

Current Version compiled February 14, 2019.

Contents

1	Introduction	3
2	How to Read this Manual and the Source Code	3
3	License Agreement	3
4	General Remarks	4
5	Choosing the Correct Program and Variables	5
6	Overview of Programs	5
6.1	Concept	6
6.2	Evaluation_TDS.m	6
6.2.1	Input	7
6.2.2	Output	7
6.3	Evaluation_TDS_Pellets.m	8
6.3.1	Input	8
6.3.2	Output	9
6.4	Evaluation_TRTS.m	9
6.4.1	Input	10
6.4.2	Output	11
6.5	Assigment_sheet.m	12
7	Installation and Needed Programs	13
8	General Info	14
9	Explanation of Really Important/Confusing Variables	15
9.1	teflon	15
9.2	dteflon	15
9.3	dpila	16
9.4	averaging_in_time, used in Evaluation_TDS_Pellets.m.m	16
9.5	frequenzmitteln used in all programs	16
9.6	ordnerspeichern used in all, defined in Assigment_sheet.m or elsewhere!	16
9.7	load_substrat, part of Evaluation_TDS.m and Evaluation_TRTS.m .	16
9.8	tapecell part of Evaluation_TDS.m and tapecell part of Evaluation_TRTS.m	16

9.9	lorentzianfit called via optimization toolbox from Evaluation_TDS_Pellets.m	17
9.10	nafion.literature used in Evaluation_TRTS.m	17
9.11	zwischenplotten used in all	17
9.12	thinfilm in Evaluation_TRTS.m	17
9.13	bruggeman used in Evaluation_TRTS.m	17
10	Explanation of Functions	18
10.1	faxis.m(time , <i>number of points</i>)	18
10.2	retrieval_of_d_mark3(thickness,modelparameters); called by TDS and Evaluation_TDS_Pellets.m via optimization toolbox	18
10.3	retrieval_of_n_airref_tapecell, called via fmincon from Evaluation_TDS.m, only called if tapecell=true	19
10.4	retrieval_of_n_airref_nooptimization(n,k), used in Evaluation_TDS.m and Evaluation_TDS_Pellets.m	19
10.5	retrieval_of_n_airref_thinlayer_tapecell.m	19
10.6	retrieval_of_n_airref_thinlayer_pila.m	20
10.7	retrieval_of_n_airref_thinlayer_Mk2_Phasesep, part of Evaluation_TDS.m	20
10.8	retrieval_of_sigma, part of Evaluation_TDS.m	20
10.9	retrieval_of_n_relativtoteflon_nooptimization.m, used in Evaluation_TDS_Pellets.m	21
10.10	importfiles_script.m and import_for_timfiles used in Evaluation_TRTS.m, respectively in Evaluation_TDS.m, Evaluation_TDS_Pellets.m	21
10.11	new_Eon_durch_Eoff_no_model_v0.m called in Evaluation_TRTS.m via optimization toolbox	21
10.12	new_Eon_durch_Eoff_DS_v0.m used in Evaluation_TRTS.m via opti- mization toolbox	21
10.13	DSfit.m called via optimization toolbox in Evaluation_TRTS.m	22
10.14	DS_seperate.m via fittoolbox in Evaluation_TRTS.m	22

1 Introduction

This software package includes three programs and associated functions. This means about 5000 lines of code. The program was written for our experiments and WILL need adjustment to handle your data! I would be surprised if unzipping this code and pressing the run-button would do the job. You will need to invest a few hours in this! So this paragraph briefly describes why it is worth investing some time in this program.

The programs provide well tested function that calculate refractive index from Terahertz - Time Domain Spectroscopy (THz-TDS) measurements or Transient Terahertz Spectroscopy (TRTS) measurements. These function DO NOT need any model or assumptions. All you need to know is the thickness of the sample and the substrate.

The programs use multiple different approaches. This allows to directly compare for example, the thin-film approximation with the results gained without a model. And with results calculated assuming Drude-Smith model for example.

The programs provide detailed reports and intermediate figures. This allows for easy tracking of bugs and fixing of these.

Parts of the program in particular the variable names are in German. German is a beautiful language and the best way to learn it is from a badly documented source code!

2 How to Read this Manual and the Source Code

(Cyan) color is used for global variables, while usual text font is used for variables. The source code has several references to my lab notebook, for example JN98 or JN 2 20. We might scan and upload some of these page, or even tex them. However, as long as you do not have any access to my lab notebook, I suggest to read the published papers, they explain some of the details.

Most variables are explained in the source code, using % comments. Important or somehow counterintuitive variables are explained in this manual. If a variable is not explained in this document nor the source code, and you cannot find out what this variable does, chances are good it does not do anything at all.

3 License Agreement

All rights reserved according to GPL-3 (or any later version) license, Jens Neu
February 14, 2019:

Copyright © 2019 Jens Neu

This program is free software: you can redistribute it and/or modify it under

the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

4 General Remarks

- The Software is "as is".
 - I appreciate reported bugs and suggestion to improve the program.
 - We are no software developer, so please be patient if there are errors.
 - The software can be used for non-commercial purposes, free of charge.
 - The software can be modified without limitation, as long as not commercially used.
 - Proper scientific practice:
 - * When you use data generated by this software, please be sure to cite the paper in which this software and the underlying equations are presented
 - * [3] "Exploring the solid state phase transition in DL-norvaline with terahertz spectroscopy" by Jens Neu and Coleen T. Nemes and Kevin P. Regan and Michael R.C. Williams and Charles A. Schmuttenmaer published in Physical Chemistry Chemical Physics, 20, p276-283, 2017 for **Evaluation_TDS_Pellets.m**
 - * [5] "Tutorial: An Introduction to Terahertz Time Domain Spectroscopy (THz-TDS)" by Jens Neu and Charles A. Schmuttenmaer, published in Journal of Applied Physics, 124,22 2018 for **Evaluation_TDS_Pellets.m** and for **Evaluation_TDS.m**
 - * [4] "Applicability of the Thin-film Approximation in Terahertz Photoconductivity Measurements" by Jens Neu and Kevin Regan and John R. Swierk and Charles A. Schmuttenmaer published in Applied Physics Letters, 113,22, 2018 for **Evaluation_TRTS.m**
- Any typos in this manual and the comments are intended. However, if some parts are wrong to a degree that makes understanding difficult, please feel free to contact us and we will fix it.

- Most variables and functions are explained as comments in the source code. The more complicated ones are discussed in this read.me file.
- We will create a FAQ section, maybe as part of this manual. Therefore, when you contact us with bug reports or questions, please indicate if this email can be posted in the FAQ section.

We strongly recommend that the programs are tested with well-known materials. For example, TDS measurements of silicon wafers, or TRTS of Silicon or GaAs.

5 Choosing the Correct Program and Variables

In the final version all programs will be merged into one. However, this will take while. See the THz-TDS tutorial for supported geometries [5].

The programs can be used for the following geometries:

1. Air/Substrate/Sample/Air in this case the tape cell function should be turned off (tapecell = false) (**Evaluation_TDS.m**)
2. Air/Substrate/Sample/Substrate/Air in this case the tape cell function should be turned on (tapecell = true) (**Evaluation_TDS.m**)
3. Air/Sample just set dteflon = 0 (**Evaluation_TDS.m**)
4. Air/Sample in pressed pellet. This is particularly interesting if you use a transparent polymer (teflon, PE) and mix a strongly absorbing material in it. The software applies Bruggeman and Maxwell Garnett to calculate the inclusion material (aka, the actual sample material) (**Evaluation_TDS_Pellets.m**)
5. Optical pump THz probe: Air/Sample (**Evaluation_TRTS.m**)
6. Optical pump THz probe: Air/Substrate/Sample (**Evaluation_TRTS.m**)
7. Optical pump THz probe: Air/Substrate/Sample/Substrate/Air (**Evaluation_TRTS.m**)

6 Overview of Programs

Please note that most variable names have historical origin:

- The substrate is always called "Teflon", do not worry about that. If you use other substrate, just insert correct thickness and refractive index.
- The same substrate is however sometimes also called sio2 (in Evaluation_TRTS.m).

- The sample is usually called pila, or amino, or DLN .
- In Evaluation_TRTS.m the sample is called sno2 .
- Some variables have German names.

6.1 Concept

The programs are handling frequency resolved THz data, either from TDS or TRTS measurements. The data is imported together with known values of substrate thickness, refractive index, and so on. The calculations solve Transmission-equations in the frequency domain. The solving in the frequency domain is usually discretized. This means you define a starting frequency (`fanfang`) a step size (`frequenzschritt`) and an end point (`fende`).

The program uses a large number of `global variables`. These are needed to transfer data from the main program to functions. In particular, for the optimization routines. Commonly anything you give to a function as an attribute in an optimization will be optimized. However, this should not happen to the measured data! This is the data and must not be changed! Therefore, global variables transfer the data to the function, and also some results back. Be extraordinary careful when changing a global variable, in particular renaming it!

6.2 Evaluation_TDS.m

This is the core program of the package. The purpose of this program is to calculate the refractive index of a sample of known thickness. This is in general a fairly challenging calculation, if no model is used. Without a model the transmission function for each frequency point needs to be inverted. Several publications use a separation approach for that. Meaning they assume that the phase of the THz field is only defined by the real part of the refractive index, and the amplitude solely by the absorption. This assumption is wrong and not made in this program!

The real part of the refractive index must be considered when the amplitude transmission is evaluated. This transmission includes reflection losses on the interface sample/air. These reflection effects are considered in the program. The Fresnel transmission coefficient is complex, and the imaginary part of this coefficient will change the phase of the transmitted wave. This phase change is also considered in the program, which allows a precise calculation of the complex refractive index, for loss-free and lossy sample materials, without prior assumption of the losses!

The user of this software will not have to choose whether the refractive index is mainly real or the sample is highly absorbing. The program will use the temp-data to approximate the real part of the refractive index, this approximation is then used as starting point of the numerical refractive index retrieval.

As there is no model linking the refractive index to the frequency ($n(\omega)$) each refractive index is independent. Meaning for any frequency point the complex refractive index is calculated. Therefore the program is written in a retrieval-loop that calculates the refractive index for each frequency point. Starting by a user specified frequency (`fanfang`) with a specified step size (`frequenzschritt`) until reaching the specified end point (`fende`). This means that depending on step size and range hundreds of calculations are performed. So the user should choose carefully what step size is needed.

The big advantage of not using any model is, that no assumptions need to be made before hand. This provides an unbiased result which then can be checked against models. This is in general preferable to assuming a model in the beginning as for a completely un-known material any assumptions are premature.

The big disadvantage is that each frequency point is independent. Therefore, the noise is directly transferred into the result. While models allow for an averaging of noise, the no model calculation will result in noisy individual points. This makes the calculation in itself less stable. Another point is that the phase is only defined $\pm 2\pi, \pm 4\pi, \pm 6\pi, \dots$, meaning the solution is not unique. This problem is addressed by determining the refractive index in the time domain and using this as a starting point of the calculations. The maximum change is limited to $\pm 20\%$ from one frequency point to the next. This avoids any massive jumps, associated with 2π jumps. It however could also suppress sharp resonances in the sample! The frequency step size must be chosen small enough to capture such resonances!

6.2.1 Input

Mandatory input:

- TDS measurement of air and/or reference substrate.
- If no reference measurement is supplied, the refractive index of the substrate needs to be supplied.
- TDS measurement of the sample.
- Thickness of the sample (can be optimized via built-in fitting routine).
- Thickness of the substrate (can be optimized via built-in fitting routine).

6.2.2 Output

The program will create the following output

- Refractive index of the sample, calculated without model.

- Refractive index of the substrate, calculated without model (can also be loaded from previous experiments!).
- Optimized Substrate Thickness, using Sellmeier, Cauchy, or Debye Model.
- Optimized Sample Thickness, using Sellmeier, Cauchy, or Debye Model.
- Absorption Coefficient .
- Conductivity (needs some model assumptions).

6.3 Evaluation_TDS_Pellets.m

This program assumes a sample material pressed into a pellet with a host material. We usually use Teflon, however PE, and other material have been used by other groups. The program needs the mixing ration of sample and host material (in weights/masses). Furthermore, the density of host and sample is needed. This means you need several pure host-material samples, press them and measure the thickness in dependence of the weight. This calibration curve is then used to determine the host-material density (in our case the thickness per mass is used in $\frac{\mu\text{m}}{\text{mg}}$). The sample density can be gained from XRD unit cell measurements and need to be handed over in the assignment sheet. Please make sure to also insert the diameter of the used pellet press dye.

The pellet version is a sister-program to TDS. The general working principle is identical. The main difference is that this program also uses effective medium theory (EMT). EMT is used to calculate the refractive index/absorption coefficient of a sample materials (like amino acids) pressed into a host material (like Teflon). This additional calculation also needs the accurate mixing ration (volume fractions) of the materials.

This program is poorly commented. Some of the variables are explained in Evaluation_TDS. There is a massive similarity between both programs. Future versions of this software will merge both programs in one. For now, if a variable is not well commented in Evaluation_TDS_Pellets.m, check Evaluation_TDS.m for documentation.

6.3.1 Input

Mandatory input:

- TDS measurement of air and/or reference substrate.
- If no reference measurement is supplied, the refractive index of the substrate needs to be supplied.

- TDS measurement of the sample.
- Thickness of the sample (can be optimized via built-in fitting routine).
- Thickness of the substrate (can be optimized via built-in fitting routine).

Additional input if effective medium theory is used

- Volume fraction of inclusion in the host material. The current version calculates the volume fraction from insert density and mass.

6.3.2 Output

The program will create the following output

- Complex Refractive index of the sample, calculated without model.
- Complex Refractive index of the inclusion material, calculated with Bruggeman and Maxwell-Garnett Eq.
- Complex Refractive index of the substrate, calculated without model (can also be loaded from previous experiments).
- Absorption Coefficient of the inclusion.

6.4 Evaluation_TRTS.m

This program is written for evaluation of frequency resolved, optical pump, THz probe data. The experimental condition assumes that two measurements are performed, one measurement without photoexcitation (called off). The second measurement is the transmission with photoexcitation (on). In the program we assume that this on measurement is performed as a difference measurement. Meaning that the lock-in was locked onto the on/off modulation frequency and the difference between on and off is measured. This result is then combined with the off-measurement to provide the actual on measurement, instead of the difference signal. If the user uses another chopper scheme the input data handling must be modified. See the figures activated by `zwischenplotten = true`, and compare these result with the import into previously used programs.

The program supports a large variety of different conditions.

- Air/Sample
- Air/Substrate/Sample
- Air/Substrate/Sample/Substrate/Air

- sample fully excited (thin sample) and also partially excited (thick sample)

This means you need to choose the correct Boolean variables to ensure that the program calculates the actual conditions. See the comments in the source code and the detailed explanation of the variables in this manual to choose the correct system.

The program will also need the following input of your system.

- Sample Thickness [dpila](#).
- Sample Complex-Refractive index without photo-excitation (nmean_sno2_test, nmean_sno2, kmean_sno2_test, kmean_sno2).
 - These values should be determined with TDS measurements.
 - It is possible to import literature values here.
 - It is also possible to use frequency independent refractive indices here.
- Substrate Thickness [dteflon](#).
- Substrate Complex Refractive index (nteflon).
- Optical penetration length /absorption length for the pump beam.
 - This value is compared to the sample thickness.
 - Depending on whether the pump beam is fully absorbed or not the correct quantum efficiency is calculated.
 - The correct model is chosen, whether the material is fully or partially photo-excited. See [4] for a detailed description.

6.4.1 Input

Mandatory input:

- TRTS measurement of Sample (referred to as "on").
- TDS Measurement of Sample (referred to as "off").
- The current version uses a difference input signal for the "on" case and then builds the actual measurement "on" as input_on + input_off. See description of the program for details.
- Refractive index of Sample.
- Thickness of Sample.

- Refractive index of Substrate.
- Thickness of Substrate.

optional input:

- Optical penetration length of pump beam, needed to calculate quantum efficiency and to use the correct model, described in J. Neu, APL 113 (2018) [4]

6.4.2 Output

The program will create the following output:

- Refractive index (complex) of the photoexcited material (Calculated without a model).
- Refractive index (complex) of the photoexcited material (Calculated with Drude-Smith model).
- Photo-conductivity (complex) (Calculated without a model, assumption that ϵ of the sample is not changed due to photo excitation).
- Photo-conductivity (complex) (Calculated with Drude-Smith model, assumption that ϵ of the sample is not changed due to photo excitation).
- Photo-conductivity (complex) (Calculated with Thin-Film Approximation).
- Can use effective medium theory to calculate conductivity of inclusion.

It will also perform the following fits

- Photo-conductivity (complex) (Calculated without a model) fit with Drude-Smith (Fit Optimization Toolbox).
- Photo-conductivity (complex) (Calculated with Thin-Film Approximation) fit with Drude-Smith.
- Photo-conductivity (complex) (Calculated without a model) fit with Drude-Smith (Fit LSQ).
- Photo-conductivity (complex) (Calculated without a model) fit with Drude-Smith modified by Tyler et al. [1].
- Photo-conductivity (complex) (Calculated with Thin-Film Approximation) fit with Drude-Smith modified by Tyler et al. [1].

6.5 Assignment_sheet.m

Evaluation_TDS uses the assignment sheet as slave, the pellet program uses the assignment sheet as master. Meaning in Evaluation_TDS, you pick the measurements in the main program, in Evaluation_TDS_Pellet, the Assignment_sheet.m has a vector `temprangeall = []`, which hands over the points to the Evaluation program. So in this case you need to open the assignment to pick the different series. This will be adjusted in a newer version.

- This file connects a unique sample ID to the corresponding raw-data.
- This files needs to be edited whenever a new measurement is added.
- The big advantage is: The main programs will not need to be edited. So you only need to add your new measurement in this file, not in the main-files.
- The sample identifier is called "temp" and must be a number.
 - This is historically, because the program was written for temperature resolved TDS
 - * The temp can have any values! So for example:
 - * Number of Sample (for example 3) * 1000
 - * Real Temperature
 - * Spot on the sample or repetition.
 - * For example, the third sample, measured at the second position at room temperature would have a "temperature" of $(3 * 1000) + 300 + 2 = 3002$.
 - * What numbers you use does not matter, but they need to ID each sample/measurement uniquely!
- Each temp must be assigned to 3 files. Sample, Reference, and Air
- The main programs can be used with only two measurements (Air/Sample or Reference/Sample). However, the assignment_sheet needs 3. So if you only use two and measured two, you have to lie to the assignment_sheet and assign the missing measurement to either of the other two! Alternatively, you can reprogram everything.
- The big advantage of this ID-system is, that the main program can run on vectors and open a large number of files/measurements in a loop.
 - For example, Sample 3, for different temperatures can be defined as:

- $\text{Temprangeall} = [300 \ 200 \ 100 \ 50] + 3000$; Which will open 3300, then 3200, then 3100, and 3050. It will calculate the refractive index of these four measurements and plot everything in the same figure.
- The user can add additional parameters to the function to allow for a better tailored assignment, or use one sheet per experiment/sample type.

7 Installation and Needed Programs

The following things need to be done before you can run the program.

- Matlab
 - The program is tested with 2016a and 2017a, however matlab is usually pretty good with forward compatibility.
 - The optimization toolbox (Matlab) Tested with Version 7.6.
 - For some operations you will need Curve Fitting Toolbox tested with Version 3.5.5.
- Functions:
 - You will need a function to import your data files, I recommend use the build-in matlab import feature to create a function for future need.
 - The function ”‘Import_for_timfiles’” can be used as template, however it imports our LabVIEW .txt. files and you will need to modify it.
 - The function needs to import a time vector and an amplitude vector.
 - The function needs to import a time vector and an amplitude vector.
- The program uses an assignment file. This file associates a measurement number with the actual files needed. For example: Teflon_at_300 K would be assigned to Jan16_001, Jand16_002, were one measurement is reference, the other sample. This file needs to be built by the user. Historically, the function only support temperature input, and each sample type had its own assignment file.
 - Historically, each measurement has a temperature assigned! Therefore, the usual structure is
 - * Number of Sample (for example 3) * 1000
 - * Real Temperature
 - * Spot on the sample or repetition.

- * For example, the third sample, measured at a second position at room temperature would have a "temperature" of $(3 * 1000) + 300 + 2 = 3002$.
- * What numbers you use does not matter, but they need to ID each sample/measurement uniquely!
- Please be sure to link YOUR file naming here. Our files are called Jan16_001.tim for example. Adjust this to open the files saved by your spectrometer.

8 General Info

- Data Format:
 - The used units are inconsistent within the program!
 - Time is in ps, and needs to be in ps in the input. So if needed, reformat the input function.
 - Frequency in THz, except otherwise stated.
 - Some thicknesses are μm , but converted in meter for most calculations
 - All fit parameter are in "number close to 1 units". This is simply because most fit routines are more stable if they are close to unity. Each function has a comment that tells you if it is fs, ps, or other units.
- You will not need all implemented features!
 - The program has a rather large Header with dozens of Boolean "tokens", they are used to toggle certain functions on and off. Most of them should be explained in the source code.
 - For example: "zwischenplot" generates figures to basically any intermediate steps, this feature is helpful to figure out what went wrong, for example if the time data is properly imported.
- Several Functions of the program are not yet implemented
 - This means, there are already Boolean variables to switch them on, but this will/might not do anything. The problem is that most of these variables are global and might be used in sub- functions. So I was not brave enough to delete them.
 - This is usually explicitly stated as a comment when this variable is defined.

- If you feel the need for such a function, feel free to implement it.
 - Historical naming scheme!
 - The substrate is always called ”‘Teflon’”, do not worry about that. If you use other substrate, just insert correct thickness and refractive index.
 - The same substrate is however sometimes also called sio2.
 - The sample is usually called pila, or amino, or DLN.
 - The first 100-150 lines are the header in which different Boolean variables allow you to choose the proper data processing. This will be needed to adjust to the particular measurement.
 - The program supports positive and negative time direction, make sure you pick the correct one.
 - The program uses Electrical Engineering nomenclature as the equations were inspired by [7]. This also means that phase, time, and the signs of refractive index and conductivity might be different than what you usually use!
- !Test this program with well known materials!

9 Explanation of Really Important/Confusing Variables

The explained Variables are either very important or too complicated to simply explain them in the source code. The majority of variables is commented in the source code.

9.1 `teflon`

This is a Boolean variable. Many functions are similar for substrate and sample calculations. This variable is set prior to calling the function and then passed to the function. This way the function ”‘knows’” if it calculates substrate or sample now and picks the correct equations and experimental results.

9.2 `dteflon`

The thickness of the substrate in micrometers.

9.3 **dpila**

The thickness of the sample in micrometers.

9.4 **averaging_in_time**, used in **Evaluation_TDS_Pellets.m.m**

Our experimental system measures multiple time traces (iterations). These are averaged into a single time trace but we also store the full data-set for each iteration. This variable allows to process each trace individually. This will take longer but might be worth it, if single traces have errors, like jitter, or laser issues during the measurements.

9.5 **frequenzmitteln** used in all programs

This Boolean variable defines whether the frequency domain is averaged or not. The program ALWAYS discretizes the data with a step-size specified in **frequenzschritt** (GHz). However, this can either mean that the datapoints in between are ignored (**frequenzmitteln** = false) or averaged (**frequenzmitteln**=true). For small step sizes both cases are identical, for larger step sizes I suggest averaging to minimize noise and increase the stability of the calculations.

9.6 **ordnerspeichern** used in all, defined in **Assignment_sheet.m** or elsewhere!

This specifies the folder in which the data output is stored. Make sure that this folder exists.

9.7 **load_substrat**, part of **Evaluation_TDS.m** and **Evaluation_TRTS.m**

This if case is crucially important. If true, the program loads previously measured data for the substrate. This is helpful if always the same substrate is used. In this case a large number of substrate measurements exists and can be averaged to be loaded into the program. This minimizes the error/uncertainty caused by the substrate measurement. Commonly we use this function.

9.8 **tapecell** part of **Evaluation_TDS.m** and **tapecell** part of **Evaluation_TRTS.m**

Usually the program assumes that the experiment geometrie is: Air, Substrate, Sample, Air. If **tapecell** is true, the geometry is: Air,Substrate,Sample,Substrate,Air

. So a sandwich geometry is assumed when `tapecell` is true. The sample does not need to be inbetween tape. It can support any symmetric geometry.

9.9 `lorentzianfit` called via optimization toolbox from `Evaluation_TDS_Pellets.m`

When `lorentzfit` is true this function is called. The function is not tested!

9.10 `nafion_literature` used in `Evaluation_TRTS.m`

Boolean variable and an associated if case. When true, this case uses literature values for the host material and previously measured values (or lit. values) of the inclusion material. This case is interesting when the material is hosted in a THz and optical transparent matrix material. See for example Jake paper that is still not published.

The host can be any material, the variable is only called Nafion for historical reasons.

9.11 `zwischenplotten` used in all

Boolean. If true a large number of figures is generated. For example: For each measurement the time trace, the FFT, the phase, the normalized spectrum. This is particular interesting when you are debugging or trying to figure out what went wrong. In general, when everything works fine, I suggest setting it to false.

Keep in mind that matlab can only handle 500-600 figures!

9.12 `thinfilm` in `Evaluation_TRTS.m`

Boolean. This variable toggles the thin film equation on/off. This equation can be used to calculate the conductivity. However, please double check if the thin film approximation is valid for your sample! If the thin film conductivity and the no-model conductivity are different, the no-model wins!

9.13 `bruggeman` used in `Evaluation_TRTS.m`

A Boolean variable. If true effective medium theory (Bruggeman) is used. This is a "all or nothing" variable, meaning currently you can either use effective medium theory everywhere in the program or nowhere.

10 Explanation of Functions

In these programs two different types of functions are used. The commonly used functions in which you hand over some parameters and get some other back, and optimization functions. These functions are either called directly if they perform analytical calculations or they are called from the optimization toolbox (fmincon...). Usually, optimization functions are called `retrieval_XXX`. The one exception is `retrieval_of_n_airref_nooptimization.m` which calculates the refractive index analytically, hence it is not called via `fmincon`, although it does something very similar.

10.1 `faxis.m(time, number of points)`

This program generates the frequency axis (in THz), when a time axis in ps is handed over. The first variable to hand over must be the time vector. The second variable is optional. Make sure that you hand the same number of points to this function as you use in `fft()`. Be advised that the number of points should be a potent of 2^x to allow for faster fft processing. This function and also `fft` can be used for zero-padding, a technique which increases the point density in the fft. Be aware that this increase does not increase the accuracy of the measurement, it only helps to read out the results better and more accurate. The only way to increase the signal is to measure longer, or slower.

10.2 `retrieval_of_d_mark3(thickness,modelparameters)`; called by TDS and `Evaluation_TDS_Pellets.m` via optimization toolbox

This function optimizes/fits the thickness and is called via the optimization toolbox command `fmincon`. To allow for the thickness optimization, a model needs to be assumed. These models are not well established in the THz range and should be chosen with care. In general, the Sellmeier equation has been found to be valid for most non-absorbing materials in the visible range. However, how accurate this model is for transparent THz materials is unclear to me. This `fmincon` command minimizes the output of the function, which is defined as: $\chi = \sum(\text{abs}(S21_{\text{target}} - S21))$; where $S21_{\text{target}}$ is the measured value and $S21$ is calculated in the function and depends on the handed over variables.

Be sure to start this with reasonable values. So measure the sample thickness mechanically and provide reasonable boundaries.

The program supports

- The Debye model with up to three resonances (this model is chosen by the global variable [debyemodel](#))
- The Cauchy model with two “resonances” and a single absorption term ([cauchy](#))
- The Sellmeier equations with two terms and a single absorption term ([sellerie](#))
- The Lorentz Model with a single resonance ([lorentzian](#))

!The Program is only tested with ([datenfensternfuerfit](#) = true). This function truncates the equation prior to the first etalon. Meaning the equation is correct if the substrate/sample is thick enough that the reflection in the material are not measured!

10.3 retrieval_of_n_airref_tapecell, called via fmincon from Evaluation_TDS.m, only called if tapecell=true

It is a TDS retrieval function. It calculates the refractive index of the substrate used in a tapecell. So the geometrie is air/substrate/substrate/air. I am not sure why I wrote a function for that. But well it works I think.

10.4 retrieval_of_n_airref_nooptimization(n,k), used in Evaluation_TDS.m and Evaluation_TDS_Pellets.m

This function is part of a loop over all frequency points. This frequency-loop is defined by [frequenzrange](#) = [fanfang](#):([frequenzschritt](#)/1000):[fende](#); where [fanfang](#),[fende](#), and [frequenzschritt](#) are defined in the header of the main program. [fanfang](#),[fende](#) are in THz, [frequenzschritt](#) in GHz. For each frequency point the refractive index of the sample or substrate is calculated. This calculation is analytically, and can only be used if the imaginary part of the refractive index is small !

!This equation is only valid if the imaginary part of the sample/substrate is small! This assumption is valid for most substrates and for sample material diluted in a host matrix as for example Teflon. A detailed explanation can be found in the SI of [3]. Do not use this for lossy materials! I cannot stress this enough!

10.5 retrieval_of_n_airref_thinlayer_tapecell.m

Called via fmincon, the measurement results and the thickness of the substrate/sample are handed over via global variables. The function calculates the geometry:

Air/Substrate/Sample/Substrate/Air.

The function calculates the refractive index of the sample material without a model! The calculation is performed point-wise. The calculation is numerical so rather instable. The main program uses the time-traces and the thickness to create reasonable starting parameters for the refractive index. The program also checks if there are 2π jumps. These jumps are saved in the variable `mbranchpick`. And plotted in figure(230+lauf), where lauf is a loop variable ensuring that each measurement is in a different figure. This function can handle high-loss materials, i.e. conductive samples!

10.6 retrieval_of_n_airref_thinlayer_pila.m

Called via `fmincon`, the measurement results and the thickness of the sample are handed over via global variables. The assumed geometry is: Air/Substrate/Sample/Air. The function calculates the refractive index of the sample material without a model! The calculation is performed point-wise. The calculation is numerical so rather instable. The main program uses the time-traces and the thickness to create reasonable starting parameters for the refractive index. The program also checks if there are 2π jumps. These jumps are saved in the variable `mbranchpick`. And plotted in figure(230+lauf), where lauf is a loop variable ensuring that each measurement is in a different figure. This function can handle high-loss materials, i.e. conductive samples!

10.7 retrieval_of_n_airref_thinlayer_Mk2_Phasesep, part of Evaluation_TDS.m

This function assumes a low loss material and separates phase and amplitude. The results of this function should be carefully evaluated to ensure that the low-loss assumption is met. The results of this calculation are stored in "npilaa", the results of the previous calculation that supports losses are `nDLN`. Compare both results, physically `nDLN` is the correct one that supports loss, however, it is a purely numerical approach that is prone to failure. `npilaa` on the other hand is analytical, therefore more stable, but has the rather strong assumption that the imaginary part of the refractive index is small. For a detailed discussion see the supplementary information of [3]

10.8 retrieval_of_sigma, part of Evaluation_TDS.m

This function uses the thin-film approximation (see [2, 6, 8, 4]) In general the results should be seen with some caution. Please be sure that the thin-film approximation criteria are met! This is basically the opposite of `retrieval_of_n_airref_thinlayer_Mk2_Phasesep.m` in which low losses are assumed!

10.9 retrieval_of_n_relativtotefflon_nooptimization.m, used in Evaluation_TDS_Pellets.m

This function has as input the calculated or loaded refractive index of the host material and the results from the pellet measurement. The function then calculates the refractive index of the pure material using the Maxwell Garnett equation and the Bruggeman equation. Please note that both eq. have advantages and disadvantages and the user must find out what is the better description of the experimental system! It supports only one host material and one inclusion (sample) material. In general Bruggeman can be used for samples with multiple materials, which is not implemented in this function. If you feel the need for this feature, please feel free to implement it.

10.10 importfiles_script.m and import_for_timfiles used in Evaluation_TRTS.m, respectively in Evaluation_TDS.m, Evaluation_TDS_Pellets.m

These function will not work on your data! They are using the syntax that our LabVIEW program uses to save the THz data. So replace them with a function that can read time and amplitude out of your raw data files. The time must be in picoseconds! The amplitude in any unit.

10.11 new_Eon_durch_Eoff_no_model_v0.m called in Evaluation_TRTS.m via optimization toolbox

This function calculates the photoexcited refractive index of the sample. The input is air-reference measurement, sample photoexcited measurement and the refractive index of the non-photo-excited materials. These are transported to the function via global variable.

10.12 new_Eon_durch_Eoff_DS_v0.m used in Evaluation_TRTS.m via optimization toolbox

Nearly all functions and refractive index calculations used in these three programs DO NOT assume a model. This is one of the few functions that assumes a model! The Drude-Smith model is optimized/fitted on the experimental transmission data. So the refractive index is described via drude-smith equation. The result can be compared to the "no model" results. This is particularly valuable because "no model" results are sensitive to noise. The drude-smith model in the retrieval itself is less sensitive. So if the Durde-Smith model describes your material decently, you

should get an excellent agreement between the refractive indices calculated with different methods.

10.13 DSfit.m called via optimization toolbox in Evaluation_TRTS.m

This function fits the Drude-Smith model to the conductivity, which was calculated without any model. In contrast to the previous one: This fit is performed after calculating the conductivity without model!

10.14 DS_seperate.m via fittoolbox in Evaluation_TRTS.m

Basically the same as DSfit.m. But another fit routine. Should provide similar results as the previous one.

References

- [1] T. L. Cocker, D. Baillie, M. Buruma, L. V. Titova, R. D. Sydora, F. Marsiglio, and F. A. Hegmann. Microscopic origin of the drude-smith model. *Phys. Rev. B*, 96(20):205439, November 2017.
- [2] M. Liang, Z. Wu, L. Chen, L. Song, P. Ajayan, and H. Xin. Terahertz characterization of single-walled carbon nanotube and graphene on-substrate thin films. *IEEE Trans. Microw. Theory Techn.*, 59(10):2719–2725, Oct.
- [3] Jens Neu, Coleen T. Nemes, Kevin P. Regan, Michael R.C. Williams, and Charles A. Schmuttenmaer. Exploring the solid state phase transition in dl-norvaline with terahertz spectroscopy. *Phys. Chem. Chem. Phys.*, 20:276–283, 2018.
- [4] Jens Neu, Kevin Regan, John R. Swierk, and Charles A. Schmuttenmaer. Applicability of the thin-film approximation in terahertz photoconductivity measurements. *Appl. Phys. Lett.*, 113(22):233901, 2018.
- [5] Jens Neu and Charles A. Schmuttenmaer. Tutorial: An introduction to terahertz time domain spectroscopy (thz-tds). *J. Appl. Phys.*, 124(22):231101, 2018.
- [6] Martin C. Nuss, Keith W. Goossen, Paul M. Mankiewich, and Megan L. OMalley. Terahertz surface impedance of thin yba2cu3o7 superconducting films. *Appl. Phys. Lett.*, 58(22):2561–2563, 1991.

- [7] Sophocles J. Orfanidis. *Electromagnetic Waves and Antennas*. Rutgers University, 2006.
- [8] M. Walther, D. G. Cooke, C. Sherstan, M. Hajar, M. R. Freeman, and F. A. Hegmann. Terahertz conductivity of thin gold films at the metal-insulator percolation transition. *Phys. Rev. B*, 76:125408, Sep 2007.