# Vector Fitting in Tidy3D

Alec Lau, Charles Wojcik

**Abstract**

In Finite-Difference Time-Domain (FDTD) simulations, the refractive index of dispersive materials needs to be fitted first because the standard FDTD algorithm assumes non-dispersive materials, where the refractive index is constant across all frequencies. However, many materials exhibit dispersion, meaning their refractive index varies with the frequency of the electromagnetic wave. To accurately model dispersive materials in FDTD simulations, the frequency-dependent refractive index or permittivity must be incorporated into the simulation. This is typically done by fitting the refractive index data to an analytical model, such as the Lorentz, Debye, Drude, or pole-residue models. Once the analytical model for the refractive index has been obtained through fitting, it can be incorporated into the FDTD algorithm using auxiliary differential equations or recursive convolution techniques. This allows the simulation to accurately capture the frequency-dependent behavior of the dispersive material and produce more accurate results. This article introduces the fast vector fitting procedure in Tidy3D. The Tidy3D material fitting plugin was written by Casey Wojcik, Ph.D.

## Motivation

A material's refractive index often depends on frequency. The relationship between frequency and permittivity $f(\omega)$ is the **dispersion**, which can be approximated by

functions like the following:

$$f(\omega) \approx \sum_{n=1}^{N} \frac{c_n}{\omega - a_n} + d$$

for some $c_n, a_n, d$, and $h$, where $d, h$ are real and $c_n, a_n$ are either real or have conjugate pairs.

These functions are called **pole-residue models**. Popular dispersion models in this form are Lorentz models and Debye models. It turns out that both of these models are special cases of the following model, given by [3]:

$$\epsilon(\omega) = \epsilon_\infty - \sum_{i=1}^{N} [\frac{c_i}{j\omega + a_i} + \frac{c_i^*}{j\omega + a_i^*}]$$

(which fits in our $f(\omega)$ form, where $c_n, a_n$ are conjugate pairs). In the time domain, this yields an exponentially decaying function that can easily be implemented in the FDTD timestep (the interested reader is referred to [5].) If we have a fit of real material dispersions, we can accurately model the responses of a large number of frequencies in a single FDTD simulation.

# Goal

We want to approximate the frequency response function $f(\omega)$ as a pole-residue model:

$$\sum_{n=1}^{N} \frac{c_n}{\omega - a_n} + d + \omega h$$

where the $c_n$s and $a_n$s are either real or conjugate pairs.

Methods for approximating dispersion models include finding a quotient of suitable polynomials, in e.g. the Levy method, the Sanathanan-Koerner method, and the Noda

method. The problem with these methods is that the problem scales poorly with frequency, and is thus a poor choice for a general application. Another method of fitting (Bode) approximates the magnitude of frequency response, but the error can be significant. A further method (Levenberg-Marquardt) uses gradient descent and the Gauss-Newton method to iteratively improve parameters. The problem with this method, however, is that it can get stuck in a local minimum.

Vector fitting is both general and robust to the above shortcomings.

# Method

Here we will go over the main ideas in vector fitting. We will skip over bookkeeping details regarding special cases, but the interested reader is referred to the cited papers.

## Function Approximation

The way to do this is to choose a function approximating a pole-residue function

$$\sigma(s) \approx \sum_{n=1}^{N} \frac{\tilde{c_n}}{s - \overline{a_n}} + \tilde{d}$$

and multiply this with $f(s)$. For this function, we'll choose the poles $\overline{a_n}$ and solve for the residues $\tilde{c_n}$ associated to those pole. Then $\sigma(s)f(s)$ approximates the desired poles-residue form

$$\sigma(s)f(s) \approx \sum_{n=1}^{N} \frac{c_n}{s - \overline{a_n}} + d + sh := (\sigma f)(s)$$

We can find this approximation by recasting this problem as a least-squares prob-

lem:

$$(\sum_{n=1}^{N} \frac{c_n}{s-\overline{a}_n} + d + sh) \approx (\sum_{n=1}^{N} \frac{\tilde{c}_n}{s-\overline{a}_n} + \tilde{d}) f(s)$$

$$\Rightarrow Ax = b$$

where

$$A[k,:] = [\frac{1}{s_k - \overline{a}_1}, ..., \frac{1}{s_k - \overline{a}_N}, \frac{-f(s_k)}{s_k - \overline{a}_1}, ..., \frac{-f(s_k)}{s_k - \overline{a}_N}, 1, s_k],$$

$b_k = f(s_k)\tilde{d}$, and

$$x = \begin{pmatrix} c_1 \\ \vdots \\ c_N \\ \tilde{c}_1 \\ \vdots \\ \tilde{c}_N \\ d \\ h \end{pmatrix}$$

is the solution vector. This $Ax = b$ problem can then be solved by Singular Value Decomposition.

Note that here we're solving for the residues associated to the poles we've chosen for $\sigma(s)$.

## Recap

We find an approximation function $\sigma(s)$ in pole-residue form by solving

$$(\sigma f)(s) \approx f(s)\sigma(s),$$

where both $(\sigma f)(s)$ and $\sigma(s)$ are therefore in pole-residue form, as $Ax = b$. From here we can get the residues associated to the poles we've chosen for $\sigma(s)$.

## Solving for the poles of $f(s)$

Once we know both $(\sigma f)(s)$ and $\sigma(s)$, we can solve for an approximation of $f(s)$ by rewriting $(\sigma f)(s)$ and $\sigma(s)$ as $h\prod_{n=1}^{N+1}\frac{s-z_n}{s-\overline{a_n}}$ and $\prod_{n=1}^{N}\frac{s-\tilde{z_n}}{s-\overline{a_n}}$ for some $z_n, \tilde{z_n}$, respectively, so

$$f(s) \approx \frac{(\sigma f)(s)}{\sigma(s)} \approx h\frac{\prod_{n=1}^{N+1} s - z_n}{\prod_{n=1}^{N} s - \tilde{z_n}}$$

This shows us that the zeros $\tilde{z_n}$ of $\sigma(s)$ are (approximately) the poles of $f(s)$. These zeros are solved as the eigenvalues of the matrix

$$\begin{pmatrix} \overline{a_1} & & \\ & \ddots & \\ & & \overline{a_N} \end{pmatrix} - \frac{1}{\tilde{d}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} \tilde{c_1} & \dots & \tilde{c_n} \end{pmatrix}$$

If the $\tilde{d}$ we found is too small to work efficiently, we re-solve with a set minimum $\tilde{d}$ value.

## Iteration

Keep in mind that, so far, we've only calculated the zeroes and residues of $\sigma(s)$ associated to the poles we've chosen for $\sigma(s)$, so our approximation for the poles we've found for $f(s)$ is only as good as the poles we've chosen for $\sigma(s)$ initially. To further refine our approximation, we repeat the process starting with a new choice of

$$\sigma(s) \approx \sum_{n=1}^{N} \frac{\tilde{c_n}'}{s - \tilde{z_n}} + \tilde{d}$$

where our choice of poles $\tilde{z_n}$ are the zeros of the previous $\sigma(s)$ (poles for $f(s)$). By continuing this process, by the least-squares method, we can converge on poles suitable for fitting $f(s)$. According to [2], convergence usually takes 2-3 iterations.

### Note on Passivity

A vector fitted function can still lead to an unstable simulation. Absorbed power must be always be positive for simulation stability, so $Im(\epsilon(s))$ must be $\geq 0$. We can enforce this by linearizing the target function around our fitted $\epsilon(s)$. This creates a least-squares problem subject to intermediate eigenvalue constraints. This is then solved with quadratic programming. The interested reader is referred to [6] for details.

# Acknowledgements

The Tidy3D material fitting plugin was written by Charles Wojcik, Ph.D., Research Scientist at Flexcompute.

# References


[1] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," IEEE Trans. Power. Deliv. 14, 3 (1999).

[2] B. Gustavsen, "Improving the pole relocation properties of vector fitting," IEEE Trans. Power Deliv. 21, 3 (2006).

[3] M. Han, R. W. Dutton, and S. Fan, "Model dispersive media in finite-difference time-domain method with complex-conjugate pole-residue pairs," IEEE Microwave and Wireless Component Letters, vol. 16, pp. 119-121 (2006).

[4] E. S. Bañuelos-Cabral, J. A. Gutiérrez-Robles, and B. Gustavsen, "Rational Fitting Techniques for the Modeling of Electric Power Components and Systems Using MATLAB Environment," InTech, Dec. 06, 2017. doi: 10.5772/intechopen.71358.

[5] A. Taflove and S. C. Hagness, "Computational Electrodynamics: The Finite-Difference Time-Domain Method," 2nd ed. Norwood, MA: Artech House, 2000, ch. 9.

[6] B. Gustavsen and A. Semlyen, "Enforcing Passivity for Admittance Matrices Approximated by Rational Functions," IEEE Trans. Power. Deliv. 16, 1 (2001)