# Authors’ response to Reviewers

**Response from the Authors to** the comments from the **Anonynous Referee #1**:

*Dear anonymous referee*

*First, thank you for your interest in our publication and in the ALG software tool. We also believe that a broader scientific community might be interested in using the provided tool. We kindly appreciate your comments and advices, which have served us to improve the quality of the manuscript.*

*Regarding your comments, we have been taken them carefully into consideration. Specifically, we have reviewed our paper, trying to improve the description of the functionality of the model. Please find in the table below the main changes of our paper, which are also highlighted in red in the updated manuscript to facilitate the revision process.*

*We trust that the comments raised through the revision process have been duly addressed to your satisfaction.*

*Kind regards,*

*The Authors*

***Reviewer #1 comment #1 (1.1): I feel there is a pressing need for a substantial reorganization of Section 3***

We appreciate your comment and the remark on the reorganization of Section 3. After reading your comments, we have made a substantial change on the structure of Section 3. We think that, with these changes, the concepts behind Section 3 have been clarified. Please observe the changes (in red) in our manuscript in Section 3. Particularly, see our replies to your specific comments in the following comments.

***1.2: The authors present two figures to illustrate the programmatic flow of their ALG model. Those models flow diagrams define a natural process for describing the functionality of the model byt the text didn’t clearly follow the diagrams […] As an example, consider Figure 2 ("LUT Configuration"). I presume this is the same as the "LUT config." box referenced in the GUI box of figure 1, but it’s never explicitly stated.***

We acknowledge that the old Figure 1 might be confusing on the logic and purpose of the Section 3.2 (i.e. to describe ALG’s graphical user interface). We have now modified Figure 1 (page 6) to focus only on the graphical user interface. We also modified the caption in Figure 2 caption (also in page 6) to explicitly state this link between “LUT configuration” boxes.

***1.3: Within Figure 2 itself, there are numbered boxes, but those numbers are never explicitly made use of in the prose: either omit the numbers because they don’t add value to your discussion, or mention them in the text because they do.***

We understand the comment from the Referee. However, the numbers are already used explicitly in the prose and each box has a dedicated paragraph. See “Firstly” (line 154 using previous version numbering), “Secondly” (line 162), “Thirdly” (line 169), “Fourthly” (line 172) and “Finally” (for the 5th step; line 175). In order to improve readability and link with the numbers in Figure 2, we have now changed those ordinal numbers (i.e., firstly, secondly…) by expressions such as “In step 1 (Generic configuration), …” (see lines 155, 162, 169, 173 and 177 in the updated manuscript).

We believe that, with this change, it is more explicit and clear the link between the steps in Figure 2 and the paragraphs below.

***1.3: (Section 3- Proposal) Consider turning Figure 1 into a high-level architecture diagram with just a few boxes (the three boxes currently in GUI, the mention of an output config file, and a single box representing "Internal Functions"). I count five potential boxes: mention each of the functions of these five boxes in a logical sequence. For two of those boxes (LUT config and Internal Functions), you can be very brief and conclude with "We will discuss LUT Configuration in greater detail in section 3.x"***

We appreciate the recommendation given by the Referee. We realize that the key problem in understanding and following the logic of ALG is on the position of Figure 1 within Section 3.2 (*ALG graphical interface*). We have decided to update Figure 1 (see page 6 of the updated manuscript), only including the three blocks of the graphical user interface: Software configuration, LUT configuration and Help system. These three blocks are then described in Section 3.2 in lines 150-153 (Software configuration), lines 154-188 (LUT configuration) and lines 190-193 (Help system).

***1.4: (Section 3 – Proposal) Then, take your existing Figure 2 ("LUT Configuration") and give a brief, sequential discussion of the purpose of each of the five boxes. The text is mostly already in the manuscript, but should be organized around the figure itself. The figure has five boxes: I should see five paragraphs - or, more generally, five logical groupings of information.***

Please notice our response to your comment above (1.2). We have explicitly numbered the paragraphs that describe each step within the LUT configuration GUI (Figure 2) (see lines 155, 162, 169, 173 and 177 in the updated manuscript).

***1.5 (Section 3 – Proposal) Finally, the bottom half of your Figure 1 ("ALG - InternalFunctions") can be turned into it’s own Figure 3. It should follow Figure 2 in the sense that you have to configure before you can run. Once again, five boxes means five paragraphs. Some of this text will be new and some can be derived from the existing section 3.***

We have followed the recommendation of the reviewer and included a Figure 4 (page 8) with focus on the ALG’s internal functions for RTM model execution and LUT generation. The new figure includes numbered boxes whose content is further described in the paragraphs below. In order to facilitate readiness, we have added expressions such as “In step 1…”, which serve to identify the paragraph with respect to the boxes in Figures 4 (see lines 193, 205, 210 and 212).

***1.6 (Additional comments) Lines 161-168… This discussion belongs somewhere else. You’re telling the user that once you’ve built the LUT on a discrete grid, you can interpolate to any continuous value within that grid without losing much accuracy. That information should appear in the paper, but Section 3 really focuses how the model gets constructed (not used). I’d propose deferring this paragraph until you’ve completed your description of how the LUT was created in the first place.***

We thanks the suggestion but we don’t fully agree on moving the Lines 161-168 in Section 3.2. In our opinion, these lines are not (in principle) related to the interpolation and how the LUT is used. In fact, we find that it is important to describe here the difference between “discrete” and “continuous” variables. Each of them has a slightly different interface to configure the variable values in the LUT, and so they are important to mention here as they are related with how to use ALG. Notice that the name “discrete” refers to variables that can only take specific (integer) values (e.g. an aerosol model can only be rural, maritime or urban). The name “continuous” indicates that a variable can take any value (real numbers) (e.g., the solar zenith angle can be any value between 0º and 90º).

***1.7 (Additional comments) Lines 198-200 …: this discussion belongs somewhere else. The second set of lines discusses details of how the interpolation occurs. If you aggregate these two blocks, you end up with a useful description of how ALG gets used in real-world situations***

Thanks for the suggestion. However, we consider that the description of the interpolation functions fit well within the description of the LUT node distribution. Moving this text somewhere else would imply having a new section (3.4) with only a few lines explaining the interpolation methods provided to users of ALG.

***1.8 (Additional comments) Lines 207-218 + Lines 230-232: This is an important paragraph that tells the user exactly what you store in your LUT. Give motivation as to why these particular quantities.***

This is indeed a good observation. We have rephrased the introductory sentence (see lines 215-216) to give further motivation on the interest of having the LUT defined as atmospheric transfer functions.

***1.9 (Additional comments) Also, you list several quantities in the LUT as bullet points, then switchgears to discuss a technical issue that differs between codes, and then back to completing LUT quantities, this time as numbered items. Why not complete the bulleted list first and then follow up with the difficult LUT entry.***

We have followed the suggestion by the reviewer and moved the lines 230-232 (in the original paper line numbering) right after the list of atmospheric transfer functions (now in lines 230-233) in order to complete the description of the LUT content.

***1.10 (Additional comments) Finally, this is the spot where the authors have the opportunity to tell the reader the hard work that has clearly been done to make different radiative transfer codes, with different types of outputs, fill the exact same LUT.***

We have added a sentence to stress that the harmonization of RTM outputs is one of the key aspects of the constructions of LUTs in ALG (see lines 234-235).

***1.11 (Additional comments) Lines 219-230: The text is fine, but please lead in with a motivating sentence. Something like "Most values stored in the ALG LUT can be obtained directly from standard RTM outputs. The exception is TOA radiance: obtaining this value differs depending on the RTM being used***

In fact, most RTM provide the TOA radiance spectrum as an output. The lines here were meant to support the idea that the atmospheric transfer functions can be used in forward and backward (atmospheric correction) simulations. Once said that, we agree that these lines should be reorganized in the text. We have decided to introduce equation 1 (now in line 220) before listing the content of the LUT in order to motivate why the atmospheric transfer functions are used to create the LUT. Equation 2 must come after the explanation on the complexity to harmonize the different RTM outputs, since in 6SV the atmospheric transfer functions have slightly a different meaning (due to the uncoupling of absorption and scattering in 6SV) (see lines 238-239 and equation 2 in line 242).

We believe that the reorganization of the text results in a more clear structure to follow the understanding of ALG.

***1.12 (Additional comments) Figure 4: Please define "SI" somewhere (in the figure or in the text)***

Thanks for pointing to this missing acronym. SI (Sensitivity Index) is now included in the text (see line 257).

**Response from the Authors to** the comments from the **Anonynous Referee #2**:

*Dear anonymous referee*

*First, we would like to thank you for your interest in reviewing our publication. That being said, we clearly see the strengths of our work and tend to disagree with some of your remarks. Please allow us to show our arguments and eventual discrepancies to your comments. We hope that, through iteration, we can improve the quality of the paper.*

*Kind regards,*

*The authors*

***2.1 The paper is a technical software […] description and that the results of the model intercomparison […] are not very meaningful since the authors do not explain the reasons for discrepancies between the codes***

We acknowledge that this paper is a technical software description, as it is stated in the goals of our paper (introduction section at lines 48-49: “to describe the ALG tool from a functional and software design perspective”) and in the conclusions section (line 322 “the main design concept and features of ALG have been described”). The intercomparison study is presented in this paper as an exercise (or application example) of how ALG could facilitate the tedious tasks of writing consistent input, running RTMs efficiently and harmonizing RTM outputs. That being said, we consider that the presented global sensitivity analysis results offer an additional comparative analysis of the studied RT models.

***2.2 The compared quantity (global sensitivity analysis) is not well suited for a model intercomparison, because it does not give much insight in reasons for discrepancies***

We consider that global sensitivity analysis can help to understand how the key input variables drive the spectral outputs. The global sensitivity analysis approach is able to quantify the sensitivity to each of the model variables and their interactions, which cannot be done using doing local analysis limited to a few wavelengths and geometric/atmospheric conditions (e.g. [Kotchenova et al. 2008]). Global sensitivity analysis provides therefore a statistical perspective of the entire (global) differences between models. Thus, these analyses provide a useful tool when used in combination to more specific/local analysis like [Kotchenova et al. 2008]. In fact, we also partly carried out the same analysis of Kotchenova et al. in the second part of Section 4 (lines 282 to 296). Our results are compatible with Kotchenova et al and demonstrates the consistency of the simulations. By no means we aim to replace the analysis done in previous validation papers, but to give a complementary analysis from the global sensitivity analysis perspective. Maybe the title of the paper is misleading. Accordingly, we have decided to change it to “Global sensitivity analysis comparison of …”.

***2.3 providing exactly the same input is a difficult task, and obviously, the presented toolbox cannot generate exactly the same input***

We acknowledge the difficulty of this task. As you well indicate, RT models can have different description in their parameterization (e.g. aerosol optical properties, gaseous absorptions, vertical profiles or even definition of viewing/illumination geometric conditions). We are aware of that and that is why, in the last 3 years of ALG development, we have taken a special care of ensuring consistency of model input configuration as much as the implemented RT models allow it. To do that, we have (1) studied carefully the RT user manuals, (2) done our internal analysis of input configuration and comparison of model outputs, and (3) contacted model developers (MODTRAN, libRadtran and 6SV) for user-support. The difficulty (or even impossibility) of ensuring exactly the same model input configuration using ALG is also not different than any other researcher would face when implementing their own RT model input generation, running and comparison scripts.

***2.4 I think that it is important that one works directly with the RT model instead of using a wrapper which hides the specific features of the individual models […] …the danger is that the users do not understand the physics behind radiative transfer, which they learn better when they work directly with the RT models***

We respect your opinion but tend to disagree. It is clear that users who want to have a deeper insight of the model will always work with the RT models directly if needed. Still, they can also benefit from the use of ALG e.g. to alleviate the complex tasks of model configuration and of execution of large datasets. Instead of being applied to ALG, this comment could be also applied to any other software tools designed for a specific RT model such as libRadtran's GUI, Ontar's PcModWin, MOSPAT, AEROgui and others (cited in the paper). However, nobody doubts about the usefulness of these tools to facilitate model configuration and execution. In that sense, ALG is not much different from these reference software tools, but goes beyond in offering a tool that harmonizes the RT model streamlining and parameter definitions (whenever possible). The toolbox does not hides the specific features of the individual models as the RTM input files written by ALG are always stored and available to users. Moreover, the tool is freely available to the community and users can participate as developers, having access to the (Matlab) code repository in GitLab. They can therefore report any eventual software bug or disagreement in input configuration and model outputs. We would like to add that the tool can potentially be used for education purposes (e.g. to master and PhD students in Remote Sensing). Under supervision of an expert/professor, students can learn the basics of atmosphere radiative transfer and get familiar with the inputs and outputs of RT models.

***2.5 I am a little skeptical whether the tool is really useful for the scientific community***

ALG v2.0 (and precursor unpublished versions) has been used in various scientific and industrial projects with related publications (see Section 5). Particularly, ALG is used in several ESA's FLEX mission activities such as end-to-end mission simulation (scene generation) and atmospheric correction. The toolbox is also used in a CNES project, in which atmospheric RT model comparison (MODTRAN6, libRadtran, 6SV, ARTDECO and SOSabs) and large database generation are key tasks. These examples shows the utility of ALG in practical scientific and industrial projects. Moreover, we have informally contacted and demonstrated ALG in front of RT model developers and users, and we have always received positive feedback and demonstration of interest in the tool. In addition, ALG is conceptually based on the ARTMO toolbox (www.artmotoolbox.com), which is a similar tool for the operation of a wide variety of vegetation (leaf-canopy) RT models. ARTMO is actively being used in several European scientific and industrial studies and several publications, conference presentations and tutorials demonstrate the interest of (and usefulness for) the scientific community. Our experience in ARTMO makes us confident that the tool can therefore be of interest for the remote sensing and atmospheric scientific community.

To conclude, we are confident that a broader scientific community might be interested in using the provided tool and that its use should not hamper the user understanding of RT models.

**Response from the Authors to** the comments from the **Executive Editor**:

***3.1 Currently the manuscript fails to comply with very significant parts of the GMD requirements for source code publication. GMD requires the public and persistent archiving of all of the source code on which a manuscript depends.***

We would like to thank you for informing us about the GMD policy on code and data availability. We realized that that indeed our paper was not correctly fulfilling the GMD policy as also indicated in the recent editorial note (https://doi.org/10.5194/gmd-12-2215-2019) and the full details on the requirements for data and code availability (https://www.geoscientific-model-development.net/about/code_and_data_policy.html)

We have carefully reviewed the information in these websites as well as the recommendations you kindly send us by personal e-mail. Accordingly, we have made available the source code of ALG v2.0 (used in this paper) as well as the generated data and scripts using the storage provided by Zenodo under the following link: https://doi.org/10.5281/zenodo.3555575. We have also updated the manuscript on the Code availability section (lines 340-346).

The source code is therefore freely available for the community through the Zenodo repository as well as in our own project website (www.artmotoolbox.com). We think that, with these modifications, the manuscript is now in line with the GMD policy on source code and data availability and thus that our manuscript is compliant for its publication in GMD.

# Comparative analysis of atmospheric radiative transfer models using the Atmospheric Look-up table Generator (ALG) toolbox (version 2.0)

Jorge Vicent[1,2], Jochem Verrelst[2], Neus Sabater[3], Luis Alonso[2], Juan Pablo Rivera-Caicedo[4], Luca Martino[5], Jordi Muñoz-Marí[2], and José Moreno[2]

[1]Magellium, Toulouse, France. Web: http://www.magellium.com.
[2]Image Processing Laboratory, Universitat de València, 46980 Paterna (Valencia), Spain.
[3]Finnish Meteorological Institute, Erik Palménin aukio 1, FI-00560 Helsinki (Finland).
[4]Secretary of Research and Graduate Studies, CONACYT-UAN, 63155 Tepic, Nayarit, Mexico
[5]Departamento de Teoria de la Señal y Comunicaciones, Universidad Rey Juan Carlos, 28943 Fuenlabrada (Madrid), Spain.

**Correspondence:** Jorge Vicent (jorge.vicent@uv.es)

**Abstract.** Atmospheric radiative transfer models (RTMs) are software tools that help researchers in understanding the radiative processes occurring in the Earth's atmosphere. Given their importance in remote sensing applications, the intercomparison of atmospheric RTMs is therefore one of the main tasks to evaluate model performance and identify the characteristics that differ between models. This can be a tedious tasks that requires a good knowledge of the model inputs-outputs and generation of large databases of consistent simulations. With the evolution of these software tools, their increase in complexity bears implications towards their use in practical applications and model intercomparison. Existing RTM-specific graphical user interfaces are not optimized for performing intercomparison studies of a wide variety of atmospheric RTMs. In this paper, we present the *Atmospheric Look-up table Generator* (ALG) version 2.0, a new software tool that facilitates generating large databases for a variety of atmospheric RTMs. ALG facilitates consistent and intuitive user interaction to enable running model executions and storing RTM data for any spectral configuration in the optical domain. We demonstrate the utility of ALG to perform intercomparison studies and global sensitivity analysis of broadly used atmospheric RTMs (6SV, MODTRAN, libRadtran). We expect that providing ALG to the research community will facilitate the usage of atmospheric RTMs to a wide range of applications in Earth Observation.

## 1 Introduction

Atmospheric radiative transfer models (RTMs) have deeply helped in understanding the radiation processes occurring in the Earth's atmosphere (Dubovik and King, 2000; Iacono et al., 2008). RTMs are physically-based computer models that numerically describe the absorption, emission and scattering processes in the ultraviolet to microwave region. Therefore, they are widely used in Earth observation scientific and technological applications, such as: (i) sensor/mission design (Kerekes et al., 1999; Verhoef and Bach, 2012; Verstraete et al., 2015), (ii) atmospheric chemistry (Theys et al., 2007; Dubovik et al., 2011), (iii) meteorology and climatology (Forster et al., 2011), (iv) atmospheric correction (Richter, 1996; Cooley et al., 2002; North et al., 2008), and (v) atmospheric physics (Stamnes et al., 1988). Over time and through continuous improvements, these models have increased in realism from simple semi-parametric equations (e.g., Seidel et al., 2010) towards advanced RTMs that allow for explicit 3D representations of complex interactions in the atmosphere. Some examples include: 6SV (Vermote et al., 1997), libRadtran (Mayer and Kylling, 2005; Emde et al., 2016), MODTRAN (Berk et al., 2006, 2014), MOMO (Fell and Fischer, 2001) and RTTOV (Saunders et al., 2018).

Given the importance of atmospheric RTMs for remote sensing applications, their intercomparison is one of the main tasks in order to determine their performance and identify the characteristics that differ between models (Kotchenova et al., 2008; Seidel et al., 2010; Proud et al., 2010; Callieco and Dell'Acqua, 2011). The process of comparing various atmospheric RTMs can be a tedious task that requires a good knowledge of the model inputs-outputs and the generation of large database of consistent simulations. Indeed, the evolution of RTMs towards more advanced models has resulted in an increase in complexity and intepretability of these models, which bears implications towards practical implementation of intercomparison studies. To overcome this limitation, Graphical User Interfaces (GUI) have been developed to facilitate RTM use and execution. A few examples of these GUIs can be found for 6SV (Matarrese et al., 2015; Wilson, 2013), MODTRAN (Schläpfer, 2016; Berk et al., 2017) or libRadtran (Mayer and Kylling, 2017). These well-documented tools allow complete access to all functionalities and configuration parameters of the models they were designed for, including user-support and continuous updates. However, each of these GUIs are customized for their specific RTM; and none can be used to define and run simulations for multiple RTMs in a consistent manner. In addition, they are not designed to easily precompute large databases, which are important due to the high computational burden for performing statistical analysis (Verrelst et al., 2016) or running these models in a pixel-per-pixel basis (Gastellu-Etchegorry et al., 2003; Guanter et al., 2009). Altogether, these GUIs are not fully offering practical solutions for the implementation of atmospheric RTMs in Earth Observation applications and, in particular, for model intercomparison. Users of atmospheric RTMs are therefore obliged to develop their own specific scripts to create datasets, which are typically: (1) limited to a handful input variables and (2) hardly extensible to other RTMs.

In an attempt to facilitate the consistent simulation of databases for a wide range of atmospheric RTMs, we developed the *Atmospheric Look-up table Generator* (ALG). ALG is a Matlab-compiled software package that allows generating Look-Up Tables (LUT) based on a suite of atmospheric RTMs. Namely, a LUT consists of a collection of input atmospheric conditions and corresponding generated RTM spectral outputs (see Section 3.3 for further details). ALG provides consistent and intuitive user interaction for defining model configuration, running and storing RTM data for any spectral configuration in the optical domain. The main objectives of this paper are therefore: (1) to describe the ALG tool from a functional and sofware design perspective, thereby giving the reader an overview of the implemented features and generated LUT data; and (2) to perform a comparison study between the models implemented in ALG: MODTRAN (v5 and v6), 6SV v2.1 and libRadtran v2.0.2.

The remainder of this work is structured as follows: Section 2 gives an overview of the currently implemented atmospheric RTMs and associated graphical interfaces. Section 3 describes the ALG software design and its main features. Section 4 provides a comparative analysis of the implemented atmospheric RTMs. Section 5 summarizes a few applications as examples of the usage of ALG. Finally, Section 6 concludes with an outlook of on-going and planned functionalities to be implemented in future versions of ALG.

## 2 Overview of existing atmospheric RTMs and associated GUIs

In this section we briefly describe the key features of the atmospheric RTMs compatible with ALG version 2.0 and their associated user interfaces.

### 2.1 MODTRAN

Developed by Spectral Science Inc. (www.modtran.spectral.com), MODTRAN (Berk et al., 2006, 2014) is one of the most widely used RTMs by scientists and commercial organizations with multiple applications in Earth Observation. MODTRAN solves the atmospheric radiative transfer (RT) equation with accurate simulations of the coupled absorption and scattering effects (Stamnes et al., 1988; Goody et al., 1989). These simulations are calculated in a stratified spherically-symmetric atmosphere consituted of vertical profiles of molecules and suspended particles (aerosols). Accordingly, MODTRAN combines the effects of molecular and particulate absorption/emission and scattering, surface reflections and emission, solar/lunar illumination, and spherical refraction. The calculated spectral outputs include direct and diffuse transmittance, top-of-atmosphere (TOA) radiance fluxes, solar/lunar irradiance, horizontal fluxes, cooling rates, etc. These outputs extend from the ultraviolet to the long wavelength infrared spectral range (0.2-200 µm) and are provided at a resolution up to $0.1\ \mathrm{cm}^{-1}$ (0.01-0.1 $\mathrm{nm}$ in the VIS-SWIR spectral range) for the narrow band simulations, or even higher using the line-by-line capabilities (Berk and Hawes, 2017). With over 30 years of heritage, MODTRAN has been extensively validated, and it continues to be maintained and upgraded (Berk et al., 2015).

Several GUIs are made available by commercial companies such as Spectral Sciences Inc., Ontar's PcModWin (www.ontar.com) and ReSe's MODO (Schläpfer, 2016). All these tools consist of a graphical front-end that wraps around MODTRAN, facilitating user interaction and model configuration from scratch and thus leveraging the use of MODTRAN. These GUIs give access to a wide range of input parameters such as definition of vertical profiles, geometric conditions and spectral configuration. Users can therefore format the input files to run MODTRAN and display the resulting simulations through interactive plotting panels. Some of these tools also allow running several simulations through the GUI, manually varying the configuration of every new simulation or through parameter series of one parameter at a time. Despite these capabilities, none of these tools are customized to generate large LUTs of MODTRAN simulations.

### 2.2 6SV

6S was developed in the 90s (Vermote et al., 1997). Since then, it has been applied to process broadband resolution instruments (e.g., El Hajj et al., 2008; Matthews et al., 2010; Liu et al., 2012). 6S solves the RT equation based on the method of successive orders of scattering (Lenoble, 1985), with a decoupling of the absorption and scattering effects by molecules and particulates. These numerical approximations are performed in a stratified plane-parallel atmosphere consituted of vertical profiles of molecules and aerosols. The calculated spectral outputs include direct and diffuse transmittance in the sun-to-target and target-to-sensor directions, spherical albedo, atmospheric path radiance and TOA radiance fluxes. These outputs extend from the spectral range between 0.3-4 µm at a resolution of 2.5 nm. The latest updates of the code account for polarization in the atmosphere (Kotchenova et al., 2006; Kotchenova and Vermote, 2007).

The only GUI dedicated to 6S known by the authors is its official website (6s.ltdri.org). Under its section *Run 6SV*, users can define the input configuration and run the code to retrieve the 6S input and output files directly from the web browser. Accordingly, the generation of multi-parametric LUTs is not feasible with this online GUI. In order to overcome this limitation, Py6S was developed (Wilson, 2013). Py6S is a Python-based Application Programming Interface that provides (1) user-friendly model setting, (2) run and plotting capabilities, and (3) ability to import external data (e.g. atmospheric profiles). As such, Py6S can be integrated in any Python code facilitating the direct usage of 6S in data processing algorithms or for LUT generation.

### 2.3 LibRadtran

The libRadtran software package is a collection of algorithms for atmospheric radiative transfer calculations (www.libradtran.org) and thus used for various applications in the field of remote sensing, atmospheric physics and climatology. LibRadtran implements different solvers of the RT equation that allow computing (polarized) radiances, irradiances and actinic fluxes in the solar and thermal spectral regions (Mayer and Kylling, 2005; Emde et al., 2016). The most recent updates include new features such as: (1) simulation of the Raman scattering, (2) new parameterization of molecular absorption and aerosol optical properties, or (3) Monte-Carlo solver of the RT equation. The flexible design of libRadtran makes it a powerful and versatile tool for research tasks. Furthermore, libRadtran includes a Python-based graphical user interface that simplifies the usage of the model. The GUI has similar functionalities as those previously discussed for MODTRAN. As such, it is not possible to run a large set of simulations and compile LUTs for later use in data processing applications.

### 2.4 OPAC

Despite of not being an atmospheric RTM per se, the OPAC package is a widely used software tool that provides aerosol optical properties in the 0.25 and 40 µm spectral range (Hess et al., 1998; Koepke et al., 2015). OPAC calculates the extinction, scattering, and absorption coefficients, the single scattering albedo, the asymmetry parameter, and the phase function. These optical properties are calculated for a set of 10 pre-defined aerosol models and user-defined mixtures, thus expanding the existing capabilities of atmospheric RTMs.

Similar to the previously defined RTMs, OPAC operates on the basis of input/output files. In order to facilitate its use, several GUIs have been developed that are compatible with OPAC. MOSPMAP is a toolbox, linked with libRadtran, for the optical modelling of complex aerosols, including pre-calculated optical properties of single aerosol particles as those in the OPAC package (Gasteiger and Wiegner, 2018). A user-friendly web interface was developed for MOPSMAP facilitating online calculations. The AEROgui tool (Pedrós et al., 2014) is a similar GUI package that can be used to obtain the optical properties of a mixture of aerosol particles. Accordingly, AEROgui expands the current capabilities of OPAC by also providing a user interface to facilitate user definition of new aerosol mixtures. However, in most cases, there is no direct and straighforward way to include the OPAC output data into atmospheric RTM simulations.

## 3 The Atmospheric Look-up table Generator (ALG) tool

In this section we start identifying the key software functionalities (Sect. 3.1). Then we introduce the ALG graphical interface and how it is used to configure a new LUT (Sect. 3.2). Finally, we describe how ALG automatically generates a LUT and its content (Section 3.3).

### 3.1 Key software functionalities

The primary goal of ALG is to provide a scientific software package that fills the gaps observed in the previously analyzed tools. In particular: (1) Each existing GUI is compatible with only one specific atmospheric RTM (e.g., PcModWin for MODTRAN) and cannot be used to configure and run simulations for other RTMs. (2) These tools are not intended to run a large number of simulations and thus creating LUTs. (3) The inputs and outputs of each atmospheric RTM are generally not consistent between each other, adding an extra layer of complexity when using or comparing various models.

Accordingly, ALG offers the following key functionalities:

1. ALG functions as a wrapper for running atmospheric RTMs, providing a graphical tool in which users can select the input configuration (i.e., atmospheric, geometric and spectral). In this way, ALG keeps the same functionality as all the previously described tools.

2. ALG facilitates the integration of additional atmospheric RTMs. In its current version 2.0, ALG is compatible with MODTRAN5 (Berk et al., 2006), MODTRAN6 (Berk et al., 2014), 6SV version 2.1 (Vermote et al., 1997) and libRadtran version 2.0.2 (Emde et al., 2016).

3. The GUI is common to all the implemented models so that it facilitates the configuration of a wide variety of atmospheric RTM.

4. LUT design with ALG is a flexible process in which users can select a RTM, its input atmospheric variables and values.

5. ALG automatically processes and harmonizes all the RTM input and output data into the final LUT file. With this functionality, ALG facilitates the intercomparability between atmospheric RTMs and the possibility to alternate between models in a data processing algorithm (e.g., for atmospheric correction).

6. ALG provides a help system and a set of tutorials to facilitate users with the installation and operation of the software.

### 3.2 ALG graphical interface

ALG's graphical interface provides users the tools to configure the software, to run the RTM simulations and to construct the final LUT. It is divided into three main elements (see Fig. 1): the *Software configuration*, the *LUT configuration* and the *Help system* GUIs.

The *Software configuration* GUI facilitates the user to edit software aspects of ALG such as: (1) the path to the executable RTM files, (2) the default folder to store the output data, and (3) the default CPU-cores used to run a RTM. In addition, users can

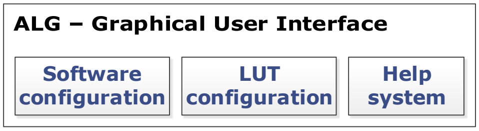


**Figure 1.** Main blocks of ALG's graphical user interface.

add new RTM input variables and edit their default values. This software configuration GUI also permits editing and storing the spectral configuration of existing and user-defined remote sensing instruments to ease the generation of sensor-specific LUTs.

In its core interface (*LUT configuration*) users can select the RTM input variables and values used to run the simulations and to store the spectral outputs into the LUT. This GUI is based on the commonalities found in Section 2, with extended functionalities that allow running a large set of simulations. The LUT configuration GUI is divided in five main subsequent steps as shown in Fig. 2 and further described in the paragraphs below.

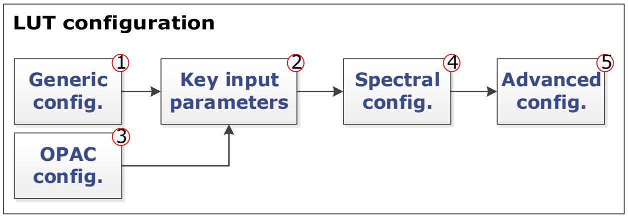
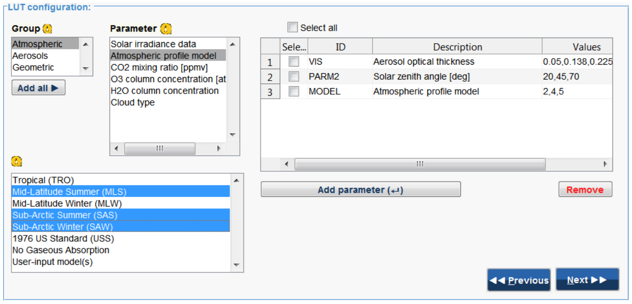


**Figure 2.** LUT configuration steps accesible through ALG's graphical interface (see Fig. 1).

In step 1 (*Generic configuration*), it is selected the atmospheric RTM used to run the simulation and the sampling method to distribute the LUT nodes (i.e., collection of points of input atmospheric and geometric variables). Several methods are implemented to distribute LUT nodes: (a) systematic gridded combinations of all input values, typically applied in atmospheric correction algorithms (e.g., Guanter et al., 2009); (b) scattered near-random and homogeneous sampling of the input variable space based on Latin Hypercube Sampling (McKay et al., 1979), Sobol distribution (Bratley and Fox, 1988) and Halton distribution (Kocis and Whiten, 1988); or (c) automatic gradient-based distribution (Vicent et al., 2018). Parallel instances of the selected atmospheric RTM are invoked in order to speed up the process of generating large LUTs (Brazile et al., 2008).

In step 2 (*Key input parameters*), ALG allows users to introduce selected atmospheric and geometric variables and their values (see Fig. 3). In ALG, input variables are divided into two types: discrete and continuous. Discrete variables are those that can only take on a certain number of values. Typical examples of discrete variables are the atmospheric profile, the aerosol model or the extraterrestrial solar irradiance. Continuous variables can have any value within an allowed range. Typical examples of continuous variables are the columnar water vapor (CWV), the aerosol optical thickness (AOT) or the solar/viewing zenith angle (SZA, VZA). For continuous variables, their values are varying between an user-input minium/maximum range and, in case of gridded sampling, distributed according to a selected distribution (linear, logarithmic, exponential or cosine).



|  | Sele... | ID | Description | Values |
|---|---|---|---|---|
| 1 | □ | VIS | Aerosol optical thickness | 0.05,0.138,0.225 |
| 2 | □ | PARM2 | Solar zenith angle [deg] | 20,45,70 |
| 3 | □ | MODEL | Atmospheric profile model | 2,4,5 |

**Figure 3.** The *key input parameters* of the LUT configuration GUI (see step 2 in Fig. 2 allows users to introduce input model variables and their values.

In step 3 (*OPAC configuration*), ALG implements a back-end interface with OPAC (v3.1) database, expanding the pre-defined aerosol models with a comprehensive database of aerosol optical properties (i.e., extinction, absorption and phase function). For OPAC aerosol models, users can create new aerosol mixtures described by their particle number density from a set of basic components.

In step 4 (*Spectral configuration*), the spectral configuration of the RTM simulations is introduced. Users can set the desired spectral range and resolution, eventually at non-contiguous spectral intervals, saving computation time and disk storage of unwanted wavelengths. A set of predefined spectral configuration of common satellite instruments or user-defined sensors can be loaded.

Finally, in step 5 (*Advanced configuration*), the user has access to advanced RTM configuration parameters (e.g., selection of radiative transfer solver, printed output files). These parameters largely depend on the selected RTM.

All these LUT configuration parameters are stored in a `.xml` file that is later used by ALG's internal functions (see Section 3.3) to automatically run the RTM simulations and construct the final LUT. This configuration file can be loaded by ALG, allowing users to edit and re-run previous simulations e.g., by adding new atmospheric variables, changing the spectral configuration or modifying advanced settings. It worth also noticing that the LUT configuration interface is common for all implemented RTMs and the software harmonizes the naming and definition of atmospheric and geometric parameters to all models.

Additionally, ALG's GUI provides access to the help system with information about: (1) how to install the software and third-party RTMs, (2) how to generate a new LUT, (3) sample cases (tutorials) with practical applications of the use of the software, and (4) implemented RTMs and input variables. The ALG help system is based on Matlab® help browser developed by ©The MathWorks, Inc.

### 3.3 ALG internal functions. Look-up table generation

After setting the LUT configuration (see Section 3.2), ALG implements a set of back-end functionalities to automatically generate the output atmospheric LUT based on the input configuration (see Fig. 4).

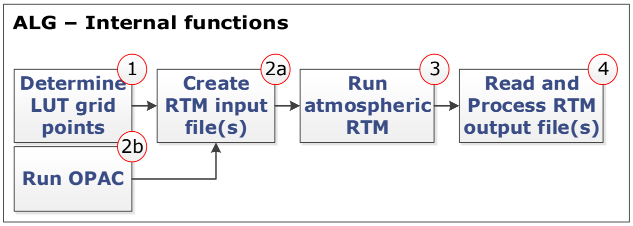


**Figure 4.** ALG's internal functions for RTM model execution and LUT generation process.

In step 1, ALG starts determining the LUT nodes of input atmospheric and geometric variables according to the selected option. Three LUT node distribution methods are implemented in ALG. The first method corresponds to a systematic (gridded) combination of all input variables and their values. Assuming $D$ selected input variables, each of them with $p_i$ values ($i$=1 to $D$), the output LUT will contain $N = \prod_{i=1}^{D} p_i$ nodes. The second method correspond to a pseudo-random distribution of nodes homogeneously covering the $D$-dimensional input space with a user-defined $N$ scattered nodes. The final method is based on an automatic node distribution algorithm, GALGA, that minimizes the error in the linear interpolation of simulated TOA radiance below a user-defined error threshold value. This gradient-based node distribution has shown to reduce interpolation errors by at least 10% and LUT size (and thus computation time) by at least 25% (Vicent et al., 2018). ALG includes a multi-dimensional interpolation function that works both with gridded and scattered data. The implemented LUT interpolation methods involve: (1) nearest neighbour, (2) piece-wise linear (Abramowitz and Stegun, 1964), (3) piece-wise cubic splines (Bartels and Barsky, 1998), (4) inverse distance weighting (Shepard, 1968), and (5) $D$-dimensional triangulation (Delaunay, 1934; Barber et al., 1996).

In step 2a, the LUT generation process continues by converting the determined combinations of atmospheric/geometric variables and user-input spectral configuration into a set of RTM input files required to build the atmospheric LUT. In this step, ALG detects if the user has selected any default or user-defined OPAC aerosol model. If so, ALG automatically runs OPAC and saves the output aerosol optical properties for a later use (see step 2b). Following the approach proposed in (Huang et al., 2016), the values of these aerosol properties, spectral configuration and additional atmospheric input variables (i.e., the LUT nodes) are written in $P$ subsets of RTM input files. In step 3 (*Run atmospheric RTM*), parallel instances of the selected RTM are then run in batch mode based on these input files.

In step 4, and once all the RTM simulations are correctly executed, ALG will finalise the LUT generation process by reading ,processing and storing the RTM output data files in the final LUT file. One of the key aspects of ALG is that it harmonizes the variety of RTM spectral outputs into a common and consistent definition of the stored LUT data. For this, ALG uses the so-called *atmospheric transfer functions*, typically used in remote sensing applications. These atmospheric transfer functions permit uncoupling the radiative transfer effects of the between the surface and atmosphere and thus are particularly useful in atmospheric correction and forward modeling (Vermote et al., 1997; Matthew et al., 2000; Guanter et al., 2009; Verhoef and Bach, 2012). In the case of a Lambertian and homogeneous surface with reflectance $\boldsymbol{\rho}$, a TOA radiance spectrum ($\boldsymbol{L}_{toa}$) can be calculated through Eq. (1):

$$\boldsymbol{L}_{toa} = \boldsymbol{L}_0 + \frac{(\boldsymbol{E}_{dir}\mu_{il} + \boldsymbol{E}_{dif})(\boldsymbol{T}_{dir} + \boldsymbol{T}_{dif})\boldsymbol{\rho}}{\pi(1 - \boldsymbol{S}\boldsymbol{\rho})} \tag{1}$$

where $\mu_{il}$ is the cosine of the SZA. The LUTs generated by ALG contain the atmospheric transfer functions used in Eq. (1) and which are described below:

- The spectrum of intrinsically reflected radiance by the Earth's atmosphere ($\boldsymbol{L}_0$ in $\mathrm{mW} \cdot \mathrm{m}^{-2}\mathrm{sr}^{-1}\mathrm{nm}^{-1}$), also called atmospheric path radiance.
- The downwelling solar irradiance spectrum at surface level, splitted by its direct ($\boldsymbol{E}_{dir}$) and diffuse ($\boldsymbol{E}_{dif}$) fluxes, both in $\mathrm{mW} \cdot \mathrm{m}^{-2}\mathrm{nm}^{-1}$.
- The atmospheric reflectance spectrum for the photons backscattered to the surface ($\boldsymbol{S}$), also known as spherical albedo.
- The upwelling direct and diffuse target-to-sensor transmittance spectra ($\boldsymbol{T}_{dir}$ and $\boldsymbol{T}_{dif}$).

In addition to these atmospheric transfer functions, the generated LUT file also includes:

- The extraterrestrial solar irradiance spectrum at 1AU Earth-to-Sun distance, $\boldsymbol{I}_0$ in $\mathrm{mW} \cdot \mathrm{m}^{-2}\mathrm{nm}^{-1}$.
- The wavelength vector at which these spectral magnitudes are calculated.
- The name and values of the input atmospheric and geometric variables for each LUT node.
- The values of the remaining (constant) parameters.

An important part of the complexity of ALG lies in being able to harmonize the different radiative transfer codes, with different types of outputs, to fill the exact same LUT. For MODTRAN simulations, these spectrally-dependent atmospheric transfer functions are automatically calculated by applying the interrogation technique presented in Guanter et al. (2009) and Verhoef and Bach (2012). In the case of libRadtran simulations, four runs are needed to compute these transfer functions (Debaecker et al., 2016). Similarly, 6SV directly provides the atmospheric transfer functions, however, with a slightly different definition due to the uncoupling of scattering and gas transmittance. The following transfer functions are used for 6SV: path radiance, at-surface total solar irradiance due to scattering ($\boldsymbol{E}_{tot}$ in $\mathrm{mW} \cdot \mathrm{m}^{-2}\mathrm{nm}^{-1}$), total gas transmittance ($\boldsymbol{T}_{gas}$), total upwelling transmittance due to scattering ($\boldsymbol{T}_{tot}$) and spherical albedo ($\boldsymbol{S}$). In this case, $\boldsymbol{L}_{toa}$ is calculated through Eq. (2):

$$\boldsymbol{L}_{toa} = \boldsymbol{L}_0 + \frac{\boldsymbol{T}_{gas}\boldsymbol{E}_{tot}\boldsymbol{T}_{tot}\boldsymbol{\rho}}{\pi(1 - \boldsymbol{S}\boldsymbol{\rho})} \tag{2}$$

**Table 1.** Key input atmospheric variables used in MODTRAN5, libRadtran and 6SV to perform the GSA.

| **Variable name** | **Min-Max** |
|---|---|
| Elevation (h): | 0 - 3 $\mathrm{km}$ |
| Aerosol optical thickness (AOT): | 0.05 - 1 |
| Ångström exponent ($\alpha$): | 0.1 - 1.5 |
| Aymmetry parameter (G): | 0.6 - 1 |
| Single Scatteting Albedo (SSA): | 0.75 - 1 |
| Water vapor (CWV): | 1 - 4 $\mathrm{g} \cdot \mathrm{cm}^2$ |
| Ozone (O3): | 0.25 - 0.45 $\mathrm{atm} - \mathrm{cm}$ |

## 4 Model intercomparison

As a first step for the RTM intercomparison study, we carried a global sensitivity analysis (GSA) of atmospheric RTM simulations. GSA allows to identify the key input variables driving the spectral output and variables of lesser influence. By identifying variables of lesser influence, models and generated LUTs can be greatly simplified, which facilitates applications such as inversion of biophysical parameters and atmospheric correction. In short, sensitivity analysis algorithms determine the effect of changing the value of one or more input variables, and observing the effect that this has on the RTM output. GSA, where the role of all input variables and their interactions are analyzed, have been successfully applied in vegetation and atmospheric RTMs (Verrelst et al., 2016; Vicent et al., 2017).

Here, we used ALG to generate a set simulations in order to analyse the relative impact of key atmospheric variables into TOA radiance. Three LUTs of MODTRAN5, libRadtran and 6SV simulations were generated. They consist of 2000 samples distributed with a Latin Hypercube sampling and covering the entire 400-2500 $\mathrm{nm}$ spectral range at 15 $\mathrm{cm}^{-1}$ (0.24-9 $\mathrm{nm}$) for MODTRAN and libRadtran and 2.5 $\mathrm{nm}$ for 6SV. These LUTs vary the atmospheric conditions as summarized in Table 1, with remaining geometric fixed to SZA=30°, VZA=0° and a relative azimuth angle (RAA) of 0°. For all the 2000 combinations, the atmospheric transfer functions were coupled with a typical vegetation spectrum simulated with PROSAIL model (Jacquemoud et al., 2009) based on Eq. (2) using ARTMO's TOC2TOA toolbox (Verrelst et al., 2019).

Figure 5 shows, through the total sensitivity index (SI), the relative importance of each input variable at TOA radiance for typical vegetation spectrum.

In general, all three RTMs show similar GSA results, indicating that they simulate similarly the processes of absorption and scattering. In these models, the driving variables are those related with the aerosol particles (AOT, $\alpha$, G and SSA), which cause the scattering and thus path radiance and diffuse transmittance along the entire spectral range. The Ångström exponent increases its relative importance as wavelength increases from 550 $\mathrm{nm}$, which is the anchor wavelength at which the AOT is defined. The surface altitude has its major influence ($\sim 80\%$) at the bottom of the $O_2$-A absorption ($\sim$760 $\mathrm{nm}$) since the aborption is mostly driven by the surface pressure. As expected, the importance of CWV is localized at the specific wavelengths

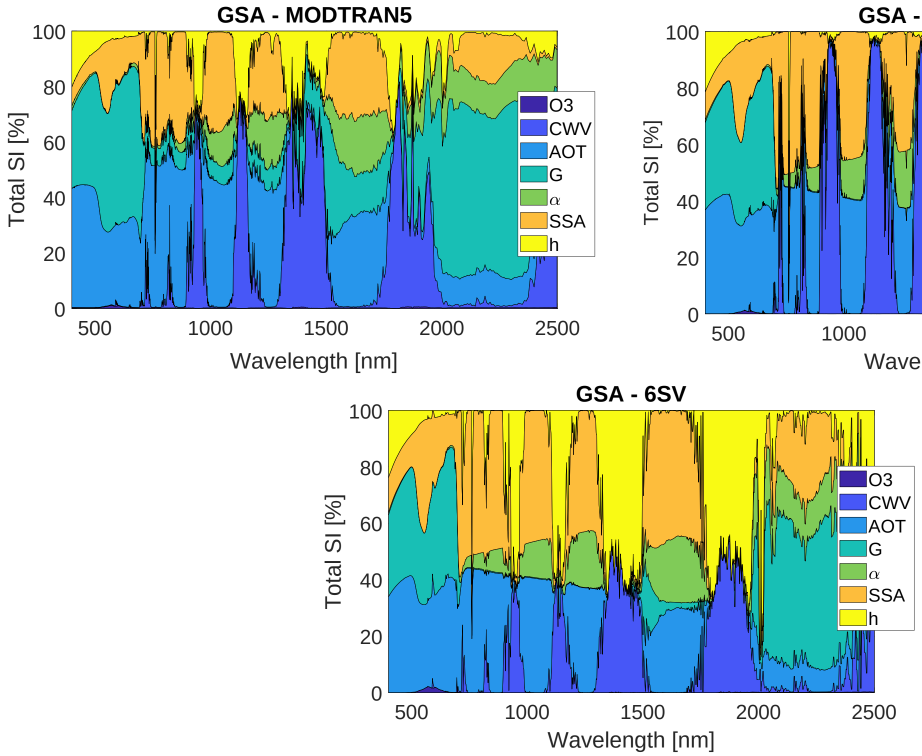


**Figure 5.** MODTRAN5, libRadtran and 6SV GSA results of main atmospheric properties at TOA radiance.

of $H_2O$ absorptions. All models also show a sudden decrease of the relative importance of the scattering processes (through the variables G and AOT) after $\sim$720 $\mathrm{nm}$. Indeed, according to Eq. (1), the high reflectance values of vegetation in the near infrared spectral region reduce the influence of the atmospheric path radiance (most affected by the scattering processes) with respect to the surface-reflected radiance. Despite of these similarities, the GSA figures also show some discrepancies, particularly on the lower importance of the aerosol absorption (through the SSA variable) in MODTRAN5 for wavelengths higher than $\sim$720 $\mathrm{nm}$. The MODTRAN5 model also shows some sensitivity $(5-10\%)$ to the asymmetry asymmetry parameter (G) in the 720-1300 $\mathrm{nm}$ spectral range while it is nearly 0% in libRadtran and 6SV. Important differences also appear on the relative sensitivity of surface elevation and CWV within the $H_2O$ bands. In fact, both variables compete to influence the strength of the $H_2O$ absorption, the CWV through its influence on the amount of $H_2O$ in the atmospheric column and surface elevation directly on the definition of the optical path of photons. In this case, 6SV shows higher dependency on the surface elevation than MODTRAN and libRadtran due to uncoupled scattering and absorption effects in 6SV. In 6SV, the $H_2O$ absorption only affects to the direct Sun-target-sensor transmittance component, which is dependent on both the CWV and optical path (and thus surface elevation). In MODTRAN and libRadtran, the multiple scattering increases the optical path of photons and thus the absorption by $H_2O$, which makes the model more sensitive to the CWV than surface elevation. However, MODTRAN and libRadtran still show differences in the relative sensitivity to CWV versus surface elevation, which might indicate differences in the implementation of the coupled absorption-scattering processes at these strong absorption features.

To further prove the usefulness of ALG to perform RTM intercomparison, we secondly repeated the study in (Kotchenova et al., 2008). Here, we compared 6SV simulations against MODTRAN's DISORT (Stamnes et al., 1988) (8 streams) and Isaac's 2-streams (Isaacs et al., 1987) RT solvers and libRadtran (DISORT solver with 8 streams). The simulations were performed with a US Standard 1976 atmospheric profile, the OPAC's continental average aerosol model optical properties, two values of AOT (0.2 and 0.8) and the same range of illumination/observation conditions described in (Kotchenova et al., 2008). The simulated LUTs were used to calculate the intrinsic atmospheric reflectance. The atmospheric reflectance from MODTRAN and libRadtran ($\rho'$) was compared with the simulated by 6SV ($\rho'_{6sv}$) according to the following cost function:

$$\delta(\tau, VZA, \lambda) = \frac{100}{N} \sum_{SZA} \sum_{RAA} \frac{|\rho'_{6sv} - \rho'|}{\rho'_{6sv}} \tag{3}$$

where, for sake of simplicity, we have omitted the dependency of the reflectance with the AOT ($\tau$), VZA, SZA, RAA and wavelength ($\lambda$). Figure 6 shows the results of the average relative differences for two wavelenghts ($\lambda$=412 $\mathrm{nm}$ and 670 $\mathrm{nm}$).

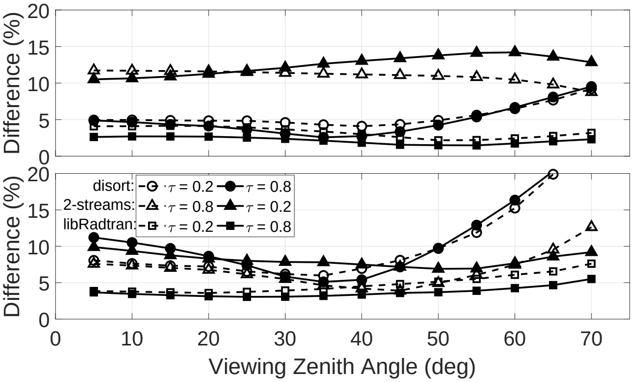


**Figure 6.** Average relative differences between 6SV and MODTRAN DISORT, Isaac's 2-streams and libRadtran DISORT for two wavelengths, 412 $\mathrm{nm}$ (top) and 670 $\mathrm{nm}$ (bottom), as function of VZA and AOT.

The results are compatible with those presented in (Kotchenova et al., 2008), showing differences (at 412 $\mathrm{nm}$) of 5-10% with respect to 6SV mostly due to the simulation of polarization in 6SV and the calculation of multiple-scattering by the Henyey–Greenstein aerosol phase function. These effects are also seen when using the Isaac's 2-streams radiative solver in MODTRAN, now with errors up to 15%. The discrepancies with respect to libRadtran are rather constant, with errors around 3-4%, probably since libRadtran introduces the phase function calculatd by OPAC for the simulation of scattering effects.

## 5 Other applications

As described in the section 3, ALG facilitates the usage of atmospheric RTMs and the generation of large LUTs of atmospheric transfer functions. Users can integrate these LUTs into a wide range of applications.

One of these applications is in End-to-end mission performance simulators (E2ES). E2ES are software tools that reproduce all aspects of satellite missions including the platform orbit/attitude, synthetic scene generation, sensor behavior, ground image processing and product evaluation (Kerekes et al., 1999; Segl et al., 2012). These tools are used by remote sensing scientists and engineers to support trade-off studies, to prepare of system calibration tests and to optimize data processing algorithms. As part of the European Space Agency's FLEX E2ES (Vicent et al., 2016), pre-computed MODTRAN-based LUTs generated with ALG are used to simulate the radiance signal as would be observed by FLEX mission instruments (Tenjo et al., 2017).

Another typical application of atmospheric LUTs is on the retrieval of aerosol physical and optical properties (Dubovik et al., 2002; Huang et al., 2015). From a satellite data processing perspective, aerosols are one of the main atmospheric components that must be accounted for when performing atmospheric correction (Thompson et al., 2018). In this frame, we studied the impact of aerosol type variability in the atmospheric correction within the $O_2$ absorption regions (Vicent et al., 2017). The goal was to determine whether the use of parametric approximations in aerosol properties can be used to peform the atmospheric correction in the $O_2$ absorptions. ALG was used to simulate several datasets with varying aerosol types, optical properties and vertical distribution.

The applicability of ALG is not limited to spaceborne instruments, but they is also suitable for the analysis of airborne and proximal sensing (e.g., flux towers, unmanned aerial vehicles). In our recent publication (Sabater et al., 2018), we studied the impact of path length in proximal sensing measurements of downwelling irradiance and at-sensor radiance, and their impact on sun-induced fluorescence retrieval. The study focused on remote sensing instruments placed at 2-50 m height over the surface. ALG was used to facilitate the running of MODTRAN simulation, which varied the instrument height and the SZA for standard atmospheric conditions.

Altogether, these few examples demonstrate the versatility of ALG to address multiple remote sensing applications based on the use of atmospheric RTMs.

## 6 Conclusions and future work

In this paper the main design concept and features of ALG have been described along with an intercomparison study for the atmospheric RTMs 6SV, MODTRAN and libRadtran. The a priori tedious tasks of (1) writting consistent input files, (2) running the RTMs in an efficient manner, (3) compiling and harmonizing the various model output files into ready-to-use LUT files, and (4) perform a model sensitivity analysis was largely simplified using the developed ALG tool and its compatibility with the ARTMO software framework (Verrelst et al., 2012). The sensitivity analysis results indicate that, overall, the various atmospheric RTMs simulate similarly the absorption and scattering processes for the selected atmospheric variables. However, there are still important differences in the sensitivity analysis that must be analysed in more detail.

Other practical applications, such as scene generation, atmospheric data analysis and atmospheric correction (Thompson et al., 2018), can also benefit from the use of ALG. A few application examples were presented, demonstrating the software capabilities to generate consistent LUTs for several atmospheric RTMs, with a wide range of input atmospheric variables, nodes distribution and spectral configurations. ALG is an ongoing work and regularly updated with new added functionalities and tools. The following upgrades are in the pipeline: (1) including the polarization data calculated by the 6SV code and the polRadtran and Mytic solvers in libRadtran, (2) implement functions to develop emulators of atmospheric transfer functions, (3) generation of LUTs of TOA radiance spectra for non-Lambertian and non-homogeneous surfaces, (4) implementation of additional RTMs such as RRTOV and SOS, and (5) compatibility with Linux and MacOS systems. Summarizing, ALG can become an useful tool to facilitate research on atmospheric radiative transfer, as well as opening the use of atmospheric RTMs to wider research communities and applications such as for climate studies, atmospheric physics and chemistry and remote sensing data processing.

*Code availability.* The exact version of the ALG (v2.0) used to produce the results used in this paper is archived on Zenodo (https://doi.org/10.5281/zenodo.3555575), as are input data and scripts to run the model and produce the plots for all the simulations presented in this paper. The current version of ALG is freely available from the project website (http://ipl.uv.es/artmo) under the GNU General Public License v3 (see http://www.gnu.org/licenses/). The software package has been developed in Matlab® R2018a and it is compatible with Windows operating systems. The tool is also provided as a stand-alone compiled executable file so that users not having a licence of Matlab can still run the software. Accordingly, users must first install the corresponding Matlab Runtime (MCR version 9.5, 64-bits). In addition, the help system of ALG includes a set of guidelines to install and compile the compatible atmospheric RTMs.

*Author contributions.* JVi and NS designed the ALG tool. JVi, JRC and JMM developed the ALG tool and performed the simulations. JVi and JVe designed the simulation experiments and carried them out. JVi prepared the manuscript with contributions from all co-authors.

*Competing interests.* The authors declare that they have no conflict of interest.

*Acknowledgements.* This work was partially carried out in the frame of ESA's project *FLEX L1B to L2 Algorithms Development study* with ESA-ESTEC contract no. AO/1-8897/17/NL/MP. Jochem Verrelst was supported by the European Research Council (ERC) under the ERC-2017-STG SENTIFLEX project (grant agreement 755617).

## References

Abramowitz, M. and Stegun, I.: Handbook of Mathematical Functions, in: Applied Mathematics Series, Volume 55., chap. 25.2, National Bureau of Standards, Washington, USA, 1964.

Barber, C., Dobkin, D., and Huhdanpaa, H.: The Quickhull Algorithm for Convex Hulls, ACM Transactions on Mathematical Software, 22, 469–483, 1996.

Bartels, R.H., B. J. and Barsky, B.: Hermite and Cubic Spline Interpolation, in: An Introduction to Splines for Use in Computer Graphics and Geometric Modelling, 2nd ed., chap. 3, pp. 9–17, Morgan Kaufmann., San Francisco, CA (USA), 1998.

Berk, A. and Hawes, F.: Validation of MODTRAN6 and its line-by-line algorithm, Journal of Quantitative Spectroscocy and Radiative Transfer, 203, 542–556, 2017.

Berk, A., Anderson, G., Acharya, P., Bernstein, L., Muratov, L., Lee, J., Fox, M., Adler-Golden, S., Chetwynd, J., Hoke, M., Lockwood, R., Gardner, J., Cooley, T., Borel, C., Lewis, P., and Shettle, E.: MODTRAN$^{TM}$5: 2006 update, in: Proceedings of SPIE - The International Society for Optical Engineering, vol. 6233 II, 2006.

Berk, A., Conforti, P., Kennett, R., Perkins, T., Hawes, F., and Van Den Bosch, J.: MODTRAN6: A major upgrade of the MODTRAN radiative transfer code, Proceedings of SPIE - The International Society for Optical Engineering, 9088, https://doi.org/10.1117/12.2050433, 2014.

Berk, A., Conforti, P., and Hawes, F.: An accelerated line-by-line option for MODTRAN combining on-thefly generation of line center absorption within 0.1 cm-1 bins and pre-computed line tails, Proceedings of SPIE - The International Society for Optical Engineering, 9472, https://doi.org/10.1117/12.2177444, 2015.

Berk, A. and, B. J., Hawes, F., Perkins, T., Conforti, P., Anderson, G., Kennett, R., and Achary, P.: MODTRAN6.0 User's Manual, Spectral Sciences, Inc., Contract No. FA9453-12-C0262; Data Itel A007, 193 pp., 2017.

Bratley, P. and Fox, B.: Algorithm 659 Implementing Sobol's Quasirandom Sequence Generator, ACM Transactions on Mathematical Software, 14, 88–100, 1988.

Brazile, J., Richter, R., Schläpfer, D., Schaepman, M., and Itten, K.: Cluster versus grid for operational generation of ATCOR's modtran-based look up tables, Parallel Computing, 34, 32–46, 2008.

Callieco, F. and Dell'Acqua, F.: A comparison between two radiative transfer models for atmospheric correction over a wide range of wavelengths, International Journal of Remote Sensing, 32, 1357–1370, 2011.

Cooley, T., Anderson, G., Felde, G., Hoke, M., Ratkowski, A., Chetwynd, J., Gardner, J., Adler-Golden, S., Matthew, M., Berk, A., Bernstein, L., Acharya, P., Miller, D., and Lewis, P.: FLAASH, a MODTRAN4-based atmospheric correction algorithm, its applications and validation, in: International Geoscience and Remote Sensing Symposium (IGARSS), vol. 3, pp. 1414–1418, 2002.

Debaecker, V., Louisand, J., Müller-Wilm, U., and Gascon, F.: Generation of Look-Up-Tables for the atmospheric correction module of Sentinel-2 Level 2A processor (Sen2Cor) using libRadtran and comparison with MODTRAN, in: Living Planet Symposium, vol. 1909, 2016.

Delaunay, B.: Sur la sphère vide. A la mémoire de Georges Voronoï, Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na, pp. 793–800, 1934.

Dubovik, O. and King, M.: A flexible inversion algorithm for retrieval of aerosol optical properties from Sun and sky radiance measurements, Journal of Geophysical Research Atmospheres, 105, 20 673–20 696, 2000.

Dubovik, O., Holben, B., Eck, T., Smirnov, A., Kaufman, Y., King, M., Tanré, D., and Slutsker, I.: Variability of absorption and optical properties of key aerosol types observed in worldwide locations, Journal of the Atmospheric Sciences, 59, 590–608, 2002.

Dubovik, O., Herman, M., Holdak, A., Lapyonok, T., Tanré, D., Deuzé, J. L., Ducos, F., Sinyuk, A., and Lopatin, A.: Statistically optimized inversion algorithm for enhanced retrieval of aerosol properties from spectral multi-angle polarimetric satellite observations, Atmospheric Measurement Techniques, 4, 975–1018, https://doi.org/10.5194/amt-4-975-2011, 2011.

El Hajj, M., Bégué, A., Lafrance, B., Hagolle, O., Dedieu, G., and Rumeau, M.: Relative radiometric normalization and atmospheric correction of a SPOT 5 time series, Sensors, 8, 2774–2791, 2008.

Emde, C., Buras-Schnell, R., Kylling, A., Mayer, B., Gasteiger, J., Hamann, U., Kylling, J., Richter, B., Pause, C., Dowling, T., and Bugliaro, L.: The libRadtran software package for radiative transfer calculations (version 2.0.1), Geoscientific Model Development, 9, 1647–1672, 2016.

Fell, F. and Fischer, J.: Numerical simulation of the light field in the atmosphere-ocean system using the matrix-operator method, Journal of Quantitative Spectroscopy and Radiative Transfer, 69, 351–388, https://doi.org/10.1016/S0022-4073(00)00089-3, 2001.

Forster, P., Fomichev, V., Rozanov, E., Cagnazzo, C., Jonsson, A., Langematz, U., Fomin, B., Iacono, M., Mayer, B., Mlawer, E., Myhre, G., Portmann, R., Akiyoshi, H., Falaleeva, V., Gillett, N., Karpechko, A., Li, J., Lemennais, P., Morgenstern, O., Oberländer, S., Sigmond, M., and Shibata, K.: Evaluation of radiation scheme performance within chemistry climate models, Journal of Geophysical Research Atmospheres, 116, 2011.

Gasteiger, J. and Wiegner, M.: MOPSMAP v1.0: a versatile tool for the modeling of aerosol optical properties, Geoscientific Model Development, 11, 2739–2762, 2018.

Gastellu-Etchegorry, J., Gascon, F., and Esteve, P.: An interpolation procedure for generalizing a look-up table inversion method, Remote Sensing of Environment, 87, 55–71, 2003.

Goody, R., West, R., Chen, L., and Crisp, D.: The correlated-k method for radiation calculations in nonhomogeneous atmospheres, Journal of Quantitative Spectroscopy and Radiative Transfer, 42, 539–550, 1989.

Guanter, L., Richter, R., and Kaufmann, H.: On the application of the MODTRAN4 atmospheric radiative transfer code to optical remote sensing, International Journal of Remote Sensing, 30, 1407–1424, 2009.

Hess, M., Koepke, P., and Schult, I.: Optical Properties of Aerosols and Clouds: The Software Package OPAC, Bulletin of the American Meteorological Society, 79, 831–844, 1998.

Huang, F., Zhou, J., Tao, J., Tan, X., Liang, S., and Cheng, J.: PMODTRAN: a parallel implementation based on MODTRAN for massive remote sensing data processing, International Journal of Digital Earth, 9, 819–834, 2016.

Huang, R.-J., Zhang, Y., Bozzetti, C., Ho, K.-F., Cao, J.-J., Han, Y., Daellenbach, K., Slowik, J., Platt, S., Canonaco, F., Zotter, P., Wolf, R., Pieber, S., Bruns, E., Crippa, M., Ciarelli, G., Piazzalunga, A., Schwikowski, M., Abbaszade, G., Schnelle-Kreis, J., Zimmermann, R., An, Z., Szidat, S., Baltensperger, U., El Haddad, I., and Prévot, A.: High secondary aerosol contribution to particulate pollution during haze events in China, Nature, 514, 218–222, 2015.

Iacono, M., Delamere, J., Mlawer, E., Shephard, M., Clough, S., and Collins, W.: Radiative forcing by long-lived greenhouse gases: Calculations with the AER radiative transfer models, Journal of Geophysical Research Atmospheres, 113, 2008.

Isaacs, R. G., Wang, W.-C., Worsham, R. D., and Goldenberg, S.: Multiple scattering lowtran and fascode models, Appl. Opt., 26, 1272–1281, 1987.

Jacquemoud, S., Verhoef, W., Baret, F., Bacour, C., Zarco-Tejada, P., Asner, G., François, C., and Ustin, S.: PROSPECT + SAIL models: A review of use for vegetation characterization, Remote Sensing of Environment, 113, S56–S66, 2009.

Kerekes, J. P., Baum, J., and Farrar, K.: Analytical model of hyperspectral system performance, Proceedings of SPIE - The International Society for Optical Engineering, 3701, 155–166, 1999.

Kocis, L. and Whiten, W.: Computational Investigations of Low-Discrepancy Sequences, ACM Transactions on Mathematical Software, 23, 266–294, 1988.

Koepke, P., Gasteiger, J., and Hess, M.: Technical Note: Optical properties of desert aerosol with non-spherical mineral particles: data incorporated to OPAC , Atmospheric Chemistry and Physics, 15, 5947–5956, 2015.

Kotchenova, S. and Vermote, E.: Validation of a vector version of the 6S radiative transfer code for atmospheric correction of satellite data. Part II: Homogeneous Lambertian and anisotropic surfaces, Applied Optics, 46, 4455–4464, 2007.

Kotchenova, S., Vermote, E., Matarrese, R., and Klemm Jr., F.: Validation of a vector version of the 6S radiative transfer code for atmospheric correction of satellite data. Part I: Path radiance, Applied Optics, 45, 6762–6774, 2006.

Kotchenova, S., Vermote, E., Levy, R., and Lyapustin, A.: Radiative transfer codes for atmospheric correction and aerosol retrieval: intercomparison study, Applied Optics, 47, 2215–2226, 2008.

Lenoble, J.: Radiative transfer in scattering and absorbing atmospheres: Standard computational procedures, A. Deepak Publishing, Vol. 300, Hampton, VA, USA, 1985.

Liu, J., Pattey, E., and Jégo, G.: Assessment of vegetation indices for regional crop green LAI estimation from Landsat images over multiple growing seasons, Remote Sensing of Environment, 123, 347–358, 2012.

Matarrese, R., Roger, J.-C., Kotchenova, S., Morcrette, J., Tanré, D., Deuzé, J., Herman, M., and Vermote, E.: MODIS Land Surface Reflectance - Science Computing Facility, University of Maryland, http://6s.ltdri.org/, [accessed 08/04/2018], 2015.

Matthew, M., Adler-Golden, S., Berk, A., Richtsmeier, S., Levine, R., Bernstein, L., Acharya, P., Anderson, G., Felde, G., Hoke, M., Ratkowski, A., Burke, H., Kaiser, R., and Miller, D.: Status of atmospheric correction using a MODTRAN4-based algorithm, Proc. SPIE - Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VI, 4049, 4049 – 4049 – 9, 2000.

Matthews, M., Bernard, S., and Winter, K.: Remote sensing of cyanobacteria-dominant algal blooms and water quality parameters in Zeekoevlei, a small hypertrophic lake, using MERIS, Remote Sensing of Environment, 114, 2070–2087, https://doi.org/10.1016/j.rse.2010.04.013, 2010.

Mayer, B. and Kylling, A.: Technical note: The libRadtran software package for radiative transfer calculations - description and examples of use, Atmospheric Chemistry and Physics, 5, 1855–1877, 2005.

Mayer, B. Emde, C. G. J. and Kylling, A.: LibRadtran website, http://www.libradtran.org/, [accessed 08/04/2018], 2017.

McKay, M., Beckman, R., and Conover, W.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics, 21, 239–245, 1979.

North, P., Brockmann, C., Fischer, J., Gomez-Chova, L., Grey, W., Heckel, A., Moreno, J., Preusker, R., and Regner, P.: MERIS/AATSR synergy algorithms for cloud screening, aerosol retrieval and atmospheric correction, in: European Space Agency, (Special Publication) ESA SP, 666 SP, 2008.

Pedrós, R., Gómez-Amo, J., Marcos, C., Utrillas, M., Gandía, S., Tena, F., and Lozano, J. M.: AEROgui: A Graphical User Interface for the Optical Properties of Aerosols, Bulletin of the American Meteorological Society, 95, 1863–1871, 2014.

Proud, S., Fensholt, R., Rasmussen, M., and Sandholt, I.: A comparison of the effectiveness of 6S and SMAC in correcting for atmospheric interference of Meteosat Second Generation images, Journal of Geophysical Research, 115, D17 209, 2010.

Richter, R.: A spatially adaptive fast atmospheric correction algorithm, International Journal of Remote Sensing, 17, 1201–1214, 1996.

Sabater, N., Vicent, J., Alonso, L., Verrelst, J., Middleton, E. M., Porcar-Castell, A., and Moreno, J.: Compensation of Oxygen Transmittance Effects for Proximal Sensing Retrieval of Canopy–Leaving Sun–Induced Chlorophyll Fluorescence, Remote Sensing, 10, 2018.

Saunders, R., Hocking, J., Turner, E., Rayer, P., Rundle, D., Brunel, P., Vidot, J., Roquet, P., Matricardi, M., Geer, A., Bormann, N., and Lupu, C.: An update on the RTTOV fast radiative transfer model (currently at version 12), Geoscientific Model Development, 11, 2717–2737, 2018.

Schläpfer, D.: MODO: MODTRAN for Remote Sensing Applications, ReSe Applications, http://www.rese.ch, [accessed 08/04/2018], 2016.

Segl, K., Guanter, L., Rogass, C., Kuester, T., Roessner, S., Kaufmann, H., Sang, B., Mogulsky, V., and Hofer, S.: EeteS - The EnMAP end-to-end simulation tool, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5, 522–530, 2012.

Seidel, F., Kokhanovsky, A., and Schaepman, M.: Fast and simple model for atmospheric radiative transfer, Atmospheric Measurement Techniques, 3, 1129–1141, 2010.

Shepard, D.: Two-dimensional interpolation function for irregularly- spaced data, Proc 23rd Nat Conf, pp. 517–524, cited By 1561, 1968.

Stamnes, K., Tsay, S.-C., Wiscombe, W., and Jayaweera, K.: Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media, Appl. Opt., 27, 2502–2509, 1988.

Tenjo, C., Rivera, J., Sabater, N., Vicent, J., Alonso, L., Verrelst, J., and Moreno, J.: Design of a generic 3D scene generator for passive optical missions and its implementation for the ESA's FLEX/Sentinel-3 tandem mission, IEEE Transactions on Geoscience and Remote Sensing, 55, pp–pp, 2017.

Theys, N., Van Roozendael, M., Hendrick, F., Fayt, C., Hermans, C., Baray, J.-L., Goutail, F., Pommereau, J.-P., and De Maziere, M.: Retrieval of stratospheric and tropospheric BrO columns from multi-axis DOAS measurements at Reunion Island (21°S, 56°E), Atmospheric Chemistry and Physics, 7, 4733–4749, 2007.

Thompson, D. R., Natraj, V., Green, R. O., Helmlinger, M. C., Gao, B.-C., and Eastwood, M. L.: Optimal estimation for imaging spectrometer atmospheric correction, Remote Sensing of Environment, 216, 355 – 373, 2018.

Verhoef, W. and Bach, H.: Simulation of Sentinel-3 images by four-stream surface-atmosphere radiative transfer modeling in the optical and thermal domains, Remote Sensing of Environment, 120, 197–207, 2012.

Vermote, E., Tanré, D., Deuzé, J., Herman, M., and Morcrette, J.-J.: Second simulation of the satellite signal in the solar spectrum, 6S: an overview, IEEE Transactions on Geoscience and Remote Sensing, 35, 675–686, 1997.

Verrelst, J., Romijn, E., and Kooistra, L.: Mapping Vegetation Density in a Heterogeneous River Floodplain Ecosystem Using Pointable CHRIS/PROBA Data, Remote Sensing, 4, 2866–2889, 2012.

Verrelst, J., Sabater, N., Rivera, J., Muñoz Marí, J., Vicent, J., Camps-Valls, G., and Moreno, J.: Emulation of leaf, canopy and atmosphere radiative transfer models for fast global sensitivity analysis, Remote Sensing, 8(8), 673, 2016.

Verrelst, J., Vicent, J., Rivera-Caicedo, J. P., Lumbierres, M., Morcillo-Pallarés, P., and Moreno, J.: Global Sensitivity Analysis of Leaf-Canopy-Atmosphere RTMs: Implications for Biophysical Variables Retrieval from Top-of-Atmosphere Radiance Data, Remote Sensing, 11, 2019.

Verstraete, M., Diner, D., and Bézy, J.-L.: Planning for a spaceborne Earth Observation mission: From user expectations to measurement requirements, Environmental Science & Policy, 54, 419 – 427, https://doi.org/http://dx.doi.org/10.1016/j.envsci.2015.08.005", 2015.

Vicent, J., Sabater, N., Tenjo, C., Acarreta, J., Manzano, M., Rivera, J., Jurado, P., Franco, R., Alonso, L., Verrelst, J., and Moreno, J.: FLEX End-to-End Mission Performance Simulator, IEEE Transactions on Geoscience and Remote Sensing, 54, 4215–4223, 2016.

Vicent, J., Sabater, N., Verrelst, J., Alonso, L., and Moreno, J.: Assessment of Approximations in Aerosol Optical Properties and Vertical Distribution into FLEX Atmospherically-Corrected Surface Reflectance and Retrieved Sun-Induced Fluorescence, Remote Sensing, 9, 2017.

Vicent, J., Alonso, L., Martino, L., Sabater, N., Verrelst, J., Camps-Valls, G., and Moreno, J.: Gradient-based Automatic Look-Up Table Generator for Radiative Transfer Models, IEEE Transactions on Geoscience and Remote Sensing, pp. 1–9, 2018.

Wilson, R.: Py6S: A Python interface to the 6S radiative transfer model, Computers & Geosciences, 51, 166 – 171, 2013.