Exploring SimSmith

SimSmith
&
Time Domain Reflectometry
(Using Impedance Measurements)

Ward Harriman

ae6ty

SimSmith Is A Platform

Yes, it does Smith, SWR, and Power charts.

But

It also does Waveforms

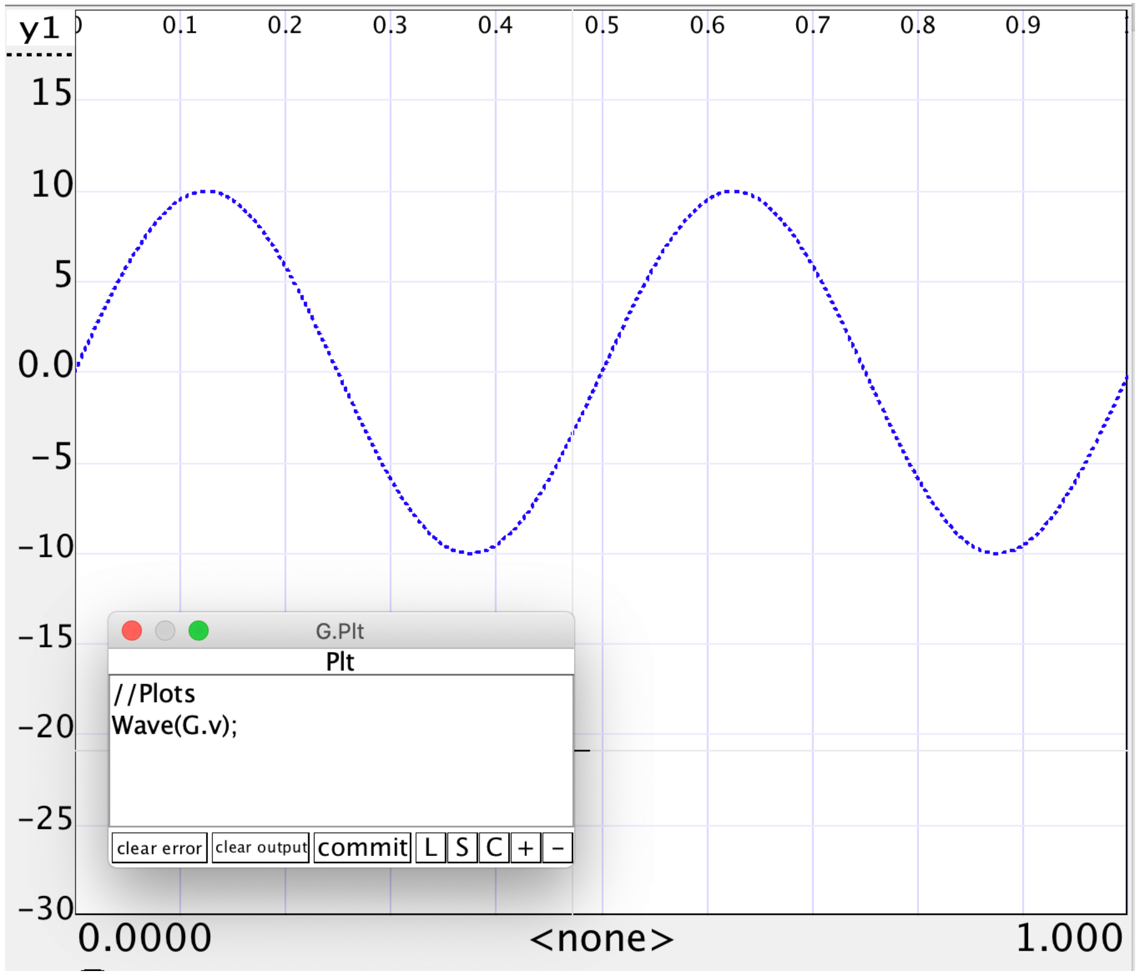
and

Can be used to develop other algorithms

Waveforms

SimSmith can display basic waveforms. Here is the simplest example.

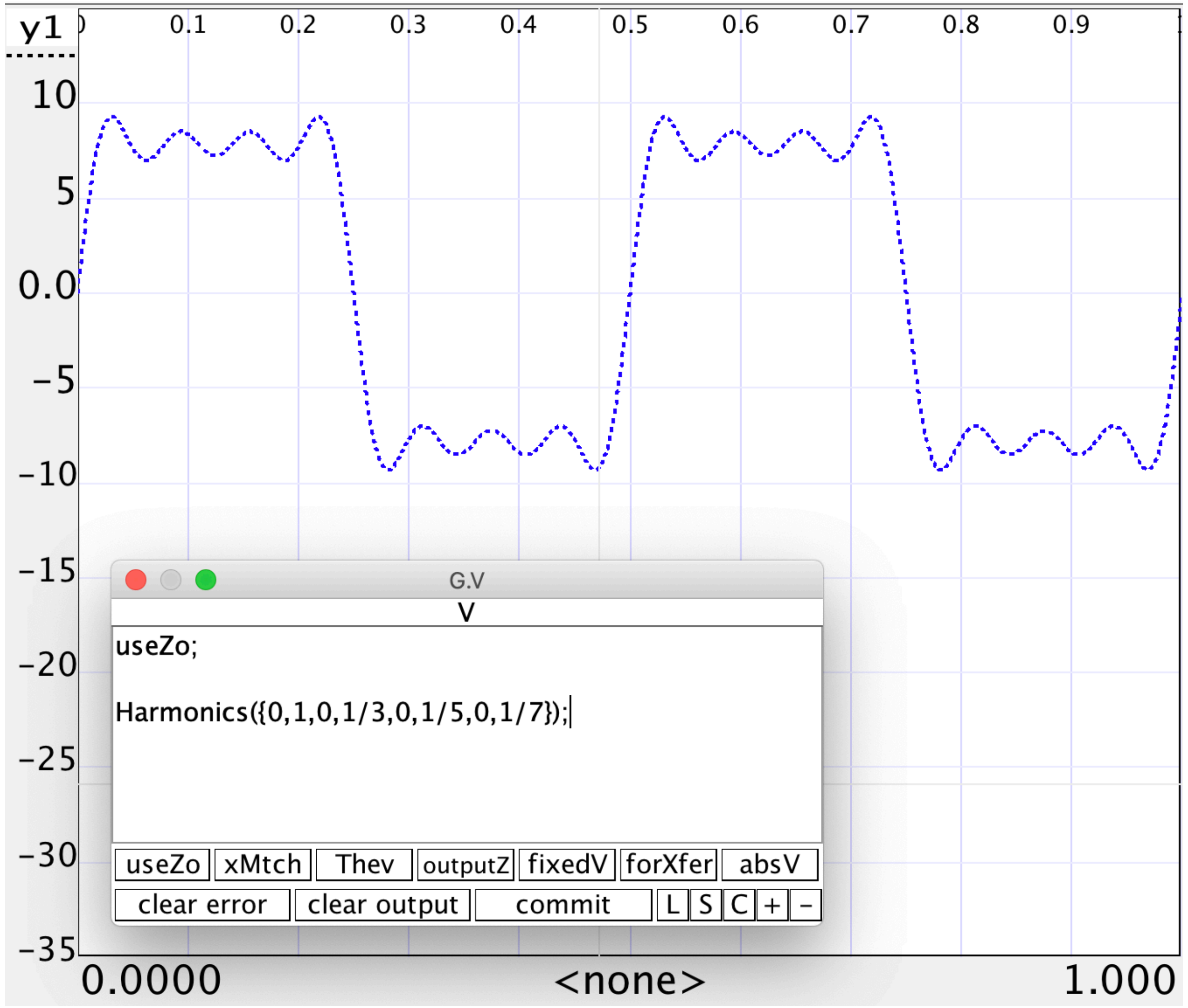
SimSmith always displays two cycles of the waveform (unless you've zoomed in).



Waveform Harmonics

There are times when the desired excitation is not a simple sine wave. For example, an amplifier might put out a square wave.

SimSmith can describe the waveform using the ‘Harmonics’ function.

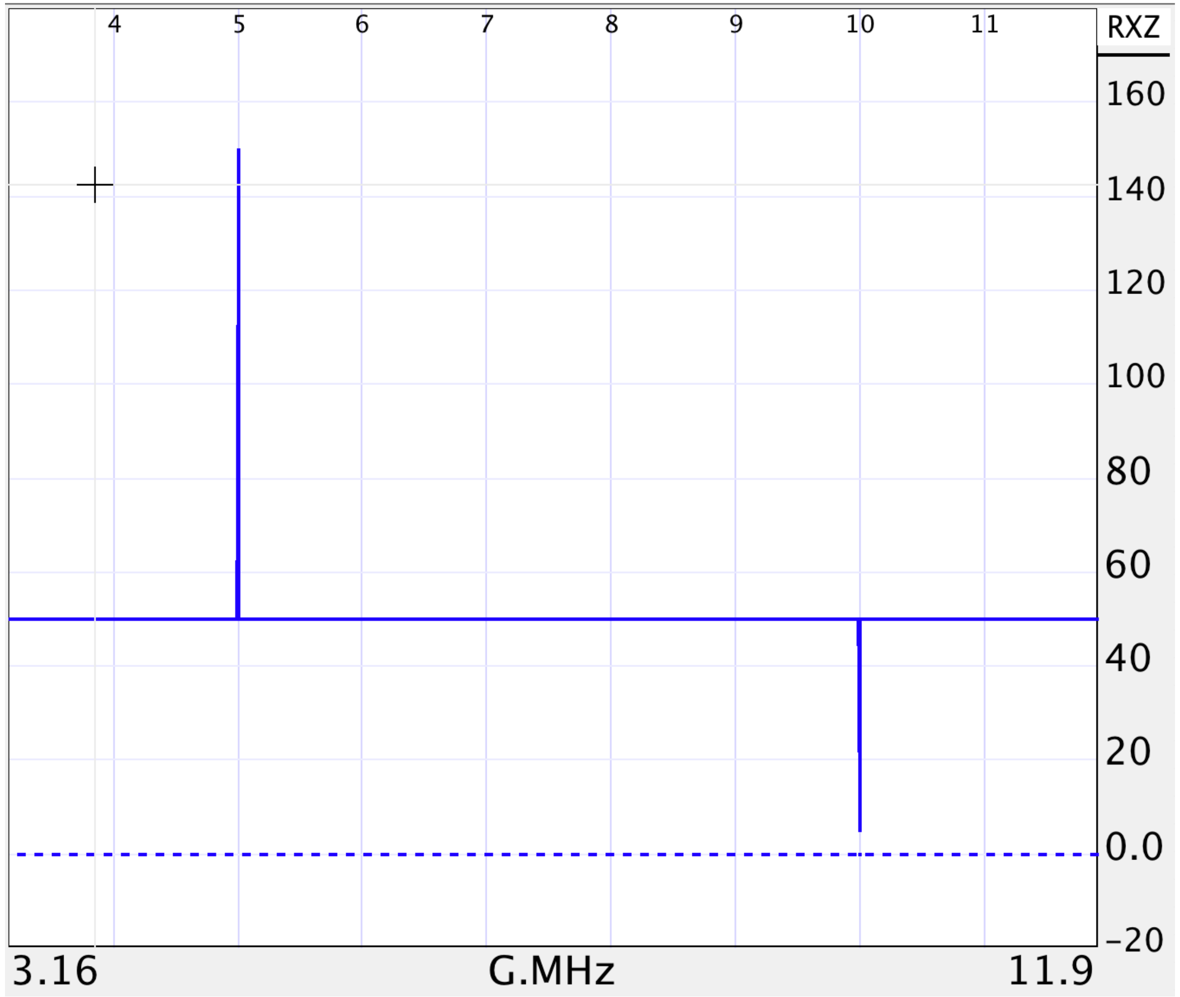


Leveraging

So… how can we leverage this?

How does time domain reflectometry work?

- Send a very short pulse down the line
- Observe when a reflection occurs
- Interpret the reflection

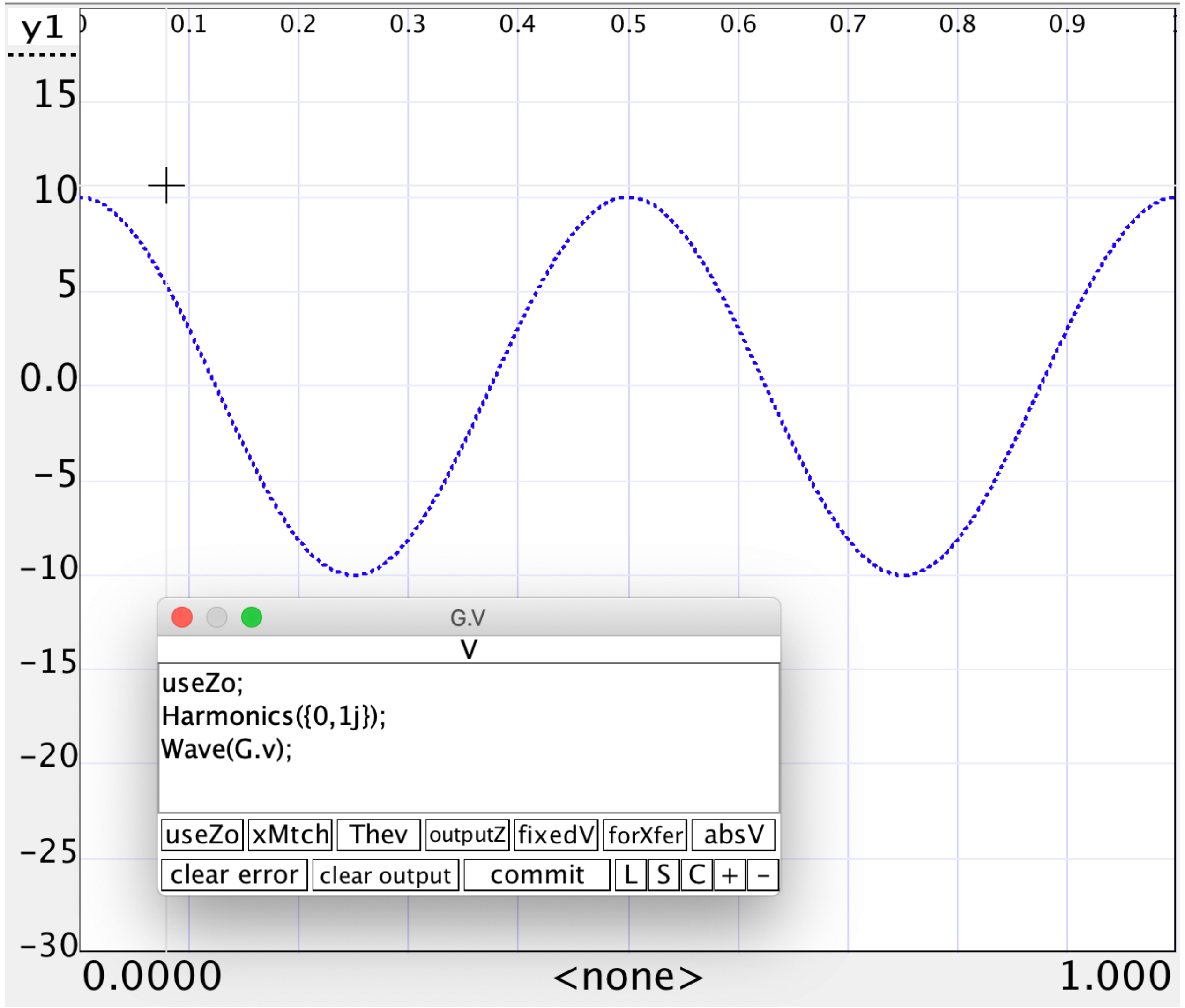


BUT… my VNA doesn't do pulses

Then, again, neither does SimSmith!

But, we can use the Harmonics() command to synthesize a pulse.

Notice '1j' moves peak to be at center.

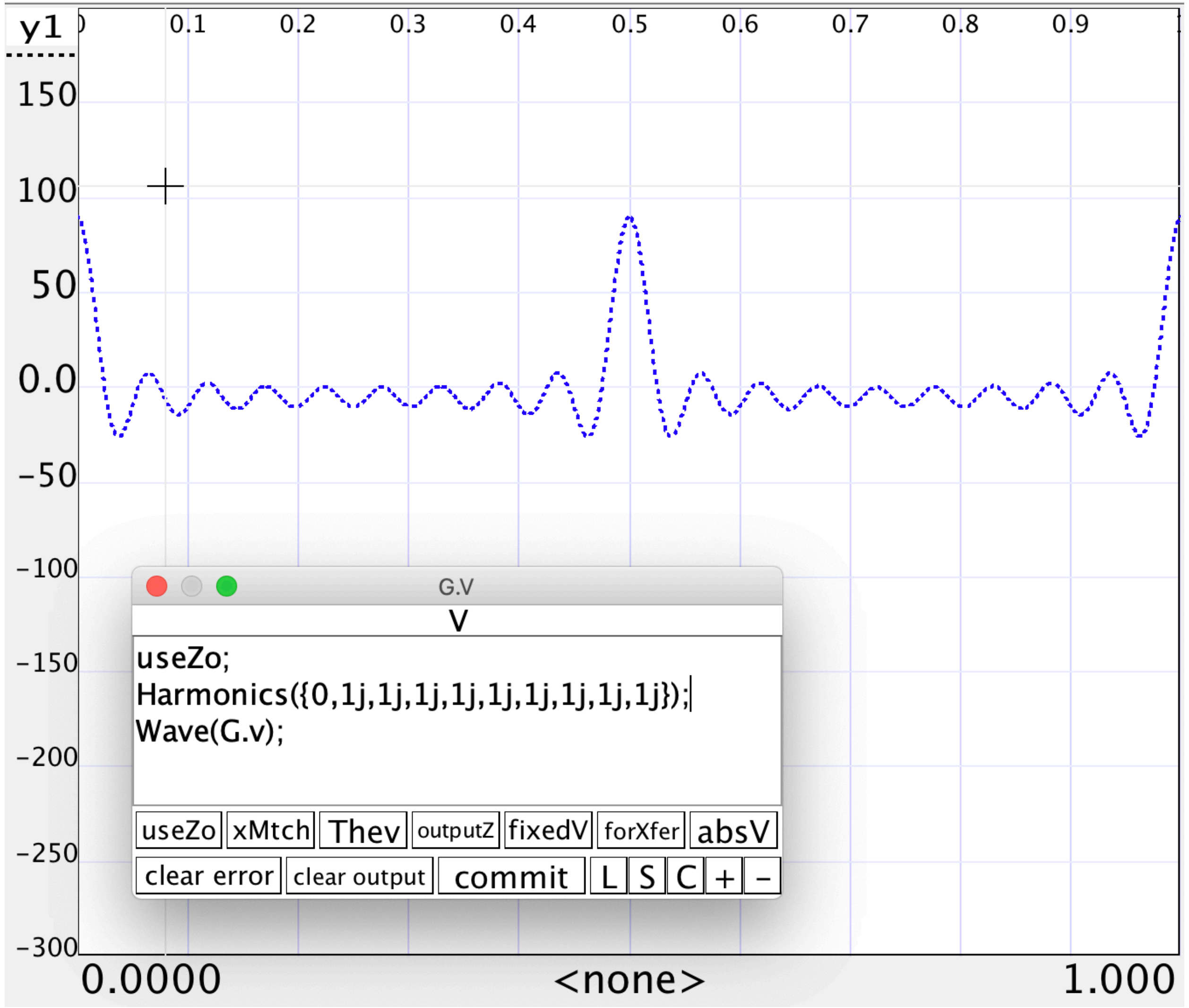


Now let's make a pulse:

Simply add higher harmonics…

We're getting there….

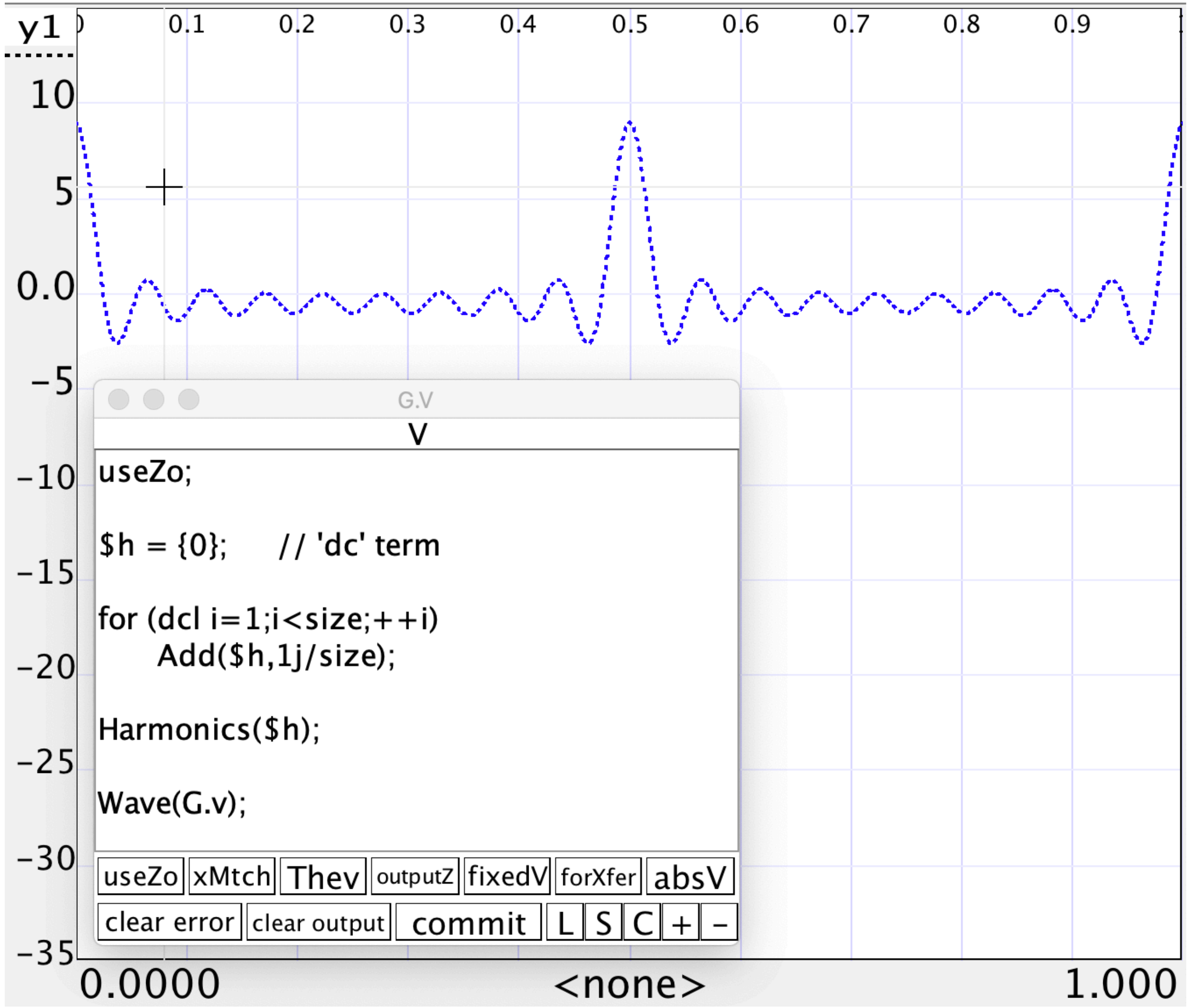
It gets tedious adding terms…



So we'll write a little program:

Use a parameter and make a list of the desired length…

Note that the harmonics are scaled by 'size' so the pulse doesn't grow taller as we add terms.



Many terms

Here are 256 terms.

There is some ringing around the pulse; a common effect.

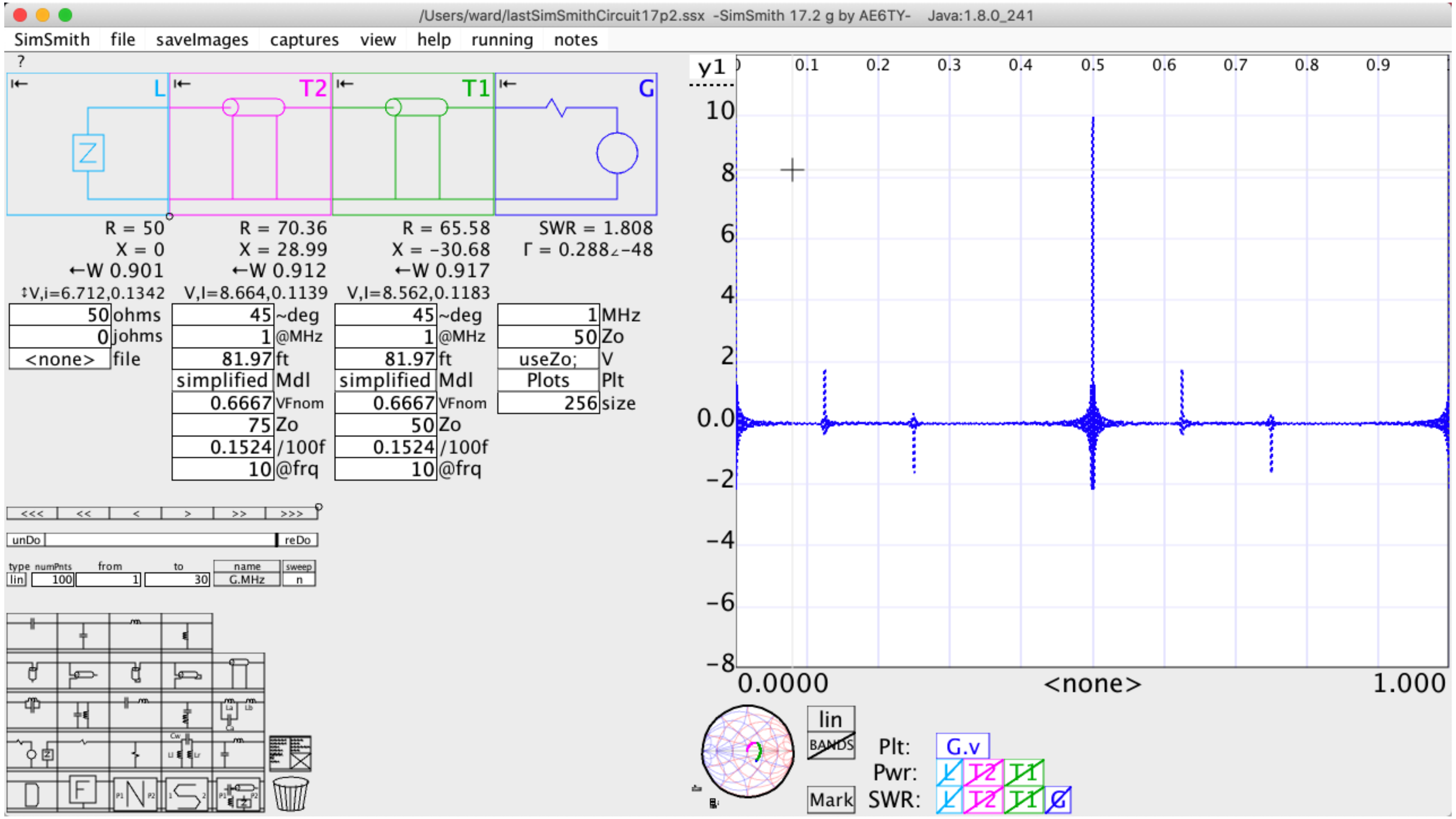
Now, lets use it

We won't use an impedance file just yet…

Let's use a simple circuit to verify functionality.

(positive pulses…)

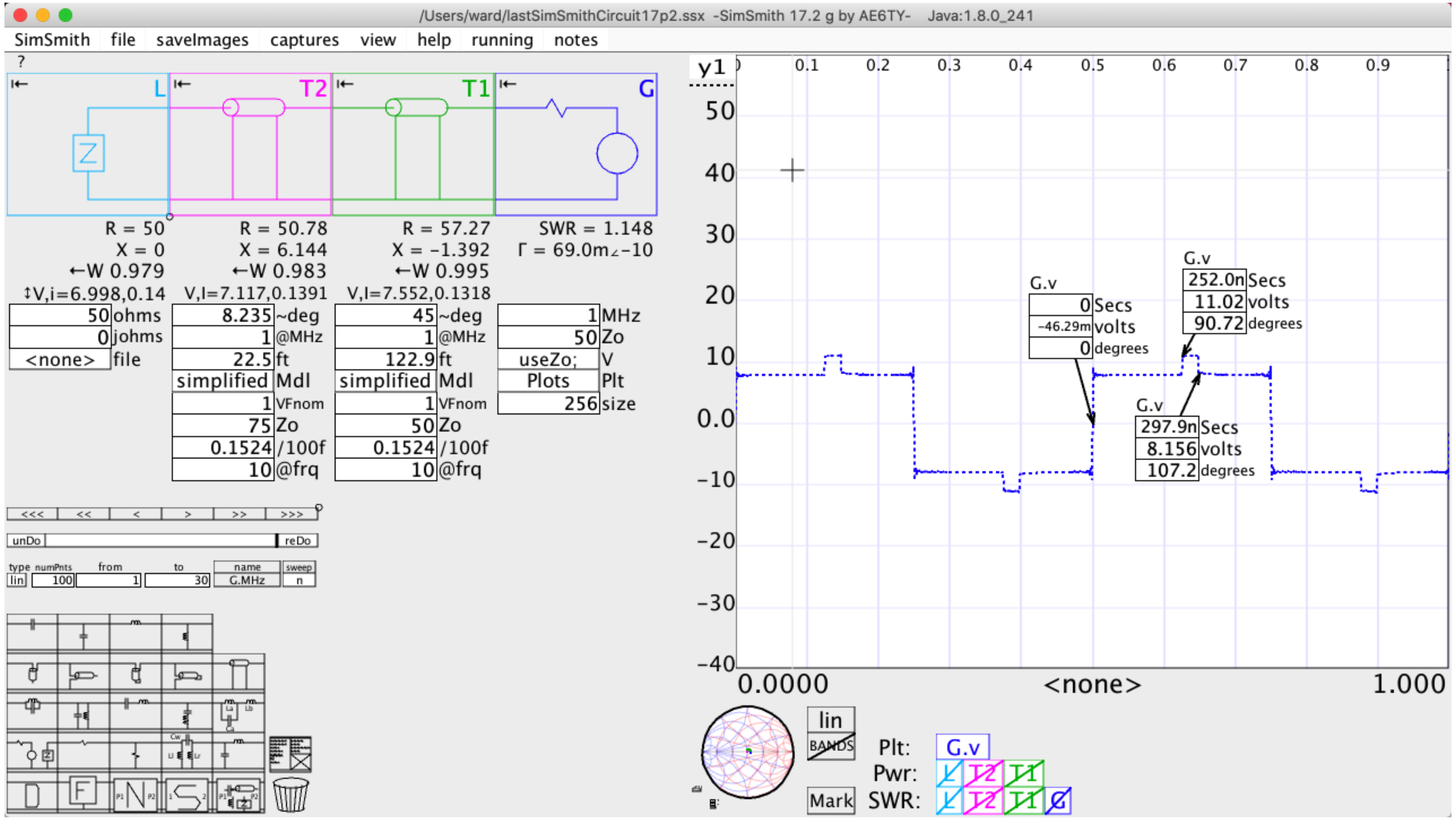
(negative pulses…)



Different Excitation

We can use a square wave to see different details.

Here I changed the length of the second line. The markers show it is '45ns' long. At 1000ft/second… that makes the line about 22.5 feet long.

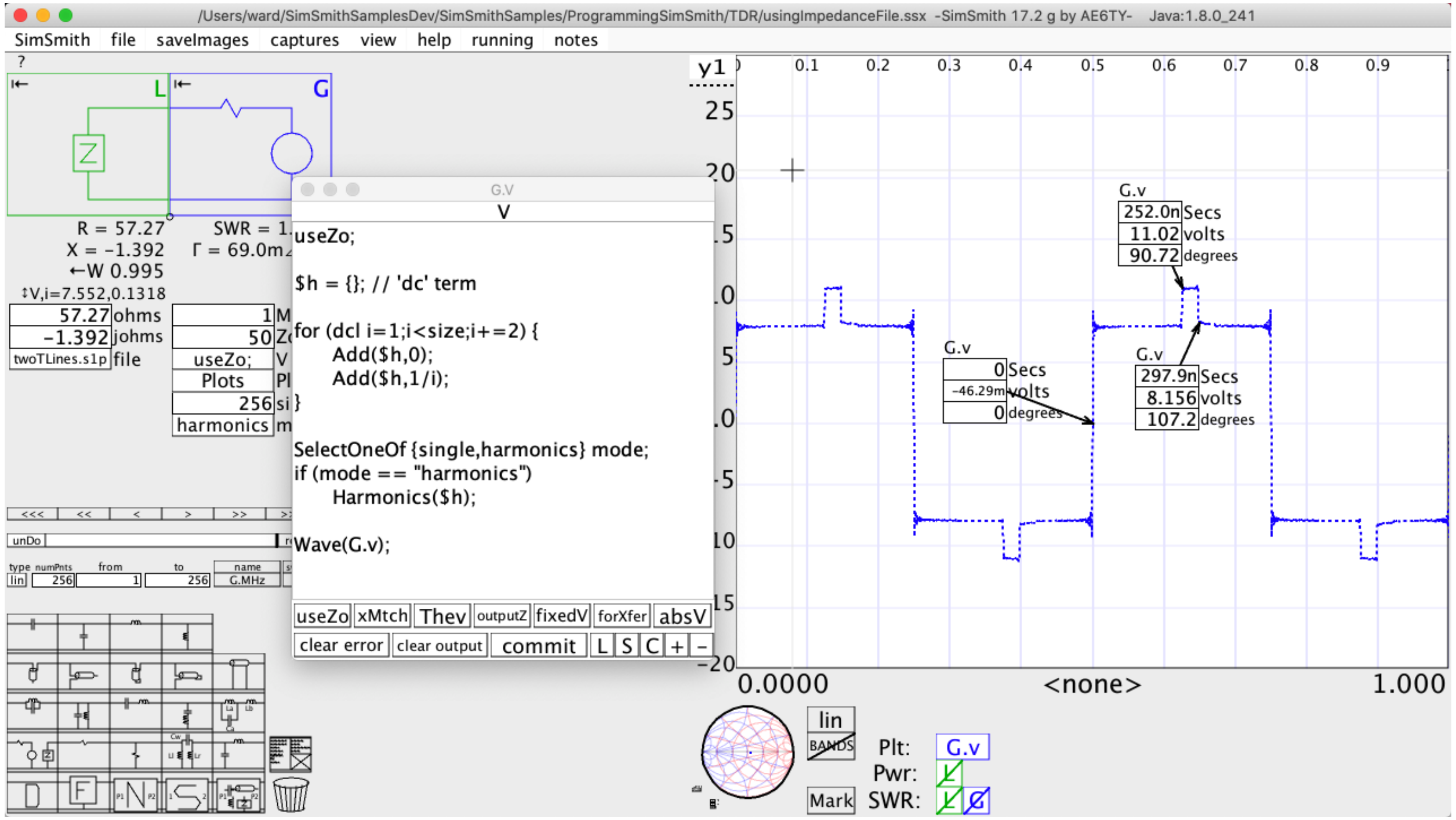


Using Impedance file

I wrote out an ‘s1p’ file that recorded the impedance seen by the generator…

This is just the kind of thing your VNA would measure.

The answer is the same.



Other ways.

SimSmith provides a very general-purpose programming language.

Of key important here:

- Reading Impedance Files
- Computing the ‘Inverse Fourier Transform’

Why the latter?

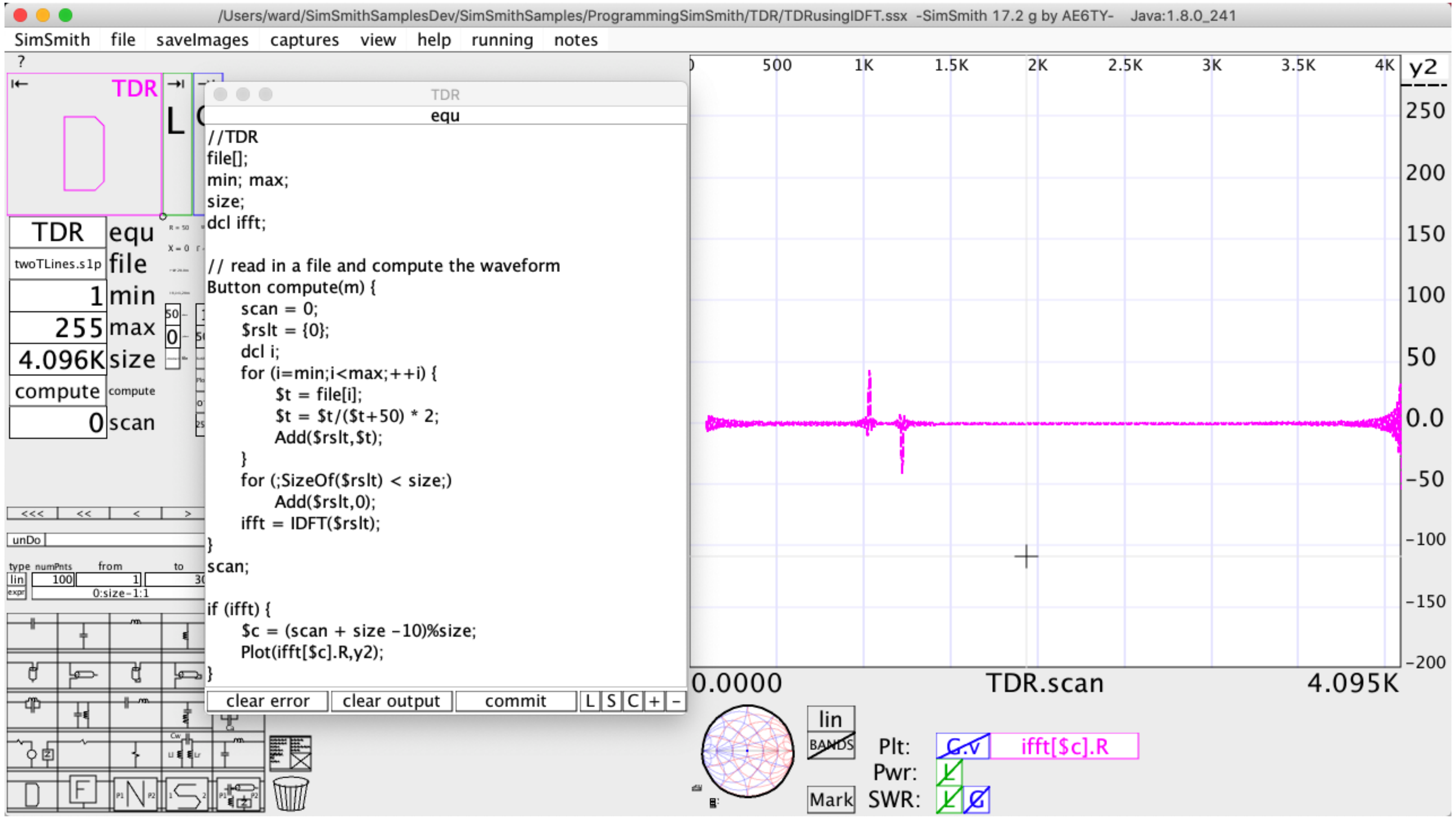
- We need to translate of bunch of frequency indexed measurements into a time based measurement….
- That’s exactly what the ‘Inverse Fourier Transform’ does.

Using SimSmith's IDFT

Basically, we need to read in the impedance file which is just a bunch of frequency measurements.

Compute the inverse Fourier Transform

Draw the results.



The pieces:

Start with an empty list of ‘impedance/frequemcy measurements’ in ‘file’.

Scan through them converting each impedance into a voltage.

Pad things to the size length you’d like the waveform to have.

Compute the IDFT

Display the results.

Generating the Square Wave Equivalent

Is left as an exercise for the reader.

Hint: the values from the impedance file need to be scaled using the DFT of a square wave.

The DFT of a square wave is:

$$\sum_{1}^{n} 1/(2n-1)$$

(Which we already saw….)

Wrap Up

Can use Wave and Harmonics to implement TDR

Can use VNA ‘impedance’ files as fodder for TDR

Can use Fourier and Inverse Fourier transforms to do the same

With a little programming, SimSmith can be used to explore a variety of algorithms well beyond basic Smith, SWR, and Power transfer problems.