Subwavelength Electromagnetic Confinement

BY

SHYAM PRABU AROKIASWAMY
B.E., Anna University, India, 2008

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2013

Chicago, Illinois

Defense Committee:

Jingjing Li, Chair and Advisor
Sharad R. Laxpati
Danilo Erricolo

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisor Professor Jingjing Li for the continuous support of my research; and for his motivation, enthusiasm and immense knowledge. His guidance helped me throughout my research and towards writing this thesis.

I would also like to thank my thesis committee Professor Sharad Laxpati and Professor Danilo Erricolo for their encouragement and insightful comments.

Furthermore, I am grateful to my colleague Rui Yang for his helpful discussion and critical review of this thesis report. Also thanks to the members of XSEDE TACC who helped me resolve the issues during the execution of code in the cluster environment.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation grant number OCI-1053575.

SPA

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

XSEDE	Extreme Science and Engineering Discovery Environment
TACC	Texas Advanced Computing Center
EM	Electromagnetics
GA	Genetic Algorithm
FDTD	Finite-Difference Time-Domain
DNA	Deoxyribonucleic acid
FFT	Fast Fourier Transform
FIFO	First In First Out
MPI	Message Passing Interface
SPMD	Single Program Multiple Data
GNU	GNU's Not Unix
PF	Purcell Factor
SPP	Surface Plasmon Polaritons
TM	Transverse Magnetic
MEEP	MIT Electromagnetic Equation Propagation
APS	Antennas and Propagation Society

SUMMARY

In recent years the use of optimization techniques to determine superior solutions to engineering problems is highly emphasized in the design process. The application of electromagnetic (EM) theory to radiation and scattering problems often either requires or benefits from the use of optimization. Among the many traditional optimization techniques, Genetic Algorithm (GA) proves to be effective for EM device optimization. GAs are robust, stochastic-based search methods that mimic the process of natural evolution. The rule of the game is survival of the fittest.

In this research, a dielectric-based grating structure is designed and it is capable of tightly focusing the radiation at the point of interest. The design parameters for the dielectric structure are obtained from the optimal solution achieved by running the GA within the given constraints. The optimization is carried out in a clustered environment where the dielectric structure is simulated in MEEP, an open source FDTD package to determine the electric and magnetic field patterns. The strong confinement of radiation with the presence of dielectric grating structure is evident from the time domain results. The near field time-domain results are transformed to frequency domain to obtain the far-field radiation pattern for the designed dielectric structure. This illustrates that the radiation pattern steers toward the end-fire view from the broadside view of the dipole pattern. The dielectric grating structure also seems to display the characteristics of 'superlens' and hence can be experimented for high-resolution sub-diffraction imaging. The effect of radiation enhancement is also observed by calculating the Purcell Factor (PF)

with the presence of the dielectric structure. Due to its high spontaneous emission, there is significant reduction in lifetime of atoms leading to full radiative decay resulting in 100% quantum efficiency. This designed dielectric grating can be easily fabricated and embedded with optical devices due to its planar form giving it a competitive advantage over its peers.

An application of this dielectric grating structure is in noninvasive hyperthermia, a type of cancer treatment where body tissue is exposed to high temperatures. Ideally, this kills the cancer cells with minimal injury to normal tissues. However, normal tissues often get destroyed because the radiation is not confined to the tumor cells. Hence, it is necessary that the radiation should be tightly focused to a spot of small size, which can be achieved with the optimally designed dielectric grating structure. It can also find its use in other applications requiring concentrated radiation.

CHAPTER 1

INTRODUCTION

Radiation Confinement has been studied vigorously in both electromagnetics and optics regime for various kinds of applications involving optical fibers, solar cells, optical sensors, integrated circuits and storage of information. Different structures were experimented by researchers including gratings, subwavelength apertures, optical cavities, photonic crystals, metal nano structures and nano particles (1), (2), (3), (4), (5), (6), (7). Often, the physics behind radiation enhancement in these structures are attributed to the presence of Surface Plasmon Polaritons (SPP) (8). Surface Plasmons or SPP's are excited electrons oscillating at the boundary between two metal/dielectric surfaces due to the illumination by light. These excited surface plasmons travel on the surface of the structure outward but are non-radiative usually. When these start to become radiative, there is high intensity in the local density of the states leading to the enhancement in radiation. The detailed discussion of SPP's and their characteristics can be found in (9), (10), (11), (12), (13).

The use of micron scale circular dielectric cylinders and spheres to attain a sub-diffraction focus directly behind the structure called 'Photonic Nanojets' has also been experimented in (14) and (15). It has been showed that the photonic nanojets enhance the backscattering of light by nano particles located within the jets. The properties of the photonic nanojets are reviewed in detail in (16). With the development of transformation optics (17), researchers have shown that subwavelength confinement of light is possible (18). Optical cavities were also

designed by employing transformation optics, to attain the subwavelength confinement of light (19). The use of metamaterials to confine light is also studied in (20).

While most of the field confinement and enhancement methods in literature seem to be convincing, they are often overlooked. The usage of optical cavities to enhance the radiation suffers from material imperfections and fabrication issues and there is a huge probability that the radiation can get trapped inside the cavity. Other structures also face issues due to losses in metals. Hence, we need a simpler structure to concentrate the radiation, which is also easy to fabricate and has very minimal or no losses. In this work, we have used dielectric grating with circular ring grooves for confining the radiation. Even though, gratings with subwavelength apertures were used to confine light, most of the works discussed here employs metals or plasmonic structures with periodic patterning. The designs of the structures for confinement and enhancement were mostly based on theoretical assumptions. We decided to give a degree of freedom in the design process by placing constraints only on the radius of the structure. The power of an optimization algorithm is exercised in the design process to gain competitive advantage over contenders. The selection and implementation of the optimization algorithm is discussed in chapters 2 and 3. The simulation results that display strong electromagnetic confinement and enhancement are discussed in chapter 4.

After the design of the structure for radiation confinement, it would also be interesting to exploit other characteristics of the designed structure. The effect of the structure on the dipolar emission is studied by employing the reciprocity principle of electromagnetics. The radiation pattern is steered towards the end-fire view from the usual broadside dipole radiation

pattern, which are demonstrated from the far field results. Purcell Factor is also calculated to see the effects on spontaneous emission. Analysis of these results can be found in Chapter 4. Conclusions and future work are reviewed in chapter 5.

CHAPTER 2

GENETIC ALGORITHM OPTIMIZATION

Optimization is a process of achieving best results with limited resources. In any engineering design, researchers employ different optimization methods to get the best solution. The optimization methods employed vary from application to application. When classical optimization algorithms are too slow, heuristic search algorithms are used to overcome their limits because these are developed with a focus on being capable of escaping local optima. Of the search heuristic algorithms in practice, the set of evolutionary algorithms seems to outperform. In this chapter, a brief description to the most popular type of evolutionary algorithm called Genetic Algorithms is presented. GA's are widely studied, experimented and applied in many fields and various researchers have already attempted to explain GA and its advantages in detail, which can be found in (21), (22), (23), (24) and (25).

2.1 Genetic Algorithms

Genetic Algorithms belong to the class of stochastic search optimization methods and were developed in 1970's by John Holland at University of Michigan (26). They fall specifically under the classification of evolutionary algorithms, as they are inspired by Darwin's theory of evolution by natural selection. In May 1995, Salvatore Mangano stated that 'Genetic algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, solutions you might not otherwise find in a lifetime'.

All known living organisms are made up of cells, which form the basic building blocks of life. These cells contain an organized structure of coiled DNA called chromosomes containing genes. Genes hold the information to build and maintain organisms' cells and pass genetic traits to offsprings. Similarly, the implementation of GA begins with encoding of the problem's design parameters in the chromosome. These chromosomes are normally binary-encoded where every chromosome is a string of bits 0 or 1. Now a group of individuals are created to form a population where the bits in the chromosomes are selected at random. Each individual is designated a fitness value which determines its success in life. The process of defining the fitness evaluation function and coding the design parameters can be said to be the two important processes in GA. In the process of evolution, the individuals in the population are allowed to reproduce to form an offspring, which inherits the characteristics from its parents. In the same way, reproduction is allowed in genetic algorithm by a three-step process:

1. Selection
2. Crossover
3. Mutation

2.1.1 Selection

Selection determines which individuals are allowed to reproduce and which ones deserve to die. This eliminates the unfit individuals from the population. There are many different algorithms developed to make selection more efficient and robust. These include tournament selection, roulette wheel selection and rank selection. The simplest of these selection algorithms

is tournament selection, which is used in our implementation. In tournament selection, N individuals from the population are chosen at random to conduct a tournament. The winner is the individual with the highest fitness and is selected as parent 1. This process is repeated to get the next parent to continue with the processes of crossover and mutation.

2.1.2 Crossover and Mutation

Crossover is done by selecting best parents from the mating pool and exchanging their genes. A probability value is set based on which crossover is performed allowing the degree of freedom to the algorithm. The crossover point is selected randomly and is of different types. The most-widely used type is single point crossover and the probability value assumed is usually assumed to be higher than or equal to 0.7. Mutation is the process of altering the bits in the binary-encoded chromosome to maintain diversity in the population. Mutation is also performed based on the probability and is usually kept low, around 0.1. The processes of mutation and crossover are illustrated in Figure 1.

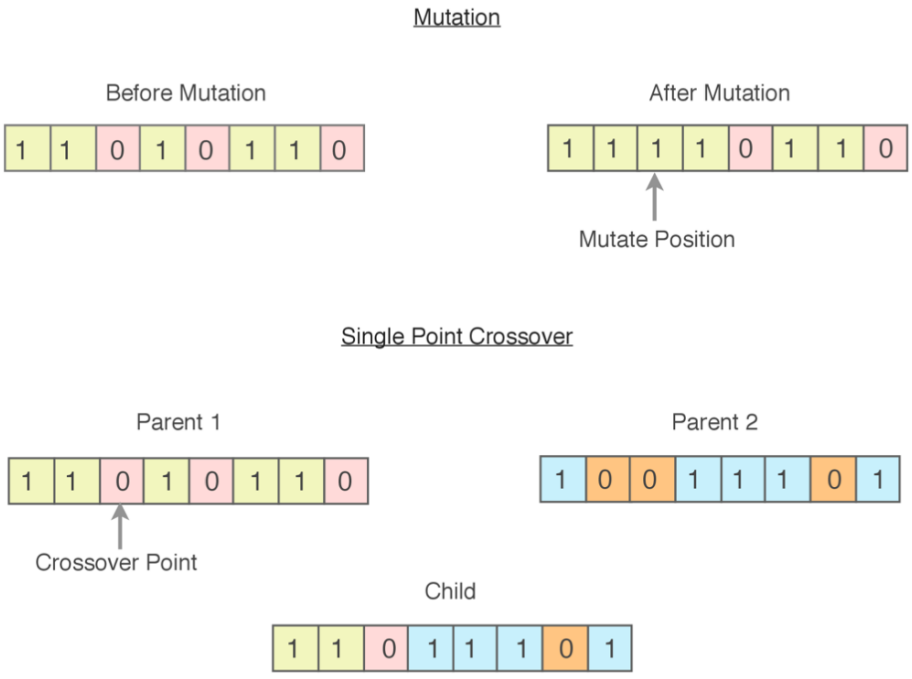


Figure 1. Mutation and Crossover

2.1.3 Simple GA

The process of reproduction is repeated until the size of the population reaches the specified value. At any time the size of the population is kept constant, which means that the unfit individuals are replaced by new offsprings by the above-mentioned process. This process is repeated over several generations until the termination criterion is met as specified. This termination criterion can be set based on the time or fitness evaluations. The simplest genetic algorithm is illustrated in Figure 2 (27).

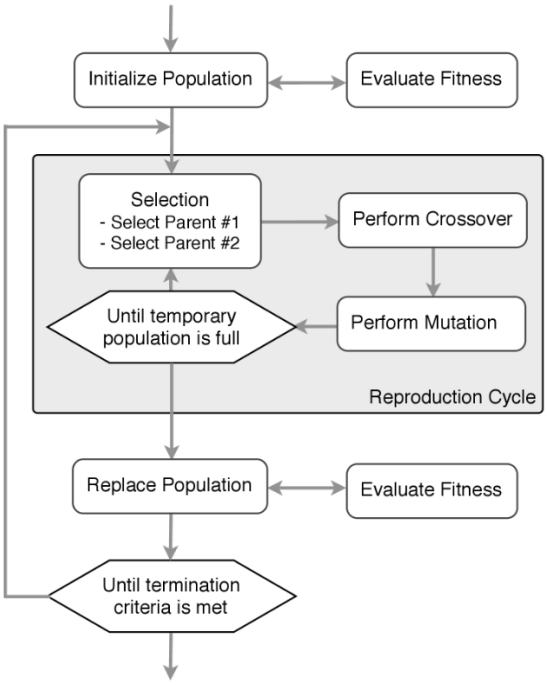


Figure 2. Simple Genetic Algorithm

2.2 Parallel Realization of Genetic Algorithms

Parallel implementation of genetic algorithms has become necessary because of the complex optimization problems resulting in high memory usage and large I/O. With the advancement in the CPU architecture along with the high speed interconnects, the usage of high performance computing resources has increased widely for implementing optimization algorithms. During the parallelization process, care should be taken in writing the code to make the effective use of the multiple processors and memory so that the communication between processors is kept minimal. The three main flavors in which genetic algorithm can be implemented in parallel comprise of

1. Master-Slave approach
2. Coarse-grained or Island GA
3. Fine-grained GA

The easiest and simplest implementation is the master-slave approach, which is used in our implementation. Here the master starts the execution and controls communication and evolution of the solution set. Master sends the individuals of the population to the slaves for fitness evaluations, which are executed in parallel thereby increasing the speed of the algorithm. As and when a slave evaluates the fitness values, a new individual is sent by the master to the slave for evaluation. This process is repeated until the slave receives the termination signal from its Master. Although this method is slower and loses advantages associated with separate evolution of subpopulations, it is adopted mainly due to its ease of implementation. The other two approaches: Coarse-grained and Fine-grained GA's divide the population into subpopulation to evolve independently to explore a wider range of the solution space. A detailed explanation of these two parallel genetic algorithm implementations can be found in (28). The approaches mentioned here can also be combined together for better convergence and fast execution.

2.3 GA Optimization in Electromagnetics

The usage of genetic algorithms is well established in electromagnetics. They have been widely used in electromagnetic problems involving antenna design (29), (30), antenna miniaturization (31), inverse scattering problems (32), optimizing array patterns (33), optimizing element spacing in arrays (34), sidelobe reduction in array pattern synthesis (35), reducing

electromagnetic coupling in shielded enclosures (36), Fiber bragg optimization (37) and optimal design of dielectric gratings (38). Many more applications are discussed in (28) and (39), the two main authors whose contributions have been vital to the area of electromagnetics and are highly commendable.

CHAPTER 3

IMPLEMENTATION FRAMEWORK

3.1 Cluster Computing

Very often applications need more computing power than a normal desktop computer can provide. To overcome this, we form a cluster by connecting high performance workstations through high speed network. This allows us to run programs that needs high computational power, large memory usage and high I/O usage in parallel by distributing the load among its peers. Some brief tutorials on parallel computing can be found in (40), (41), (42). A typical architecture of the computing node in a cluster is shown in Figure 3 (43).

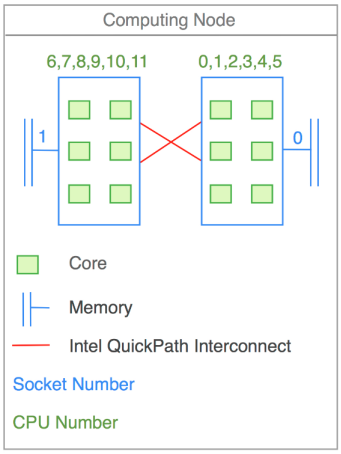


Figure 3. Computing Node - Architecture

3.1.1 Memory Architecture

One of the basic classification of parallel systems is based on memory. This leads us to two distinct hardware architecture models for computing: shared memory and distributed memory. Shared memory architecture has multiple processors that share the same memory resources where changes made by one processor is visible to all other processors. Processors in distributed memory systems have their own local memory and the changes made in local memory have no effect on the memory of other processors. This eliminates race conditions and the need for synchronization. Now when it comes to programming these systems, each architecture has its own parallel programming models but our scope is limited to distributed memory architecture.

3.1.2 Message Passing Interface

The parallel programming methodology in the distributed memory systems is implemented through MPI. MPI is a de facto standard designed for exchanging information between processes. It delineates a collection of routines for facilitating communication among the processes in a distributed memory system. It offers standardization, portability, performance and rich functionality in a number of high quality implementations. The basic outline of a MPI program follows these three general steps:

1. Initialize Communications
2. Communicate to share data between processes
3. Exit in a clean fashion

In Python (44), initialization is handled when importing mpi4py package (45). An instance of the intra communicator is obtained by calling MPI_COMM_WORLD and the number of processes in a communicator and the process rank can be obtained with the methods Get_size() and Get_rank(). For sending and receiving data, blocking (send(), recv()) or non-blocking (Isend() and Irecv()) functions can be invoked. MPI finalization is carried out by python itself when python processes exit.

3.2 Execution Framework

With the Genetic Algorithm completely developed in open source GNU Octave (46) to make the best use of its numerical computation abilities, MPI is implemented in python using the mpi4py package to exploit its simple but effective approach to object oriented programming with dynamic typing and binding. The framework is illustrated in Figure 4. The communication between GNU Octave and master python process is carried out through named pipe (also known as FIFO for its behavior). Named pipe is an extension to the traditional pipe concept in Unix.

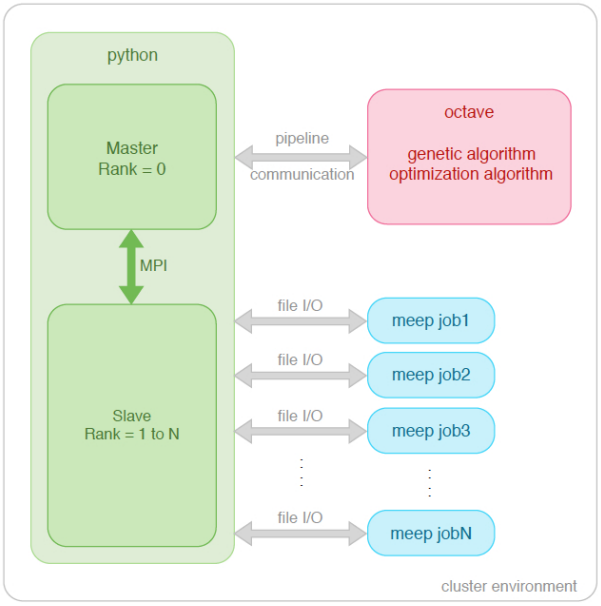


Figure 4. Implementation Framework

Two separate processes can access the pipe by name - one process can open it as a reader, and the other as a writer. The information exchange (fitness value in this case) between slave python process and MEEP (47) is through file I/O.

3.3 Code Progression

The python MPI code in Appendix A is submitted for execution through the batch system by means of a job script. The execution path engaging the effort of one slave in evaluating the fitness is depicted in Figure 5. When the execution starts, the MPI program spawns specified number of processes where each process is designated by its rank. The process whose rank is 0 is the master and all other processes are slaves. The execution control is clarified in the below steps:

1. The master python process starts the octave program which contains the implementation of genetic algorithm.
2. The octave program initiates the genetic algorithm with a random population and sends the details of the population to the master python process via named pipe.
3. After receiving the entire population from octave, master python process splits the work among the slaves that are ready.
4. The slave python processes receive the details of a child from the population and parse its chromosome to get the design parameters for the dielectric structure.
5. The slave python processes then create the MEEP control file(.ctl) and a shell script file. A sample MEEP control file is shown in Appendix B. The control file acts as an input to MEEP and the script file is used to run MEEP. MEEP is executed in parallel by using MPI with the specified number of processes and by binding the processes to cpu cores. This strategy makes effective use of the hardware resources.
6. MEEP writes its output to a file and the python slave process opens the file in read mode to parse the fitness value using regular expressions.
7. The fitness is then sent to master python process which in turn is exchanged with octave for execution of the genetic algorithm.

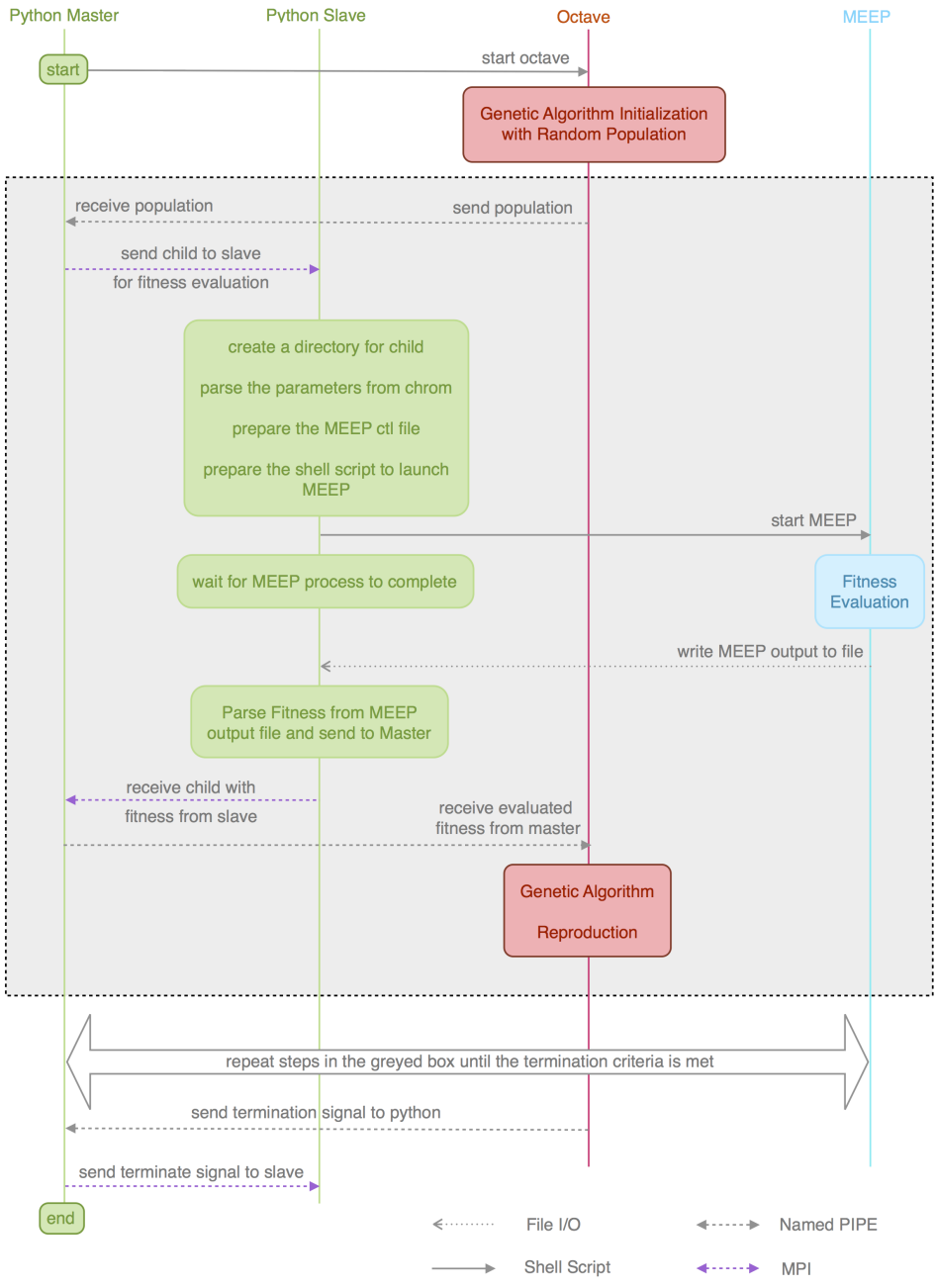


Figure 5. Code Progression Sequence chart

8. During the execution of the algorithm, the population is evolved over multiple generations by reproduction and the fitness of each child is evaluated by sending their details to the master python process. These details are then sent to the slave processes that are waiting for the work from Master. The slaves start the MEEP process which evaluates the fitness to be sent to Octave.

9. The previous step is repeated until the termination criteria is met in the genetic algorithm as mentioned in the octave program.

10. Once the termination criteria is met, octave program completes execution and sends terminate signal to master process. Now the master passes the command to all the slaves to die and it completes the execution.

11. During the execution of the genetic algorithm, the complete fitness evaluations of the entire population consisting of all generations along with the best individuals are saved for future use.

CHAPTER 4

SIMULATION RESULTS

4.1 Strong Radiation Confinement

Researchers have designed optical structures to focus the radiation mainly based on theoretical formalisms. The focus on non-periodic grating structures has gained focus only recently after the researchers from HP labs (48) developed the sub-wavelength grating mirrors with non-periodic patterning to show it has great focusing abilities. This is further endorsed by the simulation of non-periodic 1D and 2D grating structures in (49). In this work, instead of designing the dielectric grating structure based on our notions, the degree of freedom has been given to the Genetic Algorithm to optimize the design. The upshot of the process is a non-periodic dielectric grating which embraces/reinforces the results.

As discussed in chapter 4, defining the fitness evaluation function in a genetic algorithm is crucial, because this is the solitary measure that quantifies the solution to a specified problem. Here individual denotes the solution to the problem under study. In our case, fitness function is defined as the ratio of electric field density at the test point to the integral of the square of the electric field over the entire surface of the grating at the test point. By running the algorithm over many iterations in the designed framework as discussed in chapter 3, the fittest individual is selected for designing the dielectric structure. The binary-encoded chromosome of the individual is parsed to obtain thickness of the substrate and g rooves along with the dimension and position

of the grooves. The electric and magnetic field components are evaluated inside the specified volume by illuminating the grating with the TM polarized gaussian beam source. The typical execution environment in MEEP is shown in Figure 6. All the simulations are carried out at 195.75 THz (λ=1.532 μm) but can be scaled to any wavelength. A typical cylindrical dielectric grating when fabricated looks like the one in Figure 7. The designed dielectric grating can be seen in Figure 8(a).

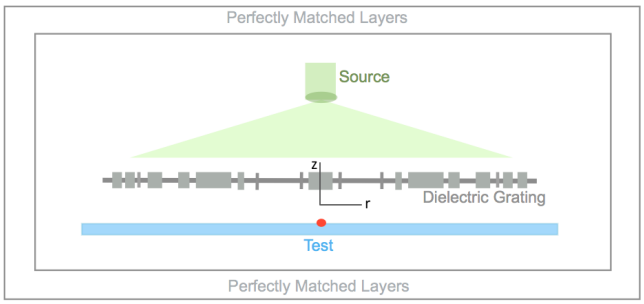
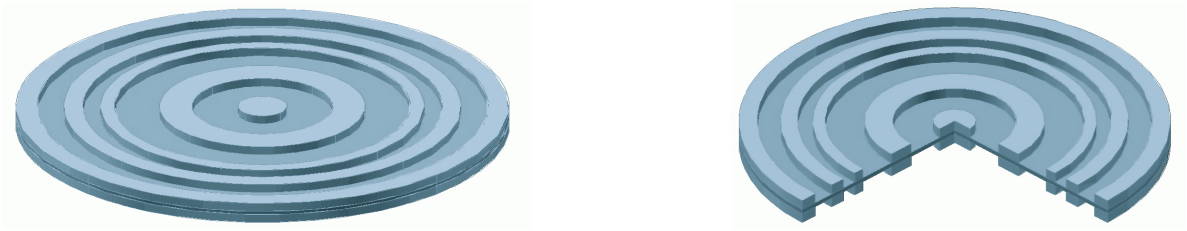


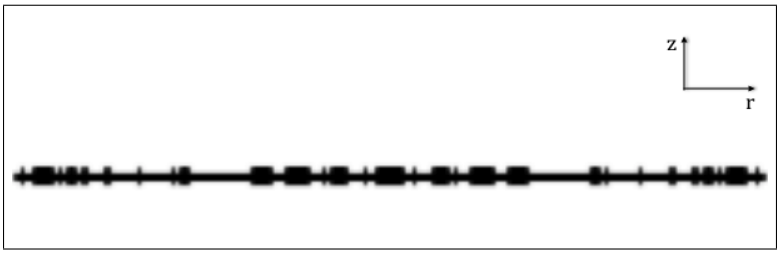
Figure 6. Meep Environment



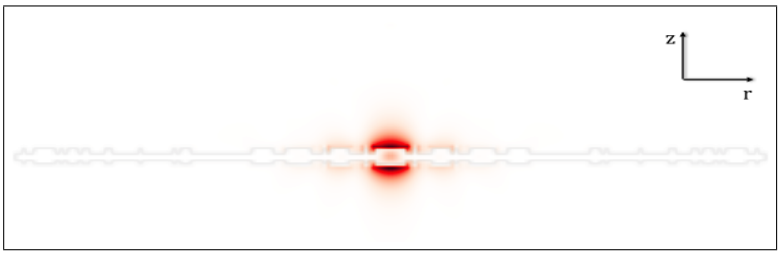
(a) Schematic View

(b) Perspective View

Figure 7. 3D view of the cylindrical dielectric grating



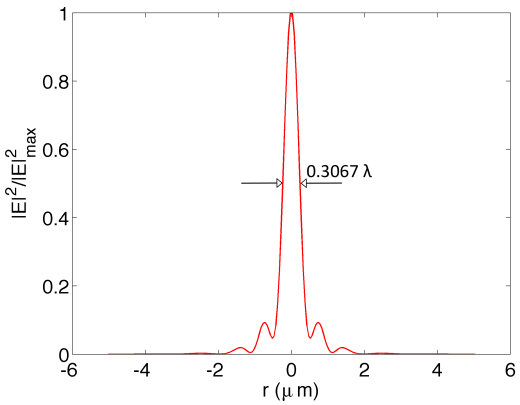
(a) 2d view of dielectric grating with thickness = 0.3125μm and radius = 5μm (Refractive Indices: White - 1, Black - 3.48)



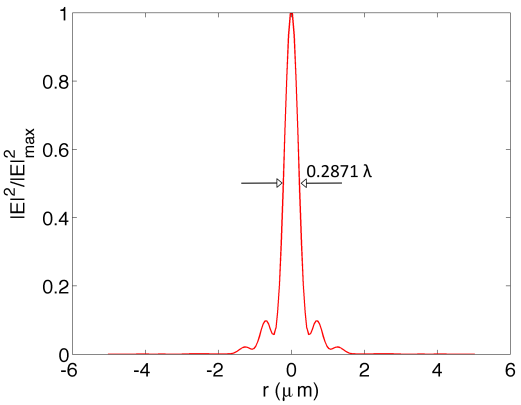
(b) $|\mathbf{E}|^2$ field pattern

Figure 8. Dielectric grating optimized for maximum transmission

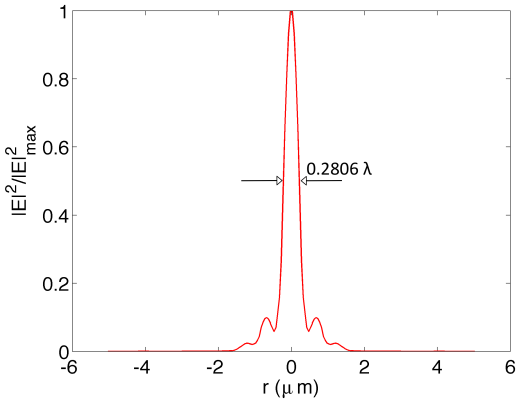
This is a cylindrical structure uniform in ϕ direction and symmetric across r direction with non-uniform grooves. The square of the electric field $|\mathbf{E}|^2$ pattern calculated in the specified volume is shown in Figure 8(b). As it can be seen from Figure 9, the value of $|\mathbf{E}|^2$ is very high at the center of the grating (r=0) and diminishes as we move away from the center. The width of the beam is strongly narrowed and tightened as it gets diffracted from the grating with a full width half maximum of $0.28\,\lambda$. The field enhancement in each case is expressed by calculating the ratio of $|\mathbf{E}|^2_{max}$ with grating to $|\mathbf{E}|^2_{max}$ without grating.



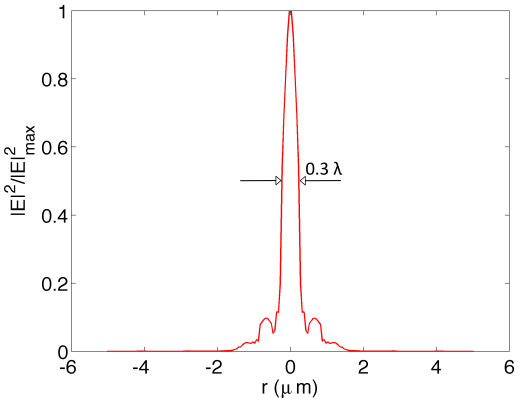
(a) At 0.25λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 155.96$



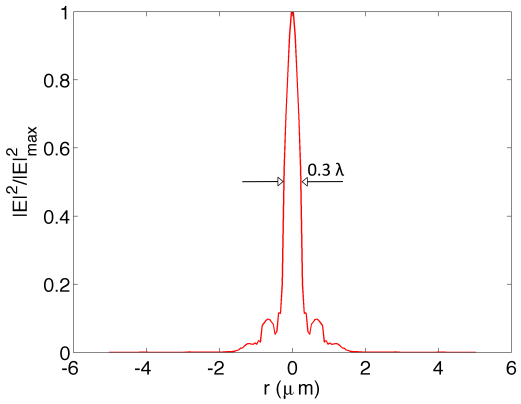
(b) At 0.1λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 589.49$



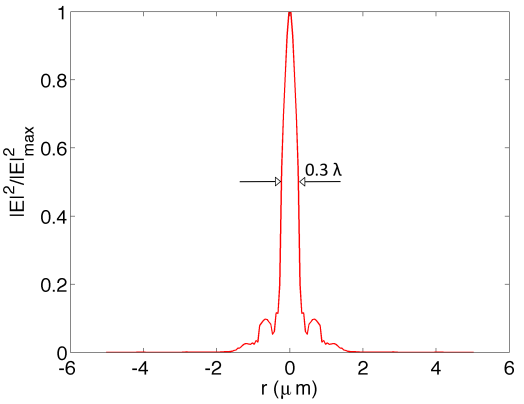
(c) At 0.05λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 1282.19$



(d) At 0.025λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 1938.10$



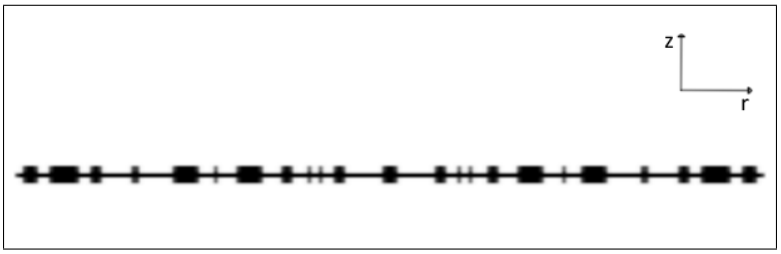
(e) At 0.0125λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 1938.10$



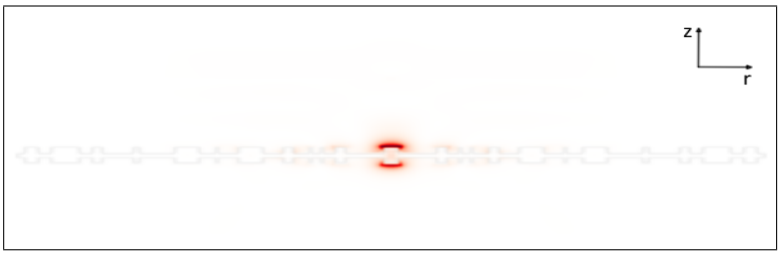
(f) At 0.01λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 1938.10$

Figure 9. FWHM of the normalized $|\mathbf{E}|^2$ distribution along r (in μm) at different distances measured from the bottom of the grating to the test point

With the same design constraints, the optimization is carried out with the goal of achieving maximum transmission on the other side (ability of the grating to reflect the incoming radiation). The optimized dielectric grating can be seen in Figure 10(a) and the $|\mathbf{E}|^2$ field pattern is shown in Figure 10(b).



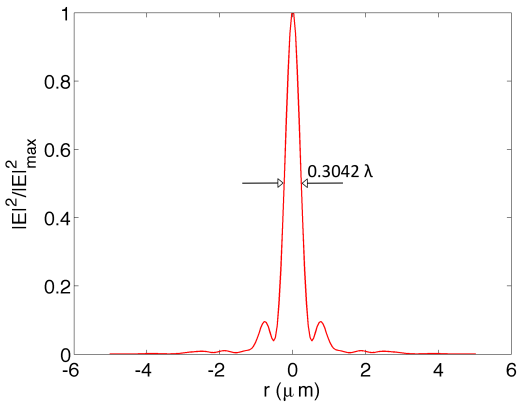
(a) 2d view of Dielectric Grating with thickness = 0.5μm and radius = 5μm (Refractive Indices: White - 1, Black - 3.48)



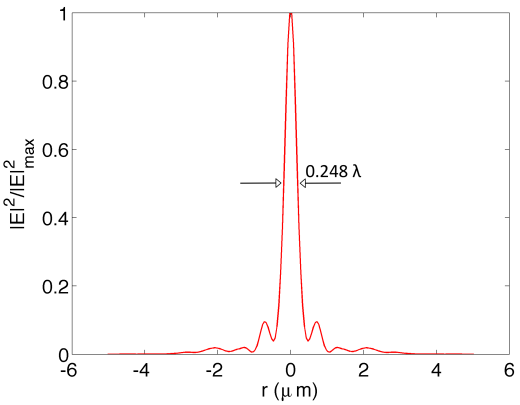
(b) $|\mathbf{E}|^2$ field pattern

Figure 10. Dielectric grating optimized for reflection

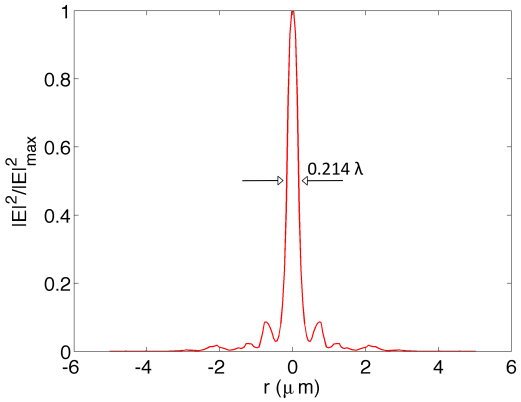
The $|\mathbf{E}|^2$ field distribution at different distances from the top of the grating is shown in Figure 11. The FWHM calculated in this case is very less ($\approx 0.21\,\lambda$). Although the intensity of the field is enhanced, it is less than that observed in the previous case and hence depending on the application, the appropriate design can be employed.



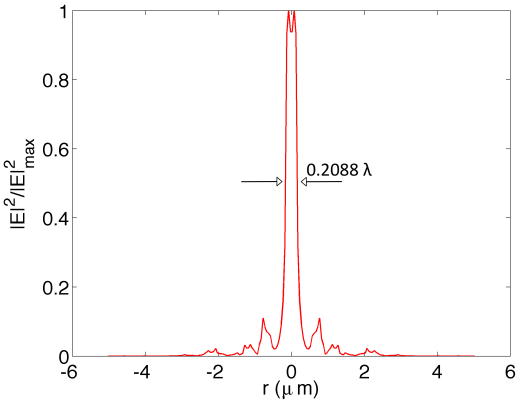
(a) At 0.25λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 40.77$



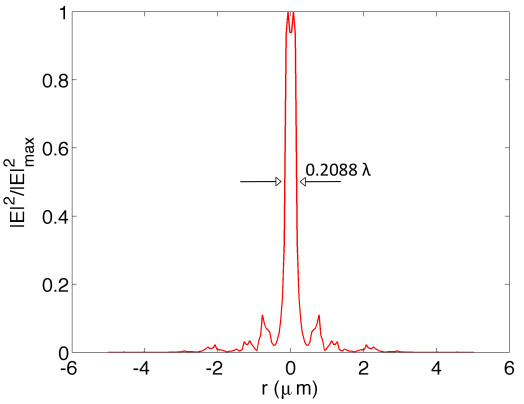
(b) At 0.1λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 138.67$



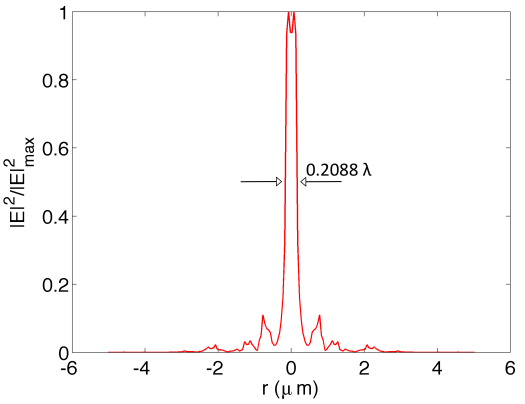
(c) At 0.05λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 358.51$



(d) At 0.025λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 625.44$



(e) At 0.0125λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 625.44$



(f) At 0.01λ with $|\mathbf{E}|^2_{max}/|\mathbf{E}_0|^2_{max} = 625.44$

Figure 11. FWHM of the normalized $|\mathbf{E}|^2$ distribution along r (in μm) at different distances measured from the top of the grating to the test point

In both the cases discussed here, there is an extraordinary confinement in radiation also leading to strong field enhancement. The confinement in radiation is because of the high refractive index of the grating (silicon) compared with the surrounding medium (air). The field improvement is due to the resonant enhancement of the evanescent waves. These characteristics of the designed dielectric structure is similar to what is observed in 'superlens' (50), a perfect lens that uses metamaterials to go beyond the diffraction limit.

4.1.1 Applications

The confined beam from the grating structure finds its usefulness in many applications including non-invasive cancer treatment. A typical setup for microwave ablation uses either a waveguide or an antenna array to achieve a focused microwave spot (51) and (52). However, current applicators based on conventional microwave device suffers from diffraction and the size of the spot has a theoretical lower limit of $\sim \lambda$, where λ is the propagating wavelength of the microwave inside human tissue. At the frequency used (0.9-2.4GHz), λ is between 2 to 5cm in most human tissues according to the measure data from (53), (54) and (55). This would be too large for tumors that are often a few centimeters in size. Hence there is a need for new devices that are capable of tightly focusing the microwave beam to a spot much smaller than is currently achievable. We have accomplished this goal with our optimal design of the non-periodic dielectric grating structure.

Other applications of the designed dielectric structure include but are not limited to: 1. Coupling the wave in to the waveguide thereby increasing the coupling efficiency, 2. Solar Cells, 3. Confining light in optical fibers, 3. In radiation implosion, a process that uses high

levels of electromagnetic radiation to compress a target, which is useful in generating power with fusion energy by inertial confinement approach (56).

4.2 Far Field Behavior

It is also interesting to study the effect of the designed dielectric grating on the emitter placed nearby the structure. To do this, a point dipole source polarized to z direction is placed at the focus to examine the total radiated power from the dipole and the far zone radiation pattern. The behavior might be predicted according to reciprocity theorem of electromagnetics. The theorem states that (57), when there are two sets of sources $\mathbf{J}_1, \mathbf{M}_1$ and $\mathbf{J}_2, \mathbf{M}_2$ that are allowed to radiate simultaneously or individually inside the same medium that is enclosed by a sphere of infinite radius and produce fields $\mathbf{E}_1, \mathbf{H}_1$ and $\mathbf{E}_2, \mathbf{H}_2$, assuming that the sources are positioned within a finite region and that the fields are observed in the far field, then

$$\iiint_V (\mathbf{E}_1 \cdot \mathbf{J}_2 - \mathbf{H}_1 \cdot \mathbf{M}_2)\, dv' = \iiint_V (\mathbf{E}_2 \cdot \mathbf{J}_1 - \mathbf{H}_2 \cdot \mathbf{M}_1)\, dv'$$

Notice that, in the context of antenna, the reciprocity is usually interpreted as the transmitting radiation pattern is the same as the receiving pattern. However, here we are interested in the influence of the grating to a dipole source and we apply the reciprocity theorem in its original form. Using this reciprocity theorem, we expect a strong far-zone radiation in the z direction. This is unusual because here the z direction is the end-fire direction of the dipole, while a dipole source in free space is expected to radiate in the broadside. The far field pattern is obtained by

carrying out near field to far field transformation in frequency domain as explained here. From (58), we get in the far zone

$$\mathbf{E}(\hat{\mathbf{r}}) = \frac{e^{-jkr}}{4\pi r} \oiint_S [j\omega\mu\hat{r} \times (\hat{r} \times \mathbf{J}_s(\mathbf{r}')) + jk\hat{r} \times \mathbf{M}_s(\mathbf{r}')]e^{jk\hat{r}\cdot r'} dS' \tag{4.2.1}$$

$$\mathbf{H}(\hat{\mathbf{r}}) = \frac{e^{-jkr}}{4\pi r} \oiint_S [j\omega\mu\hat{r} \times (\hat{r} \times \mathbf{M}_s(\mathbf{r}')) - jk\hat{r} \times \mathbf{J}_s(\mathbf{r}')]e^{jk\hat{r}\cdot r'} dS' \tag{4.2.2}$$

where $\mathbf{J}_s(\mathbf{r}') = \hat{n}' \times \mathbf{H}_s$ and $\mathbf{M}_s(\mathbf{r}') = \mathbf{E}_s \times \hat{n}'$ are the equivalent electric and magnetic surface currents. In the far zone, only transverse components of the fields exist. To take the transverse components, cross product (Equation 4.2.1) by $\hat{r}$ from the right and we get E_θ and E_ϕ respectively, as

$$E_\theta = -j\frac{e^{-jkr}}{4\pi r} \oiint [\omega\mu\mathbf{J}_s \cdot \hat{\theta} + k\mathbf{M}_s \cdot \hat{\phi}]e^{jk\hat{r}\cdot r'} dS' \tag{4.2.3}$$

$$E_\phi = -j\frac{e^{-jkr}}{4\pi r} \oiint [\omega\mu\mathbf{J}_s \cdot \hat{\phi} - k\mathbf{M}_s \cdot \hat{\theta}]e^{jk\hat{r}\cdot r'} dS' \tag{4.2.4}$$

The surface area in (Equation 4.2.3) and (Equation 4.2.4) can be written as the sum of surface areas on the 2 discs of the cylinder and the cylinder itself.

$$E_{\theta,\phi} = \underset{topdisc,S_1}{\oiint} + \underset{bottomdisc,S_3}{\oiint} + \underset{cylinder,S_2}{\oiint} \tag{4.2.5}$$

Now consider an infinitesimal area element on the surface of the cylinder and on the disc of the cylinder as in Figure 12. We can write their areas as $dS_2' = \rho\, d\alpha\, dz\, \hat{\rho}$ and $dS_{1,3}' = \rho\, d\alpha\, d\rho\, \hat{z}$.

The next step is to calculate $\mathbf{J}_s \cdot \hat{\theta}, \mathbf{M}_s \cdot \hat{\phi}, \mathbf{J}_s \cdot \hat{\phi}, \mathbf{M}_s \cdot \hat{\theta}$ in (Equation 4.2.3) and (Equation 4.2.4). For surface of the cylinder, S2 and surface of the top disc, S1, the electric and magnetic current densities can be written as:

$$\mathbf{J}_2 = \hat{n}' \times \mathbf{H}_2 = \hat{\rho} \times \hat{\phi} \cdot \mathbf{H}_\phi = \hat{z}\mathbf{H}_\phi \implies \mathbf{J}_2 \cdot \hat{\theta} = \hat{z} \cdot \hat{\theta}\mathbf{H}_\phi = -sin\theta\, \mathbf{H}_\phi \tag{4.2.6}$$

$$\mathbf{M}_2 = \mathbf{E}_2 \times \hat{n}' = (\hat{\rho}\, \mathbf{E}_\rho + \hat{z}\, \mathbf{E}_z) \times \hat{\rho} = \hat{\phi}\mathbf{E}_z \implies \mathbf{M}_2 \cdot \hat{\phi} = \hat{\phi} \cdot \hat{\phi}\mathbf{E}_z = \mathbf{E}_z \tag{4.2.7}$$

$$\mathbf{J}_1 = \hat{n}' \times \mathbf{H}_1 = \hat{z} \times \hat{\phi} \cdot \mathbf{H}_\phi = \hat{\rho}\mathbf{H}_\phi \implies \mathbf{J}_1 \cdot \hat{\theta} = -\hat{\rho} \cdot \hat{\theta}\mathbf{H}_\phi = -cos\theta\, \mathbf{H}_\phi \tag{4.2.8}$$

$$\mathbf{M}_1 = \mathbf{E}_1 \times \hat{n}' = (\hat{\rho}\, \mathbf{E}_\rho + \hat{z}\, \mathbf{E}_z) \times \hat{z} = -\hat{\phi}\mathbf{E}_\rho \implies \mathbf{M}_1 \cdot \hat{\phi} = -\hat{\phi} \cdot \hat{\phi}\mathbf{E}_\rho = -\mathbf{E}_\rho \tag{4.2.9}$$

The dot product in the above equations is calculated with the aid of Figure 13. Similarly the equivalent surface currents on the bottom disc surface, S3 can be obtained as $\mathbf{J}_3 \cdot \hat{\theta} = cos\theta\, \mathbf{H}_\phi$ and $\mathbf{M}_3 \cdot \hat{\phi} = \mathbf{E}_\rho$.

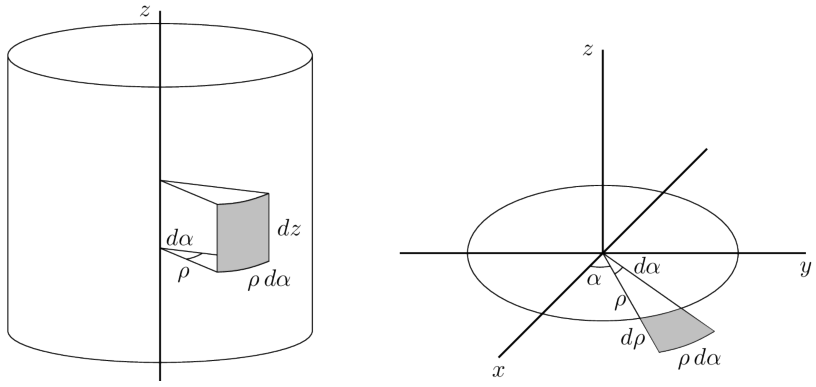


Figure 12. Area element for a cylinder and disc

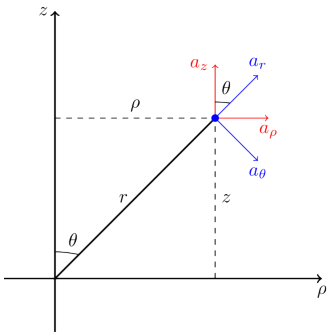


Figure 13. Relationship between cylindrical and spherical coordinates

The final step is to determine the exponential term in (Equation 4.2.3) and (Equation 4.2.4).

$$\hat{r} \cdot r' = (\hat{x}\,\sin\theta cos\phi + \hat{y}\,sin\theta sin\phi + \hat{z}\,cos\theta).(\hat{x}\,\rho cos\alpha + \hat{y}\,\rho sin\alpha + \hat{z}\,b) \tag{4.2.10}$$

$$\hat{r} \cdot r' = sin\theta\,cos\phi\,\rho cos\alpha + sin\theta\,sin\phi\,\rho sin\alpha + b\,cos\theta \tag{4.2.11}$$

Now expanding (Equation 4.2.3), we have

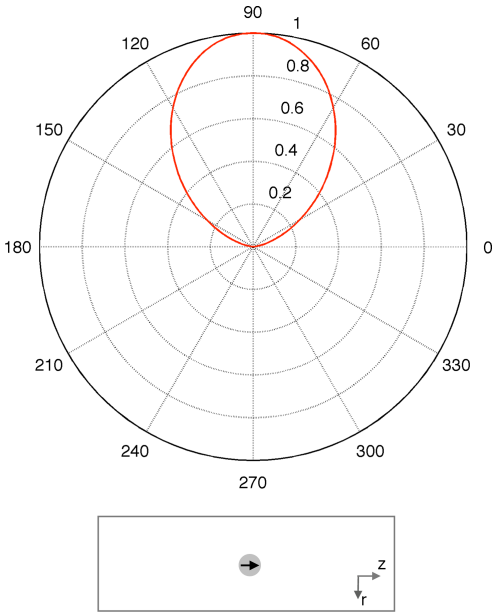
$$E_\theta\Big|_{@S1} = -jk\frac{e^{-jkr}}{4\pi r}\iint[-\eta\,cos\theta\,\mathbf{H}_\phi - \mathbf{E}_\rho]\cdot$$
$$exp\{jk(sin\theta\,cos\phi\,\rho cos\alpha + sin\theta\,sin\phi\,\rho sin\alpha + b\,cos\theta)\}\rho\,d\alpha\,d\rho \tag{4.2.12}$$

$$E_\theta\Big|_{@S2} = -jk\frac{e^{-jkr}}{4\pi r}\iint[-\eta\,sin\theta\,\mathbf{H}_\phi + \mathbf{E}_z]\cdot$$
$$exp\{jk(sin\theta\,cos\phi\,\rho cos\alpha + sin\theta\,sin\phi\,\rho sin\alpha + b\,cos\theta)\}\rho\,d\alpha\,dz \tag{4.2.13}$$

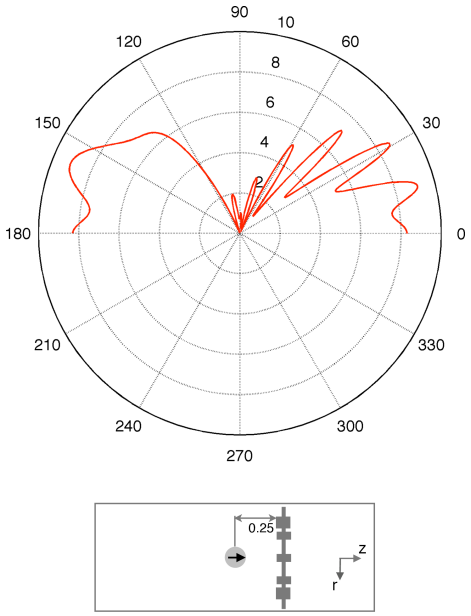
$$E_\theta\Big|_{@S3} = -jk\frac{e^{-jkr}}{4\pi r}\iint[\eta\,cos\theta\,\mathbf{H}_\phi + \mathbf{E}_\rho]\cdot$$
$$exp\{jk(sin\theta\,cos\phi\,\rho cos\alpha + sin\theta\,sin\phi\,\rho sin\alpha + b\,cos\theta)\}\rho\,d\alpha\,d\rho \tag{4.2.14}$$

Using (Equation 4.2.12), (Equation 4.2.13), (Equation 4.2.14) in (Equation 4.2.3), we can determine E_θ. Similarly, E_ϕ can be calculated.

From the above transformation, we can plot the E-field radiation pattern. To verify the transformation, a dipole is placed in the volume with no dielectric structure to obtain a donut pattern as in Figure 14(a). Now with the presence of the dielectric structure, far field pattern is obtained as in Figure 14(b). As it can be seen, the beam is steered towards the end-fire view from the normal broadside view of the dipole pattern which is unusual and is interesting in quantum domain. There is also considerable enhancement in the electric field intensity which can be attributed to the transformation of evanescent waves into propagating waves by the dielectric structure. By changing the distance between the source and the grating and/or changing the design parameters, we might be able to steer the beam in the desired direction.



(a) Dipole Pattern - Broadside view



(b) Presence of Grating (End-fire view)

Figure 14. Far-Field Radiation Pattern (Note: Figure (b) is normalized to the maximum intensity in Figure (a))

4.3 Purcell Enhancement

The possibility of modifying the rate of spontaneous emission life time was first proposed by Purcell (59). This phenomenon called as Purcell Effect describes the amount of spontaneous emission enhancement in a resonant structure. This factor can be directly related to the efficiency of the designed dielectric structure and hence calculation of this value is of vital importance in our work. The emission rate is proportional to the local density of optical states which in turn is proportional to the power emitted by the source, so we can obtain the Purcell factor in an environment simply by ratio of the power emitted by a dipole source in the environment with the presence of dielectric structure to the power emitted by the dipole in a homogenous environment (60).

$$\text{Purcell Factor} = \frac{\text{Total Radiated Power(in the presence of grating)}}{\text{Total Radiated Power of Dipole}}$$

There is a considerable Purcell enhancement at the design frequency as it can be seen from the Purcell Factor vs frequency plot as in Figure 15. This shows high spontaneous emission resulting in full radiative decay and this means that the quantum efficiency is 100%. Often Purcell enhancement comes with dissipative and material loss but since dielectric is used in this design, the structure incurs no such loss. Furthermore, the design is planar and so it is easy to fabricate and inexpensive. These distinguishing characteristics gives the designed dielectric structure an edge, making it a suitable candidate to be integrated in optical devices.

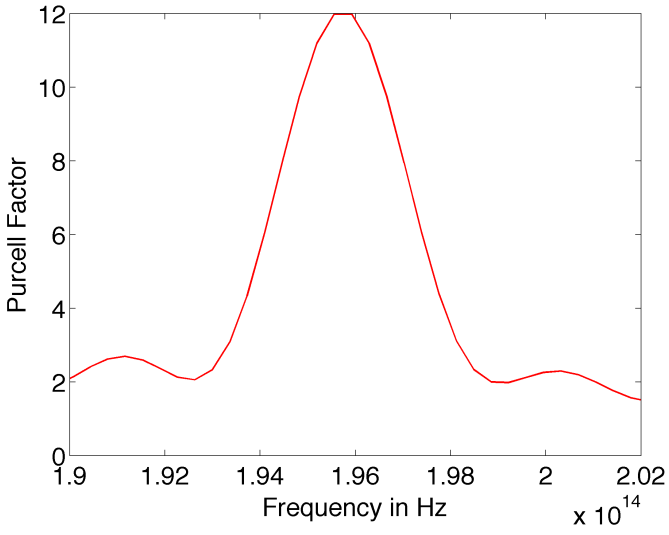


Figure 15. Frequency vs Purcell Factor

CHAPTER 5

SUMMARY AND FUTURE WORK

In this work, the design and characteristics of dielectric gratings with subwavelength grooves have been discussed with a particular emphasis on non-periodic patterning. These gratings have been shown to be able to focus the microwave radiation tightly on a spot suitable for radiation confinement applications. The design also displayed the characteristics of a superlens and hence can be experimented for sub-diffraction imaging by going beyond the conventional diffraction limit. The flat physical geometry not only makes them promising for radiation confinement applications and to embed in optical devices, but also presents opportunities for other possible applications.

In chapters 2 and 3, the optimization algorithm and the execution framework which can be easily adopted for any electromagnetic optimization were described. The parallel genetic algorithm framework presented in this research along with the use of open source tools can be easily leveraged for any kind of electromagnetics optimization problem. In the future, one could readily adapt this framework to other applications by 1: modifying the design parameters and 2: defining the fitness evaluation function accordingly. The other parallel genetic algorithm methodologies discussed in chapter 2 can be implemented and added to the available framework, which increases the degree of freedom to the beneficiary.

Chapter 4 discusses the FDTD simulation results of using the dielectric structure optimized for transmission and reflection. The strong confinement in radiation is demonstrated by cal-

culating the FWHM of the electric filed intensity variation. This also showed that there is a significant enhancement in the electric field intensity as a result of resonance in the designed structure.

Finally, the reciprocity of the design is also investigated and the far field behavior of the structure is also investigated. The Purcell enhancement calculations show significant spontaneous emission characteristics in the presence of the dielectric structure resulting in 100% quantum efficiency. There is no material loss in using the dielectric structure and the fabrication is simple and inexpensive. Hence these kinds of dielectric structures have superior advantages when compared to its peers like optical cavities and metal nano-structures that suffer from fabrication complexities and material losses.

APPENDICES

Appendix A

MPI PYTHON PROGRAM

```
#!/usr/bin/python

from mpi4py import MPI
from sys import argv as sysargv
import config
import result
import generalUtils
import parallelVariables
import os
import scipy
import shlex
import subprocess
import sys
import time


class GA_Individual:

    def __init__(self, ID=scipy.int32(1), chrom=scipy.array([0, 0],
                                dtype=scipy.uint32), fitness=0.0):
        self.ID = scipy.int32(ID)
        self.chrom = scipy.uint32(chrom)
        self.fitness = fitness

    def get_ID(self):
        return self.ID

    def get_chrom(self):
        return self.chrom

    def get_fitness(self):
        return self.fitness

class GeneNumError(Exception):

    def __init__(self, GeneNum, InputGeneNum):

        Exception.__init__(self)
        self.GeneNum = GeneNum
        self.InputGeneNum = InputGeneNum

class GA_Population:

    def __init__(self, chromlength=config._chromLength):
        self.ChromLength = scipy.int32(chromlength)
```

Appendix A (Continued)

```
        self.GeneNum = scipy.int32(scipy.ceil(scipy.float64(chromlength)
                                                              / 32.0))
        self.LastID = 0
        self.Individuals = []

    def add_individual(self, newind=GA_Individual()):
        if len(newind.chrom) != self.GeneNum:
            raise GeneNumError(self.GeneNum, len(newind.chrom))

        self.Individuals.append(newind)
        self.LastID += scipy.int32(1)
        return len(self.Individuals)

    def get_bestfit(self):
        Bestfit = 0
        for indvidual in self.Individuals:
            if individual.get_fitness() > Bestfit:
                Bestfit = individual.get_fitness()

        return Bestfit

    def get_individuals(self, indlist=[]):
        IndividualList = []
        if len(indlist) == 0:
            IndividualList = self.Individuals
        else:
            for ind in indlist:
                IndividualList.append((self.Individuals)[ind])

        return IndividualList

    def get_individual(self, ind):
        return (self.Individuals)[ind]

    def print_individuals(self, indlist=[]):
        for ind in indlist:
            print 'ID: ', ind.get_ID(), 'Chrom: ', ind.get_chrom()

    def get_popsize(self):
        return len(self.Individuals)

    def delete_individuals(self, indlist):
        for ind in indlist:
            self.Individuals.pop(ind)

Comm_World = MPI.COMM_WORLD
GA_Size = Comm_World.Get_size()
GA_Rank = Comm_World.Get_rank()
time.sleep(60)
chromlength = config._chromLength
Cores_4_Optim = config._coresPerJob  # Cores_4_Optim should > 1
assert Cores_4_Optim > 1
```

Appendix A (Continued)

```
genenum = config._geneNo
PopulationSize = config._populationSize
TotalFitEvaluation = config._totalFitEvaluation
Restart_Every = config._restartEvery
RetainSize = config._retainSize
TotalRunTimeStr = 1200

if GA_Rank == 0:
    outstr = ('TotalRunTime = ', TotalRunTimeStr)
    print outstr

# This is specific to SGE. Might change if a different job scheduler is used
JobIDStr = (os.environ)['JOB_ID']

PIPE_PO_Pre = 'PIPE_PtoO' + '_' + JobIDStr
PIPE_OP_Pre = 'PIPE_OtoP' + '_' + JobIDStr
PIPE_PO = PIPE_PO_Pre + '_' + str(GA_Rank)
PIPE_OP = PIPE_OP_Pre + '_' + str(GA_Rank)

if os.path.lexists(PIPE_PO):
    if os.path.exists(PIPE_PO):
        os.unlink(PIPE_PO)
    else:
        os.remove(PIPE_PO)

if os.path.lexists(PIPE_OP):
    if os.path.exists(PIPE_OP):
        os.unlink(PIPE_OP)
    else:
        os.remove(PIPE_OP)

os.mkfifo(PIPE_PO)
os.mkfifo(PIPE_OP)

if GA_Rank == 0:

    octcmd = './ga_MST_start ' + PIPE_PO_Pre + ' ' + PIPE_OP_Pre + ' ' + \
        str(chromlength) + ' ' + str(PopulationSize) + ' ' + \
        str(Restart_Every) + ' ' + str(RetainSize) + ' ' + \
        str(TotalFitEvaluation) + ' ' + \str(TotalRunTimeStr)
    args = shlex.split(octcmd)
    octprocess = subprocess.Popen(args)

    print 'PY.GA_Rank == 0. Octave process for global GA control started.\n'
    population = GA_Population(chromlength)

    if not os.path.exists(config._fitRecordFile):
            print 'PY.GA_Rank==0. Could not find fitnessrecord.mat file.\n'
            infile = open(PIPE_OP, 'rb')
        for loop in range(PopulationSize):
            NewIndv_ID = scipy.fromfile(infile, dtype=scipy.int32, count=1)[0]
```

Appendix A (Continued)

```
        NewIndv_chrom = scipy.fromfile(infile, dtype=scipy.uint32,
                                                    count=genenum)
        NewIndv = GA_Individual(ID=NewIndv_ID, chrom=NewIndv_chrom,
                                fitness=0.0)
        population.add_individual(NewIndv)

    infile.close()
    print 'PY. GA_Rank == 0.  Initial population received.\n'
else:
    print 'PY. GA_Rank == 0.  Population loaded from the existing \
                    fitnessrecord.mat file.\n'

# Initializing Environment for Parallel Job
parallelParams = generalUtils.initializeEnvironment()
parallelParams.displayParallelParams()

SolvedPop = GA_Population(chromlength)

while True:  # Doing optimization until breaks
    if not os.path.exists(config._fitRecordFile):
        print 'PY. GA_Rank == 0.  No fitnessrecord.mat file.\n'
        for WorkerRank in range(1, GA_Size):
            Message_Status = MPI.Status()
            Message_Flag = Comm_World.Iprobe(source=WorkerRank, tag=1,
                status=Message_Status)  # Probe JobDone signal

            if Message_Flag:  # Job is done
                      # Take the JobDone message
                MessageTake = Comm_World.recv(source=WorkerRank, tag=1)
                SolvedIndv = Comm_World.recv(source=WorkerRank, tag=11)
                SolvedPop.add_individual(SolvedIndv)
                print 'PY. Received the child with evaluated fitness \
                                from worker...'

            if population.get_popsize() == 0:
                break

            Message_Flag = Comm_World.Iprobe(source=WorkerRank, tag=2,
                status=Message_Status)  # Probe Head ready signal

            if Message_Flag:  # Worker is ready
                    # Take the Head ready signal
                MessageTake = Comm_World.recv(source=WorkerRank, tag=2)
                Individual = population.get_individual(0)
                JobID = Individual.get_ID()
                JobChrom = Individual.get_chrom()
                HostFile = (parallelParams.hostFileForSlots)[WorkerRank-1]
                coreMapping = (parallelParams.coreMappings)[WorkerRank-1]

                Comm_World.send(JobID, dest=WorkerRank, tag=15)
                Comm_World.send(JobChrom, dest=WorkerRank, tag=16)
                Comm_World.send(HostFile, dest=WorkerRank, tag=17)
```

Appendix A (Continued)

```
                Comm_World.send(coreMapping, dest=WorkerRank, tag=18)
                population.delete_individuals([0])

    outfile = open(PIPE_PO, 'wb')

    SolvedSize = SolvedPop.get_popsize()
    print 'Sending solved size to octave : ', SolvedSize
    #SolvedSize could be zero.If so, this serves as a hand shaking signal.
    scipy.array([SolvedSize], dtype=scipy.int32).tofile(outfile)

    if SolvedSize >= 2:
        for popind in range(SolvedSize):
            CurrentIndividual = SolvedPop.get_individual(0)
            JobID = CurrentIndividual.get_ID()
            scipy.array([JobID], dtype=scipy.int32).tofile(outfile)
            JobChrom = CurrentIndividual.get_chrom()
            JobChrom.tofile(outfile)
            JobFitness = CurrentIndividual.get_fitness()
            scipy.array([JobFitness],dtype=scipy.float64).tofile(outfile)
            SolvedPop.delete_individuals([0])

    outfile.close()
    print 'Reading Evolved Population from Octave...'
    infile = open(PIPE_OP, 'rb')  # receive the updated population
    popsize = scipy.fromfile(infile, dtype=scipy.int32, count=1)[0]

    if popsize < 0:
        print 'GA_Rank == 0. Negative popsize received.  Terminating.'
        infile.close()
        break
    else:
        for loop in range(popsize):
            NewIndv_ID = scipy.fromfile(infile, dtype=scipy.int32,
                    count=1)[0]
            NewIndv_chrom = scipy.fromfile(infile, dtype=scipy.uint32,
                    count=genenum)
            NewIndv = GA_Individual(ID=NewIndv_ID, chrom=scipy.array(
                NewIndv_chrom, dtype=scipy.uint32), fitness=0.0)
            population.add_individual(NewIndv)
        infile.close()

outstr = 'PY.  GA_Rank == 0.  Terminating the optimization ... '
print outstr

    # to send terminate signal to Workers
for WorkerRank in range(1, GA_Size):
    # Receive Head-Ready or JobDone signal.
    MessageTake = Comm_World.recv(source=WorkerRank, tag=MPI.ANY_TAG)
    if MessageTake == 1:  #JobDone signal. Receive and dump the done job.
        print 'PY. Last Job Done from slave : ', WorkerRank
        # Receive Head-Ready signal
        MessageTake = Comm_World.recv(source=WorkerRank, tag=MPI.ANY_TAG)
```

Appendix A (Continued)

```
        DummyID = scipy.int32(-1)
        DummyChrom = scipy.zeros((1, genenum), dtype=scipy.uint32)
        DummyCoreMapping = scipy.int32(-1)
        DummyHostFile = 'No Host'
        # A Job with ID == 0 indicates termination.
                DummyID = Comm_World.send(DummyID, dest=WorkerRank, tag=15)
                # A Job with ID == 0 indicates termination.
        DummyChrom = Comm_World.send(DummyChrom, dest=WorkerRank, tag=16)
        DummyHostFile = Comm_World.send(DummyHostFile, dest=WorkerRank,
                                        tag=17)
        DummyCoreMapping = Comm_World.send(DummyCoreMapping, dest=WorkerRank,
                                        tag=18)

else:

    while True:  # Doing optimization until breaks
                # send Team head ready signal.  The data sent is the tag
        Comm_World.send(scipy.int32(2), dest=0, tag=2)
        JobID = Comm_World.recv(source=0, tag=15)
        JobChrom = Comm_World.recv(source=0, tag=16)
        HostFile = Comm_World.recv(source=0, tag=17)
        coreMapping = Comm_World.recv(source=0, tag=18)

        if JobID < 0:
            break

        print 'PY. GA Rank :', GA_Rank, 'Job ID : ', JobID, 'JobChrom : ',
                JobChrom, 'HostFile : ', HostFile, 'core Mapping : ',
                coreMapping, '\n'

        (jobDirName, jobOutput) = generalUtils.get_JobName(JobID,
                                                config._workDir)

        result = generalUtils.chromToParameters(JobChrom)
        if result.errorGauge:
            print result.error
            # send Job-Done signal.  The data sent is the tag
            Comm_World.send(scipy.int32(1), dest=0, tag=1)
            Comm_World.send(GA_Individual(ID=JobID, chrom=JobChrom,
                                        fitness= -1), dest=0, tag=11)
            continue

        parameters = result.output
        (MEEPFileName, MEEPFileSta, MEEPFileMessage) = generalUtils.
                make_MeepFile(parameters, jobDirName, config._workDir)

        if MEEPFileSta == 3:
            print 'PY. Creating a job directory for the job failed \
                        for the job : ', jobDirName
        elif MEEPFileSta == 2:
            print 'PY. Creating superlens.ctl file failed for the \
```

Appendix A (Continued)

```
                    job : ', jobDirName
elif MEEPFileSta == 1:
    print 'PY. Creating cattmp1.ctl file failed for the \
                    job : ', jobDirName

if MEEPFileSta != 0:
    # send Job-Done signal.  The data sent is the tag
    Comm_World.send(scipy.int32(1), dest=0, tag=1)
    Comm_World.send(GA_Individual(ID=JobID, chrom=JobChrom,
                            fitness= -1), dest=0, tag=11)
    continue;

MEEPLinkFile = 'meep-mpi'  # the meep command to execute the ctl file

CoreMappingStr = str(int(coreMapping[0]))
for coreloop in range(1, len(coreMapping)):
    CoreMappingStr = CoreMappingStr + ':' + str(int(coreMapping


jobFile = os.path.join(config._workDir, jobDirName,
                                                config._jobSubmission)
jobFileID = open(jobFile, 'w')
jobFileID.write('#!/bin/bash \n')
jobFileID.write('module unload openmpi \n')
jobFileID.write('module load mvapich2 \n')
jobFileID.write('module load meep \n')
jobFileID.write('mpirun_rsh -np %d -hostfile %s MV2_CPU_MAPPING=
               %s %s %s >%s \n'% (Cores_4_Optim, HostFile,
               CoreMappingStr, MEEPLinkFile, MEEPFileName, jobOutput))
jobFileID.close()

result = generalUtils.sysCommand('chmod u+x ' + jobFile)
if result.errorGauge:
    print 'ERROR: The mode of the jobFile could not be changed \
                    to executable. Job not submitted. Job-', jobDirName
    # send Job-Done signal.  The data sent is the tag
    Comm_World.send(scipy.int32(1), dest=0, tag=1)
    Comm_World.send(GA_Individual(ID=JobID, chrom=JobChrom,
                                    fitness= -1), dest=0, tag=11)
    continue

meepcmd = './' + jobFile
args = shlex.split(meepcmd)
meepProcess = subprocess.Popen(args)
(out, err) = meepProcess.communicate()
if err == None:
    JobFitness = generalUtils.parseFitnessFromMeep(JobID,
                                     config._workDir)
else:
    meepProcess.kill()
    print 'Meep Process did not complete in the given time \
                    for the Job ID : ', JobID
```

Appendix A (Continued)

```
            JobFitness = scipy.int32(-1)
        # send Job-Done signal.  The data sent is the tag
        Comm_World.send(scipy.int32(1), dest=0, tag=1)
        Comm_World.send(GA_Individual(ID=JobID, chrom=JobChrom,
                                      fitness=JobFitness), dest=0, tag=11)
        print 'PY. GA_Rank == ', GA_Rank, '. Result for JobID = ',
                JobID, '. Fitness = ', JobFitness, ' sent back to
                GA_Rank ==0.'

os.unlink(PIPE_PO)
os.unlink(PIPE_OP)
print 'GA_Rank ', GA_Rank, ' done!'
```

Appendix B

MEEP CONTROL FILE

```
(print "Job Started")
(define-param thick 0.3125);
(define-param thick_midlayer 0.15625);

(define-param fcen 0.6525); pulse center frequency  1550 nm
(define lambda0 (/ 1 fcen)); Wavelength, in MEEP unit

(define grat_size 5.0)
; the distance from the upper side of the grating to the source
(define-param source_to_grat 2)
; the distance from the lower side of the grating to the test surface
(define-param test_to_grat (* 0.25 lambda0))
(define-param padr 2)
(define-param padz_up 0.5)
(define-param padz_down lambda0)
(define-param dpml 1) ; pml thickness
(define-param sr (+ grat_size padr dpml) )
(define-param sz (+ dpml padz_down test_to_grat thick source_to_grat
                                                  padz_up dpml))
(print "\n")
(print "sr=" sr "\n")
(print "sz=" sz "\n")

(define low_z (- 0 (/ sz 2)))
; center of the grating structure
(define grat_center (+ low_z dpml padz_down test_to_grat (/ thick 2)))
; position of the source
(define source_z (+ grat_center (/ thick 2) source_to_grat))
(define test_z (- grat_center (/ thick 2) test_to_grat))

(define-param eps_silicon (* 3.48 3.48))  ; dielectric constant of slab
(define-param eps_background (* 1 1))  ; dielectric constant of environment

(set! dimensions CYLINDRICAL)
(set-param! m 0)
(set! geometry-lattice (make lattice (size sr no-size sz)))
(set! default-material (make dielectric (epsilon eps_background)))

(define (groove edge1 edge2)
        (list
                (make block (center (* 0.5 (+ edge1 edge2)) 0 grat_center)
                            (size (- edge2 edge1) infinity thick)
                           (material (make dielectric (epsilon eps_silicon)))
                )
                (make block (center (- 0 (* 0.5 (+ edge1 edge2))) 0 grat_center)
```

Appendix B (Continued)

```
                        (size (- edge2 edge1) infinity thick)
                        (material (make dielectric (epsilon eps_silicon)))
                )
        )
)

(set! geometry
        (append
                (groove 0 0.2)
                (groove 0.3 0.35)
                (groove 0.55 0.8)
                (groove 0.85 0.9)
                (groove 1.05 1.4)
                (groove 1.55 1.85)
                (groove 2.65 2.8)
                (groove 2.85 2.9)
                (groove 3.3 3.35)
                (groove 3.7 3.8)
                (groove 4 4.1)
                (groove 4.15 4.3)
                (groove 4.35 4.4)
                (groove 4.45 4.75)
                (groove 4.85 4.9)
                (list
                     (make block (center 0 0 grat_center);
                         (size (* 2 grat_size) infinity thick_midlayer)
                         (material (make dielectric (epsilon eps_silicon)))
                        )
                )
        )
)

;  Temporal width of the sinusoidal source
(define-param temporal_width 10)
; size of the magnetic current
(define-param source_size (+ grat_size padr))

(define-param gb_w0 grat_size)
(define gb_z0 (* (* pi gb_w0 gb_w0) fcen))
(define gb_wzi (* gb_w0 (sqrt (+ 1 (/ (* source_to_grat source_to_grat)
                                                (* gb_z0 gb_z0))))))
(define gb_rzi (+ source_to_grat (/ (* gb_z0 gb_z0) source_to_grat)))
(define  gb_pcoef (/ pi (/ gb_rzi fcen)))
(define gb_mcoef (/ -1.0 (* gb_wzi gb_wzi)))

(print "pcoef=" gb_pcoef "\n")
(print "mcoef=" gb_mcoef "\n")

(define (source_amp p)
        (exp (* (+ gb_mcoef (* 0+1i gb_pcoef)) (* (vector3-x p)
                                                (vector3-x p))))
)
```

Appendix B (Continued)

```
(set! force-complex-fields? true)
(set! sources
        (list
                (make source
                        (src  (make continuous-src (frequency fcen)
                                        (width temporal_width) )  )
                        (component Hp)
                        (center 0 0 source_z) (size source_size infinity 0)
                        (amp-func source_amp)
                )
        )
)

(set! pml-layers (list (make pml (thickness dpml))))
(set-param! resolution 20)

(define-param tim 1000)  ;running time

(define (ESquare r Er Ez) (+ (* (magnitude Er) (magnitude Er))
                             (* (magnitude Ez) (magnitude Ez)) ));

(define (ESquare_max_output)
        (print "ESquare maximum: "
                (get-field-point D-EnergyDensity (vector3 0 0 test_z)
                )
                "\n"
        )
)

(define (ESquare_Integral)
        (print "ESquare Integral: "
                (integrate-field-function (list Er Ez) ESquare (volume (size
                        (* grat_size 2) infinity 0) (center 0 0 test_z))
                )
                "\n"
        )
)

(begin
         (run-until tim
                (at-end
                        ESquare_max_output
                        ESquare_Integral
                )
         )
        (run-until (/ 1 fcen)
                    (in-volume (volume (size (* grat_size 2) infinity sz)
                                    (center 0 0 0))
                        (at-end
                                    (to-appended "ESq" ESquare_output)
                                    (to-appended "eps" output-epsilon)
```

```
                                )
                        )

        )
)
(print "Job Done")
```

CITED LITERATURE

1. Gerard, D., Wenger, J., Devilez, A., Gachet, D., Stout, B., Bonod, N., Popov, E., and Rigneault, H.: Strong electromagnetic confinement near dielectric microspheres to enhance single-molecule fluorescence. Optics Express, 16(19):15297–15303, September 2008.

2. Devilez, A., Bonod, N., Wenger, J., Gerard, D., Stout, B., Rigneault, H., and Popov, E.: Three-dimensional subwavelength confinement of light with dielectric microspheres. Optics Express, 17(4):2089–2094, February 2009.

3. Cubukcu, E., Kort, E. A., Crozier, K. B., and Capasso, F.: Plasmonic laser antenna. Applied Physics Letters, 89(9):093120–093120–3, 2006.

4. Garcia-Vidal, F. J. and Martin-Moreno, L.: Transmission and focusing of light in one-dimensional periodically nanostructured metals. Physical Review B, 66(15):155412, October 2002.

5. Imani, M. and Grbic, A.: Subwavelength focusing with a corrugated metallic plate. In IEEE Antennas and Propagation Society International Symposium, 2009. APSURSI '09, pages 1–4, 2009.

6. Lockyear, M. J., Hibbins, A., Sambles, J. R., and Lawrence, C. R.: Surface-topography-induced enhanced transmission and directivity of microwave radiation through a subwavelength circular metal aperture. Applied Physics Letters, 84(12):2040–2042, 2004.

7. Li, J., Fiorentino, M., Fattal, D., and Beausoleil, R. G.: Strong optical confinement between non-periodic flat dielectric gratings. 106(9).

8. Ritchie, R. H.: Plasma losses by fast electrons in thin films. Physical Review, 106(5):874–881, June 1957.

9. Gramotnev, D. and Bozhevolnyi, S.: Plasmonics beyond the diffraction limit. Feb 2010.

10. Maier, S. A. and Atwater, H. A.: Plasmonics: Localization and guiding of electromagnetic energy in metal/dielectric structures. Journal of Applied Physics, 98(1):011101–011101–10, July 2005.

CITED LITERATURE (Continued)

11. Barnes, W. L., Dereux, A., and Ebbesen, T. W.: Surface plasmon subwavelength optics. Nature, 424(6950):824–830, August 2003.

12. Barnes, W. L.: Surface plasmon polariton length scales: a route to sub-wavelength optics. Journal of Optics A: Pure and Applied Optics, 8(4):S87, April 2006.

13. Pitarke, J. M., Silkin, V. M., Chulkov, E. V., and Echenique, P. M.: Theory of surface plasmons and surface-plasmon polaritons. Reports on Progress in Physics, 70(1):1, January 2007.

14. Chen, Z., Taflove, A., and Backman, V.: Photonic nanojet enhancement of backscattering of light by nanoparticles: a potential novel visible-light ultramicroscopy technique. Optics Express, 12(7):1214–1220, April 2004.

15. Li, X., Chen, Z., Taflove, A., and Backman, V.: Optical analysis of nanoparticles via enhanced backscattering facilitated by 3-d photonic nanojets. Optics Express, 13(2):526–533, January 2005.

16. Heifetz, A., Kong, S.-C., Sahakian, A. V., Taflove, A., and Backman, V.: Photonic nanojets. Journal of computational and theoretical nanoscience, 6(9):1979–1992, September 2009. PMID: 19946614.

17. Ward, A. J. and Pendry, J. B.: Refraction and geometry in maxwell's equations. Journal of Modern Optics, 43(4):773–793, 1996.

18. Pendry, J. B., Aubry, A., Smith, D. R., and Maier, S. A.: Transformation optics and subwavelength control of light. Science (New York, N.Y.), 337(6094):549–552, August 2012. PMID: 22859483.

19. Ginis, V., Tassin, P., Soukoulis, C. M., and Veretennicoff, I.: Confining light in deep subwavelength electromagnetic cavities. (0911.4216), November 2009. Phys. Rev. B 82, 113102 (2010).

20. Grbic, A., Merlin, R., Thomas, E., and Imani, M.: Near-field plates: Metamaterial Surfaces/Arrays for subwavelength focusing and probing. Proceedings of the IEEE, 99(10):1806–1815, 2011.

21. Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Boston, MA, USA, Addison-Wesley Longman Publishing Co., Inc., 1st edition, 1989.

CITED LITERATURE (Continued)

22. Holland, J. H.: Genetic algorithms. Scientific American, pages 66–72, 1992.

23. Whitley, D.: A genetic algorithm tutorial. Statistics and Computing, 4(2):65–85, 1994. Cited By (since 1996):391.

24. Obitko, M.: Introduction to Genetic Algorithms. `http://www.obitko.com/tutorials/genetic-algorithms`, 1998.

25. Fernandez, J. J.: The genetic programming notebook. `http://www.geneticprogramming.com`, 2013.

26. Holland, J. H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, 1975.

27. Johnson, J. and Rahmat-Samii, V.: Genetic algorithms in engineering electromagnetics. IEEE Antennas and Propagation Magazine, 39(4):7–21, 1997.

28. Haupt, R. L. and Haupt, S. E.: Practical genetic algorithms. Hoboken, N.J., John Wiley, 2004.

29. Villegas, F., Cwik, T., Rahmat-Samii, Y., and Manteghi, M.: A parallel electromagnetic genetic-algorithm optimization (EGO) application for patch antenna design. IEEE Transactions on Antennas and Propagation, 52(9):2424–2435, 2004.

30. Jones, E. and Joines, W.: Design of yagi-uda antennas using genetic algorithms. IEEE Transactions on Antennas and Propagation, 45(9):1386–1392, 1997.

31. Azadegan, R. and Sarabandi, K.: A novel approach for miniaturization of slot antennas. IEEE Transactions on Antennas and Propagation, 51(3):421–429, 2003.

32. Qing, A.: Electromagnetic inverse scattering of multiple two-dimensional perfectly conducting objects by the differential evolution strategy. IEEE Transactions on Antennas and Propagation, 51(6):1251–1262, 2003.

33. Ares-Pena, F., Rodriguez-Gonzalez, J., Villanueva-Lopez, E., and Rengarajan, S.: Genetic algorithms in the design and optimization of antenna array patterns. IEEE Transactions on Antennas and Propagation, 47(3):506–510, 1999.

CITED LITERATURE (Continued)

34. Haupt, R.: Optimized element spacing for low sidelobe concentric ring arrays. IEEE Transactions on Antennas and Propagation, 56(1):266–268, 2008.

35. Yan, K.-K. and Lu, Y.: Sidelobe reduction in array-pattern synthesis using genetic algorithm. IEEE Transactions on Antennas and Propagation, 45(7):1117–1122, 1997.

36. Macpherson, R. I. and Ryan, N. J.: Reducing Electromagnetic Coupling in Shielded Enclosures Using a Genetic Algorithm - Finite-Difference Time-Domain Solver. Defense Technical Information Center, 2002.

37. Tremblay, G., Gillet, J.-N., Sheng, Y., Bernier, M., and Paul-Hus, G.: Optimizing fiber bragg gratings using a genetic algorithm with fabrication-constraint encoding. Journal of Lightwave Technology, 23(12):4382–4386, 2005.

38. Johnson, E. G. and Abushagur, M. A. G.: Microgenetic-algorithm optimization methods applied to dielectric gratings. Journal of the Optical Society of America A, 12(5):1152–1160, May 1995.

39. Rahmat-Samii, Y. and Michielssen, E.: Electromagnetic optimization by genetic algorithms. New York, J. Wiley, 1999.

40. Blaise Barney, Lawrence Livermore National Laboratory: Introduction to Parallel Computing. `https://computing.llnl.gov/tutorials/parallel_comp/#MemoryArch`, 2013.

41. Cornell Virtual Workshop: High Performance Computing Topics. `https://www.cac.cornell.edu/VW/default.aspx`, 2013.

42. National Center for Supercomputing Applications: High Performance Computing and CyberInfrastructure (CI). `http://www.citutor.org`, 2013.

43. Xsede: Xsede ranger user guide. `https://www.xsede.org/tacc-ranger`, 2012. [Online; accessed 13-August-2013].

44. Python community: PSF/Python. `http://www.python.org`, 2013.

45. Dalcin, L.: MPI for Python. `http://mpi4py.googlecode.com`, 2012.

46. Octave community: GNU/Octave. `www.gnu.org/software/octave/`, 2012.

CITED LITERATURE (Continued)

47. Oskooi, A. F., Roundy, D., Ibanescu, M., Bermel, P., Joannopoulos, J. D., and Johnson, S. G.: MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method. Computer Physics Communications, 181:687–702, January 2010.

48. Fattal, D., Li, J., Peng, Z., Fiorentino, M., and Beausoleil, R. G.: Flat dielectric grating reflectors with focusing abilities. Nature Photonics, 4(7):466–470, July 2010.

49. Lu, F., Sedgwick, F. G., Karagodsky, V., Chase, C., and Chang-Hasnain, C. J.: Planar high-numerical-aperture low-loss focusing reflectors and lenses using subwavelength high contrast gratings. Optics Express, 18(12):12606–12614, June 2010.

50. Pendry, J. B.: Negative refraction makes a perfect lens. Physical Review Letters, 85(18):3966–3969, October 2000.

51. Sharma, R., Wagner, J. L., and Hwang, R. F.: Ablative therapies of the breast. Surgical oncology clinics of North America, 20(2):317–339, viii, April 2011. PMID: 21377586.

52. Fenn, A. J.: Adaptive phased array thermotherapy for cancer. Boston, Artech House, 2009.

53. Gabriel, C., Gabriel, S., and Corthout, E.: The dielectric properties of biological tissues: I. literature survey. Physics in Medicine and Biology, 41(11):2231, November 1996.

54. Gabriel, S., Lau, R. W., and Gabriel, C.: The dielectric properties of biological tissues: II. measurements in the frequency range 10 Hz to 20 GHz. Physics in Medicine and Biology, 41(11):2251, November 1996.

55. Gabriel, S., Lau, R. W., and Gabriel, C.: The dielectric properties of biological tissues: III. parametric models for the dielectric spectrum of tissues. Physics in Medicine and Biology, 41(11):2271, November 1996.

56. Laser Inertial Fusion Energy(LIFE): Lawrence Livermore National Laboratory. `https://life.llnl.gov/what_is_life/index.php`.

57. Balanis, C. A.: Advanced engineering electromagnetics. New York, Wiley, 1989.

58. Jin, J.-M.: The finite element method in electromagnetics. New York, Wiley, 2002.

CITED LITERATURE (Continued)

59. Purcell, E.: Spontaneous emission probabilities at radio frequencies. In Physical Review, volume 69, page 681, 1946.

60. FDTD Solutions Knowledge Base: Cavities and Resonators: Lumerical Solutions, Inc. http://docs.lumerical.com/en/fdtd/cavity_purcell_factor.html.

VITA

Name:	Shyam Prabu Arokiaswamy
Education:	M.S., Electrical and Computer Engineering University of Illinois at Chicago, IL Aug 2013
	B.E., Electronics and Communication Engineering Anna University, Chennai, India May 2008
Experience:	Richard Daley Library, University of Illinois at Chicago Graduate Assistant - Systems 2012 - 2013
	Tata Elxsi, India Engineer - Broadcast Domain 2009 - 2011
Awards:	IEEE APS Pre-Doctoral Research Award 2013
Membership:	IEEE Student Member