analytical chemistry

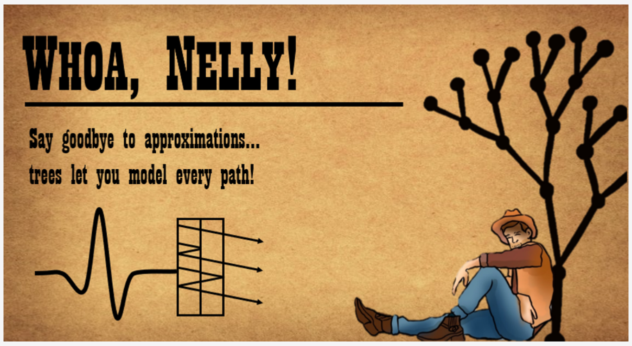
Nelly: A User-Friendly and Open-Source Implementation of Tree-Based Complex Refractive Index Analysis for Terahertz Spectroscopy

Uriel Tayvah,* Jacob A. Spies, Jens Neu,* and Charles A. Schmuttenmaer

Cite This: *Anal. Chem.* 2021, 93, 11243−11250

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information


WHOA, NELLY!
Say goodbye to approximations...
trees let you model every path!

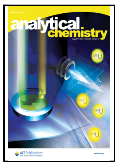
ABSTRACT: Terahertz (THz) spectroscopy is a powerful tool for unambiguously extracting complex-valued material properties (e.g., refractive index, conductivity, etc.) from a wide range of samples, with applications ranging from materials science to biology. However, extracting complex refractive indices from THz time-domain spectroscopy data can prove challenging, especially for multilayer samples. These challenges arise from the large number of transmission-reflection paths the THz pulse can take through the sample layers, leading to unwieldy strings of Fresnel coefficients. This issue has often been addressed using various approximations. However, these approximations are only applicable to specific classes of samples and can give erroneous results when misapplied. An alternative to this approach is to programmatically model all possible paths through the sample. The many paths through the sample layers can be modeled as a tree that branches at every point where the paths diverge, i.e., whenever the pulse can either be transmitted or reflected. This tree can then be used to generate expressions relating the unknown refractive index to the observed time domain data. Here, we provide a freely available open-source package implementing this method as both a MATLAB library and a corresponding graphical user interface, which can also be run without a MATLAB license (https://github.com/YaleTHz/nelly). We have tested this method for a range of samples and compared the results to commonly used approximations to demonstrate its accuracy and wide applicability. Our method consistently gives better agreement than common approximations.

■ INTRODUCTION

Many important physical processes leave spectroscopic signatures in the terahertz (THz) region of the electromagnetic spectrum (0.1−10 THz = 3−333 cm^{-1} = 0.1−10 ps). These include the nanoscale conductivity critical to emerging materials,[1] collective motions in molecules[2,3] and proteins,[4] phonons, polarons,[5] and magnons.[6] Of the various THz spectroscopy techniques that can probe these processes, THz time-domain spectroscopy (THz-TDS) and time-resolved terahertz spectroscopy (TRTS) provide especially rich insights because they coherently detect the broadband THz radiation (i.e., measure both the sign-resolved amplitude and the phase of the electric field), giving access to the full complex dielectric spectrum of the material of interest. This is particularly useful in the study of the frequency-dependent conductivity, since various conductivity models have distinct features in their complex form, e.g., the negative imaginary conductivity in the Drude−Smith model.[7] However, it is not necessarily trivial to extract this useful dielectric information from the measured time-domain data since the expressions that relate the two can become fairly complex. This problem is particularly pronounced for multilayered samples, where the expression must include terms for each of the many possible transmission-reflection paths through the sample layers. While these expressions can be simplified with various approximations,[8−10] these do not apply to all samples and can give erroneous results when misapplied. To avoid these approximations, several works have described methods that generate and process the full expressions programmatically for arbitrary sample geometries.[11−13] However, these works were not accompanied by software implementations of the algorithms discussed, limiting their usefulness.

Developments in recent years have exacerbated this issue, further increasing the need for accurate and user-friendly THz-TDS data analysis. First, researchers have turned their attention to novel materials and sample geometries for which traditional approximations may not be appropriate.[9,10] At the same time,

Received: May 20, 2021
Accepted: July 26, 2021
Published: August 6, 2021



ACS Publications

https://doi.org/10.1021/acs.analchem.1c02132

THz-TDS and TRTS spectrometers have become commercially available, bringing what was once a specialist technique to a wider range of laboratories. These two developments have placed competing demands on THz data analysis techniques. Properly analyzing novel samples requires the development of better approximations or more advanced data analysis techniques, increasing the complexity of THz data analysis. However, the increasing complexity of THz-TDS data analysis poses challenges for newer users. Since THz-TDS may be only one of many techniques that they use, they may not have the time to devote to developing appropriate approximations or techniques. The software presented here seeks to reconcile these competing needs, by providing an approximation-free method which can be easily used by any researcher.

The approach presented here avoids approximations by using a tree structure to model all possible transmission-reflection paths the pulse can take through the sample. This technique is applicable to any sample geometry without either modification or approximation. While this technique is somewhat involved, the MATLAB package provided here—Nelly, in reference to Fres*nel* coefficients—and its accompanying graphical user interface (GUI)—Cordouan, in reference to the first lighthouse to utilize a Fresnel lens—provide an accessible interface that any researcher can use.

We begin with a brief overview of THz-TDS measurements and the expressions that relate them to a sample's dielectric properties. We then illustrate how these expressions can become unwieldy and discuss common approximations that have been used to simplify them, as well as the shortcomings of these approximations. Finally, we describe the method our software uses to accurately extract dielectric information without approximation and benchmark it against common approximations.

EXPERIMENTAL SECTION

Numerical Implementation in MATLAB. The code described in this manuscript was developed in MATLAB and was tested for versions between 2016b and 2021a. It can also be used without a MATLAB license. The code is available at https://github.com/YaleTHz/nelly. For users with a MATLAB license, the library can be accessed by downloading the package and following the directions in the README file, which is available in the Supporting Information (SI) and on the GitHub repository. For users without a MATLAB license, the library can also be used in Python using the freely available MATLAB runtime. The GUI, called Cordouan, was built in MATLAB's App Designer and acts as a user-friendly front end for the Nelly library, allowing users to import their own data, set up sample and reference geometries, run the Nelly data analysis method, and export the results. It has been tested for MATLAB versions 2018b−2021a. As with the library, users without a MATLAB license can use the free MATLAB runtime to run the GUI in a standalone executable that is included in the package.

THz Spectroscopy Measurements. THz-TDS measurements were performed to benchmark the data analysis method. The spectrometer used has been described in detail previously.[14] Briefly, THz-TDS measurements were made on a home-built spectrometer based on a Ti:Al_2O_3 oscillator (Spectra Physics MaiTai, 800 nm center wavelength, ~10 nJ pulse energy, ~35 fs pulse duration, and 42 MHz repetition rate) and a pair of photoconductive antennae (Batop). The water sample was measured in a quartz cuvette with a 100 μm path length (Starna Cell 48-Q-0.1), and a measurement on the empty cuvette was used as a reference.

Finite-Element Simulations. Finite-element simulations of THz pulses passing through various sample types were performed to benchmark the method described here against various approximations. The simulated electric fields were processed with the method discussed here as well as several common approximations to retrieve the complex refractive index. These data were compared with the known dielectric parameters provided in the simulation. Simulations were carried out in CST Studio Suite 2020, a finite-element solver for Maxwell's equations.[15] The CST files and macros used are provided in the SI. For all calculations, the time-domain solver was used with electric boundary conditions in the X-direction and magnetic boundary conditions in the Y-direction to ensure a linearly polarized terahertz pulse that propagated in the Z-direction.

Implementing Approximations. We compared Nelly against four common approximations. The first considers only propagation terms and is implemented in the `just_propagation` function included with the package. The next two approximations add terms for reflection losses (i.e., transmission Fresnel coefficients) and single-layer etalons. These are implemented using options within Nelly that exclude certain paths, e.g., paths that include reflections for the reflection losses case and paths that include cross-layer etalons for the single-layer etalon case. Finally, the Tinkham equation[8] is implemented in the `tinkham` function that is included with the package.

RESULTS AND DISCUSSION

Description of Theory. THz-TDS systems measure the sign-resolved electric field of a THz pulse as a function of time. A typical experiment consists of two measurements: a pulse passed through the sample of interest and a pulse passed through a well-characterized reference or blank substrate material. The dielectric properties of the sample can then be extracted by comparing the two resulting pulses. Specifically, we calculate the change in each frequency component's amplitude and phase by Fourier transforming each pulse and taking the complex ratio of the two. This quantity is referred to as the transfer function (TF) and is defined as $\mathrm{TF}(\omega) = \frac{E_{\mathrm{sample}}}{E_{\mathrm{reference}}}$, where E_{sample} and $E_{\mathrm{reference}}$ are the complex-valued Fourier-transformed spectra of the sample and the reference, respectively, and ω is the angular frequency. The change measured in TF occurs because of the following three processes: propagation through the layers of the sample and reflection and transmission at the interfaces between them. The effect of each of these processes can be expressed in terms of simple expressions related to the sample's refractive index, i.e., Fresnel coefficients for reflection and transmission at the interfaces and complex exponential terms for propagation and absorption through the layers. Given a path through the layers of the sample, we can generate an expression for the change in amplitude and phase across the spectrum as a function of the sample's refractive index by multiplying the appropriate reflection, transmission, and propagation terms. For example, for the thick pellets often used in THz-TDS experiments, the sample pulse first transmits through an air−sample interface, then propagates through the sample, and finally transmits through the sample−air interface, while the reference pulse simply propagates through air. Figure 1 shows a schematic of this example, showing a cartoon of the observed signal and

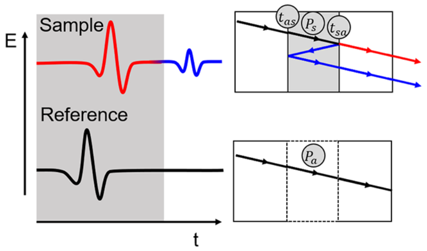


Figure 1. A schematic illustrating a THz-TDS transmission measurement of a thick sample. The left panel shows the sample and reference pulses within a possible measurement time window that is shaded in gray, which allows one to ignore the etalon shown in blue. The right panel shows the paths through the sample that correspond to the observed pulses (e.g., the blue reflection pulse at the left corresponds to the blue reflection path at the right). The terms above the paths make up their transfer functions (see eq 1). Note that the beams are shown at an angle only for ease of visualization. The software discussed here assumes a normal incidence.

corresponding pathways through the sample and reference for a measurement of a thick sample pellet. The main pulse is shown in red, while the first internal reflection (or etalon) is shown in blue.

Based on the transmission, reflection, and propagation pathways shown in Figure 1 and excluding the etalon pulse that arrived outside the measurement time (blue path), the expression for the transfer function can be written as

$$\frac{E_s}{E_a} = \mathrm{TF}(\omega, n_s) = \frac{t_{as}(\omega)P_s(\omega)t_{sa}(\omega)}{P_a(\omega)} \tag{1}$$

The subscripts a and s used in eq 1 and shown in Figure 1 refer to "air" and "sample," respectively. The terms $t_{jk} = \frac{2n_j(\omega)}{n_j(\omega) + n_k(\omega)}$ are the Fresnel coefficients for transmission from layer j to layer k, and the $P_j(\omega) = e^{-i\omega d_j n_j/c}$ terms describe propagation through layer j, where j and k are generalized to represent two adjacent layers. d_j and n_j are the thickness and the refractive index of layer j, respectively. Once we have such an expression, we can extract the complex refractive index of the sample using an optimization routine to find the refractive index n_s that minimizes the deviation between $\mathrm{TF}(\omega,n_s)$ and $\mathrm{TF}_{\mathrm{measured}}(\omega)$ for each frequency point ω.

However, not all samples will yield such simple expressions. Additional terms arise when we are forced to consider etalons due to decreased layer thicknesses. For example, in eq 1 we ignored the second path (i.e., the etalon) shown in blue in Figure 1, since reflections at the air−pellet interfaces will either not reach the detector (e.g., they propagate back toward the source) or will have to take at least two round trips through the thick pellet, leading to a large time delay between the main pulse and the reflected pulse. We can thus remove reflections by truncating the time window in which the data are collected (as in the shaded time window in Figure 1). However, if our pellet were thinner (less than ~100 μm), the time delay between the main and reflected pulses would be very small (less than ~1 ps for $n_s < 2$), making the two signals inseparable. For materials with sharp resonances, issues may arise even for thick samples with a good separation between the etalons. Such samples exhibit long-lasting ringing, which cannot be truncated without artificially broadening the resonances. When these reflection paths cannot be excluded in the time domain, we must account for them by adding additional expressions to the transfer function. When the layer is thin enough, we can assume that an infinite number of reflections will appear in the time window and thus compress the resulting infinite sum into a relatively simple expression of the form $\mathrm{FP}_{ijk}(\omega) = \frac{1}{1 - r_{jk}P_j^2 r_{ji}}$, which accounts for the etalons within layer j due to reflections at its interfaces with bordering layers i and k (also called Fabry−Perot (FP) reflections).[16] Here $r_{jk} = \frac{n_k - n_j}{n_j + n_k}$ is the Fresnel coefficient for reflection at the interface between layers j and k, and P_j is the propagation term for layer j. Such expressions have been used successfully in a range of works discussing refractive index extraction from single-layer samples.[17−20] Indeed, such an analysis is accurate for many cases where we can write out the transfer function, and Peretti et al. have provided an open source package to implement this approach for several sample geometries.[21] Such approaches are limited to predefined sample geometries, however, and as we add more layers to the sample itself[22] or to other features of the setup (e.g., a cuvette containing a liquid sample[23]), the transfer function can become complicated. Each of these additional layers will require its own set of Fresnel coefficients and may require Fabry−Perot terms as well. A particularly difficult situation arises when we have adjacent thin layers. In these cases, we need not only additional Fresnel coefficients and Fabry−Perot terms for each layer but also terms describing reflections involving both layers (see the path described in Figure 2). These terms are not as easily compressed as the single-layer Fabry−Perot terms and lead to lengthy transfer function expressions.

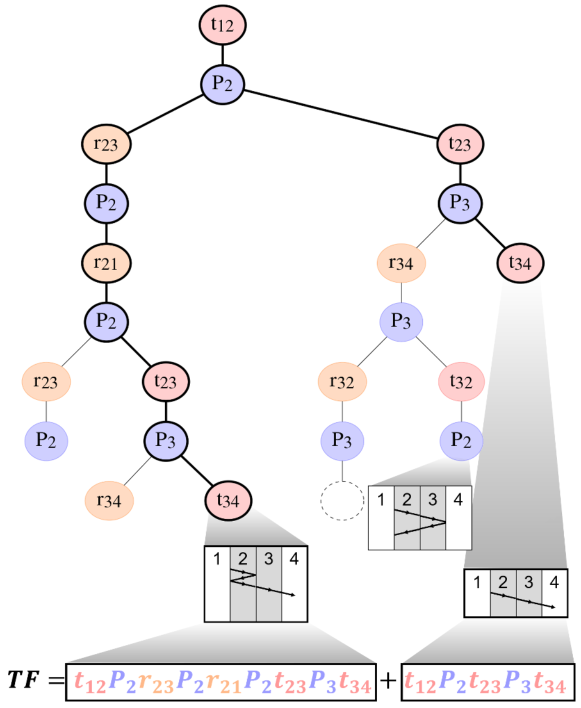

t12
P2
r23
t23
P2
P3
r21
r34
t34
P2
P3
r23
t23
r32
t32
P2
P3
P3
P2
r34
t34
1 2 3 4
TF = $t_{12}P_2r_{23}P_2r_{21}P_2t_{23}P_3t_{34}$ + $t_{12}P_2t_{23}P_3t_{34}$

Figure 2. The tree structure, which is used to enumerate all the paths the pulse can take through a two-layer sample. These paths are then summed to construct the full transfer function. Each node represents a transmission, reflection, or propagation event, for example, t_{ij} presents transmission at the interface between layers i and j. Each inset shows the path represented by the corresponding node. The empty dashed node shows one of the places where the tree building process has been terminated by a time or amplitude cutoff at that node. The expression at the bottom gives the transfer function (TF) that this tree would yield.

Overview of Approximate Methods. Facing these ballooning expressions, one approach has been to simplify these expressions by exploiting the properties of specific types of samples. For example, if the layer of interest is thick, the simplest approximation assumes that the propagation term dominates and ignores all reflection and transmission terms.[24] For the opposite case of an optically thin and highly conductive layer, a Taylor expansion of the transfer function yields the widely used Tinkham formula.[8,25] Using such approximations has substantial benefits: these approximations often yield closed-form expressions for the complex refractive index, which greatly simplify data analysis. However, these approximate approaches are not without their drawbacks, which arise because these approximations are rooted in particular assumptions about the sample. For example, while the Tinkham formula can be used for thin conductive films, it is not applicable in other cases, such as photoconductive films thicker than ~10 μm,[26] samples with a high dark conductivity,[10] or samples with background phonon modes.[9] Thus, before analyzing samples, researchers must carefully consider which of the approximations are appropriate. At best, this introduces extra effort; at worst, it can introduce erroneous conclusions into the literature when misuses of the approximations proliferate. Even if researchers are careful to apply only the appropriate approximations, the progress of the field may demand new approximations, since compelling new materials may have properties outside of the region where common approximations are valid. Researchers would then be left to choose from a bewildering range of possible approximations.

Fortunately, however, approximation-free approaches have surfaced a number of times in the literature.[11−13] Instead of addressing the problem of unwieldy strings of Fresnel coefficients by means of approximations, these works recognize that assembling and processing these expressions is precisely the sort of straightforward but tedious task that computers are well suited to. That is, these works discussed programs that take in any arbitrary sample geometry, i.e., the thicknesses and refractive indices of the layers in the sample, consider all possible paths the pulse could take through the sample, and construct the appropriate transfer function based on these paths. Thus, by minimizing the difference between the generated transfer function and the experimental results, it is then relatively straightforward to calculate the unknown complex refractive index of a sample without the need to use approximations. While this approach is more accurate, it is also much more difficult to implement than the closed-form approximations discussed above, and previous works utilizing this approach have not included any source code,[11−13] leaving interested groups to implement it for themselves. Facing this barrier, the adoption of the method has been limited. To address this issue, we provide an open-source implementation of this method in MATLAB along with a graphical user interface for our MATLAB library. Below, we will provide some details about our implementation and improvements upon existing algorithms, benchmark our method against previous approximations, and provide an overview of how users can use the software provided here.

Implementation Details. Given a particular sample geometry, i.e., the thickness for each layer in the sample as well as any known refractive indices, the goal is to create a transfer function that connects the refractive indices of the sample's layers to the amplitude and phase changes a THz pulse will experience upon passing through the sample. Upon emerging from the sample, the pulse will have been split into a number of reflections in addition to the main pulse. Therefore, to create the transfer function, the program must: (1) enumerate all possible paths through the sample, (2) determine which paths reach the detector, and (3) calculate the amplitude and phase changes associated with each of these paths. The final transfer function will then be the sum of the transfer functions associated with each path that reaches the detector. This transfer function is then used to extract the refractive index by numerically finding the value of the unknown refractive index that most closely matches the predicted transfer function to the transfer function observed experimentally.

As with previously described methods,[11−13] the implementation presented here uses a tree structure to accomplish these tasks. Figure 2 shows one possible tree for a simple two-layer sample (layers 1 and 4 represent the air surrounding the sample).

The tree representation consists of two types of nodes: layer nodes (P) and interface nodes (t or r), which represent transmission or reflection at an interface, respectively. For example, any path through the sample must begin by transmitting from the initial air layer (layer 1) into the first sample layer (layer 2). This is represented by the root node denoted t_{12}. Having transmitted from layer 1 to layer 2, the path will then proceed through layer 2. Thus, the root node has only one child, the layer node P_2, which represents propagation through layer 2. Here the paths diverge. Having passed through layer 2, the pulse reaches the 2 → 3 interface and will now split into a reflected portion and a transmitted portion. These are represented by interface nodes r_{23} and t_{23}, respectively. The algorithm continues as before from each of these nodes, following interface nodes into the natural layers and splitting layer nodes into the reflections and transmissions generated at their interfaces with the next layer. In this way, i.e., recursively generating the children for each node, the algorithm continues building the tree until we meet certain termination criteria (discussed below). Once built, the tree contains an enumeration of all possible paths through the sample; each node represents a path, which can be found by following the parent edges up to the root, i.e., all the events that led to that node. Thus, this recursive process of building the tree enumerates all possible paths through the sample, which is the first task mentioned above. Using this tree, the second task, i.e., determining which paths reach the detector, is straightforward. To reach the detector in a transmission geometry, the pulse must transmit through the interface between the final layer of the sample and the medium surrounding the sample. Finding the paths that reach the detector is then simply a matter of finding the nodes that represent this transmission, e.g., all t_{34} nodes in Figure 2. The transfer function associated with each node can then be found by taking the product of all the nodes on the path between the node in question and the root node. These products are shown for each of the t_{34} nodes in Figure 2, where the nodes contributing to the product are outlined in black. The final transfer function, which is simply a sum of these products, is also shown.

As mentioned above, an important factor in building the tree is determining when to terminate branches, since the recursive process would otherwise generate an infinite tree. One straightforward approach adopted by Cassar et al.[11] and Wilk et al.[12] is to simply propagate all branches to a certain depth (i.e., terminate all branches when the tree reaches a certain height). However, this can introduce a large number of extraneous nodes at a considerable computational cost. An accurate representation of the thinnest layers will require a higher depth than the thicker layers, but this higher depth will be applied to all layers regardless. Since the number of nodes increases exponentially with the depth, the number of extraneous nodes can grow extremely large. Greenall et al. avoided this by terminating on a branch-by-branch basis, determining whether or not to create children for each node.[13] We follow this approach, terminating on the basis of a time cutoff and an amplitude cutoff. As shown in Figure 1, only a certain window in the time domain is measured and processed, so any reflections that arrive outside this time window are excluded. Similarly, each succeeding reflection will be attenuated in an amplitude, and at a certain point this amplitude will be below the instrument's detection limit. Thus, when building the tree the program keeps track of the time and amplitude corresponding to each node's path and terminates any node whose time or amplitude would make it undetectable. For the amplitude cutoff, for example, the first node is assigned amplitude $A = 1$. Then, when generating the children for a given parent node, the child is given amplitude $A_{\text{child}} = |A_{\text{parent}}\text{TF}_{\text{child}}|$, where TF_{child} is t_{ij}, r_{ij}, or P_i depending on the type of the child node. If the amplitude A of a given node is less than the given value of A_{cutoff}, no children are generated. The value of A_{cutoff} can be determined by the noise floor of the instrument, for example. Similarly, the root node starts at time $t = 0$. Since propagating through layer i will add time $\frac{n_i d_i}{c}$ to the time elapsed, the children of any layer node will be given $t_{\text{child}} = t_{\text{parent}} + \frac{n_i d_i}{c_i}$. Any nodes with $t > t_{\text{cutoff}}$ are terminated, where t_{cutoff} is determined by the data collection time window. Finally, any branches that correspond to the pulse leaving the sample on the entrance side are also discarded, as these signals never reach the detector. For example, in Figure 2, t_{21} would represent a transmission back out of the front of the sample and away from the detector.

A summary of the full process is as follows: at each frequency point, we assume a trial value for the unknown refractive index and construct the full tree. From the tree, we determine the predicted transfer function value for the trial refractive index. We then adjust the trial value using an optimization routine to minimize the deviation between the predicted transfer function and the measured value, constructing a new tree for each trial value.

Benchmarking against Common Approximations. To demonstrate the importance of this method, we compared it against a range of commonly used approximations. Figure 3 shows the results of applying various approximations to THz-TDS measurements of water in a quartz cuvette (see the Experimental Section for details). Even for this strongly absorbing sample where propagation terms might be expected to dominate, Figure 3 shows reasonably significant deviations between the various approximations and Nelly, particularly at low frequencies.

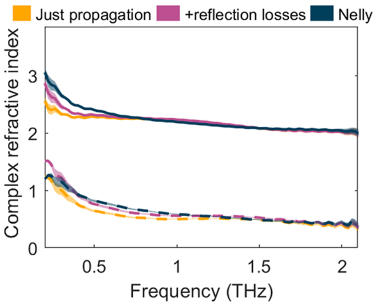


Figure 3. The refractive index extracted with Nelly is shown in dark blue and is compared with the refractive indices obtained with two common approximations. For the "just propagation" trace shown in orange, only propagation terms are considered. The "+Reflection losses" trace shown in magenta considers both propagation terms and transmission terms at each interface. The real and imaginary part of the refractive indices are shown as solid and dashed lines, respectively.

To further test Nelly, we simulated sample and reference pulses for various geometries in CST Microwave Studio, a finite-element electromagnetic field simulation software package. We then used various methods to extract the refractive index from these simulated data. By comparing the values extracted with the dielectric parameters specified in the simulations, we can assess the accuracy of each method for each sample geometry. This approach has the advantage of giving us a "true" refractive index against which we can benchmark our results (i.e., the dielectric parameters specified in setting up the simulation). The results of this comparison are shown in Figure 4, where the difference between the expected results and the results for each method are highlighted in red and blue (real and imaginary, respectively). While various approximations perform well for particular samples, the method discussed here gives accurate results in all cases.

The sample geometries were chosen to highlight situations in which each approximation is commonly applied. The first sample is a thick pellet with a small absorption feature around 0.3 THz. Here, all processes except propagation will have a negligible contribution to the overall changes in the amplitude and the phase. Because the sample is thick, we can remove any etalons by creating a window in the time domain and safely ignore them. While there will still be reflection losses at the front and back interfaces (i.e., t_{as} and t_{sa} terms, respectively), the phase and amplitude associated with these will be small relative to the propagation term, which is again due to the thickness of the sample. As expected, all approximations work reasonably well in this case.

In the next sample—water in a cuvette—the simpler approximations begin to fail. With a thinner sample layer, the relative contribution of the Fresnel transmission terms grows, and the etalons can no longer be removed in the time domain and must be considered. Thus, we only achieve accurate results once we have added the single-layer etalon terms. This can also be observed in the experimental results shown in Figure 3. The final geometry, which is common for TRTS experiments, introduces additional complications. These experiments measure the on − off difference in the time domain THz pulse as the sample is subjected to a chopped photoexcitation pulse. From such measurements, the refractive index of the photoexcited layer can be extracted. The thickness of this layer is typically taken to be the penetration length of the material undergoing photoexcitation. In the simulations shown here, we seek to extract the photoconductivity of a thin sample, where the pump-beam penetration length is such that the sample is not fully photoexcited but instead has a significant finite value that precludes the use of the Tinkham equation (often referred to as the thin-film approximation). Thus, we have a thin layer of the photoexcited material next to a thin layer of the remaining nonphotoexcited material. As with the previous thin sample, we cannot ignore transmission terms or single-layer etalons, so the first two approximations (propagation only and added reflection losses) do not perform well. Unlike the previous sample, however, including terms for single-layer etalons does not resolve the issues. This is because the adjacent thin layers in this

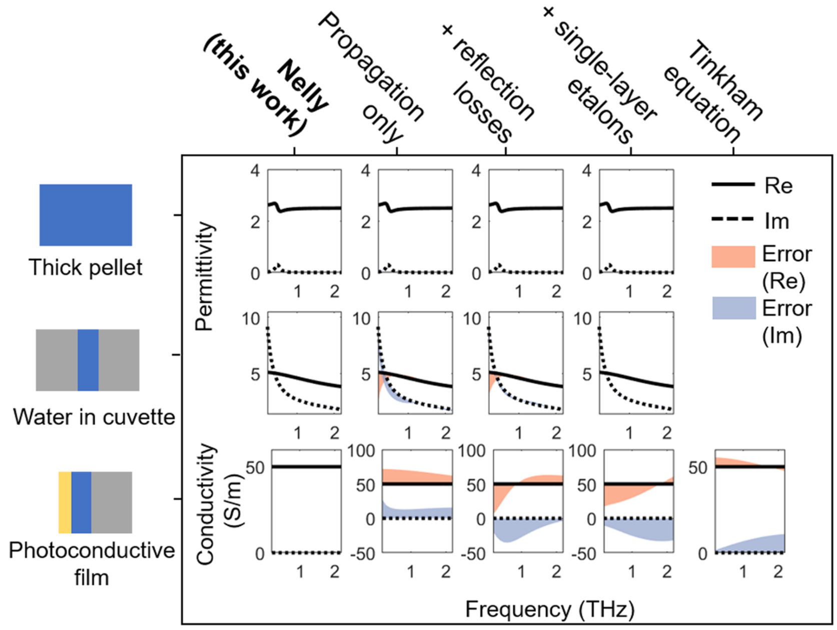
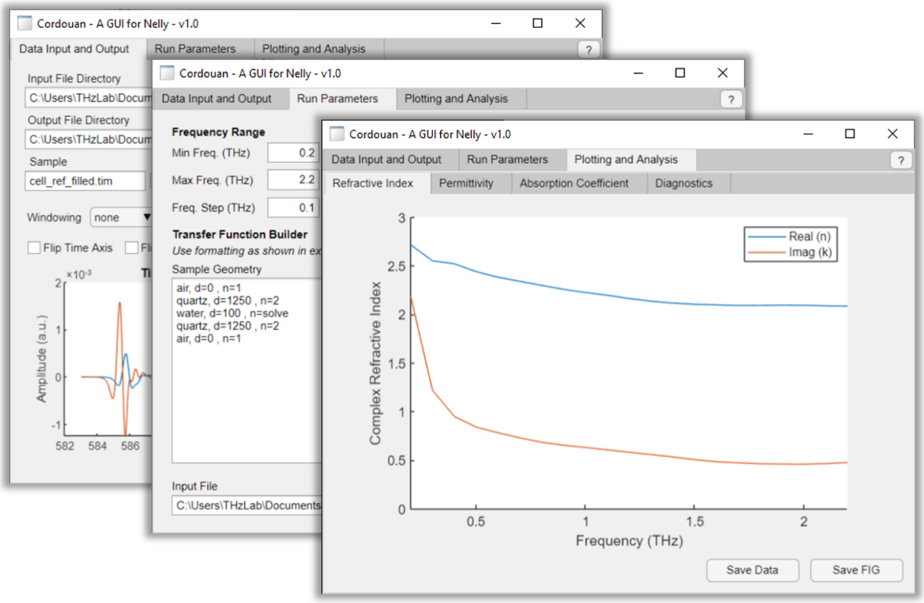


Figure 4. For a range of simulated samples modeling common sample types (Teflon mixed with an absorbing amino acid, an aqueous solution in a cuvette, and a photoexcited layer), the refractive index extracted with Nelly is compared with the refractive indices extracted from the same data with common approximations. The black lines show the dielectric parameters specified in the simulation (solid for real and dashed for imaginary). For each refractive index extraction method, the deviation between the input value and the extracted values is shaded in red (real) and blue (imaginary). The residuals are shown in Figure S2. The "Propagation only" column shows the results obtained with a transfer function that only includes the propagation terms. The next two columns each add additional terms, i.e., transmission terms (t_{ij}), and Fabry−Perot terms (FP_{ijk}), respectively. The first column shows the results from Nelly, illustrating the excellent agreement for the method presented here and the avoidance of inaccuracies associated with the commonly used approximations.



Input File Directory: C:\Users\THzLab\Docum

Output File Directory: C:\Users\THzLab\Docum

Sample: cell_ref_filled.tim

Windowing: none

□ Flip Time Axis □ Fl

Frequency Range

Min Freq. (THz) 0.2

Max Freq. (THz) 2.2

Freq. Step (THz) 0.1

Transfer Function Builder

Use formatting as shown in ex

Sample Geometry

```
air, d=0 , n=1
quartz, d=1250 , n=2
water, d=100 , n=solve
quartz, d=1250 , n=2
air, d=0 , n=1
```

Input File: C:\Users\THzLab\Documents

Figure 5. Screenshots of Cordouan showing the process of using the GUI to run Nelly. From left to right are the tabs for data input and output, run parameters, and data analysis. These tabs guide the user through the process of importing their data, configuring settings to run Nelly, building a transfer function for their specific sample geometry, and analyzing and exporting results.

sample will also give cross-layer etalons. Taking these into account using the method discussed here finally yields accurate results. For this sample, we also compare our results with the commonly used Tinkham equation. This approximation gives better results than the other approximations but still a noticeable error (11% error in the real part, for example), which can be avoided with our method. This illustrates the strength of the presented algorithm, particularly for emerging materials.

Using the Software. While similar methods have been demonstrated previously,[11−13] this work is the first to include software that other research groups can use to apply the method. The software package, Nelly, offers users two interfaces with which to analyze THz spectroscopy data: a MATLAB library and a GUI called Cordouan, both of which can be used with or without a MATLAB license. Users of the MATLAB library can process their data by constructing an input file that specifies the sample and reference geometries and passing this along with their experimental data as an argument to the `nelly_main` function. The input file also specifies parameters such as the frequency range in which to extract the refractive index and the amplitude cutoff for reflections (discussed above). A sample input file is included and discussed in greater detail in the README file (SI). The `nelly_main` function outputs the extracted complex-valued refractive index (`n_fit`) as a function of frequency in THz (`freq`) as well as some intermediate results from the processing to help users to diagnose any issues. The library has been designed to be modular, allowing for easy extensibility. For example, the task of building the tree that is discussed above is delegated to a self-contained class (`tf_node`). Thus, while `nelly_main` assumes a transmission geometry by default, users interested in reflection geometries could input their geometry into the `tf_node` class and then select the appropriate nodes in the resulting tree for their transfer function, that is, nodes representing transmission out of the first layer rather than the last. We assume here that the reflection is normal to the input wave, but in principle it could also be rewritten for arbitrary angles of incidence. We welcome such modifications and will happily review extensions like this for inclusion in the package. Users can submit modifications for review through pull requests to the package's GitHub repository (https://github.com/YaleTHz/nelly/pulls). Users who encounter issues can find help in the manual included in the Supporting Information, as well as by posting an issue on the GitHub page or contacting the authors.

We have also implemented a GUI called Cordouan to allow easy access to the features provided by Nelly. Cordouan is organized into three different tabs that guide the user through the process of importing data, setting parameters for running Nelly, analyzing the data, and finally exporting the results. Figure 5 shows each of the tabs and illustrates the typical workflow, where the user imports data, checks data, sets parameters, and then views the resulting complex refractive index (see the SI for more details). In addition to setting basic parameters such as FFT settings and frequency limits, Cordouan includes an easy-to-use method for building transfer functions for a particular geometry, as shown in the middle tab in Figure 5. The user can input their sample geometry using an easily interpretable syntax, which is then converted to the structure required for Nelly. In addition, the user can use the parameters that they input into Cordouan, including the transfer function builder, to save a new input file that can be loaded for later use. After running Nelly, Cordouan automatically plots the results in terms of the complex refractive index, the complex permittivity, and the absorption coefficient. In addition, diagnostic data are provided for the user to quantify how well the experimental and calculated transfer function match and to inspect the raw fast Fourier-transformed amplitude and phase spectra of the sample and reference. From here, raw data can be saved for further processing (e.g., calculating the complex conductivity) and plotting or the plots can be saved directly in a variety of different file formats.

■ CONCLUSION

As commercially available THz spectrometers open up the technique to a wide range of research groups, the need for accurate and user-friendly data analysis methods has become ever more acute, especially as researchers have turned their attention toward novel materials where commonly used approximate methods do not apply. The software package presented here (Nelly) addresses this need. Our method considers all relevant reflections within the sample, avoiding potentially inappropriate assumptions and making it applicable to any sample geometry. For a range of common samples, we demonstrate that Nelly is in excellent agreement with the expected dielectric properties even when common approximations are not. The presented algorithm is implemented as a MATLAB library and a GUI, making it accessible to a wide range of users. Both the library and GUI can also be used without a MATLAB license. It is our hope that this will allow researchers in the field to study a wider range of samples, unconstrained by the limitations of available data processing methods.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.analchem.1c02132.

User manual for Nelly (PDF)

Current version of the described programs; the most up-to-date version can be found at https://github.com/YaleTHz/nelly (ZIP)

User manual for Cordouan (PDF)

Zip file containing CST projects for simulation (ZIP)

■ AUTHOR INFORMATION

Corresponding Authors

Uriel Tayvah − *Department of Chemistry and Energy Sciences Institute, Yale University, New Haven, Connecticut 065111, United States;* orcid.org/0000-0002-2759-131X; Email: uriel.tayvah@yale.edu

Jens Neu − *Department of Chemistry and Energy Sciences Institute, Yale University, New Haven, Connecticut 065111, United States; Department of Molecular Biophysics and Biochemistry and Microbial Sciences Institute, Yale University, New Haven, Connecticut 06511, United States;* orcid.org/0000-0002-1054-0444; Email: Jens.Neu@yale.edu

Authors

Jacob A. Spies − *Department of Chemistry and Energy Sciences Institute, Yale University, New Haven, Connecticut 065111, United States;* orcid.org/0000-0002-0148-4823

$^\nabla$**Charles A. Schmuttenmaer** − *Department of Chemistry and Energy Sciences Institute, Yale University, New Haven, Connecticut 065111, United States;* orcid.org/0000-0001-9992-8578

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.analchem.1c02132

Notes

The authors declare no competing financial interest.
$^\nabla$Deceased July 26, 2020.

■ ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant 1954453. U.T.T. acknowledges the support of the National Science Foundation Graduate Research Fellowship (DGE-1752134). The authors would like to thank Gary W. Brudvig for his input on this work, Daiwei Zhang for beta testing parts of the software, and Sarah Ostresh for contributing to the cover art.

REFERENCES

(1) Spies, J. A.; Neu, J.; Tayvah, U. T.; Capobianco, M. D.; Pattengale, B.; Ostresh, S.; Schmuttenmaer, C. A. *J. Phys. Chem. C* **2020**, *124* (41), 22335−22346.

(2) Neu, J.; Schmuttenmaer, C. A. *J. Infrared, Millimeter, Terahertz Waves* **2020**, *41*, 1366.

(3) Neu, J.; Stone, E. A.; Spies, J. A.; Storch, G.; Hatano, A. S.; Mercado, B. Q.; Miller, S. J.; Schmuttenmaer, C. A. *J. Phys. Chem. Lett.* **2019**, *10* (10), 2624−2628.

(4) Falconer, R. J.; Markelz, A. G. *J. Infrared, Millimeter, Terahertz Waves* **2012**, *33* (10), 973−988.

(5) Ziwritsch, M.; Müller, S.; Hempel, H.; Unold, T.; Abdi, F. F.; van de Krol, R.; Friedrich, D.; Eichberger, R. *ACS Energy Lett.* **2016**, *1* (5), 888−894.

(6) Grishunin, K.; Huisman, T.; Li, G.; Mishina, E.; Rasing, T.; Kimel, A. V.; Zhang, K.; Jin, Z.; Cao, S.; Ren, W.; Ma, G.-H.; Mikhaylovskiy, R. V. *ACS Photonics* **2018**, *5* (4), 1375−1380.

(7) Smith, N. V. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2001**, *64* (15), 155106.

(8) Glover, R. E.; Tinkham, M. *Phys. Rev.* **1957**, *108* (2), 243−256.

(9) La-o-vorakiat, C.; Cheng, L.; Salim, T.; Marcus, R. A.; Michel-Beyerle, M.-E.; Lam, Y. M.; Chia, E. E. M. *Appl. Phys. Lett.* **2017**, *110* (12), 123901.

(10) Ulatowski, A. M.; Herz, L. M.; Johnston, M. B. *J. Infrared, Millimeter, Terahertz Waves* **2020**, *41* (12), 1431−1449.

(11) Cassar, Q.; Chopard, A.; Fauquet, F.; Guillet, J.; Pan, M.; Perraud, J.; Mounaix, P. *IEEE Trans. Terahertz Sci. Technol.* **2019**, *9* (6), 684−694.

(12) Wilk, R.; Pupeza, I.; Cernat, R.; Koch, M. *IEEE J. Sel. Top. Quantum Electron.* **2008**, *14* (2), 392−398.

(13) Greenall, N. R.; Li, L. H.; Linfield, E. H.; Davies, A. G.; Cunningham, J. E.; Burnett, A. D. *2016 41st International Conference on Infrared, Millimeter, and Terahertz waves (IRMMW-THz)* **2016**, 1−2.

(14) Williams, M. R. C.; Aschaffenburg, D. J.; Ofori-Okai, B. K.; Schmuttenmaer, C. A. *J. Phys. Chem. B* **2013**, *117* (36), 10444−10461.

(15) Weiland, T. *Int. J. Numer. Model. Electron. Netw. Devices Fields* **1996**, *9* (4), 295−319.

(16) Neu, J.; Schmuttenmaer, C. A. *J. Appl. Phys.* **2018**, *124* (23), 231101.

(17) Duvillaret, L.; Garet, F.; Coutaz, J.-L. *IEEE J. Sel. Top. Quantum Electron.* **1996**, *2* (3), 739−746.

(18) Dorney, T. D.; Baraniuk, R. G.; Mittleman, D. M. *J. Opt. Soc. Am. A* **2001**, *18* (7), 1562.

(19) Pupeza, I.; Wilk, R.; Koch, M. *Opt. Express* **2007**, *15* (7), 4335−4350.

(20) Scheller, M.; Jansen, C.; Koch, M. *Opt. Commun.* **2009**, *282* (7), 1304−1306.

(21) Peretti, R.; Mitryukovskiy, S.; Froberger, K.; Mebarki, M. A.; Eliet, S.; Vanwolleghem, M.; Lampin, J.-F. *IEEE Trans. Terahertz Sci. Technol.* **2019**, *9* (2), 136−149.

(22) van Mechelen, J. L. M. *Langmuir* **2014**, *30* (43), 12748−12754.

(23) Tielrooij, K. J.; Timmer, R. L. A.; Bakker, H. J.; Bonn, M. *Phys. Rev. Lett.* **2009**, *102* (19), 198303.

(24) Baxter, J.; Schmuttenmaer, C. A. Time-Resolve Terahertz Spectroscopy and Terahertz Emission Spectroscopy. In *Terahertz Spectroscopy: Principles and Applications*; Dexheimer, S. L., Ed; CRC Press: Boca Raton, FL, 2017; pp 73−115.

(25) Lloyd-Hughes, J.; Jeon, T.-I. *J. Infrared, Millimeter, Terahertz Waves* **2012**, *33* (9), 871−925.

(26) Neu, J.; Regan, K. P.; Swierk, J. R.; Schmuttenmaer, C. A. *Appl. Phys. Lett.* **2018**, *113* (23), 233901.