Analysis of Photonic Crystal Wave Containment

S. David Lively
Southern Methodist University

Periodic dielectric structures may be used to direct and contain electromagnetic waves. The ability of a crystal to affect wave behavior depends on the several factors, including the shape, scale and distribution of dielectrics as well as the relative wavelength of an incident wave. Simulation techniques such as FDTD may be used to model periodic dielectric structures and their response to different wavelengths in order test suitability of a given structure for a particular application. We present an analysis of three structures, their resonant frequencies, and software modifications to enable efficient simulation of such structures.

The Finite Difference Time Domain method is a popular, robust method to simulate and analyze the propagation of electromagnetic waves within different media and geometries. While typically applied to non-periodic structures such as dielectric slabs and complex circuits, the algorithm may also be used to simulate periodic structures.

For this document, the existing GoLightly simulator was modified to facilitate periodic structures. The goals of such modifications include:

1. Compatibility with modern hardware
2. Unit cell-based simulation definitions
3. Tiling of image-based crystal unit cells
4. Multi-resolution simulation
5. Source wavelength sweep
6. Interactive monitor charting and analysis
7. Summary data export

The most-recent GoLightly implementation, created in 2015, utilized a combination of C++, NVIDIA CUDA and OpenGL. While effective at the time it was created, several factors indicated that a reimplementation was required for this project. The available hardware – specifically, a 2019 16” MacBook pro with AMD GPU – presented several challenges.

CUDA (Compute Unified Device Architecture), the GPGPU development ecosystem developed by NVIDIA, requires access to an NVIDIA GPU. In addition, the CUDA version used by GoLightly is nearly a decade old. Updating the application to use a more modern CUDA version would have required substantial work which would not benefit the task at hand. In addition, CUDA kernels are based on the Cg shader language which is very similar to HLSL[1] used by Unity3D.

Given that available hardware consisted of a MacBook Pro with an AMD GPU, it seemed logical to port the CUDA kernels to HLSL. This is platform-agnostic system that may be cross-compiled for NVIDIA, AMD and Intel GPUs, and, through Unity3D, is available for all modern operating systems.

GoLightly’s user interface requires a complex CUDA/OpenGL interoperation mechanism. MacOS support for OpenGL has been deprecated by Apple. While it is still possible to use OpenGL on MacOS, implementation of the FDTD kernels in HLSL precludes use of OpenGL. While the FDTD kernels could have been re-written in GLSL (OpenGL Shader Language), the similarities between CUDA and HLSL made HLSL a more suitable platform.

Modern game development platforms such as Unity3D are well-suited for the creation of cross-platform GPU-centric applications with complex user interfaces. Several factors

[1] High Level Shader Language, used by DirectX and Unity3D (rebranded as Shader Lab).

informed the decision to reimplement GoLightly in Unity3D.

- Cross-compilation support of HLSL to Metal, the preferred graphics API on MacOS
- Rich library of user interface software components
- Rapid development iteration cycle.
- Support for MacOS, Linux and Windows

The decision to reimplement GoLightly in Unity3D yielded many benefits including reduced software complexity. The C++ GoLightly implementation contained a substantial amount of code purely to manage the user interface. Unity3D provides much of this functionality by default, leading to the exclusion of many thousands of lines of source code.

Once the Unity3D implementation reached feature parity with the C++ version, some modifications were required to facilitate support for periodic structures. The prior version relied on the scale-invariant nature of Maxwell's equations that describe the physical behavior of electromagnetic waves.

For instance, the FDTD algorithm defines the size of a Yee cell as 1/10th of the wavelength of the highest frequency source in the simulation. All physical constants such as **u0**, **eps0**, and **c** are defined as 1. While these assumptions simplify some calculations in FDTD, they present issues when dealing with structures defined in real physical units. Since analysis of the dielectric lattice requires a frequency sweep to identify resonant wavelengths, it is convenient to define the Yee grid in terms of feature size rather than wavelength[2].

To that end, we define the normal wavelength of the simulation in terms of the smallest feature size in the simulation. For the first structure to be analyzed[3], the smallest feature is a rod with diameter 480nm. We define the normal wavelength[4] of the simulation to be 1/10 of that value, or 48nm. The frequency scanning portion of the simulation may then use wavelengths as low as 10*48um= 480nm. (For each of the lattice simulations to be performed, the minimum wavelength of interest is on the order of 600nm.)

Having determined a rational way to convert physical units to the unitless parameter space used by GoLightly, a few other modifications required.

GoLightly uses an image-based model definition format to eliminate any dependence on complex CAD applications or scripting. While well-suited to represent complex, arbitrary dielectric geometries, investigation of the lattices of interest required a certain flexibility not easily adapted to the image-based model definition. Namely, it was required to repeatedly adjust the Yee grid resolution when exploring potential solutions. This required the creation of many versions of the lattice unit cell tiles.

Rather than performing this process manually, we wrote a Unity3D script to generate the unit cells which were then exported to the PNG model definition format used by GoLightly. At that point, we determined that generating a tile, exporting it to a PNG image, reloading and decoding that image and using it to populate the dielectric material array used by the FDTD kernels was, to be frank, redundant, inefficient and suboptimal. Since the routines to generate a tile had already been created, we chose to use it to directly populate the FDTD dielectric array. While this is a departure from GoLightly's design ethos[5], it simplified this use case.

---

[2] Choosing a minimum wavelength that is proportional the smallest feature of interest would also have been valid.

[3] A square lattice of rods with spacing of 1.2um, diameter 480um.

[4] FDTD assumes that the size of a Yee cell is at most 1/10th of the shortest wavelength in the simulation.

[5] If the application were meant to handle general simulation cases, an image-based tile format would be advantageous.

Once the model generation system was complete, a software component was created to re-run the simulation with different conditions in a loop. Specifically, the source wavelength was incremented in a loop with user-defined wavelength band and number of steps to be calculated. Results were collected by summing the RMS power of each location within a series of FDTD monitors over 4000 time steps[6,7].

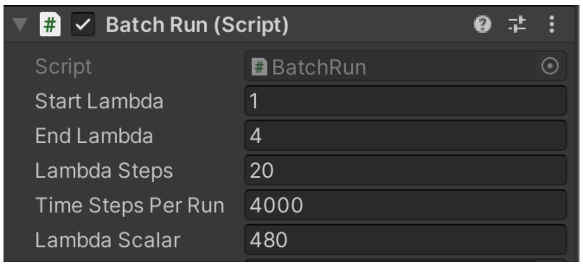

Figure 1 Frequency Sweep Parameter Dialog

The additional requirement of user-defined FDTD domain resolution[8] necessitated the implementation of a Model Provider component responsible for defining the domain size, Yee cell size, and other parameters as indicated in Figure 2.

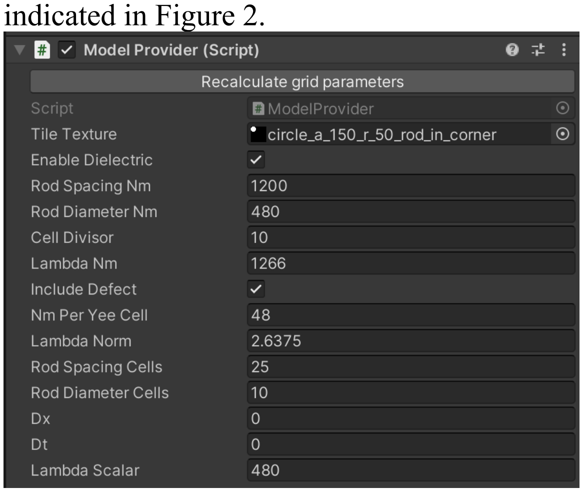

Figure 2 Model Definition Parameters

Figure 2 lists the model input parameters for the rod array lattice simulation. The "Lambda Norm" indicates the relative source wavelength, defined as the ratio of lambda (nm) to 10 * nm per cell. The Cell Divisor parameter indicates the minimum source wavelength in terms of Yee cells. For the rod array experiment, the "Include Defect" parameter indicates whether the rod removal defect should be included in the resulting dielectric array.

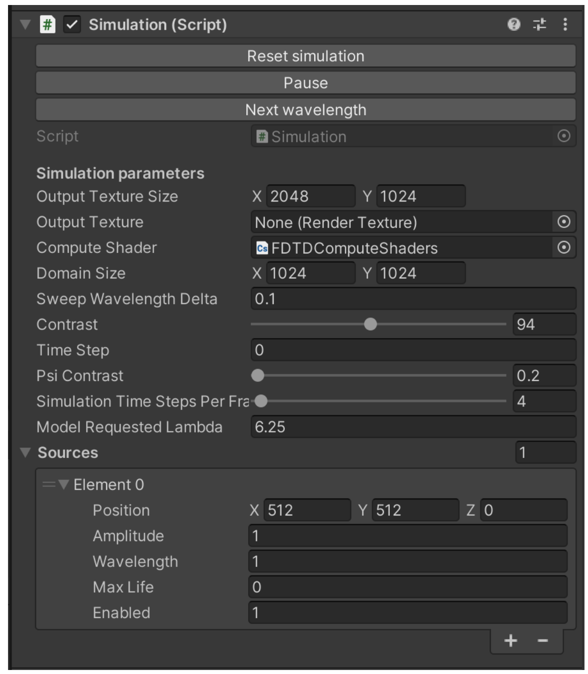

Figure 3 Simulation Configuration

The Simulation component (Figure 3) provides a configuration interface to control some interface items, indicating the size of the simulation domain in Yee cells, the variant of the FDTD compute shaders to be used, the current time step (updated while a simulation executes) and the number of FDTD frames (time steps) to be calculated between user interface updates. Generating

6 4000 steps was experimentally determined to be the approximate minimum number of frames required for the monitor values to reach steady state.

7 In FDTD, a time step is defined as the ration of c to the minimum source wavelength, where c is assumed to be 1.

8 The resolution of the domain in this case is nm per square Yee cell.

an image from a electric or magnetic field array requires substantial compute resources and is done infrequently to increase throughput and reduce execution time.

Experiments and Results

For this project, three crystal definitions were analyzed at various wavelengths in order to determine resonance. These are detailed below.

Experiment 1: A square lattice of rods in air with spacing a= 1.2um, radius r=0.24um and relative epsilon of 8.9.

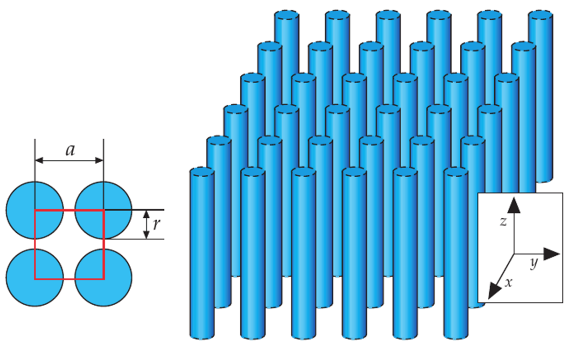

Figure 4 A square lattice of cylindrical rods. a=1.2um. r=0.24um.

A Yee cell size of 48nm[9] was chosen to minimize aliasing and ensure that features (rods) occupied enough Yee cells to accurately represent their interaction with the minimum wavelength defined as 480nm. Sweeping the input wavelength from 480nm to 1800nm in steps of 72nm yielded the results show in Figure 5.

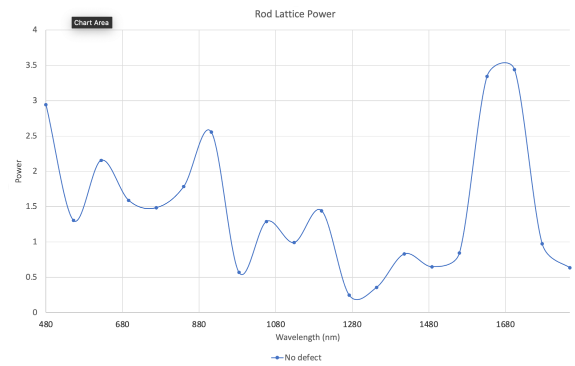

Figure 5 RMS output of a square lattice of rods

In this case, the area of interest is the minimum monitor sum indicating maximum confinement or resonance. Experimentally, this was determined to be a source wavelength of 1266nm[10].

Figure 6 shows the TMz E-field distribution.

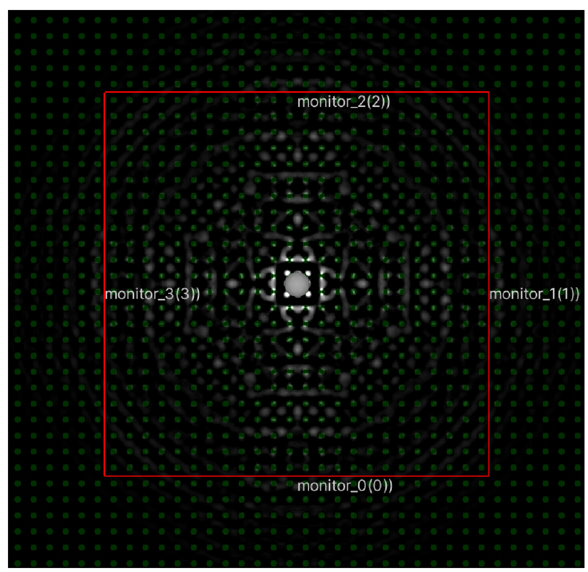

Figure 6 Ez field for square rod array

The bright area at the center of the field plot indicates a high-magnitude Ez value. The wave remains confined within the array.

---

[9] The Yee cell size is 10% of the rod diameter, or 48nm.

[10] Given the wavelength sweep resolution of 72nm, this value could be 1266 +- 36nm.

For comparison, the same experiment was run with a missing rod adjacent to the source position. The lattice is indicated in Figure 7.

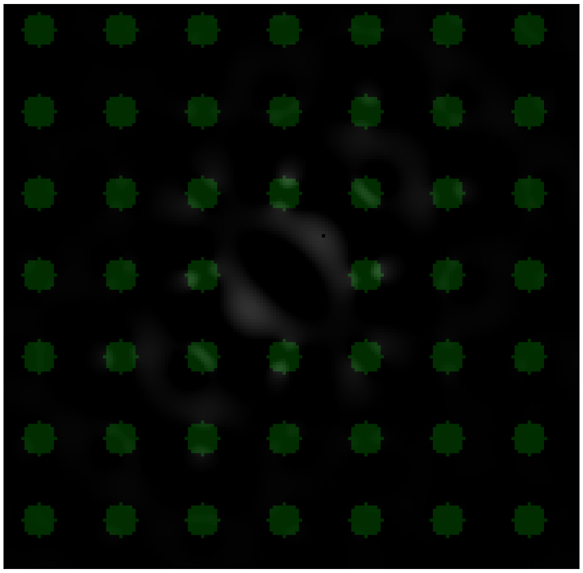

Figure 7 Rod lattice with missing rod defect

Note the missing rod at the center of the field. The same lattice was tested against the resonant wavelength of 1266nm. The resulting Ez field is portrayed in Figure 8.

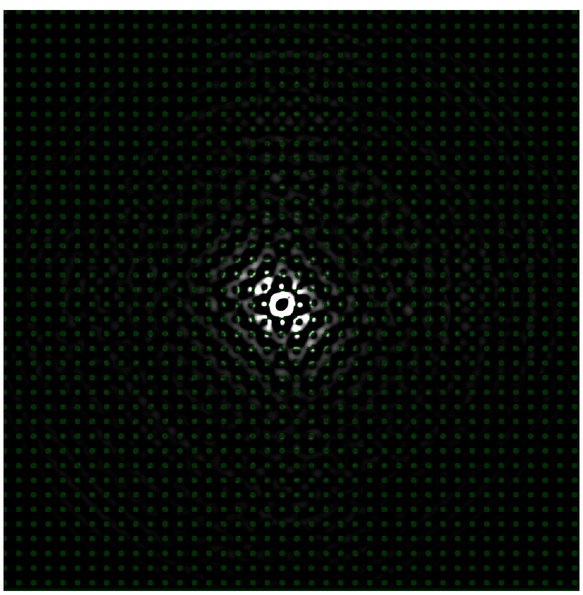

Figure 8 TMz Ez field - Square lattice after 4000 frames

Note the field asymmetry. This is due to the source location being slightly offset relative to the missing rod. This defect creates a resonant cavity which allows the wave to bounce within the empty space. The crystal itself still contains the wave just as in the version without the defect, but allows for some interesting things to happen in the interior.

The results of the perfect array and array with defect are shown in Figure 9.

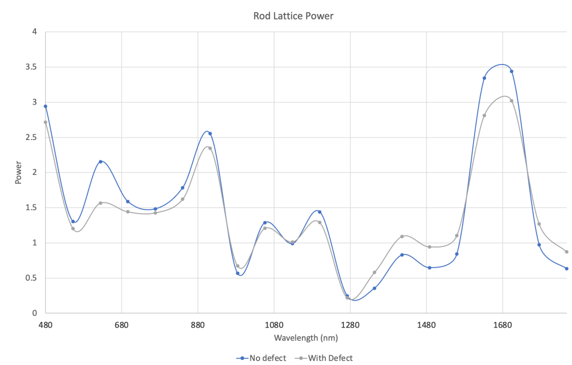

Figure 9 Rod lattice power output - defect vs non-defect

Note that both versions have approximately equal resonant wavelengths, the most interesting of which occurs at approximately 1266nm. Differences in the peak power at different wavelengths may be accounted for by the directionality of the source waves. This is due to the offset source position relative to the rod lattice. In the defect-free experiment, the source was perfectly centered between a square of four rods. In the defect experiment, a single rod was removed, leaving the source slightly off-centered relative to the lattice.

These experiments indicate that the simulator correctly models wave confinement and may be used to identify resonant values given a lattice definition and a range of wavelengths.

For the final experiment, a square lattice of veins was analyzed. A square lattice of veins where a = 1.2μm, r=0.2a, and epsr = 8.9.

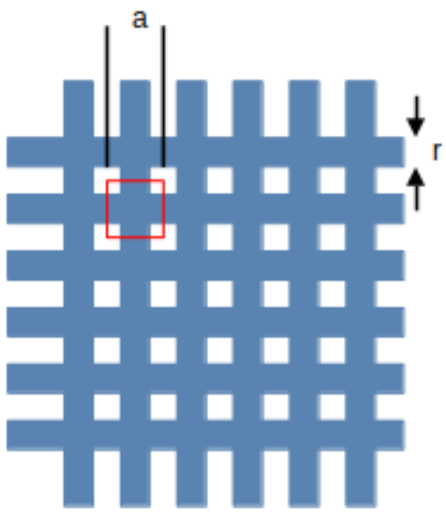

Figure 10 Square lattice of veins. a=1.2um, r=0.24um

This required modification of the Model Provider script to generate a unit cell consisting of a waffle-like pattern as indicated by the red square in Figure 10.

A wavelength sweep from relative lambda of 48nm to 1800nm with a step resolution of 48nm[11]

References

[1]Pollock, C. R., and Michal Lipson. *Integrated Photonics* . Boston: Kluwer Academic , 2003. Print.

---

[11] Experimentation indicats that the square lattice contains smaller wavelengths