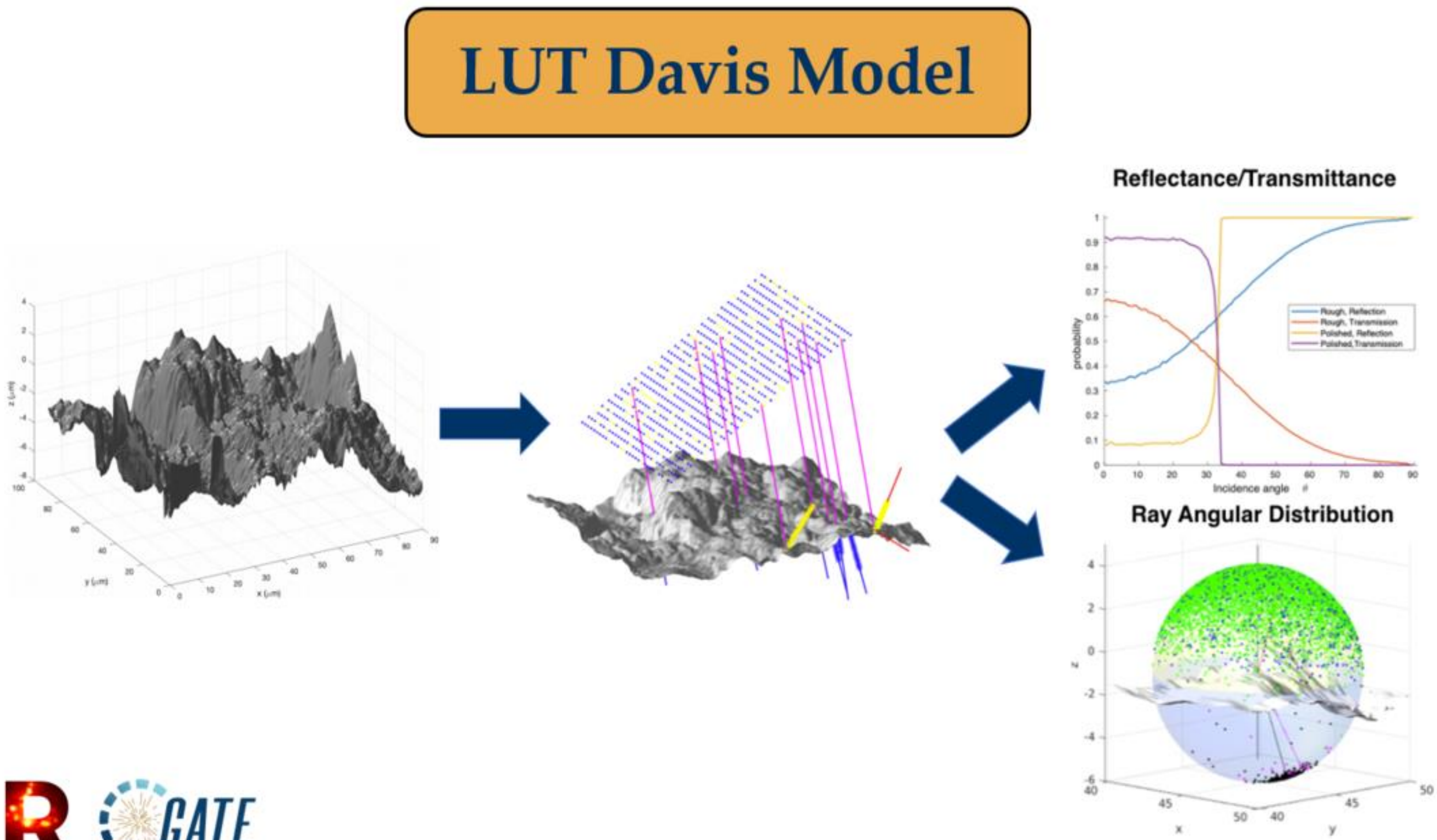


Davis LUT Model Standalone v 1.0 User's Guide

Carlotta Trigila
Eshani Moghe
Emilie Roncali

Table of Contents

Introduction

The Davis Model is a look-up-table (LUT) model of crystal reflectance computed from 3D measurement of crystal surface.

Given a surface sample, the algorithm computes the angular distribution of reflectance and transmittance and the angular distribution of reflected and transmitted rays as a function of incidence angle. The LUT computation includes the reflector, the coupling medium, and the photon tracking between the two interfaces. This algorithm allows an accurate modelling of photon interactions with crystal surfaces with or without a reflector and overcomes some of the limitations of the UNIFIED model and of the experimental approach of Janecek and Moses.

The LUT Davis model has been validated and implemented in GATE v8.0. Subsequent releases include LUTs of a rough and a polished crystal surface without reflector, with a Lambertian reflector (Teflon) and an air- and grease-coupled specular reflector (ESR). While these LUTs allow to perform accurate light transport simulations with good agreement with the experimental data in term of normalized light output, the need for finer detector modeling requires the computation of more customized LUTs.

To give researchers the possibility to simulate more realistic optical data in a specific configuration, we implemented our reflectance model in a standalone graphical interface. Using this GUI, the user will be able to generate GATE LUTs with their customized surface, crystal and coupling or with the ones present within the GUI database. Moreover, a set of reflectors is included in the database together with the possibility of generating LUTs of mixture of coupling media to consider the probability of having non-ideal crystal-reflector-photodetector couplings.

In this User's Guide, we present the standalone graphical interface in all its principal features.

Previous work

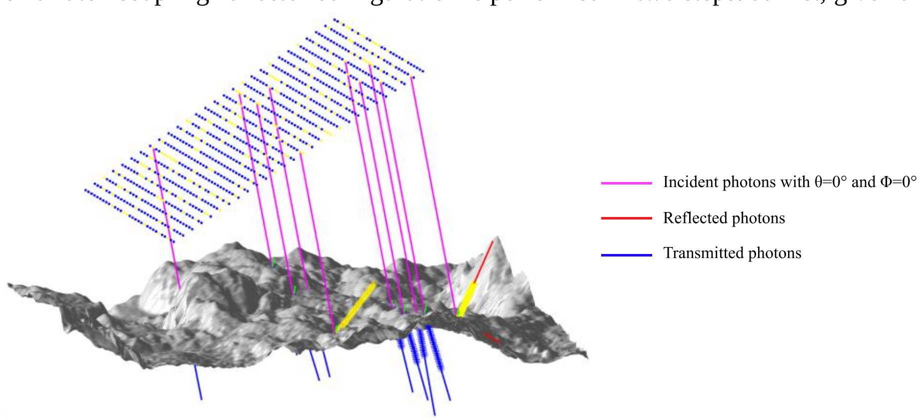
A detailed description of the algorithm at the foundation of the standalone application, experimental validation, and implementation in GATE can be found in the following works:

[1] Roncali E and Cherry S 2013 Simulation of light transport in scintillators based on 3D characterization of crystal surfaces *Phys. Med. Biol.* **58** 2185–98

[2] Roncali E, Stockhoff M and Cherry S R 2017 An integrated model of scintillator-reflector properties for advanced simulations of optical transport *Phys. Med. Biol.* (https://doi.org/10.1088/1361-6560/ aa6ca5)

[3] Stockhoff M, Jan S, Dubois A, Cherry S and Roncali E 2017 Advanced optical simulation of scintillation detectors in gate v8.0: first implementation of a reflectance model based on measured data Phys. Med. Biol. 62 L1–8

Briefly, the LUT Davis algorithm computes the reflection and transmission probabilities and the ray angular distributions as a function of incidence angles (0° to 90°) of crystal surfaces 3D scanned with an atomic force microscopy (AFM), with or without reflector. To simulate the optical photons emitted isotropically after a gamma interaction and to compute the crystal reflectance and the photons direction, the 3D surface is virtually illuminated with a collimated beam of ~2000 photons per angle, each with a specific wavelength randomly extracted by the emission spectrum of the selected crystal. The collimated beam impinges the surface with an *incident polar angle* θ varying between 0° and 90° with an angular sampling of 1°. To ensure sufficient sampling of the surface local slope, the beam is also rotated around the global normal to the surface (azimuthal angle ϕ varying from 0° to 360°) for each polar angle. Each photon is tracked down to the surface using a convergence method. The photon's probability to be reflected or refracted by the surface at a specific incident angle is evaluated using Fresnel equations with respect to the local surface normal vector.
The LUT of a scintillator-coupling-reflector configuration is performed in two steps: at first, given a crystal



(its scanned surface, its emission spectrum, and its index of refraction as a function of the wavelength) and a coupling medium (with its index of refraction), the *scintillator-coupling* reflectance and transmittance LUTs are computed. Then, the transmitted ray information stored in the transmittance LUT are reused to evaluate the influence of a reflector (defined by its reflectance and by the angular distribution of reflection) and generate the *scintillator-coupling-reflector* LUTs (also named reflector LUT). In this way, multiple reflectors LUTs can be computed using the previously computed scintillator-coupling LUT, allowing an important computational time gain when multiple configurations of the same surface are needed.

Standalone download and technical information

Developed with App Designer (MATLAB 2019b), the LUT Davis Model Standalone can be installed and used on Windows, MacOsX, and Linux independently of MATLAB, thanks to the use of MATLAB Runtime [https://www.mathworks.com/products/compiler/matlab-runtime.html].
The installers for the three operating systems can be downloaded from the UC Davis box at:

https://github.com/LUTDavisModel/Standalone-Application-Installers-User-Guide.git

We suggest referring to this guide for a detailed description of its features and for more technical information.

For any problem/suggestion on the app, on this User's Guide or if you need any help, please contact us at:

LUTAPPSUPPORT@ucdavis.edu

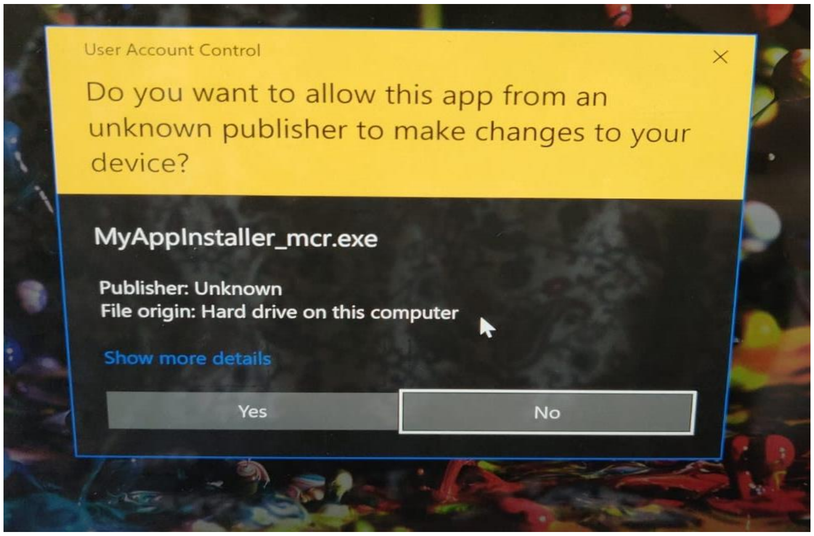
with a detailed description of your problem.

Standalone Installation

Once the installer is downloaded according to your operating system:

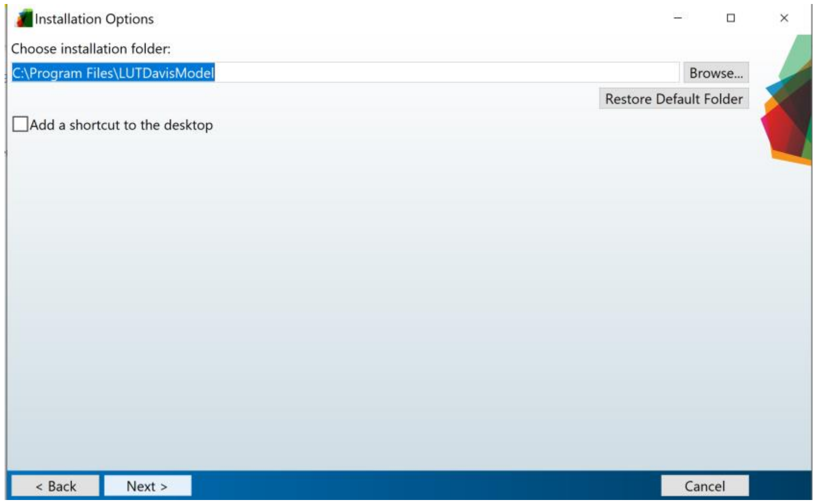
- Double click on the installer.

  NOTE: We are currently working on resolving the "Unknown publisher" issue, as shown below. Until then, please allow this publisher to make changes.

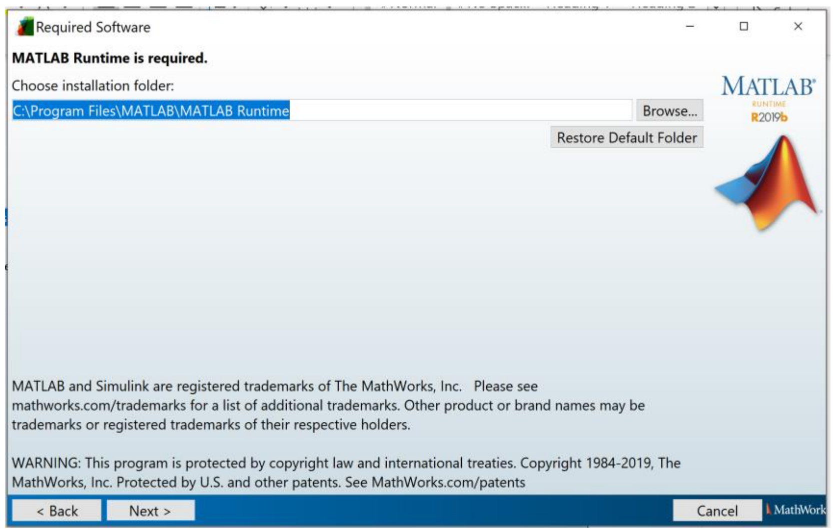


- After a few seconds, the "LUTDavisModel Installer" window will open. Click next.

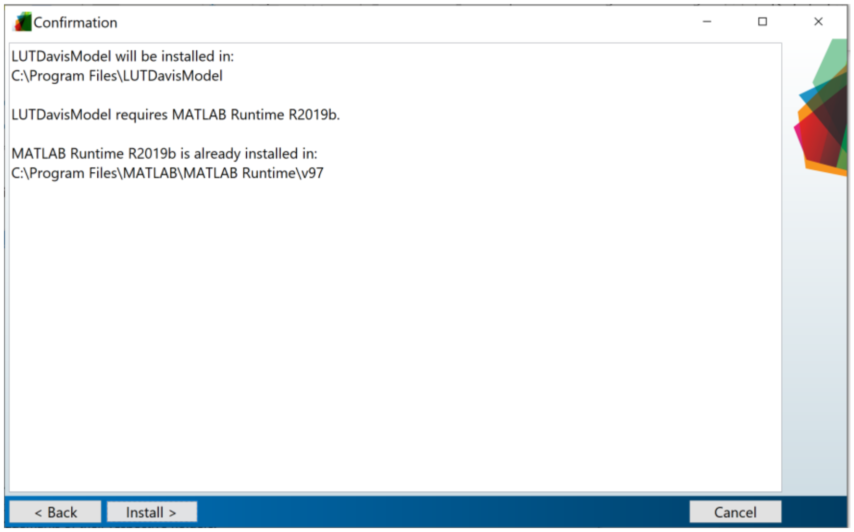
- Choose a location to install your folder. Usually, a default location is selected. Click next.



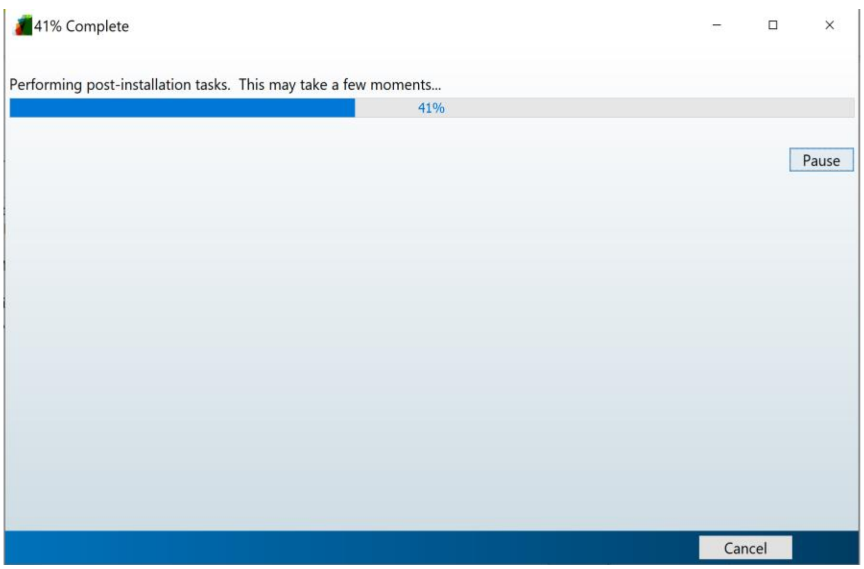
- MATLAB Runtime installation is then required. Choose a location for the Runtime. Usually, a default location is selected. Click next.



NOTE: If Runtime is already installed and you are trying to reinstall it, a window will appear as shown below:



- Finally, the application will start installing. For the application to install, it will take few seconds if Runtime is already on the computer and up to 20-30 minutes if Runtime must be downloaded and installed.



- Once completed, the Installation window will close.

  For Windows/MacOS users: Type "**LUTDavisModel**" on your laptop's search bar and click on the application. Upon clicking, it may take a few moments before the application loads and opens.

For Linux users: Go in the installation folder /application/ and run the LUTDavisModel executable from the terminal (./LUTFavisModel). If the exe does not work, please follow the installation notes about the LD_LIBRARY_PATH.

Standalone Structure

Welcome to the LUT Davis Model Standalone!

The Standalone is composed of five panels:

1) **Generate Scintillator-Coupling LUTs**
Generate the LUT using the surface, scintillator and coupling characteristics.

2) **Generate Reflector LUTs**
Include the reflector in the LUT computation.

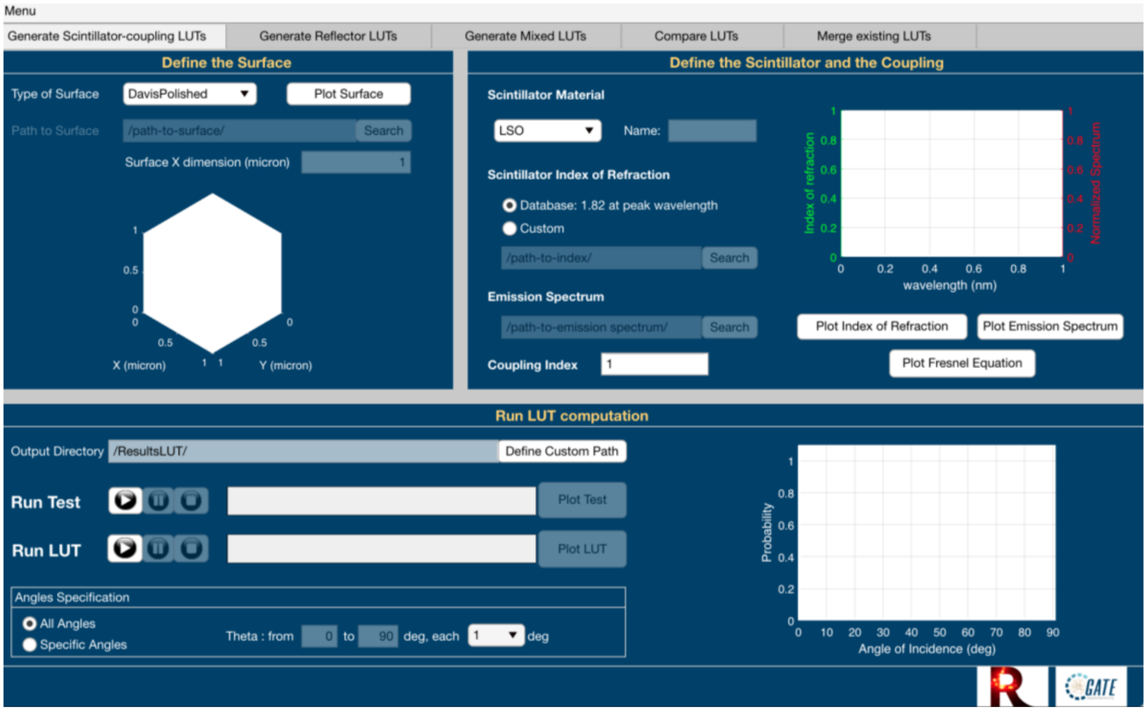
3) **Generate Mixed LUTs**
Generate a mixture of coupling media in the interface between the crystal surface and the reflector or photodetector.

4) **Compare LUTs**
Plot and qualitatively compare already computed LUTs.

5) **Merge existing LUTs**
Merge and create GATE readable LUTs from several non-completed LUT.

Each of panel is further described.

Generate Scintillator-Coupling LUTs

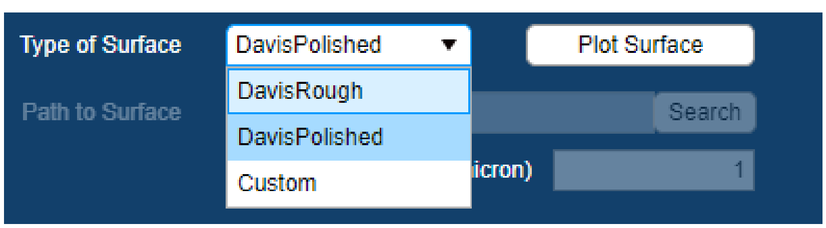


The figure below shows the first panel of the standalone. It allows the scintillator-coupling LUT generation (bottom subpanel) after defining the surface (top-right subpanel), the crystal and the coupling medium (top-left subpanel).

Firstly, the 3D information of the crystal topography must be defined from the "Define the Surface" subpanel.

Define the Surface

From the *"Type of surface"* drop down, one can choose to use a scanned surface in the standalone database or a customized 3D surface (e.g. obtained from a different scanning modality).



If using:

- DavisRough: 90×90 μm^2 rough surface with height varying between ~−3 μm and 3 μm. Find more details about the surface in our previous work [1,2].
- DavisPolished: 45×90 μm^2 polished surface with height varying between ~−0.05 μm and 0.05 μm. It is used as default surface. Find more details about the surface in our previous work [1,2].
- Custom: a 2D matrix containing the height of the surface must be uploaded as a *".txt"* file using the *"Search"* button. Moreover, the surface X dimension in micron must be given as an integer.

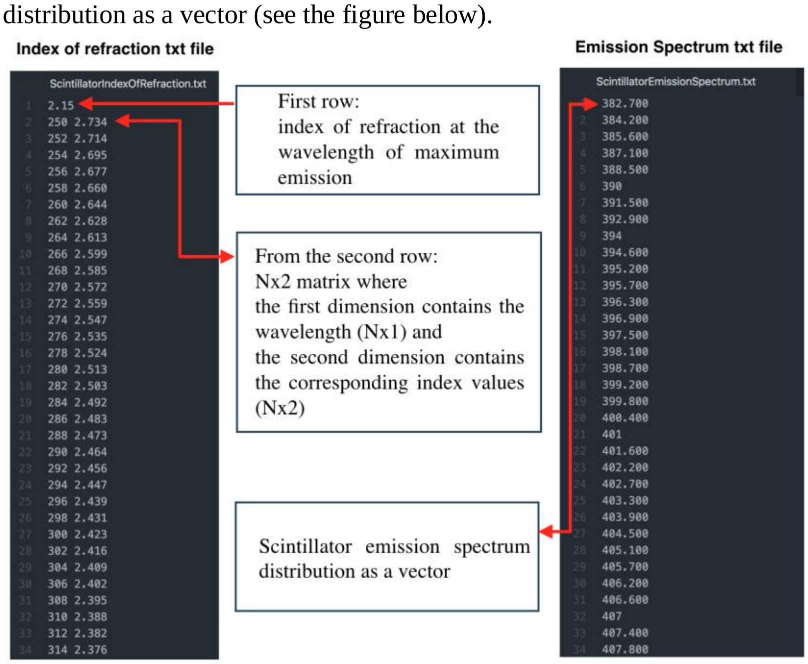
The surface can be plotted using the *"Plot Surface"* button.

Define the Scintillator and the Coupling

The scintillator and the coupling must be characterized using the "**Define the Scintillator and the Coupling**" subpanel. The algorithm needs the scintillator to be defined by means of its name, emission spectrum, and index of refraction as a function of the wavelength. The user can choose from the "*Scintillator material"* drop down:

- A scintillator available in the database:
  The scintillators included are LSO, BGO, $LaBr_3$ YAG and YGG. LSO is the default scintillator. Its index of refraction as a function of the wavelength and its emission spectrum will be taken from the values in the database. Despite choosing a scintillator from the database, a user can choose to customize the Index of refraction. It must be uploaded as a 2xN matrix saved in a *".txt"* file, where the first dimension contains the wavelength and the second one contains the corresponding index values. Constant index of refraction values as a function of the wavelength are also accepted.

- A custom scintillator:
  To do that, different things must be defined:
  - *Name*:
    useful for the results' output folder name
  - *Index of refraction as a function of the wavelength*:
    as a '.txt' file containing (see the figure below):
    1) in the first row, the value of the index of refraction at the wavelength of maximum emission
    2) then, starting from the second row, a Nx2 matrix where the first dimension contains the wavelength (Nx1) and the second dimension contains the corresponding index values (Nx2).
    Constant index of refraction values as a function of the wavelength are also accepted.
  - *Emission spectrum*:
    requires to be uploaded in a *".txt"* file containing the scintillator emission spectrum distribution as a vector (see the figure below).

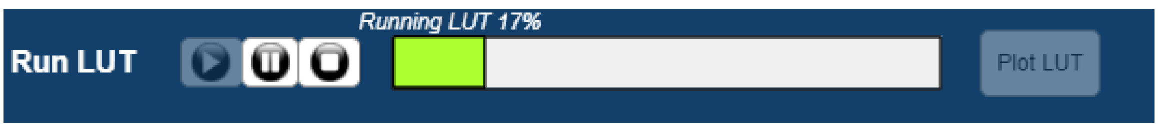


Both the emission spectrum and the index of refraction can be superimposed using their specific plot buttons (“*Plot Index of Refraction*” and “*Plot Emission Spectrum*” buttons).
Regarding the coupling medium, its index of refraction is needed as a value greater than 0. The default value is 1 for air. Other values can be used: e.g. Air = 1; Optical grease = 1.57; Glass = 1.5; Generic glue = 1.5; MeltMountTM = 1.582; TiO2 = 1.61.

Once the indices of refraction of the crystal and the coupling medium are defined, Fresnel equations can be computed and plotted (“*Plot Fresnel equations*” button) to visualize the probability for a photon to be reflected or transmitted by a flat surface.

Run LUT computation

The scintillator-coupling computation can be started via the “*Run LUT computation*” subpanel. After defining the output folder to save the results, the user can run a test (“*Run Test*”) and/or run the complete LUT (“*Run LUT*”) computation using the corresponding play buttons ( ).
In all cases, the computation can be paused ( ) or stopped ( ) when needed.
For all computations, a green loading bar helps the user to visualize the degree of computation of the algorithm.



The “*Run Test*” functionality consists in running the LUT algorithm using low statistics in terms of incident photon angular distribution. This, by virtually illuminating the surface with a beam orientation defined by the polar angle, Theta, varying 0° and 90°, with a sampling of 5°. For each theta, the beam is then characterized by only four azimuthal angle Phi (between 0° and 360°, each 90°). More details about the LUT algorithm can be found in our previous publications [1,2].
Computing test LUTs could provide insight on the reflectance and transmittance dependency with the incident polar angle theta for selected configuration. It can also give an estimate of the total computation time. The reflectance and transmittance results can be plotted using the *Plot Test* button and can be superimposed to the Fresnel equations for evaluation. *It is important to note that this is only a test and that GATE LUTs WILL NOT BE available at the end of the computation test.*

The “*Run LUT*” functionality allows to run the scintillator-coupling LUTs computation after defining the “Angles Specification”.
If choosing:

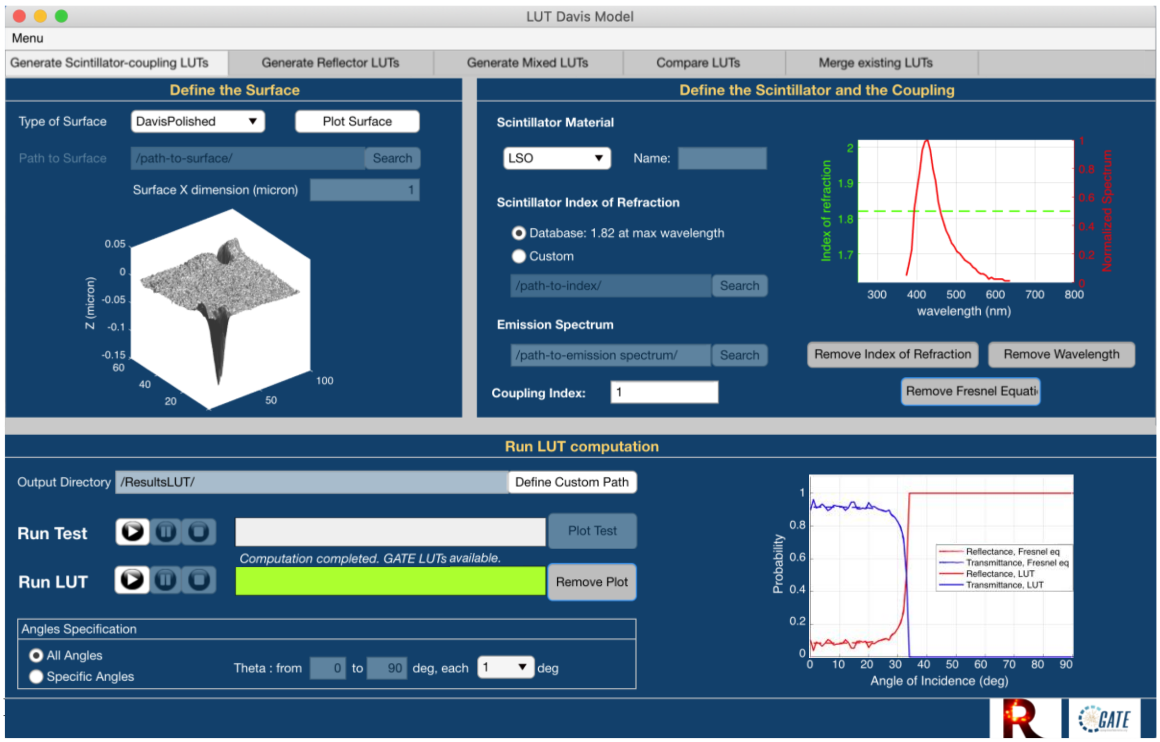
- “***All Angles***”, the complete LUT computation can be performed. It is done with the polar angle Theta going from 0° to 90° and with a sampling of 1°. It is the default angles specification. At the end of this computation, GATE LUTs <u>WILL BE </u>saved in the output folder and can be used in GATE.
- “***Specific Angles***”, only a few select angles will be computed as defined by the user. Values must be set in the corresponding Input boxes. The user is also able to choose the polar angle sampling (refer to the Introduction for more details). At the end of this computation, GATE LUTs <u>WILL NOT BE</u> available. Note: if you want to run only one angle, write it in both Input boxes (e.g. “Theta from 25° to 25°”).

  This computation could be useful if a user wants to have analyze specific incident angles or if for any reason the *"All Angles"* computation crashes. In both cases, users can compute only the remaining polar angles (values from 0° to 90° are needed) and GATE LUTs can be generated within the last panel of the app, “**Merge existing LUTs**”. As will be further explained, it allows to merge the results of multiple incomplete LUT computations after checking their compatibility (same surface, scintillator, coupling) as explained in section **Merge existing LUTs**.

At the end of all computations, the reflectance and transmittance can be plotted using the *"Plot LUT"* button and superimposed with the Fresnel equations.
*NOTE: GATE LUTs will be available only if the complete LUT computation is run with **"All Angles"** specifications or after merging multiple **"Specific Angles"** computations.*

EXAMPLE 1: RUNNING A "Run LUT" SCINTILLATOR-COUPLING LUT with "All Angles"

The figure below shows an example of LUT computation where we choose a configuration composed by the polished surface, LSO as scintillator material, and air as coupling.

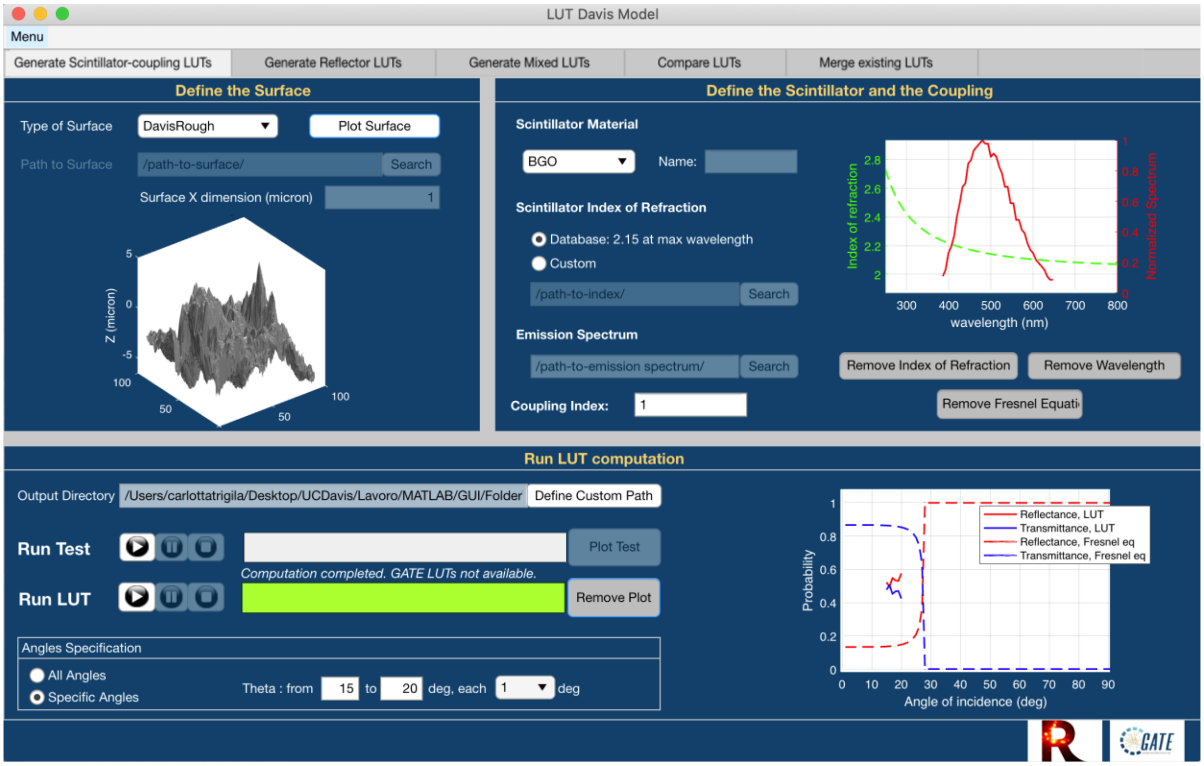


- *Log file*: containing few information about the computation;
- ***Two GATE LUTs, if available.***

NOTE: Once your results are generated, to use them further in the app, please DO NOT CHANGE THE FOLDER NAME.

EXAMPLE 2: RUNNING A "Run LUT" SCINTILLATOR-COUPLING LUT with "Specific Angles"

When running a ***"Specific Angles"*** computation, the user can plot the results but GATE LUTs will not be available. The same happens when performing a "*Run Test*".

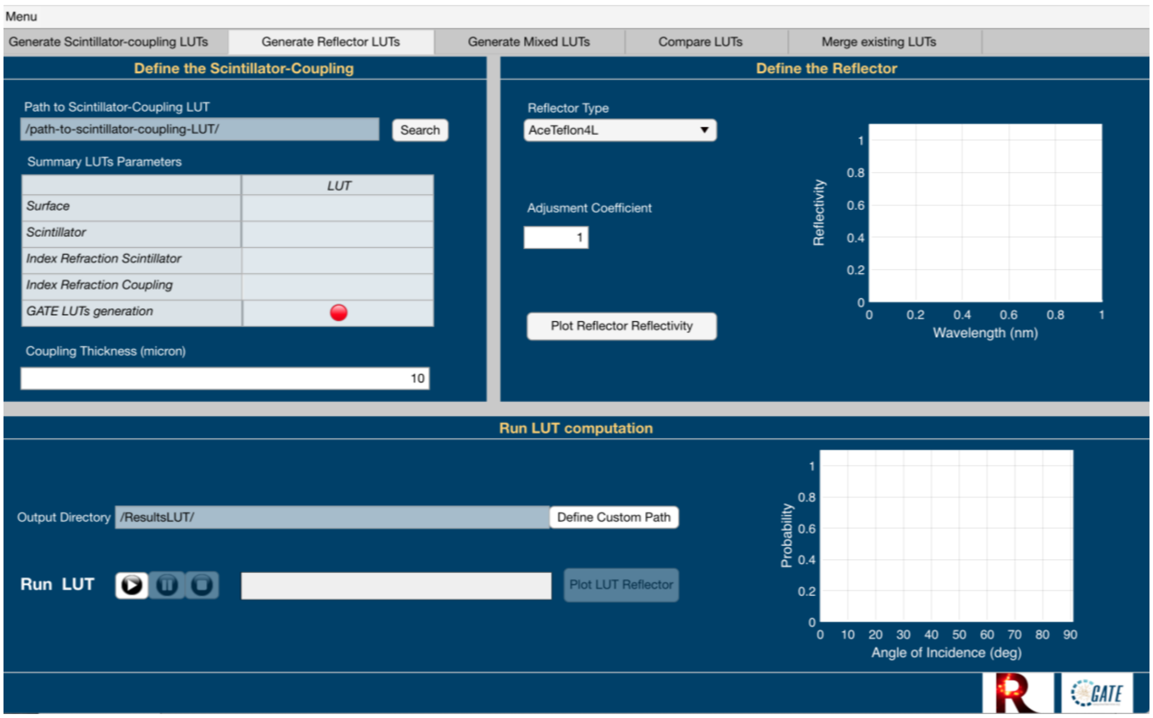


NOTE:

- *GATE LUTs are not available*
- *The results are noise, since low polar angle statistics has been used*
- *The reflectance and transmittance with a rough surface do not follow Fresnel equations, since they describe the probabilities to be reflected and transmitted by a flat surface*

Generate Reflector LUTs

The figure below shows the second panel of the app, useful to compute the GATE LUT with reflector.

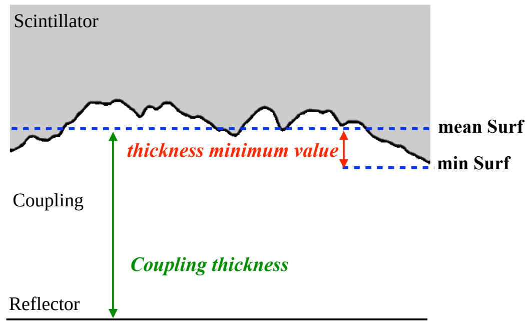


To start the computation, the user must upload a previously computed scintillator-coupling LUT from the subpanel “Define the Scintillator-Coupling” whose characteristics will be summarized in a summary table (figure 2 up-left). A colored light button visually indicates if the reflector GATE LUT will be generated, dependently on the scintillator-coupling LUT characteristics.

If the light is:

- **Green**: You can run the LUT computation to generate and save all the GATE LUTs with reflector. It happens when “*All Angles*” simulations from part1 are used, as shown in the *EXAMPLE 1b*.
- **Yellow**: The Reflector LUT angle computation can be performed, but the GATE LUTs won't be computed since it appears that the Scintillator-Coupling folder used does not contain all the needed input angle (from 0 to 90, with a regular sampling), as shown in the *EXAMPLE 2b*.
  To fix this problem, run the LUT Scintillator-Coupling missing incident angles and use the "Merge existing LUTs" panel to merge your results.
- **Red**: The Reflector LUT cannot be computed.

After choosing the Scintillator-Coupling, the coupling thickness minimum value (distance between the mean height and the minimum height of the surface, see figure below) is automatically displayed in the “Coupling Thickness” edit field. The coupling medium thickness must be set as an integer (in micron) with a value higher than the minimum displayed (See the *EXAMPLE 1b* for 0.1265 micron polished surface and the *EXAMPLE 2b* for 4.23 micron rough surface).



From the "**Define the Reflector**" subpanel, the reflector type to be used must be chosen; its reflectivity can be plotted and manually decreased through a scaling factor. This allows the user to simulate a reflector which efficiency deteriorated for example, as can happen with Teflon impregnated with grease.

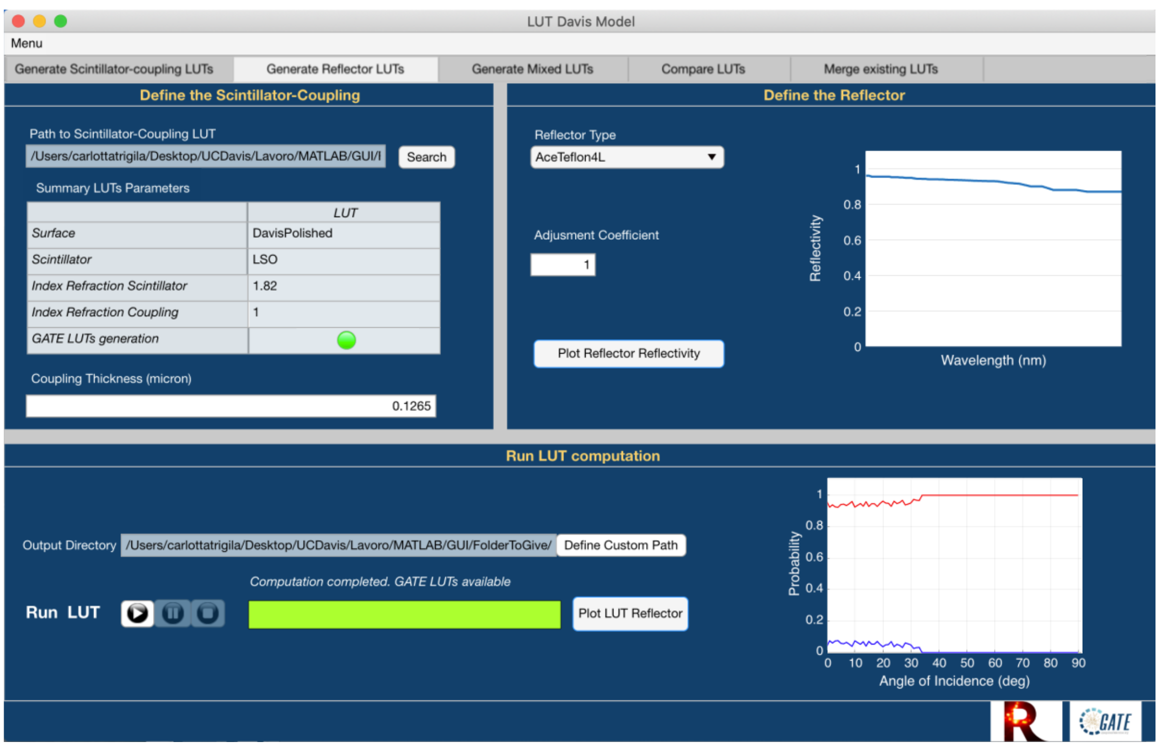
The reflectors available in the standalone database are Teflon, ESR, PTFE2L, PTFE4L, PTFE8L. More details about them can be found in our previous work [2].

After defining the output directory, the LUT can be started, paused, or stopped using the corresponding buttons and the results can be plotted using the plot button.

<u>EXAMPLE 1b: RUNNING A Reflector LUT with a "All Angles" Scintillator-Coupling LUT</u>

Green button:
if using a LUT as the one described in the EXAMPLE 1

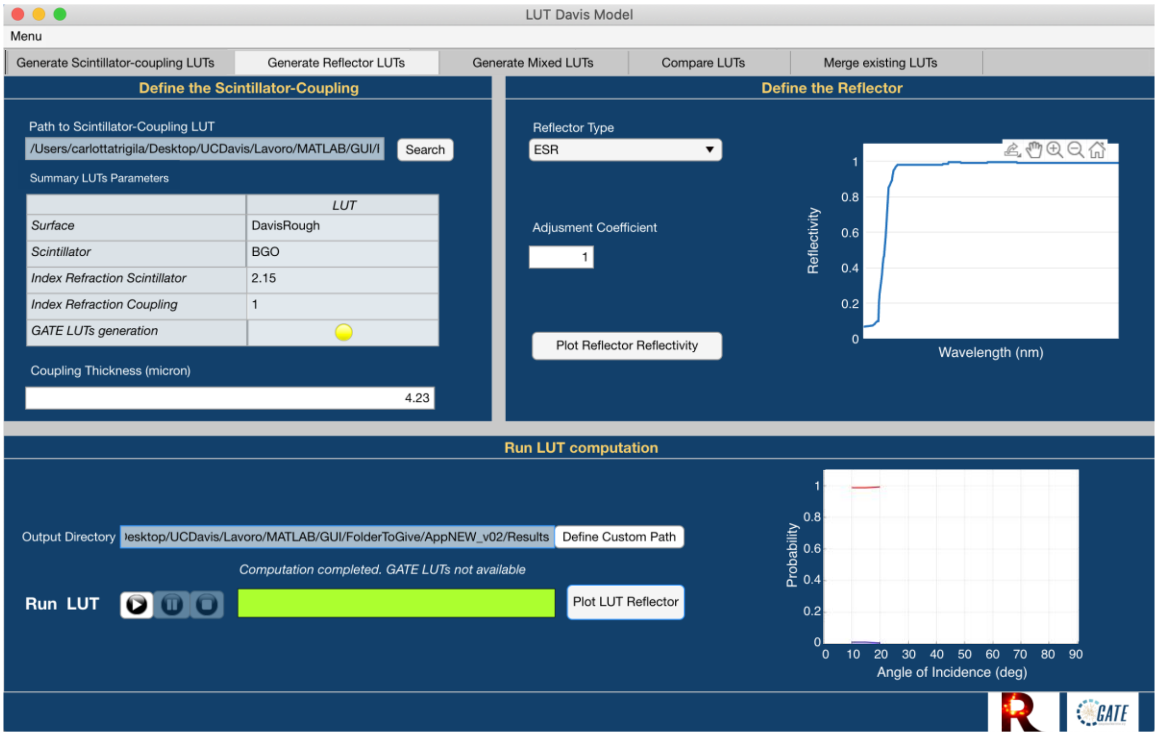
Reflector LUT computed with Teflon as reflector



EXAMPLE 2b: RUNNING A Reflector LUT with a "Specific Angles" Scintillator-Coupling LUT

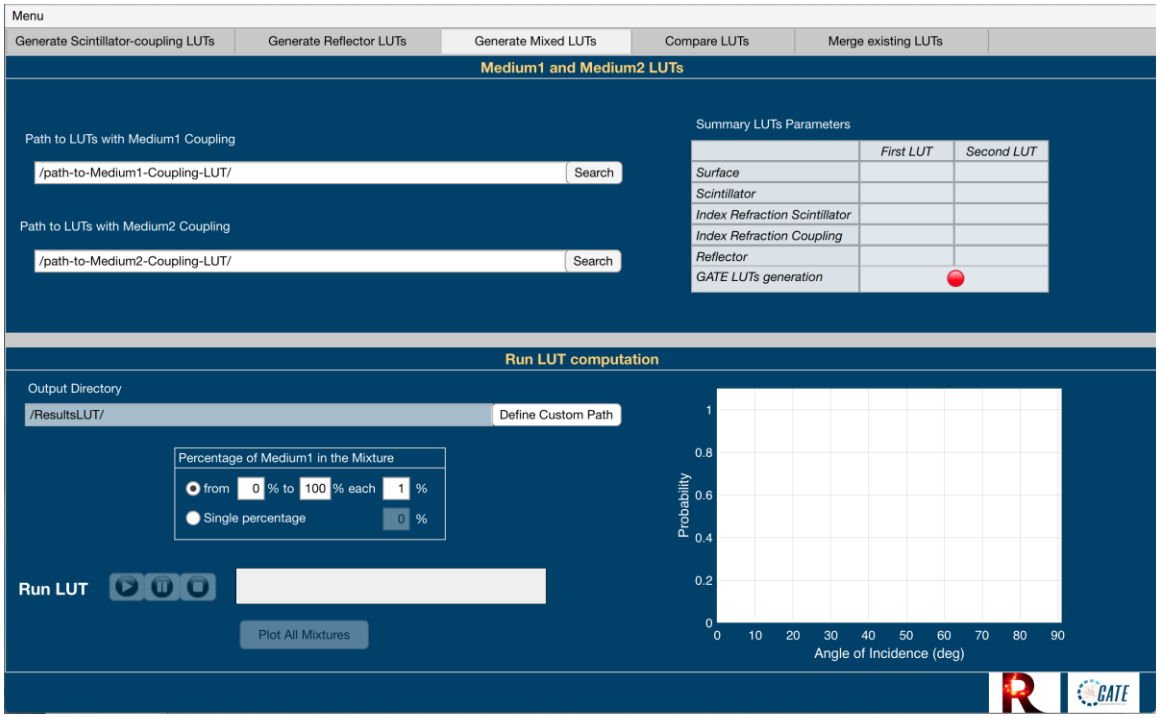
Yellow button:
if using a LUT as the one described in the EXAMPLE 2

Reflector LUT computed with ESR as reflector



Generate Mixed LUTs

The LUT algorithm gives the possibility to generate LUT with a mixture of coupling media (e.g. air and grease). The mixture of two coupling media is performed by using two pre-generated LUTs and fixing the mediums' relative percentages. Using two scintillator-coupling LUTs computed with air and grease, the mixed LUTs allow to simulate grease coupling with some air inhomogeneities. While using two reflector LUTs, one can consider the case in which an amount of optical grease, when pushed away from the photodetector-crystal face, inserts between the crystal and the reflector, thus causing the interface to be far from an ideal crystal-air one. This can be done through the third panel of the app, as shown in the figure below.

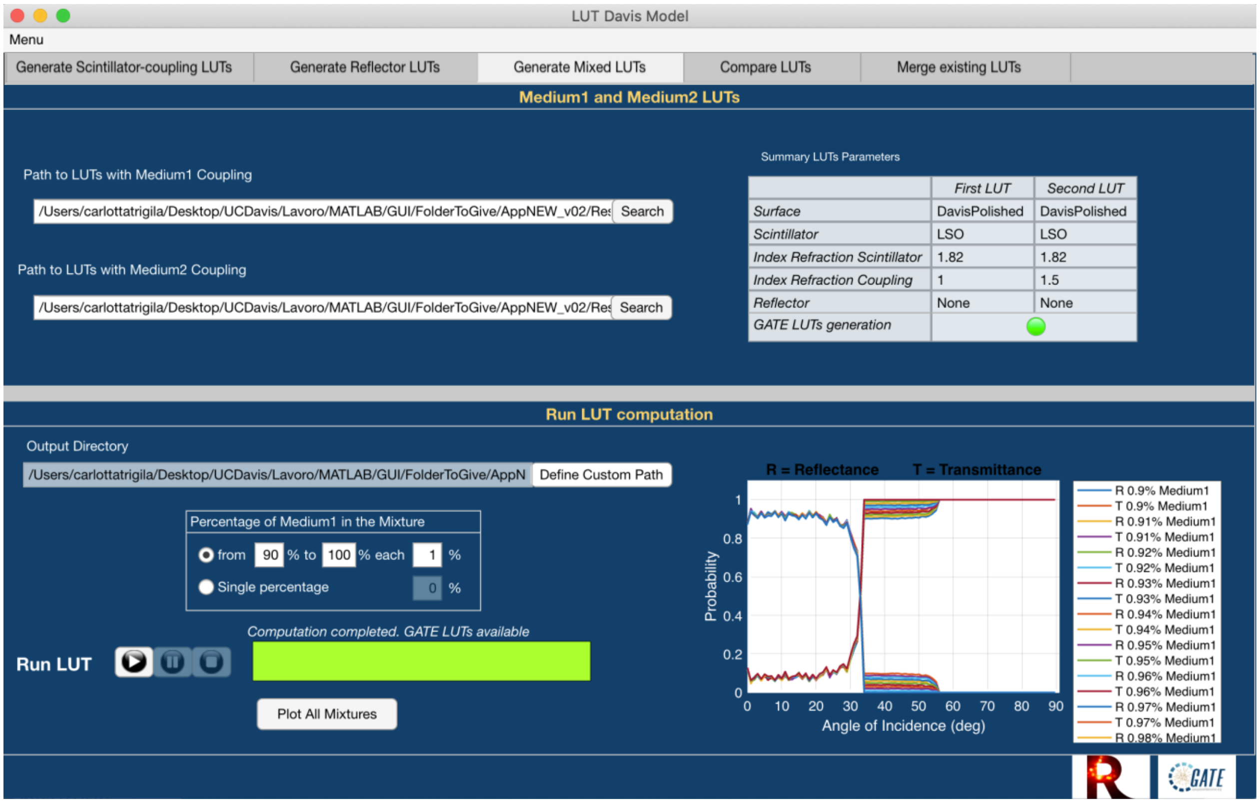


First, two LUTs must be uploaded in the "*Medium1 and Medium2 LUTs*" subpanel. Mixtures can be computed only if the two LUTs are compatible. If creating a mixture from two scintillator-coupling LUTs, they must have the same surface and scintillator but different coupling media whereas if created from two Reflector LUTs, they must have the same surface, scintillator and reflector but different coupling media. A summary of the selected LUT is automatically displayed in the Summary LUTs Parameters table. A light will give the information about the generation of GATE LUTs. If:

- Green: the GATE LUT will be available.
- Yellow: the mixtures are computed and can be plotted, but no GATE LUT will be available.
- Red: the mixed LUTs cannot be computed since the uploaded LUTs are incompatible

After selecting the output folder, the relative percentage of the media in the mixture must be defined as a range of percentage or as a single value (e.g. see the following EXAMPLE 1c).
Then, the computation can be started, paused, or stopped using the corresponding buttons. The results can be visualized using the *"Plot All Mixtures"* button.

EXAMPLE 1c: RUNNING Coupling Mixed LUTs with "All Angles" Scintillator-Coupling LUT made with Air and Grease (from 90% to 100% of air in the mixture)



The mixtures have been created by looking for the folders (previously computed all angles scintillator-coupling LUTs):

1) *201013_095151_LUTScintCoupl_LSO_DavisPolished_1.82-1*
2) *201013_104505_LUTScintCoupl_LSO_DavisPolished_1.82-1.5*

The results are saved in a folder named:

201013_154443_Mixtures_ScintCoupl_Index1-1.5_DavisPolished_LSO

which contains several folders:

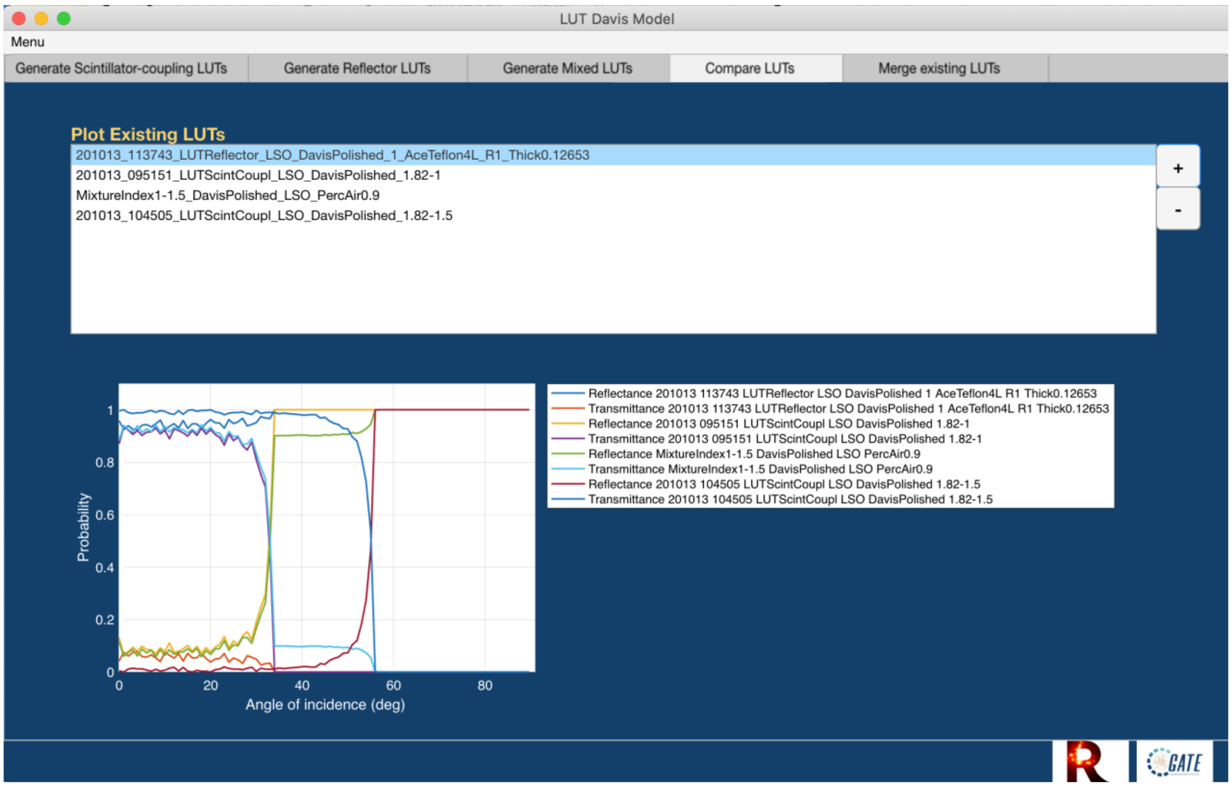
MixtureIndex1-1.5_DavisPolished_LSO_PercAir0.9
…
MixtureIndex1-1.5_DavisPolished_LSO_PercAir0.95
…
MixtureIndex1-1.5_DavisPolished_LSO_PercAir1

Whose contents are the same as described before.
To simulate grease coupling with some air inhomogeneities, users could use the same input LUTs and fix 0% to 10% of air, each 0.5%.

Compare LUTs

The fourth app panel allows to plot the reflectance and transmittance curves of previously computed LUT for visual comparison. Using the buttons can directly add + or remove - a file.

Plotted in the figure below are:
1) Scintillator-coupling LUT with LSO, Polished Surface coupled with Grease
2) Scintillator-coupling LUT with LSO, Polished Surface coupled with Air
3) Reflector LUT with LSO, Polished Surface coupled with Air and Teflon
4) Mixture of the LUTs 1) and 2) with 95% of air



Merge existing LUTs

The last panel gives the possibility to generate GATE LUTs starting from more than two folders of the <u>same exact configuration:</u>

- Same surface
- Same crystal
- Same reflector (if merging Reflector LUTs)

when all polar angles theta is available. It can be done by clicking the button + and uploading different folders.
This computation could be useful if a user needs to have GATE LUTs from multiple incomplete LUT computations, as introduced in **Run LUT computation** subsection.

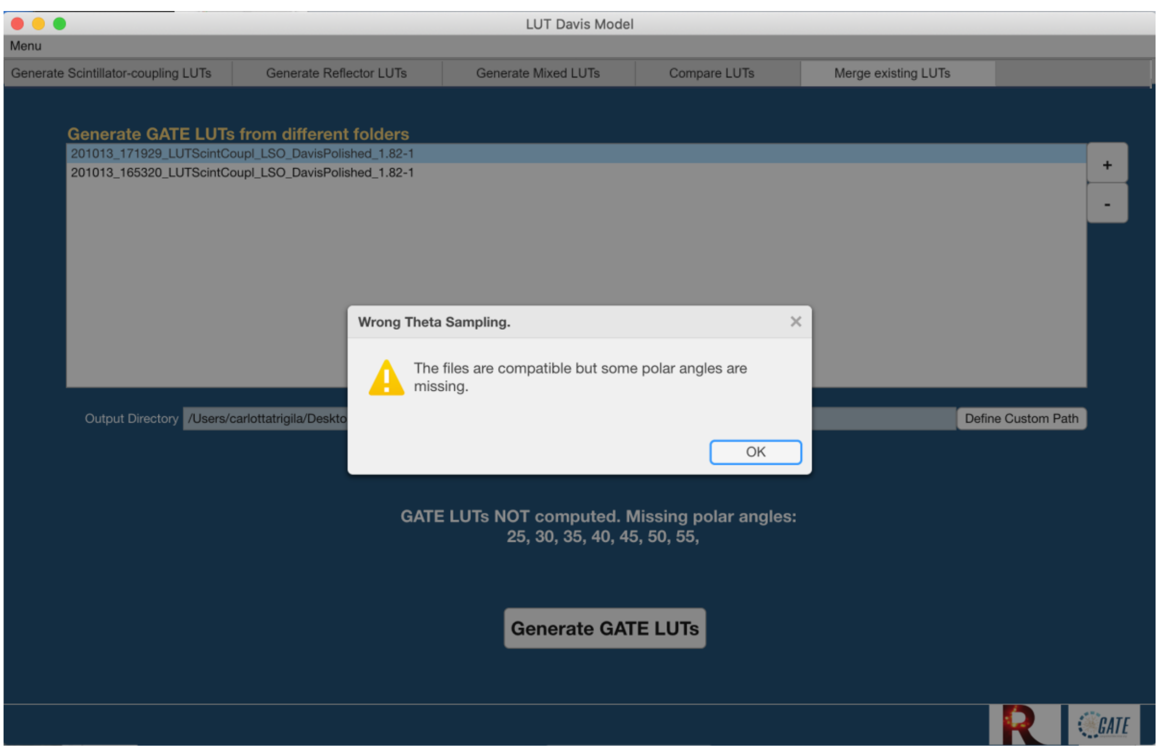
<u>EXAMPLE OF MERGING LUTs TO GENERATE GATE LUTs</u>

Four folders will be used as an example:

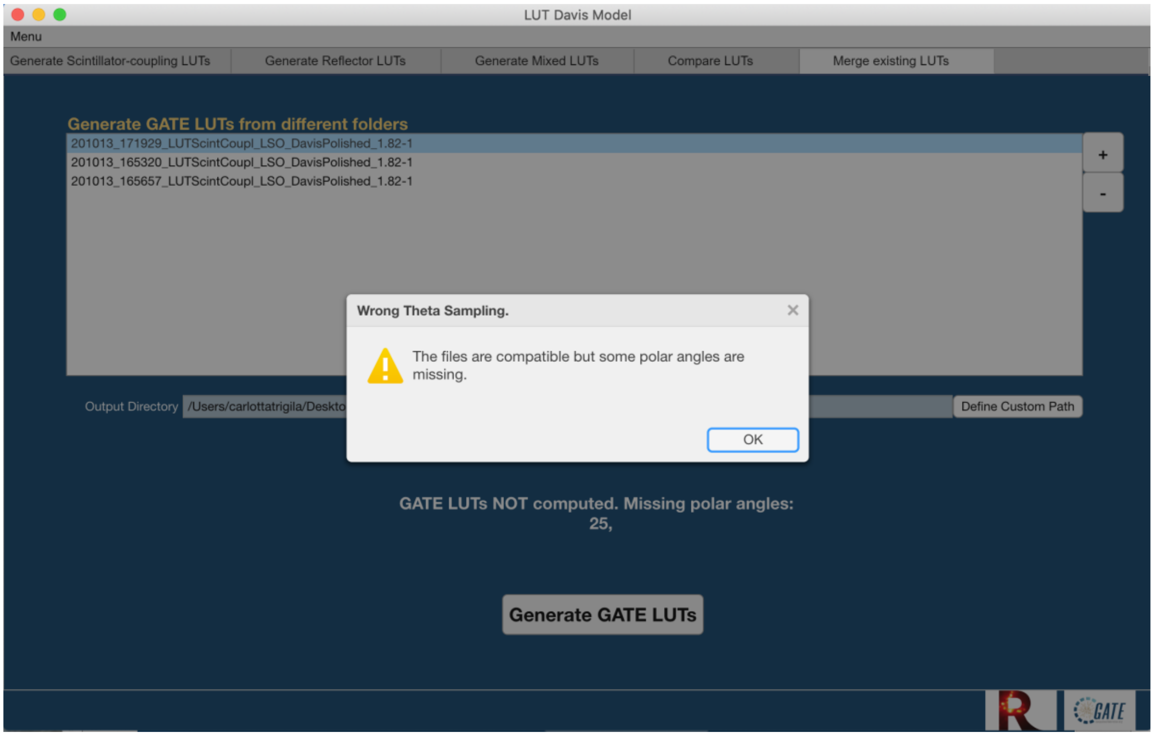
1) *201013_165320_LUTScintCoupl_LSO_DavisPolished_1.82-1*: from 0° to 20 ° each 5°
2) *201013_165657_LUTScintCoupl_LSO_DavisPolished_1.82-1*: from 30° to 60 ° each 5°
3) *201013_171502_LUTScintCoupl_LSO_DavisPolished_1.82-1*: 25°
4) *201013_171929_LUTScintCoupl_LSO_DavisPolished_1.82-1*: from 60° to 90 ° each 5°

We performed the merging by:

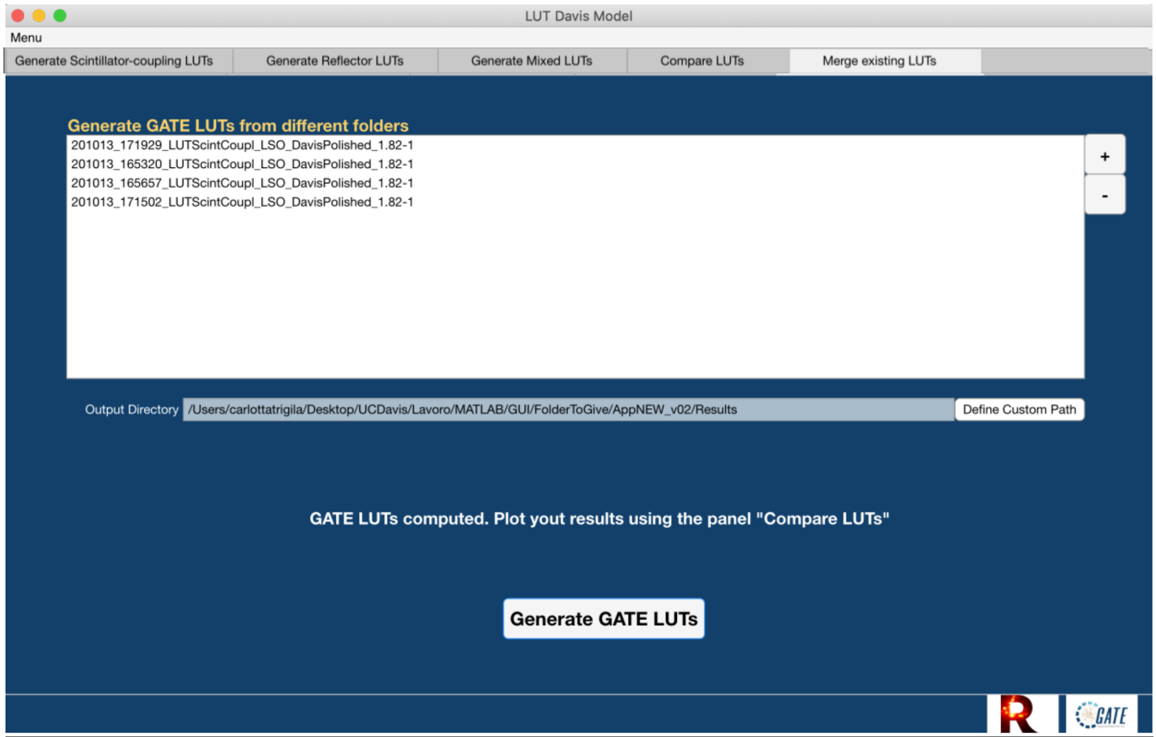
- Adding folder 1 and folder 4 and clicking on “Generate GATE LUTs”:



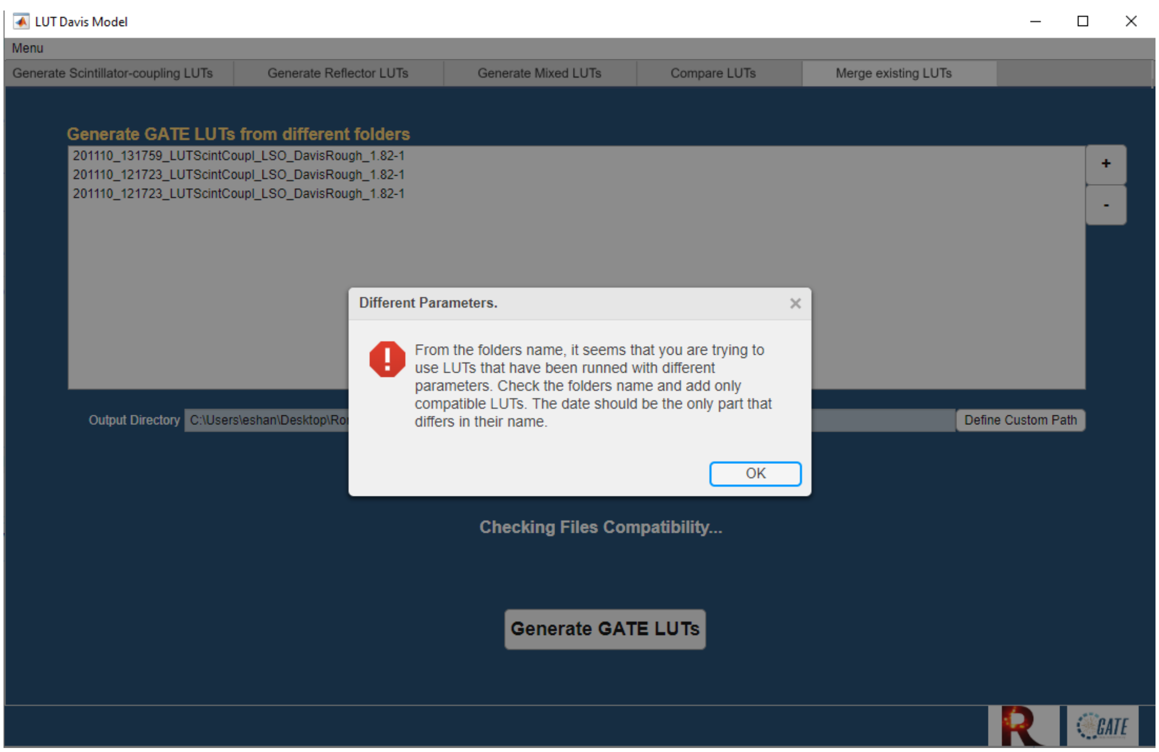
- Adding folder 3:



- Adding folder 4: GATE LUTs are finally generated



<u>NOTE: If the folders are not consistent, an alert will remove the non-compatible folder</u>



<u>NOTE: you cannot generate LUTs if Test LUT or Mixtures LUT are being used.</u>

For any questions, advice, or issues, please contact us at:

LUTAPPSUPPORT@ucdavis.edu