Contents

Chapter 6 — Online Content

Articles
- Mathematical Stability Problems in Modern Non-Linear Simulation Programs by Dr. Ulrich Rohde, N1UL and Rucha Lakhe
- The Dangers of Simple Usage of Microwave Software by Dr. Ulrich Rohde, N1UL and Hans Hartnagel
- Harmonic Balance by Dr. Ulrich Rohde, N1UL, Ajay Poddar, and Matthias Rudolph
- Using Simulation at RF by Dr. Ulrich Rohde, N1UL
- Harmonic Balance by Dr. Ulrich Rohde, N1UL, Ajay Poddar, and Matthias Rudolph

Tools and Data
- SON and data files for the Electromagnetic Analysis of RF Circuits section

Note that the color image files, SON/data files, and the paper by Rohde are in the Chapter 6 Working Files folder “Chapter 6 — Online Content”

Chapter 6

Electronic Design Automation (EDA)

This chapter, originally titled Computer-Aided Design (CAD), provides an overview of common simulation and design tools at the circuit component level and for electromagnetic modeling. These tools enable the hobbyist to harness some of the circuit simulation power employed by professional electronic and RF engineers in the product and system design cycle. These tools sets are known as Electronic Design Automation software.

Jean-Francois Debroux addresses circuit simulation tools. Dr. Ulrich Rohde, N1UL, surveys issues associated with linear and nonlinear RF simulation and contributes three extensive papers with the online content. Jim Rautio, AJ3K, presents an overview of electromagnetic (EM) simulation.

The purpose of this chapter is not to provide detailed instructions for using any particular software package, but to explain the basic operations, limitations and vocabulary for circuit and RF design software. Software to aid design and analysis in specialized areas, for example filter design, switchmode power supplies, transmission lines, and RF power amplifiers, is covered in other chapters.

6.1 Circuit Simulation Overview

Circuit simulation is a branch of a more general computer-based technique: Simulation. If the laws of physics and a technical device or system can be described appropriately for a computer program (a simulator) to process them (that is running a simulation) the result mimics the behavior of the actual system without the need to build it.

Because computers are more and more powerful and simulation programs are more and more efficient, simulation is now widely used in the development of almost every product. There are simulators in fields like mechanics, electronics, and many others. Here, the focus will be placed on circuit simulation.

Circuit simulators have been around for decades. Significant progress has been made. One of the milestones was *SPICE* developed at the University of California at Berkeley in the 1970s. *SPICE* is an acronym standing for Simulation Program with Integrated Circuit Emphasis. Most circuit simulators today still use the *SPICE* syntax and philosophy even though their capabilities and performance have improved by orders of magnitude.

Many circuit simulators today are associated with *schematic capture* programs that allow drawing schematics the way it used to be done by hand. (See the **Construction** chapter's section on PCB Layout for more information on schematic capture.) The simulation *results viewer* allows graphically displaying voltages and currents similar to the way an oscilloscope or a spectrum analyzer does.

6.1.1 The Design Cycle with Simulation

Using simulation has a significant impact on the development cycle. The classic cycle is shown in **Figure 6.1**. The process starts with a specification that defines the goal in terms of functionalities and performances. Generally speaking, functionalities define architecture (the way parts are assembled and connected) while performance parameters (power, speed, etc.) define parts selection and size. After the project sketch is defined is the time to build it which requires buying the parts and assembling everything.

Then comes the test phase, where you compare the prototype to the specification. To determine whether the prototype matches the specifications, you must have the right test equipment. If the prototype performs well, the project is done (hobby projects usually don't go to production), but if something is wrong, a potentially painful and costly phase begins to identify what's wrong, fix the issues, and eventually iterate the process until everything is okay.

If you're using a simulation, then you modify the cycle as in **Figure 6.2**. This cycle is slightly more complex, but it saves time and money. You don't need to buy parts or test equipment during the first phase — the simulator does it all. After you've run the simulation successfully, having addressed any issues flagged during earlier attempts, you are ready to begin

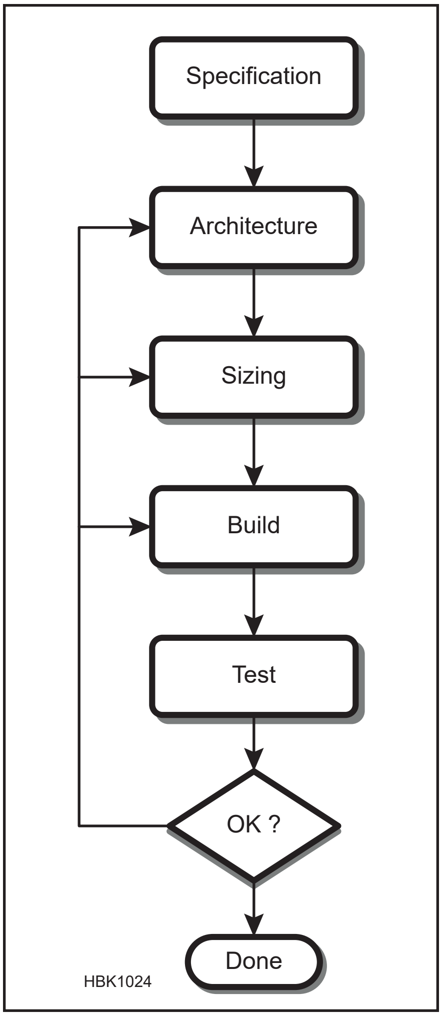


Figure 6.1 — Classic design cycle.

the implementation phase. The simulation should normally reduce the risk of damaged parts and limit the number of iterations you must go through during the implementation.

For interested readers / To go further

en.wikipedia.org/wiki/Computer_simulation

en.wikipedia.org/wiki/SPICE

www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator

6.1.2 Hobby Versus Professional Circuit Simulation Tools

The market for simulation tools has evolved in recent years because systems are getting more and more complex with more and more interactions requiring very complex and expensive software packages that target professionals.

On the other hand, open-source software is now very common and addresses almost every domain. A number of open-source circuit simulators exist with active communities improving them. *Ngspice* is an example. These tools are available under a variety of licensing models (gpl, lgpl, etc...) The same is true for schematic capture, printed circuit board (PCB) design, and IC layout. *Kicad*, *KLayout* and *gEDA* are some examples. These open-source tools may not offer the same performance as professional tools, but they are a good choice for hobbyists.

Another family of circuit simulators has appeared with tools like *LTSPICE*. It was developed by the IC manufacturer Linear Technology (now part of Analog Devices) for users to validate an application using the company's products before buying the components and building a board. These circuit simulators are free but not open-source. They come with a comprehensive library of components and are associated with schematic capture and a waveform viewer for ease of use.

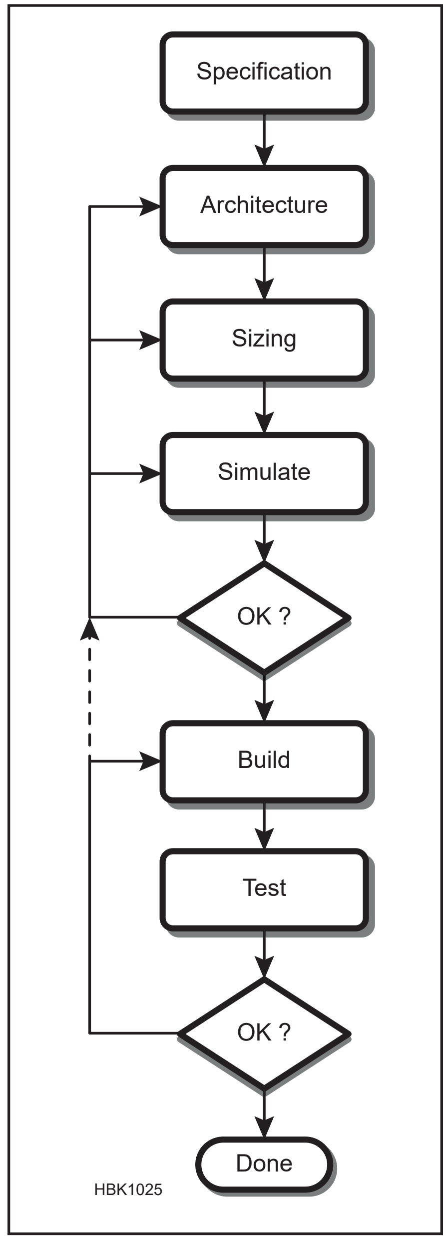


Figure 6.2 — Design cycle with simulation.

6.2 Interests and Limitations of Circuit Simulation

There are four major benefits to simulating a circuit rather than building it:

- It is fast
- It is cost effective
- It gives deep insight on circuit behavior
- It is safer

But there are limitations:

- Only what is modeled can be simulated
- Only what is measured can be monitored
- Only analysis types supported by a given simulator can be performed

It is important to understand these points well so we will focus on them.

Simulation is fast and cost effective: With today's free circuit simulators and the power of available computers, it takes only minutes to draw a simple schematic and simulate it. No need to buy components, no delay to get them, no need for an expensive signal generator — the circuit simulator has all that inside!

Simulation gives deep insight: Once the simulation is done, every voltage and every current in any circuit device can be displayed. This is not so easy to do on a board. And these "measurements" do not add any perturbation to the circuit. Simulation can be done at different temperatures or taking into account component tolerances giving some information on circuit robustness.

Simulation is safer: There is no risk in simulating a high voltage or a high-current circuit. No smoke is produced in case a component is not used properly!

Simulation is model limited: A circuit simulator uses models to describe components. A model is a set of equations with parameters that describes the relationship between voltages and currents.

For instance, a resistor model is defined by

Ohm's law: V = I × R. This model has only one parameter: R, the resistor value. Such a resistor is absolutely stable over temperature since it has no temperature coefficient.

A more realistic resistor would be described by the same Ohm's law, but in addition, R would depend on temperature. For example, R = R0 (1 + TC1 (Temp – T0), where R0 is nominal resistance at a nominal temperature, T0.

Simulation can only show what is described and what is watched: In a circuit simulator, components are connected to *nets*, a connection between components. (The list of all connections is called a *netlist*.) By definition, all components sharing a net are connected by a zero-ohm resistance which does not exist in reality. If the actual connections exhibit some resistance and inductance these must be added explicitly to the model or as separate components. In the same way, if two nets exhibit some parasitic capacitance between each other it must be added explicitly.

Taking into account parasitic elements must be done wisely. (See the **RF Techniques** chapter for more about parasitic components.) There is no need to consider connection resistance in a circuit that draws micro-amps. Similarly, there is no need to take into account parasitic L and C at low frequencies.

Model limitations and parasitic elements are the major sources of difference between simulation and experiment. For example, if a 0.25-W resistor dissipates 2 W in a breadboard circuit, it will probably smoke quickly. But in a simulation, it does not. The user must check dissipated power and eventually use that information to choose the proper resistor size. Some simulators have a feature called "Smoke alarm" that can highlight overloaded components provided that the model includes some voltage/current/power limits.

Simulation is limited to available analysis: Many simulators inherit the analysis types that were defined by *SPICE*. But these do not cover all the fields of circuit analysis. For instance, noise analysis in *SPICE* is limited to small signal ac simulation. So, *SPICE* is not suited to perform a mixer noise analysis since such a circuit's operation is fundamentally non-linear.

6.2.1 Circuit Simulation Process

To simulate a circuit, you need six things:

• A circuit description so that the simulator knows what to simulate

• Models for the components used in the circuit

• One or more simulation commands defining the simulation to be performed

• A circuit simulator

• A computer

• A method

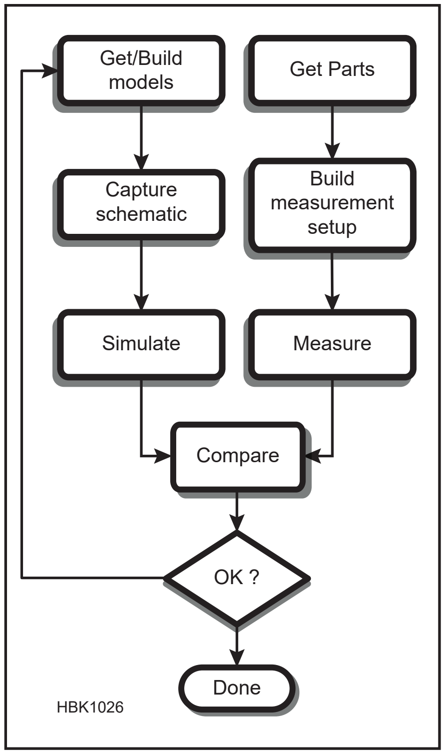


Figure 6.3 — Model validation process.

SIMULATION METHOD

It might seem strange to start with the sixth item, but the five preceding topics are pointless without a method. The method strongly depends on the context and goal.

In the professional world, circuit simulators are embedded in *frameworks* that manage schematic, layout, parasitic extraction, etc. Models are usually validated by dedicated support teams. This organization and these tools are expensive, but as a result, simulation results are reliable, prototypes operate as expected in most cases, and production yield is high and consistent.

For the hobbyist, the situation is very different. Models are supplied without a real description of validity, and parasitic elements are hard to predict, so the prototype may behave somewhat differently from the simulation. A model validation phase shown in **Figure 6.3** is usually required, especially when it comes to high frequency applications.

The principle is to compare simulation results to measurement results and trim the model until the difference is acceptable. Trimming differs according to model origin. If it is a manufacturer model, trimming consists mainly of adding environmental parasitics such as PCB trace resistance, inductance, and capacitance.

If it is a homemade model, trimming implies changing model parameters and adding environmental parasitic elements. It is a much more complex process that requires a good knowledge of physics, electronics, and the underlying math.

The key here is not to blame the simulator or the model or the test setup if differences exist. The important thing is to understand why there are differences and what can influence these differences so as to bring the simulation schematic and model closer and closer to the test setup until results match. It is also important to keep in mind that the issue can be located on the test side and that the setup might require improvement.

CIRCUIT DESCRIPTION

Most circuit simulators accept a netlist input. The netlist format was popularized by *SPICE* in the 1970s even though the "*SPICE* netlist" is in fact a components and commands list. Here is a description of *SPICE* netlists syntax for those who need to write netlists manually or need to read and understand netlists.

An example of how a schematic is described by a netlist: On the left in **Figure 6.4**, the schematic shows an ac voltage source driving a first order low-pass filter. Components and nets are labeled. Here, there are three components and two nodes, plus ground. On the right is the circuit netlist. It is a text form descrip-

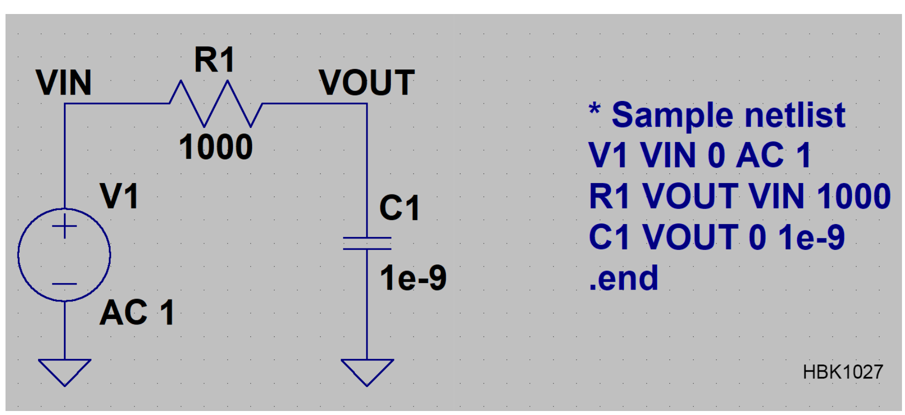


Figure 6.4 — A simple schematic with its netlist description at right.

tion of a schematic that can be parsed by the simulator to build an internal description that can be simulated.

- *SPICE* netlists are not case sensitive.
- A *SPICE* netlist file is basically organized as separate lines.
- Each line ends with a return or newline character.
- Each line contains different fields. Fields are separated by a space character.
- The line's first character defines the line's function.

SPICE uses SI units (volts, amperes, ohms, farads, henrys, hertz, seconds, etc.). It does accept letters for exponent (k for kilo, p for pico, etc.). Confusion is possible with "m" that stands for milli while "meg" stands for mega, and "mil" stands for 0.001 inch. A good practice to avoid this confusion is to use a floating-point format (1.5E3 for 1.5 k). The default temperature for *SPICE* is 27 °C (300 K, 80.6 °F). It can be changed and even swept by dedicated commands.

Comment Lines

The netlist file's first line is always considered to be a title line and is not interpreted. It is treated like a comment line but appears in detailed result files. Any line beginning with an asterisk (*) character is a comment line. It is a good practice to begin the title line with an asterisk so that an error is not caused if a blank line is unintentionally added at the beginning of the file.

Continuation Lines

A line beginning with a plus (+) character is a continuation of previous line. There can be more than one continuation line.

Component Lines

A line beginning with a letter (A to Z or a to z) is an instance of a component. The line's first letter defines the component type. For instance, C stands for a capacitor and R stands for a resistor. The line's first field (beginning with the identifier letter) defines the component name or designator, such as R1, L4, D2, etc. The name can contain letters and numbers and must be unique. Other fields define the net names the component is connected to. The number of nets depends on the component type and model. Most recent simulators accept alphanumerical net names, but the old *SPICE* syntax allowed only numeric net names. Additional fields depending on the component nature define model name, sizing, etc.

The following is a reduced components list from strict *SPICE* syntax. Many more recent simulators have additional letters for additional component types.

- C: Capacitors. Next two fields are net names. Then, either a value or a model name followed by a value.
- D: Diodes. Next two fields define Anode and Cathode net names in that sequence. Then a model name and optionally a multiplier
- E: Voltage-controlled voltage sources. Next two fields are output nodes followed by two fields defining controlling nodes and finally a field defining gain value.
- F: Current-controlled current sources. Next two fields are output nodes followed by one field defining controlling source name and finally a field defining gain value.
- G: Voltage-controlled current sources. Next two fields are output nodes followed by two fields defining controlling nodes and finally a field defining gain value.
- H: Current-controlled voltage sources. Next two fields are output nodes followed by one field defining controlling source name and finally a field defining gain value.
- I: Independent current sources. Next two fields are output nodes followed by current type and parameter(s).
- J: JFETs.
- K: Inductor coupling factor.
- L: Inductors.
- M: MOSFETs.
- Q: Bipolar transistors. Next four fields define Collector, Base, Emitter, and Substrate node names in that sequence. Then a field defines model name and optionally a multiplier.
- R: Resistors.
- T: Transmission lines.
- V: Voltage sources.
- X: Subcircuit calls. Next fields are subcircuit pins. Last field is subcircuit name.

For a detailed description of these components, the simulator reference manual is the most valuable source of information. A description of model syntax is available in the *LTSPICE* guide:

www.analog.com/media/en/simulation-models/spice-models/LTspiceGettingStartedGuide.pdf?modelType=spice-models.

6.2.2 Models

Some components like resistors may simply have a value, but components like diodes cannot be defined by a single value. However, all diodes share some common characteristics. They all behave in a similar way so they can be described by the same set of equations. However, a small signal diode and a power rectifier do not have the same characteristics. To reach that goal, circuit simulators use parameterized models. For instance, a diode instance can be:

D1 Node1 Node2 D1N4148

This defines that diode D1 has its anode connected to node Node1, its cathode connected to node Node2, and that it behaves according to model D1N4148. So, there must be a model definition in the netlist:

.MODEL D1N4148 D (is=2.52E-9 rs=.568 n=1.752 cjo=4E-12 m=.4 tt=20E-9)

In contrast, a fast rectifier like MURS320 has the following model:

.MODEL MURS320 D (is=1.06E-9 rs=.0111 n=1.367 cjo=135E-12 m=.45 tt=45E-9)

(Note that these lines begin with a period. These are netlist commands, described in the following section.)

These two diodes use different parameter values that are passed to the equations so that they behave differently. A quick look at the parameters shows that the biggest difference between these two diodes lies in series resistance, rs, and junction capacitance at 0 V, cjo. A detailed description of all *SPICE* model equations can be found online but in most cases these equations are written for simulators to use, not to make it easy for humans to understand them.

As mentioned earlier, *SPICE* stands for Simulation Program with Integrated Circuit Emphasis. From the beginning, it has been IC-oriented. There is a difference between a discrete transistor and a transistor that is embedded in an IC.

- A discrete transistor is usually packaged separately. Even if empty, the package exhibits significant parasitic capacitance between pins and significant series inductance on each pin.
- A transistor in an IC has no package, only the IC is usually packaged. So, parasitic capacitance is about two orders of magnitude smaller. The same applies for inductance since connections are much shorter inside ICs. But in an IC, the transistor is built on the same substrate as other components and has then an isolation diode between collector and substrate.

For a discrete transistor, especially if it is to be used at high frequencies, the model describing the transistor itself should be embedded in a package description including parasitic Ls and Cs. Not doing so is the major source of differences between simulation and measurements for high frequency designs. **Figure 6.5** shows what a high frequency device model should look like

A discrete transistor model should include a package parasitic components description. Sometimes they do, especially for high frequency devices, but sometimes they don't. Users should check this to minimize risks of getting a circuit that does not work as expected. The following description shows what is modeled in *SPICE* and what is not.

For bipolar transistors, the *SPICE* model includes series resistance on collector, base, and emitter. These are named RC, RB, and RE in the model parameter list. But the model does not include series inductance. If package has significant leads length and if operating frequency is high, these inductances

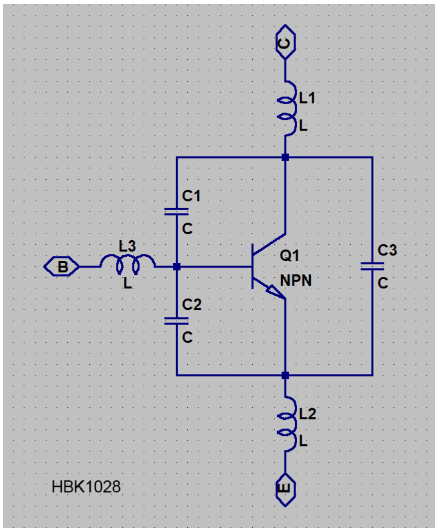


Figure 6.5 — Transistor model showing parasitic inductance and capacitance.

should be added. Fortunately, high frequency transistors usually come in short leads or no-lead packages.

The model does include both B-C and B-E junction capacitances, but these are the internal junction capacitance. The reverse-biased junction (usually B-C) capacitance results from the depleted region and depends on C-B voltage. For the forward-biased junction, capacitance (usually B-E) results from the depleted region that is very narrow and from the minority carrier transit time. At low collector current, depleted region capacitance dominates and is fairly stable with current while at high current, transit time capacitance dominates and is proportional to current.

The model does not include the fixed package parasitic capacitances. Connecting these fixed parasitic capacitances in parallel with the transistor's varying capacitances results in larger values that vary less. Unfortunately, especially for low frequency devices, manufacturers provide tweaked C_{BE} and C_{BC} values to include package capacitance values. The result is a poorly matched description of how these values vary with a signal. A more accurate model consists of putting together a transistor model as a chip without additional parasitic elements, and a package model extracted by measuring an empty package. This was the basis of the now discontinued ARRL model.

Nothing can prevent the user from building their own model. But this requires adding package parasitic elements through the difficult work of measuring them and building the transistor chip model through the even more difficult task of getting bare transistors and measuring them.

Many models (as well as parameters for models) can be found on manufacturer websites. These models should be used carefully as they are usually supplied "as is", without documentation or domain of validity information. However, they can give valuable results at least for low frequency applications.

PASSIVE COMPONENT PHYSICAL MODELS

A number of simulators have introduced "physical models" as opposed to *SPICE* ideal models. *SPICE* is somewhat hybrid with respect to this issue. It does include, for instance, series resistance on active devices pins (with values defaulting to 0) but not in passive devices.

Passive physical models include parasitic elements such as series resistance or inductance to provide a more realistic modeling. But there is no standard definition, and such models actually provide more confusion than accuracy!

The key is to be aware of what the simulator is modeling and what it is not, and to add elements missing from the simulation to make it realistic. The **RF Techniques** chapter shows component construction and explains the effect of parasitic elements of component behavior. Fortunately, the size reduction provided by SMD components has a significant effect on reducing parasitic elements.

DEPENDENT MODELS

Professional simulators, especially in the field of IC design, provide powerful tools for analyzing *process parameter spreading*. This allows simulation of how a given step in the manufacturing process can impact each component differently.

Manufacturing a transistor or an IC is a complex process with many steps using physical-chemical operations resulting in tolerances at each step. That applies for oxide thickness or doping level, for instance. As a result, devices parameters exhibit tolerances. But as one process step may affect several devices, tolerances are at least partly coupled so that they cannot be analyzed separately.

Usually, for simplicity, process step tolerances are grouped in so-called "corners" to reduce the number of combinations. Corners are often defined by their effects. For instance, tolerances for all the steps can be analyzed to find which combination results in the fastest IC operation. These are grouped in a corner called "Fast," another corner is "Slow," and so on. The tools then use model parameters that are computed at run time from requested process corners, making the model almost impossible to read. For simpler tools such as those used by hobbyists, models stay quite simple and easy to read but do not offer process corner analysis.

6.2.3 Command Lines

A line beginning with a dot (.) is a simulation command. (See the .MODEL lines in the previous section.) Most commands define simulation to be performed or contain simulation directives. These commands are associated with fields defining simulation parameters. For instance, .DC VIN 0 1 0,01 tells the simulator to perform a dc sweep of voltage source VIN from 0 V to 1 V in 10 mV steps.

A special command, .SUBCKT is used to define a subcircuit, allowing a hierarchical description of a complex circuit.

A reduced commands list from strict *SPICE* syntax follows. Many more recent simulators have additional commands or, confusingly, have modified the significance of some commands. Be sure you understand a command's function and syntax before you use it.

SIMULATION SETUP COMMANDS

These commands are interpreted prior to running the analysis. They define global variables, parameters, or libraries for models.

- .LIB: Defines a model library. Only the required devices are loaded.
- .STEP: Defines a parameter sweep. Analysis is performed once for each defined parameter value.
- .TEMP: Defines temperature.

ANALYSIS COMMANDS

- .AC: Defines small signal analysis frequency sweep. Parameters are sweep type, start frequency, stop frequency, and step. Operating point is computed first, the small signal equivalent circuit for non-linear devices is extracted, and ac analysis is performed on the circuit's equivalent small signal model.
- .DC: Defines dc sweep analysis. A voltage, a current, or other parameter can be swept to plot any output versus the swept value. It is a static sweep with no timing involved.
- .NOISE: Defines small signal noise analysis. Parameters are output, name of input source, and a step parameter. Frequency range is defined by .AC command. Summary is plotted every "step" value of AC point.
- .OP: Defines operating point analysis.
- .TRAN: Defines transient analysis. Parameters are start time, stop time, and time step.

POST-PROCESSING COMMANDS

- .FOUR: Defines a Fourier series analysis on TRAN results.
- .SENS: Defines a sensitivity analysis on OP results.

6.2.4 Example: Series Voltage Regulator

This and the following sections will show different examples of simulation methodology to design and evaluate most aspects of circuit operation. These examples are based on *LTSPICE* for its ease of use and power.

The target specification is:

- Input voltage: 15 V to 25 V
- Output voltage: 12 V +/- 5%
- Output current: 0 to 1 A

The first step is choosing an architecture. For simplicity, and since the specification is not stringent, the design can be a Zener reference, open loop regulator. The initial schematic is shown in **Figure 6.6**.

The specification states that the pass transistor should be able to handle at least 1 A and 25 V. Simple calculations show that at maximum input voltage (25 V) and maximum current, the pass transistor should be able to dissipate P = 25 – 12 × 1 = 13 W.

R1 should bias the Zener diode and supply pass transistor base current. The Zener should be biased between 5 and 20 mA to benefit from a low dynamic impedance. Base current would be load current divided by current gain, β or beta.

$$R1 = \frac{V_{in} - V_{ref}}{I_{bias} + \frac{I_{load}}{\beta}}$$

The next step is choosing a pass transistor that can handle 1 A and dissipate at least 13 W. A safety factor of 2 or 3 sets the requirement to 3 A and 40 W. Devices like the old BD241C or 2N3055 fit that requirement. Transistor datasheet minimum β is 25, so base current could be as large as 40 mA. This is more than a Zener can handle. There are two design options to increase pass transistor β:

- Using a Darlington configuration at the expense of an increase voltage drop
- Using a composite NPN-PNP configuration.

The second option is preferable but requires a power PNP together with a small signal NPN. A BD242C or 2N2955 that are complementary to the BD241C or 2N3055 are suitable. A 2N2222 is fully capable of delivering the PNP base current. 2N2222 minimum β at 40 mA is 50, so base current should not increase more than 1 mA. This value is much lower than Zener current.

Output voltage will be the Zener voltage minus 2N2222 V_{BE} (about 0.7 V, temperature dependent). An elegant solution is using a 12 V Zener diode in series with a diode-connected 2N2222 that more or less compensates for the pass driver. The tuned-up schematic is shown in **Figure 6.7**.

Models for the BD242C and 2N2955 can be found at **onsemi.com** and copied into the schematic. 2N2222 and 12 V Zener models are available in *LTSPICE*. The following simulations have been done with a BD242C power transistor, but results are very similar with the 2N2955.

Table 6.1 shows an example netlist.

It is now time to simulate this circuit in order to check the assumptions used to build it.

Unfortunately, *LTSPICE* allows only one simulation at a time. This is why the schematic doesn't show simulation commands. To work through multiple commands, you

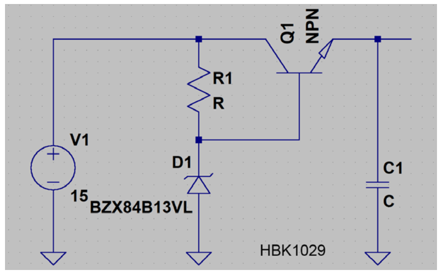


Figure 6.6 — Initial schematic for a series voltage regulator circuit.

Table 6.1
***LTSPICE* Netlist**

```
* C:\Users\JF\Desktop\Travail\LTC\DiscreteSeriesRegulator.asc
V1 N001 0 15
R1 N001 N004 500
I1 N002 0 0.5 AC 1
Q2 N003 N004 N002 0 2N2222
Q3 N002 N003 N001 0 Qbd242c
R2 N001 N003 100
Q4 N004 N004 N005 0 2N2222
D2 0 N005 BZX84C12VL
.lib C:\Users\JF\Documents\LTspiceXVII\lib\cmp\standard.dio
.lib C:\Users\JF\Documents\LTspiceXVII\lib\cmp\standard.bjt
.MODEL Qbd242c pnp
+IS=1e-09 BF=144.298 NF=0.85 VAF=15.6106
+IKF=1.39466 ISE=8.61573e-12 NE=3.32689 BR=0.1
+NR=0.75 VAR=19.1801 IKR=10 ISC=3.98813e-12
+NC=3.60783 RB=0.1 IRB=0.1 RBM=0.1
+RE=0.0001 RC=0.0601441 XTB=0.828305 XTI=1
+EG=1.05 CJE=3.18622e-10 VJE=0.49128 MJE=0.481942
+TF=1e-08 XTF=4.23532 VTF=7.00536 ITF=0.001
+CJC=3.12657e-10 VJC=0.630571 MJC=0.450516 XCJC=0.799996
+FC=0.8 CJS=0 VJS=0.75 MJS=0.5
+TR=9.57121e-06 PTF=0 KF=0 AF=1
.backanno
.end
```

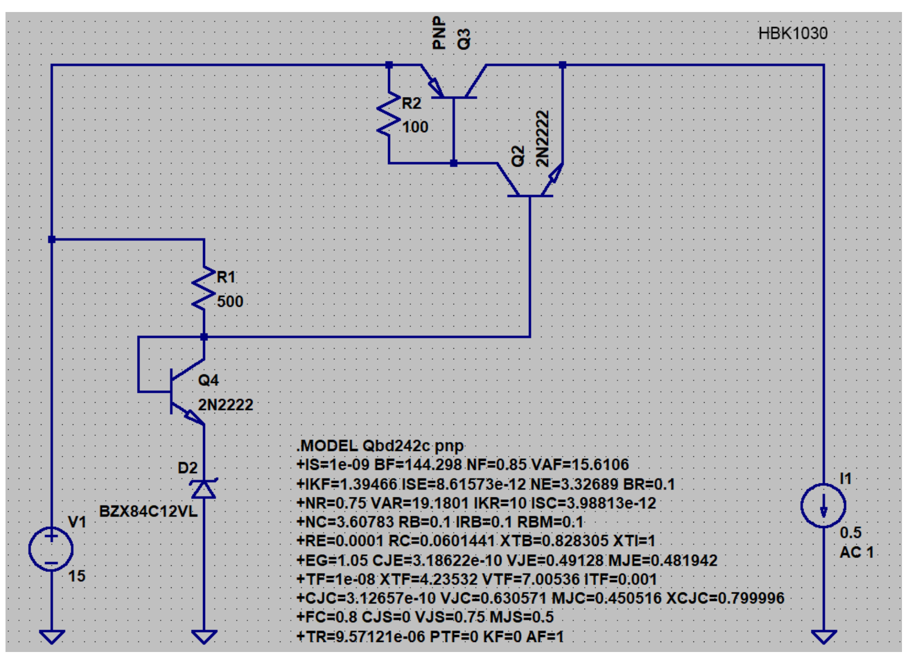


Figure 6.7 — Regulator circuit with improved pass transistor drive capability.

add a command, run the simulation, then change the command.

Obviously, running one command at a time is a cumbersome process. To streamline changing commands, here's a useful tip: turn commands into comments so that the simulation ignores them, but you have a full list of the commands you want to run in each of your subsequent simulations. To comment a command:

1. Right click on the command.
2. Choose "Cancel" in the dialog box that pops up.
3. Choose "Comment" in the second dialog box.

This way you can enter all of the commands at once while turning on only the command you want to run in the current simulation.

SWEEPING OUTPUT CURRENTS

Command is:

.DC I1 0 1 0.001

Load current is swept from 0 to 1 amp with 1 mA steps. **Figure 6.8** shows the output voltage versus load current.

The output voltage is fairly stable, but at very small loads, output voltage rises close to input voltage. This is definitely a design flaw: At no load, both transistors V_{BE} are zero, so output which is V_{IN} minus the sum of the two V_{BE} is equal to V_{IN}. To avoid this problem, a small current in the mA range should be added, for instance by connecting a 12 kΩ resistor on the output. **Figure 6.9** shows that with the small extra load, the output voltage is okay.

SWEEPING INPUT VOLTAGE FOR VARIOUS LOAD CURRENTS

Commands are:

.STEP I1 0 1 0.5 Load current stepped from 0 to 1 A by 0.5 A steps.

.DC V1 15 25 0.01Input voltage swept from 15 to 25 V by 10 mV steps.

Figure 6.10 shows that as expected, output voltage increases with input voltage and decreases with load current.

OUTPUT IMPEDANCE VERSUS FREQUENCY

To measure output impedance, a 1 A ac current is superimposed on the ac load current. Then ac output voltage gives a direct reading of output impedance. One should not be confused that ac current can be larger than dc current. It does not mean current changes direction. It just means that a small-signal model is computed from the operating point and then the linear model is used to compute ac response. That's one benefit of simulation as directly superimposing ac and dc currents would be hard to do with real components!

Commands are:

.STEP I1 0 1 0.5 Load current stepped from 0 to 1 A by 0.5 A steps.

.AC DEC 20 1 1E8Frequency is swept from 1 Hz to 100 MHz with 20 steps per decade.

Figure 6.11 shows that low frequency output impedance depends on current. At high current, it is fairly low but does increase with frequency. The usual fix is adding a capacitor across the output. Pay attention to the Equivalent Series Resistance (ESR) of that capacitor and eventually for large values of capacitance, the Equivalent Series Inductance (ESL).

The simulation shows output impedance

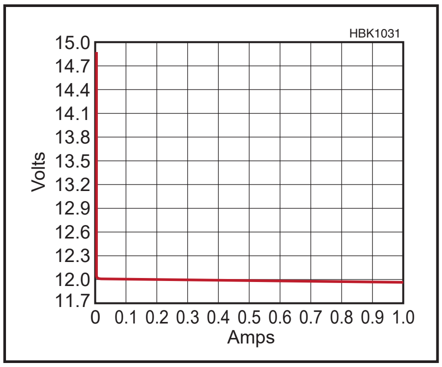


Figure 6.8 — Output voltage versus load current of the circuit in Figure 6.7.

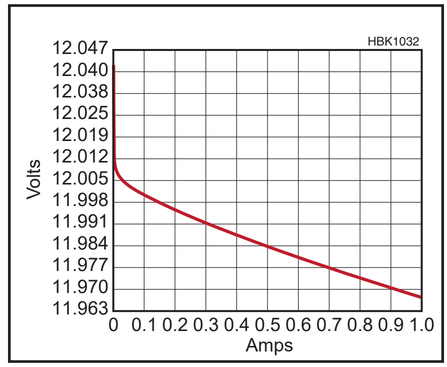


Figure 6.9 — Output voltage versus load current with a 12 kΩ load.

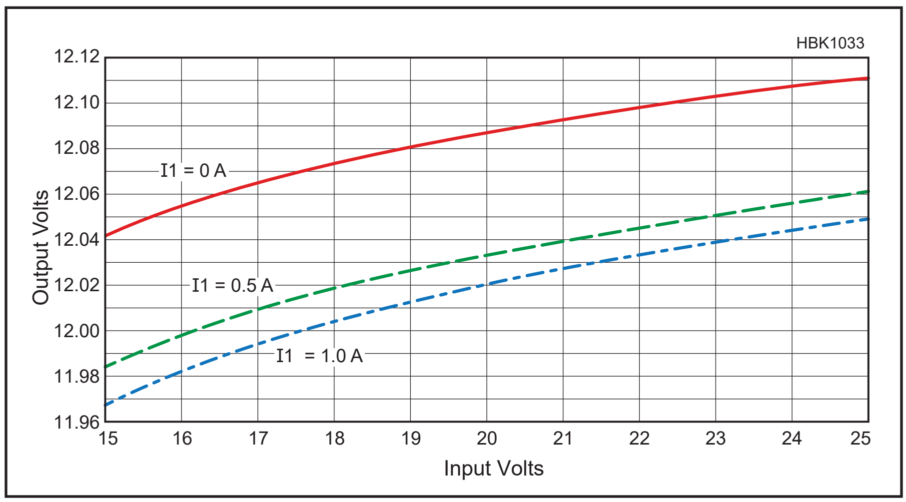


Figure 6.10 — A swept simulation of output voltage for a range of input voltages.

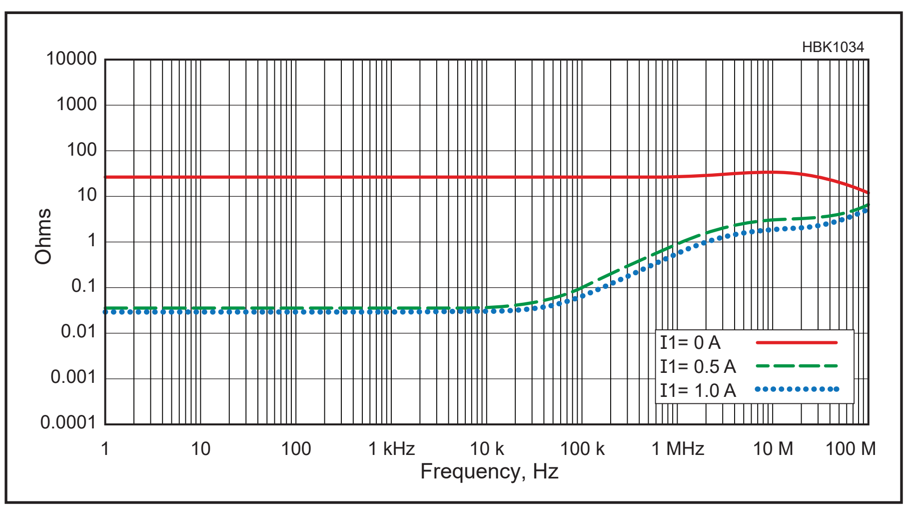


Figure 6.11 — Output impedance versus frequency from 1 Hz to 100 MHz.

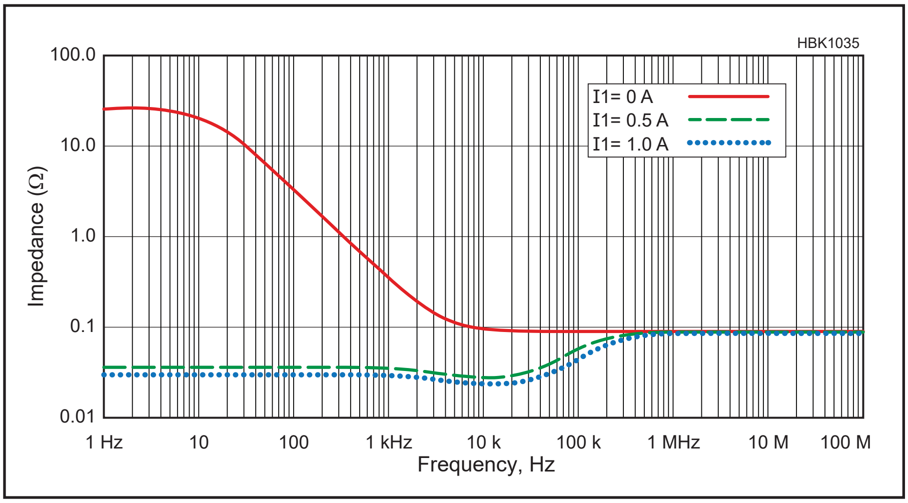


Figure 6.12 — The effect on output impedance of adding a 470 µF capacitor at the output.

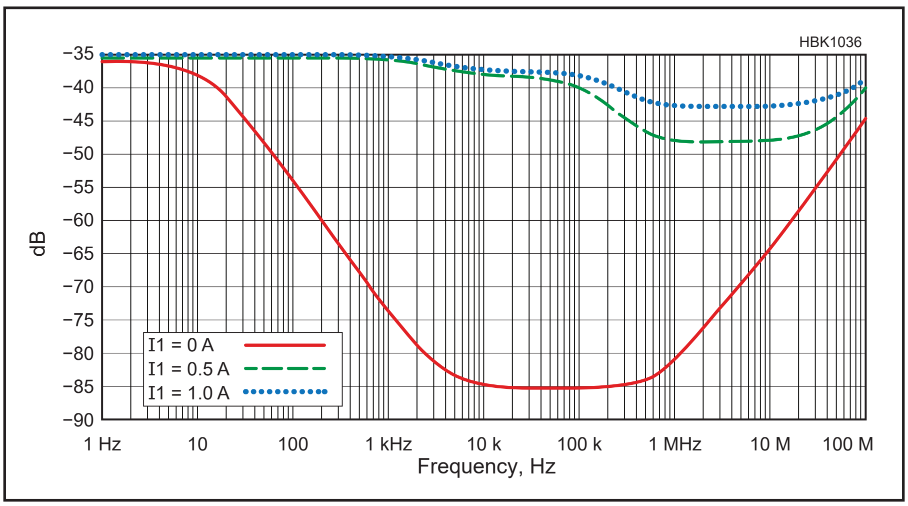


Figure 6.13 — Rejection of ac voltage on the input versus frequency from 1 Hz to 100 MHz.

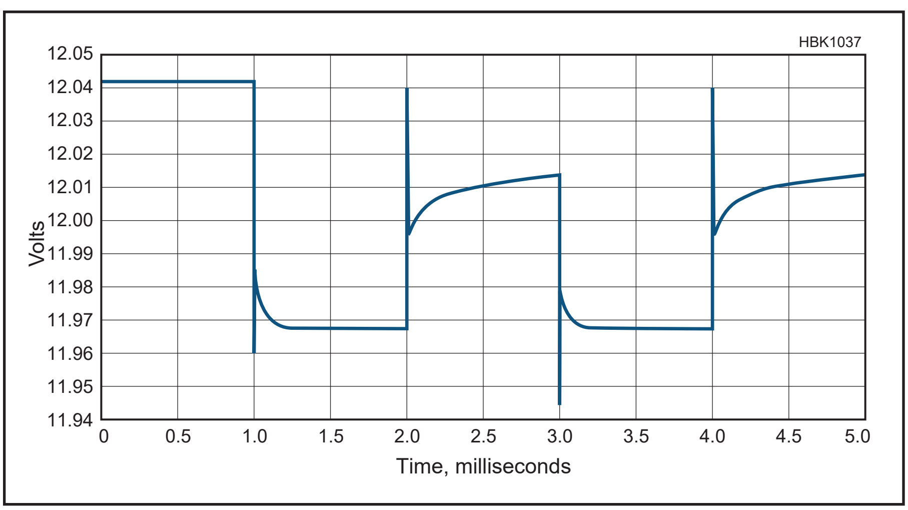


Figure 6.14 — Output voltage transients resulting from load current changes.

up to 100 MHz. Keep in mind that if wiring or PCB traces have significant length, parasitic inductance and resistance will affect this result. Add them to the schematic to get more realistic results.

Figure 6.12 shows the effect of a 90 mΩ ESR, 470 µF capacitor connected across the output.

The improvement is significant above 100 kHz. This example shows that simulation can reveal many things, but requires including realistic components, especially for parasitic elements. For instance, make sure that wires and traces exhibit some resistance (and inductance) while wiring this supply or drawing a PCB.

INPUT REJECTION VERSUS FREQUENCY

A 1-V ac source is superimposed on the input dc voltage. Output ac voltage (in dBV by default) gives a direct reading of input rejection.

Commands are:

.STEP I1 0 1 0.5 Load current stepped from 0 to 1 A by 0,5 A steps.

.AC DEC 20 1 1E8Frequency is swept from 1 Hz to 100 MHz with 20 steps per decade.

Figure 6.13 shows that input rejection is better than –35 dB.

LOAD TRANSIENT

Load current is pulsed:

I1 N002 0 PULSE(0 1 1e-3 1e-6 1e-6 999e-6 2e-3)

Command is:

.TRAN 5e-3

Figure 6.14 shows that output voltage exhibits some transient changes on load current transitions but remains quite stable.

FINAL SCHEMATIC

The final schematic after optimization is given in **Figure 6.15**.

CONCLUSION

Using *LTSPICE*, it is possible to check a circuit's operation and to optimize it with respect to its specifications. As you will have noted, it is easy to characterize a circuit using a simulator as it does not require complex (and expensive) instruments and allows measurements that are quite difficult to do on a bench.

However, you must remember that a simulator simulates only what it is given as input. In particular, non-ideal components should be modeled properly, and connections should be taken into account if relevant.

6.2.5 Example: Fourth Order Band-Pass Filter

The example circuit in **Figure 6.16** is taken from the **AF and RF Filters** chapter's section on Passive LC Filters.

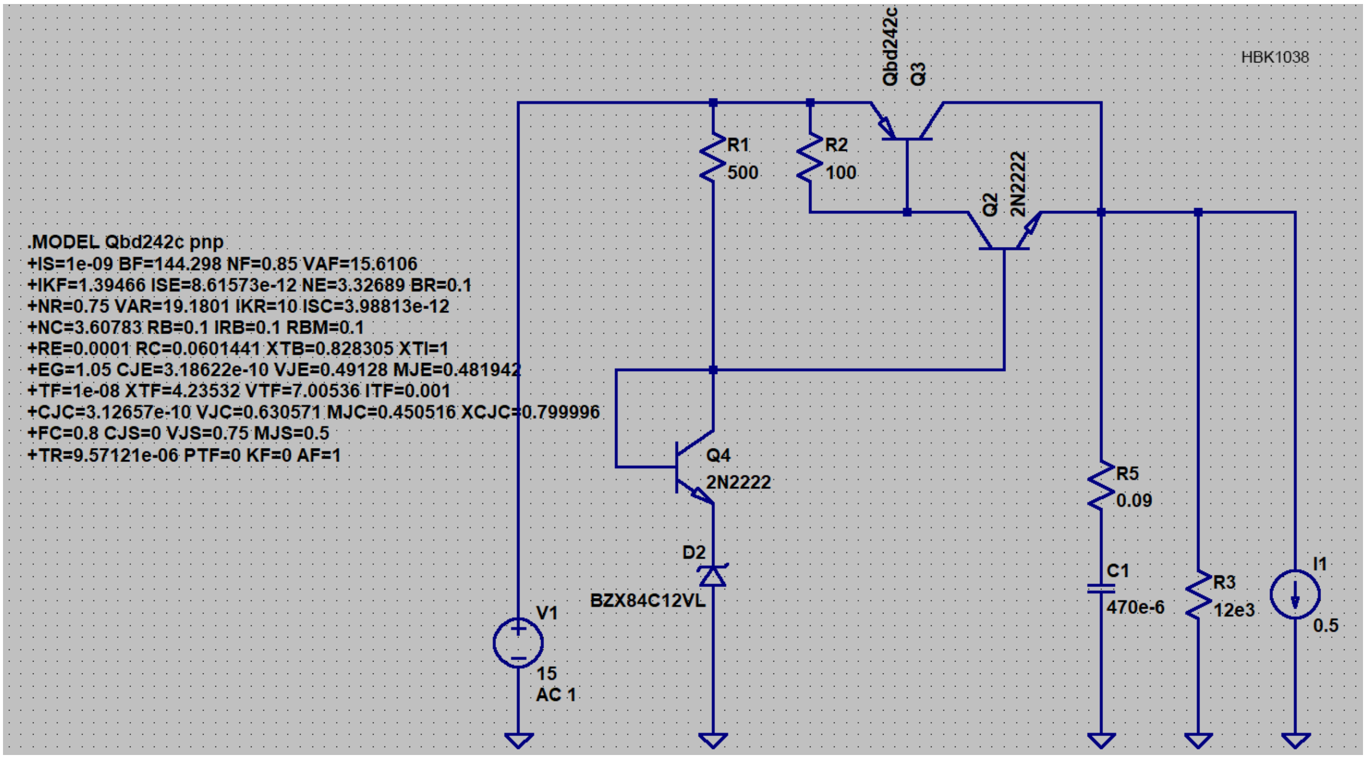


Figure 6.15 — The final schematic.

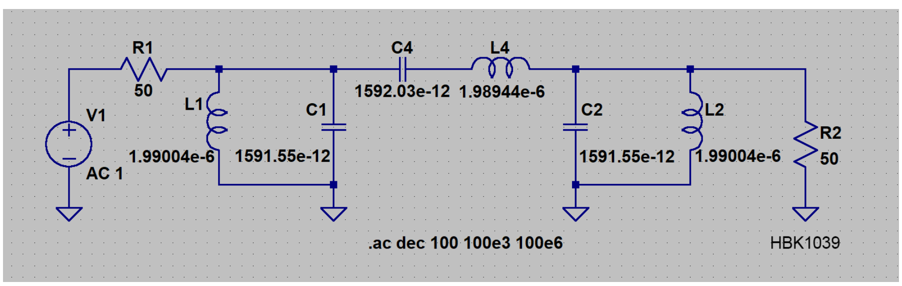


Figure 6.16 — A fourth-order band-pass LC filter.

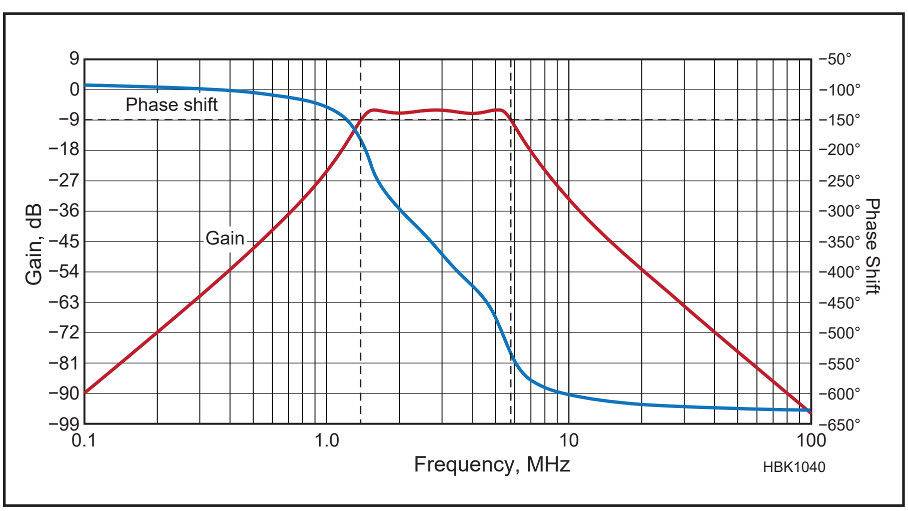


Figure 6.17 — Frequency response of the filter in Figure 6.16.

FREQUENCY RESPONSE

Figure 6.17 shows the simulated frequency response. The center frequency is 2.825 MHz and gain at that frequency is –6.02 dB. –3 dB (with respect to center gain) bandwidth extends from 1.386 MHz to 5.778 MHz. Depending on the actual value of the components and circuit wiring, parasitic elements may affect this result on the high frequency side.

INPUT IMPEDANCE

Figure 6.18 shows the simulated input impedance. Input impedance is extracted by displaying input voltage divided by input current. It is close to 50 Ω at the center frequency but fluctuates over frequency.

The component values on the schematic result from computations but are non-standard and would certainly be hard to find or create. Two issues then arise:

• What if values are rounded to standard values for availability?

• What if components values are affected by tolerances?

CHOOSING NORMALIZED COMPONENT VALUES

LTSPICE has an easy answer to both questions. Using normalized values of 2 μμH for the three inductors and 1600 pF for the three capacitors, the central frequency shifts to 2.82 MHz (–50 kHz from theoretical component values) and gain at that frequency is unaffected. The –3 dB bandwidth extends from 1.374 MHz to 5.744 MHz (shifts of

-40 to -60 kHz). That makes sense since both L and C values were rounded to the closest normalized value, and both happened to be slightly larger than their theoretical values.

EFFECT OF COMPONENTS TOLERANCES

For the effect of tolerance, we can use the Monte-Carlo feature of *LTSPICE* even though, in a difference from some other simulators, it does not have a dedicated Monte-Carlo command.

Monte-Carlo analysis is a broad class of algorithms used to study problems involving random variables. In circuit simulation, this technique uses random numbers to mimic what happens when combining components affected by tolerances. Depending on the actual value of each component that is randomly defined by tolerance, the circuit may behave differently.

To achieve this analysis, components' values are defined by a parameter involving a nominal value, a tolerance, and a random variable that is used to generate an actual value. If a circuit exhibits a significant sensitivity to tolerances, it may behave improperly in some cases. That's an issue for production as it affects yield, but it's also an issue for hobbyists as the particular item they build may not work properly.

Figure 6.19 shows the commands to perform the analysis. Tolerance is defined to be 5% on all components (.PARAM TOL = 0.05) A dummy parameter is then swept in order to run a number of simulations. For each simulation, a new set of random variables is computed but nothing else changes. (.STEP PARAM X 0 100 1) This command defines that simulation is run 101 times with parameter X changed for each run. Since parameter X is not used in the circuit, it has no effect. **Figure 6.20** shows the schematic using components with tolerance variations included.

For each component, its value is computed by multiplying the nominal value by 1 plus a Gaussian random number. That random number is generated by the function *Gauss(stddev)* where *stddev* stands for the Gaussian function standard deviation. Since 99% of values fit within ±3 times the standard deviation, defining that standard deviation as tolerance is a good approximation.

Formulas used in parameters can be used directly in component values but that makes the schematic much harder to read. Note that for clarity, nominal component values could be added as comments for reference.

Figure 6.21 is the simulated frequency re-

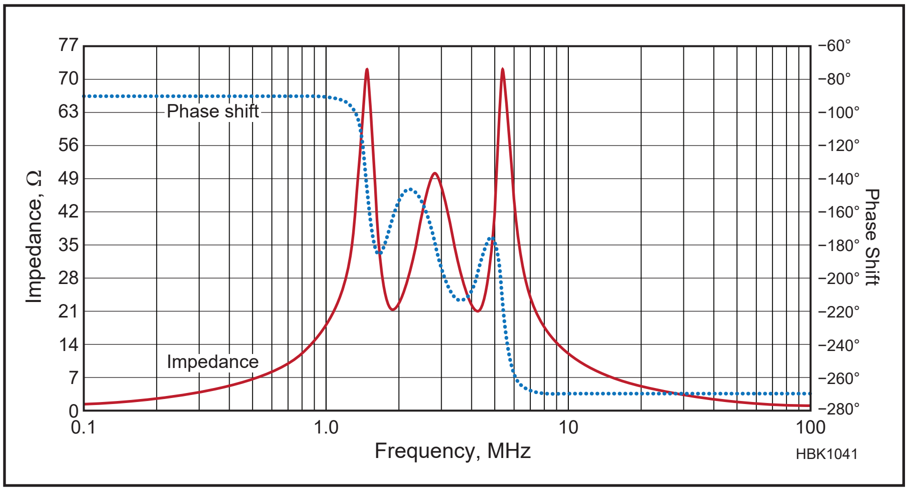


Figure 6.18 — Input impedance versus frequency of the filter in Figure 6.16.

```
HBK1042
.PARAM TOL=0.05

.param valC1={1591.55e-12*(1+gauss(tol/3))}
.param valL1={1.99004e-6*(1+gauss(tol/3))}
.param valC2={1591.55e-12*(1+gauss(tol/3))}
.param valL2={1.99004e-6*(1+gauss(tol/3))}
.param valC4={1592.03e-12*(1+gauss(tol/3))}
.param valL4={1.98944e-6*(1+gauss(tol/3))}

.step param X 0 100 1 ; a dummy parameter to cycle Monte Carlo runs
```

Figure 6.19 — Commands to perform Monte-Carlo analysis of the filter by randomly changing component values.

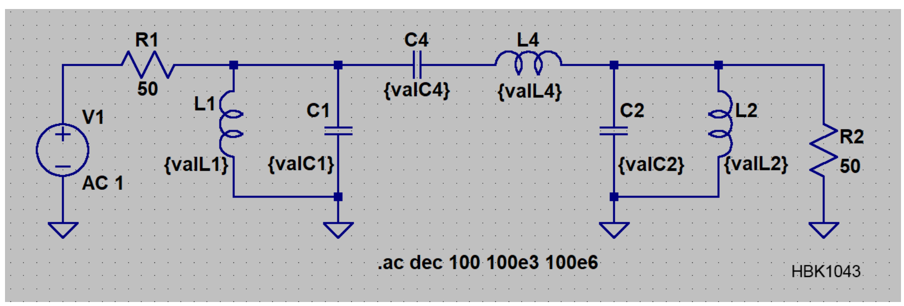


Figure 6.20 — The schematic of the filter showing the components with random value variations.

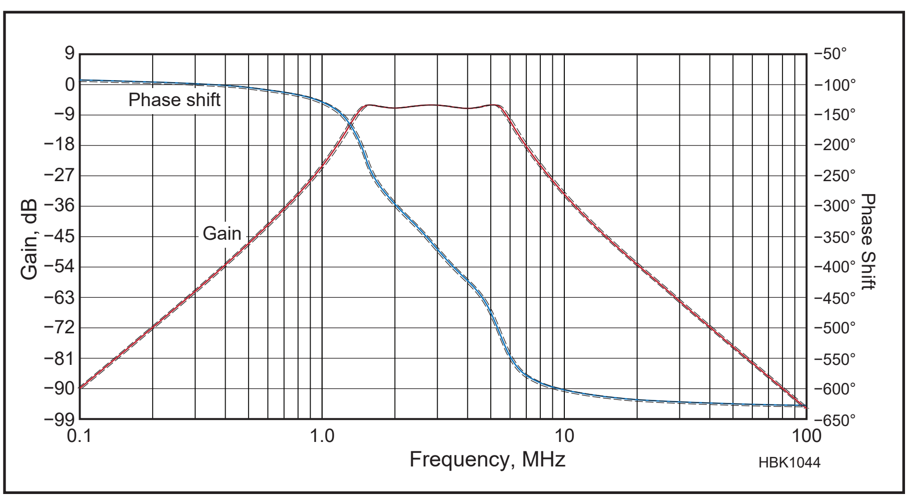


Figure 6.21 — The results of the Monte-Carlo analysis of the filter.

sponse for 101 runs. It shows some variations. Should these variations be unacceptable for the application, components with tighter tolerance should be used.

CONCLUSION

This example shows an additional benefit of simulators: detecting potential production yield even before building the first prototype.

6.2.6 Example: Crystal Oscillator

When it comes to simulating a crystal oscillator, a number of issues arise, and some designers may just give up the task. But if addressed properly, a crystal oscillator can be simulated and give valuable insight on its operation. Some classic issues of crystal oscillator simulation follow.

CRYSTAL MODEL AND PARAMETERS

A crystal equivalent schematic is shown in **Figure 6.22**. The schematic shows a series RLC branch in parallel with a capacitor. The series branch is the electrical representation of the crystal mechanical resonator, as transformed by the piezoelectric effect. (See the **Oscillators and Synthesizers** chapter for more information about crystal parameters.)

Intuitively, one can feel that at low frequencies, when inductor impedance is zero and capacitor impedance is large with respect to resistance, the equivalent impedance is that of the two capacitors in parallel. The same way, at high frequencies, when inductor impedance is large, equivalent impedance is that of the parallel capacitor only. In between, things are more difficult to guess, calculation is a bit complex but fortunately, simulation can help. **Figure 6.23** shows the impedance versus frequency of the crystal's equivalent schematic.

Overall, impedance follows a 1/f law, confirming the capacitive behavior at high and low frequencies. Around 25 MHz, two resonances occur. A series resonance where impedance reaches a minimum and a parallel resonance where impedance reaches a maximum. The series resonance frequency is lower than the parallel resonance frequency. This is because series resonance involves only L1 and C1 while the parallel resonance involves L1 and C1 in series with C2, resulting in a slightly lower capacitance value (since C2 is much larger than C1). Both resonances are very narrow because the crystal's quality factor, Q, is very high.

DEFINING CRYSTAL MODEL PARAMETERS

A concern here is to find the component values to use in the crystal equivalent schematic. Crystal manufacturers rarely supply these values. Fortunately, there is a relatively simple method to find them. In most cases, C2 and R1 are specified in the crystal data sheet. R1 is named series resistance and C2 is named parallel or shunt capacitance, not to be confused with the load capacitance.

Now what about L1 and C1 values? L1 can be estimated from R1 and Q values. R1 is generally available. Q is sometimes specified but it stands in the 10,000 to 100,000 region. We'll see later that this value is not critical.

From the equation for series resonator Q, L1 can be found:

$$Q = \frac{2\pi f L1}{R1} \text{ so } L1 = \frac{QR1}{2\pi f}$$

C1 and L1 define the series resonance that is either defined as or very close to the crystal parallel frequency.

$$F0 = \frac{1}{2\pi\sqrt{L1C1}} \text{ so } C1 = \frac{1}{(2\pi F0)^2 L1}$$

Finally, knowing F0, Q and R1 and C2 is sufficient to calculate the missing values L1 and C1.

The equivalent schematic used so far was extracted from a 25 MHz miniature SMD crystal from Mouser Electronics with a specified series resistance R1 = 40 Ω and shunt capacitance C2 = 7 pF. Assuming a Q of 40,000 leads to L1 = 0.0101 H rounded to 0.01 H. And then C1 = 4,05 fF. The model is complete! (Note the very large value of L1 and very small value of C1 required by the high Q!)

THE OSCILLATOR

There are many options for a crystal oscillator and a Colpitts type using a follower NPN is used in **Figure 6.24**. A simple bias is used to fix dc base voltage around 3.3 V and dc collector current around 5 mA. Loop gain is defined by a capacitive divider that also defines crystal load capacitance.

OSCILLATOR LOOP GAIN

First, let's look at loop gain by connecting a series ac source and **Figure 6.25** shows the gain versus frequency (zoomed into the region of interest). This shows that gain is larger than 1 at a frequency where phase is 0° so the conditions for oscillation are met.

OSCILLATION STARTUP

Now, let's look at base and emitter signals in the transient simulation of **Figure 6.26**. These two signals are perfectly stable, but the oscillator seems not to work! What happened? The oscillation condition is met but oscillator does not start.

In fact, a real circuit always has some noise. This noise power spreads over frequency and then has a non-zero value at the frequency where oscillation condition is met. So, oscillation may start from noise and increase in amplitude over time until it reaches some

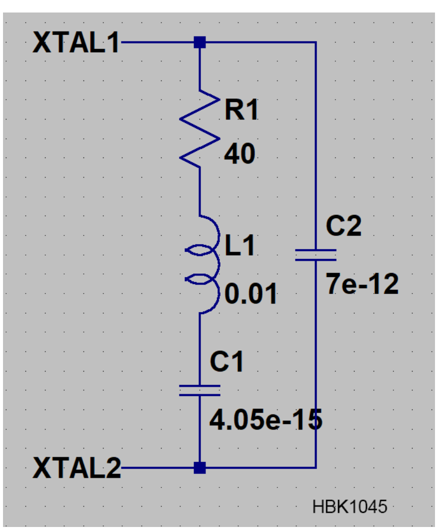


Figure 6.22 — A crystal's equivalent model circuit.

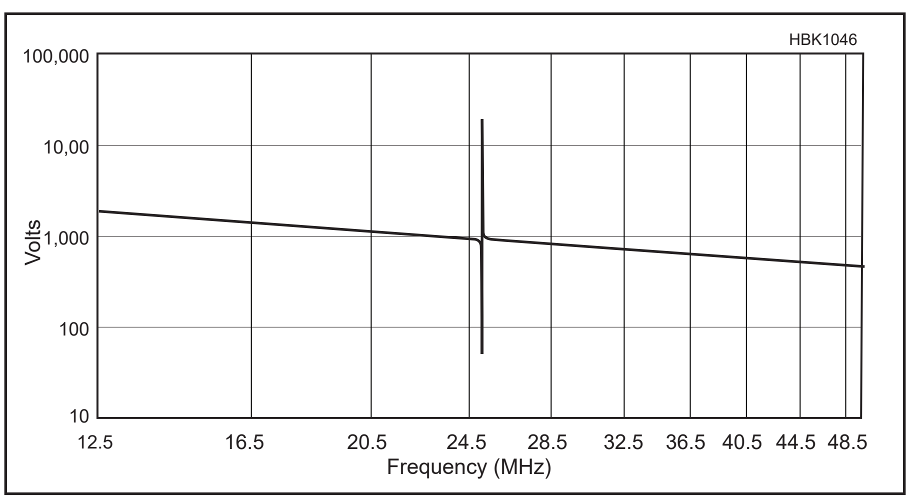


Figure 6.23 — Impedance versus frequency of the crystal in Figure 6.22.

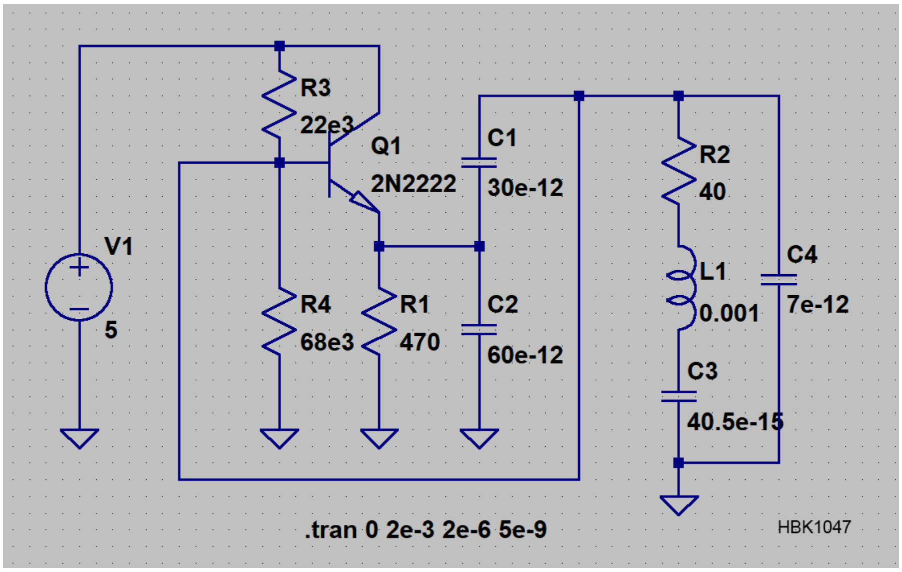


Figure 6.24 — The schematic for a Colpitts crystal oscillator.

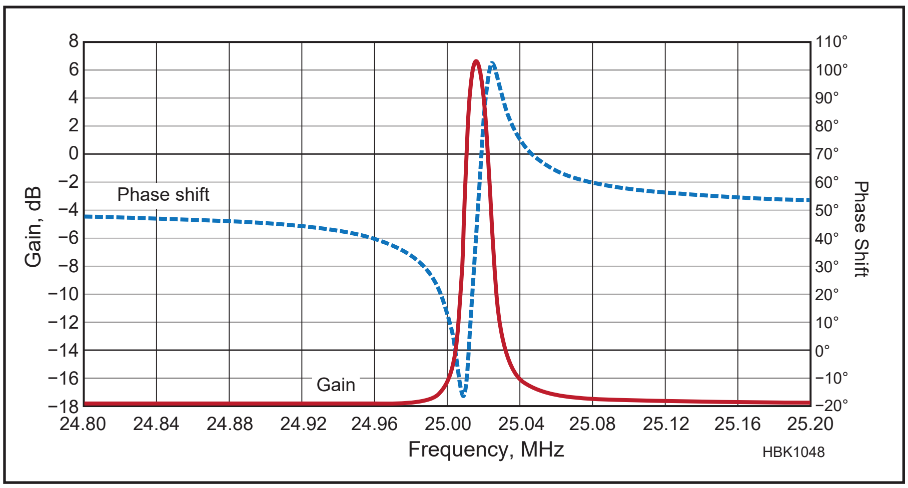


Figure 6.25 — Loop gain of the oscillator circuit in Figure 6.24.

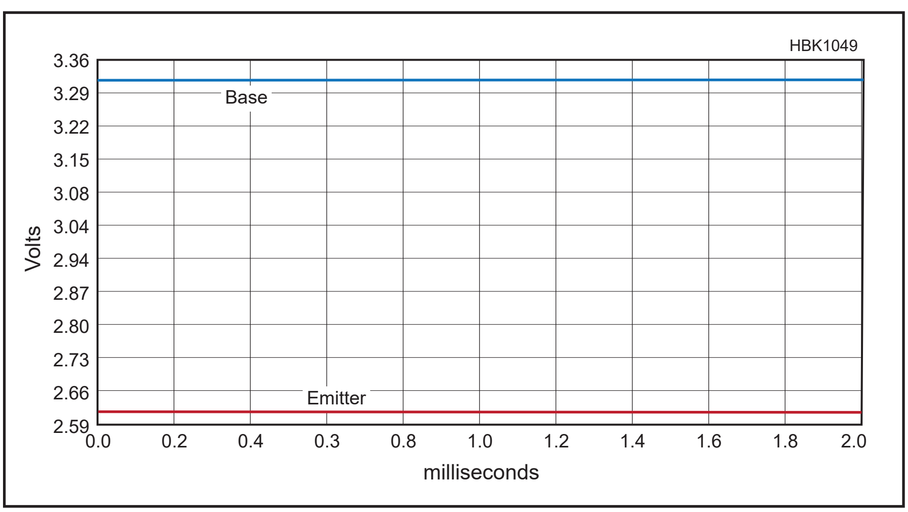


Figure 6.26 — Base and emitter signals for the oscillator circuit in Figure 6.24.

non-linearity that limits amplitude.

In a transient simulation, there is no explicit noise and computation noise is lower than default accuracy so oscillation cannot start. Could this be related to simulator settings?

SIMULATOR SETTINGS

Forcing accuracy to 1E-9 instead of default 1E-3 and limiting time steps to 5 ns so that there are at least several points per period does not improve the situation.

"SNAP KICKING"

A classic method of simulating an oscillator starting up is "putting a snap" in the crystal to cause it to start. **Figure 6.27** shows how this can be done by adding one period of a current source in parallel with the crystal. After that transient, current is zero and has no more effect.

Even with that trick, after a large initial transient and despite an oscillation at the right frequency, the oscillation is damped and would stop after some time as shown in **Figure 6.28**. What happened? In fact, Q is so large that it would require a very high dynamic range to handle the actual amplitude increase from one period to the next. So, what if Q is reduced by a factor of 10?

REDUCING Q FACTOR

This is easily done by dividing L1 value by 10 and multiplying C1 by 10 to keep the oscillation frequency the same value. Now, with a lower quality factor and nothing else changed, the oscillator does start as **Figure 6.29** shows. Oscillation amplitude increased exponentially with time until it reaches a limitation by circuit non-linearity.

CONCLUSION

When it comes to simulation of harmonic oscillators that operate through loop small-signal instability, adding a short transient is often necessary to cause the oscillation to actually start in transient simulation. In cases where the Q factor is very large, artificially reducing it helps and speeds up transient simulation. Oscillation conditions are correctly predicted by ac small-signal analysis. In the case of relaxation oscillators, the type using hysteresis in the loop, they always start properly in transient simulations but cannot be analyzed with ac simulation as they have no stable operating point.

6.2.7 Non-*SPICE* Simulators

SPICE set a standard for circuit simulation and many simulators evolved from *SPICE*. The breakthrough was such that most simulators that did exist before *SPICE* simply disappeared. For quite some time, electronic circuit simulation was completely ruled by *SPICE*,

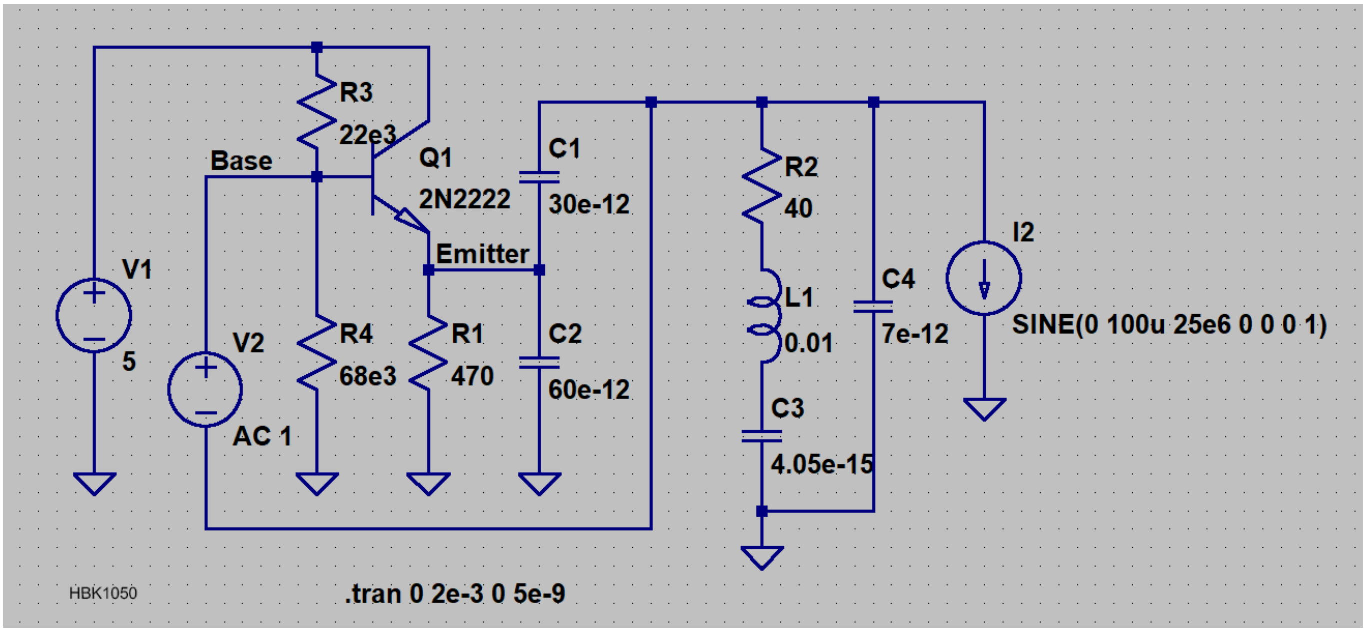


Figure 6.27 — Adding a "snap" transient to the crystal as a current source that is active for only one simulation period.

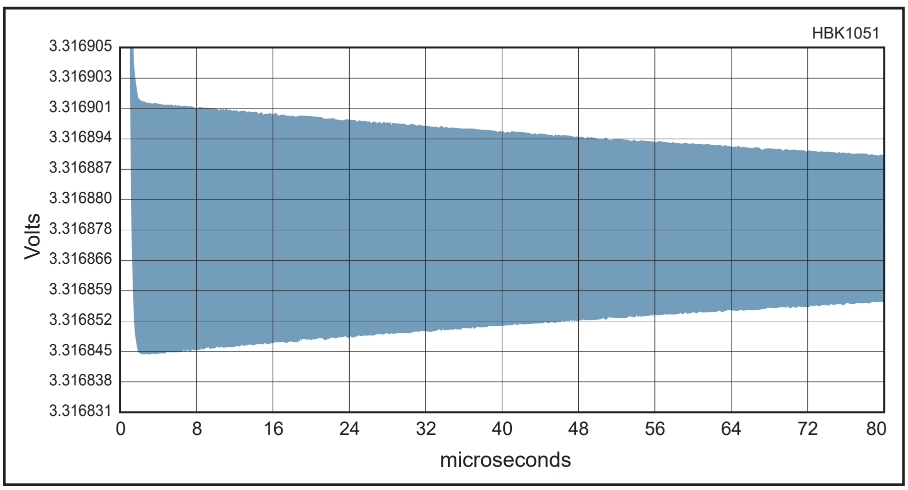


Figure 6.28 — Output waveform for the oscillator with the snap transient.

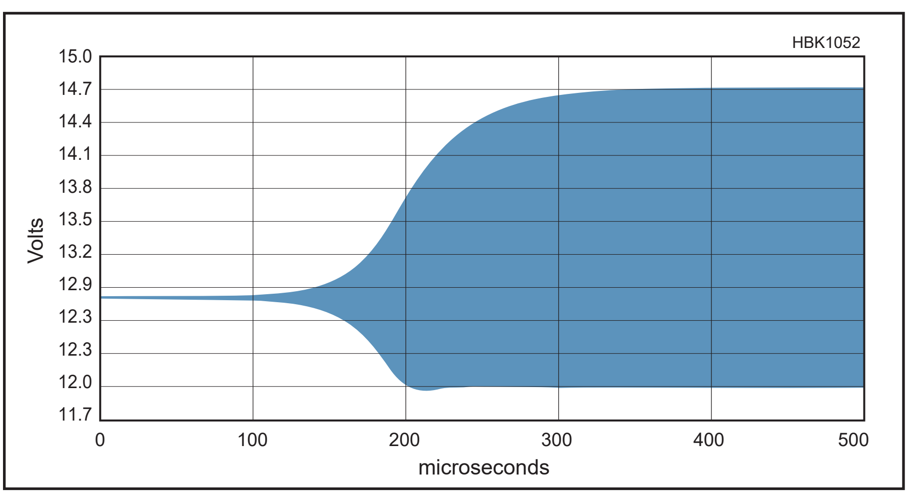


Figure 6.29 — Final output waveform for the oscillator after crystal Q was reduced.

SPICE compatibility, and extensions to *SPICE*. But as already stated, *SPICE* types of analysis suffer from some intrinsic limitations. As a result, some recognition of other needs slowly emerged and non-*SPICE* circuit simulators appeared.

Probably the most significant concept having been developed is *Harmonic Balance*. This approach is so successful at fixing some *SPICE*-type analysis issues that some major commercial *SPICE* products have included Harmonic Balance features and have even significantly extended and improved it. Harmonic Balance is discussed in more detail in this chapter's section on the **Limitations of Simulation at RF**.

Another field is *Behavioral Simulation.* This approach has been integrated in most *SPICE*-derived simulators through syntax extensions without changing simulation types. This is also true for mixed signal, analog-digital simulation.

ELECTROMAGNETIC SIMULATORS

Another type of simulator does not target circuit simulation but extracts equivalent schematics or S-parameters from a physical structure description. These are *electromagnetic simulators*. Confusion exists because some of these simulators allow mixing physical description and active devices described by appropriate models. Above frequencies such that propagation effects can be neglected, electromagnetic simulators are mandatory. For professional development there are a number of (expensive) tools. Electromagnetic simulators are discussed later in this chapter.

6.3 Limitations of Simulation at RF

Experienced users of circuit simulation software are wary of using any software near or outside the boundaries of circuits and parameters for which it was intended and tested. RF simulation can present just such situations, leading to software failure and unrealistic results. Introduced and summarized in this section, several detailed papers by Dr. Ulrich Rohde, N1UL, exploring simulation at RF are provided with the online content. The papers are:

- "Using Simulation at RF" by Rohde, a survey of the issues of RF simulation and the techniques used in current modeling programs.
- "The Dangers of Simple Usage of Microwave Software" by Rohde and Hartnagel, a discussion of inaccuracies introduced by device parameter measurement and model characteristics.
- "Mathematical Stability Problems in Modern Non-Linear Simulations Programs" by Rohde and Lakhe, presenting various approaches to dealing with nonlinear circuit simulation.
- "Transistor Applications from RF to Microwave Frequencies" by Rohde, Poddar, and Rudolph, a comprehensive paper showing examples circuit simulation techniques through microwave frequencies.
- "Harmonic Balance" by Rohde, Poddar, Rudolph, a discussion of how harmonic balance modeling techniques work.

In addition, there are many online resources to help you obtain trustworthy simulation results with a simulator designed for RF. For the interested reader with some technical background, the online paper "Introduction to RF Simulation and its Application" by Ken Kundert (**hkenkundert.com/docs/bctm98-RFsim.pdf**) provides an introduction to RF simulation methods and how they account for the characteristics of RF circuits when generating common RF measurements. The website The Designer's Guide (**www.designers-guide.org**) also provides many tutorials, technical guides, models, and other resources for analog and RF simulation users.

While the precise lower bound of "RF" is ill-defined, RF effects start at about 100 kHz. This was first noticed as self-resonance of high-Q inductors for receivers. In response, Litz wire was invented in which braided copper wires were covered with cotton and then braided again to reduce self-resonance effects.

As frequencies get higher, passive elements will show the effects of parasitic elements such as lead inductance and stray capacitance. At very high frequencies, the physical dimensions of components and their interconnections reach an appreciable fraction of the

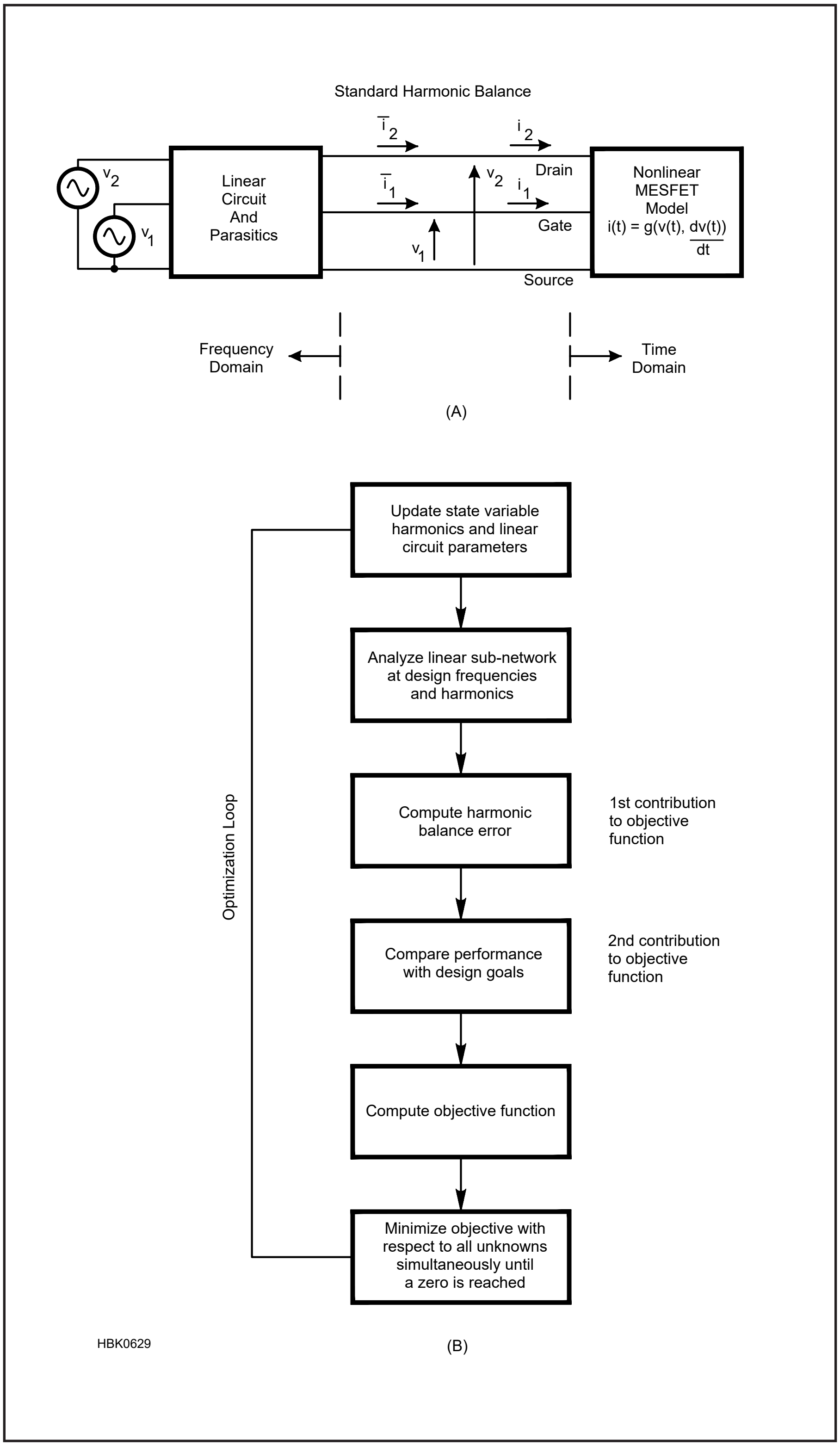


Figure 6.30 — (A) MESFET circuit partitioned into linear and nonlinear sub-circuits for harmonic balance analysis. Applied gate and drain voltages, and relevant terminal voltages and currents, are indicated. (B) Flowchart of a general purpose harmonic balance design algorithm that includes optimization.

signal wavelength and their RF performance can change drastically.

RF simulators fall in the categories of *SPICE*, harmonic balance (HB) programs, and EM (electromagnetic) programs. The EM simulators are more exotic programs. Two types are common, the 2D (2.5) or 2-dimensional and the full 3-dimensional versions. They are used to analyze planar circuits, including vias (connections between layers) and wraparounds (top-to-ground plane connections), and solid-shapes at RF. They go far beyond the *SPICE* concept.

6.3.1 *SPICE*-Based Simulators

SPICE was originally developed for low frequency and dc analysis. (Modern *SPICE* programs are based on *SPICE3* from University of California — Berkeley.) While doing dc, frequency, and time-domain simulations very well, *SPICE*-based simulation has some problems. The time-domain calculation uses the very complex mathematics of the Newton-Raphson solution to nonlinear equations. These methods are not always stable. All kinds of adjustments to the program settings may be necessary for the calculations to converge properly. Knowledge of the specifics of different types of electronic circuits can assist the user in finding an accurate solution by specifying appropriate analysis modes, options, tolerances, and suitable model parameters. For example, oscillators require certain initializations not necessary for amplifiers and bipolar transistors may need different convergence tolerances than do MOS circuits. Generally, *SPICE* finds a solution to most circuit problems. However, because of the nonlinearity of the circuit equations and a few imperfections in the analytical device models, a solution is not always guaranteed when the circuit and its specification are otherwise correct.

The next problem at RF is that the basic *SPICE* simulator uses ideal elements and some transmission line models. As we approach higher frequencies, where the lumped elements turn into distributed elements and special connecting elements become necessary, the use of the standard elements ends. To complicate matters, active elements such as diodes and transistors force the designer to more complex simulators. Adding the missing component elements leads to highly complex models and problems of convergence in which the simulator gives an error advising of a numerical problem or — more likely — by failing to generate a solution.

SPICE also has problems with very high-Q circuits and noise analysis. Questions of the noise figure of amplifiers or phase noise of an oscillator cannot be answered by a *SPICE*-based program accurately. Noise analysis, if not based on the noise correlation matrix approach, will not be correct if the feedback capacitance (Im(Y_{12}), the imaginary component of Y-parameter Y_{12}) is significant at the frequencies involved. Analysis of oscillators in *SPICE* does not give a reliable output frequency and some of the latest *SPICE* programs resort to some approximation calculations.

6.3.2 Harmonic Balance (Hb) Simulators

There are now nonlinear microwave simulators using the *harmonic balance* (HB) technique. Harmonic balance (HB) analysis is performed using a spectrum of harmonically related frequencies or tones, similar to what you would see by measuring signals on a spectrum analyzer. The fundamental frequencies are the frequencies whose integral combinations form the spectrum of harmonic frequency components used in the analysis. On a spectrum analyzer you may see a large number of signals, even if the input to your circuit is only one or two tones. The harmonic balance analysis must truncate the number of harmonically related signals so it can be analyzed on a computer.

HB analysis performs steady-state analysis of periodically excited circuits. The circuit to

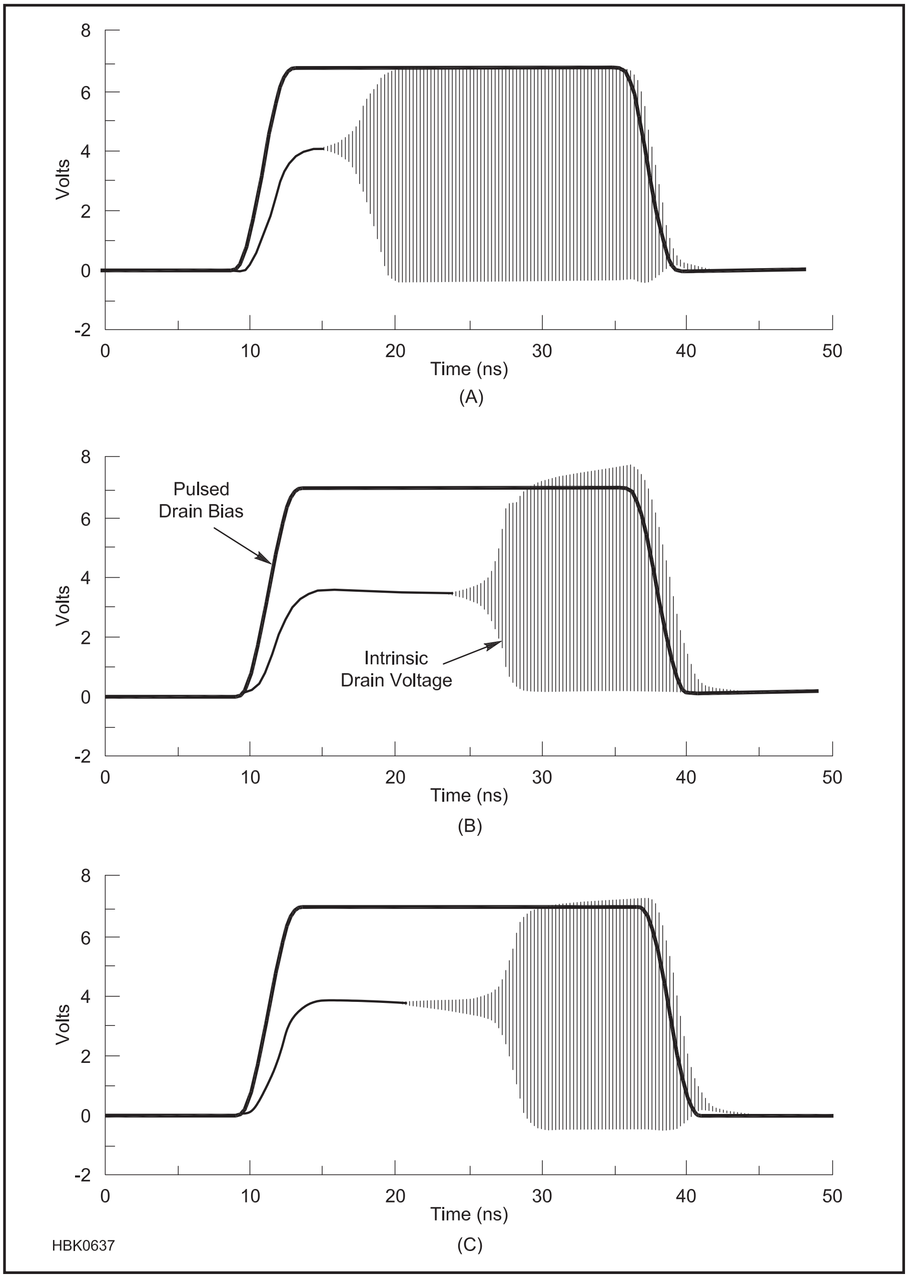


Figure 6.31 — (A) is the initial simulation of a *SPICE*-based simulator. (B) is the correct response of a pulsed microwave oscillator obtained by harmonic balance simulation using the Krylov-subspace solution. (C) is the *SPICE*-based simulation after 80 pulses of the drain voltage.

be analyzed is split into linear and nonlinear sub-circuits. The linear sub-circuit is analyzed in the frequency domain by using distributed models. This enables straightforward intermodulation calculations and mixer analysis. The nonlinear sub-circuit is calculated in the time domain by using nonlinear models derived directly from device physics. This allows a more intuitive and logical circuit representation.

Because of its nonlinear optimization capabilities, HB is uniquely suited to the task of oscillator design, mixers, and similar circuits that are difficult or impossible to simulate with linear techniques. The harmonic balance method seeks a solution to a steady-state nonlinear design problem by iteratively solving for a set of variables, referred to as *state variables*. The state variables can typically be chosen as the voltages at the linear–nonlinear interface in a circuit partitioned into linear and nonlinear segments. They are expressed as the phasor components, and their harmonics, of a sinusoidal excitation frequency. The state variables are usually found iteratively by a gradient-based technique which seeks a simultaneous solution for Kirchhoff's equations applied to the linear and nonlinear sides of the network separately.

Analysis parameters such as Number of Harmonics specify the truncation and the set of fundamental frequencies used in the analysis. The fundamental frequencies are typically not the lowest frequencies (except in the single-tone case) nor must they be the frequencies of the excitation sources. They simply define the base frequencies upon which the complete analysis spectrum is built.

Figure 6.30A diagrams the harmonic balance approach for a MESFET amplifier. (The online content paper by Rohde, Poddar, and Rudolph presents an in-depth example of HB analysis and simulation of a microwave oscillator for the interested reader.) Figure 6.30B charts a general-purpose nonlinear design algorithm that includes optimization. Modern analysis tools that must provide accurate phase noise calculation should be based on the principle of harmonic balance.

Modern HB programs have developed solutions for handling very large numbers of transistors (>1 million transistors) through efficient math solutions. Memory management through the use of matrix formulations reduces the number of internal nodes and solving nonlinear equations for transient analysis are some of the key factors to their success.

6.3.3 Contrasts in Results

The following time domain analysis is a good example of differences between *SPICE* and harmonic balance simulation. A microwave oscillator is keyed on and off and a transient analysis is performed. When using the standard *SPICE* based on *SPICE3*, the initial calculation shows an incorrect response after one iteration as seen in **Figure 6.31A**. It takes about 80 pulses (80th period of the pulsed drain voltage) until the simulation attains the correct answer (Figure 6.31C) of the Krylov-subspace-based harmonic balance in Figure 6.31B.

The frequencies involved need not be in the GHz range. Oscillators, in particular, can

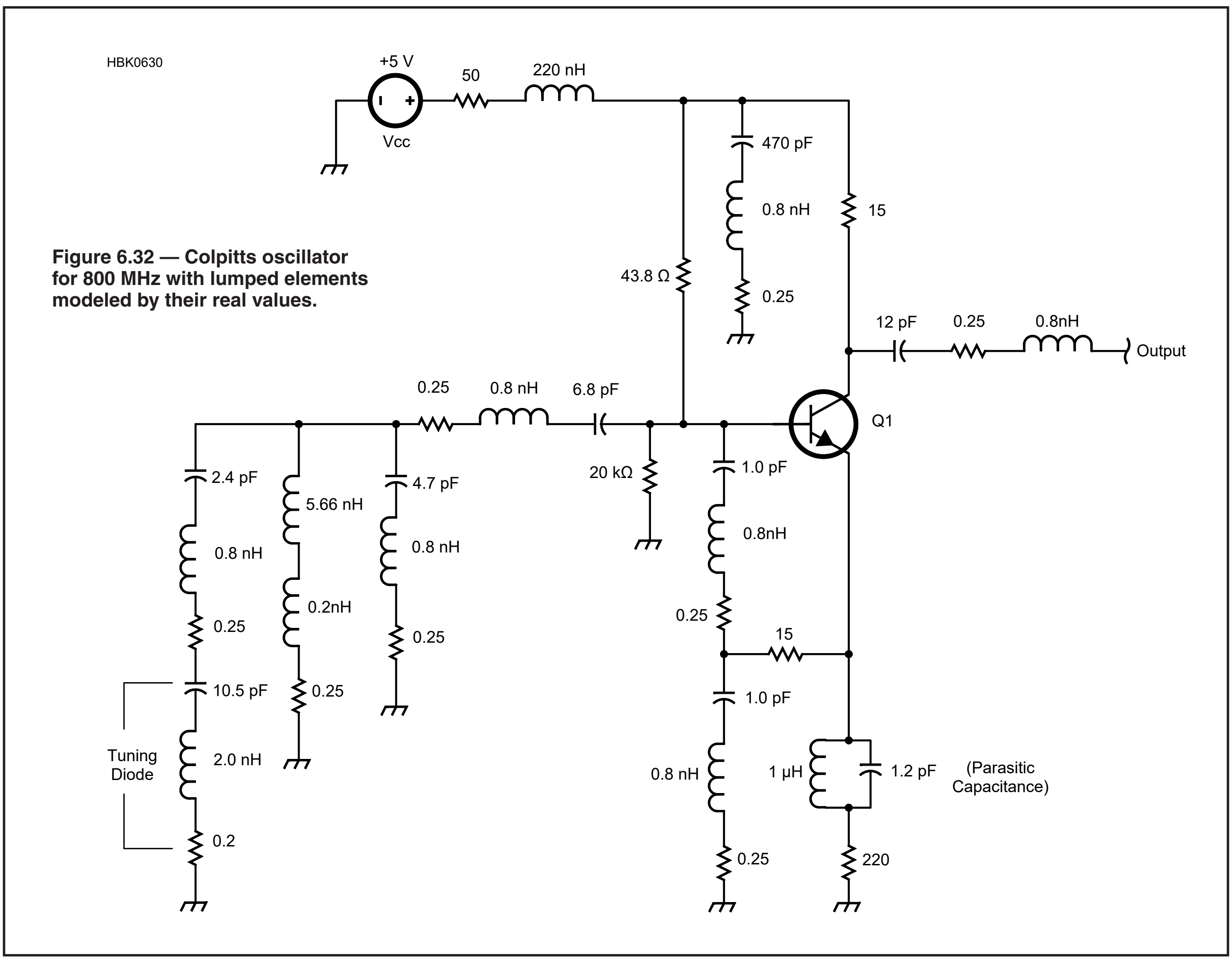


Figure 6.32 — Colpitts oscillator for 800 MHz with lumped elements modeled by their real values.

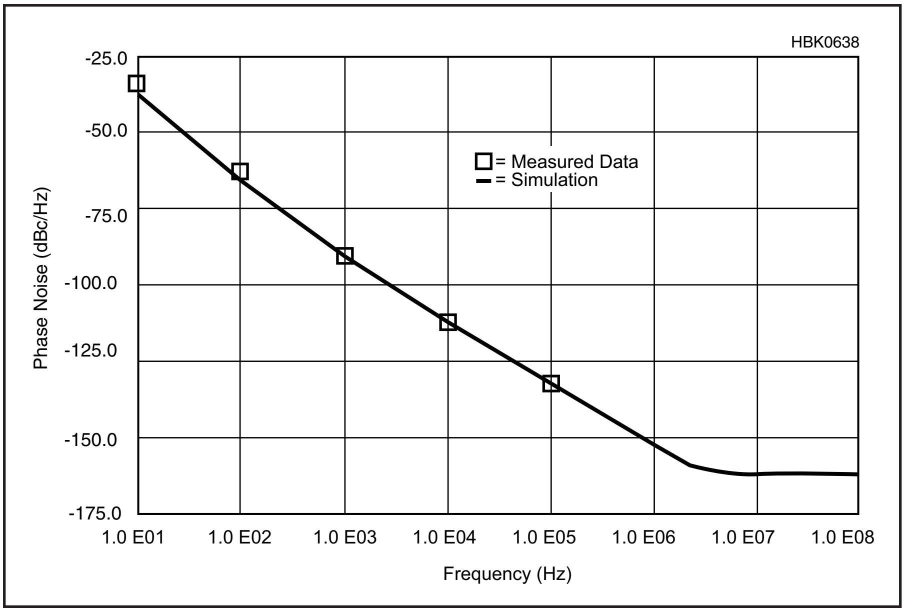


Figure 6.33 — Comparison between predicted and measured phase noise for the oscillator shown in Figure 6.32.

be very difficult to analyze at any frequency as shown by simulations of a low-MHz phase shift oscillator and a 10 MHz Colpitts oscillator in the referenced papers.

Validating the harmonic balance approach, **Figure 6.32** shows a BJT microwave oscillator entered into the schematic capture module of a commercially available HB simulator (Ansoft *Serenade 8.0*); **Figure 6.33** compares this oscillator's simulated phase noise to measured data. HB analysis gives similarly accurate results for mixers.

6.3.4 RF Simulation Tools

PSPICE: OrCAD *PSPICE* is a popular version available from Cadence PCB Solutions (**www.orcad.com**) and a "Lite" version is available for free download. (The Lite version is limited in the size and complexity of designs that can be created.) *AIM-spice* (**www.aim-spice.com**) is a PC version of *SPICE* with a revised user interface, simulation control, and with extra models.

A student version can be downloaded. Table 6.1 earlier in this chapter shows other free *SPICE* offerings.

There are a number of PC based *SPICE* programs in the $1000 range, but they are designed more for switching power supplies and logic circuits optimization than RF. *ICAP4* (**www.intusoft.com/demos.htm**) and *MICROCAP* (**www.spectrum-soft.com/index.shtm**) both have demonstration/evaluation versions available for download.

Agilent, AWR, Ansys, and Synopsis offer very modern mixed-mode CAD tools and they combine the concept of *SPICE* with the advanced technologies. These are professional quality tools, but if one can arrange to make use of them through a friend or associate, the results are worth investing the time to learn their use.

Linear Technologies also offers a free version, *LTSPICE*, in support of its switchmode products but handles all types of circuits and has a large number of users. (For a *QST* tutorial and a set of references and downloadable component libraries, see the Reference entries for Silver and Würth Elektronik.)

6.4 Electromagnetic Analysis of RF Circuits

Computers can be used to solve the equations that govern electromagnetic fields. This first became technically feasible in the late 1970's and is now a well-developed technology. A wide variety of software is available, all developed over decades of intense research. Today, you can take advantage of the fruits of this research without learning the equations and without writing any software. In this section we provide some historical background, discuss the different types of EM analysis techniques, and outline how electromagnetic software is used today. Then we show how you can use some of these tools for your RF designs.

6.4.1 Historical Background

In 1865 James Clerk Maxwell, **Figure 6.34**, published the equations that govern electric and magnetic fields. Maxwell's equations, as they are now called, put in solid mathematical terms the following principles:

1. An electric field that changes with time creates a magnetic field that changes with distance and is exactly at right angles to the electric field.
2. A magnetic field that changes with time creates an electric field that changes with distance and is exactly at right angles to the magnetic field.

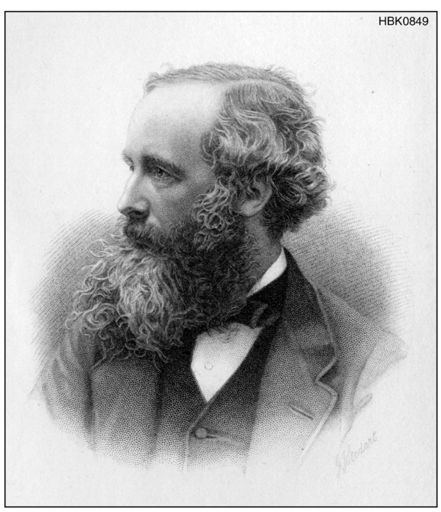
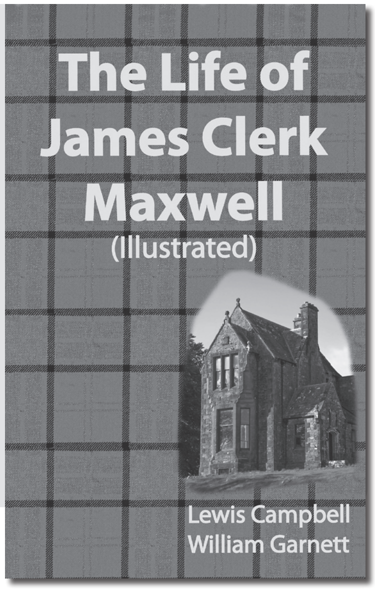


Figure 6.34 — James Clerk Maxwell (1831 – 1879) gave us the equations which govern the interaction between changing electric and magnetic fields. These equations can be solved on a computer, revealing how our RF and microwave circuits will perform.

Maxwell quickly recognized an important implication of his equations. If both an electric and magnetic field could somehow exist at the same time and at the same place and be exactly at right angles to each other, both changing with time and with distance in just the right way, each would continually create the other and a propagating wave would result. Using his equations, he calculated the speed of that hypothetical "electromagnetic" wave. Amazingly, he saw that it was virtually identical to the mechanically measured speed of light. His conclusion was earth shaking, "that light is an electromagnetic disturbance propagated through the field according to electromagnetic laws." However, Maxwell felt that the chances of using electricity to actually

A Biography of Maxwell

The best-known biography of James Clerk Maxwell is the first one that was written, *The Life of James Clerk Maxwell*. The authors, Lewis Campbell and William Garnett, knew Maxwell personally and they concentrate on his youth. We re-live young Maxwell's wonder at multi-colored light reflecting from an oil slick on a puddle of water; his experiments with Iceland Spar and thus learning about polarization of light; and other personal experiences that lead to his becoming, along with Einstein and Newton, one of the three greatest physicists of all time. Unique material from the second edition is also presented along with a few pages of modern material on Maxwell. A delight to read, you will treasure this book forever.



create such a wave would be difficult if not impossible. For more information about Maxwell, go to the James Clerk Maxwell Foundation website (**www.clerkmaxwellfoundation.org**). Several very readable biographies are available, including the excellent text in the **sidebar**.

After Maxwell's death, a German professor, Heinrich Hertz, was experimenting with a spark gap. Almost by accident, he realized he had succeeded in creating and detecting electromagnetic waves. He then refined how he created and detected the waves and conducted experiments reflecting, diffracting, focusing, and polarizing electromagnetic waves. This provided definitive proof that Maxwell's equations are correct. However, Hertz felt there was little if any practical use for such waves. As we all know, an Italian entrepreneur, Guglielmo Marconi, thought otherwise and was spectacularly successful in commercializing use of electromagnetic waves for communications. Today, every time we key our radio, work a DX station, or even just talk on our cell phone, we honor these amazing electromagnetic pioneers.

Figure 6.35 shows how an electromagnetic wave propagates. We key our handheld transceiver (HT) and it produces an RF (i.e., changing with time) voltage across the antenna terminals. For example, if it produces 10 volts, and the terminals are 1 cm apart, we have an electric field of 10 volts/cm. This is indicated by the long vertical arrow coming out of the HT. (The arrow indicates the strength and direction of the electric field right at the antenna terminals. The electric field does not physically extend out of the HT!) This electric field is changing with time, so it creates a magnetic field that changes with distance and is at right angles to the electric field. Since the electric field at the antenna (that creates the magnetic field) is changing with time, the resulting magnetic field also changes with time. That means the magnetic field then creates an electric field that changes with distance. The electric field and magnetic field keep creating each other, resulting in an electromagnetic wave. The wave travels into space until it runs into something.

Science marches on and modern physics has also shown us that Maxwell's equations are wrong. You may be familiar with the concept of a *photon*, the particle of light. A photon is indivisible. We cannot have half of a photon. The light we see can only be composed of an integer number of photons. A sensitive photo-multiplier tube can receive one photon, or two, or three, but never 2.853 photons.

Maxwell's equations, summarized in the **sidebar**, form a *field theory*. There are no photons in Maxwell's equations. Mathematically, we can easily write the equations for an electromagnetic wave with the energy of 2.853 photons. But we know that such a wave cannot exist. It is here that something called *quantum electrodynamics* (QED) takes over. This is part of quantum theory. The mathematics gets really, really hard. No user-friendly software exists for solving the QED equations. QED is wildly weird…but it actually works! See the sidebar presenting a very readable explanation of QED by its co-inventor, Richard Feynman. But fortunately, for virtually everything we do, it is not needed. The number of photons is so incredibly huge, we can just pretend that we are dealing with electric and magnetic fields. So, we use Maxwell's equations for our work with radio waves. It is this last point that we discuss in this chapter.

We are especially fortunate to have Maxwell's equations. In nearly all other fields of science and engineering, exact solutions to practical problems are often impossible. In contrast, given a precisely known passive RF circuit, we can use Maxwell's equations to get precise results for how it will operate. (Active circuits, typically with transistors, are another matter. This is dealt with elsewhere in the book.) Appropriate circuits include fil-

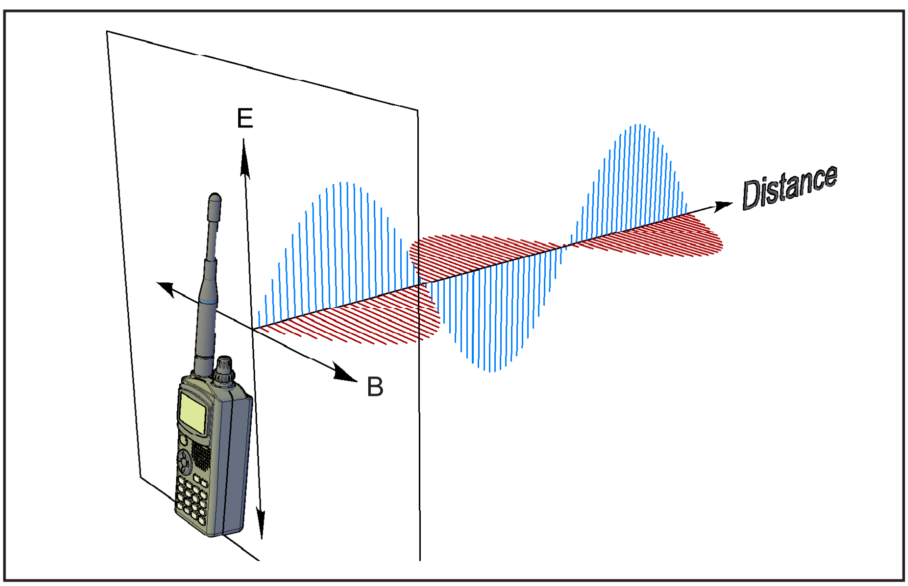


Figure 6.35 — We create one of Maxwell's electromagnetic waves when we key our HT. That generates a voltage across the antenna terminals, which creates a time-varying electric field, which creates a magnetic field that varies with both time and distance, which creates more electric field, which creates more magnetic field, and on and on. One and one-half cycle of an electromagnetic wave is shown.

The Basic Elements of Maxwell's Equations

The complete set of Maxwell's equations can be described as, "A set of four simultaneous partial differential equations operating on four vector fields in four dimensions and driven by two scalar fields." The four dimensions are the three spatial dimensions (x, y, z) and time. The four vector fields are electric, **E** and **D**, and magnetic, **H** and **B**. The two scalar fields are electric and magnetic charge. (Magnetic charge has yet to be discovered, but that does not stop the mathematics.)

You may also know these equations as laws named for scientists who originally discovered the relationships. It was Maxwell who brought them all together into one set of equations that define electromagnetics. (For an explanation of vectors, scalars, and what the various symbols mean, see "Radio Mathematics" in the online content.)

Gauss's Law for Electric Fields:

$$\nabla \bullet \mathbf{D} = \rho$$

Gauss's Law for Magnetic Fields:

$$\nabla \bullet \mathbf{B} = 0$$

Faraday's Law: $\nabla \times E = \frac{\partial B}{\partial t}$

Ampere – Maxwell Law:

$$\nabla \times H = \mathbf{J} + \frac{\partial D}{\partial t}$$

Where ρ represents the density of electric charge and **J** represents a distribution of electric current (moving charge). There are also forms of Maxwell's Laws that are written as integrals, that incorporate relativity, that use mathematical objects known as tensors, and various other ways. The four equations above, though, are the most widely known.

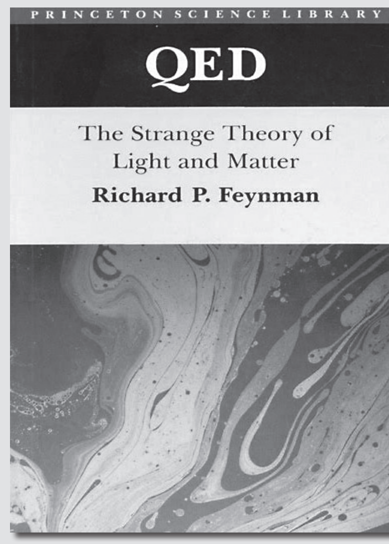
J is a vector because current has a direction. A distribution of *electric* charge changing with time is exactly what makes electric current. So, for RF, we can use either current or charge (both of which can vary with time) to drive the equations. All sorts of other abstract mathematical quantities are also helpful, including some strange beasts known as vector and scalar potentials. Fortunately, we can often simplify things by assuming that **D** is proportional to **E** (**D** = ε**E**), and **B** is proportional to **H** (**B** = μ**H**). Then, we have only have two vector fields to deal with, but even that is still a big problem.

By doing this extra math, Maxwell's equations and all their vector and scalar fields are turned into a big pile of discrete numbers that can be stored on and crunched in a computer. It has only taken a couple decades with hundreds — if not thousands — of researchers all over the world, but it seems like today we are pretty much there.

For a detailed introduction to Maxwell's equations, if you have some calculus background see *A Student's Guide to Maxwell's Equations* by Daniel Fleisch from Cambridge Press. It is an excellent tutorial on this important topic.

Quantum Electrodynamics (QED)

Maxwell's electromagnetic theory has no place for photons, i.e., "particles" of light. QED predicts everything Maxwell's theory does, and it does so assuming that electromagnetic waves are composed of particles, particles that we call photons. *QED — The Strange Theory of Light and Matter* by Richard Feynman, the world-famous Nobel laureate and popularizer of science, explains QED in simple terms that anyone can understand. When you read this book, which is available from many sources, be prepared to enter a world that is unlike anything we have ever known in our day-to-day lives.



ters, matching networks, couplers, power dividers, connectors, transformers, and any passive structure involved in handling RF.

All this gradually started to become possible in the 1970's. Computers were just starting to come into wide use. Large corporate time-share computers saw some of the first EM software ever developed. The first IBM-PC put serious computing power (for the day) in front of a lot of eager, bright-eyed experimenters. We were off to the races!

6.4.2 Types of Electromagnetic Analysis

Maxwell liked to think "analogically," reasoning that completely different areas of physics and nature somehow use similar equations. For example, he suspected that incompressible fluid flow (think of a submarine powering through the ocean depths) and electrostatics (think about rubbing a balloon on your hair) use the same equations. Really? How could that possibly be? It sounds ridiculous, but he gave it a try anyway and he derived the equations for incompressible fluid flow and found that they magically worked just fine for electrostatics… Absolutely mind-boggling when we think about it. That was his first step in coming up with the full set of Maxwell's equations.

We can use analogical thinking in solving Maxwell's equations on a computer too. The problem is that we have continuous electric and magnetic fields everywhere. In contrast, a computer can only store numbers, and we must be careful that we do not overrun the computer's memory capacity. The same is true for images. To get an image into a computer, we take a picture with a camera. The camera converts the continuous image into discrete *pixels*. There are three numbers (red, green, and blue colors which Maxwell was the first to figure out) for each pixel. In this form, our picture is just a bunch of numbers. We can now store the picture on any computer and use software to process the picture as we like. So, if analogical thinking works for this case, maybe we can convert our electromagnetic problem into pixels.

And indeed, we can. First, the fields and currents are *discretized*, that is, they are divided into small subsections, just like an image being converted to pixels. However, an image is usually two-dimensional while EM fields are three-dimensional in space, and they vary with time. Thus, we actually have to deal with pixels in four dimensions. The analogy would be a holographic movie. To make the problem worse, we have to discretize the electric field, the magnetic field, the charge (or, equivalently, the current), and even the circuit itself. That is a lot of numbers!

Fortunately, today's computers have gotten really good at storing lots of numbers. Once we have the numbers stored, then, just like when we do image processing, we can use software to do what we want with the numbers. This is where the last few decades of

EM research has concentrated. There are two ways to do this: volume meshing and surface meshing. We discuss each below.

VOLUME MESHING — FDTD

The easiest to understand is *finite difference time domain* (FDTD) analysis. In its simplest form, we divide the entire volume of the RF structure into tiny rectangular hexahedral cells (six sides, e.g., cubes, rectangular cuboids, etc.). This is called *meshing*. Note that we must also mesh all of space that might have any EM fields. This includes any air (e.g., free space) surrounding the structure. Next, we assume an initial condition. This is the RF input into the structure, like when you first key your transmitter. Let's say we have 1 volt of electric field at the input right at the start (time: $t = 0$). This input is a single impulse, we put it there for only a single instant (i.e., time step). This is like hitting a bell to start it ringing.

And indeed it does ring. Changing the input voltage, which we placed on an edge of one of the hexahedrals, from 0 volts to 1 volt right at the start gives us a changing electric field. In addition to meshing the structure, we also have to discretize time, so we pick a tiny time-step. Then the software uses Maxwell's equations to calculate exactly how much changing magnetic field is created by the changing electric field at the input. In addition, the software knows — because of Maxwell's equations — that this changing magnetic field must be at right angles to the changing electric field and it must also change with distance. This changing magnetic field is now created on the edges of every hexahedral surrounding the hexahedral edge that has the changing electric field input.

In the next time-step, we have both changing magnetic fields and changing electric fields near the original input. Each hexahedral edge with a changing magnetic field creates a changing electric field in all surrounding hexahedral edges. Each changing electric field creates a changing magnetic field in all surrounding edges. Using Maxwell's equations, the software calculates how much electric and magnetic field is created and the process is repeated. Soon (at the speed of light) electric and magnetic fields propagate throughout the entire structure. For every hexahedral in the structure, the computer now stores a list of numbers that tell us how the electric and magnetic fields change as a function of time. If there is a little bit of loss (and in real life, there is always at least a little bit of either conductor, dielectric, or radiation loss), the ringing electric and magnetic fields gradually die out to zero, and the solution is complete.

Often, we are not interested in the electric and magnetic fields in the entire circuit. For example, we might be interested only in an input and an output. We still need to calculate all the field information everywhere to get to the final desired result. But at any given time-step during analysis, only the last several time-steps of complete field information are needed. In this case, we discard all internal field information after it is no longer needed for calculating the next time-step. In other cases, we actually want to see the complete internal fields as they develop during the analysis and we keep that information available, usually by storing it conveniently on the computer's hard drive. **Figure 6.36** shows a typical FDTD result. Bright shades indicate high electric fields as a continuous RF wave at a single frequency propagates along a complex rectangular waveguide circuit. (**Color versions of the figures are available as online content.**) The field strength is shown across the entire cross-section of the guide at the ports, and on a horizontal plane in the middle of the waveguide along its length. The hexahedral meshing is so fine that it is not visible in this image.

A little more detail is important. Due to problems calculating rate-of-change with time (known in calculus as a *time derivative*) and rate-of-change across space (a *gradient*), there are actually two full and nearly identical discretizations of the structure being analyzed. Both discretizations cover the same volume. The only difference is one discretization is offset from the other by one-half of a hexahedral. The changing electric field is calculated on one discretization, and the changing magnetic field is calculated on the other. This means that every sample of magnetic field is midway between all adjacent samples of the electric field. Likewise, the electric field calculation is always for points that are midway between all adjacent samples

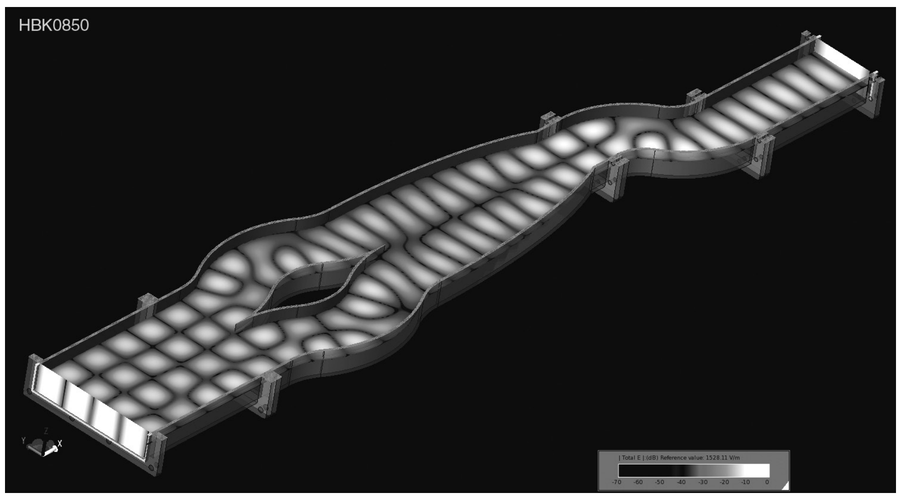


Figure 6.36 — A complex rectangular waveguide mode converter as analyzed by FDTD. Electric field is shown with bright shades indicating high field strength. Input is on the right. (Courtesy Remcom, with permission.)

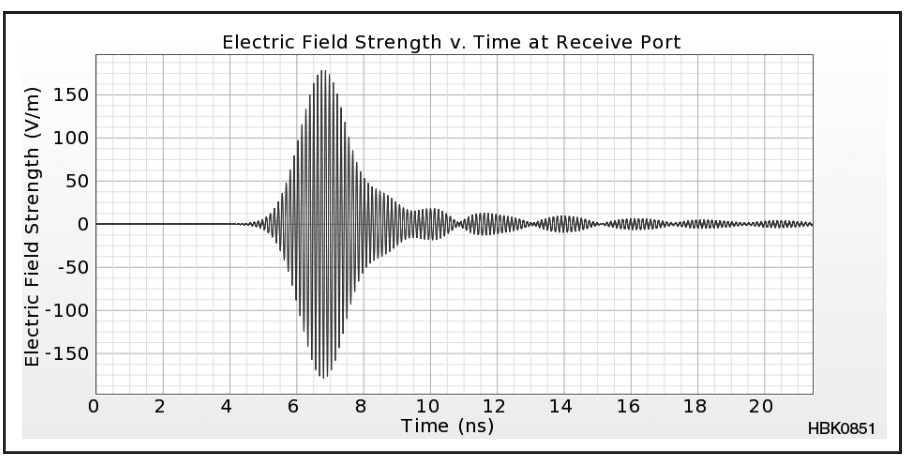


Figure 6.37 — The FDTD calculated time-domain response to a brief RF pulse on the input of the mode converter of Figure 6.36 resembles the ringing of a bell. This is called the *impulse response*. (Courtesy Remcom, with permission.)

of the magnetic field. Without this special modification, there can be stability problems.

In practice, the input is typically not a simple impulse. A pure impulse — an infinitely short pulse — can cause numerical instability. So, we typically use a "gentler" input, like a Gaussian pulse. This kind of pulse is the well-known "bell curve" used to curve test grades or to represent the probability of many random processes. If we are especially interested in results over a specific band of frequencies, we can use the Gaussian pulse to modulate an RF carrier centered on the band of interest.

For most circuits, the electric and magnetic fields at a given point, plotted as a function of time, look just like what we would expect for a ringing bell, or a ringing resonant LC circuit, **Figure 6.37**. For this case, the input is a Gaussian pulse modulating an RF carrier. The rapid oscillation is from the RF carrier, and the envelope is the result of the "bell" ringing. This is closely related to what professionals call the *impulse response*, the impulse being a single instant of electric field that started all the fields going. While impressive, this is not of much use in designing our circuit. We typically need a frequency response.

The input pulse that started everything ringing actually has a wide band of frequencies present in it. Think about the spark gaps that Hertz and Marconi, as well as early hams, used to generate electromagnetic waves. They were very broad band. If we think of our circuit as being a filter, the output is just a filtered version of that broad band impulse we put on the input. The spectrum of that output is the filter response that we want to see. The computer can take the numbers that were calculated for the impulse response and simulate a spectrum analyzer by using a mathematical algorithm, i.e., a mathematical recipe, called a Fourier transform. The fastest way of doing a Fourier transform on a computer is creatively called the fast Fourier transform, or FFT.

The software then performs an FFT on the impulse response, and we have the current and voltage output of our circuit. This output is then converted to S-parameters and plotted, **Figure 6.38**. The indicated *S21* is the ratio of the amplitude of the output voltage wave (i.e., coming out of port 2) to the amplitude of the input voltage wave (i.e., going into port 1). *S11* is the reflection coefficient, i.e., the ratio of the reflected wave to the input wave.

There are a number of time domain approaches similar to FDTD. Another approach is called TLM, *transmission line method*. Maxwell's equations are modeled as a fine mesh of short transmission lines. However, the basic idea of all time domain volume meshing EM tools is similar to what we describe above. The entire volume is meshed, an input pulse is applied, the time-domain response is calculated one time-step at a time, and then the output is (usually) converted from time-domain to frequency-domain.

VOLUME MESHING — FINITE ELEMENT METHOD (FEM)

Mechanical engineers use the *finite element method* (FEM) extensively. Materials can only take so much stress and strain before they fail. FEM allows designers to rapidly try out many different structure designs on the computer. Then they build the best one, where "best" might be based on the most strength for the least weight, or size, or cost, etc.

Stresses and strains are vector fields, just like electromagnetic fields. A *vector* has a magnitude and a direction. For example, an electric vector field at a given point might be "5 V/m in the *x* direction." A "field" means that the vector is defined over a region, like inside a waveguide, or even everywhere in all of space. So, by analogical thinking, it seems we might be able to apply FEM to our RF and microwave problems too. And indeed, we can, but the devil is in the details. Like all numerical EM techniques, it has taken world-class researchers several decades to reach today's high level of maturity. (A list of math tutorials is provided in "Radio Math" in the online content.)

FDTD is a *time domain* analysis. In other words, the EM analysis evaluates the EM fields as they evolve in time. We get the full frequency response by performing an FFT on the calculated impulse response.

For RF work, FEM tools typically provide *frequency domain* results. In this case, we specify a frequency, and the FEM software analyzes the response at that frequency. Then we specify a second frequency and rerun the analysis for it. This process repeats until we have results for all frequencies of interest.

The advantage of frequency domain analysis is that we analyze only the frequencies that we want. In addition, we now need to solve only for the three spatial dimensions. We already know that the fields all vary in time with a sine wave at a given frequency. We are however, restricted to working with linear circuits. Given input at a single frequency, linear circuits have only that frequency present everywhere in the circuit. An amplifier or a mixer would have several frequencies bouncing around even if only one frequency is input. Given only one frequency is present everywhere, our EM analysis now needs to solve only for the magnitude and phase of that sine wave. The advantage of time domain analysis is that we get all the frequencies (by using the FFT on the impulse response) with a single analysis.

FEM starts just like FDTD, by discretizing the structure we are analyzing. As with FDTD, FEM must mesh all space that contains electromagnetic fields, including air. For RF work, FEM tools mesh our structure into tetrahedra. **Figure 6.39** shows an example of a multi-pin connector meshing. The rectangular hexahedral mesh used for structures in FDTD often leaves a "staircase" appearance. The staircase can be reduced by making the mesh very fine, but this increases memory requirements and analysis time. With pictures, the pixels also give a staircase appearance. But with today's cameras, the pixels are so tiny (and there are so many of them) that they are hard to see without magnification.

FEM can use tetrahedra of any size. Thus, curving surfaces appear to be meshed into triangles (the exposed surface of each tetrahedra). This is often a better representation of the surface than is possible with the hexahedral cells of FDTD. In addition, since different size tetrahedra can be used as desired, the accuracy of the mesh can be varied. Very

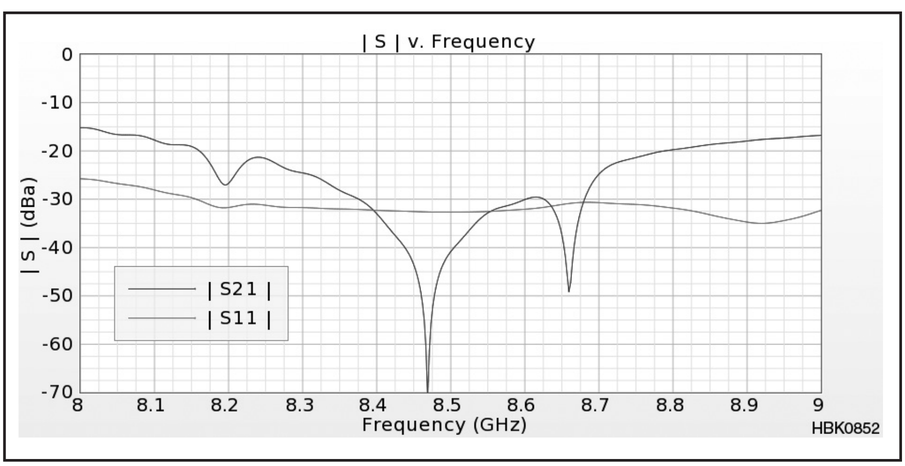


Figure 6.38 — Performing a Fourier transform on the circuit response is equivalent to using a spectrum analyzer to see the frequency domain response of the mode converter given the broad band impulse used to excite the filter. (Courtesy Remcom, with permission.)

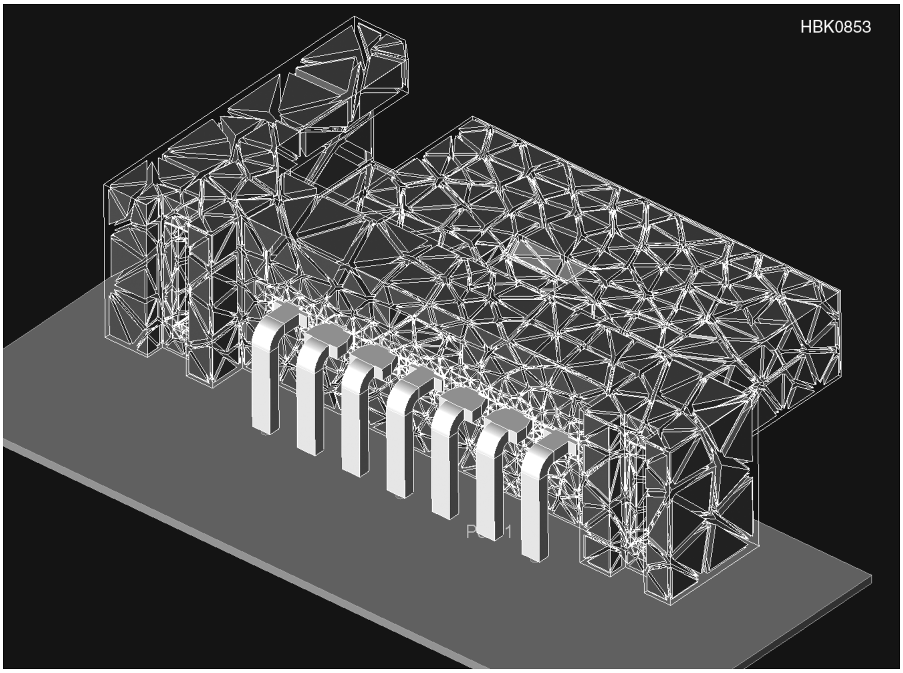


Figure 6.39 — Extremely arbitrary structures can be analyzed with FEM. Here we see the meshing for a PCB multi-pin connector. High accuracy analysis at high frequency is critical when high performance is needed. The entire volume is meshed with tetrahedra. For clarity, the tetrahedra used for meshing are outlined only where they intersect with a surface of the connector. (Courtesy Keysight, with permission.)

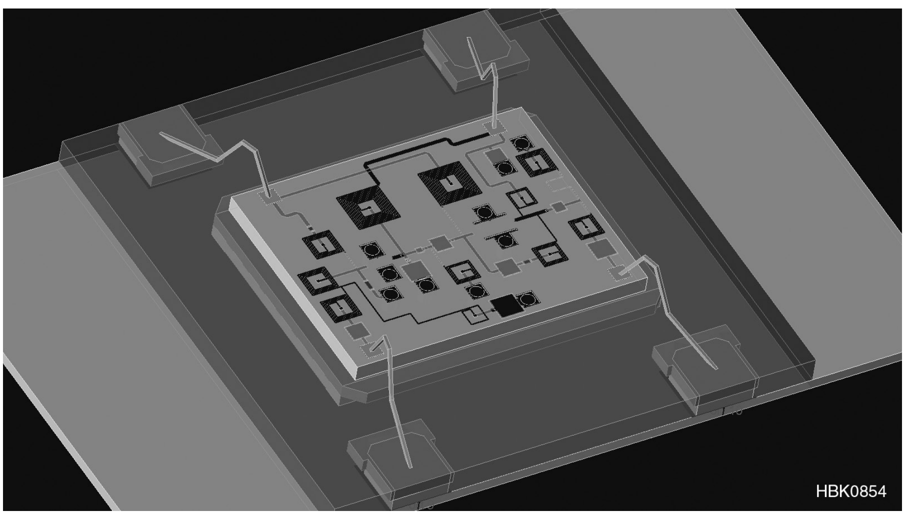


Figure 6.40 — This RF integrated circuit is mounted on a carrier with wire bonds making connections to the outside world. This is a view of the circuit prior to meshing. FEM can analyze the entire structure including all electromagnetic interactions. (Courtesy Keysight, with permission.)

tiny tetrahedra can be used where the highest accuracy is needed, and large tetrahedra can be used elsewhere. In fact, an initial solution can be performed, and the error estimated everywhere. Adaptive meshing then refines the mesh (i.e., makes more and smaller tetrahedra) wherever the error is too high. Thus, the fine mesh, which can consume a lot of analysis time and computer memory, is used only where it is needed. Care should be taken because the location of the fine mesh might require modification depending on the frequency of analysis and on how the structure is excited.

The mathematics of FEM can get very involved, and there are many ways to implement it. High level, we must cast the problem into what is called a *variational form*. For example, an equation for the energy in, say, a microwave filter, given some boundary conditions (like an RF voltage source on the input), is a variational form. FEM then finds the EM fields consistent with Maxwell's equations and the given boundary conditions that give the minimum total energy. That field is the solution.

We can intuitively see why the minimum energy solution is the correct solution. Imagine a length of thin, moderately wide conductor on a printed circuit board. Next, we place static electric charge on the conductor. Electrons are all negative, they basically do not like (repel) each other. They all try to push each other to opposite edges of the conductor, at least as best as they can. We now have the highest charge on the edges of the conductor and lower charge towards the center of the conductor. This configuration of charge gives the minimum potential energy for the charge. This is the solution that FEM finds. It makes sense: highest charge on the edges means we have the highest voltage there too. If we add too much charge, the sharp edges are where the arcs and sparks will come from (do not try this at home!).

Mathematically, the situation for RF is nearly identical. It is just that instead of charge, we have RF current flowing. The RF current flows most strongly on the edges of a planar (e.g., printed circuit board [PCB]) conductor. In fact, if we run too much RF current, the edges literally start evaporating. This can happen even with low power if the conductor dimensions are very small, for example, on a silicon RF integrated circuit (SiRFIC) as is widely used in cell phones. In this case, the evaporation process is called *electromigration*. SiRFIC designers add a layer of insulator on top of the RF conductors because of this problem. Otherwise, our cell phone would quickly fail. **Figure 6.40** shows a drawing of a typical RFIC made using GaAs (instead of silicon) mounted on a carrier. This was prepared using CAD (*computer aided design*) software and is ready for FEM meshing.

Thus, FEM is similar to an optimization problem. As such it is often iterative. It will try an initial (usually fast, coarse mesh, high error) analysis and evaluate the error (i.e., non-minimum energy) everywhere in that solution. Then the FEM software refines the meshing and refines the solution. This process repeats until the error is below a user-specified threshold. If you feel you need a better solution, just make the error threshold smaller. For high-accuracy work, it is best to refine the solution until all FEM calculated fields and currents appear smooth (error appears as noise in the fields). Sharp edges (as in PCB and SiRFIC) should have smooth high current. However, results can be useful even if error is still apparent in the field display. **Figure 6.41** shows a high-quality result for a complex coaxial connector-to-PCB transition.

As we can see above, a major strength of the volume meshing techniques, like FEM and FDTD, is that they can easily analyze just about anything no matter what shape it is. These techniques excel by being able to handle extremely arbitrary structures, provided (as with all EM analyses) the problem does not get too big for our computer memory or our willingness to wait for an answer.

Next, we consider a technique that can handle much larger problems, but only if the problem is appropriate for the technique.

SURFACE MESHING — METHOD OF MOMENTS (MOM)

Like FEM and FDTD, *Method of Moments* (MoM) was developed in the 1960s. MoM was fully formalized and popularized in the west by Prof. Roger Harrington of Syracuse University. He named it after similar work that had previously been performed in Russia. The equations needed to understand Method of Moments can be difficult. Fortunately, the concept is easy to understand.

Imagine we have a planar circuit, say a filter on a PCB. In the first step, we discretize (i.e., divide into small subsections) only the metal of the circuit. In contrast with volume meshing techniques, we do not mesh the substrate and we do not mesh the air around the substrate. We mesh only the metal of the circuit. Next, we put current (at a given RF frequency) on one subsection of that circuit. We calculate how much voltage that bit of RF current induces on all other subsections. We can view this as one tiny transmitting antenna (the subsection with current) and all the other subsections are tiny receiving antennas. Typically, high voltage is induced on nearby subsections, and lower voltage on more distant subsections. The effect of the PCB dielectric material is included in this calculation. (This is the step where the math can get really complicated. But all that has already been done. You don't have to worry about it!)

This calculation (i.e., current on one subsection induces voltage on all other subsections) is repeated for all subsections in the circuit, one at a time. All this information is stored on the computer in the form of a matrix (i.e., like a spreadsheet with rows and columns). Next, the computer puts current on all subsections at the same time. This, as previously calculated, induces voltage in all the subsections in the circuit. Every subsection has voltage induced in it from all the other subsections in the circuit. In fact, each subsection also has voltage induced in it due to itself. All the voltages induced in each subsection add together.

But a perfect conductor (which is our assumption for this description) cannot have any voltage at all across it! (Measure the voltage across any piece of copper. It is really close to short circuit, zero voltage. A perfect conductor would be exactly zero voltage.) So, the computer adjusts all the currents in all the subsections so that there is zero total voltage in all the subsections. (In practice, this is done by a process called *matrix inversion*.) Of course, the subsection that we are using for the input to the circuit must have some voltage across it. Otherwise, we will have zero current (and zero voltage) everywhere…no fun in that!

The currents in all the subsections that give us zero total voltage everywhere that there is

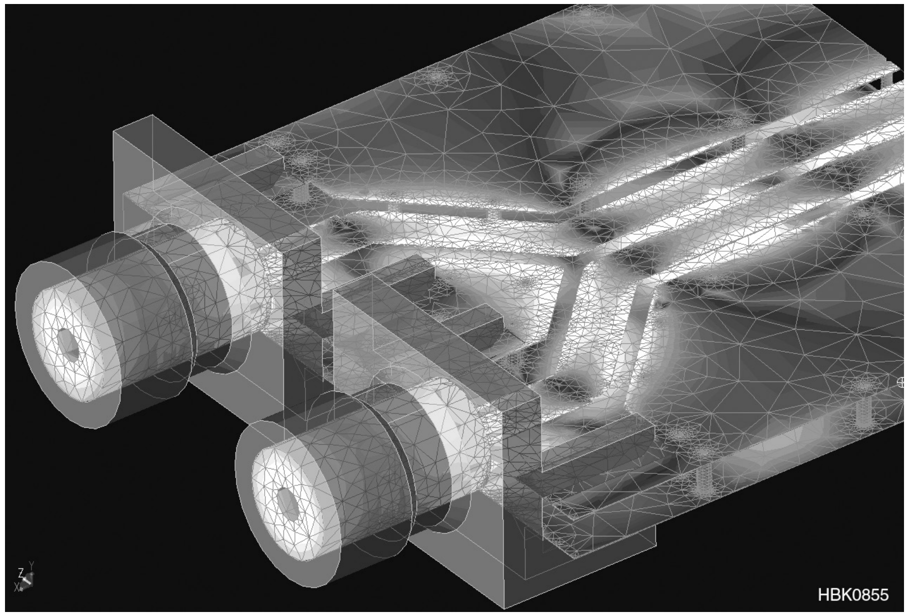
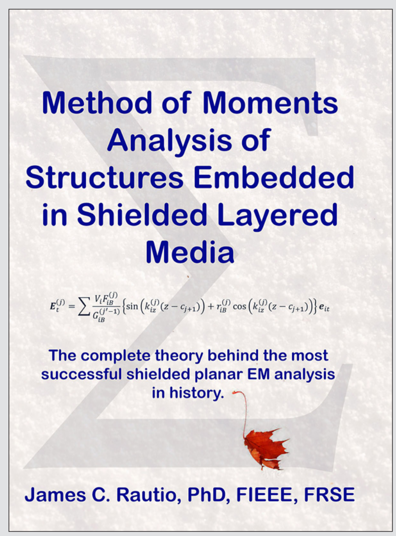


Figure 6.41 — FEM has meshed a circuit with two coaxial connectors transitioning to two PCB conductors. The PCB conductors then come close to each other to form the start of a coupled line. The shading shows current density on the conductors with bright shades indicating high current. This includes a ground plane on the underside, which we can see through the narrow slots between the PCB conductors and the ground plane that surrounds them on the top side of the substrate. This is known as *coplanar waveguide*, or CPW. Note that the current is smooth with high current on the edges, indicating a high-quality result. (Courtesy Keysight, with permission.)

The Method of Moments (MoM)

The math involved in the MoM uses vectors, matrices, and calculus no more complicated than various sine and cosine integrals. However, some derivations and results are lengthy and complicated. If you can handle this level of mathematics and write computer programs, then you can use *Method of Moments – Analysis of Structures in Shielded Layered Media* by James Rautio, AJ3K, as a recipe to write your own MoM program. MoM programs typically deal only with surface current (infinitely thin sheets of current). A recent development, fully detailed in this book, is how to use MoM for volume current, i.e., full 3D. Ultimately, this could supplement, or possibly supplant, the common volume meshing techniques like Finite Elements as MoM meshes only the structure of interest while other volume meshing techniques must also mesh all dielectric, including the air surrounding the structure of interest.



A *Sonnet* Tutorial using Port Tuning

Up to this point, we have described how EM software solves what is called the "analysis" problem. In other words, we have a planar RF circuit layout, and we need to analyze the RF circuit response. This was a huge advantage because before it became available, we would be forced to build a circuit and then measure it. Of course, it would never work quite the way we wanted it to work. Now what do we do?

Before EM software, we would get out the craft knife and little bits of copper and start tuning the circuit. Maybe one or another open circuited stub needed to be a little shorter. But if shorter made it worse, we realized we needed to make it longer. We got out the soldering iron. High-level designers could dab some silver epoxy on. Those days are gone, thank goodness.

With EM analysis, we just modify the layout on our computer to make the open stub (or whatever part of the circuit you think might improve the response) a little shorter or a little longer and then repeat the analysis. This is so much better than the old days!

As you might imagine, we are never satisfied. Each time we repeat the EM analysis, it can be a long wait, especially for the larger and more complicated circuits. We want something faster.

Over the last decade, a much faster methodology has developed. It is called *port tuning*. We can do things with EM analysis that we could never do with a circuit that we built. For example, a typical filter has a few resonators in it. These are often just half-wave long lengths of line, just like a dipole antenna, but often with nothing connected. If our filter has its passband a little bit low in frequency, we make the resonator shorter. If the passband is a little too high, we make it longer.

It turns out that port tuning lets us do this without having to repeat the entire EM analysis each time. We illustrate this using *Sonnet*®. If you would like to follow along, download the free *SonnetLite*® at **www.sonnetsoftware.com**. The following description is for *SonnetLite Version 18*. If you have a later version, the user interface might be somewhat different. If you are comfortable with mathematics and would like to explore the exact way that *Sonnet* implements the Method of Moments you can read the book referenced in the sidebar. This book also describes the recent developments that allow MoM to perform volume meshing analysis.

After you install the software, go through the "Getting Started" Guide under Help->Help and Manuals. It will take around one hour to do this. When you are finished, you can start your first port tuning project. The SON data files for the tutorial are included with this chapter's online material.

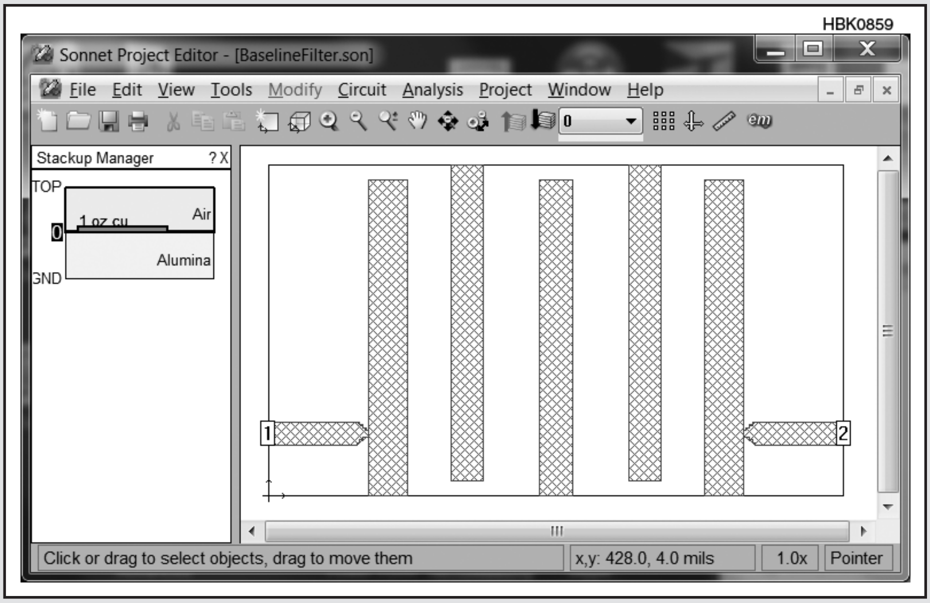


Figure 6.D — The baseline filter that we use to illustrate port tuning is made of quarter wavelength resonators, each grounded at one end to a perfectly conducting sidewall that is present at the edge of the substrate. (Courtesy Sonnet, with permission.)

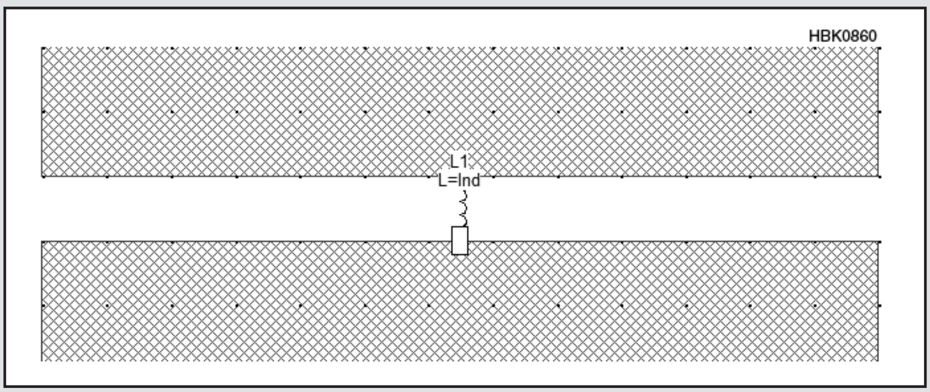


Figure 6.E — Following instructions in the text, we have placed a gap halfway along the length of the first (left) resonator. This is a zoomed detail of the inductor component that was inserted in that gap. The value of the inductance, "Ind," is a variable parameter that will be tuned. (Courtesy Sonnet, with permission.)

Upon completing the tutorial, you will have the layout of **Figure 6.D** on your *Sonnet* screen. The *Sonnet* file for this filter is "BaselineFilter.son" in the online content. This is an interdigital filter. The resonators are one quarter wavelength long with one end of each resonator shorted to ground. The antenna equivalent is a quarter wavelength vertical antenna, which also has one end shorted to ground. (Everything that touches the edge of the substrate in *Sonnet* is connected to the shielding box that contains the circuit, and this is ground.)

Select one of the resonators to tune. For this example, we select the first (left) resonator, but any resonator can be tuned with this method. Lengthening a resonator lowers the resonant frequency and shortening a resonator increases the resonant frequency. To tune the length of the resonator you selected, we need to remove a very small portion of the resonator creating a narrow gap. Then we replace the piece we removed with a small, ideal inductor. Repeating an EM analysis for every little change in a layout takes time. But we can change the small inductor and get nearly instantaneous analysis results.

Before proceeding, it is worth your

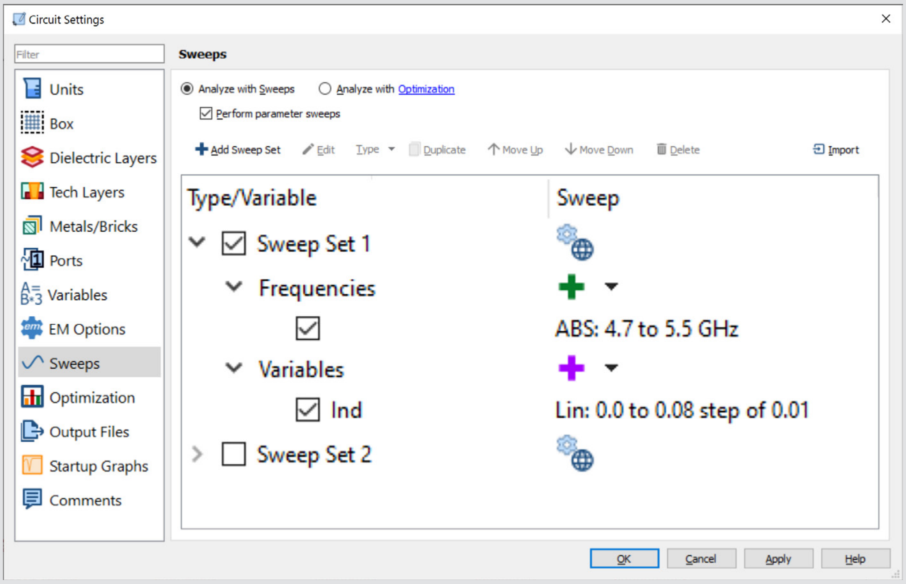


Figure 6.F — This is how your Sweep settings should look after setting up a Parameter Sweep to analyze the filter for a variety of inductor, "Ind," values. (Courtesy Sonnet, with permission.)

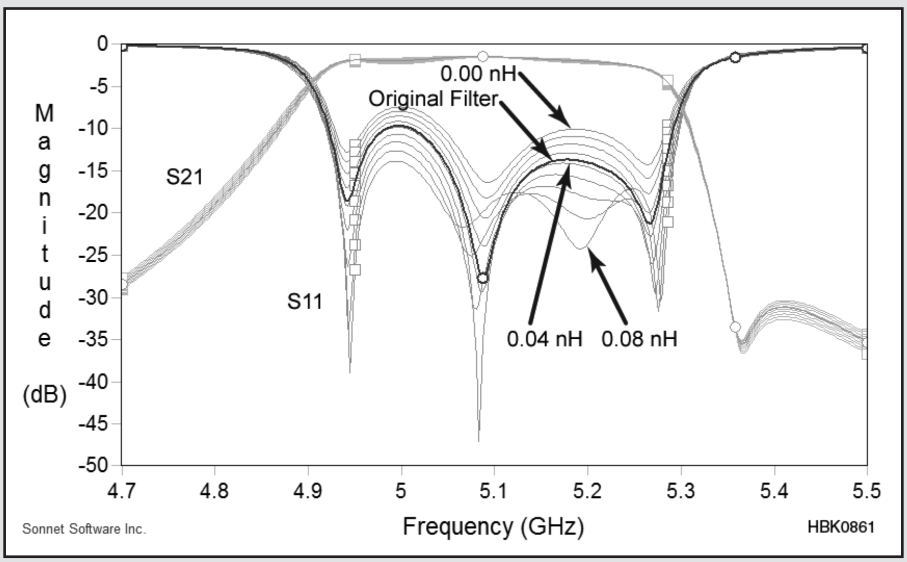


Figure 6.G — Results of running a parameter sweep for the value of "Ind" from 0 to 0.08 nH shows that 0.04 nH gives almost the same response as the original (untuned) filter. A value of 0.08 nH gives an improved reflection coefficient (S11) and indicates the first resonator should be made 0.002 inches longer (see text for details). (Courtesy Sonnet, with permission.)

time to read the section about Components in the *Sonnet* User's Guide (Help->Help and Manuals). We are inserting an "Ideal Component," an inductor. Here is step-by-step description for Version 18 of *SonnetLite*:

1) Select Edit->Divide Polygons.
2) Cut across the width of the desired resonator about halfway along the length (the exact position is not critical).
3) Select Edit->Divide Polygons again.
4) Cut across the width of the resonator again, one cell away from your first cut. A "cell" is the smallest bit of metal allowed for a given analysis. The cell size is indicated by a grid of dots. For this project, the cell size was set to 0.002 inches square. You will likely need to zoom (CTRL+ to zoom centered on the cursor.)
5) Select the one-cell-long piece of resonator and press Delete. This leaves a narrow gap across the width of the resonator.
6) Select Insert->Component->Ideal. Specify an inductor. Set the "Ground" to "Floating." We are going to add a small inductor across the narrow gap we just cut. For the value of the inductor, we will use a *Sonnet* variable. Instead of typing an actual number for the inductor value, type "Ind" as the name of the inductor value variable. "Ind" is a variable that we will sweep, i.e., *Sonnet* will automatically change the value of the variable and repeat the analysis. Next, click OK, and you are asked for more information about the variable. Nothing more is needed, so just click OK.
7)When the information windows you were just using disappear, click in the center of one side of the gap and click a second time in the center of the other side of the gap. This places the inductor across the gap.

Your layout, zoomed to the region around the gap, should resemble **Figure 6.E** ("Port Tuning Example.son" in the online content). As you have likely guessed, we are going to use the inductor that we added to this layout to replace the metal that we took out to make the gap in the resonator. If it turns out a little more inductance gives us a more desirable filter, then we make the resonator a little longer. A little less inductance means we should make the resonator a little shorter. Here, even negative inductors are allowed.

First, we must find out how much inductance corresponds to how much change in resonator length by using a parameter sweep. To do a parameter sweep, read the section in Help->Help and Manuals->User's Guide->Parameter Sweep. In *Sonnet*, click on "Settings" (that icon is near the middle of the top of the window when the circuit layout is displayed) and then on "Sweeps" and add a linear parameter sweep for "Ind" from 0.0 to 0.8 nH with a step of 0.01 nH. Your settings window should look similar to **Figure 6.F**. All of these steps have been completed in the file "Port Tuning Example" in the online content. When you are finished, save the project file and click on the "Add to queue" icon (which is right next to the "Settings" icon), select "New queue," click on the run icon in the Queue window and the analysis will proceed.

The key point in this example is that the EM analysis itself is only performed once. That is because the EM layout is the same for every inductor value. The only thing that changes is the inductor

(continued on next page)

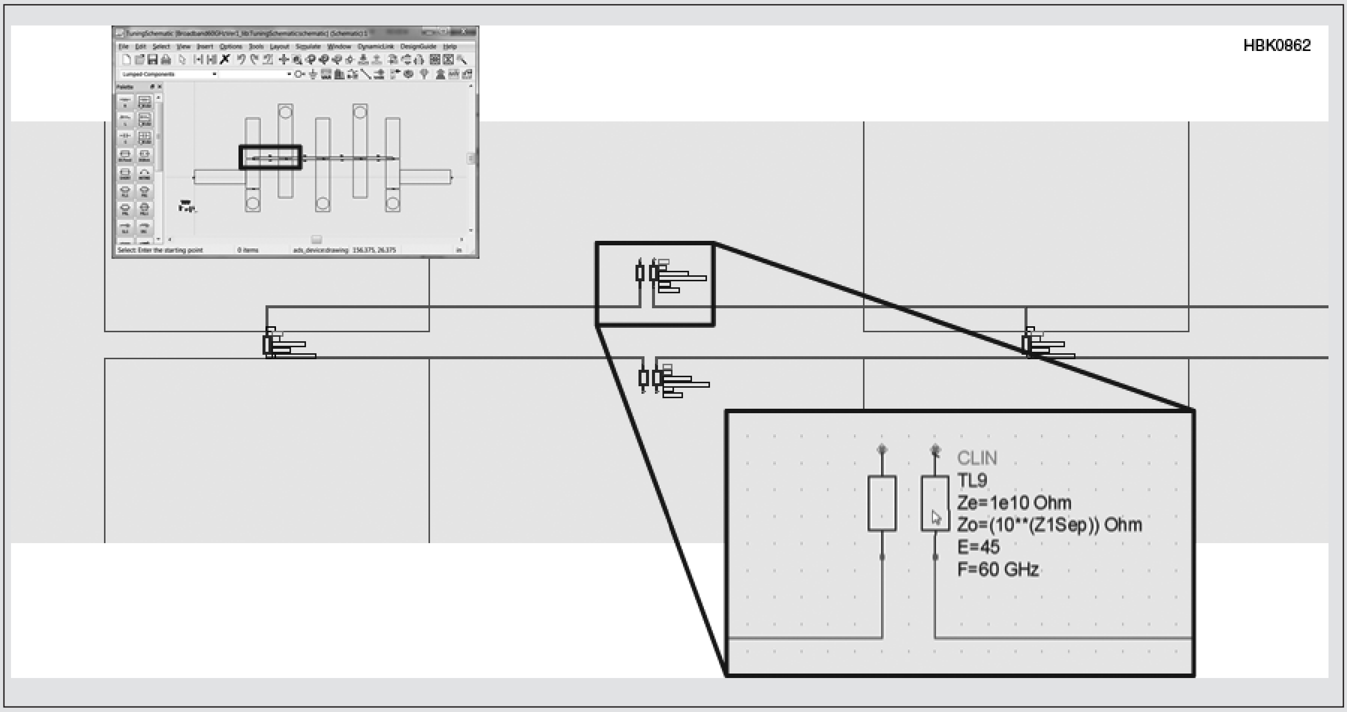


Figure 6.H — For professional use, port tuning can be used to tune all resonator lengths and couplings between resonators in order to obtain the desired response. This illustrates a filter tuned using Keysight ADS and Sonnet EM analysis. The lower right detail illustrates a circuit theory coupled line used to fine tune the coupling between two resonators. [Courtesy Sonnet, used with permission]

value. Analyzing the inductor connected into the EM analysis (which is done using something called *nodal analysis*) is evaluated rapidly. If you carefully watch the display during analysis, you will see a few seconds delay while the EM analysis executes, then all the remaining results scroll out almost instantly.

Figure 6.G shows the results of the parameter sweep, sweeping the value of the port tuning inductor from 0.0 to 0.08 nH with a step of 0.01 nH. Also plotted is the original filter response, the one that we started with before adding the tuning ports. The original filter reflection coefficient is highlighted. Reflection (S11) is about –10 dB. It turns out that the curve for Ind = 0.04 nH gives a very close match to the original filter. Thus, we conclude that the 0.002-inch gap in the resonator is equal to about 0.04 nH.

Next notice that the curve for Ind = 0.08 nH gives a reflection that is 5 dB better, or –15 dB. This tells us that we can improve the filter return loss by 5 dB if we make the input resonator another 0.002 inch (i.e., another 0.04 nH) longer.

If we take the original filter and increase the length of that resonator by 0.002 inch, a confirming EM analysis, “FinalFilter.son” in the online content, gives us almost exactly the same response as the port tuning filter with Ind = 0.08 nH. That is the filter we would build.

There are some other practical considerations when port tuning real filters. For example, in this filter the distance from the open end of the resonator to the sidewall is 0.010 inch. Making that resonator longer by 0.002 inch reduces that distance. That increases the open-end capacitance, which lowers the resonant frequency a small amount. This would become critical if the resonator must be lengthened by, say, 0.008 inch. The gap to the wall would only be 0.002 inch. Lengthening the resonator by even more would require that we move the wall.

Also, in practice, we would lengthen the resonators symmetrically. In other words, if we lengthen the input resonator, we would lengthen the output resonator by the same amount. This can be done by adding an identical inductor in an identical gap in the output resonator. The inductance of the first inductor is a variable name (“Ind”) so we can just use that same variable name for the output resonator tuning inductor. Then, both resonators are

conductor (given an input voltage in the input subsection) is the current distribution in the conductor. That is our solution. Next, we take the now known current on the input subsection and, given the voltage we assumed on that subsection, we can calculate the input impedance. Repeat this process for each frequency of interest, and we have a plot of the input impedance as a function of frequency. We now have all this information without ever having built anything.

This implementation of the Method of Moments is restricted to planar circuits only. Arbitrary 3D structures, including those illustrated in the previous section, are not appropriate for this approach. The reason planar method of moments has seen wide-spread use is that it can handle much larger planar circuits at high levels of accuracy. Appropriate planar circuits can include a large number of layers. PCBs in a computer typically use 4 to 8 layers but can have as many as 100 layers. This is no problem for the method of moments. Sometimes planar problems are referred to as having 2.5 dimensions, rather than full 3D. However, full 3D fields and full 3D current are calculated (conducting connections between layers in a circuit board are called vias and they allow current to flow vertically), so describing this kind of problem as 3D planar automatically equal.

Note that for filters, the shielding box dimensions can be critical. You should analyze the filter with the box size that you are going to use to build it. It is important to keep the box size as small as possible so that box resonances don't insert a "suck out," for example, right in the middle of your passband! You might be able to get a good-looking response by using the filter without a shielding box, but now, all those antenna-like resonators in the filter start talking with other circuits in the vicinity causing all sorts of parasitic oscillations and intermodulation distortions. On top of that, the filter rejection above the passband can degrade.

Professionals often use an EM analysis, like *Sonnet*, in conjunction with a so-called *framework*. A microwave design framework includes many different inter-connected microwave-related software tools including, for example, circuit theory analysis. For port-tuning, a designer using both the Keysight Technologies ADS framework and *Sonnet* can use any ADS component as a tuning element in Sonnet. In this case, using a transmission line model (instead of a small inductor) makes life especially easy as the designer can simply read off the required resonator change of length right from the optimized transmission line element. Figuring out how much inductance corresponds to how much resonator length is not needed. This can even be done in *Sonnet* by using a "Project" component for tuning and making a *Sonnet* net list project with the tuning transmission line in the project.

Finally, designers also often need to tune the length of every resonator, not just one or two of them. In addition, they also need to tune the coupling between resonators (i.e., adjust the gap between resonators). **Figure 6.H** shows an example of a filter tuned in this way. All of this can be done with port tuning; however, it exceeds the capability of *SonnetLite*. For those with access to a full copy of *Sonnet* and a full microwave framework (Keysight Technologies *ADS*, Cadence *Microwave Office*, or Cadence *Virtuoso*), detailed tutorials can be found at **www.sonnetsoftware.com**.

is more accurate. Fields and currents are 3D, while the circuit is formed on layered (i.e., planar) dielectric. That said, enhancing Method of Moments to include 3D arbitrary volume meshing problems has seen substantial progress recently. See, for example, the Method of Moments book described in the **sidebar**.

RF and microwave designers usually work with something called *scattering* or *S-parameters*. For example, we could take the input impedance calculated above and convert it to SWR (Standing Wave Ratio). In professional work, SWR is commonly called VSWR (Voltage Standing Wave Ratio) and often pronounced, "Viz-Wahr." While the name is slightly different, it means the same thing. We can also mathematically convert either the VSWR or input impedance into the reflection coefficient. This gives us the voltage amplitude of the reflected wave given a 1-volt incident wave. When we include the phase of the reflected wave in the calculation, it is now what we call an S-parameter.

If we are dealing with a 2-port filter, we also need to know how much of that the 1-volt wave incident on the input makes it to the output. This is called a *transmission coefficient*. Now, let's say, our antenna is not perfectly matched and reflects RF back to the output of our filter. In response, the filter output now generates a reflected wave that goes right back to the antenna. This is called an *output reflection coefficient*. Finally, a portion of the wave that was reflected from the antenna and hits the output of our filter, finds its way through the filter, and comes out of the filter's input. This is called the *reverse transmission coefficient*. So, we have four numbers now to describe our 2-port filter. These are all scattering parameters and for the benefit of our computer, we put them all in a 2×2 matrix. Even more detailed descriptions can be found elsewhere. (See the RF Techniques chapter for more information about S-parameters.)

The Method of Moments has been developed for a wide variety of problems. For example, there are a few ways to implement the Method of Moments that make it amazingly efficient for doing extremely large problems like a radar signal illuminating an aircraft, or even an entire aircraft carrier. However, this type of implementation tends not to be as useful for, say, circuits on PCBs or other planar circuits where there is a reduction in accuracy.

There are two widely used approaches that are preferred for MoM analysis of planar circuits. Both approaches can handle large planar circuits, even when they have multiple layers. One approach performs the required current-induces-voltage calculation assuming the planar circuit is unshielded. The other assumes the circuit is shielded. Both approaches are useful. The unshielded approach is a natural for unshielded circuits, like antennas. The shielded approach is a natural for shielded circuits, like a filter inside a shielding box. However, each approach can flex into the other's domain. The shielded approach benefits from a more accurate induced voltage calculation but it requires an underlying, fine pixel-like grid for meshing. The unshielded approach benefits from allowing arbitrary subsection size, which can be handy, provided the required less accurate induced-voltage calculation is not of concern.

Figure 6.42 shows a typical circuit ana-

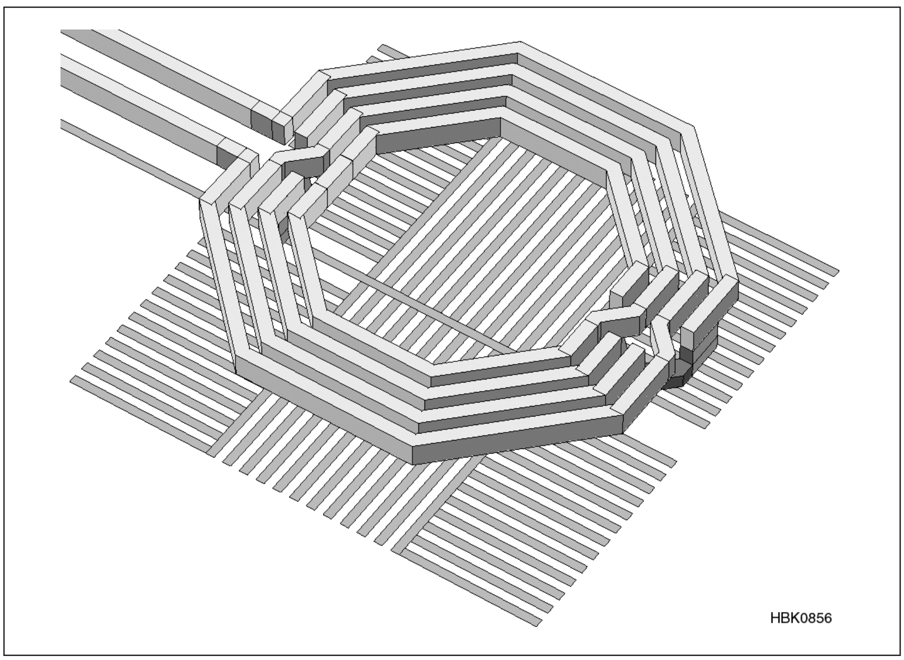


Figure 6.42 — Method of Moments (MoM) can be used to analyze planar 3D circuits like this spiral inductor over silicon. Many such inductors are present in the RF chip set for every cell phone. Silicon dioxide insulator (not shown) gives some separation between the bottom of the inductor and the top of the semi-conducting silicon. To reduce inductor losses, a grid-like ground shield is placed directly on the silicon to try to keep the inductor fields out of the silicon. (Courtesy Sonnet, with permission.)

lyzed using shielded 3D planar MoM. This is a spiral inductor on silicon, a critical component used in all cell phone RF chip sets. For example, when you make a call, the cell phone computer moves the transmitter and receiver to the channel assigned by the cell site for your call. To do this, it programs a VCO (voltage-controlled oscillator) that is inside a PLL (phase locked loop) to move to the required frequency. The VCO has an inductor as part of its tuned circuit. Compared to silicon chips, inductors are gigantic. They take up a lot of space and space (or "real estate" as the RFIC designers call it) is expensive. The solution is to make the spiral turns even smaller and use more of them. However, that increases the loss. If there is too much loss, the VCO noise goes up and tuning range goes down. In the worst case, loss might be so high that the VCO does not even start oscillating.

A typical silicon wafer fabrication ("wafer fab" for short) can easily cost $1 million. Fabricators do not have the time or the money to repeatedly build and tweak a circuit. It is critical that the VCO, and everything else on that wafer, work the first time. This is where EM analysis saves the day. The designer does all the trial and error on the computer, and then builds the circuit once.

Silicon is a lossy conductor, i.e., a semiconductor. Any RF designer knows that when we put an inductor close to a lossy conductor, the inductor loss increases. To reduce the loss, we try to get the inductor as far as possible from the silicon. A given process might have a few microns of silicon dioxide, or other insulators, deposited on top of the silicon. So, the designer puts the inductor on top of all available silicon dioxide. In **Figure 6.42**, these insulating layers are not shown, so it looks like the inductor is floating in air.

Another trick to reduce loss is to add a ground shield. This has been done for the inductor in Figure 6.42. This is the grid-like conductor pattern under the inductor. It is important that the ground shield conductors are, more or less, at right angles to the spiral inductor turns above them. If the ground shield conductors are parallel, they start to act like a shorted turn in the secondary winding of a transformer — not good!

Figure 6.43 shows the current in the surface of the silicon directly under the spiral inductor when there is no ground shield. The figure's caption explains current levels. At first, it is a surprise that the direction of the current (not shown) is radial, flowing toward or away from the inductor. This means the current is driven by the electric field around the inductor, not by the magnetic field. The designer wants this current to be as low as possible to reduce loss.

Figure 6.44 shows the substrate current in the silicon under a small portion of the inductor with a ground shield. This was analyzed using *Sonnet*, a shielded 3D planar EM tool. As mentioned above, the induced voltage calculation is very accurate for shielded tools, but a fine, underlying rectangular mesh is required. If you look closely on this zoomed image, you can see the *pixelation* caused by the fine mesh.

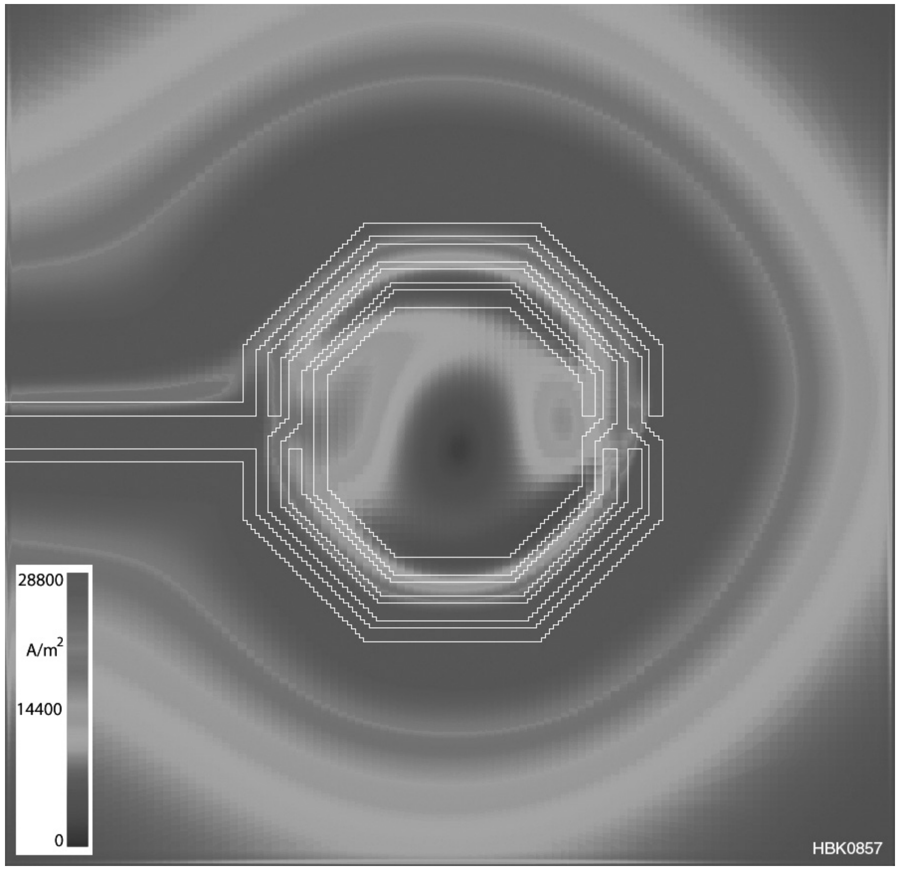


Figure 6.43 — The spiral inductor, here without the ground shield, induces current in the silicon. This current increases loss and must be minimized. Brighter shades indicate mid-level current (see the scale at the lower left). Highest current is in the dark area directly under the outer turns of the inductor. (Courtesy Sonnet, with permission.)

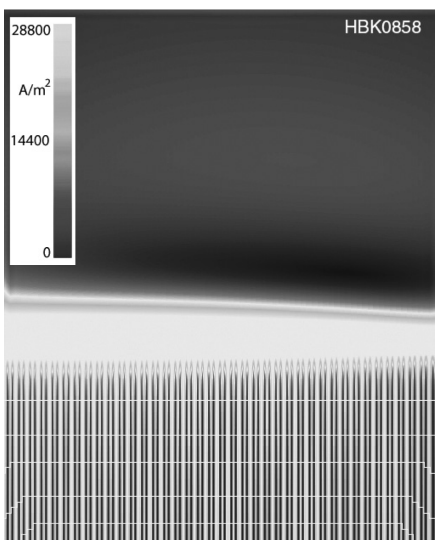


Figure 6.44 — A small section of the current in the silicon under the inductor with ground shield in place shows a strongly modified current in the silicon. Whether or not inductor loss is actually decreased, or increased, strongly depends on the ground shield design. (Courtesy Sonnet, with permission.)

Normally, we view only the current in the conductors of our circuit with planar EM tools. A description of how to view tangential (i.e., parallel to the surface of the substrate) electric field using something called "sense metal" in planar method of moments is given the Sonnet User's manual (**www.sonnetsoftware.com**), search for "sense metal." By Ohm's law, the tangential electric field on the surface of the silicon substrate is proportional to the current flowing in the silicon. If we make the sense metal surface resistance, say, 1 million ohms/square, then it will have little effect on the overall fields. The (very tiny) currents in the sense metal (due to the current flowing in the lossy silicon) are proportional to the tangential electric field required by Ohm's law.

6.4.3 Commercial Electromagnetic Analysis Tools

Below we describe most of the current RF and microwave EM software vendors that concentrate on the microwave market. These vendors have all exhibited at IMS, the International Microwave Symposium, and are viewable on the 2021 conference's Expocad Web software. IMS is the premier microwave conference in the world. Our apologies to any vendors that we missed. All product names are trademarks of the associated vendors.

Ansoft (a division of Ansys): Widely used finite element code specialized for microwave work, *HFSS* (High Frequency Structure Simulator). Ansoft provides a variety of other EM tools, including planar unshielded method of moments. All tools are integrated into a comprehensive microwave design framework with a wide variety of circuit theory and systems simulation tools. **www.ansys.com**

AWR (a division of Cadence): Their tool, *Axiem*, uses unshielded planar method of moments integrated into a comprehensive widely used microwave design framework (Microwave Office) incorporating a wide variety of circuit theory and systems simulation tools. **www.awrcorp.com**

CST (a division of Dassault Systèmes): A wide range of nearly every kind of EM solver available today. Their primary tool is based on a technique closely related to FDTD providing time-domain full 3D volume meshing EM analysis. Their wide range of tools are organized into various suites. For example, CST *Microwave Studio* includes transient, frequency domain, eigenmode, resonant, integral equation, asymptotic, and TLM solvers. **www.cst.com**

EM Software and Systems (a division of Altair): Their primary product is *FEKO*, based on a fast implementation of method of moments intended for very large problems. It is hybridized with several other EM techniques including finite elements, physical optics, geometric optics, and uniform theory of diffraction. Suitable for EMC, cable harnesses, antenna placement, bio-medical, microwave circuits, etc. **https://www.altair.com/feko-applications/**

IMST: Their main software product, *Empire XPU*, provides time domain 3D volume meshing FDTD analysis of microwave structures. They also offer a tool, *MultiLib*, that provides models for planar multi-layer microwave circuits that can be used directly in the Keysight Technologies microwave design framework ADS. **www.imst.com**

Integrand Software (A division of Cadence): Used widely for Si RFIC design, EMX is a 3D unshielded method of moments analysis that is tailored specifically for RF design work on silicon, including a high degree of automation. It is fully integrated into *Cadence Virtuoso*, a widely used Si RFIC design framework. **www.integrandsoftware.com**

Keysight Technologies: Two main EM tools, unshielded 3D planar method of moments EM analysis *Momentum*, and 3D volume meshing finite elements *EMPro*. Formerly Agilent Technologies Electronic Measurement Group, they offer the largest and most widely used range of integrated microwave tools including their microwave design framework, *ADS, Advanced Design System*. **https://www.keysight.com/us/en/products/software/pathwave-design-software.htm**

Mician: Mode matching tool *µWave Wizard*, uses the fact that every finite volume structure has a set of characteristic modes, sometimes also called resonances. Any structure that can be decomposed into a set of simpler building blocks that have known modes solved for by any of the many different EM tools included can be analyzed by adding all modes together and weighted so that all fields match at the boundaries between the blocks. Very fast and accurate analysis of waveguide filters, for example, is possible. A free version is available. **www.mician.com**

Microwave Innovation Group: Mode matching tool *WASP-NET*, uses the fact that every finite volume structure has a set of characteristic modes, sometimes also called resonances. Any structure that can be decomposed into a set of simpler building blocks that have known modes solved for by any of the many different EM tools included can be analyzed by adding all modes together and weighted so that all fields match at the boundaries between the blocks. Very fast and accurate analysis of waveguide filters, for example, is possible. **www.wasp-net.com**

Remcom: Provider of *XFdtd*, a full wave, 3D time domain, EM solver for antenna design and placement, mobile device design, biomedical, microwave, automotive radar, etc. Other products include *Wireless InSite,* site-specific radio propagation software for analysis of wireless communication systems, and *XGtd*, a ray-based EM analysis tool for antenna placement on large structures such as ships and aircraft. **www.remcom.com**

Sonnet Software: Main tool, *Sonnet Suites*, integrates a shielded, 3D planar, Method of Moments, EM analysis with plotting, modeling, etc., tools. Sonnet also integrates well with Keysight Technologies *ADS*, AWR *Microwave Office*, and *Cadence Virtuoso* microwave design frameworks. A free version is available. **www.sonnetsoftware.com**

WIPL-D: A range of products all based on unshielded Method of Moments. Emphasis is on fast solution of very large problems. Primary applications include large antennas, including dish antennas, antenna placement, EM compatibility analysis (EMC), and microwave circuits. A free version is available.