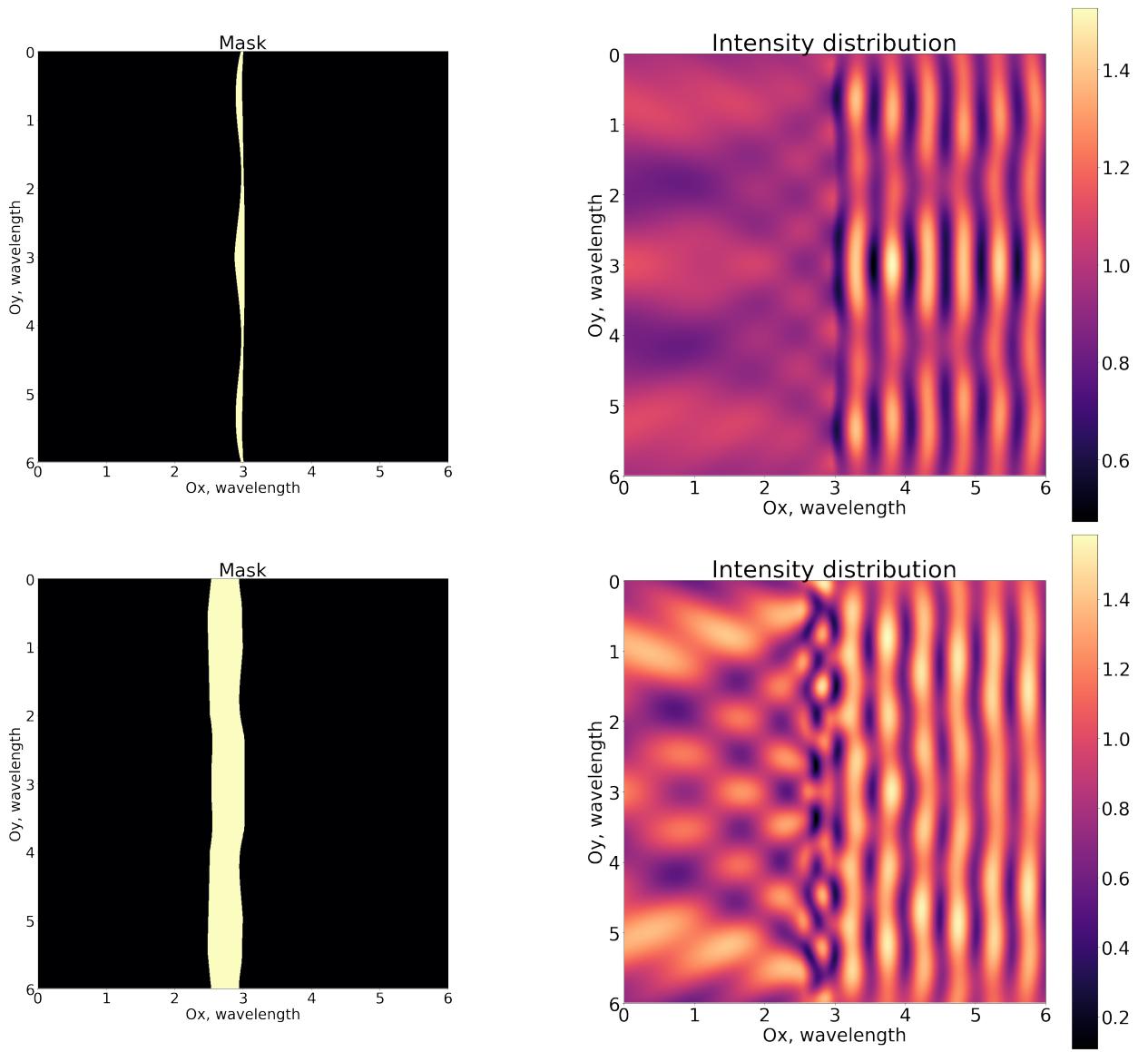


Figure 5.20: CUDA discrete gradient BESO for superlens of three fixed focuses surrounded by cutting exclusion zone expressed by half-plane

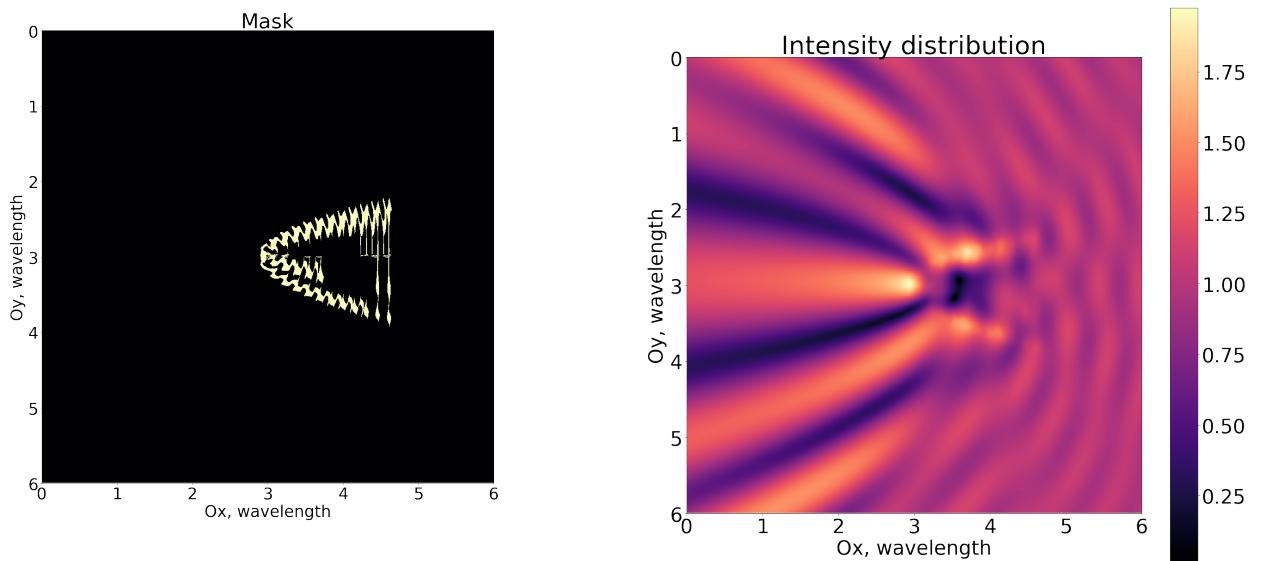


Figure 5.21: CUDA discrete gradient ESO for device of required normal intensity distribution

imaging device of required normal distribution of intensity in specific domain surrounded by exclusion zone. This exclusion zone is the Ox half-plane inside $[-\infty; 330]$ nodes where mask is defined to be empty. The resolution is 1024×1024 nodes². $N = 6$ wavelength per domain. ≈ 170.7 nodes per wavelength λ . The center of target domain is located at [5, 512]. The target is minimization of difference between required and computed intensities distributions in specific domain. Initial mask is in the figure 5.15. The ESO progress is described by mask (in the left) and respective intensity distribution (in the right). We represented first discrete gradient iteration. Total intensity increase is equal to 34.7%. The device domain's areas account for $0.582 \lambda^2$ (1.62% of design domain's area) after the mentioned iterations respectively.

5.4.3 Steepest descent

Figure 5.22 represents result of steepest descent numerical method for BESO of glass lens shape by using exclusion zone. This exclusion zone is the Ox half-plane inside $[-\infty; 10]$ nodes where mask is defined to be empty. The resolution is 1024×1024 nodes². $N = 6$ wavelength per domain. ≈ 170.7 nodes per wavelength λ . The fixed focus is located at node point [5, 512]. Target function is intensity at the focus point. Initial mask is in the figure 5.15. Initial focus intensity equal to 1.00. The BESO progress is described by masks (in the left) and respective intensity distributions (in the right). We represented 10884; 21768; 32653 steepest descent iterations. The focus intensities equal to 1.370; 2.295; 2.908 during these iterations respectively. In addition, the focus intensities increased by $\approx 37\%$; 67% ; 26% . Total intensity increase is equal to 191%. The device domain's areas account for $0.154 \lambda^2$; $0.185 \lambda^2$; $0.275 \lambda^2$ (0.43%; 0.51%; 0.77% of design domain's area) after the mentioned iterations respectively.

Figure 5.23 represents result of steepest descent numerical method for BESO of glass lens shape by using exclusion zone. This exclusion zone is the Ox half-plane inside $[-\infty; 100]$ nodes where mask is defined to be empty. The resolution is 1024×1024 nodes². $N = 6$ wavelength per domain. ≈ 170.7 nodes per wavelength λ . The fixed focus is located at node point [5, 512]. Target function is intensity at the focus point. The BESO progress is described by masks (in the left) and respective intensity distributions (in the right). We represented 0; 3176; 6353 steepest descent iterations. The focus intensities equal to 1.233; 1.241; 1.376 during these iterations respectively. In addition, the focus intensities increased by $\approx 0.6\%$; 10.9% . Total intensity increase is equal to 11.9%. The device domain's areas account for $6.086 \lambda^2$; $6.118 \lambda^2$; $6.127 \lambda^2$ (16.90%; 17.00%; 17.01% of design domain's area) after the mentioned iterations respectively.

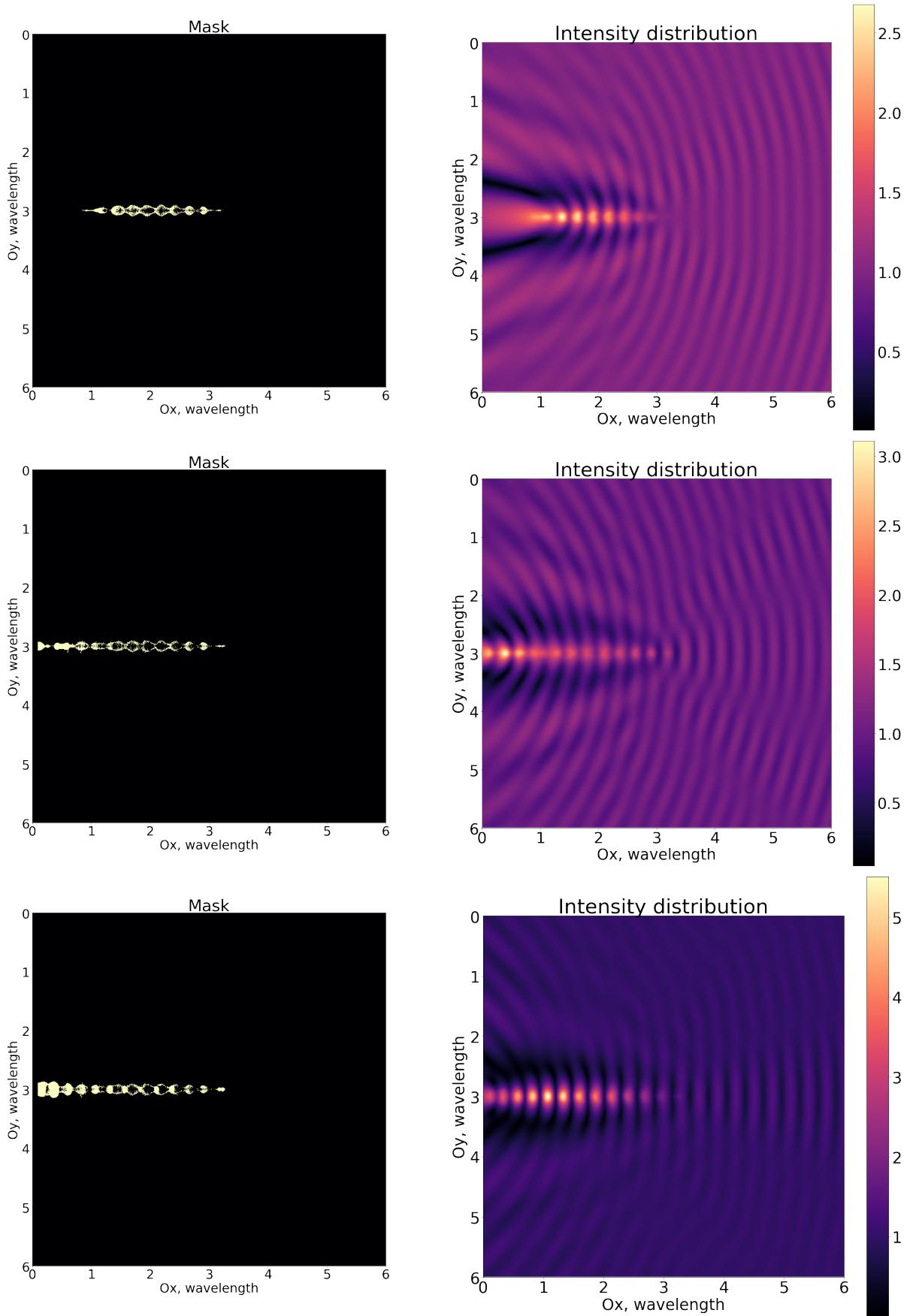


Figure 5.22: CUDA steepest descent BESO for superlens of one fixed point surrounded by exclusion zone expressed by half-plane

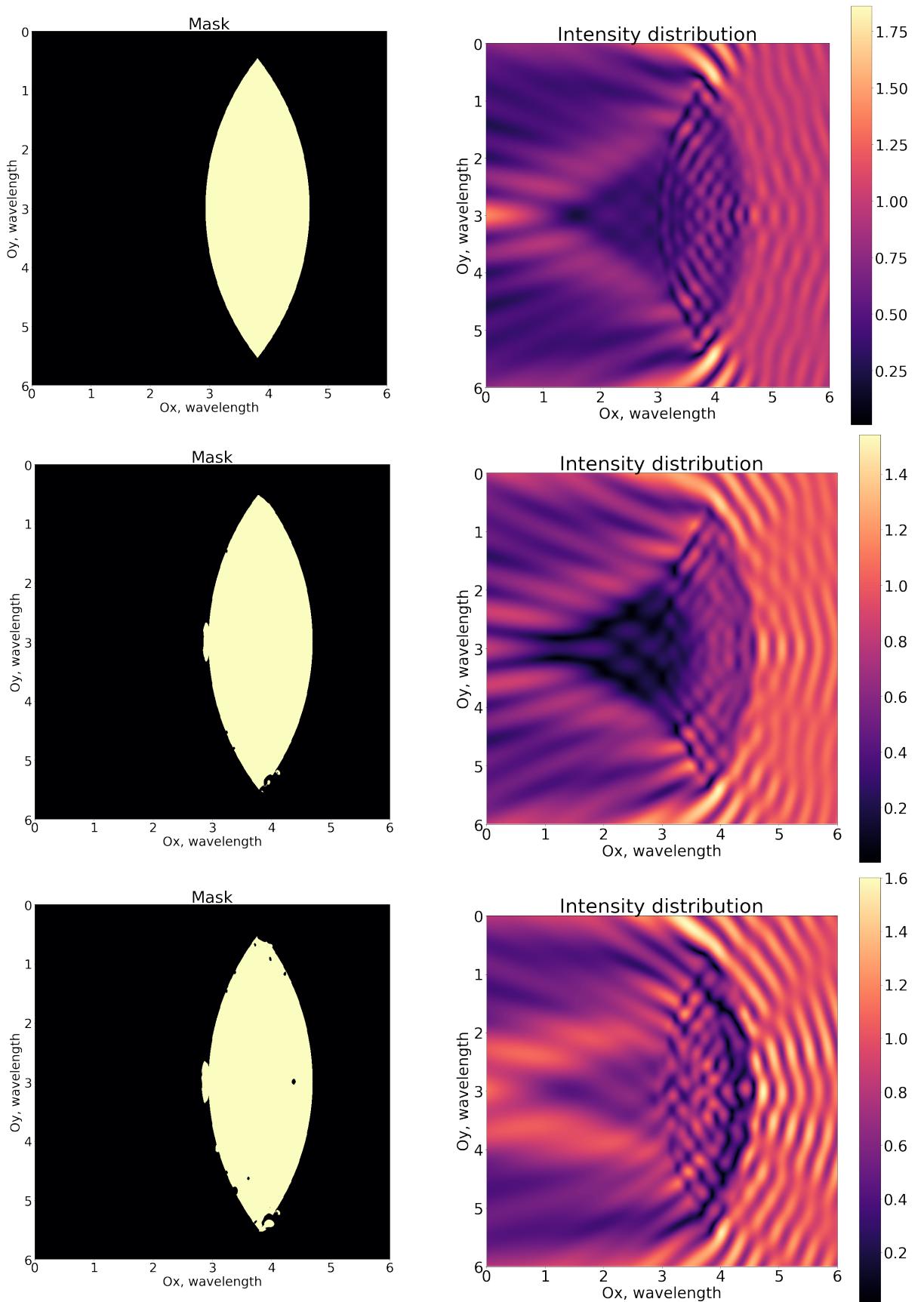


Figure 5.23: CUDA steepest descent BESO (initial mask is determined by biconvex lens) for superlens of one fixed point surrounded by exclusion zone expressed by half-plane

Chapter 6

Conclusion

We conclude that TO method used in the present work (based on FFT-GFIEM solver) has relatively low computational cost, can be applied to broad set of wave devices and allows to predict and analyze their shapes corresponding to given wave fields distributions.

It works as a black box, depends on minimal parametric set (defined by domain size and scattered field distribution), computationally stable and finds such shapes and geometries close to experimental samples.

It is practically useful because it gives efficient way of photonic devices design which is intensively carrying out in radio-element manufacturers.

The problem of erroneous surroundings of intensity maximum pixel by substance was resolved by using exclusion zone.

However, the BESO with greedy method achieves better optimization, but computing speed is decades times lower in comparison with the BESO with discrete gradient method. The developed CUDA (C++) steepest descent showed that the biggest increase in intensities in the focus surrounded by exclusion zone expressed by half-plane. In addition, steepest descent results illustrated that big increase in intensity may be achieved by materials of area equal less than $1 \lambda^2$.

In addition, CUDA (C++) paralleled GMRES may give speedup in comparison with Numpy bigger than 100. However, the very BESO has opportunities to be paralleled in more streams. For example, greedy BESO may have number of streams equal to the number of neighbours to device material in additive method and equal to the number of material bits in subtractive method. We approved that developed BESO numerical may be much more user-friendly in terms of computational time consumption and opportunities to control the design process via generalization of problem, for example, by using constraints and introduction of target function extension based on required intensity distribution.

The increasing of small inclusions of substance lack with number of wavelengths per design domain, material mask symmetry, program parallelization, device endurance test, practical checks of optical power and segmentation convergence remain unresolved problems.

Bibliography

- [1] G. Navon, Y.-Q. Song, T. Room, S. Appelt, R. Taylor, and A. Pines, “Enhancement of solution nmr and mri with laser-polarized xenon,” *Science*, vol. 271, no. 5257, pp. 1848–1851, 1996.
- [2] T. J. Vogl, M. G. Mack, R. Straub, A. Roggan, and R. Felix, “Percutaneous mri-guided laser-induced thermotherapy for hepatic metastases for colorectal cancer,” *The Lancet*, vol. 350, no. 9070, p. 29, 1997.
- [3] L. O. Severgina, N. Sorokin, A. Dymov, D. Tsarichenko, D. Enikeev, D. Kislyakov, L. Rapoport, I. Korovin, and D. Korolev, “Laser en-bloc resection of non-muscle-invasive bladder cancer: clinical and morphological specificities,” *Cancer Urology*, vol. 14, no. 3, pp. 78–84, 2018.
- [4] T. Staufer, S. Bohlen, K. Poder, T. Brümmer, F. Blumendorf, O. Schmutzler, M. Meisel, J. Osterhoff, and F. Grüner, “Development of a laser-wakefield thomson x-ray source for x-ray fluorescence imaging,” in *Laser Acceleration of Electrons, Protons, and Ions V*, vol. 11037, p. 110370M, International Society for Optics and Photonics, 2019.
- [5] P. A. Oganesyan and S. N. Shevcov, “Optimizing the design topology in the abaqus software,” *Izvestia of Samara Scientific Center of the Russian Academy of Sciences*, vol. 16, pp. 543–549, 2014.
- [6] Y. M. Temis and D. A. Yakushev, “Optimization of gte parts and components design,” *Vestnik of Samara University. Aerospace and Mechanical Engineering*, no. 3-1, pp. 183–188, 2011.
- [7] J. Liu and Y. Ma, “A survey of manufacturing oriented topology optimization methods,” *Adv. Eng. Softw.*, vol. 100, pp. 161–175, Oct. 2016.
- [8] B. E. Vasilev and L. A. Magerramova, “Analysis of the possibility of applying topological optimization in the design of uncooled turbine blades,” *Izvestia of Samara Scientific Center of the Russian Academy of Sciences*, no. 3-1, pp. 139–147, 2015.
- [9] C. Maxwell, “Scientific papers,” vol. II, p. 175, 1895.
- [10] A. G. M. Michell, “Lviii. the limits of economy of material in frame-structures,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 8, no. 47, pp. 589–597, 1904.

- [11] M. P. Bendsøe and N. Kikuchi, “Generating optimal topologies in structural design using a homogenization method,” *Computer methods in applied mechanics and engineering*, vol. 71, no. 2, pp. 197–224, 1988.
- [12] S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.
- [13] S. A. Burns, *Recent advances in optimal structural design*. ASCE Publications, 2002.
- [14] U. Schramm and M. Zhou, “Recent developments in the commercial implementation of topology optimization,” in *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials* (M. P. Bendsøe, N. Olhoff, and O. Sigmund, eds.), (Dordrecht), pp. 239–248, Springer Netherlands, 2006.
- [15] U. Schramm, H. Thomas, M. Zhou, and B. Voth, “Topology optimization with altair optistruct,” in *Proceedings of the Optimization in Industry II Conference, Banff, Canada*, 1999.
- [16] Altair OptiStruct, “Optistruct 11.0 user manual,” *Altair Engineering Inc, Troy, MI*, 2011.
- [17] Abaqus Scripting User’s Manual, “Abaqus 6.11,” <http://simulia.com>, vol. 89, no. 2080, p. v6.
- [18] ANSYS Academic Research Mechanical, release 18.1 ed.
- [19] W. P. Rodden and E. H. Johnson, *MSC/NASTRAN aeroelastic analysis: user’s guide; Version 68*. MacNeal-Schwendler Corporation, 1994.
- [20] R. Lahey *et al.*, “MSC/NASTRAN Reference Manual: Volume I, II, III,” *The MacNeal-Schwendler Corporation*, 1995.
- [21] J. D. Deaton and R. V. Grandhi, “A survey of structural and multidisciplinary continuum topology optimization: post 2000,” *Structural and Multidisciplinary Optimization*, vol. 49, pp. 1–38, Jan 2014.
- [22] G. Roelkens, D. Van Thourhout, R. Baets, R. Nötzel, and M. Smit, “Laser emission and photodetection in an inp/ingaasp layer integrated on and coupled to a silicon-on-insulator waveguide circuit,” *Optics express*, vol. 14, no. 18, pp. 8154–8159, 2006.
- [23] A. W. Fang, H. Park, O. Cohen, R. Jones, M. J. Paniccia, and J. E. Bowers, “Electrically pumped hybrid algainas-silicon evanescent laser,” *Optics express*, vol. 14, no. 20, pp. 9203–9210, 2006.
- [24] E. Yablonovitch, “Inhibited spontaneous emission in solid-state physics and electronics,” *Physical review letters*, vol. 58, no. 20, p. 2059, 1987.

- [25] S. John, “Strong localization of photons in certain disordered dielectric superlattices,” *Physical review letters*, vol. 58, no. 23, p. 2486, 1987.
- [26] C. Hafner, C. Xudong, J. Smajic, and R. Vahldieck, “Efficient procedures for the optimization of defects in photonic crystal structures,” *JOSA A*, vol. 24, no. 4, pp. 1177–1188, 2007.
- [27] A. Chutinan and S. Noda, “Waveguides and waveguide bends in two-dimensional photonic crystal slabs,” *Physical review B*, vol. 62, no. 7, p. 4488, 2000.
- [28] M. Palamaru and P. Lalanne, “Photonic crystal waveguides: Out-of-plane losses and adiabatic modal conversion,” *Applied Physics Letters*, vol. 78, no. 11, pp. 1466–1468, 2001.
- [29] A. Talneau, P. Lalanne, M. Agio, and C. M. Soukoulis, “Low-reflection photonic-crystal taper for efficient coupling between guide sections of arbitrary widths,” *Optics letters*, vol. 27, no. 17, pp. 1522–1524, 2002.
- [30] I. Ntakis, P. Pottier, and R. M. De La Rue, “Optimization of transmission properties of two-dimensional photonic crystal channel waveguide bends through local lattice deformation,” *Journal of applied physics*, vol. 96, no. 1, pp. 12–18, 2004.
- [31] A. Chutinan, M. Okano, and S. Noda, “Wider bandwidth with high transmission through waveguide bends in two-dimensional photonic crystal slabs,” *Applied physics letters*, vol. 80, no. 10, pp. 1698–1700, 2002.
- [32] S. Olivier, H. Benisty, M. Rattier, C. Weisbuch, M. Qiu, A. Karlsson, C. Smith, R. Houdré, and U. Oesterle, “Resonant and nonresonant transmission through waveguide bends in a planar photonic crystal,” *Applied physics letters*, vol. 79, no. 16, pp. 2514–2516, 2001.
- [33] S. Fan, S. G. Johnson, J. Joannopoulos, C. Manolatou, and H. Haus, “Waveguide branches in photonic crystals,” *JOSA B*, vol. 18, no. 2, pp. 162–165, 2001.
- [34] J. Moosburger, M. Kamp, A. Forchel, S. Olivier, H. Benisty, C. Weisbuch, and U. Oesterle, “Enhanced transmission through photonic-crystal-based bent waveguides by bend engineering,” *Applied Physics Letters*, vol. 79, no. 22, pp. 3579–3581, 2001.
- [35] Y. Zhang and B. Li, “Ultracompact waveguide bends with simple topology in two-dimensional photonic crystal slabs for optical communication wavelengths,” *Optics letters*, vol. 32, no. 7, pp. 787–789, 2007.
- [36] J. Smajic, C. Hafner, and D. Erni, “Design and optimization of an achromatic photonic crystal bend,” *Optics Express*, vol. 11, no. 12, pp. 1378–1384, 2003.

- [37] W. J. Kim and J. D. O'Brien, "Optimization of a two-dimensional photonic-crystal waveguide branch by simulated annealing and the finite-element method," *JOSA B*, vol. 21, no. 2, pp. 289–295, 2004.
- [38] J. Smajic, C. Hafner, and D. Erni, "Optimization of photonic crystal structures," *JOSA A*, vol. 21, no. 11, pp. 2223–2232, 2004.
- [39] D. Erni, D. Wiesmann, M. Spühler, S. Hunziker, E. Moreno, B. Oswald, J. Fröhlich, and C. Hafner, "Application of evolutionary optimization algorithms in computational optics," in *ACES*, vol. 15, Citeseer, 2000.
- [40] J. S. Jensen and O. Sigmund, "Systematic design of photonic crystal structures using topology optimization: Low-loss waveguide bends," *Applied physics letters*, vol. 84, no. 12, pp. 2022–2024, 2004.
- [41] P. I. Borel, A. Harpøth, L. H. Frandsen, M. Kristensen, P. Shi, J. S. Jensen, and O. Sigmund, "Topology optimization and fabrication of photonic crystal structures," *Optics Express*, vol. 12, no. 9, pp. 1996–2001, 2004.
- [42] L. H. Frandsen, A. Harpøth, P. I. Borel, M. Kristensen, J. S. Jensen, and O. Sigmund, "Broadband photonic crystal waveguide 60 bend obtained utilizing topology optimization," *Optics Express*, vol. 12, no. 24, pp. 5916–5921, 2004.
- [43] A. Lavrinenko, A. Têtû, L. H. Frandsen, and P. I. Borel, "Optimization of photonic crystal 60 degree waveguide bends in the slow light regime for broadband transmission," in *Quantum Electronics and Laser Science Conference*, p. QFC6, Optical Society of America, 2006.
- [44] J. S. Jensen, O. Sigmund, L. H. Frandsen, P. I. Borel, A. Harpøth, and M. Kristensen, "Topology design and fabrication of an efficient double 90/spl deg/photonic crystal waveguide bend," *IEEE Photonics Technology Letters*, vol. 17, no. 6, pp. 1202–1204, 2005.
- [45] P. I. Borel, L. H. Frandsen, A. Harpøth, M. Kristensen, J. S. Jensen, and O. Sigmund, "Topology optimised broadband photonic crystal y-splitter," *Electronics Letters*, vol. 41, no. 2, pp. 69–71, 2005.
- [46] O. Sigmund and K. Hougaard, "Geometric properties of optimal photonic crystals," *Physical review letters*, vol. 100, no. 15, p. 153904, 2008.
- [47] R. Matzen, J. S. Jensen, and O. Sigmund, "Topology optimization for transient response of photonic crystal structures," *JOSA B*, vol. 27, no. 10, pp. 2040–2050, 2010.

- [48] W. Xie, P. Niay, P. Bernage, M. Douay, J. Bayon, T. Georges, M. Monerie, and B. Poumellec, “Experimental evidence of two types of photorefractive effects occurring during photoinscriptions of bragg gratings within germanosilicate fibres,” *Optics Communications*, vol. 104, no. 1-3, pp. 185–195, 1993.
- [49] “b,”
- [50] A. Cherkaev, *Variational methods for structural optimization*, vol. 140. Springer Science & Business Media, 2012.
- [51] J. E. Cadman, S. Zhou, Y. Chen, and Q. Li, “On design of multi-functional microstructural materials,” *Journal of Materials Science*, vol. 48, no. 1, pp. 51–66, 2013.
- [52] M. P. Bendsøe, “Optimal shape design as a material distribution problem,” *Structural optimization*, vol. 1, no. 4, pp. 193–202, 1989.
- [53] M. Zhou and G. Rozvany, “The coc algorithm, part ii: Topological, geometrical and generalized shape optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 89, no. 1-3, pp. 309–336, 1991.
- [54] G. I. Rozvany, M. Zhou, and T. Birker, “Generalized shape optimization without homogenization,” *Structural optimization*, vol. 4, no. 3-4, pp. 250–252, 1992.
- [55] O. Sigmund, “Materials with prescribed constitutive parameters: an inverse homogenization problem,” *International Journal of Solids and Structures*, vol. 31, no. 17, pp. 2313–2329, 1994.
- [56] O. Sigmund, “Tailoring materials with prescribed elastic properties,” *Mechanics of Materials*, vol. 20, no. 4, pp. 351–368, 1995.
- [57] L. S. R. Krishna, N. Mahesh, and N. Sateesh, “Topology optimization using solid isotropic material with penalization technique for additive manufacturing,” *Materials Today: Proceedings*, vol. 4, no. 2, Part A, pp. 1414 – 1422, 2017. 5th International Conference of Materials Processing and Characterization (ICMPC 2016).
- [58] M. Y. Wang, X. Wang, and D. Guo, “A level set method for structural topology optimization,” *Computer methods in applied mechanics and engineering*, vol. 192, no. 1-2, pp. 227–246, 2003.
- [59] X. Wang, Y. Mei, and M. Y. Wang, “Level-set method for design of multi-phase elastic and thermoelastic materials,” *International Journal of Mechanics and Materials in Design*, vol. 1, no. 3, pp. 213–239, 2004.

- [60] A. Wilkins, V. Challis, and A. Roberts, “Isotropic, stiff, conducting structures via topology optimisation,” 2007.
- [61] V. Challis, A. Roberts, and A. Wilkins, “Design of three dimensional isotropic microstructures for maximized stiffness and conductivity,” *International Journal of Solids and Structures*, vol. 45, no. 14-15, pp. 4130–4146, 2008.
- [62] S. Zhou, W. Li, G. Sun, and Q. Li, “A level-set procedure for the design of electromagnetic metamaterials,” *Optics express*, vol. 18, no. 7, pp. 6693–6702, 2010.
- [63] Y. M. Xie and G. P. Steven, “A simple evolutionary procedure for structural optimization,” *Computers & structures*, vol. 49, no. 5, pp. 885–896, 1993.
- [64] Y. M. Xie and G. P. Steven, “Basic evolutionary structural optimization,” in *Evolutionary structural optimization*, pp. 12–29, Springer, 1997.
- [65] S. Patil, S. W. Zhou, and Q. Li, “Design of periodic microstructural materials by using evolutionary structural optimization method,” in *Advanced Materials Research*, vol. 32, pp. 279–283, Trans Tech Publ, 2008.
- [66] O. Querin, G. Steven, and Y. Xie, “Evolutionary structural optimisation (eso) using a bidirectional algorithm,” *Engineering computations*, vol. 15, no. 8, pp. 1031–1048, 1998.
- [67] X. Yang, Y. Xei, G. Steven, and O. Querin, “Bidirectional evolutionary method for stiffness optimization,” *AIAA journal*, vol. 37, no. 11, pp. 1483–1488, 1999.
- [68] X. Huang and Y. Xie, “Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method,” *Finite Elements in Analysis and Design*, vol. 43, no. 14, pp. 1039–1049, 2007.
- [69] X. Huang and M. Xie, *Evolutionary topology optimization of continuum structures: methods and applications*. John Wiley & Sons, 2010.
- [70] N. F. Wang and X. M. Zhang, “A solid isotropic material with parallel penalization method for structural topology optimization with multiple materials,” in *2016 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, pp. 1–6, July 2016.
- [71] X. Huang and Y. Xie, “Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials,” *Computational Mechanics*, vol. 43, no. 3, p. 393, 2009.
- [72] X. Huang and Y. Xie, “Evolutionary topology optimization of continuum structures with an additional displacement constraint,” *Structural and Multidisciplinary Optimization*, vol. 40, no. 1-6, p. 409, 2010.

- [73] G. I. Rozvany, “Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics,” *Structural and Multidisciplinary optimization*, vol. 21, no. 2, pp. 90–108, 2001.
- [74] G. I. Rozvany, “A critical review of established methods of structural topology optimization,” *Structural and multidisciplinary optimization*, vol. 37, no. 3, pp. 217–237, 2009.
- [75] X. Huang and Y.-M. Xie, “A further review of eso type methods for topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 41, no. 5, pp. 671–683, 2010.
- [76] X. Huang and Y. Xie, “Optimal design of periodic structures using evolutionary topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 36, no. 6, pp. 597–606, 2008.
- [77] X. Huang, Z. Zuo, and Y. Xie, “Evolutionary topological optimization of vibrating continuum structures for natural frequencies,” *Computers & structures*, vol. 88, no. 5-6, pp. 357–364, 2010.
- [78] J. P. Mercier, G. Zambelli, and W. Kurz, *Introduction to materials science*. Elsevier, 2012.
- [79] X. Huang and Y. Xie, “Evolutionary topology optimization of continuum structures with an additional displacement constraint,” 2009.
- [80] X. Huang, Y. Xie, and G. Lu, “Topology optimization of energy-absorbing structures,” *International Journal of crashworthiness*, vol. 12, no. 6, pp. 663–675, 2007.
- [81] X. H. Huang and Y. Xie, “Bidirectional evolutionary topology optimization for structures with geometrical and material nonlinearities,” *AIAA journal*, vol. 45, no. 1, pp. 308–313, 2007.
- [82] X. Huang and Y. Xie, “Topology optimization of nonlinear structures under displacement loading,” *Engineering structures*, vol. 30, no. 7, pp. 2057–2068, 2008.
- [83] T. Søndergaard, *Green's Function Integral Equation Methods in Nano-Optics*. CRC Press, 2019.
- [84] M. Abramowitz, *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*,. New York, NY, USA: Dover Publications, Inc., 1974.
- [85] F. Murat and L. Tartar, “Optimality conditions and homogenization. nonlinear variational problems (isola d’elba, 1983),” *Research Notes in Mathematics*, vol. 127, 1985.
- [86] T. Belytschko, S. Xiao, and C. Parimi, “Topology optimization with implicit functions and regularization,” *International Journal for Numerical Methods in Engineering*, vol. 57, no. 8, pp. 1177–1196, 2003.

- [87] J. Norato, R. Haber, D. Tortorelli, and M. P. Bendsøe, “A geometry projection method for shape optimization,” *International Journal for Numerical Methods in Engineering*, vol. 60, no. 14, pp. 2289–2312, 2004.
- [88] A. Y. Piggott, J. Lu, T. M. Babinec, K. G. Lagoudakis, J. Petykiewicz, and J. Vučković, “Inverse design and implementation of a wavelength demultiplexing grating coupler,” *Scientific reports*, vol. 4, p. 7210, 2014.
- [89] I. B. Minin, E. E. Nuzhin, A. I. Boyko, M. S. Litsarev, and I. V. Oseledets, “Evolutionary Structural Optimization Algorithm based on FFT-GFIEM Solver for Inverse Design of Wave Devices,” in *2018 Vth International Conference on Engineering and Telecommunication (EnT)*, IEEE. (accepted).

Appendix A

Formulas

A.1 FFT-GFIEM discretization

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}_0(\mathbf{r}) + \int_{\Omega} \mathbf{G}(\mathbf{r}, \mathbf{r}') \cdot (\varepsilon(\mathbf{r}') - \varepsilon_{ref}) \cdot \mathbf{E}(\mathbf{r}') d^3 r' \quad (\text{A.1})$$

$$\mathbf{G}(\mathbf{r}, \mathbf{r}') = k_0^2 \cdot \frac{i}{4} \mathbf{H}_0^{(1)}(|k| |\mathbf{r} - \mathbf{r}'|) \quad (\text{A.2})$$

$$\forall i \in \overline{1, N^2} \quad \mathbf{E}_i = \mathbf{E}_{0,i} + \sum_{j=1}^{N^2} \mathbf{G}_{ij} (\varepsilon_j - \varepsilon_{ref}) \cdot \mathbf{E}_j \longrightarrow \quad (\text{A.3})$$

Write vector by vector in one big vector

$$\begin{aligned} & \longrightarrow \mathbf{E} = \mathbf{E}_0 + \mathbf{G}(\varepsilon - \varepsilon_{ref}) \circ \mathbf{E} \iff \\ & \iff \mathbf{E} = \mathbf{E}_0 + \mathbf{G} \cdot \left(\varepsilon - \underbrace{\varepsilon_{ref}}_{=1} \right) \cdot \text{diag}(\mathbf{m}) \cdot \mathbf{E} \end{aligned} \quad (\text{A.4})$$

$$\underbrace{(\mathbf{I} - \mathbf{G} \cdot (\varepsilon - 1) \cdot \text{diag}(\mathbf{m}))}_{\equiv \mathbf{A}(\mathbf{m})} \underbrace{\mathbf{E}}_{\mathbf{E}(\mathbf{m})} = \underbrace{\mathbf{E}_0}_{= \text{const}} \quad (\text{A.5})$$

$$\mathbf{A}(\mathbf{m}) \mathbf{E}(\mathbf{m}) = \mathbf{E}_0 \quad (\text{A.6})$$

$$\mathbf{E}(\mathbf{m}) - (\varepsilon - 1) \cdot \underbrace{\mathbf{G} \cdot \text{diag}(\mathbf{m}) \cdot \mathbf{E}(\mathbf{m})}_{=\text{FFT_matvec(BTTB}(\mathbf{G}), \text{diag}(\mathbf{m}) \cdot \mathbf{E}(\mathbf{m}))} = \mathbf{E}_0 \quad (\text{A.7})$$

A.2 Fast (FFT-based) GMRES method to solve linear equation systems based on Toeplitz matrix

A.2.1 4-dimensional cubic Green's matrix tensor G^{4D}

Green's matrix tensor G^{4D} is determined by the following formula:

$$\begin{aligned} G_{i_1 i_2 j_1 j_2}^{4D} &= \gamma(r_{i_1 i_2 j_1 j_2}), \\ r_{i_1 i_2 j_1 j_2} &= ((\zeta + i_1 - i_2)^2 + (j_1 - j_2)^2)^2 = r_{i_1 - i_2, j_1 - j_2} \\ \Rightarrow G_{i_1 i_2 j_1 j_2}^{4D} &= G_{i_1 - i_2, j_1 - j_2}^{4D} \quad \forall i_1, i_2, j_1, j_2 = \overline{1, n} \end{aligned} \quad (\text{A.8})$$

where correction ζ in this research paper was equal to $-\frac{1}{2}$. Thus, G^{4D} is the multilevel Toeplitz matrix.

A.2.2 Fast (FFT) matvec linear operator for G matrix by vector multiplication

The G -matrix by arbitrary p -dimensional vector x_a multiplication problem is the following: $G_m = G \cdot x_a$, where G_m is the result of this multiplication to be $p \times p$ matrix.

Usually matrix-by-vector multiplication requires $\mathcal{O}(p^2)$, but the same results may be obtained using 2D $n \times n$ fast (FFT) matvec for multilevel Toeplitz matrix G^{4D} . For this 2D FFT matvec one have to do $n \times 1D$ FFTs in each axis so that's $\mathcal{O}(p \cdot \log n)$. In addition, in the case of usual G -matrix by vector multiplication there is a storage problem with matrix G . For example, for $n = 64$: G - matrix storage requires p^2 float elements storage that corresponds to 2 gigabytes of Random-Access Memory while mentioned FFT matvec requires to storage $(2 \cdot n - 1) \times (2 \cdot n - 1)$ G_e^{2D} instead of G and $(2 \cdot n - 1) \times (2 \cdot n - 1)$ X_e instead of x_a . Thus, both these problems were resolved using 2D FFT matvec linear operator $\mathbf{G}*$ for matrix G by vector x_a multiplication in this research.

Firstly, this linear operator creates $n \times n$ G^{2D} -matrix, such that $G_{i,j}^{2D} \equiv G_{\Delta i, \Delta j}$, $\forall i = \overline{1, n}, \forall j = \overline{1, n}$, where $G_{\Delta i, \Delta j}$ was represented in (A.8). Secondly, the linear operator extend G^{2D} -matrix (Figure 2.3) to $G_e^{2D} = \begin{bmatrix} G^{2D} & G_{-2}^{2D} \\ G_{-1}^{2D} & G_{-1,-2}^{2D} \end{bmatrix}$, where -1 means to eliminate first row and to inverse row order while -2 means to eliminate first column and to inverse the column order. Thirdly, the linear operator converts p -element x_a to $n \times n$ X_a that, next, extends to $X_e = \begin{bmatrix} X_a & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & 0_{(n-1) \times (n-1)} \end{bmatrix}$, where $0_{n_1 \times n_2}$ is the zero $n_1 \times n_2$ matrix. Finally, the result of this linear

operator is computed by using the following formula:

$$G^{4D} \cdot X_a = \left[\left\{ G_e^{2D+} \cdot X_e^+ \right\}^- \right]_{\overline{1,n}, \overline{1,n}} \equiv G_m^{2D},$$

$$\text{vec}(G_m^{2D}) = G_m,$$

Thus, $\mathbf{G} * x_a \equiv \text{vec} \left(\left[\left\{ G_e^{2D+} \cdot X_e^+ \right\}^- \right]_{\overline{1,n}, \overline{1,n}} \right)$

(A.9)

where $[(\cdot)]_{\overline{1,n}, \overline{1,n}}$ means to take only first i, j -th elements of (\cdot) -matrix, where $i = \overline{1, n}$, $j = \overline{1, n}$; G_m^{2D} is $n \times n$ matrix.

A.2.3 Linear operator $\mathbf{A}(\mathbf{m})$ for $A(m)$ -matrix

\mathbf{A} is based on $\mathbf{G} * (\cdot)$, according to (3.5) and (A.9):

$$\begin{aligned} \mathbf{A}(\mathbf{m}) \cdot x_a &= \mathbf{I} - k^2 \cdot (\varepsilon - 1) \cdot \mathbf{G} * \mathbf{diag}(\mathbf{m}) \cdot x_a \equiv \\ &= \mathbf{I} - \chi \cdot \mathbf{G} * \mathbf{m} \circ x_a, \\ \chi &= k^2 \cdot (\varepsilon - 1) \end{aligned} \quad (A.10)$$

where \circ —is the Hadamard product, \mathbf{I} — is the linear operator that corresponds to the identity matrix I ; $\mathbf{diag}(\mathbf{m})$ —is the linear operator that corresponds to the diagonal matrix $\text{diag}(m)$.

In this research built-in GMRES algorithm of *scipy.sparse.linalg*-library was used to solve linear equation systems. The required absolute tolerance of this algorithm, $\kappa = 10^{-5}$ was achieved for 31 iterations. In addition, relative residual norms decreased monotonously exponentially with time and monotonously linearly with iteration number.

A.3 BESO generalization development

$$I_{max_of_max} = \max_{\mathbf{m} \in \{0,1\}^{N^2}} \left| i^* \equiv \arg \max_{i \in \overline{1, N^2}} |\mathbf{E}_i(\mathbf{m})| \right|^2$$

subject to

$$\left\{ \begin{array}{l} \underbrace{\mathcal{C}_1(\text{vec}(\{\mathbf{i}/\mathbf{m}_i = 1\}))}_{\text{Constraints for position of device development}} \\ \underbrace{\mathcal{C}_2(\mathbf{i}_{\text{set}}^*)}_{\text{Constraints for position of required electric field distribution}} \end{array} \right.$$
(A.11)

$$\frac{\partial \overbrace{|\mathbf{E}^*|^2}^{I_{max}}}{\partial \mathbf{m}} = 2 \operatorname{Re} \left\{ \mathbf{J}(\mathbf{E}^*(\mathbf{m})) \overline{\mathbf{E}^*(\mathbf{m})} \right\}. \quad (\text{A.12})$$

$\mathbf{A}(\mathbf{m})\mathbf{J}(\mathbf{E}(\mathbf{m})) = \mathbf{B}(\mathbf{m})$, where

$$\mathbf{J}(\mathbf{E}(\mathbf{m})) \equiv \frac{\partial \mathbf{E}(\mathbf{m})}{\partial \mathbf{m}} \quad (\text{A.13})$$

$$\mathbf{B}(\mathbf{m}) \equiv (\varepsilon - 1) \cdot \mathbf{G} \cdot \operatorname{diag}(\mathbf{E}(\mathbf{m}))$$

$$\begin{cases} i^* = \arg \max_{i \in \overline{1, N^2}} |\mathbf{E}_i(\mathbf{m})| \\ \mathbf{y}_i = \begin{cases} 1, & \text{if } i = i^* \\ 0, & \text{otherwise} \end{cases} \\ \mathbf{A}^T(\mathbf{m}) \cdot \mathbf{z}(\mathbf{m}) = \mathbf{y} \\ \mathbf{J}^T(\mathbf{E}^*(\mathbf{m})) = (\varepsilon - 1) \cdot \operatorname{diag}(\mathbf{E}(\mathbf{m})) \cdot \mathbf{G} * \mathbf{z}(\mathbf{m}) \end{cases} \quad (\text{A.14})$$

$$\begin{aligned} \|\Delta \mathbf{I}_i(\mathbf{m})\| &= \|w \cdot \mathbf{I}_{i \in \mathbf{i}_{set}^*}^{\text{required}}(\mathbf{m}) + b - \mathbf{I}_{i \in \mathbf{i}_{set}^*}^{\text{numerical}}(\mathbf{m})\| \rightarrow \min_{w \in \mathbb{C}, b \in \mathbb{C}} \\ \forall i \in \mathbf{i}_{set}^* \quad \|\Delta \mathbf{I}_i(\mathbf{m})\| &\rightarrow \min_{\mathbf{m} \in \{0,1\}^{N^2}} \end{aligned} \quad (\text{A.15})$$

A.3.1 Suggested generalized algorithm for BESO

First, start with true allowed addings and subtractings

$$\text{Allow_add}_i = \text{True} \quad \forall i \in \overline{1, n^2}$$

$$\text{Allow_sub}_i = \text{True} \quad \forall i \in \overline{1, n^2}$$

Then, continue with falsifications of them iteratively on $i^* \in \mathbf{i}_{\text{set}}^*$

$$\left\{
 \begin{array}{l}
 \text{if } \Delta \mathbf{I}_i(\mathbf{m}) > 0 \\
 \hline
 \text{false subtract substance from existing substance: } \text{Allow_sub}_i = \text{False}, \text{ where } \mathbf{J}_i(\mathbf{E}_{i^*}(\mathbf{m})) > 0 \\
 \text{false add substance neighboured to existing substance: } \text{Allow_add}_i = \text{False}, \text{ where } \mathbf{J}_i(\mathbf{E}_{i^*}(\mathbf{m})) < 0 \\
 C_1 (\text{vec}(\{\mathbf{i}/\mathbf{m}_i = \mathbf{1}\})) \\
 C_2 (\mathbf{i}_{\text{set}}^*) \\
 \dots \text{ may be, another additional constraints} \\
 \\
 \text{if } \Delta \mathbf{I}_i(\mathbf{m}) < 0 \\
 \hline
 \text{false add substance neighboured to existing substance: } \text{Allow_add}_i = \text{False}, \text{ where } \mathbf{J}_i(\mathbf{E}_{i^*}(\mathbf{m})) > 0 \\
 \text{false subtract substance from existing substance: } \text{Allow_sub}_i = \text{False}, \text{ where } \mathbf{J}_i(\mathbf{E}_{i^*}(\mathbf{m})) < 0 \\
 C_1 (\text{vec}(\{\mathbf{i}/\mathbf{m}_i = \mathbf{1}\})) \\
 C_2 (\mathbf{i}_{\text{set}}^*) \\
 \dots \text{ may be, another additional constraints}
 \end{array}
 \right. \tag{A.16}$$

Finally, add substance where **Allow_add** is True and subtract substance where **Allow_sub** is True. It may be repeated a lot of times till the BESO is not converged.

A.4 Derivation of formulas for fast computations of Jacobian matrix row $J(x_{i^*})$

According to (3.4), the suggested TO problem for $T \equiv D$ is the following:

$$\begin{aligned}
 \phi &= \max_{i \in g} |x|_i^2(m), \\
 \text{subject to} & \begin{cases} m_i \in \{0, 1\}, \forall i \in \overline{1, p}, \\ g = \{i \mid m_i = 0 \text{ } i \in \overline{1, p}\} \end{cases}
 \end{aligned} \tag{A.17}$$

TO problem formulations (3.4) and (A.17) are equivalent. Let one consider the gradient of the equivalent objective function ϕ for maximal element of field vector solution $x(m)$. According to (A.17):

$$\begin{aligned}\frac{\partial|x^*|^2}{\partial m} &= \frac{\partial\overline{x^*(m)}}{\partial m}x^*(m) + \frac{\partial x^*(m)}{\partial m}\overline{x^*(m)} = \\ &= \frac{\partial x^*(m)}{\partial m}\overline{x^*(m)} + \frac{\partial x^*(m)}{\partial m}\overline{x^*(m)} = \\ &= 2\operatorname{Re}\left\{J(x^*(m))\overline{x^*(m)}\right\}.\end{aligned}\quad (\text{A.18})$$

Thus, only the i^* -th **row** of Jacobian matrix $J(x(m)) \equiv \frac{\partial x(m)}{\partial m}$ should be known, where $i^* = \arg \max_{i \in g} |x_i(m)|$.

The partial derivative of the first matrix equation of (3.5) is

$$A(m)J(x(m)) = B(m) \quad (\text{A.19})$$

where $J(x(m)) \equiv \frac{\partial x(m)}{\partial m}$, $B(m) \equiv k^2 \cdot (\varepsilon - 1) \cdot G \cdot \operatorname{diag}(x(m))$. One took into account the 3D cubic $p \times p \times p$ -dimensional I^{3D} tensor by p -dimensional $x(m)$ vector multiplication is the $\operatorname{diag}(x(m))$ (I^{3D} is the sparse 3D cubic $p \times p \times p$ -dimensional tensor with p non-zero elements that are equal to 1 and located on the main diagonal of the I^{3D} : $I_{iii}^{3D} = 1, \forall i \in \overline{1, p}$).

However, according to the (A.18), for solving optimization problem (3.4) one will calculate only the **row** of Jacobian matrix $J(x(m))$ using the suggested formula derived from (A.19):

$$\begin{aligned}J^T(x(m)) \cdot A^T(m) &= B^T(m) \Leftrightarrow \\ \Leftrightarrow J^T(x(m)) \cdot y &= B^T(m) \cdot z(m),\end{aligned}\quad (\text{A.20})$$

where system was multiplied on the right by special vector $z(m)$ that depends on sparse vector y ,

such that: $y = \begin{cases} 1, & \text{if } i = i^* \\ 0, & \text{otherwise} \end{cases}$ and $y = \cdot z(m)$. Thus, (A.20) will be the following:

$$J^T(x(m)) \cdot y = J^T(x^*(m)) = B^T(m) \cdot z(m) \quad (\text{A.21})$$

Let one consider $B^T(m) \cdot z(m)$. According to (3.5) and (A.19):

$$\begin{aligned}B^T(m) \cdot z(m) &= k^2 \cdot (\varepsilon - 1) \cdot \operatorname{diag}(x(m)) \cdot \\ &\quad \cdot G \cdot z(m) = k^2 \cdot (\varepsilon - 1) \cdot \operatorname{diag}(x(m)) \cdot G * z(m) \\ \Rightarrow \mathbf{J}^T(\mathbf{x}^*(\mathbf{m})) &= k^2 \cdot (\varepsilon - 1) \cdot \operatorname{diag}(\mathbf{x}(\mathbf{m})) \cdot \mathbf{G} * z(m)\end{aligned}\quad (\text{A.22})$$

because G is approximately (relative error is 10^{-7}) symmetric in our problem statement and any

diagonal matrix is symmetric. $\mathbf{J}^T(\mathbf{x}^*(\mathbf{m}))$ — is the linear operator corresponding to the matrix $J^T(x^*(m))$. Ditto, according to (3.5):

$$\begin{aligned}
 A^T(m) &= \{I - k^2 \cdot (\varepsilon - 1) \cdot G \cdot \text{diag}(m)\}^T = \\
 &= I - \text{diag}(m) \cdot G^T = I - k^2 \cdot (\varepsilon - 1) \cdot \text{diag}(m) \cdot G \\
 A^T(m) \cdot z(m) &= [I - k^2 \cdot (\varepsilon - 1) \cdot \text{diag}(m) \cdot G] \cdot z(m) = \\
 &= z(m) - k^2 \cdot (\varepsilon - 1) \cdot \text{diag}(m) \cdot G * z(m) \\
 \Rightarrow \mathbf{A}(\tilde{\mathbf{m}}) &= \mathbf{I} - k^2 \cdot (\varepsilon - 1) \cdot \text{diag}(\mathbf{m}) \cdot \mathbf{G}*,
 \end{aligned} \tag{A.23}$$

where $\mathbf{A}(\tilde{\mathbf{m}})$ —is the linear operator that corresponds to the matrix A^T ; \mathbf{I} — is the linear operator that corresponds to the identity matrix I ; $\text{diag}(\mathbf{m})$ —is the linear operator that corresponds to the diagonal matrix $\text{diag}(m)$; and $\mathbf{G}*$ is the linear operator that executes the FFT matvec for multiplication of matrix G by vector represented in the next to $\mathbf{A}(\tilde{\mathbf{m}})$ operator.

Thus, according to (A.20), (A.21), (A.22) and (A.23), one resolves the following system:

$$\begin{cases} i^* = \arg \max_{i \in g} |x_i(m)| \\ y_i = \begin{cases} 1, & \text{if } i = i^* \\ 0, & \text{otherwise} \end{cases} \\ \mathbf{A}(\tilde{\mathbf{m}}) \cdot z(m) = y \\ \mathbf{J}^T(\mathbf{x}^*(\mathbf{m})) = k^2 \cdot (\varepsilon - 1) \cdot \text{diag}(\mathbf{x}(\mathbf{m})) \cdot \mathbf{G}*z(m) \end{cases} \tag{A.24}$$

Described equations were implemented in Python programming language. In order to implement evolutionary structural algorithm, we used discrete gradient method and greedy method separately. Both methods starts building up device topology domain ω with the central pixel $r_{i_c j_c}$ of design domain Ω ($M = \begin{cases} 1, & \text{if } i = i_c \ j = j_c \\ 0, & \text{otherwise} \end{cases}$). Then they iteratively add substance next door to the device topology domain boundary $\partial\omega$ so that target function ϕ increases. Furthermore, they optionally can subtract substance so that target function ϕ increases. Moreover, they optionally may center ω in Ω in order to keep opportunity to build up ω longer. However, greedy method adds only 1 pixel that requires calculation of FFT matvecs the number of which is the number of neighbors. Nevertheless, discrete gradient method requires two FFT matvecs (for Jacobian matrix row calculation) to execute optimal adding or optimal subtracting (see (A.10) and (A.24)). Moreover, discrete gradient method have the ability to execute these operations for all possible pixels.