Politechnika Łódzka
Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki
Instytut/Katedra Centrum Kształcenia Międzynarodowego

PRACA DYPLOMOWA INŻYNIERSKA

Design and Simulation of Periodic Ring Resonator Structures

Emmanuel Odongo
181341

Opiekun pracy
dr hab. inż. Sławomir Hausman

Łódź, luty, 2020

Politechnika Łódzka
Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki
Instytut/Katedra Centrum Kształcenia Międzynarodowego

PRACA DYPLOMOWA INŻYNIERSKA

Projektowanie i symulacja periodycznych struktur rezonatorów pierścieniowych

Emmanuel Odongo
181341

Opiekun pracy
dr hab. inż. Sławomir Hausman

Łódź, luty, 2020

Contents

1 Introduction

1.1 Problem statement

There are several ways in which the geometries of periodic ring resonator structures can be changed in order to alter their electromagnetic characteristics. Simulation software such as XFdtd* may be used to narrow down the iterations of a particular design prior to manufacturing physical prototypes. While XFdtd has a GUI that enables a user to manually model and tweak the parameters of the aforementioned structures, changing particular aspects of their geometries could prove to be a non-trivial task.

The aim of this thesis is to provide a method for rapidly prototyping periodic ring resonator structures.

To achieve a better prototyping experience, a different interface to the one currently provided in XFdtd, needed to be designed. This interface was conceptualised with the following goals in mind:

- Restricting certain geometric transformations that would fundamentally change the resonator structure.
- Making common geometric transformations simple to perform.
- Keeping the cognitive overhead to a bare minimum.

1.2 Motivation

Periodic ring resonator structures are widely used in AMC-backed antennas. The ring resonator structure is typically considered to be a two-dimensional structure due to its relatively small height. A conductive ground plate is positioned parallel to the resonator. In between the resonator and ground plate sits a substrate that accounts for most of the bulk of the AMC.

The effects that altering various aspects of the resonator geometry may have on the performance of an antenna backed by an AMC, are not always immediately obvious. These effects could vary widely depending on the particular geometries in question.

* XFdtd is an electromagnetic simulation software developed by Remcom.

AMCs have a number of unique properties that make them quite suitable at tackling various challenges in antenna design and operation. For instance, AMCs tend to have a very high surface impedance, so much so, that the term AMC is often used interchangeably with the term High Impedance Surface (HIS).

This high impedance makes AMCs suitable for use as ground planes in vertical monopoles. As compared to using a flat sheet of metal as the ground plane, the AMC has significantly less multipath interference since it suppresses the surface waves generated in the ground plane [1].

AMCs also have the unique property of in-phase image currents. Taking the example of a horizontal wire antenna – a setup similar to the vertical monopole mentioned previously with the key difference being that the antenna is bent so that it lies parallel to the ground plate – an AMC ground plane exhibits less backward reflection and a smoother radiation pattern than a flat metal ground plane [1, 2].

The surface resonance of AMCs is heavily influenced by the coupling between its various elements [1], making antennas that utilise an AMC ground plane difficult to detune. This property makes AMCs an ideal material for constructing Radio Frequency Identification (RFID) tags that can be use to track metallic objects [3].

Furthermore, Jafargholi et al. demonstrate how AMCs can be used to improve radiation patterns, increase radiation efficiency and improve gain in both monopole and dipole antennas [4].

1.3 Scope

The scope of this thesis covers the design of AMC cells using a custom-built application, the simulation of the subsequent designs using the XFdtd simulation software and analyses of the simulation results.

The focus of this endeavour will be determining the effects of varying the geometric underpinnings of AMC cells on their electromagnetic properties.

1.4 Backrgound information

Perfect Electric Conductors (PECs) have a reflection coefficient of -1, which means that they reflect incident electromagnetic waves with a phase shift of

180 deg and an amplitude equal to the incident wave. In order to construct an antenna with the minimal amount of interference, the ground plate must be placed at a distance at least $\lambda/4$ from the resonator structure, where λ is the wavelength of the incident wave.

Perfect Magnetic Conductors (PECs) on the other hand, do not add a phase shift to incident waves. They have a reflection coefficient of +1. The consequence of this characteristic is that the ground plate may be placed at an arbitrarily small distance from the antenna without causing destructive interference between the incident and reflected waves.

PECs do not occur in nature. It is not possible to construct one. The closest approximation is an AMC.

Both PECs and AMCs make decent reflectors that can be used to make an antenna unidirectional. However, AMC-backed antennas can have a much lower profile than PEC-backed antennas, since the dielectric thickness is not constrained by the wavelength of the incident wave.

For a very narrow range of frequencies, AMC-backed antennas introduce a phase shift of $0°$. For a phase shift of $\pm 90°$ in the reflected signal, their is some constructive interference. This range is considered to be the bandwidth of the antenna. Outside of this range, destructive interference occurs [5].

Padilla et al describe 2 classes of AMCs: the first based on volume and the second based on surface distribution of inhomogeneities [6]. The scope of this thesis encompasses only the second class. More specifically, it will cover the effects of variations in the geometries of AMCs on their electromagnetic properties.

Metamaterials are artificial composites of materials such as metals and dielectrics. The metamaterials that are of most academic interest are those that exhibit electromagnetic properties that are not found in natural materials. The physical properties of metamaterials can be tailored for specific applications.

By creating unit cells whose scale is significantly smaller than the expected operating wavelength and composing said cells planarly or volumetrically, the resulting metamaterial composite is tuned to a particular wavelength and behaves like a homogenous material [7, 8]. In practice, the AMC behaves like a Frequency Selective Surface (FSS) that is backed by a PEC. Variations in dielectric loading, cell geometry and cell spacing determine the spectral response of the FSS, with inductive elements generally contributing to a high-pass/band-pass filter response and capacitive elements contributing to

alow-pass/band-stop filter response [9, 10, 11].

2 Methodology

To aid in the rapid prototyping of various ring resonator structures, a tool was needed that would provide a way to quickly change various parameters and apply the changes in the simulation software. I settled upon building an application with a GUI that could provide visual feedback of the parameters being changed.

Due to my familiarity with the language, I opted for a JavaScript-based solution. The end product is a web application that can run in a web browser*.

* The web application targets ECMAScript 5 compliant browsers, which includes most major browsers released after 2013 [12]

2.1 Structure of the application

2.1.1 GUI Layout

The layout of the web-based application's GUI, consists of three main parts:

- General configurator
- Parameter modifier
- Renderer

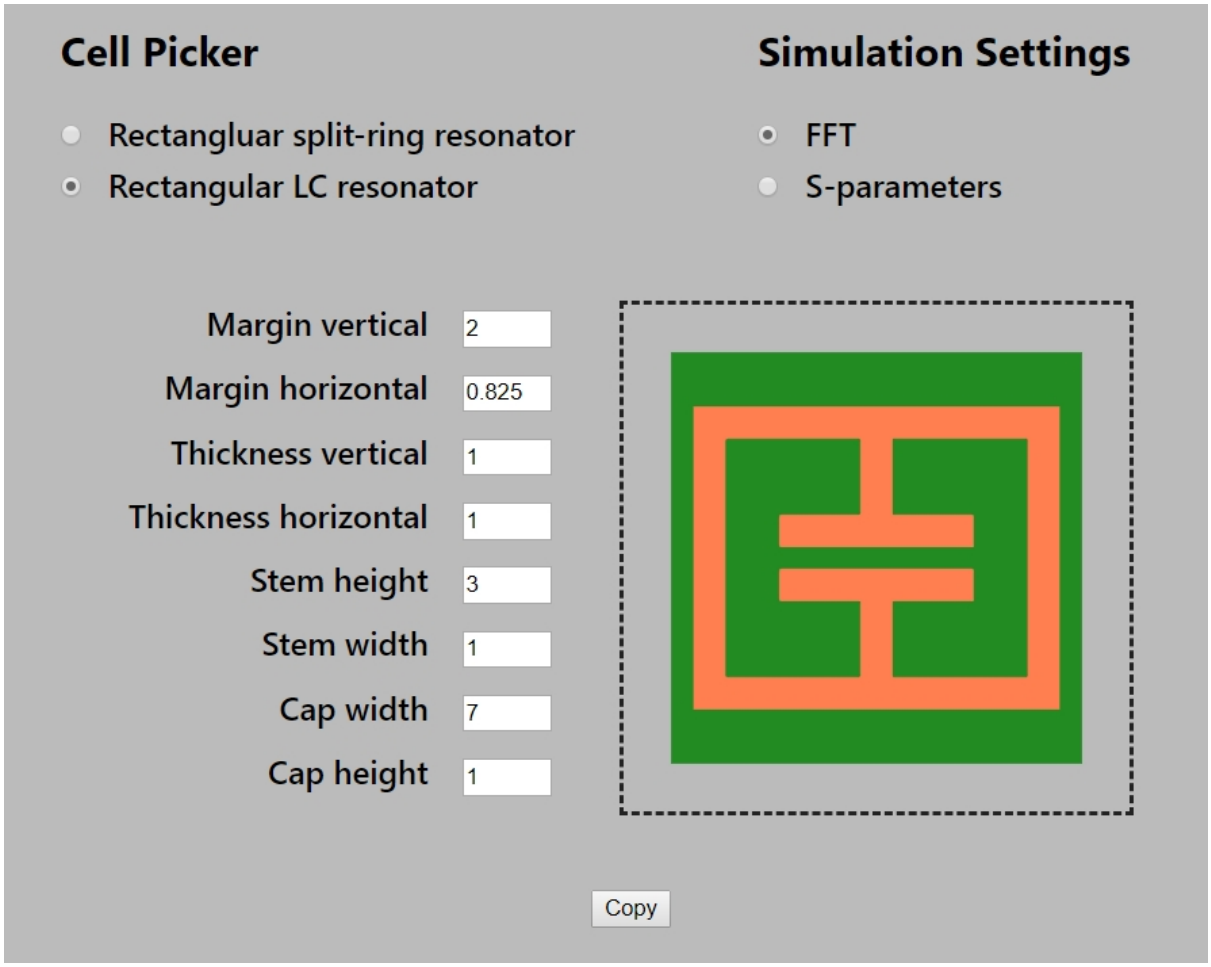


Figure 1: GUI of the application used to design AMC cells

General configurator

The general configurator is further subdivided into two sections:

- Shape selector
- Output selector

Cell picker

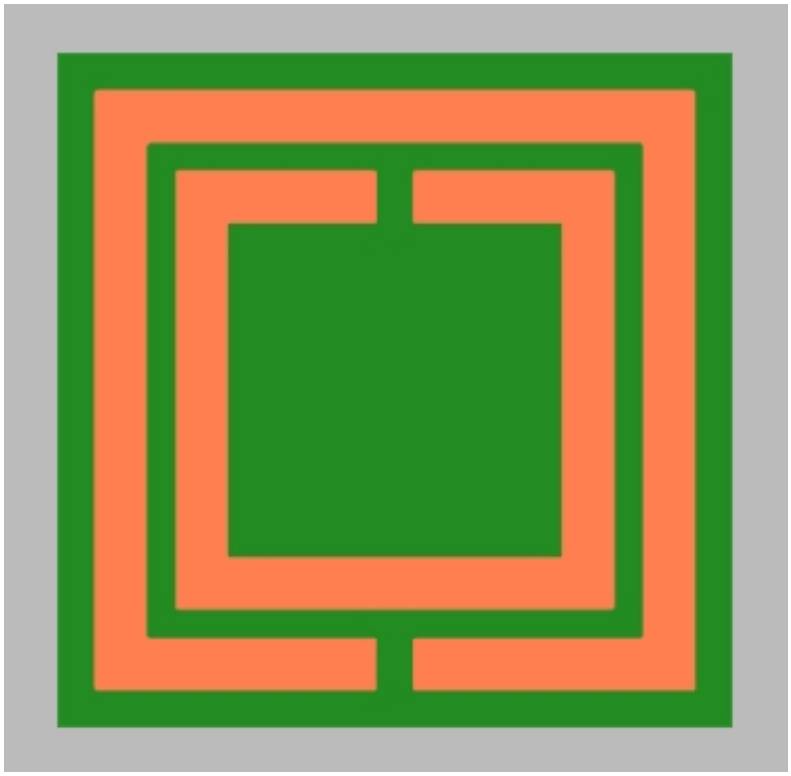
The cell picker allows for switching between the predefined AMC cells that are provided. These shapes need to be programmed individually. It consists of a list of radio buttons and is located in the top-left portion of the GUI as depicted in *Figure 1*

Figure 2 shows the visual representations of the shapes as they appear in the AMC cell design program, having their parameters set to their arbitrarily chosen default values.

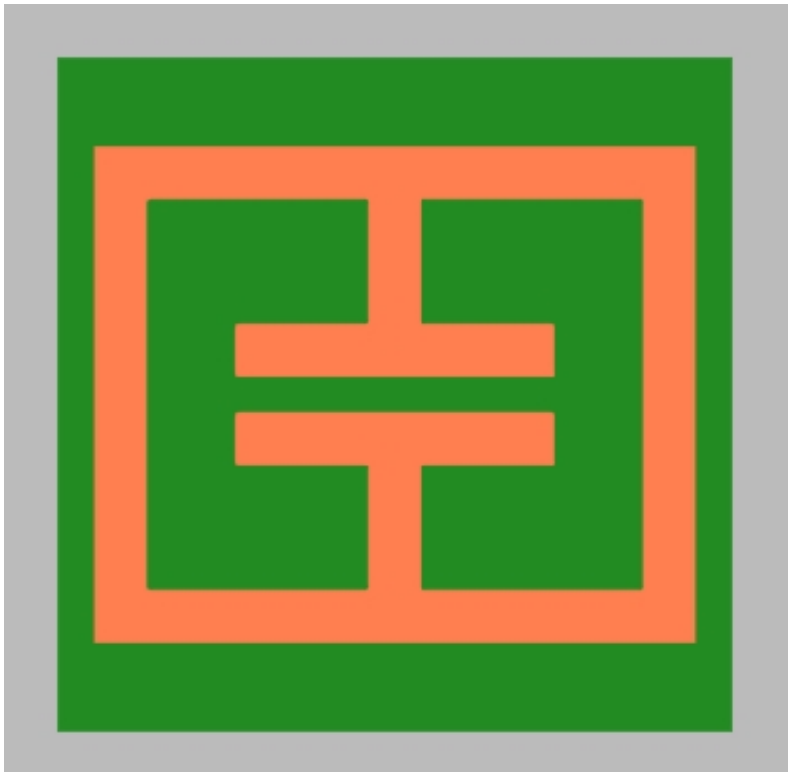
The shapes available in the shape selector are listed below:

- Rectangular-shaped split-ring resonator (*Figure 2a*)
- T-shaped resonator (*Figure 2b*)

Each shape is assigned to a radio button, meaning that only one shape can be selected at a time.



(a) Rectangular-shaped split-ring resonator



(b) T-shaped resonator

Figure 2: Shapes of AMC cells available in the shape selector

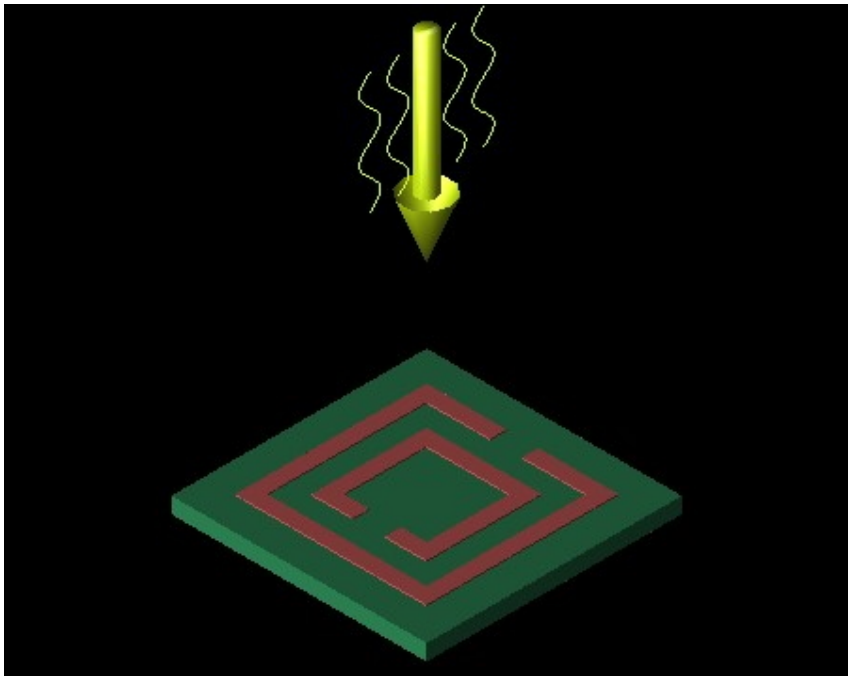
Within the application, shapes are represented as objects with three core parts:

- Parameters
- Vertex positions
- Constraints

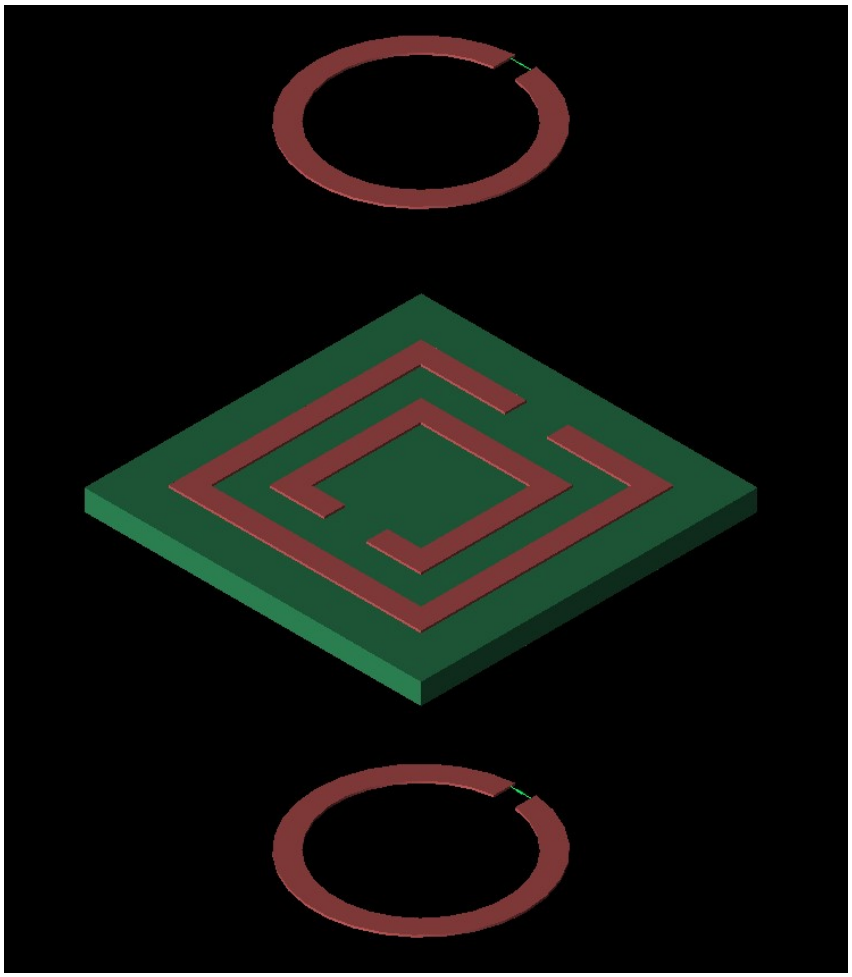
Parameters – These are the parametrised values used to describe the geometry of a cell. The parameters were selected in such a way as to maintain symmetry along the vertical and horizontal axes. This design decision was made since it greatly simplifies the logic needed to update the shape when parameter values are edited. The only common parameters between the various shapes, are the *horizontal* and *vertical outer margins*. Theses two parameters determine the shortest distance between the conductive part of the geometry and the cell boundary in the horizontal and vertical axes respectively.

Vertex positions – Each point/vertex of any shape in the application is defined in terms of the respective shape's parameters. In the AMC cell design program, shapes are rendered as SVGs. SVG is an image format for vector images.

Constraints – Each shape is bundled with a set of equations that are used to determine whether the conductive parts of the shape that give it its distinct geometry are not overlapping each other. The constraints also prevent the shape from growing too large and overflowing the maximum allowed size for a cell. The constraints are ran every time the AMC cell design program receives a request to update the parameters of said shape. If all the constraints pass, the parameters of the shape are updated and the coordinates of the vertices are subsequently recalculated. Within the GUI, the shape is rerendered to reflect the change in parameters.



(a) AMC cell with an icon indicating planar waves



(b) AMC cell with two antennas on either side

Figure 3: Renders of an AMC cell in the XFdtd software

Simulation settings

The simulation settings section allows the operator to select type of simulation to be performed. Under the hood this setting chooses the appropriate excitation source to be used when simulating the AMC cell in XFdtd. By selecting **FFT**, the excitation source is set to an external planar wave that is parallel to the plane of the AMC. The direction of the planar wave is indicated by the yellow arrow in the XFdtd software as shown in *Figure 3a*. The frequency of the planar wave is swept across some predefined range. A point sensor placed above the AMC cell measures the reflected signal.

By selecting **S-parameters**, two split ring antennas are placed on either side of the AMC cell. The antenna above acts as the transmitter, while the one below acts as the receiver. This setup, shown in *Figure 3b*, allows for the simulation and subsequent calculation of the AMC cell's S-parameters.

Parameter modifier

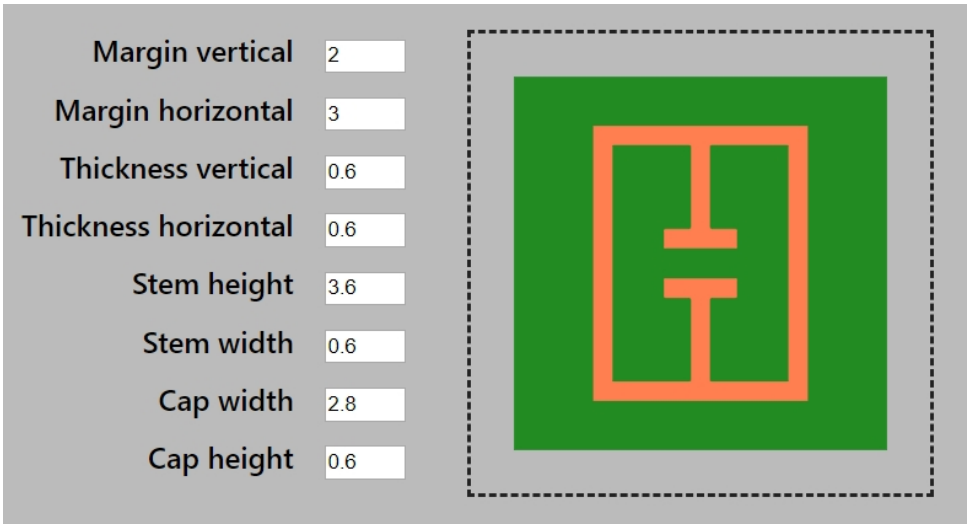
Each shape has a collection of parameters that can be altered. These parameters are listed on the left side of the interface, alongside their respective values. The available parameters depend on the geometry of the selected shape, thus not all shapes share the same parameters.

When a shape is selected, the list of parameters is updated to reflect the editable parameters of said shape. Switching between shapes resets all of its parameters to their initial values.

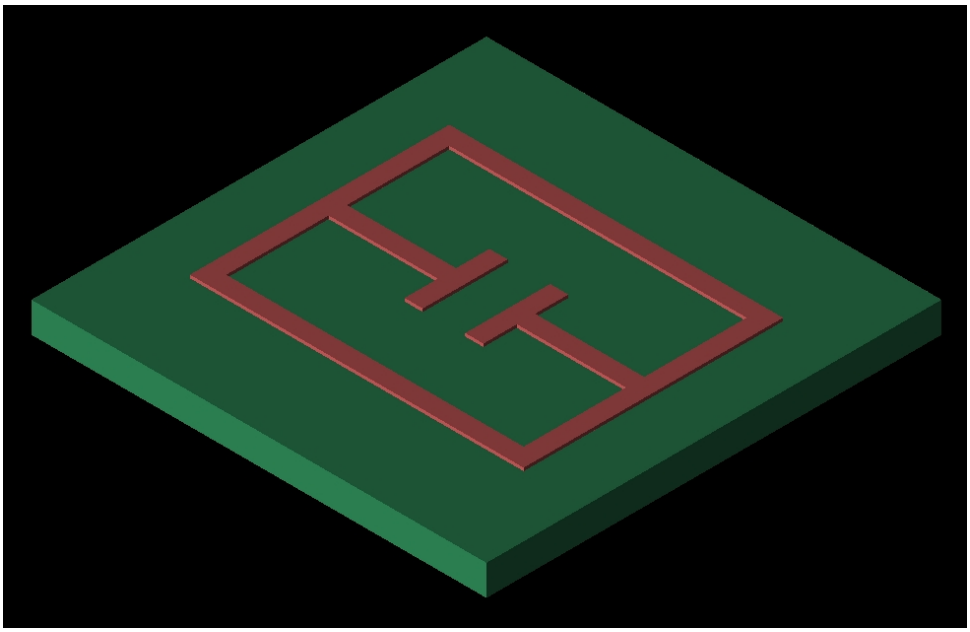
Renderer

On the right side of the interface is a live render of a single cell of the array. This render automatically updates when the shape is changed or when one of its parameters changes.

The render is 2-dimensional and is colour-coded, with orange representing the conductive material and green representing the substrate. *Figure 4* shows the same AMC cell as rendered in the design application as well as in the simulation software.



(a) Render of an AMC cell in the design application



(b) Render of an AMC cell in the XFdtd software

Figure 4: Renders of an AMC cell

Below the renderer is a button that copies the relevant code needed to generate the shown shape in XFdtd to the user's clipboard. The copied code can be pasted into a script window in the XFdtd software; upon running the script, a 3-dimensional shape is generated with the necessary materials, and a simulation is queued. Once the simulation completes, the operator may use the results to perform calculations and plot graphs.

2.1.2 Code Structure

The AMC cell design program is a web application written in JavaScript. The code is divided into 3 main blocks:

- presentational layer
- geometric definitions
- transpiler

Presentational layer

The presentational layer is responsible for rendering the GUI that is displayed to the user. It makes heavy use of a JavaScript templating engine called **lit-html**.

lit-html is part of the Polymer Project, an initiative by Google to create an open-source library for web development using Web Components. It provides a simple API for creating Hyper Text Markup Language (HTML) templates while retaining a syntax that is very close to HTML itself. It achieves this using JavaScript template strings (introduced in 2015 with the release of ECMAScript 6). Template strings provide a convenient way of performing string building by making use of interpolation. Prior to the release of this version of ECMAScript, the only way of building strings was through concatenation.

Template strings provide a concise way of expressing static and dynamic content alongside each other. In addition to this, template strings allow the building of multi-line strings without the need to explicitly use escape sequences such as new line (`\n`) and/or carriage return (`\r`).

The code snippets in *Listings 1* and *2* compare the two methods of string building.

```
var parameters = [
  {name: 'length', value: 10},
  {name: 'width', value: 8}
];

var concatenatedForm = '
  <form>' +
    '<label>' +
      parameters[0].name +
      '<input type="number" value="' + parameters[0].value = '">' +
    '</label>' +
    '<label>' +
      parameters[1].name +
      '<input type="number" value="' + parameters[1].value = '">' +
    '</label>' +
  '</form>
';
```

Listing 1: String building in ECMAScript 5 (concatenation)

```
var parameters = [
  {name: 'length', value: 10},
  {name: 'width', value: 8}
];

var interpolatedForm = `
  <form>
    <label>
      ${parameters[0].name}
      <input type="number" value="${parameters[0].value}">
    </label>
    <label>
      ${parameters[1].name}
      <input type="number" value="${parameters[1].value}">
    </label>
  </form>
`;
```

Listing 2: String building in ECMAScript 6 (interpolation)

The version in *Listing 2* uses string interpolation and is much more readable. The concatenated version is shown in *Listing 1* and requires the use of numerous quotation marks and + operators, which only serve to draw unnecessary attention away from the parts of the string that are contextually relevant.

Unnecessary line breaks could be removed from the concatenated version, resulting in fewer lines of code. However, increased line length is in itself, a factor that can make code less readable.

Geometric definitions

Due to the parametrised nature of the shapes within the AMC cell design program, the models used to represent them must be dynamic. No section of the geometries is static, hence the shapes must be represented as a set user-controlled parameters that can be combined in such a way as to sufficiently describe the overall desired geometry.

This can be, for the most part, accomplished by exposing parameters such as widths, heights and margins to the operator. It should be noted that this does not give the operator complete control over the entire geometry, since the way in which the parameters have been defined is restricted to vertically and horizontally symmetrical geometries.

Transpiler

The application is written in JavaScript, with the aim of providing a way to run it cross-platform (i.e. on different operating systems). By leveraging the ubiquity of web browsers, JavaScript applications can essentially run cross-platform.

There is one major caveat, though, which is that different browsers have varying degrees of support for all of the features of modern-day JavaScript. While this issue has largely been mitigated over the last few years by browser vendors converging on a set standard for JavaScript, not all systems have the latest versions of their web browsers installed.

Furthermore, since JavaScript is an ever-evolving language, new features are being constantly added, leaving older versions of browsers outdated.

The most widely adopted solution to the problem stated above is code transpilation. This allows developers to write their code in a modern Javascript (at the time of writing the latest iteration is ECMAScript 10) but have it work on older browsers that were released before this version of JavaScript. This is achieved by transpiling the code down to a previous version of JavaScript (typically ECMAScript 5 since this provides support to over 95% of users globally[12]).

2.2 Coding concepts

2.2.1 Data-binding

The bulk of the interactivity of the AMC cell design application makes use of a concept known as data binding. The data in question is the set of parameters that are used to define a given shape.

Each vertex making up a shape can be considered as being stored internally as a coordinate – implementation-wise, a coordinate is a pair of expressions made up of various parameters of the shape. The GUI renders editable input fields, initialising the values with the initial parameters of the shape.

The vertices are calculated and used to render the shape in the renderer in the bottom right of the GUI.

As the operator changes the value in a given input field, the corresponding parameter of the shape is changed internally. This triggers a recalculation of the vertices of the shape, causing the representation of the shape in the GUI to be rerendered.

2.2.2 Scalable Vector Graphics

Scalable Vector Graphic (SVG) is an Extensible Markup Language (XML) based image format for defining scalable vector images. Vector images differ from bitmap images in that they are infinitely scalable – information about a vector image is not lost when it is scaled up or down.

As a subset of XML, SVGs consist primarily of elements and attributes. Elements are namespaces that, in the context of the SVG layout engine, have an inherent meaning. Elements can accept attributes, which are key-value pairs. Attributes provide a way of passing arguments to elements. Elements are typically expressed using a pair of tags, meaning they have an opening tag and a closing tag. Certain elements can also have other elements nested within their tags. In the case of some elements, no closing tag is provided, and a special kind of tag called a self-closing tag is used (an example of this is the `<path>` element shown in *Listing 3*). Attributes can be applied to opening or self-closing tags.

Valid SVGs are wrapped in `<svg>` tags. In *Listings 3* and *4*, the `<svg>` attribute has 3 attributes. The `height` and `width` attributes define the dimensions in pixels that the SVG should be displayed at. The `viewbox`

attribute defines the visible area of the SVG. The first two numbers represent the top left corner of the visible area with respect to the SVG's internal coordinate system. The remaining two numbers represent the width and height of the viewable area respectively.

SVGs have a `<path>` element which has a `d` attribute. This attribute has several commands for drawing various shapes.

As previously stated, in the AMC cell design program, shapes are primarily represented as a set of vertices. The following are some of the simpler commands provided by the `d` attribute

- `M` – The **Move To** command moves the cursor to the provided coordinate (relative to the viewable area of the SVG). This coordinate is used as the starting point for drawing.
- `h` – This command draws a horizontal line whose length is equal to the provided value.
- `v` – This command draws a vertical line whose length is equal to the provided value.
- `Z` – The **Close Path** command draws a straight line between the first and last point of the `d` attribute.

Listing 3 shows how a square may be drawn using a `<path>` element.

The are more complex commands that can be used to draw arcs and bezier curves, however, they have been omitted from this overview since the AMC cell geometries under consideration consist of only straight lines. For this reason, instead of the `<path>` element, the `<polygon>` element – which has a simpler Application Programmable Interface (API) – will be used in the design application.

The `polygon` tag has a `points` attribute that takes as a value a series of points. Each successive point is connected to the preceding point using a straight line. The first and last points are also connected by a line automatically. *Listing 4* shows how a square may be drawn using a `polygon` element.

```
1 <svg width="300" height="300" viewBox="0 0 100 100">
2   <path d="M 10 10 h 80 v 80 h -80 Z" />
3 </svg>
```

Listing 3: SVG using the <path> attribute

```svg
<svg width="300" height="300" viewBox="0 0 100 100">
  <polygon points="10,10 90,10 90,90 10,90"></polygon>
</svg>
```

Listing 4: SVG using the <polygon> attribute

An SVG may contain multiple `<path>`s, `<polygon>`s and many other elements. These can be composed to form more complex shapes. Using Cascading Style Sheets (CSS), it is trivial to apply different styles to different elements in an SVG. In this way, it is possible to create a visual distinction between the conductive part of the AMC and the substrate as shown in *Figure 5*.

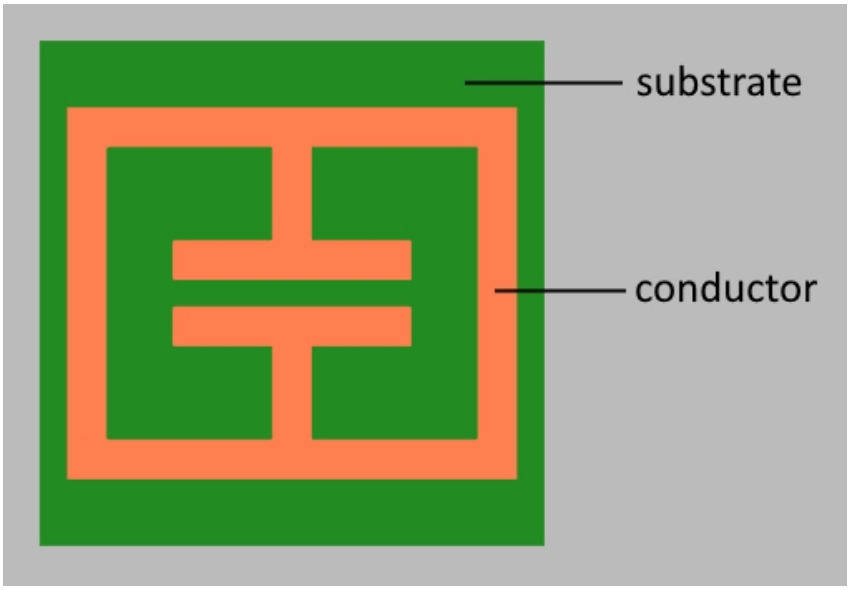


Figure 5: Colour-coded visual representation of an AMC cell

2.2.3 Exporting Structures to XFdtd

XFdtd supports a scripting language based on the ECMAScript 5 specification. JavaScript is a general purpose scripting language that also conforms to the ECMAScript specification.

The scripting language in XFdtd allows for the automation of tasks that would otherwise have to be performed manually using the GUI; it provides an API for various actions including but not limited to creating and saving projects, generating geometries, defining materials, running simulations and plotting graphs.

At the very bottom of the application GUI is a button that allows the user to

copy the generated script to their operating system's clipboard. This marks the completion of the design phase of the AMC structure.

The next step is to perform simulations using the XFdtd software. The operator begins this phase by opening said software. The aforementioned script that was copied to the system clipboard may then be pasted into a scripting window within the XFdtd software application. The operator runs the script by clicking the **Run Script** button. The relevant actionable items in the XFdtd software have been labelled in *Figure 6*.

To the right of the XFdtd GUI is a panel that contains shortcuts to open various windows within the software. The one of interest is the **Scripting** shortcut. Clicking it will open a scripting window*.

A new macro script can be created by clicking on the gear icon or using the menu (**Script > New Macro Script**). Doing so opens a new macro script tab within the scripting window. This tab has a text area into which XFdtd scripting code can be input. Below the text area is a green icon with the label **Execute Macro**. Clicking it will execute the script in the text area. The same effect can also be achieve from the menu (**Execute > Execute Macro**).

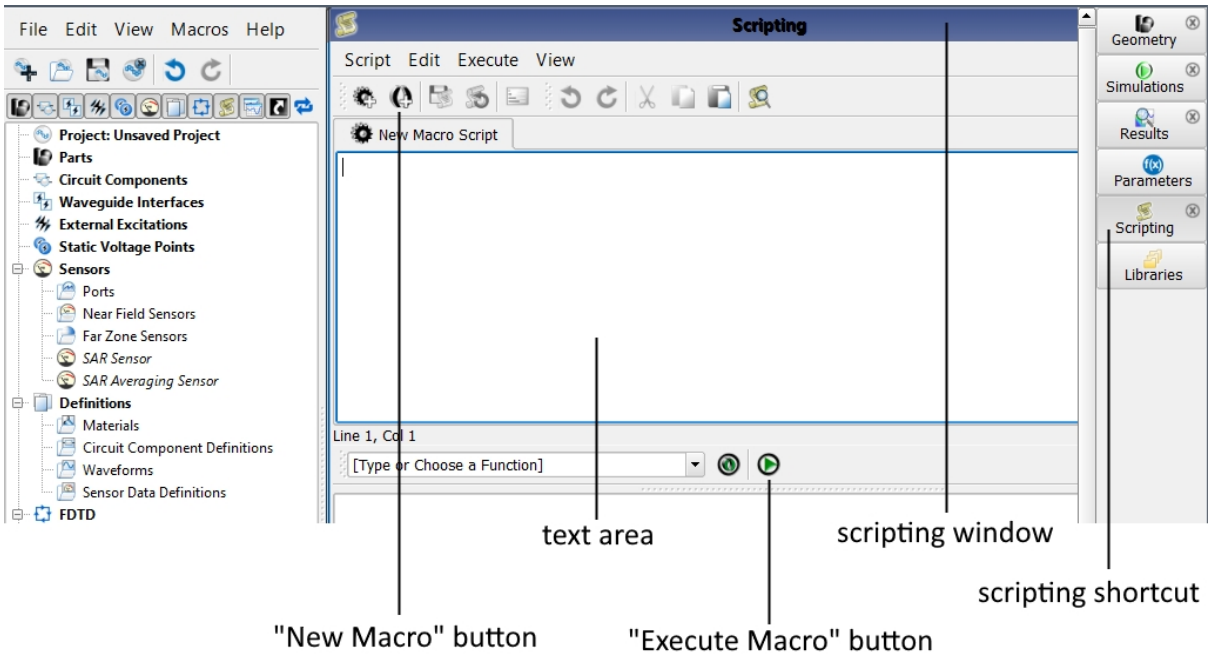


Figure 6: XFdtd GUI

* If the shortcut is not listed in the side panel, it can be toggled using the menu of the main XFdtd window (**View > Scripting**).

The script will then perform the following actions automatically:

- Set up the XFdtd project.
- Create the materials used in the model. In this case copper for conductive materials and FR4 for the AMC cell substrate.
- Create the geometry as defined in the design application.
- Set up the environment needed to perform the simulation e.g. antennas, sensors etc.
- Run the simulation as defined in the design application.

2.3 Resonator Geometries

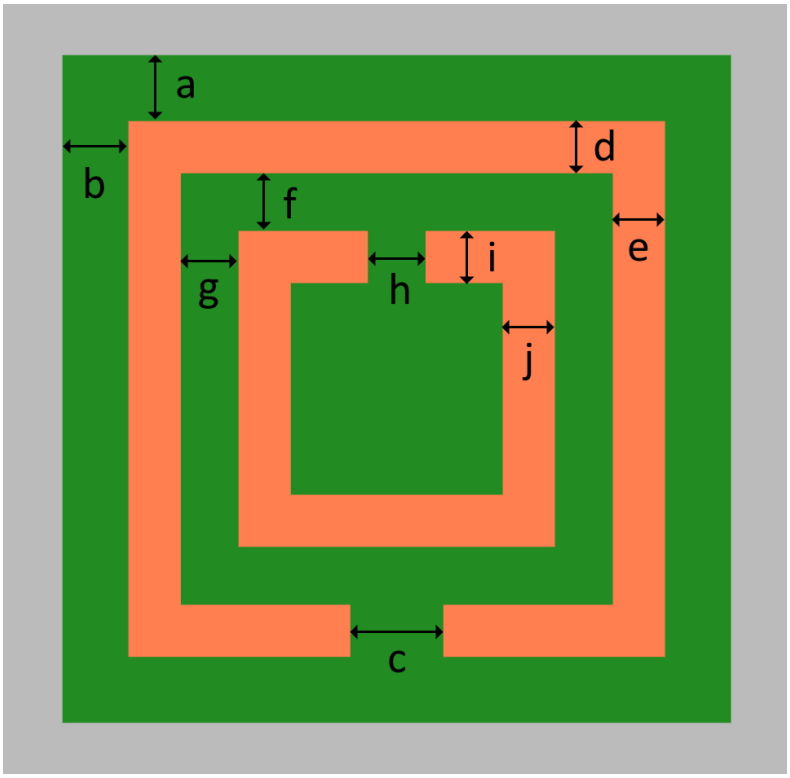
To simplify the logic that prevents shapes from overlapping, the geometries were parametrised in such a way as to make the shapes symmetrical along the concentric horizontal and vertical axes. Thus, most of the editable parameters are symmetric across at least one of the aforementioned axes; examples of these fields include *stem width* and *vertical margin*.

Figure 7 shows the various lengths of the geometries that have been parametrised. *Table 1* matches these parametrised length to the names used to reference them in the application GUI.

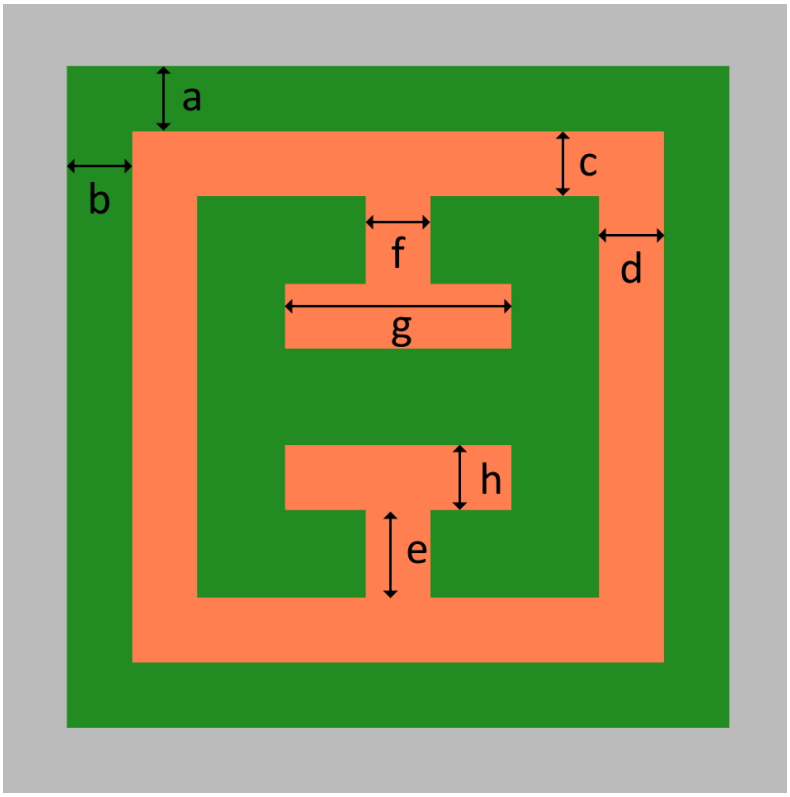
Shape	Label	Parameter
Rectangular-shaped split ring resonator	a	Outer margin vertical
	b	Outer margin horizontal
	c	Outer gap horizontal
	d	Outer thickness vertical
	e	Outer thickness horizontal
	f	Inner margin vertical
	g	Inner margin horizontal
	h	Inner gap horizontal
	i	Inner thickness vertical
	j	Inner thickness horizontal
T-shaped resonator	a	Margin vertical
	b	Margin horizontal
	c	Thickness vertical
	d	Thickness horizontal
	e	Stem height
	f	Stem width
	g	Cap width
	h	Cap height

Table 1: Parameter names as used in the application

Within the application, each of the parameters can be edited independently of the other. The units of the parameters are micrometers [μm].



(a) Parametrised lengths of the rectangular split-ring resonator



(b) Parametrised lengths of the T-shaped resonator

Figure 7: Parametrised lengths of the shapes available in the application

2.4 Simulation setup

Two setups were used in the simulation of the AMC cells.

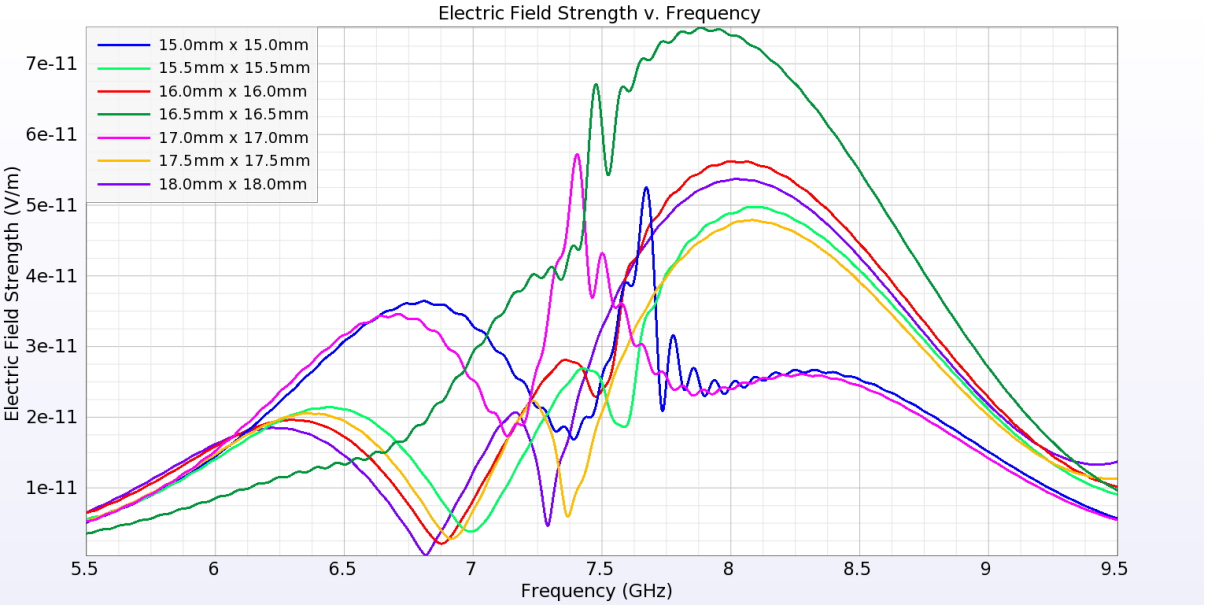
The first setup involves the AMC cell and an excitation source of planar waves that are perpendicular to the cell. The excitation source lies above the AMC while a point sensor is positioned below the AMC to measure the amount of radiation that passes through. The frequency of the excitation source is swept, to determine the resonant frequency of the AMC. In addition to this, the distance between the AMC and the point sensor is varied, to obtain results for different regions of the AMC's field – both far field and near field.

In the second setup, a transmitting antenna is placed above the AMC and a receiving antenna below it. As in the first setup, the frequency of the input source is swept. However, the input source is no longer a planar wave, but the transmitting antenna. The distance between both antennas and the AMC is varied to observe the characteristics of the AMC in its different regions. A point sensor is positioned in the same plane as the transmitting antenna to measure the return characteristics of the AMC. This setup is affected by the coupling between the antennas and the AMC.

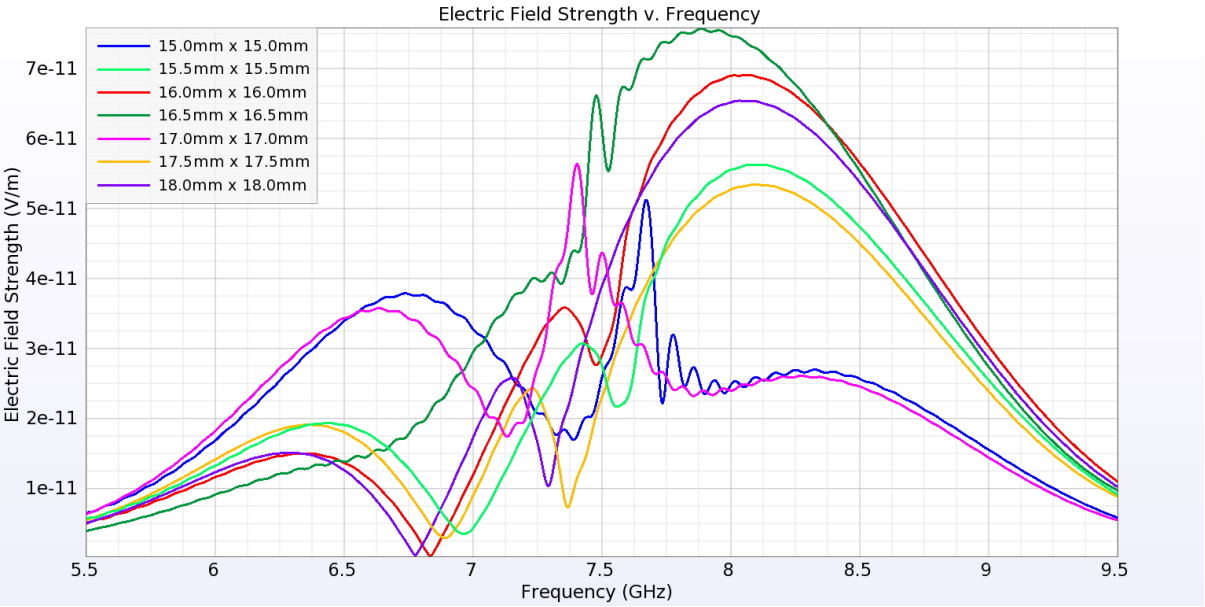
3 Results

3.1 Simulations

The first aspect of the cell geometry to be tested, will involve keeping all of the dimensions of the conductive part of the AMC cell constant. The variable will be the amount of substrate around this conductive layer. From a macro perspective, the distance between adjacent cells will be varied.

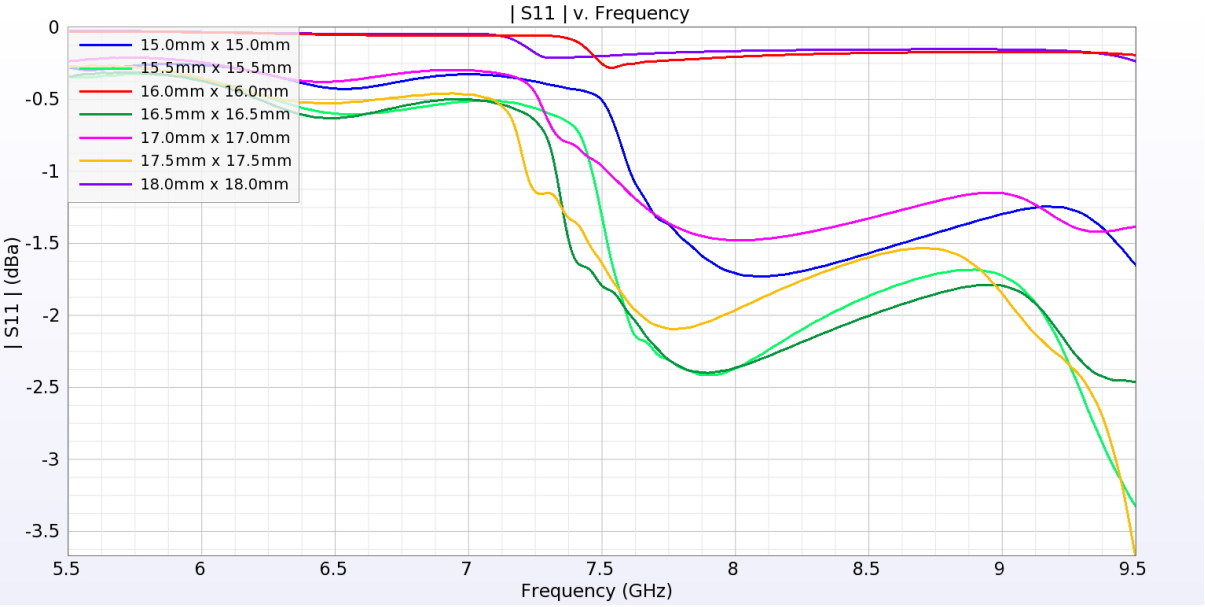


(a) Electric field strengths (Rectangular split ring)

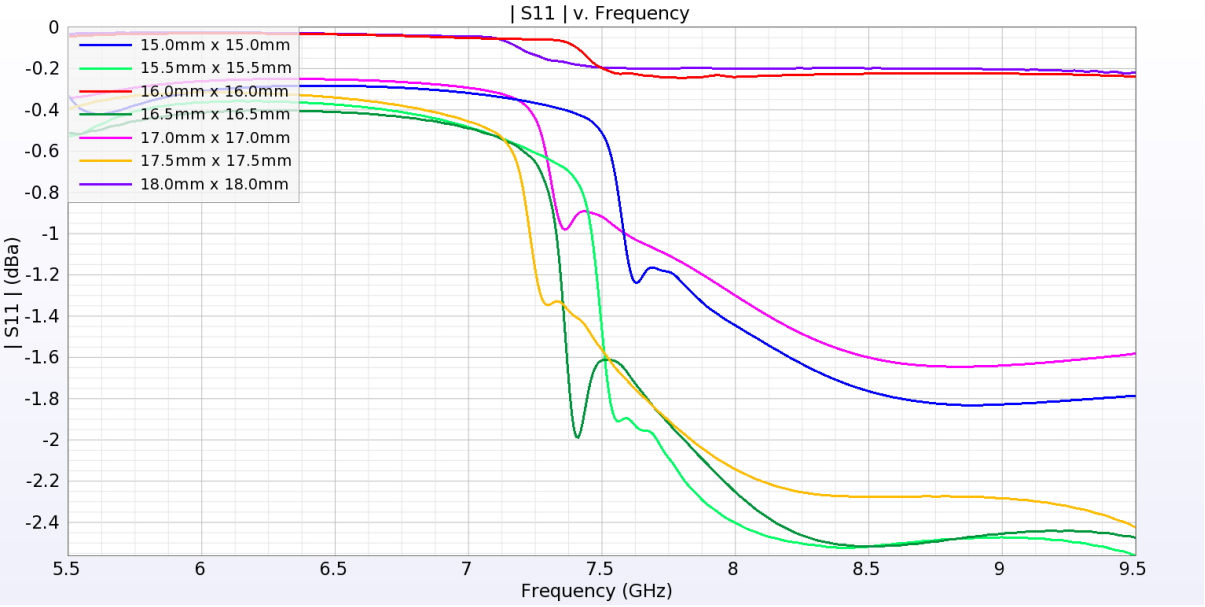


(b) Electric field strengths (T-shaped resonator)

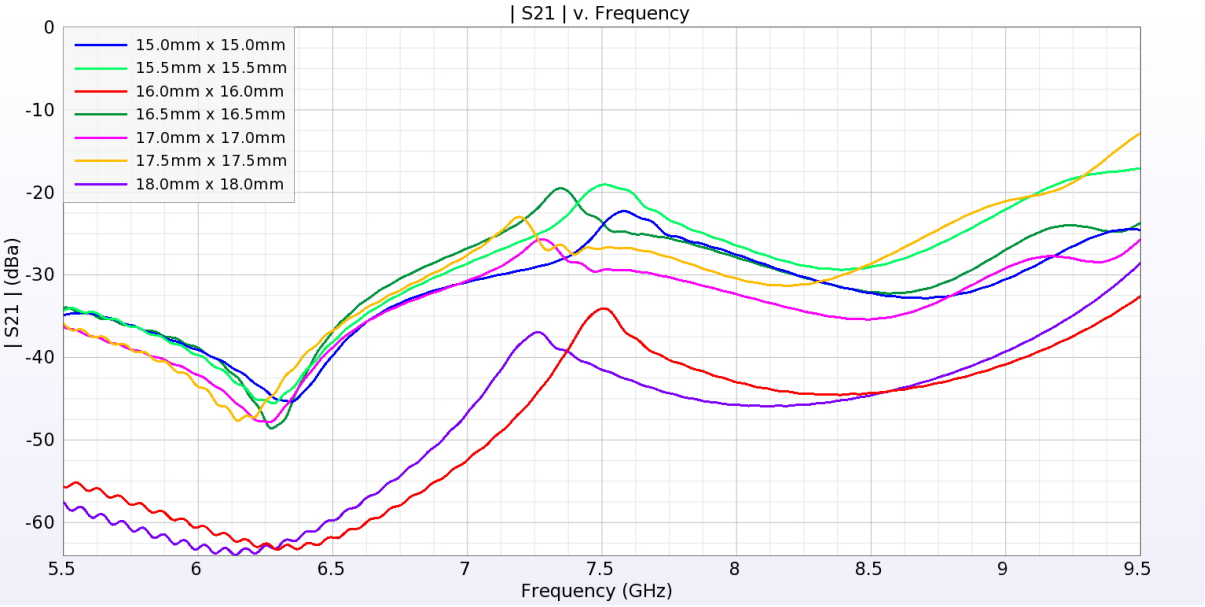
Figure 8: Effect of cell size on resonant frequency



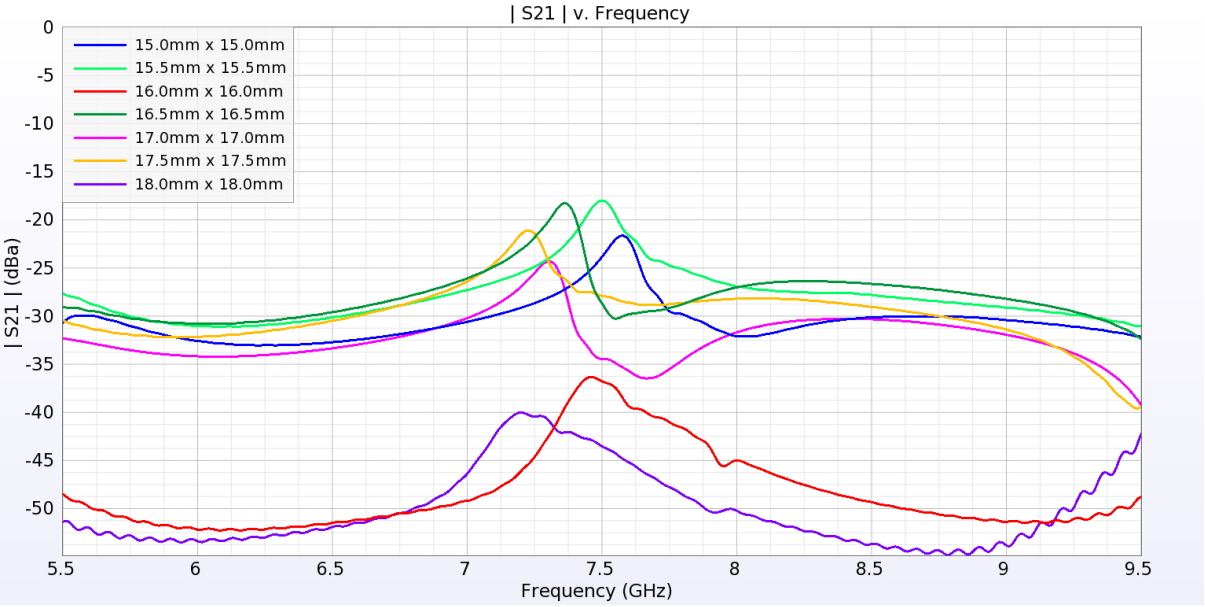
(a) S_{11} parameters (Rectangular split ring)



(b) S_{11} parameters (T-shaped resonator)



(c) S_{21} parameters (Rectangular split ring)



(d) S_{21} parameters (T-shaped resonator)

Figure 9: Effect of cell size on S parameters

Figure 8 shows the effect of varying the spacing between cells of the AMC on the antenna's resonant frequency. The size of the geometrical shape formed by the conductive material was kept constant. Only the amount of substrate surrounding each cell was varied.

It can be observed that a narrow spike in the electric field strength, around the $7.5GHz$ mark, occurs for increments of about $1.5mm$ in the dimensions of the cell. The square cell of side $15.0mm$, $16.5mm$ and $17.0mm$ all exhibit

this spike.

Based of the graphs in *Figure 9*, the cells of side $16.0mm$ and $18.0mm$ have drastically different S-parameters than the other cells. Their S_{11} parameters are fairly flat (maxing out at around $-0.2dBa$) while their S_{21} parameters are roughly $20dBa$ lower than in the other cells. The S_{11} parameters suggest that the AMC cells of side $16.0mm$ and $18.00mm$ are not as good reflectors as the other cells considered cells.

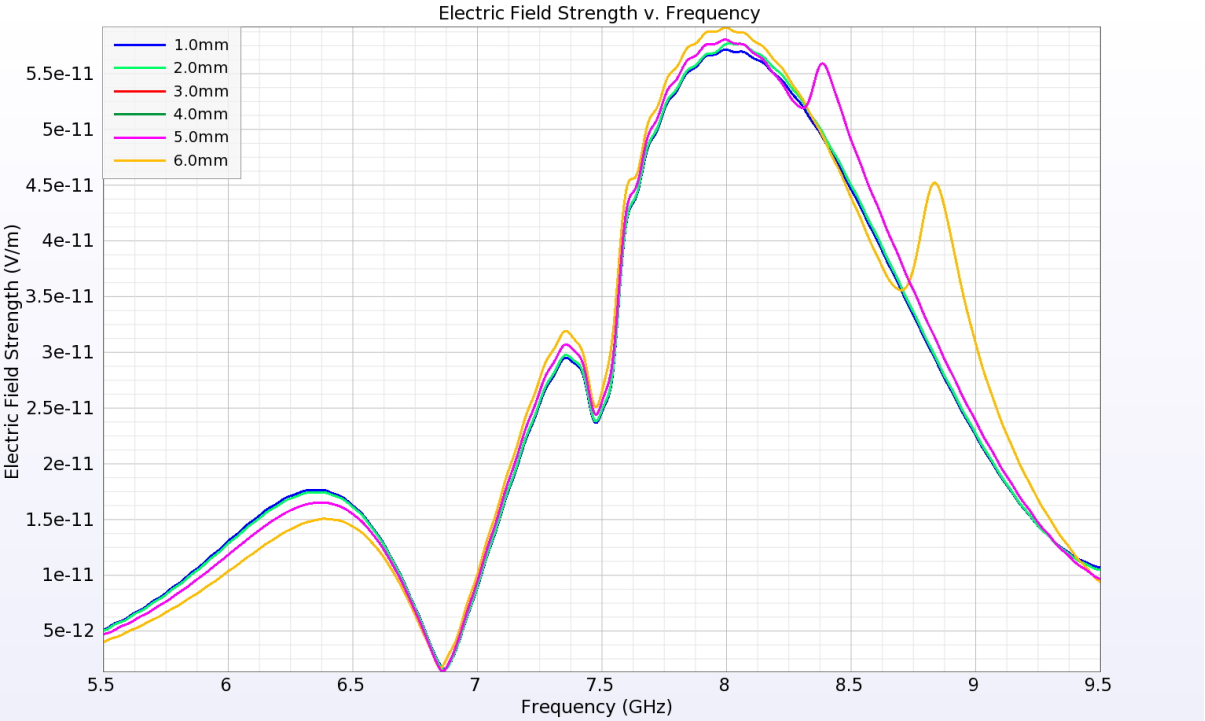
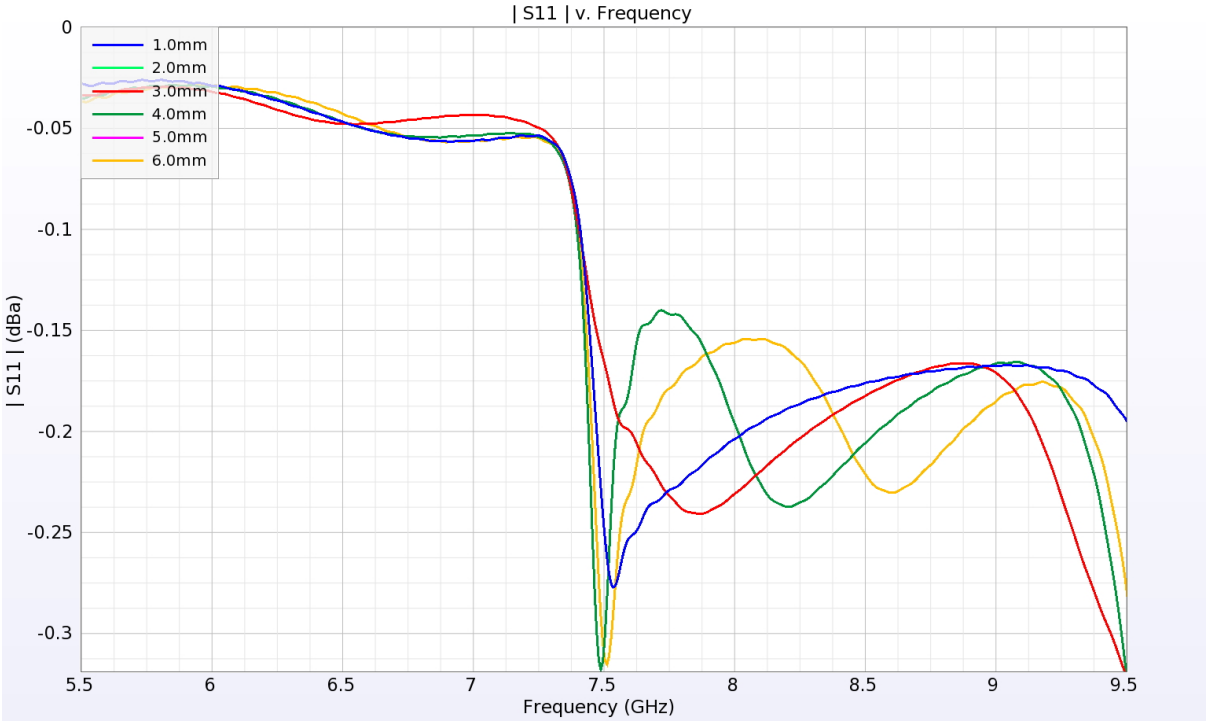
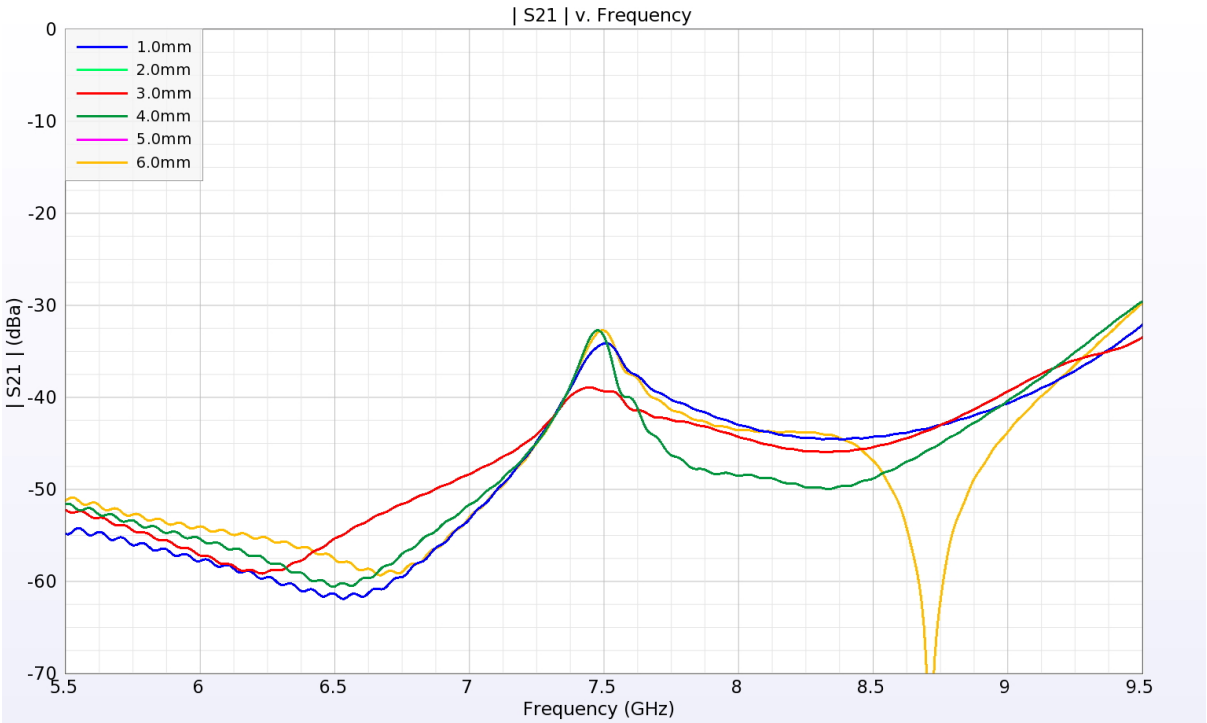


Figure 10: Electric field strength of rectangular split ring for varying gap widths



(a) S_{11} parameters



(b) S_{21} parameters

Figure 11: S-parameters of rectangular split ring for varying gap widths

Figures 10 and *11* show the effect of varying the gap width of the rectangular split ring resonator. The gap is electrically equivalent to a capacitor.

The electric field strength at a receiving antenna placed below the AMC remains largely unaffected as the width of the gap in the split rings is varied. When the width of the gap was $3.0mm$, the S_{11} parameter does not exhibit a sharp drop at $7.5GHz$, but in general the drop at this frequency increases as the width of the gap increases.

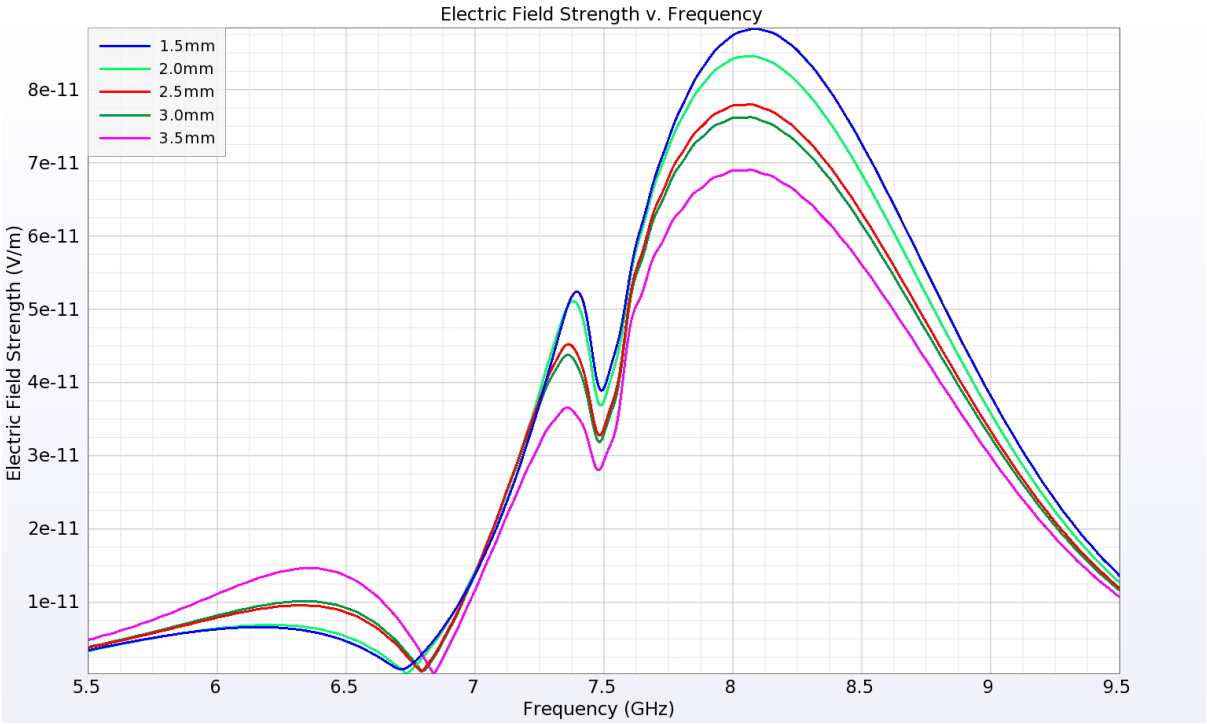
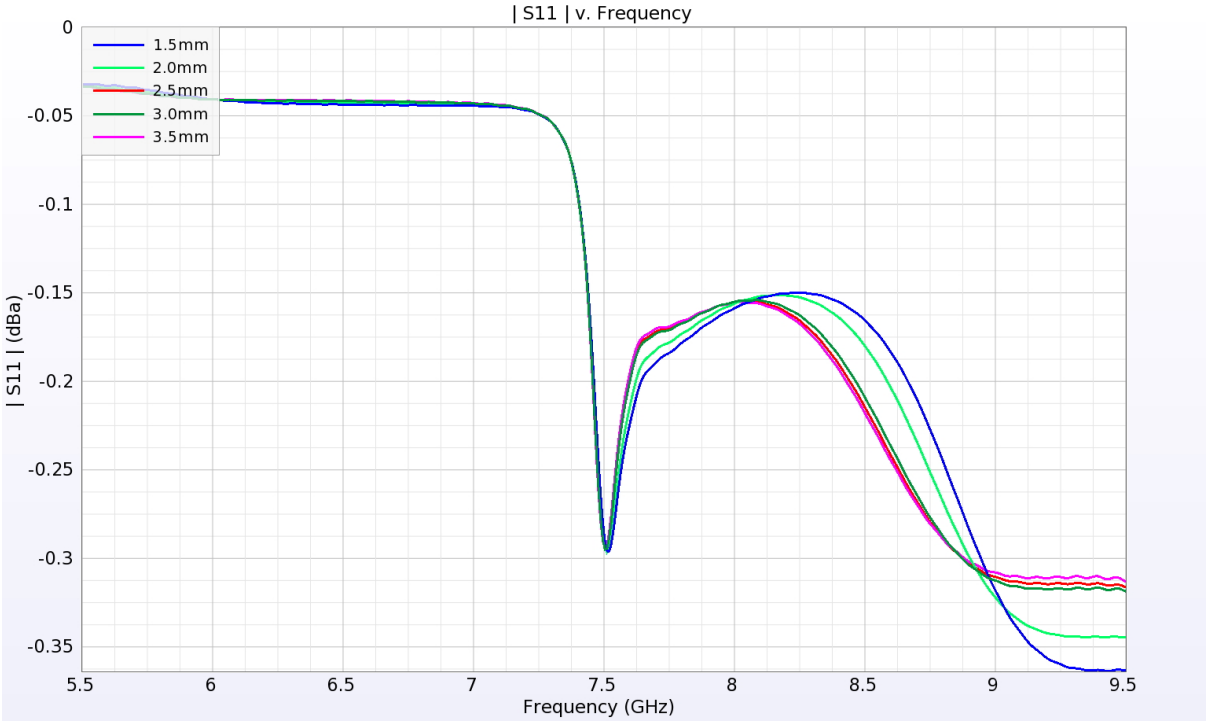
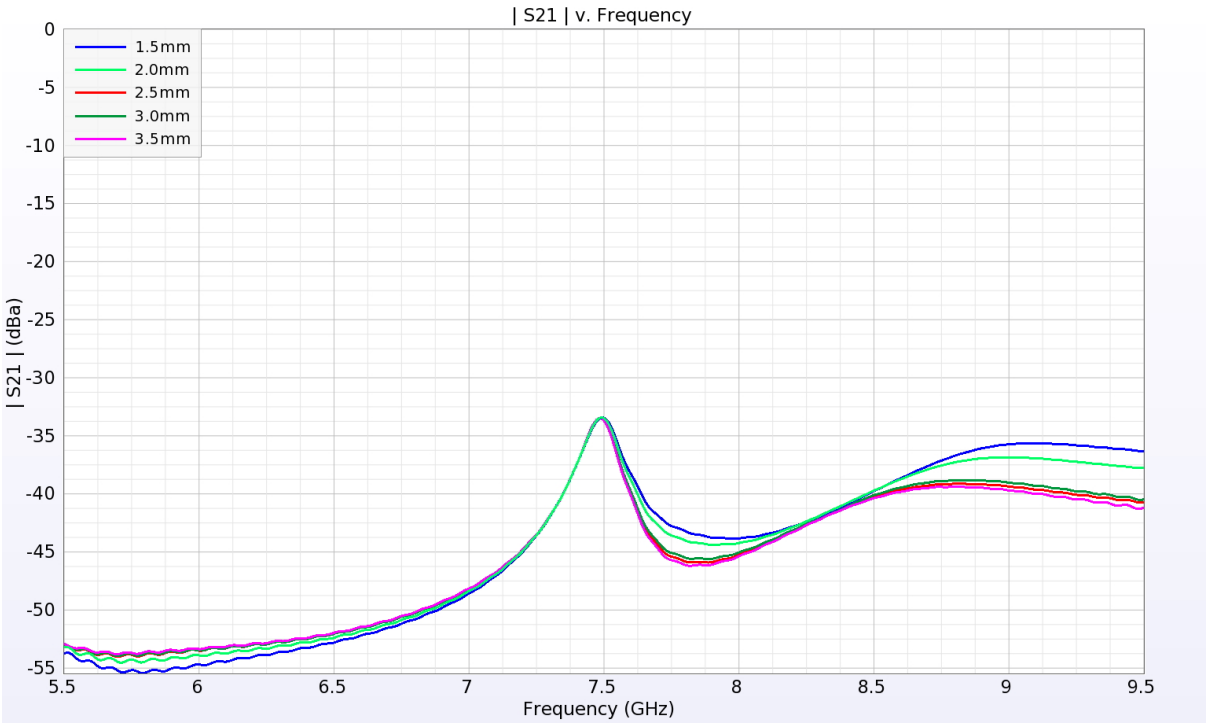


Figure 12: Electric field strength of T-shaped resonator for varying cap heights



(a) S_{11} parameters



(b) S_{21} parameters

Figure 13: S-parameters of T-shaped resonator for varying cap heights

Figures 12 and *13* show the effect of varying the cap height of the T-shaped resonator.

The cap height has little effect on the S-parameters of the AMC cells in the vicinity of $7.5GHz$ on the frequency spectrum. However, the electric field strength of at a receiving antenna placed below the AMC decreases as the cap height increases.

3.2 Verification of results

A quick verification of the results obtained in the simulations can be performed by applying some theoretical formulas. We can calculate the capacitance and inductance of a given AMC cell, the values of which then be used to calculate the resonant frequency.

The formula for determining the resonant frequency is shown in *Equation 1.*

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \tag{1}$$

What's left is to determine the formulas for capacitance, C and inductance, L that are specific to a given cell's geometry. Due to the relative abundance of research into the rectangular split ring resonator[13, 14], there are a number of approaches to calculating its capacitance and inductance. The method used is the one presented by C. Saha et al[13].

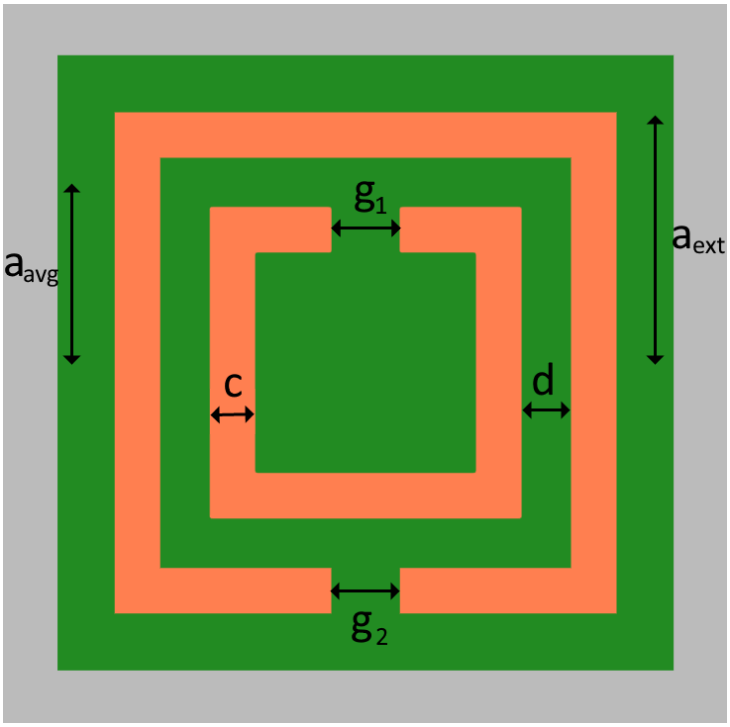


Figure 14: Rectangular split ring cell (not shown, h, the height of the substrate)

The inductance is calculated as follows

$$L = 0.0002l(2.303log_{10}\frac{4l}{c} - \gamma) \tag{2}$$

Where for the given geometry $l = 8a_{ext} - g$ and $\gamma = 2.853$.

The capacitance is calculated as follows

$$C = \left(2a_{avg} - \frac{g}{2}\right) C_{pul} + \frac{\epsilon_0 ch}{2g_1} \tag{3}$$

C_{pul} is the capacitance per unit length and is calculated as follows

$$C_{pul} = \frac{\sqrt{\epsilon_r}}{c_0 Z_0} \tag{4}$$

Where c_0 is the velocity of light and Z_0 is the impedance whose formula is shown below

$$Z_0 = \frac{120\pi}{\sqrt{\epsilon_e}} \frac{K(k)}{K(k')} \tag{5}$$

Where $epsilon_e$ is the effective permittivity and can be calculated as follows

$$\epsilon_e = 1 + \frac{\epsilon_r - 1}{2} \frac{K(k')K(k_1)}{K(k)K(k_1')} \tag{6}$$

$K(k)$ is a complete elliptical function of the first kind. $K(k')$ is its complimentary function.

$$k = \frac{\frac{c}{2}}{\frac{c}{2} + d} \tag{7}$$

$$k_1 = \frac{sinh(\frac{c}{2}\frac{\pi}{2h})}{sinh([\frac{c}{2} + d]\frac{\pi}{2h})} \tag{8}$$

For $0 \leq k \leq 0.7$ the following approximation holds

$$\frac{K(k)}{K(k')} = \left[\frac{1}{\pi} ln\left(2\frac{1+\sqrt{k'}}{1-\sqrt{k'}}\right)\right]^{-1} \tag{9}$$

For the case in which $a_{ext} = 7mm$, $a_{avg} = 5.5mm$, $c = 1mm$, $d = 1mm$, $g_1 = g_2 = 1mm$ and $h = 1mm$, *Equation 1* resolves as shown

$$f_0 = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{28nH \cdot 0.52pC}} = 1.32GHz \tag{10}$$

4 Summary

4.1 Iteration of AMC cell designs

As presented in the simulation results, changing various parameters of the geometry of AMC cells can have an impact on their performance as reflectors in antennas. These changes can be highly specific to the general geometry of the cell, making simulation a convenient, time-saving method of validating different iterations of a design.

The iteration speed comes down to two factors, the speed of the simulations themselves, and the speed with which designs can be tweaked. The AMC cell design program aimed at addressing the latter, by providing a quick and intuitive way to tweak cell designs, albeit with some restrictions in place (e.g. horizontal and vertical symmetry restrictions).

As AMCs become more mainstream, the new tools will need to be developed to aid in their development, and existing tools will need to adjusted to streamline the prototyping process. While basic in its functionality, the AMC cell design program does provide a way to speed up the prototyping phase of the design of AMC cells.

4.2 Analysis of the AMC Cell Design Program

The program developed to aid in the design of AMC cells served its purpose of being able to quickly apply tweaks to a given cell's geometry. This section covers the advantages and disadvantages of said program, as well as improvements that could be made to increase its functionality and usability.

Advantages

- **Rapid prototyping**

  The program allows parameters of a given geometry to be edited at a whim.

- **Visual feedback**

  The section of the program that renders the current state of the cell is a feature that provides the operator with a level of assurance that their changes have taken effect.

Disadvantages

- **Incompatible modelling definitions**

  Owing to the parametrisation of the geometries making up the AMC, the models are not directly exportable to other software. Likewise, pre-existing shapes cannot be imported as-is into the program. They would need to be processed into the parametrised format.

- **Geometric constraints**

  Each model of an AMC cell within the program includes not just the parameters defining its geometry, but functions which describe the valid relations between these parameters. These constraints remove the ability to change the parameters in such a way that the general geometry of the cell is fundamentally altered. While this may be a useful handrail that restricts the possible geometries that can be designed by editing the parameters, it is by far the most time-consuming process of creating the internal geometry definitions within the program.

Improvement

While the current state of the AMC cell design program was largely sufficient for the intended purpose, there exists room for improvement, not just for the current problem set that it was developed to tackle, but for scenarios with slightly different requirements. Some of these possible improvements include:

- **Utilising files for interfacing with the simulation software**

  The current implementation involves copying the the XFdtd script code to the operating system clipboard, so that it may be later ran in the simulation software. This does create a somewhat disjointed workflow in that with each iteration of the AMC cell designed in the program, the code must be manually pasted somewhere so that it is not lost when overwritten by code for the next iteration is copied to the clipboard.

  Since XFdtd also supports running scripts from locally stored files, it is possible to provide the code that needs to be run in a file rather than requiring it to be copied and pasted. This feature would likely also benefit from providing the operator with the ability to provide a filename that should be used for the downloaded file. This is a fairly simple feature to implement using the existing browser APIs.

- **Automatic graphs**

  The code generated by the AMC cell design program currently draws the shapes in XFdtd, then queues the simulations. The results of the simulation need to be plotted manually.

  The XFdtd software provides an API for creating plots of the simulation results. Knowing the desired plots for the simulations, the program could be extended to include the necessary scripts to create plots in its generated output.

- **Grouping multiple design iterations**

  As mentioned previously, each iteration of a cell design needs to be one at a time. There is no queueing system in the AMC cell design program.

  This particular feature could be implemented in two ways, the first and simpler of the two, would involve the operator manually queueing each iteration by clicking a `Queue` button, which would append the generated code for that particular iteration of the cell design to an in memory representation of program output. This could be done multiple times, and once the desired iterations had been designed, the combined output could be copied to the clipboard.

  The second method of grouping the code for multiple design iterations would involve providing the operator with more control over the queueing system. This would likely necessitate an alternative input method of cell parameters than the one currently provided in the program. A Domain Specific Language (DSL) could be created to allow the operator to specify the parameters for an iteration of an AMC cell. Within the AMC cell design program, this information (i.e. the set of parameters for each iteration) could be provided via a text input element or by uploading a text file containing this data. The generated code would then include the necessary scripts for all the provided iterations.

  Both of these methods would make it possible to obtain a single piece of code for multiple iterations of cell designs, which when ran in XFdtd, would queue simulations for each of the iterations.

References

[1] D Sievenpiper et al. “High-impedance electromagnetic surfaces with a forbidden frequency band”. In: *IEEE Transactions on Microwave Theory and Techniques* 47 (Nov. 1999), pp. 2059–2074. DOI: 10.1109/22.798001.

[2] J.R. Sohn et al. “Comparative Study On Various Artficial Magnetic Conductors For Low Profile Antenna”. In: *Progress In Electromagnetics Research* 61 (2006), pp. 27–37. DOI: 10.2528/PIER06011701.

[3] M Abu et al. “Designing Artificial Magnetic Conductor at 2.45 GHz for Metallic Detection in RFID Tag Application”. In: *International Journal of Engineering and Technology* 6 (Mar. 2014), pp. 427–435.

[4] A Jafargholi, M.R. Booket, and M Veysi. “Applications of Artificial Magnetic Conductors in Monopole and Dipole Antennas”. In: (May 2012). DOI: 10.5772/35257.

[5] X Begaud. “Wideband Antennas and Artificial Magnetic Conductors”. In: *Non-standard Antennas*. Ed. by F Le Chevalier, D Lesselier, and R Staraj. John Wiley & Sons, Inc, 2011. Chap. 9, pp. 183–200.

[6] P Padilla, J.M. Fernández, and M Sierra-Castañer. “Characterization of Artificial Magnetic Conductor Strips for Parallel Plate Planar Antennas”. In: *Microwave and Optical Technology Letters* 50 (Feb. 2008), pp. 498–504. DOI: 10.1002/mop.23092.

[7] G.V. Eleftheriades and K.G. Balmain, eds. *Negative-refraction metamaterials*. John Wiley & Sons, Ltd, 2005.

[8] R.W. Ziolkowski. “Metamaterials and Antennas”. In: *Handbook ofAntenna Technologies*. Ed. by N.C. Zhi et al. Springer Science+Business Media, 2016, pp. 287–320.

[9] D.S. Wang, C.H. Chan, and S.-W. Qu. “Frequency Selective Surfaces”. In: *Handbook ofAntenna Technologies*. Ed. by N.C. Zhi et al. Springer Science+Business Media, 2016, pp. 471–526.

[10] F Capolino, ed. *Theory and Phenomena of Metamaterials*. CRC Press, 2009.

[11] J.S. Roper. “Design Of A Circular Reflectarray With A Performance Comparableto The Typical Rectangular Reflectarray”. Georgia Institute of Technology. 2017.

[12] *Browser support for ECMAScript 5*. https://caniuse.com/#feat=es5. Accessed: 2019-08-04.

[13] C. Saha, J Siddiqui, and Y.M.M. Antar. “Square split ring resonator backed coplanar waveguide for filter applications”. In: Aug. 2011. ISBN: 978-1-4244-5117-3. DOI: 10.1109/URSIGASS.2011.6050665.

[14] Andrea Vallecchi, Ekaterina Shamonina, and Christopher Stevens. “Analytical model of the fundamental mode of 3D square split ring resonators”. In: *Journal of Applied Physics* 125 (Jan. 2019), p. 014901. DOI: 10.1063/1.5053482.