EXTRACTING FREQUENCY DOMAIN NETWORK PARAMETERS USING THE LATENCY INSERTION METHOD

BY

ROBERT F. KUMMERER

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Professor José E. Schutt-Ainé

ABSTRACT

In this thesis, a tool is presented for extracting frequency domain scattering parameters that utilizes a time domain circuit simulation algorithm called the latency insertion method (LIM). LIM is a finite difference formulation comparable with current industry standard circuit simulators, with several advantages, namely linear numerical complexity.

In this work, the theoretical underpinnings of LIM and the tool are established. The tool is comprehensively explained and validated, and its limitations are discussed. Finally, an example problem is analyzed using this new tool, and the results are presented.

To my family, for their love and support.

ACKNOWLEDGMENTS

Foremost, I would like to acknowledge my advisor, Professor Schutt-Ainé. His continued support and guidance were critical to my graduate work. When I was an undergraduate student, he helped guide me along this research path towards graduate school, which has proven to be fruitful. Through Professor Schutt-Ainé, I was exposed to many intelligent, influential, or otherwise important people, contributing to my growth in graduate school.

In addition, I thank my groupmates. Whether it was advice on classes, research, or where free food was available, my groupmates proved to be an invaluable source of information. In particular, I thank Xu Chen for his help with my two publications and Gene Shiue for his advice on LIM. Finally, I thank all of the members of the Synclesis group for their input and experience at our weekly meetings.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Statement

The simulation and computation of frequency domain macromodels for passive structures is a considerably computationally complex process. Modern integrated circuits are becoming very small and operating at higher and higher clock rates, necessitating accurate simulation to preserve signal integrity. These driving factors cause the models to become increasingly complex and difficult to simulate. Traditional circuit simulators such as Simulation Program with Integrated Circuit Emphasis (SPICE) have a superlinear run time due to required matrix inversions, and they become increasingly inefficient as the circuit size becomes extremely large. The latency insertion method (LIM) [1, 2, 3, 4, 5, 6], is a circuit simulation algorithm that has been shown to be an efficient and accurate tool for the fast simulation of very large circuits. LIM's most important advantage over SPICE is its linear numerical complexity, which leads to a much faster simulation time relative to SPICE for very large circuits.

This work sets out to extend the capabilities of LIM by introducing the capacity for frequency domain simulation. Characterizing circuit networks in the frequency domain is incredibly important and a critical function of any circuit simulator. The tool presented in this thesis leverages the simplicity and speed of LIM to generate frequency domain S-parameters. With two time domain simulations, the complete set of S-parameters can be calculated using the fast Fourier transform (FFT).

1.2 Outline

Chapter 2 details the various LIM formulations that may be used. In addition to this, implementation considerations are provided to improve the performance of LIM. In Chapter 3, the theory, implementation, and validation of the tool are discussed. Chapter 4 shows an application of the tool to a specific problem, where a power plane with material and geometric uncertainties is modeled and simulated. Much of the information in Chapters 3 and 4 has been previously published in Kummerer et al. [4, 5], and is adapted here with permission. Finally, in Chapter 5, conclusions and future work are discussed.

CHAPTER 2

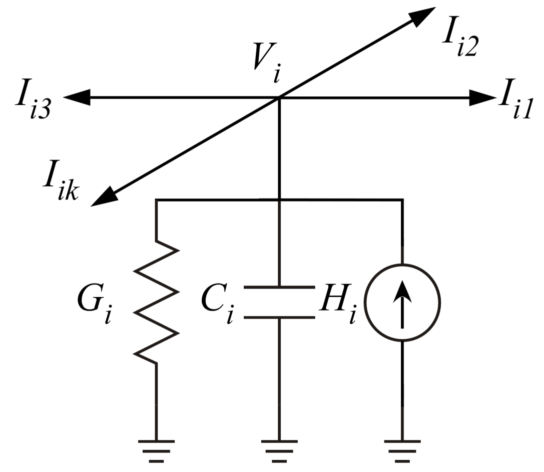
LATENCY INSERTION METHOD

LIM is a finite difference algorithm developed for the time domain simulation of circuits [1, 2, 3, 4, 5, 6]. Similar to Yee's finite difference time domain (FDTD) algorithm [7], LIM alternately calculates the current through each branch and the voltage at each node, separated by half time steps, in a leapfrogging manner. The major advantage of LIM over the industry standard circuit simulation algorithm SPICE is LIM's linear numerical complexity. The run time of LIM is directly proportional to the number of nodes and branches in a given circuit, whereas the run time of SPICE has a superlinear dependence on the number of nodes in the circuit. In the following sections, various LIM formulations and their implementations are discussed.

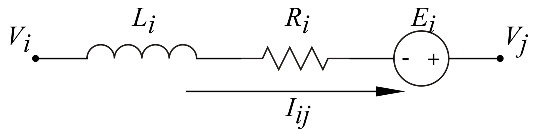
2.1 Formulation

A circuit, when being simulated using the LIM algorithm, must be broken up into nodes and branches. Voltages are defined at each node, and currents are defined through each branch. A node has a capacitance, conductance, and current source in parallel to ground, as shown in Figure 2.1(a). Nodes are connected to other nodes by branches. Branches contain a series combination of an inductance, resistance, and voltage, shown in Figure 2.1(b). If all nodes do not have capacitors to ground or all branches do not have inductors, very small fictitious ones must be inserted.

Evaluating Kirchhoff's current law (KCL) at a given node i, Equation 2.1 can be derived. Similarly, evaluating Kirchhoff's voltage law (KVL) across a branch ij yields Equation 2.2. As seen in Equations 2.1 and 2.2, the branch currents are indexed at exact time steps, while the voltages are indexed at half time steps. From these two equations, various formulations may be derived.



(a) Node topology



(b) Branch topology

Figure 2.1: LIM topologies

$$C_i\left(\frac{V_i^{n+\frac{1}{2}}-V_i^{n-\frac{1}{2}}}{\Delta t}\right)+G_iV_i^{n-\frac{1}{2}}-H_i^n=-\sum_{k=1}^{M_i}I_{ik}^n \tag{2.1}$$

$$V_i^{n+\frac{1}{2}}-V_j^{n+\frac{1}{2}}=L_{ij}\left(\frac{I_{ij}^{n+1}-I_{ij}^n}{\Delta t}\right)+R_{ij}I_{ij}^n-E_{ij}^{n+\frac{1}{2}} \tag{2.2}$$

The explicit formulation is given by directly solving for future voltage and currents from Equations 2.1 and 2.2. Solving Equation 2.1 for the future voltage at node i, $V_i^{n+\frac{1}{2}}$, yields Equation 2.3.

$$V_i^{n+\frac{1}{2}}=V_i^{n-\frac{1}{2}}+\frac{\Delta t}{C_i}\left(-\sum_{k=1}^{M_i}I_{ik}^n-G_iV_i^{n-\frac{1}{2}}+H_i^n\right) \tag{2.3}$$

Similarly, solving Equation 2.2 for the future branch current, I_{ij}^{n+1}, produces Equation 2.4.

$$I_{ij}^{n+1}=I_{ij}^n+\frac{\Delta t}{L_{ij}}\left(V_i^{n+\frac{1}{2}}-V_j^{n+\frac{1}{2}}-R_{ij}I_{ij}^n+E_{ij}^{n+\frac{1}{2}}\right) \tag{2.4}$$

For the implicit formulation, G_i and R_{ij} in Equations 2.1 and 2.2 are replaced by $G_iV_i^{n+\frac{1}{2}}$ and $R_{ij}I_{ij}^{n+1}$, respectively. Again, solving for $V_i^{n+\frac{1}{2}}$ and I_{ij}^{n+1} yields the following two equations.

$$V_i^{n+\frac{1}{2}}=\left(\frac{C_i}{\Delta t}+G_i\right)^{-1}\cdot\left(\frac{C_i}{\Delta t}V_i^{n-\frac{1}{2}}-\sum_{k=1}^{M_i}I_{ik}^n+H_i^n\right) \tag{2.5}$$

$$I_{ij}^{n+1} = \left(\frac{L_{ij}}{\Delta t} + R_{ij}\right)^{-1} \cdot \left(\frac{L_{ij}}{\Delta t} I_{ij}^{n} + V_i^{n+\frac{1}{2}} - V_j^{n+\frac{1}{2}} + E_{ij}^{n+\frac{1}{2}}\right) \tag{2.6}$$

Finally, the semi-implicit form is found by substituting $\frac{G_i\left(V_i^{n+\frac{1}{2}}+V_i^{n-\frac{1}{2}}\right)}{2}$ and $\frac{R_{ij}\left(I_{ij}^{n+1}+I_{ij}^{n}\right)}{2}$ for the conductance and resistance terms, G_i and R_{ij}.

$$V_i^{n+\frac{1}{2}} = \left(\frac{C_i}{\Delta t} + \frac{G_i}{2}\right)^{-1} \cdot \left(\left(\frac{C_i}{\Delta t} - \frac{G_i}{2}\right) V_i^{n-\frac{1}{2}} - \sum_{k=1}^{M_i} I_{ik}^{n} + H_i^{n}\right) \tag{2.7}$$

$$I_{ij}^{n+1} = \left(\frac{L_{ij}}{\Delta t} + \frac{R_{ij}}{2}\right)^{-1} \cdot \left(\left(\frac{L_{ij}}{\Delta t} - \frac{R_{ij}}{2}\right) I_{ij}^{n} + V_i^{n+\frac{1}{2}} - V_j^{n+\frac{1}{2}} + E_{ij}^{n+\frac{1}{2}}\right) \tag{2.8}$$

All of the LIM formulations mentioned above are conditionally stable, meaning that there exists a maximum time step for which this algorithm is numerically stable. Yee's FDTD has a very similar stability criterion. The worst case maximum stable time step is given by $\Delta t_{max} = \sqrt{L_{min}C_{min}}$, where L_{min} and C_{min} are the minimum inductor and capacitor values in the whole circuit. These values generally come from the fictitious elements, if they exist [2].

2.2 Implementation

The implementation of LIM is relatively straightforward, especially when only using the basic formulations shown above. More sophisticated implementations such as Block LIM [3] exist, but I did not use them because simpler methods may be used without loss of generality. In this section, the specifics of the implementation of LIM and the improvements made to increase performance are presented.

The core of LIM is simply the two updating equations, Equations 2.3 and 2.4, Equations 2.5 and 2.6, or Equations 2.7 and 2.8. The current updating equation is repeated for every branch and the voltage updating equation is repeated for every node. This is done at each time step of the simulation.

The algorithm is shown below.

Algorithm 1 LIM Algorithm

```
for $t = 0$ to Num Time Steps do
    for $node = 0$ to Num Nodes do
        Update Voltage Equation
    end for
    for $branch = 0$ to Num Branches do
        Update Current Equation
    end for
end for
```

Other than the two updating equations, there is no common framework of organizing a LIM simulation. For this thesis a relatively simple interface to get the LIM simulations running was created. Initially, an adjacency matrix was used to describe the connections between circuit nodes. The matrix is of size $N_{nodes} \times N_{nodes}$, where N_{nodes} is the number of nodes in the circuit. The matrix has entries of zero when the nodes are not connected and one when the two nodes are connected. Similarly, the branch current, resistance, and inductance are also stored in an $N_{nodes} \times N_{nodes}$ matrix. The node voltages, conductances, and capacitances are stored in a $N_{nodes} \times 1$ vector.

As the size of the circuit increases, this method becomes inefficient because the two-dimensional matrices containing branch currents, resistances, and inductances become massive and very sparse. At each time step and at each node, an entire row of these matrices would be looped through. This bogged down the simulation and was unnecessarily memory intensive, considering most of the values are zero. In order to improve this, a new data storage method for the branch information was realized by compressing the sparse square matrix into a list of lists in Python or a vector of vectors in C++ from the standard library. The adjacency list serves as a form of a branch list. The algorithm for compressing the large adjacency, resistance, and inductance matrices is given below.

This process is very simple. The algorithm simply iterates through the adjacency matrix and finds non-zero elements. The index of that non-zero element is appended onto a list with all the other indices of nodes adjacent to node i. When an item is appended to the adjacency list, the branch resistance and inductance between the two nodes are also added into their own list to be stored similarly. Now, when updating the voltage at each node, it becomes

Algorithm 2 Compress Matrices

```
adjacentNodesList = [ ]
adjacentRList = [ ]
adjacentLList = [ ]
for $i = 0$ to Num Nodes do
    nodeList = [ ]
    rList = [ ]
    lList = [ ]
    for $j = 0$ to Num Nodes do
        if A$[i, j]$ == 1 then
            nodeList.append($j$)
            rList.append($R\,[i, j]$)
            lList.append($L\,[i, j]$)
        end if
    end for
    adjacentNodesList.append(nodeList)
    adjacentRList.append(rList)
    adjacentLList.append(lList)
end for
```

less computationally difficult to calculate the current flowing into the node $\left(\sum_{k=1}^{M_i} I_{ik}^{n}\right)$.

CHAPTER 3

SCATTERING PARAMETER CALCULATION USING LIM

3.1 Theory

In this section, the theoretical underpinnings of the tool are discussed. The most important concept is that of S-parameters, as the main goal of this thesis is to present a tool that extracts them. Secondly, understanding the FFT is required, because the fundamental operation of the tool requires conversion of time domain simulation data into frequency domain S-parameters.

3.1.1 Scattering Parameters

S-parameters are a frequency domain representation of an N-port network. S-parameters completely describe a linear network by giving ratios of reflected and transmitted voltage waves, as referenced from an incident voltage wave on a given port. In order to measure any N-port network parameter, the outputs must be terminated by a certain load while the input is excited. While Z and Y parameters also offer a complete description of a network, they can be cumbersome to measure. They require open and short terminations for measurement, which are difficult to realize in situations involving active devices and high frequencies. S-parameters simply require a termination equal to the reference impedance Z_0. Z_0 is generally taken to be 50 Ω, and it will be throughout this thesis. This property makes S-parameters very attractive for characterizing networks at high frequencies.

In this work, only two-port S-parameters are discussed. The generalized root power waves of a two-port S-parameter system are given by Equations 3.1, 3.2, 3.3, and 3.4 [8]. These are called root power waves because when they are squared, the resulting value is the power of the given wave. The incident root power wave, a_1, is the excitation of the network. This incident root power wave will give rise to a reflected root power wave, b_1, and a transmitted root power wave, b_2. In a realistic measurement scenario, a_1 would be a single-frequency sinusoid with a given power, supplied by a network analyzer. The reflected and transmitted waves, b_1 and b_2, would be measured with the aid of directional couplers, splitting the backward traveling wave from the forward traveling wave. The termination on port 2 of the network must be matched to Z_0 to ensure that no power from b_2 is reflected back into the system. A graphical representation of this setup is given in Figure 3.1.

$$a_1 = \frac{V_1 + Z_0 I_1}{2\sqrt{Z_0}} \tag{3.1}$$

$$a_2 = \frac{V_2 - Z_0 I_2}{2\sqrt{Z_0}} \tag{3.2}$$

$$b_1 = \frac{V_1 - Z_0 I_1}{2\sqrt{Z_0}} \tag{3.3}$$

$$b_2 = \frac{V_2 + Z I_2}{2\sqrt{Z_0}} \tag{3.4}$$

Knowledge of a_1, b_1, and b_2 allows us compute S_{11} and S_{21} of the network. Taking the ratios of the reflected or transmitted waves with the incident wave yields these values, as shown in Equations 3.5 and 3.6. Repeating this process by matching port 1 to Z_0, exciting port 2 with the incident wave a_2, and measuring the reflected (b_2) and transmitted (b_1) waves the remainder of the S matrix can be computed. Equations 3.1, 3.2, 3.3, and 3.4 detail how the forward and backward traveling waves are calculated from voltage and current measurements at the ports of the device under test (DUT). The voltage and current measurement locations are shown in Figure 3.2.

$$S_{11} = \left.\frac{b_1}{a_1}\right|_{Z_L = Z_0} \tag{3.5}$$

$$S_{21} = \left.\frac{b_2}{a_1}\right|_{Z_L = Z_0} \tag{3.6}$$

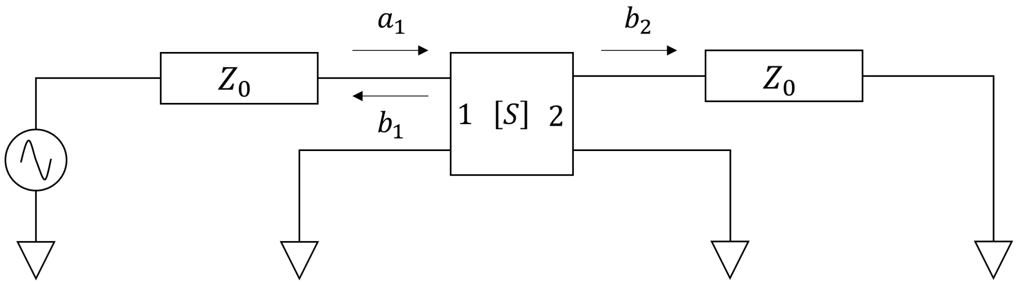


Figure 3.1: S-parameter port 1 measurement setup showing root power waves

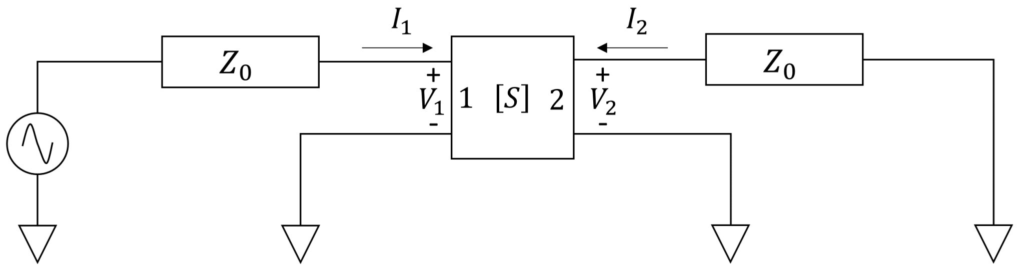


Figure 3.2: S-parameter measurement setup showing voltage and current

3.1.2 Fast Fourier Transform

The FFT forms the backbone of this tool, because it is able to take the time domain data and convert it into frequency domain data. For information about the FFT, we have relied on Manolakis and Ingle [9]. The FFT is an algorithm to exactly calculate the discrete Fourier transform (DFT), which is a mapping of discrete time domain samples to discrete frequency domain samples. It is called the fast Fourier transform because it has a numerical complexity of O(N log N) compared to O(N^2) for the DFT, where N is the number of samples in the signal. While the DFT has many uses, it is used specifically for spectral analysis in this thesis.

The exact expression for the DFT is given in Equation 3.7. X_k is the frequency domain representation of x_n, which is a time domain signal of length N. The output sequence, X_k, is also of length N. Generally, the time domain data, x_n, must be equally spaced in order to be processed correctly by the FFT. This is convenient for LIM because the time step is fixed, while SPICE has a variable time step.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{\frac{-j2\pi}{N}kn} \tag{3.7}$$

When the input to the FFT is purely real, which is the case in this thesis, $X_{N-k} = X_k^*$. Therefore, the second half of the output of the FFT is redundant and may be discarded. X_k is indexed with integers, but it is only useful when frequency data corresponds with an actual frequency. The frequency spacing Δf of each FFT bin k is given in Equation 3.8. The frequency of a given bin is then given by $k\Delta f$.

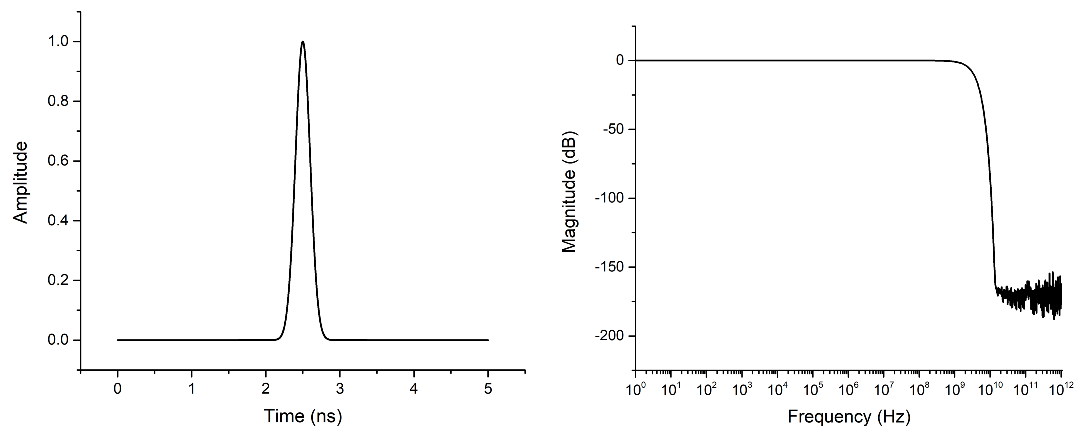
$$\Delta f = \frac{1}{N\Delta t} \tag{3.8}$$

The frequency resolution of the output of the FFT is governed by the number of time domain samples (N) and the time step of the simulation (Δt). Increasing either will increase the frequency resolution of the output. Zero padding commonly is used to increase the effective number of samples in the time domain by appending many zeros to the end of the time domain sequence. This has the effect of increasing frequency resolution. However, this does not actually improve the ability to distinguish two signals of similar frequencies. However, increasing the length of the true, non-zero-padded time domain sequence will result in an FFT output containing more information, allowing nearby signals to be more easily distinguished.

An effect of the inherent finite nature of the FFT is called spectral leakage. For example, when sampling a pure sinusoid, the effect of truncating the time domain sequence results in spreading of the signal's power across the frequency spectrum. This problem can be mitigated by using longer time domain sequences or various windows, but it cannot be perfectly removed. The problem of spectral leakage is not the focus of the thesis, but in the future, it should be taken into account for sensitivity analysis.

3.1.3 Excitation

The goal of the tool is to generate the whole spectrum of S-parameters using only two time domain simulations. In order to accomplish this, the chosen excitation must be very broadband. The most broadband signal possible is a single Dirac delta, but that is not advisable from a numerical standpoint.



(a) Time domain (b) Frequency domain

Figure 3.3: Gaussian excitation

The very sharp increase in time of the delta function has implications for the accuracy and stability of simulation being run.

A good compromise between being broadband and being numerically favorable is a Gaussian pulse. The Gaussian pulse takes the form of the commonly known Gaussian function, shown in Equation 3.9 [10]. The spectral composition of a Gaussian is another Gaussian, and it is very flat with a highly controllable roll-off point. The frequency domain representation of Equation 3.9 is given by Equation 3.10. Clearly, the choice of τ, affects the pulse width of $f\left(t\right)$ and therefore the frequency composition $f\left(\omega\right)$. By compressing the pulse width, the spectrum will necessarily broaden. An example of a Gaussian pulse and the resulting spectrum generated from an FFT are given in Figure 3.3.

$$f\left(t\right) = e^{-\frac{t^2}{2\tau^2}} \tag{3.9}$$

$$f\left(\omega\right) = \sqrt{2\pi}\tau e^{-\frac{(\tau\omega)^2}{2}} \tag{3.10}$$

There are several other types of excitations that can be used, depending on the simulation needs. A differentiated Gaussian can be used if your simulation cannot contain a DC component in the stimulus. The general form of the differentiated Gaussian is given in Equation 3.11, and its frequency domain expression is given in Equation 3.12.

$$f\left(t\right) = -\frac{t}{\tau} e^{-\frac{t^2}{2\tau^2}} \tag{3.11}$$

$$f\left(\omega\right) = j\omega\sqrt{2\pi}\tau^2 e^{-\frac{(\tau\omega)^2}{2}} \tag{3.12}$$

Additionally, a modulated Gaussian can be used to achieve a bandpass Gaussian spectrum centered around a certain frequency ω_0. The time and frequency domain functions are shown in Equations 3.13 and 3.14 [10].

$$f\left(t\right) = e^{-\frac{t^2}{2\tau^2}} \sin\left(\omega_0 t\right) \tag{3.13}$$

$$f\left(\omega\right) = -j\sqrt{\frac{\pi}{2}}\tau \left[e^{-\frac{(\tau(\omega-\omega_0))^2}{2}} - e^{-\frac{(\tau(\omega+\omega_0))^2}{2}}\right] \tag{3.14}$$

In the experiments shown in this work, only the Gaussian pulse is used. This excitation was chosen over the alternative excitations listed above due to its wideband nature. The entire spectrum of S-parameters is required, and none of the circuits demonstrated here had any problems operating at DC.

3.2 Implementation

Now that the theory is established, the implementation of the tool itself can be discussed. The basic steps are as follows: define input and output ports, append 50 Ω resistors to ground at those nodes, run a LIM simulation with the excitation at the input port, record the input and output port voltages and currents, compute time domain root power waves, convert root power waves to the frequency domain, and finally take appropriate ratios to determine S-parameters.

In LIM, there are two types of sources, voltage sources in branches between nodes and current sources from ground to a node. Using a current source for the excitation is simpler, even though theoretically the excitation is generally considered to come from a voltage source. A Norton equivalent resistance and a current source can achieve the same effect as a voltage source.

In a laboratory measurement of S-parameters, the ports of the DUT must be matched to the reference impedance of the system. The same is true for

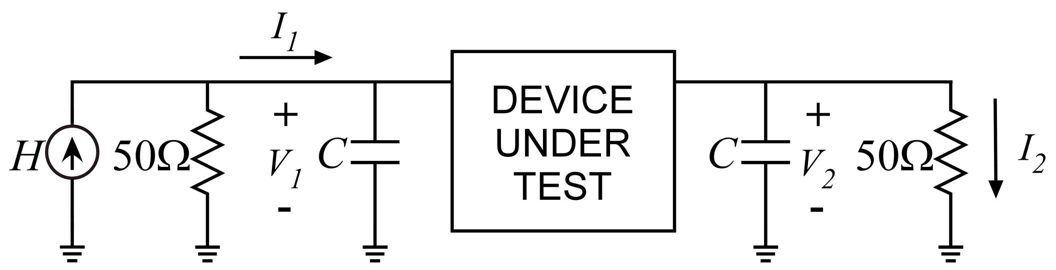


Figure 3.4: Example circuit diagram with V_1, V_2, I_1, and I_2 shown

the time domain simulation being run here. Before beginning the simulation, 50 Ω resistances are added between the input port and ground and the output port node and ground. It is worth noting that in the LIM circuit format, the resistances between a node and ground are actually considered as conductances. Care must be taken when modifying the circuit so as to incorporate a conductance of $(50\ \Omega)^{-1}$, or $0.02\ \Omega^{-1}$ and not $50\ \Omega^{-1}$.

Following the small amount of setup that must be done, the LIM simulation is run. The voltages and currents at the two ports of the network are saved at each time step during the simulation. V_1 and V_2 are simply the node voltages at the input and output ports, respectively. Finding I_1 and I_2 is slightly more involved. These currents the currents flowing into the network from both ports. For I_1, the excitation current is used, but with the current through the 50 Ω resistor to ground subtracted, because that current is not going into the circuit. I_2 is the current going into the load resistor, and can be calculated simply by dividing the output node voltage by 50 Ω. These four values are shown clearly in Figure 3.4.

The input and output voltages and currents are used in Equations 3.1, 3.3, and 3.4 to find time domain root power waves. It makes no sense to solve for a_2 in this case, because the excitation is at port 1, and only S_{11} and S_{21} are calculated from this simulation. Another simulation must be run with the excitation on port 2 to find S_{12} and S_{22}.

An FFT is run on each of the root power waves, yielding a complex frequency domain representation of those signals. Zero padding can be used to increase the frequency resolution of the FFT by decreasing frequency bin size. In practice, it was found zero padding the time domain data up to 2^{16} points or higher yielded good results. Additionally, the exact amount of zero padding can be specified to yield a specific number of points in the frequency range of interest. This, however, creates a large number of extra

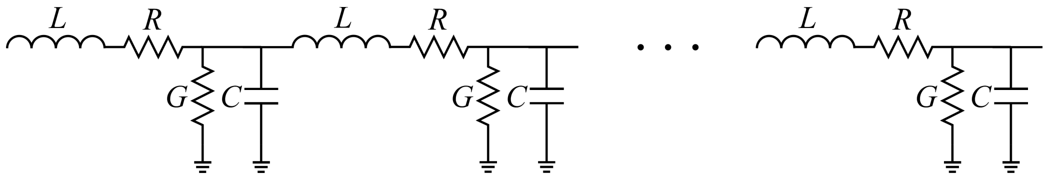


Figure 3.5: Transmission line lumped element model

points beyond the maximum frequency of interest. These values should be discarded, because they waste memory and are often inaccurate. Dividing the appropriate frequency domain root power waves yields the S-parameters, shown in Equations 3.5 and 3.6.

It is important to note that this method, in its current form, works only for linear circuits. There is no way to track conversions of input frequencies to different output frequencies, since there is only a single input excitation containing every frequency.

3.3 Validation

Validating the tool was one of the most critical steps in the whole process. Ensuring accuracy in the tool is of paramount importance, because otherwise there is no use for it. To prove accuracy, both the time domain and frequency domain results from LIM and the tool are compared to results generated using a well-known commercial circuit simulator, ADS. In this section, two passive lumped element circuits, including a transmission line and a power delivery network, are simulated, and the results are carefully compared against solutions derived analytically or generated from a commercial simulator. Additionally, the variations in results obtained from different LIM formulations and the effects of different simulation run times are studied.

3.3.1 Comparison with Analytical Solution

A good way of testing is to simulate a circuit that has a known, analytical solution. In this case, a lumped element transmission line was simulated, whose model is shown in Figure 3.5. Although this model is only an approximation for a transmission line, the circuit that was simulated had 500 elements, making the agreement quite close. The characteristic impedance of the transmission line, Z_c, is 50 Ω, the propagation velocity, v_p, is $3 * 10^8$ $\frac{m}{s}$, and the length, d, is 1 m.

From Equations 3.15 and 3.16, the per unit length inductance $\mathcal{L}$ and capacitance $\mathcal{C}$ can be found [11]. Multiplying these by transmission line length and dividing by the number of elements yields the elemental inductance L and C, which are included in Figure 3.5.

$$v_p = \frac{1}{\sqrt{\mathcal{L}\mathcal{C}}} \tag{3.15}$$

$$Z_c = \sqrt{\frac{\mathcal{L}}{\mathcal{C}}} \tag{3.16}$$

The S-parameters of a transmission line are given by the Equations 3.17 and 3.18 [12].

$$S_{11} = \frac{\left(1 - X^2\right)\Gamma}{1 - \Gamma^2 X^2} \tag{3.17}$$

$$S_{21} = \frac{\left(1 - \Gamma^2\right)X}{1 - \Gamma^2 X^2} \tag{3.18}$$

Expressions for Γ and X are given by Equations 3.19 and 3.20.

$$\Gamma = \frac{Z_c - Z_0}{Z_c + Z_0} \tag{3.19}$$

$$X = e^{-\gamma d} \tag{3.20}$$

The propagation constant, γ, is expressed in Equation 3.21.

$$\gamma = \sqrt{\left(R + j\omega L\right)\left(G + j\omega C\right)} \tag{3.21}$$

To simplify things, the transmission line was considered to be very low loss, so the propagation constant becomes $\gamma = j\omega\sqrt{LC} = \frac{j\omega}{v_p}$. Additionally, by setting $Z_c = Z_0 = 50\ \Omega$, the expressions for S_{11} and S_{21} simplify into $S_{11} = 0$ and $S_{21} = e^{-\frac{j\omega d}{v_p}}$. Figure 3.6 compares the theoretical S_{21} phase with that given by the tool. Agreement between the two is extremely good, inspiring confidence in the tool's accuracy. Only the phase of S_{21} is shown because none of the other S-parameters are interesting, as S_{11} is 0, and the magnitude of S_{21} is 1.

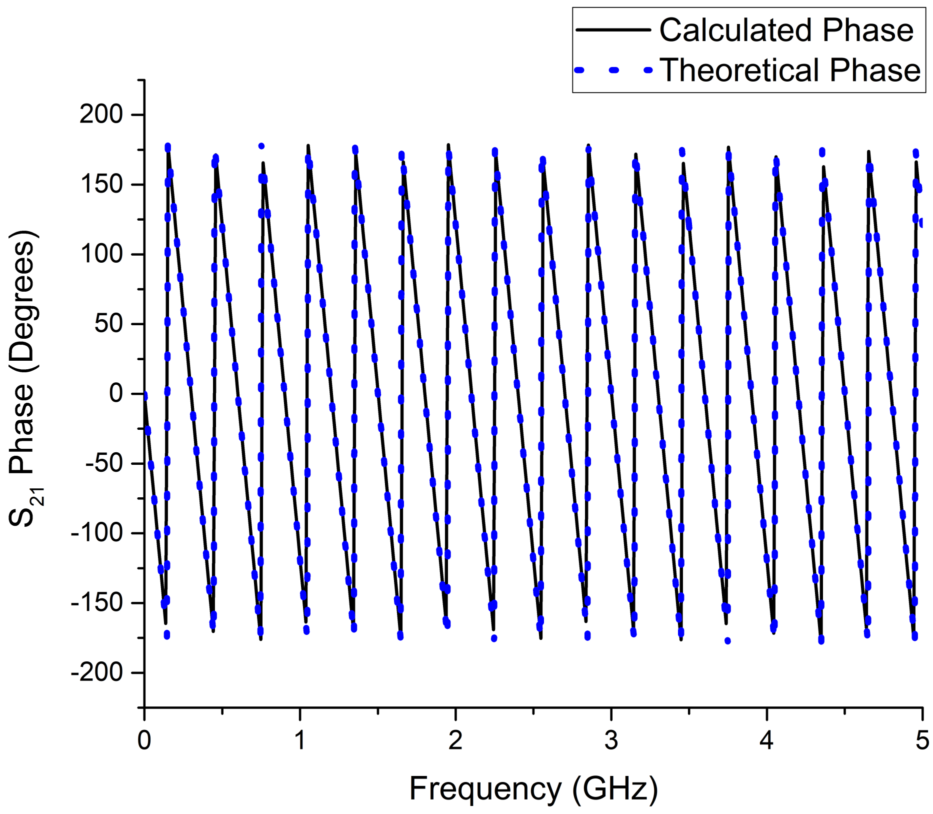


Figure 3.6: Transmission line S_{21} phase comparison

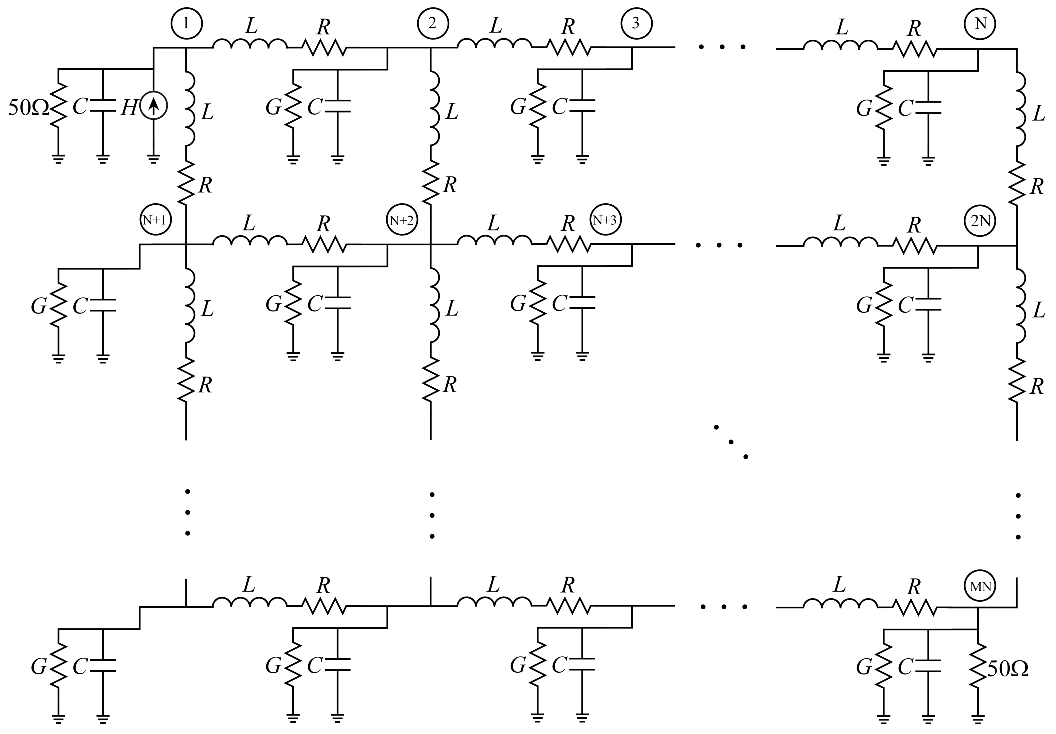


Figure 3.7: Diagram of M×N PDN circuit with excitation at port 1

3.3.2 Comparison with ADS

In this section, comparisons between the tool and commercial simulators are made. Here a lumped element model of a PDN is being simulated. More information about PDNs and this particular model is provided in Chapter 4, but for this section, the origins of the circuit are of little relevance. The chosen circuit is a 40x40 node box spring circuit, shown in Figure 3.7. The input port (port 1) is located at the top left of the circuit (node one), while the output port (port 2) is located at the bottom right of the circuit (node 1600). The resistance and inductance between nodes are $R = 130.0$ mΩ and $L = 1.885$ nH, respectively. The conductance and capacitance to ground at each node are $G = 51.00$ μS and $C = 162.3$ fF, respectively. It is a rather large and complex circuit, so it is a good case for verification against a commercial tool. Advanced Design System (ADS) from Keysight Technologies is the chosen simulation tool, because it is capable of both time domain and S-parameter simulations, and it is common and trusted throughout the industry.

Figure 3.8 compares the time domain voltage measured at port 2, as generated by ADS and LIM. This figure shows excellent correlation between the results from LIM and ADS. Confirming accurate LIM results was a crucial step

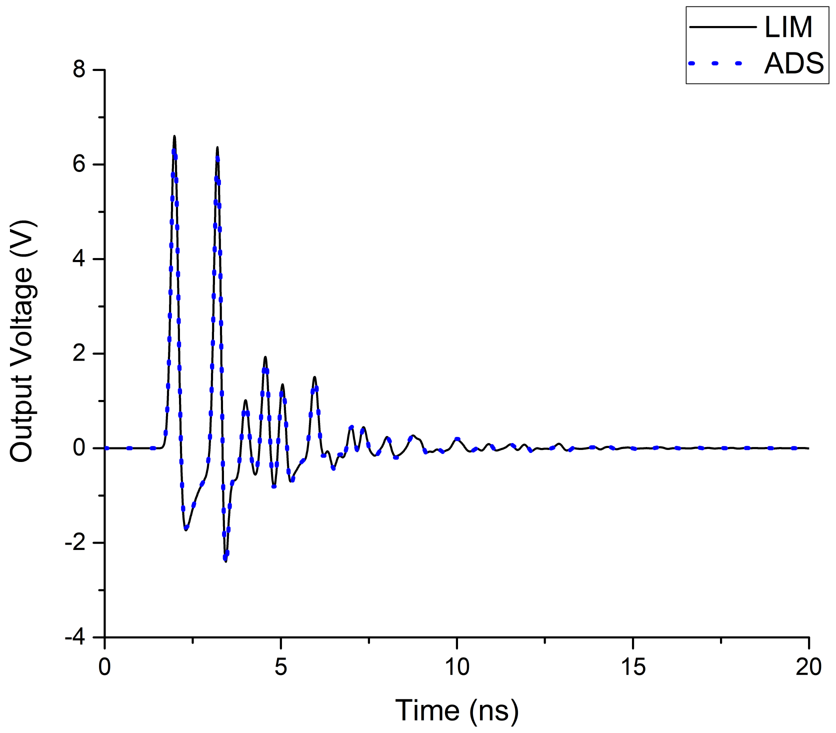


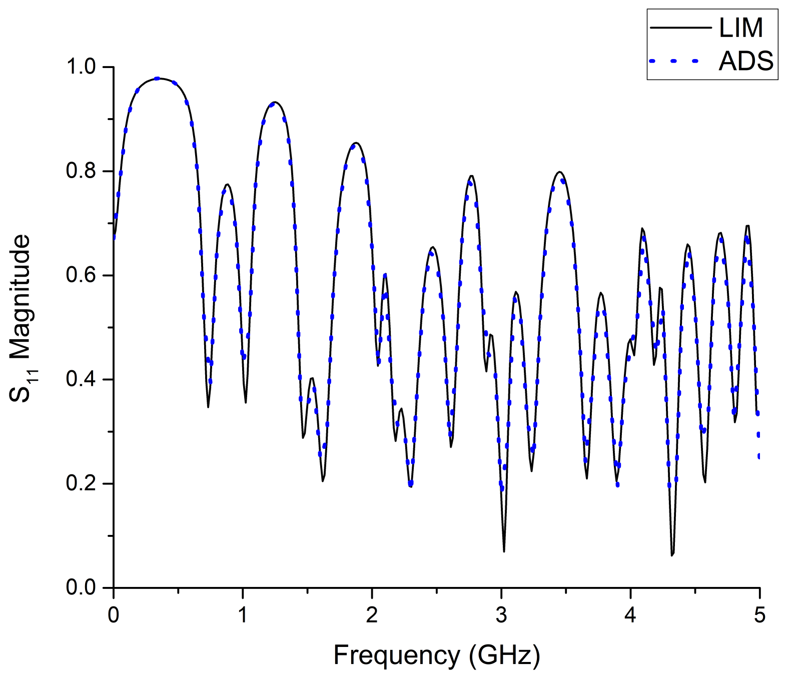
Figure 3.8: Output voltage comparison with ADS

in the process, because the accurate calculation of the circuits S-parameters is contingent upon an accurate time domain simulation. Following the time domain simulation, the S-parameters are calculated using the input and output voltages and currents. The S-parameters generated using the tool and those generated by ADS are given in Figures 3.9 and 3.10. It is clear that these agree quite closely. In this case, only S_{11} and S_{21} are given because the network is reciprocal and therefore S_{22} and S_{12} are redundant.

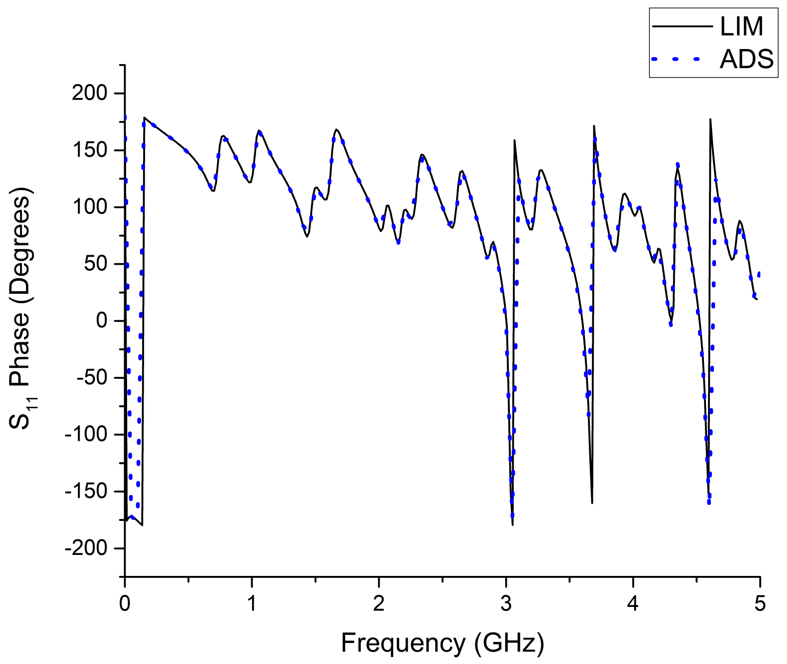
3.3.3 Variations Between LIM Formulations

LIM, being a finite difference algorithm, suffers from the same problems as any other finite difference equation. In this section, variations between results of explicit, implicit, and semi-implicit LIM formulations are compared, and the effects of these differences on the S-parameter calculations are analyzed.

Figure 3.11 shows the output voltages calculated by LIM's three basic formulations with two different time steps. The main spike is emphasized to show the minute differences between the results from the separate schemes. It is evident that the smaller time step yields more consistent results among the

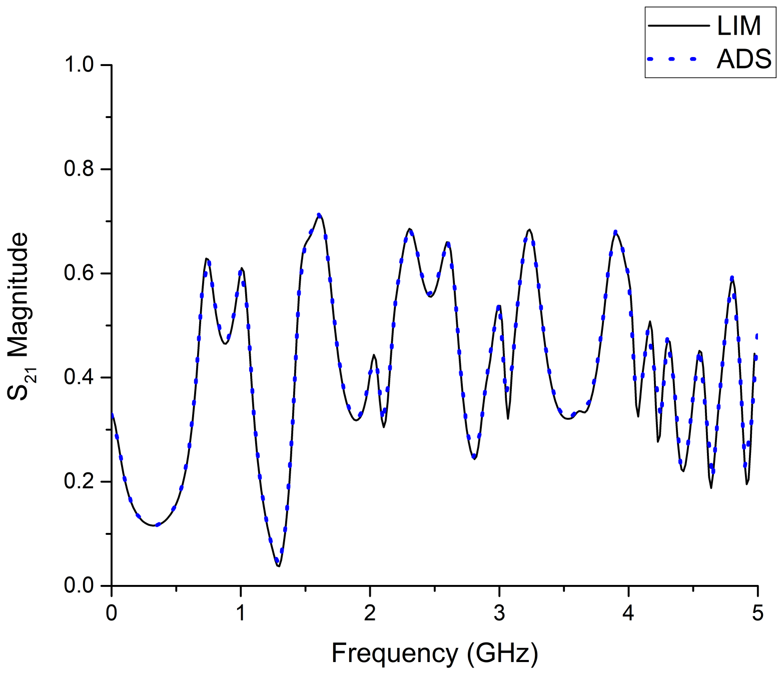


(a) Magnitude

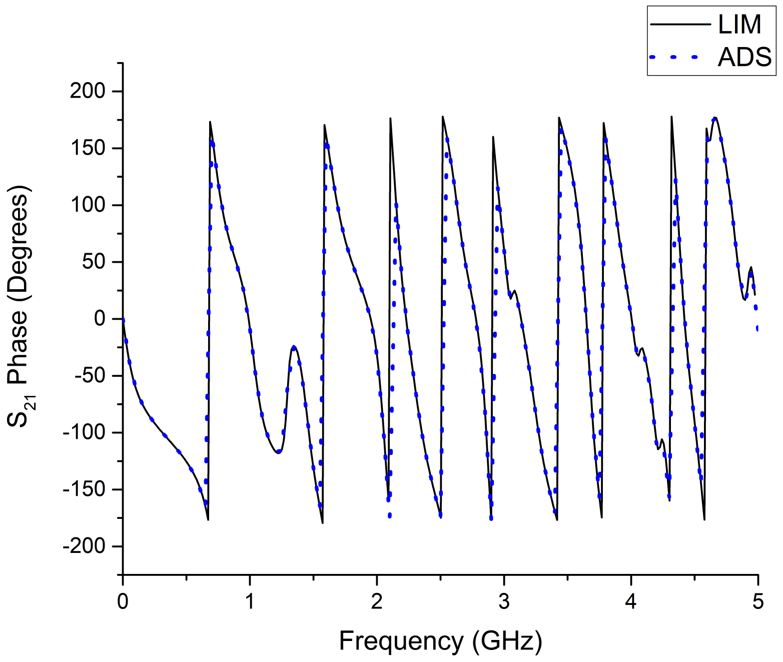


(b) Phase

Figure 3.9: S_{11} comparison with ADS



(a) Magnitude



(b) Phase

Figure 3.10: S_{21} comparison with ADS

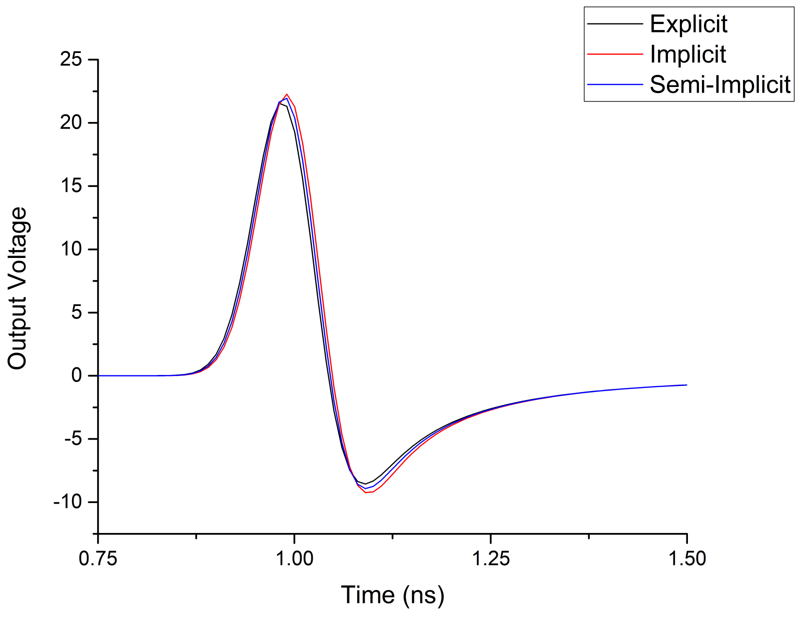
three formulations. This makes sense because as the time step approaches zero, the three finite difference schemes will converge to the same result. From Figure 3.12, it is clear that these minor variations in the time domain simulation results have serious consequences in the frequency domain. While the time step is certainly small enough in either case to ensure stability, it may not be small enough to ensure a desired level of accuracy in the resulting S-parameters. This is something the user must be aware of.

3.3.4 Simulation Run Time Considerations

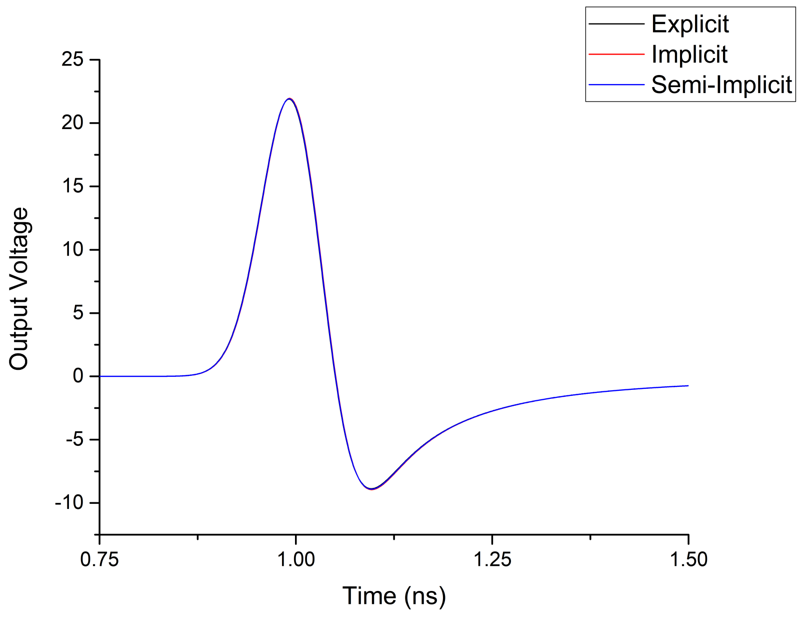
The duration of the simulation is of critical importance when S-parameters are calculated by means of an FFT. The findings indicate that the effects of the excitation must be completely extinguished before the completion of the time domain simulation. Early termination of the LIM simulation results in a loss of data and considerable effects on the resulting S-parameters.

To investigate this, three simulations were run using the same time step and explicit LIM formulation, but with three different run times: 5, 10, and 20 ns. Figure 3.13 highlights the truncation and subsequent loss of data from the three trials. Only the output voltage is shown, but the same effect is seen in the output current and input voltage and current. The three datasets are overlayed and are identical except for the simulation end time.

Shown in Figures 3.14(a) and 3.14(b) are comparisons between the S-parameters calculated using the data generated from simulations using the three different run times. From the previous section, it is shown that the 20 ns simulation yields accurate S-parameter results. Comparing the other two simulations to this one, we can see that the results become corrupted. Moving from 20 ns to 10 ns simulation time only results in a small loss in accuracy, largely due to the fact that the signal has been mostly damped out by that point in time. However, going from 10 ns to 5 ns simulation time yields very bad results. At some points, the supposedly passive circuit becomes active and has an $S_{11} > 1$. It is expected to lose accuracy here, because a lot of information is lost, which can be seen in red in Figure 3.13. These effects will vary based on the circuit in question, and the user must be vigilant to ensure the proper simulation time is set.

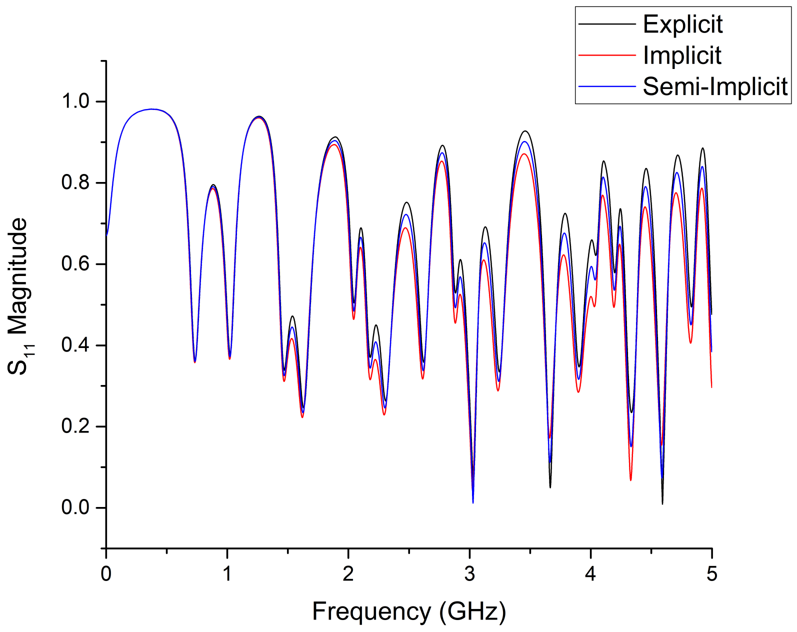


(a) $\Delta t = 10$ ps

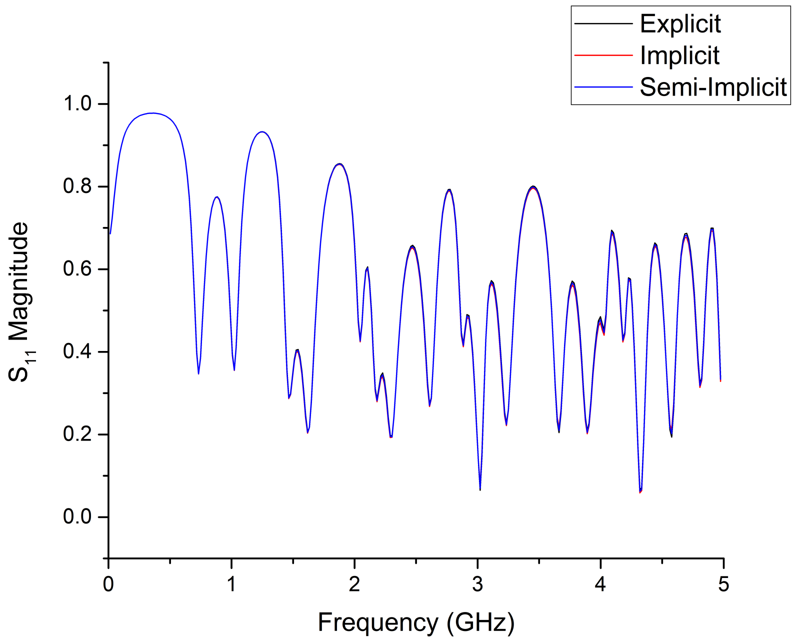


(b) $\Delta t = 1$ ps

Figure 3.11: Output voltage comparison between LIM formulations



(a) $\Delta t = 10$ ps



(b) $\Delta t = 1$ ps

Figure 3.12: S_{11} magnitude comparison between LIM formulations

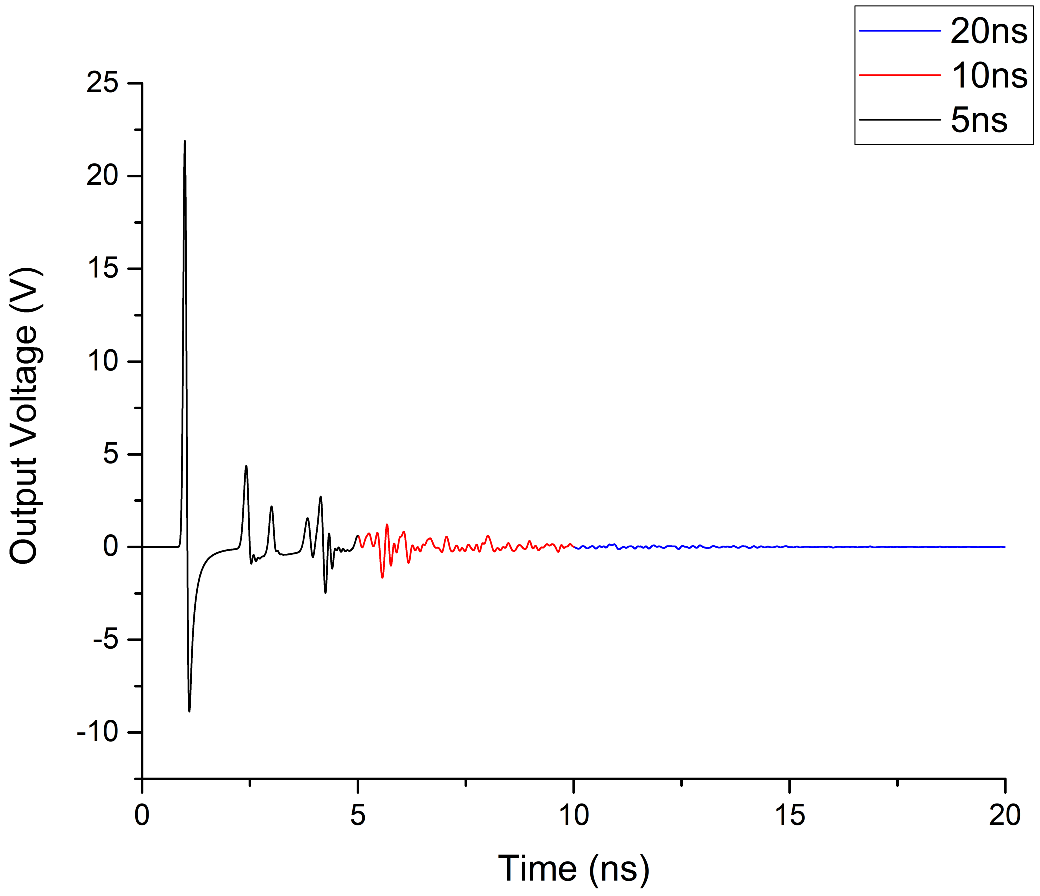
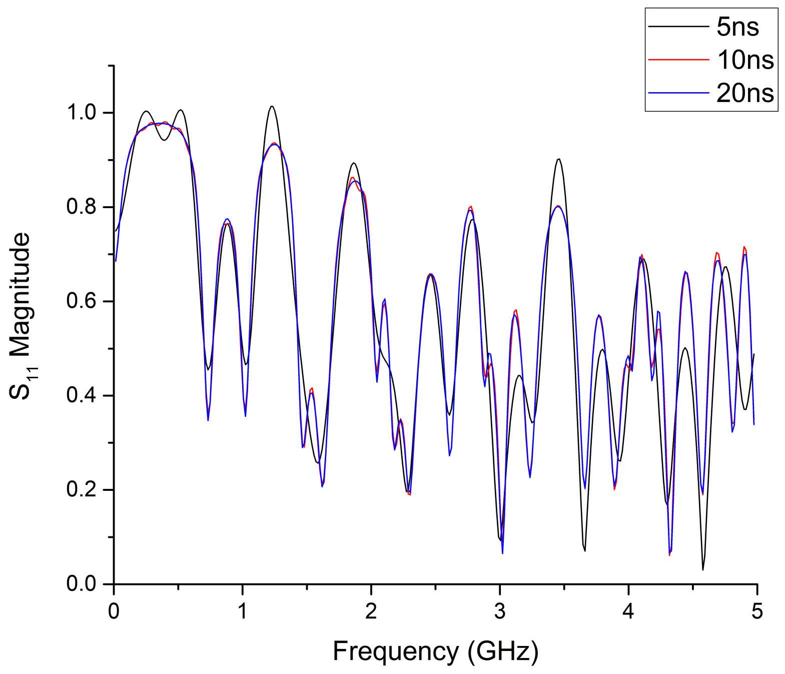
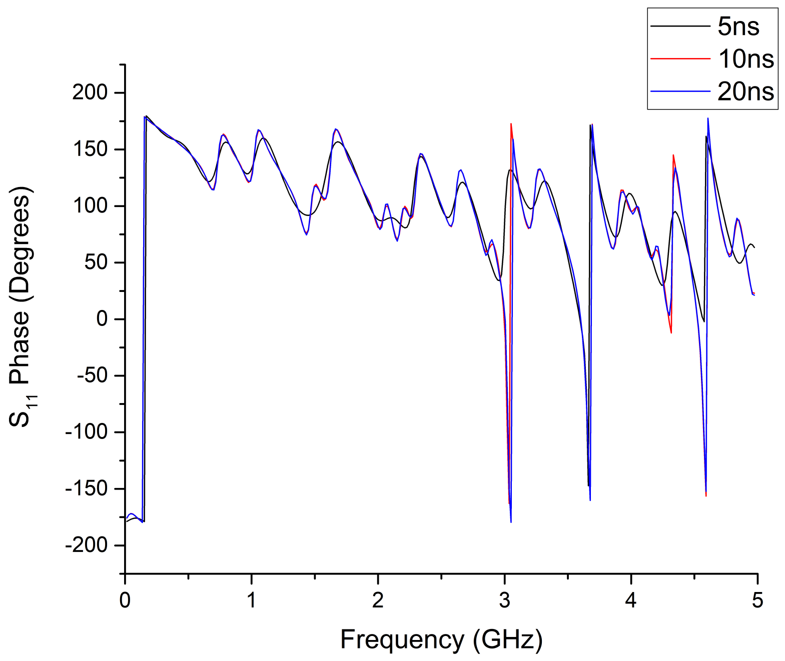


Figure 3.13: Output voltage



(a) S_{11} magnitude comparison



(b) S_{11} phase comparison

Figure 3.14: S_{11} comparison between simulation run times

CHAPTER 4

APPLICATIONS

Once validation of this tool was complete, it was applied to the problem of simulating a structure with some randomness in its material properties and physical dimensions. In this particular case, the structure of interest was a power delivery network (PDN). A basic PDN consists of a conducting power plane separated from a conducting ground plane by a dielectric. Creating a lumped element model for this circuit is necessary for simulation using LIM, due to LIM's circuit based nature. Following the modeling of the structure, the effects of the inherent uncertainties in the DUT are characterized first using Monte Carlo simulations and then using stochastic collocation [13]. This chapter discusses PDN modeling, stochastic characterization, the specific scenario that is simulated, and the results generated using the tool. This problem is also discussed in [4, 5].

4.1 Power Plane Modeling

A PDN can be thought of as a two-dimensional transmission line, and it is modeled as such [14]. When modeled as a lumped element circuit, the PDN is broken up into a number of unit cells. Figure 4.1 shows a diagram of our example PDN with important dimensions indicated. The variables$w$, h, and t are unit cell side length, plate separation, and plate thickness, respectively, and ϵ_r is the relative permittivity of the dielectric substrate.

The circuit model of a single unit cell is given in Figure 4.2. Each unit cell has a conductance and capacitance to ground and a resistance and inductance to each adjacent unit cell. The unit cells are arranged in a grid, with each non-edge cell having four neighbors above, below, left, and right of it. This forms a "box spring" type circuit, and is shown in Figure 3.7. This format makes it ideal for usage with LIM, because the circuit does not need to be modified by the addition of fictitious elements.

Given in 4.1 and 4.2 are the expressions for the unit cell capacitance and conductance, respectively. The equation for capacitance is intuitive, as it is the same as the capacitance of a parallel plate capacitor. The conductance is a frequency-dependent term and is related to the loss tangent of the dielectric.

$$C = \epsilon_0 \epsilon_r \frac{w^2}{h} \tag{4.1}$$

$$G = \omega C \tan \delta \tag{4.2}$$

Internodal inductances and resistances are calculated by Equations 4.3 and 4.4, respectively.

$$L = \mu_0 h \tag{4.3}$$

$$R = \frac{1}{\sigma t} \tag{4.4}$$

Depending on the granularity required for the simulation, the PDN can be broken up using various sized unit cells. In general, when using finite difference formulations such as the FDTD, it is recommended that the granularity of the structure be equal to or greater than 20 cells per wavelength. Increasing the number of cells per wavelength serves to reduce the numerical dispersion error [7].

4.2 Stochastic Collocation

Stochastic collocation is a method of characterizing the effects of uncertainty [13]. Compared to the Monte Carlo method, stochastic collocation is a much more sophisticated approach. A Monte Carlo simulation relies on brute force and a massive number of trials. Stochastic collocation, on the other hand, requires fewer trials and approximates the statistical moments of our output.

A major benefit of stochastic collocation is that it is non-intrusive, meaning that the underlying LIM and S-parameter equations may be preserved [15]. The core principle in stochastic collocation is the creation of a sparse grid. The sparse grid is a set of points with the same number of dimensions as the

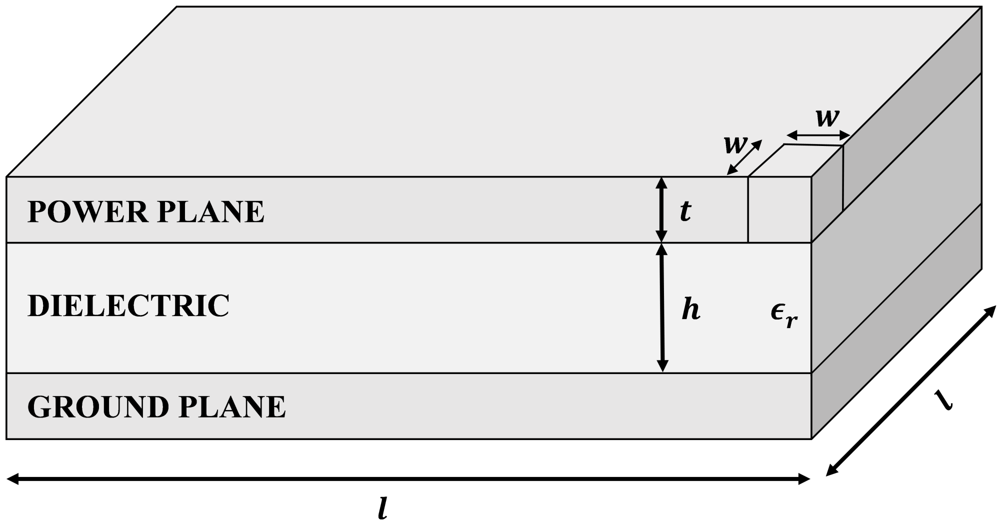


Figure 4.1: Power plane with parameters

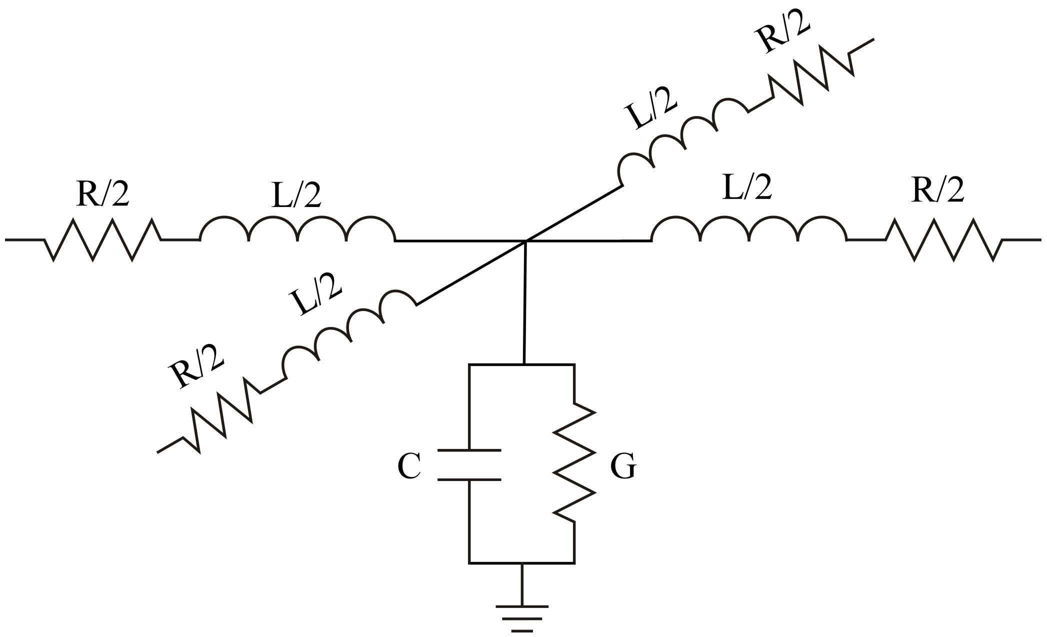


Figure 4.2: PDN unit cell

Table 4.1: Parameter values

Parameter	Mean Value	Standard Deviation
w	2.5 mm	-
l	10 cm	-
h	1.50 mm	150μm
t	150μm	15μm
ϵ_r	4.4	0.44
σ	58 MS$\cdot$m^{-1}	-

number of random variables in the problem. Each grid point consists of a tuple of values of the random variables at that given point. Depending on the level of accuracy that is required, sparse grids of different densities are sampled.

The simulation results at each of the sparse grid points is then used to generate an interpolating function that maps the random variables to the simulation output. This interpolant will approximate the results of the LIM and subsequent S-parameter simulation. An external tool called TASMANIAN [16] is used to generate the sparse grid and calculate the interpolant.

4.3 Example Problem

The PDN that is being simulated has three random variables: the power plane thickness (t), dielectric thickness (h), and dielectric constant (ϵ_r). As a reminder, the PDN is shown in Figure 4.1. Each of these random variables is normally distributed with a standard deviation (σ) of 10% of their mean values and with a cutoff after 3σ. A change in any of these parameters will result in a change in the overall S-parameters of the circuit. Table 4.1 gives the mean values of all of the parameters and the standard deviations of the random parameters.

First, a basic Monte Carlo simulation is run, using 10,000 trials. At each trial, a random sample of t, h, and ϵ_r is taken. These values are then converted into the unit cell R, L, G, and C. Due to the limitations of the tool, the frequency dependence of the dielectric conductance cannot be accounted for. For this simulation, the value of the conductance at the center frequency of the simulation was used. From there, the LIM circuit is constructed and the time domain signal is run. Using the tool, the S-parameters are extracted from the time domain data and saved. This Monte Carlo simulation will be used as a baseline for experimenting with the stochastic collocation method.

Using the Clenshaw-Curtis rule, the sparse grid is made for levels 3, 4, and 5. Each of these three levels has a different number of points, resulting in different degrees of accuracy in their ultimate results. A benefit of the Clenshaw-Curtis sparse grid is that higher level sparse grids contain points from lower level sparse grids, eliminating the need to resimulate those points if multiple levels are used.

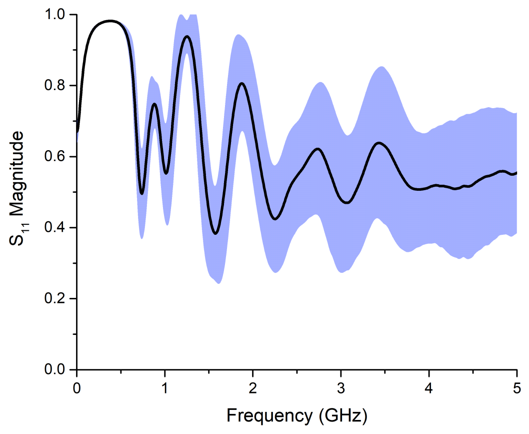
4.4 Results

The Monte Carlo simulation yielded results shown in Figure 4.3. Each of the four plots shows the mean value of the S-parameter with a black line and the 1 standard deviation range with the blue shaded area. As the frequency increases, the uncertainty in the S-parameters increases, which is understandable. The effects of the reactive elements become more pronounced as the frequency is increased, and therefore their impact on the S-parameters become more pronounced.

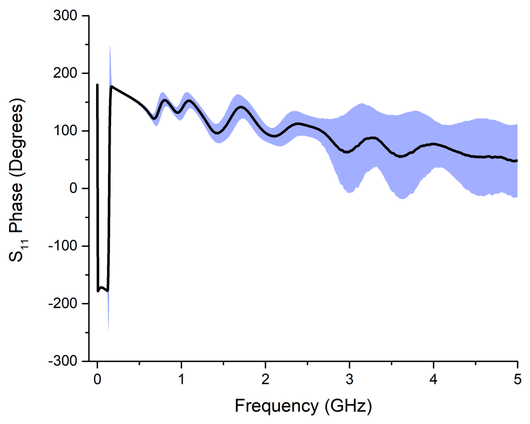
Taking the Monte Carlo simulation to be the baseline, the results generated from the three levels of sparse grid are compared. Figure 4.4 overlays the plot of the standard deviation of the magnitudes and phases of S_{11} and S_{21}. Similarly, Figure 4.5 shows the plots of the mean values of the magnitudes and phases of S_{11} and S_{21}. It is apparent that as the level increases, the approximation of the moments becomes more accurate.

The number of trials run and simulation time is given in Table 4.2. The advantages of using stochastic collocation over the a simple Monte Carlo are clear in Table 4.3. Level 5, the most accurate and most computationally expensive of the three levels used, shows a 95% reduction in run time to achieve results with acceptable error. The level can be increased to reduce error as desired.

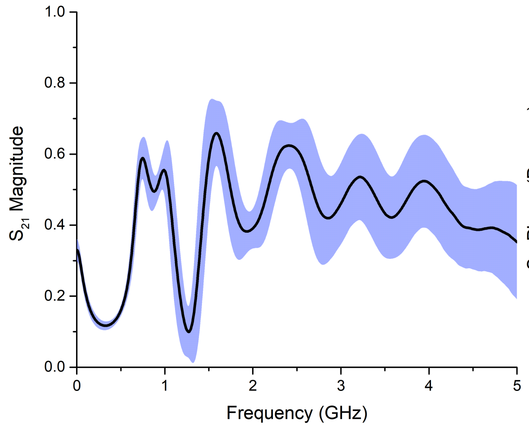
This much improved computation time does come with a price. Table 4.4 shows the mean square error between the means of the S-parameter distributions obtained through the Monte Carlo method and those obtained through the stochastic collocation method. Likewise, Table 4.5 gives the mean square error between the standard deviations of the respective distributions [4, 5].



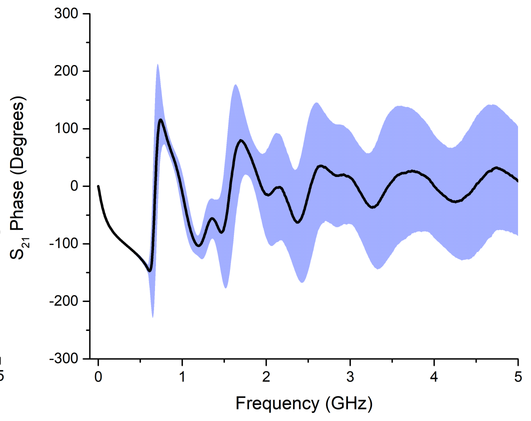
(a) S_{11} Magnitude



(b) S_{11} Phase



(c) S_{21} Magnitude



(d) S_{21} Phase

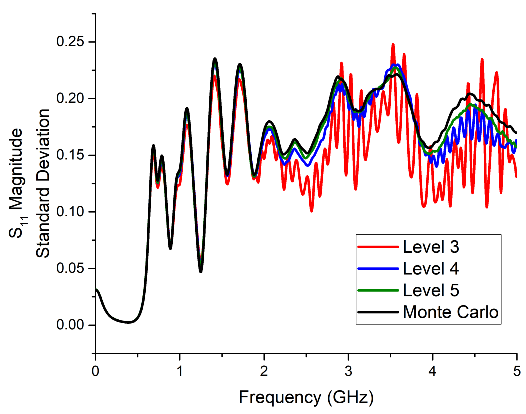
Figure 4.3: S-Parameters mean value $\pm\sigma$

Table 4.2: Times for simulating sparse grid points

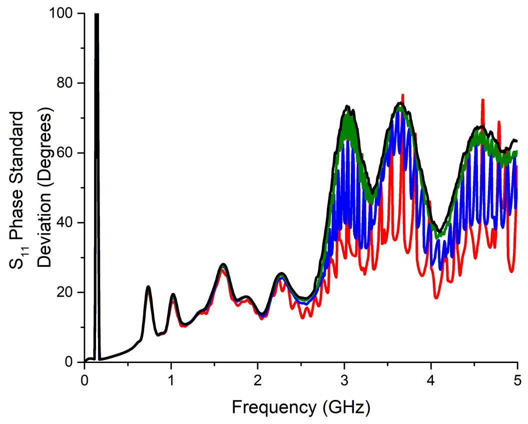
Level	Trials	Time (s)
5	441	198.45
4	117	52.65
3	69	31.05
Monte Carlo	10000	4502

Table 4.3: Times for running Monte Carlo on interpolant

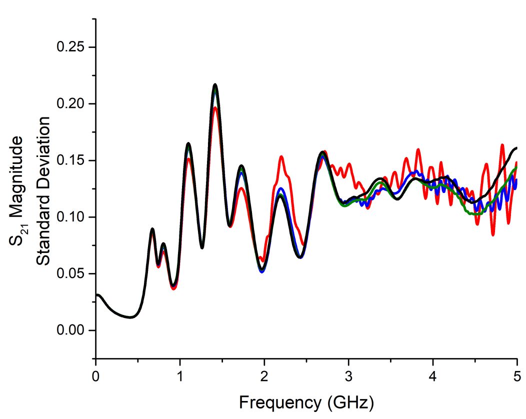
Level	Interpolant Time (s)	Total Time (s)	Time Reduction (%)
5	39	237.45	95
4	31	83.65	98
3	26	57.05	99



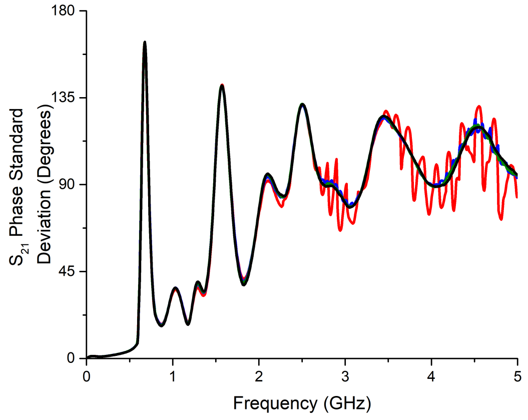
(a) S_{11} magnitude standard deviation comparison



(b) S_{11} phase standard deviation comparison

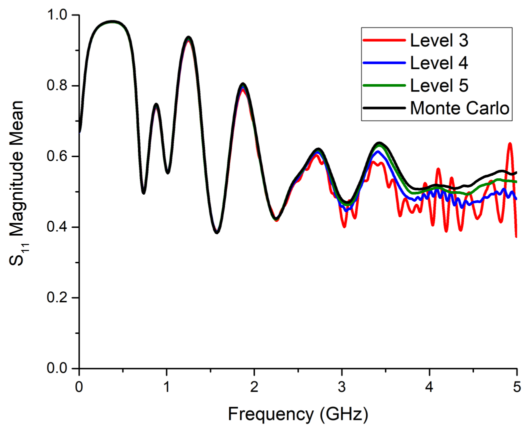


(c) S_{21} magnitude standard deviation comparison

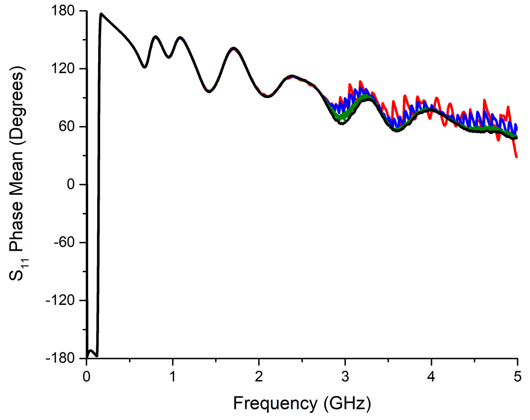


(d) S_{21} phase standard deviation comparison

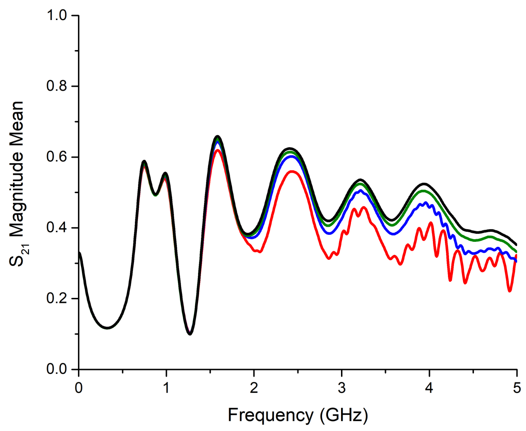
Figure 4.4: S-parameter distribution standard deviations



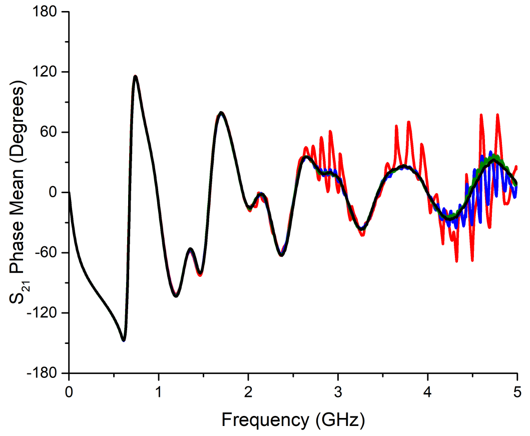
(a) S_{11} magnitude mean comparison



(b) S_{11} phase mean comparison



(c) S_{21} magnitude mean comparison



(d) S_{21} phase mean comparison

Figure 4.5: S-parameter distribution means

Table 4.4: Mean squared error of means

Level	S_{11} Mag	S_{11} Phase	S_{21} Mag	S_{21} Phase
5	1.07E-04	4.66E+00	1.62E-04	3.63E+00
4	6.88E-04	4.26E+01	1.15E-03	4.29E+01
3	1.95E-03	8.26E+01	6.38E-03	2.04E+02

Table 4.5: Mean squared error of standard deviations

Level	S_{11} Mag	S_{11} Phase	S_{21} Mag	S_{21} Phase
5	2.53E-05	6.25E+00	3.27E-05	8.49E-01
4	1.27E-04	1.15E+02	6.17E-05	2.04E+00
3	9.04E-04	3.02E+02	2.29E-04	5.56E+01

CHAPTER 5

CONCLUSION AND FUTURE WORK

This thesis has presented an approach to calculate the S-parameters of a lumped element circuit. The background and importance of the problem was given, followed by an explanation of the circuit simulator upon which this tool is built. Following a theoretical explanation of S-parameters and the FFT, an explanation of the S-parameter extraction procedure was given. Finally, this tool was applied to a problem involving the simulation of a random structure. It was shown that this tool is capable of accurately computing the S-parameters of a LIM circuit.

In the future, this approach can be applied to simulations of more complicated structures. An extremely simple addition to this is the extension to N-port networks, requiring only the addition of extra matched terminations, two additional LIM simulations per added port, and a total of N(1+N) FFT computations. Additionally, a major improvement could be made by introducing a capability of handling non-linear circuits. The current method of utilizing a broadband stimulus and the FFT would likely have to be rethought and reworked in order to accommodate non-linear circuits, due to the linear nature of S-parameters and the FFT.

REFERENCES

[1] X. Chen, M. Qiu, J. Schutt-Aine, and A. Cangellaris, "Stochastic LIM for transient simulation of circuits with uncertainties," in *Electrical Performance of Electronic Packaging and Systems (EPEPS), 2014 IEEE 23rd Conference on*, Oct 2014, pp. 29–32.

[2] Z. Deng and J. E. Schutt-Aine, "Stability analysis of latency insertion method (LIM)," in *Electrical Performance of Electronic Packaging - 2004*, Oct 2004, pp. 167–170.

[3] P. Goh, "A fast multi-purpose circuit simulator using the latency insertion method," PhD dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, 2012.

[4] R. F. Kummerer, X. Chen, J. E. Schutt-Aine, and A. C. Cangellaris, "FFT-based macromodeling of power delivery network with uncertainties using latency insertion method and stochastic collocation," in *2017 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, Dec 2017, pp. 1–3.

[5] R. F. Kummerer, X. Chen, J. E. Schutt-Aine, and A. C. Cangellaris, "FFT-based macromodeling of power delivery network with uncertainties using latency insertion method," in *2017 IEEE 21st Workshop on Signal and Power Integrity (SPI)*, 2017.

[6] J. E. Schutt-Aine, "Latency insertion method (LIM) for the fast transient simulation of large networks," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 48, no. 1, pp. 81–89, Jan 2001.

[7] J. Jin, *Theory and Computation of Electromagnetic Fields.* Hoboken, N.J.: Wiley, 2010.

[8] M. B. Steer, *Microwave and RF Design: A Systems Approach.* Raleigh, NC: SciTech Pub, 2010.

[9] D. Manolakis and V. Ingle, *Applied Digital Signal Processing: Theory and Practice.* Cambridge University Press Textbooks, 2011. [Online]. Available: https://books.google.com/books?id=vvMtvAEACAAJ

[10] J. Jin, "Lecture viewgraphs set 2," lecture slides, University of Illinois, 2018.

[11] D. Pozar, *Microwave Engineering*. Wiley, 2004. [Online]. Available: https://books.google.com/books?id=4wzpQwAACAAJ

[12] J. E. Schutt-Aine, "TL characterization," lecture slides, University of Illinois, 2018.

[13] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 357–78, 2005.

[14] M. Swaminathan and A. E. Engin, *Power Integrity Modelling and Design for Semiconductors and Systems*. Upper Saddle River, NJ: Prentice Hall, 2008.

[15] X. Chen, J. E. Schutt-Ain, and A. C. Cangellaris, "Stochastic LIM for transient simulations of printed circuit board transmission lines with uncertainties," in *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*, May 2016, pp. 2297–2304.

[16] M. Stoyanov, "User manual: Tasmanian sparse grids," Oak Ridge National Laboratory, One Bethel Valley Road, Oak Ridge, TN, Tech. Rep. ORNL/TM-2015/596, 2015.