THE PENNSYLVANIA STATE UNIVERSITY
MILLENNIUM SCHOLARS PROGRAM

DEPARTMENT OF MATERIALS SCIENCE AND ENGINEERING

Designing 3D Printed Metamaterials with MOLACO

NATALIE CUMMINGS
SPRING 2021

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Materials Science and Engineering
with honors in Materials Science and Engineering

Reviewed and approved* by the following:

Douglas H. Werner
Professor of Electrical Engineering
Thesis Supervisor

Robert A. Kimel
Professor of Materials Science and Engineering
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

Metamaterials, which gain their properties from their physical structure rather than their elemental composition, can open possibilities in a variety of different fields and applications. By utilizing the Werner lab's MOLACO (Multi-Objective Lazy Ant Colony Optimization) algorithm, nontraditional features with desired properties, such as low or zero refractive index, can be found. This project sought to adapt MOLACO, which was previously used for Frequency Selective Surfaces, for use with metamaterials, by utilizing a GitHub API which allows HFSS to be called from within MATLAB. The API works by creating a script in VisualBasic through MATLAB code which can then implement commands within HFSS. This internal modeling allows the feature to be analyzed via the Anisotropic Inversion Algorithm (AIA), which assesses the feature's correlation to the desired properties. The modeling functionality was completed and tested for 3x3x3 search spaces. This allows each meandered path to be created and analyzed in HFSS without leaving MATLAB. Future work will include the implementation of the AIA and further testing within MOLACO.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I want to thank Dr. Werner for his help, patience, and support in completing this project in the craziest of times. I truly couldn’t have done this without you.

Thanks also goes to Dr. Kimel, for being my reviewer and popping up time and time again throughout my college career in MATSE.

Immeasurable thanks to the Millennium Scholars Program, for all the support I received over my four years. College was such a better experience with all of you standing behind me and helping me to grow and move forward.

I appreciate everything the Schreyer Honors College has given me the opportunity to do, from funding my experiences to hosting fascinating guest speakers, to creating awesome honors classes.

This entire endeavor couldn’t have been done without everyone listed on this page, and my deepest thanks go out to everyone who has supported me along the way.

Chapter 1

Introduction

1.1 Background on Metamaterials

Metamaterials are a class of materials which derive their properties from their physical structure, rather than their elemental composition. Generally, these materials are comprised of one pattern that is repeated over and over again to form a surface with a certain set of desired properties. This class of materials has grown in popularity over the last decade, as examples such as Split Ring Resonators have been created and utilized in the field of transformation optics[1]. This experiment focused on optical metamaterials optimized for low or zero refractive index over a broad range of angles.

Metamaterials have been developed in the past to exhibit zero[2] or even negative[3] refractive indices, however this project seeks to find nontraditional, three dimensional structures. Rather than searching for optimal structures with a guess and check modeling approach, that work can be outsourced to the optimization algorithm and solved in the background. Additionally, other parameters such as the broad angle compatibility can be specified and optimized alongside the refractive index parameter.

Optical metamaterials interact with waves of different frequencies, and can manipulate those waves in numerous different ways, depending on the properties of their physical structure. One of the biggest appeals of optical metamaterials is their ability to exhibit properties which are not found in nature, such as negative or extremely low index refractive indices. Additionally, these properties can be optimized over a wide range of angles, leading to more versatile applications. Since the features and the periodicity of their arrangement is smaller than the waves they are intended to interact with, the entire surface can be modeled with homogeneous properties[4], simplifying the process greatly.

This optimization of desired properties takes place in the Multi-Objective Lazy Ant Colony Optimization algorithm (MOLACO), a program created by Danny Zhu, a former member of the Werner lab. This optimization algorithm will be discussed in great detail in Chapter 2 of this work, to better understand its origins, its methodology, and how it can be adapted to metamaterials.

1.2 Combining Metamaterials with 3D Printing

The interest in combining this material with 3D printing stems from two sources: the ability to print small, accurate formations and the rapid prototyping potential. To an extent, progress in metamaterials is limited by progress in its methods of manufacturing[5]. Additive manufacturing technology has improved significantly over the last decade, with clear advances in the accuracy, speed, and minimum print size. In general, a metamaterial feature must be several times smaller than the wavelength of the wave it is intended to interact with. It follows, then, that these features are often very small and can be difficult to produce. However, 3D printers can easily take on the production of complicated geometries, and new advances allow even small features to be printed accurately. The printer used in this lab, the Stratasys Objet260Connex3, can print horizontal layers as thin as 16 microns[6].

Rapid prototyping allows for the quick testing of a potentially promising feature without significant time or cost investment. The filament used by the printer to create the feature is a photopolymer, Vera White Plus, and since the design's model can be easily created and sent to the printer, there are minimal manufacturing costs. This method can also cut down on waiting times for production and shipment, as a group of 10 or more features could print in the span of a couple of hours and be ready for testing almost immediately.

1.3 Considering Engineering and Design Principles of the Project

In a project such as this one, many factors must be considered for its eventual applications into the real world. This project has few economic issues to deal with, as there are no hazardous materials to be disposed of, and much of the testing is done in a virtual setting, which reduces waste. 3D printing has been suggested for its rapid prototyping ability, in which traditional manufacturing costs can be avoided. However, there are still costs present, such as the cost of the filament, the cost of the conductive spray, and the cost of electricity to power the 3D printer.

Regarding environmental and sustainability concerns, this project could present a couple of potential issues. The features to be tested are printed from polymers, and cannot be easily repurposed. However, the amount of waste is significantly reduced from a 'guess and check' method, as only features which have been shown in modelers to exhibit the desired properties are chosen to be tested physically. Additionally, there is not likely to be significant turnover of the materials in an application setting. Once a specific feature is printed and tested, methods could be explored to choose more sustainable materials for mass production, such as utilizing filament which incorporates waste materials as a part of its makeup. These materials are unlikely to be made obsolete, as a material with a low or zero refractive index could be used in a myriad of different applications without need for serious modification.

The manufacturability of this project is heavily dependent on additive manufacturing, as mentioned above. This allows local, quick manufacture of features to test. This also allows the exact amount of material needed to be printed, rather than being forced to order a minimum number of units from a traditional manufacturing setting which may go to waste. Improvements in 3D printing technology, such as printers with higher resolution and even thinner layers would allow for creating smaller features, opening up possibilities in a wider range of the electromagnetic spectrum.

There are few concerns regarding ethical issues and health and safety. No hazardous materials are being used, nor hazardous waste being created. The risks inherent in the process are very low, as a significant portion of the experiments are performed in a virtual setting. Additive manufacturing is a very safe manufacturing process, with no significant health or safety risks. The application of conductive paint must be performed in a well-ventilated area to avoid inhalation of fumes, but otherwise there are no known health risks associated with the experiment.

Finally, the social and political impacts must be considered. This project is unlikely to raise ethical issues in even a global setting. Work in metamaterials may be driven by political motives, however, as the technology behind it could theoretically be used for cloaking at different wavelengths. This project does not aim for that specific goal, but metamaterials are often lumped into the same category, which may garner interest from the military sector. These potential biases must be considered, and the project must pursue the fundamental scientific goal, rather than a politically motivated one.

Chapter 2

Materials and Methods

2.1 Previous Work on MOLACO

MOLACO, or Multi-Objective Lazy Ant Colony Optimization, is an algorithm originally created by Danny Zhu[7] for use by the Werner group. It has been used in the past to generate nontraditional Frequency Selective Surfaces (FSS) for various applications.

MOLACO is a descendant of other algorithms such as the Ant System[8] and the Max Min Ant System[9]. Ant Colony Optimizations (ACOs) were chosen for their ability to pursue multiple objectives simultaneously by using the pheromones of the digital ants[10]. However, traditional ACOs were limited by their tendency to be space filling. Hence, MOLACO implemented 'lazy' ants to prevent overmeandering, and produce meaningful structures.

Another important distinction of MOLACO from traditional ACOs is its adaptive masking[10]. Masks limit ants to exploring only certain areas of a search space, which can ensure against overmeandering. Different sub-populations of ants are assigned to different masks, and masks that are performing best are weighted to be more likely to be used in future generations. However, to ensure good exploration, no mask is ever taken out of the rotation, it may just have a much lower subpopulation than more fruitful masks. Examples of masking are shown in the figure below.

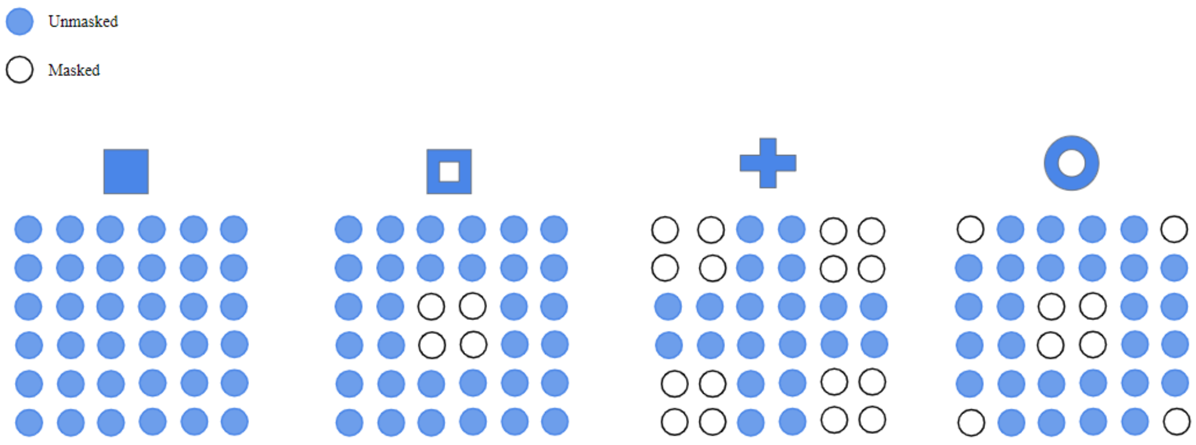


Figure 1: Masks utilized in MOLACO, shown on an example 6x6x1 search space. Masked areas are areas the digital ants cannot move through, whereas unmasked areas can be traversed. Adapted from Danny Zhu[10].

In its original exploration, the algorithm was used to generate unintuitive 3D FSS unit cell geometries. For different applications, MOLACO's objectives can be varied. The optimization parameters which were used for FSS are listed in the table below.

Table 1. Optimization parameters of MOLACO for 3D FSS. Adapted from Danny Zhu[7].

Parameter	Description
N	Number of colonies
P	Population Size
Q	Fitness scaling factor
ρ	Pheromone evaporation rate
τ_{min}	Minimum pheromone concentration
τ_{max}	Maximum pheromone concentration
FBW_d	Desired fractional bandwidth percentage
n_{freqs}	Number of frequency sample per band

n_θ	Number of incidence angles sampled

The validity of MOLACO was proven with the testing of these FSS and the above parameters. After promising nonintuitive FSS features were found, they were tested by first modeling them to confirm the desired properties. Next, they were manufactured using a 3D printer, then metallized with silver conductive spray to provide the desired properties while still using a simpler, polymer-based additive manufacturing technique. Then, these unit cells were arranged into a periodic array, which was tested to confirm agreement of experimental and predicted values of desired properties. After this testing was successful, it was considered how MOLACO may be utilized for other types of devices.

The substitution and modification of the above parameters allow MOLACO to be a generally useful tool, applicable for a wide range of applications. In fact, it may be used for the optimization of such applications as FSS, metamaterials, meander line antennas, optical antennas, or phase gradient metasurfaces. With further work and adaptation such as those taken in this project, the possibilities are many for reutilizing MOLACO in various situations.

2.2 How MOLACO Works

MOLACO runs over multiple ‘generations’, taking the best results from each generation and using those as seeds for the next generation. In theory, the more generations that are specified, the more accurate the final result will be. Algorithms such as MOLACO come in many different styles and variations and can be used to solve numerous different problems.

MOLACO utilizes the behavior of ants for its computer simulations. It builds upon prior algorithms which also used ant colonies as optimization tools, such as the Ant System[8] and the Max Min Ant System[9], discussed above. In MOLACO, the 'digital ants' move through a specified search space, leaving behind 'digital pheromones' as they do so. Future ants sometimes follow these pheromones, and sometimes chart their own path. Generally, the path of only one ant is considered, so that the path will be guaranteed to be contiguous. Multiple ants may be considered in the same search space, however, as noncontiguous parts of features may provide additional resonances or coupling behaviors. Importantly, these ants are lazy, meaning that they will stop moving after a set number of squares explored.

This is enabled via the Phantom Termination Segment, as shown in the figure below.

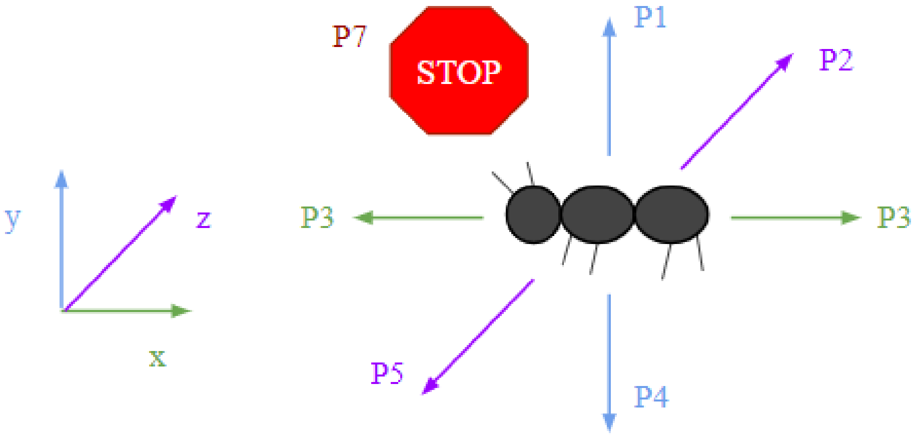


Figure 2: Potential moves of the digital ant, with probabilities P1-P6. P7 is the Phantom Termination Segment, the probability of stopping the path. Only 5 of the probabilities P1-P6 will be considered, as the ant may not backtrack. Adapted from Danny Zhu[7].

As ants are moving through the search space, they can move in one of five directions (the forbidden sixth being the direction they came from, as backtracking is prohibited). Each of these directions is weighted with their own probability, and another option is also included: not moving at all. The Phantom Termination Segment serves as a probability that the ant stops moving, ending its path.

If this end condition were not enabled, the entire search space would be explored each time, as in the Multi-Objective Ant Colony Optimization[11]. Instead, this algorithm is able to create nontraditional feature structures without over-meandering[10], which can then be tested for certain desired properties. The best feature from the first generation can be used as a basis for the next generation, and so on until the given number of generations have been completed and an ideal feature has been provided. This feature can be modeled again in HFSS or other software to confirm its desired properties, and then can be tested physically.

To test these features, several features must be additively manufactured. Then, in order to have the desired properties, these polymer features must be uniformly metallized, or coated in metal conductive paint (usually silver)[7]. The features can then be placed in an array and tested against their predicted property values. The biggest challenge with this method is the time for computation which, for an 8x8x1 search space, took as long as 9 days to compute[11].

In order to use this program for metamaterials, however, some changes must be made. MOLACO can run as usual, but the features must be built in HFSS, analyzed, passed through the Anisotropic Inversion Algorithm, and then passed back into MOLACO with its rating. Establishing and improving this process, with a focus on enabling the modeling, has been the bulk of the project detailed in this paper.

Chapter 3

Results and Discussion

The overall process I created is shown in the following flow chart.

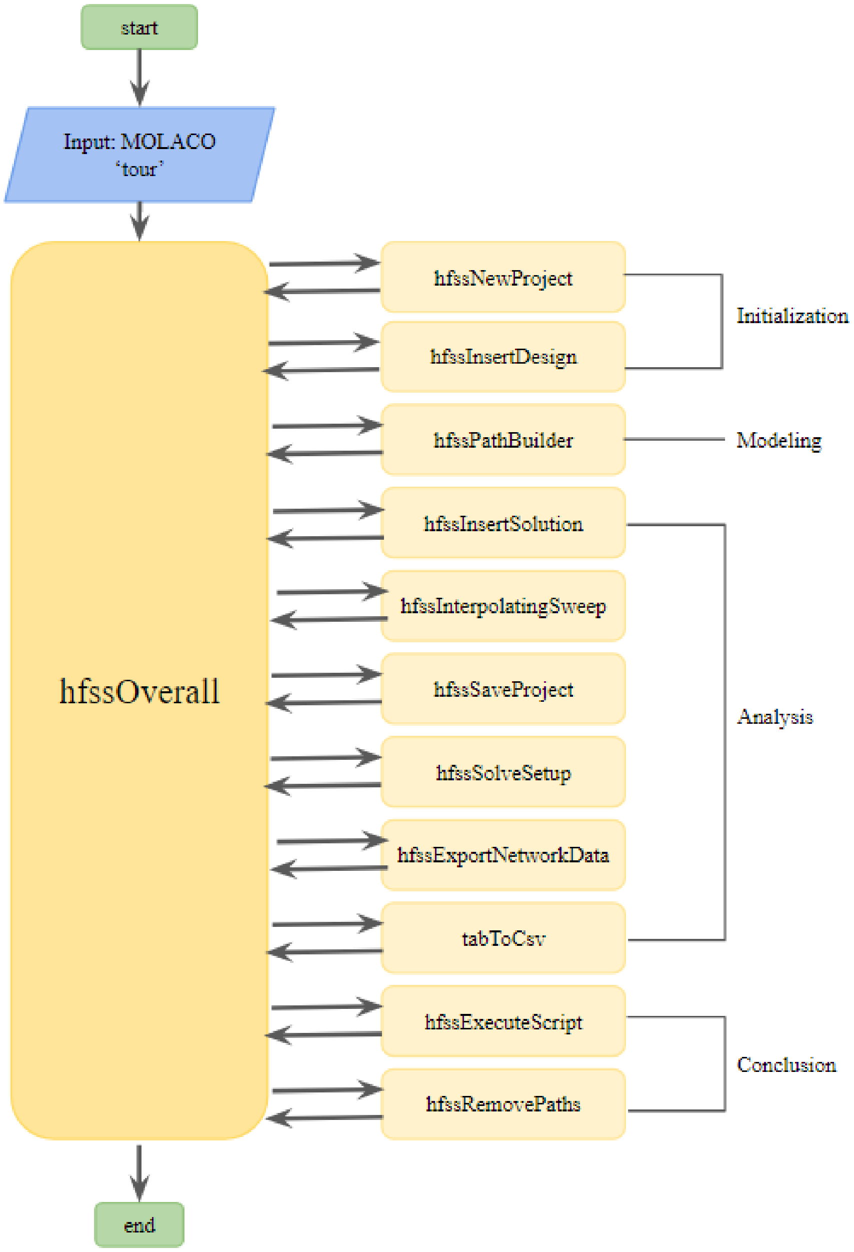
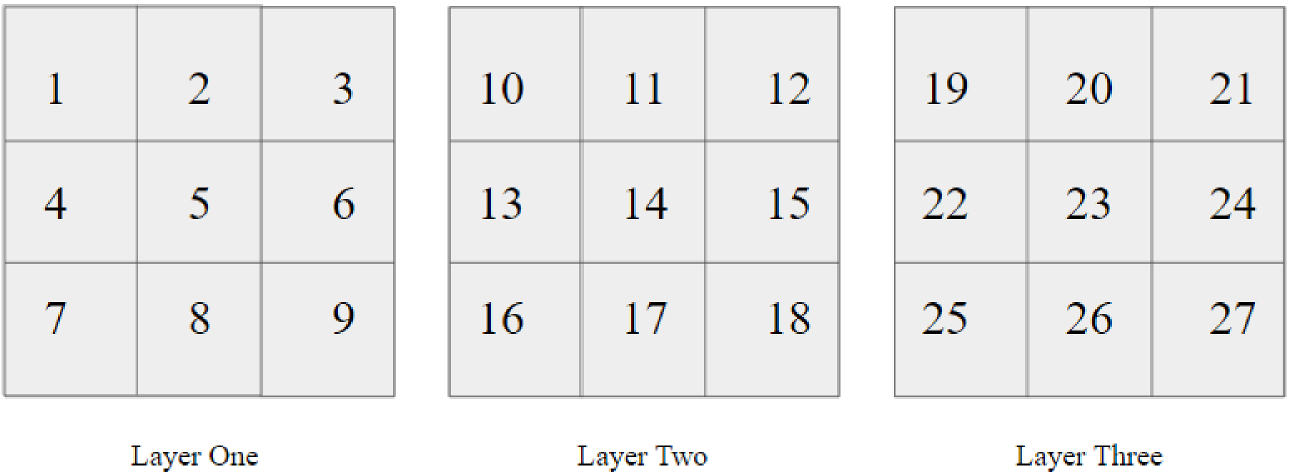


Figure 3: Flow chart of process calling HFSS within MATLAB

The code runs largely within the function hfssOverall, which calls the eleven other functions shown in the flow chart in order to build the feature in HFSS. In this section, each function will be explained in detail, clarifying how they all work together to accomplish the end result. Most of these functions are taken from the publicly available GitHub HFSS API[12].

hfssOverall

This function serves as a framework for the rest of the steps in the process. HfssOverall can be called from within MOLACO, before testCost is called. It requires the input of 'tour' from MOLACO, which is the path through the search space. For this project, the search space was limited to 3x3x3, as shown in the figure below.



1	2	3
4	5	6
7	8	9

Layer One

10	11	12
13	14	15
16	17	18

Layer Two

19	20	21
22	23	24
25	26	27

Layer Three

Figure 4: Space indices for a 3x3x3 search space, shown in three layers

On the first layer, each voxel within the search space is numbered from one to nine, labeling from left to right one row at a time. The numbering continues on the second, and then third layers. When MOLACO forms a path, it moves from one adjacent voxel to the next. For example, at square 14, the path could move to 5, 11, 13, 15, 17, or 23. This continues at each decision point until the path is complete, utilizing the probabilities discussed above for determining each movement.

Within hfssOverall, once the path has been input, the feature can begin to be modeled in HFSS. This is performed within the eleven functions called from hfssOverall. Some of them have outputs that feed back into hfssOverall, but many simply perform a task within HFSS and finish allowing the program to move on to the next function.

3.1 Initialization

The first section of hfssOverall is initialization, in which the VisualBasic file and HFSS file are established, as well as laying the groundwork for the modeling to begin.

hfssNewProject

This function creates the project within HFSS, which will be opened in each of the following functions. It requires the input of a file name, as well as the path location of HFSS on the computer. Duplicate naming conventions are not a concern, as all associated files are deleted at the end of the process, excluding data analysis files.

hfssInsertDesign

This function allows the creation of a design within HFSS, where the path will be modeled. It requires the input of the file name and the design name, any string to label the incoming design. Further modeling functions will work within the design created in this function.

3.2 Modeling

Modeling is perhaps the most important step in this process, and certainly the one which received the bulk of the focus of this process. Modeling is contained within just one function, however that function calls two other functions as well. This combination allows the path to be modeled as a series of connected cubes within HFSS.

hfssPathBuilder

This is the function that actually builds the given path, block by block. It calls another function, buildBlock, within it, as shown in the below figure.

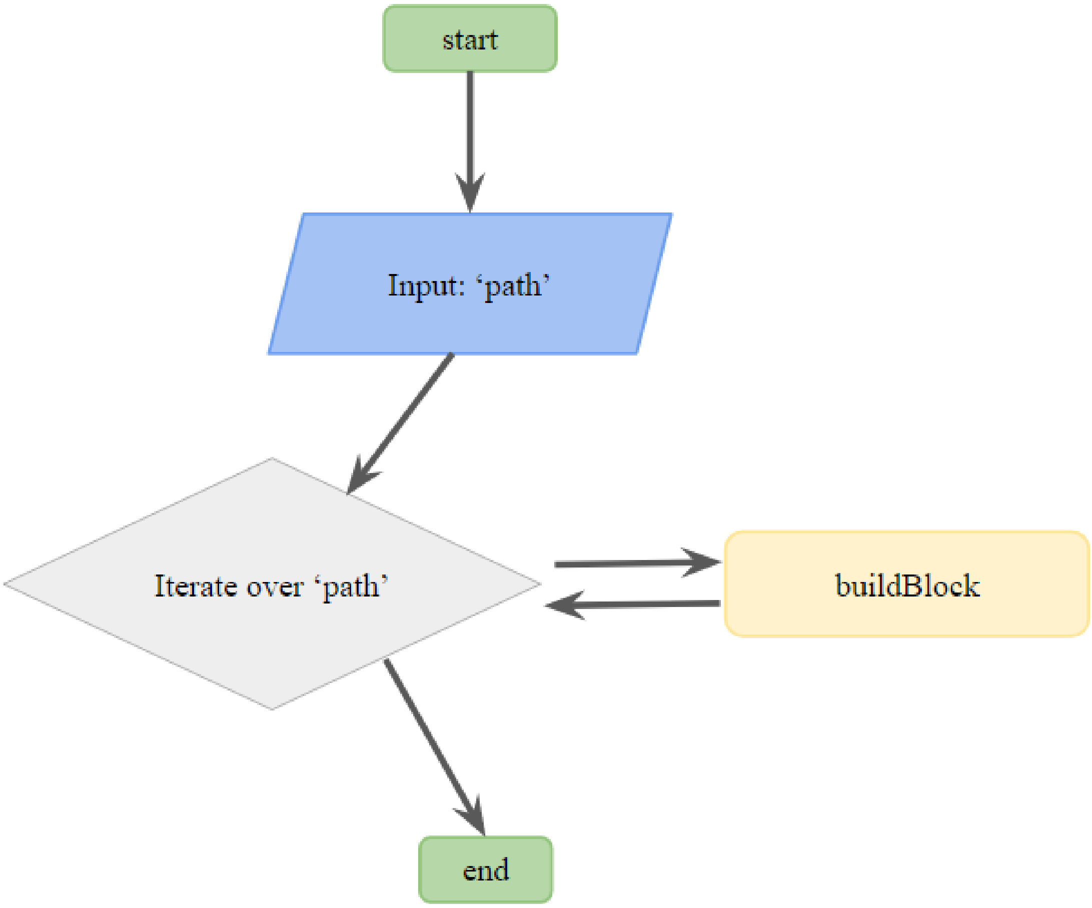


Figure 5: The flow chart for hfssPathBuilder. Not shown is hfssAssignMaterial, which takes place within buildBlock.

Within this function, the dictionary ‘antPath’ is created to match each square in the 3x3x3 search space to an x,y,z coordinate. This dictionary limits the search space to 3x3x3, but the simple creation of a dictionary for a different search space would allow a different space to be chosen. This function takes in the ‘path’ array as input and uses a for loop to iterate over it. In each iteration, it calls buildBlock in the location dictated by the search space dictionary. This creates a cube within HFSS for each iteration, with adjacent cubes from successive iterations automatically joining together. After this function has run, the path has been modeled within HFSS.

buildBlock

This nested function creates the VisualBasic code which instructs HFSS to build the blocks. It requires the file name, a name for the box (set as the index of the for loop to avoid repetition), the starting location of the cube (in x,y,z), the size of the cube (in size of x, size of y, size of z), and the units of the cube’s size (such as ‘cm’ or ‘mm’). It also calls hfssAssignMaterial, which specifies the material of the blocks being built, in this case silver. This function runs once for each voxel in the ‘path’ variable passed in from MOLACO.

3.3 Analysis

Analysis allows information to be gathered from the feature that was modeled in HFSS. It is not as simple as simply running a test, there are several functions that must be called to set up the file for this process. This section of the program has been logically implemented and included in the idealistic version of hfssOverall, but has not been thoroughly tested, as more resources were directed towards solving the issue of modeling. This sub-chapter, however, will provide a general overview of each function and how it would work were it fully implemented.

hfssInsertSolution

This function writes the VirtualBasic code to insert a solution setup into the HFSS project. It requires an input of the file name, the name of the solution setup, and the desired solution frequency (in GHz). Further analysis functions build upon this solution setup.

hfssInterpolatingSweep

This function adds an interpolating sweep to the solution set up in hfssInsertSolution. It requires the file name, a name for the sweep, the solution name, the starting frequency (in GHz), and the ending frequency (also in GHz) as inputs. Along with hfssInsertSolution, this sets up the process for solving the feature.

hfssSaveProject

A project in HFSS cannot be solved until it has been saved. Therefore, this function must be called at this point in the process, just before the feature is solved. It takes in the file name and project name, then saves it.

hfssSolveSetup

This function solves the setup created in several of the previous analysis functions. It takes in the file name and the solution name, utilizing the parameters already established by the previous functions' inputs.

hfssExportNetworkData

This function exports the data matrix of the variables S, Z, and f into a data file which can be passed on to other functions. It requires the inputs of the file name, desired name of the output file, the solution name, and the sweep name. Unfortunately, this function formats this data file as a .tab file, which is not compatible with the AIA.

tabToCsv

In order to convert the data file into the desired format within MATLAB, this function was developed to convert the .tab file to CSV. It simply takes in the name of the data file and renames it to ‘test_data.csv’.

3.4 Conclusion

hfssExecuteScript

Up until this point, HFSS has not actually been opened (even in the background). Rather, the previous functions were all writing into a VisualBasic document which is implemented in this step. This function executes that VisualBasic script, opening HFSS and performing all of the actions detailed in the script; Initialization, Modeling and Analysis.

hfssRemovePaths

This function cleans up files and directories created during this process, so that it can repeat for the many tested features without worrying about duplicate naming problems.

3.5 Case Study

Consider that 'path' is equal to [1 2 11 14 17 26 27], as shown in the figure below.

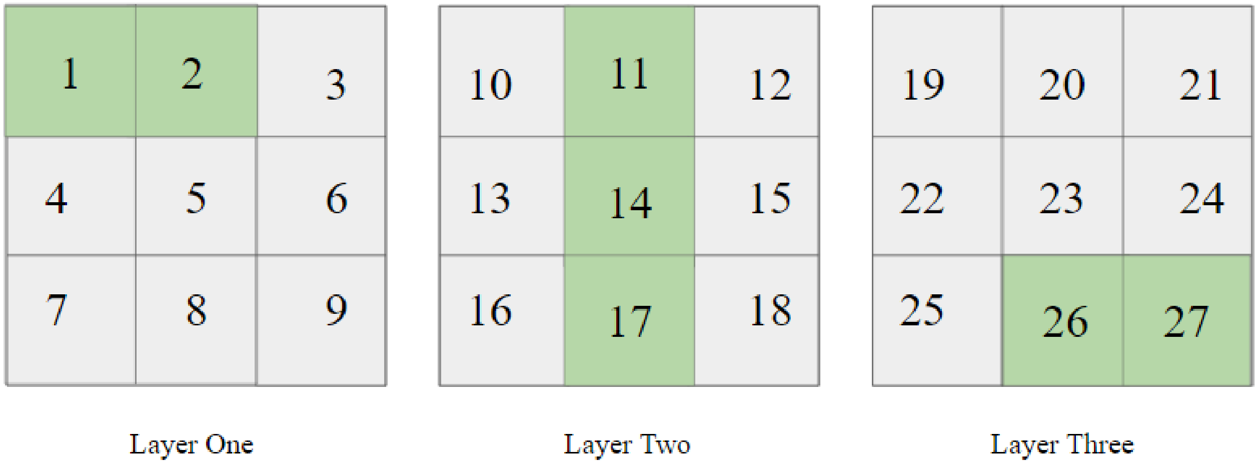


Figure 6: Path shown in green in a 3x3x3 search space, starting at 1 and ending at 27.

This path was chosen as it allows us to confirm that the program can build blocks on all three layers (e.g. 1, 11, 26), in all three rows (e.g. 2, 14, 27), and in all three columns (e.g. 1 17, 27). Since this path is highlighting the operation of the 3x3x3 matrix, the VisualBasic script was arbitrarily named 'three.vbs'. The Visual Basic script (referred to as fid within the code) may be set as any string, but continuity is important. If the fid is changed in hfssOverall, it must be changed to the same variable within hfss_path_builder as well. When ‘three.vbs’ is run, it opens HFSS, which models the path one block at a time. The fully modeled path is shown in the figure below.

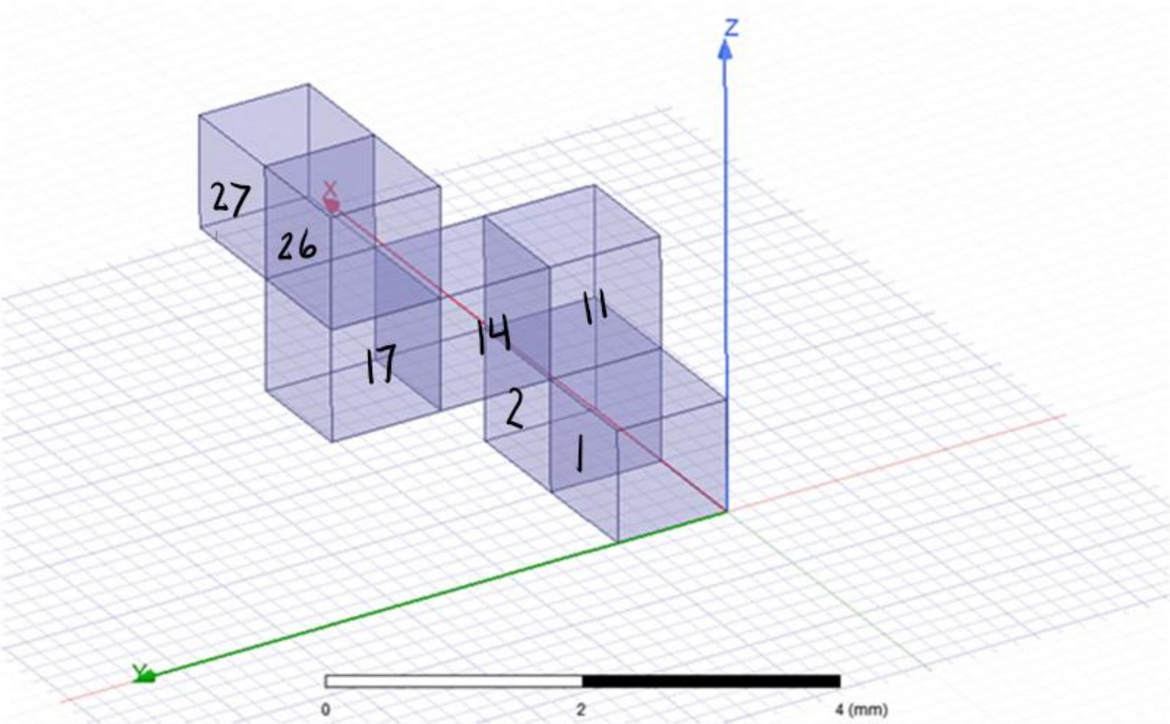


Figure 7: How the path appears when built in HFSS from MATLAB. Each block is labeled as its path identity for clarity

As we can see from Figure 5, the path has modeled correctly within HFSS. In this case each block has dimensions of 1x1x1 mm, but with simple changes to buildBlock, both the side length and units can be easily changed (e.g. to 0.1x0.1x0.1 nm). A 1 mm cube was simply chosen for clarity within the example.

Chapter 4

Conclusions and Future Work

Overall, the HFSS API is an effective way to model features in HFSS without leaving MATLAB. The process happens quickly, meaning that it should not significantly increase the computational expense. This is a favorable result, considering that computation times can already span many hours, and it would not be ideal to add significant complication onto that.

This set of functions was developed specifically for a 3x3x3 search space, but it is simple to broaden these options. A different dictionary can be made within hfssPathBuilder, allowing the user to select the desired search space. Smaller search spaces (e.g. 2x2x1) will take less time to compute, whereas a larger search space (e.g. 5x5x5) would require significantly more computation time but may yield more potential features.

Although this project planned out the usage of the HFSS API in combination with the AIA, it was only tested through the Modeling stage. Future work will include implementing the Analysis and Conclusion stages, and then passing the data gained into the AIA. Once that is confirmed to work, this code can be added directly into MOLACO to fulfill the goal of adapting MOLACO for metamaterials.

Future work may also include improvements to MOLACO itself – such as reduced computational cost and broad angle ranges. For example, if MOLACO can be further streamlined, each run will take less time and more features can be explored. Additionally, it would be preferable if the features found displayed the desired properties at a broad range of interaction angles. Developing a method to optimize for the desired property such as refractive index as well as a broad angle range would improve the efficacy of the optimized features.

MOLACO offers an incredible opportunity to discover nontraditional features for three-dimensional metamaterial features. By utilizing the HFSS API, modeling and calculations of proposed features can take place without leaving MATLAB, allowing MOLACO to be utilized for metamaterials instead of Frequency Selective Surfaces. This program will allow the group to seek out negative or low-index materials, utilizing the algorithm already in use and proven to be effective, in the search space of their choosing.

Appendix A

MATLAB Code

This appendix contains publicly accessible links to the MATLAB Code I created for this project as well as the MATLAB files borrowed from the HFSS API. In hfssOverall, functions outside of the modeling portion are commented out, but the framework is established as in Figure 1.

hfssOverall:
https://drive.google.com/file/d/1nm8Xe8OprcNOoiF-8YbGV0qdxEGleFTR/view?usp=sharing

hfssNewProject:
https://drive.google.com/file/d/1jPaF4pAWTKdTibQNqlV90JUa_xpivK6l/view?usp=sharing

hfssInsertDesign:
https://drive.google.com/file/d/1_UPGebYC_4GBDWizceUF76kRsTAaX6at/view?usp=sharing

hfss_path_builder:
https://drive.google.com/file/d/1P14KBIwWShbFf9uH-uYtTmMiHGqyaf9Q/view?usp=sharing

buildBlock:
https://drive.google.com/file/d/1IWRELrQqtsk6fbW5nT-eTyFP32ope66w/view?usp=sharing

hfssAssignMaterial:
https://drive.google.com/file/d/1B67l1gfwbCx0Mia0HR0Eu1BOt_-wqenr/view?usp=sharing

hfssInsertSolution:
https://drive.google.com/file/d/1GqoqSfxKb0bzyAHomyNT03e-hO98wbeq/view?usp=sharing

hfssInterpolatingSweep:
https://drive.google.com/file/d/1FnSTqP_gsXdgJximmUjhEWkPSmL5azuk/view?usp=sharing

hfssSaveProject:
https://drive.google.com/file/d/1lzkXjvYa-5W_-zhjXC_MZLqqH00iOyLf/view?usp=sharing

hfssSolveSetup:
https://drive.google.com/file/d/1TXoTpqUXd-BnrDKviv4epT7USN8TbmJi/view?usp=sharing

hfssExportNetworkData:
https://drive.google.com/file/d/1Hxvp_s6_Ch07s2pgxW3K0Kbo5BwL6_OV/view?usp=sharing

tabToCsv:
https://drive.google.com/file/d/1men_Et2juX1TLtsowaj47pKwWTfuz92P/view?usp=sharing

hfssExecuteScript:
https://drive.google.com/file/d/1db5QNP8lFOPpjj-OZ08DIulphuSGqhns/view?usp=sharing

hfssRemovePaths:
https://drive.google.com/file/d/12dZHmlc6XjzB5DIn-r50xhoedwBiJ-kL/view?usp=sharing

BIBLIOGRAPHY

1. Zheludev, N. I., & Kivshar, Y. S. (2012). From metamaterials to metadevices. *Nature Materials*, *11*(11), 917–924. https://doi.org/10.1038/nmat3431
2. Moitra, P., Yang, Y., Anderson, Z., Kravchenko, I. I., Briggs, D. P., & Valentine, J. (2013). Realization of an all-dielectric zero-index optical metamaterial. *Nature Photonics*, *7*(10), 791–795. https://doi.org/10.1038/nphoton.2013.214
3. Shalaev, V. M. (2007). Optical negative-index metamaterials. *Nature Photonics*, *1*(1), 41–48. https://doi.org/10.1038/nphoton.2006.49
4. Jiang, Z. H., Bossard, J. A., Wang, X., & Werner, D. H. (2011). Synthesizing metamaterials with angularly independent effective medium properties based on an anisotropic parameter retrieval technique coupled with a genetic algorithm. Journal of Applied Physics, 109(1), 013515. https://doi.org/10.1063/1.3530849
5. Zheludev, N. I. (2010). The Road Ahead for Metamaterials. *Science*, *328*(5978), 582–583. https://doi.org/10.1126/science.1186756
6. Connex3 Objet260 3D Printer for Multi-color Multi-material Models. (2021). Retrieved from https://www.stratasys.com/3d-printers/objet260-connex3
7. Zhu, D.Z. (2018). *Inverse Design of Three-Dimensional Frequency Selective Structures and Metamaterials using Multi-Objective Lazy Ant Colony Optimization.* (Unpublished doctoral dissertation). The Pennsylvania State University, University Park, PA.
8. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *26*(1), 29–41. https://doi.org/10.1109/3477.484436

9. Stützle, T., Hoos, H. H.. (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, 16, 889-914

10. Zhu, D. Z., Werner, P. L., & Werner, D. H. (2017). Design and Optimization of 3-D Frequency-Selective Surfaces Based on a Multiobjective Lazy Ant Colony Optimization Algorithm. IEEE Transactions on Antennas and Propagation, 65(12), 7137–7149. https://doi.org/10.1109/tap.2017.2766660

11. Mueller, C. (2014). Multi-Objective Optimization of Software Architectures Using Ant Colony Optimization. *Lecture Notes on Software Engineering*, *2*(4), 371–374. https://doi.org/10.7763/lnse.2014.v2.152.

12. Ramasami, V. (2020, December 9). yuip/hfss-api. Retrieved from https://github.com/yuip/hfss-api

ACADEMIC VITA

Natalie Cummings
nvc5249@psu.edu

EDUCATION

The Pennsylvania State University, State College, PA
Expected Graduation: May 2021
Bachelor of Science, Materials Science and Engineering, Minor in Spanish

HONORS/AWARDS

Student in the Schreyer Honors College **2017-2021**

Millennium Scholars Program, Cohort 5 **2017-2021**

- Scholarship program which seeks to increase diversity in students studying STEM fields

Presidential Leadership Academy **2018-2021**

- Selective program which guides students to become better leaders through seminars, debates, group projects, and many guest speakers surrounding the practice of leadership, with a focus on exploring the grey area of complicated subjects

EMSAGE Laureate **2020**

- Earth and Mineral Sciences Academy for Global Experiences: The College of EMS's recognition for high-achieving seniors, requiring demonstrated accomplishments in academic excellence, global awareness, professional and civic service, and leadership

EMS Undergraduate Poster Competition, 3rd place **2019**

SPARK Program Participant **2018**

- Learned about the many opportunities available with the help of the University Fellowship Office

WISER (Women in Science and Engineering Research) Participant **2017-2018**

- PA NASA Space Grant Consortium funded research in Dr. Beese's Lab

Phi Kappa Phi Undergraduate Research Potential Award **2017**

RESEARCH

Dr. Werner **2019-present**

- Undergraduate Researcher at Penn State
- Devised methods to transform our MATLAB ant colony optimization algorithm for Frequency Selective Surfaces to an algorithm for metamaterials, adapting a program from VisualBasic and producing new code

Dr. Shvets **2018**

- CCMR REU Participant at Cornell University
- Developed a novel transfer technique to place plasmonic metasurfaces onto the tip of an optical fiber

Dr. Marden **2017-2018**

- BIO 499: Tropical Field Ecology
- Performed biological field research in Costa Rica, focusing heavily on statistical analysis and methods of scientific communication

Dr. Beese **2017-2019**

- Undergraduate Researcher at Penn State
- Utilized basic metallography techniques such as grinding, polishing, tensile testing, and optical microscopy to analyze additively manufactured aluminum samples

EMPLOYMENT

Penn State Information Technology Services **2019-2020**

- Student Supervisor
- Enabled other employees to offer the highest quality of support
- Managed attendance, inventory, and other high-level responsibilities
- Utilized my computer background and logic skills to resolve technology issues as they emerged

Introspective Networks **2020**

- Software Engineering Intern
- Scripted programs in Python, bash, and PowerShell, culminating in the creation of a command-line operating system for one of the company's products
- Researched and drafted technical blog posts and created new pages for the company's website

EXTRACURRICULAR INVOLVEMENT

Material Advantage **2017-2021**

- Professional Society for students in Materials Science and Engineering
- Participated in science outreach events, attended lectures from professors

Millennium Society THON **2018-2021**

- Vice President, Family Relations Chair
- Helped found the organization, to give Millennium Scholars a way to give back to the community
- Participated in various volunteer events, including science outreach
- Helped raise thousands of dollars for THON
- Culminated my THON experience as a virtual dancer in THON 2021

Discovery Space **2017-2018**

- Helped to explain basic scientific concepts to children visiting the museum

TOTEMS Mentor **2018**

- Helped first-year EMS students get prepared and acclimated for their first semester through a summer program at Lake Raystown

SKILLS

MATLAB, bash, Python, C++, Java
High proficiency in Spanish
Beginner in French