

KARINE : K-distribution Atmospheric Radiation: Infrared Net Exchanges Version 3.4.5, User Manual

July 23, 2008

V. Eymet, Laboratoire d'Energétique, Université Paul Sabatier, 31062
Toulouse, France

Contents

1	Introduction	3
1.1	Purpose of the code	3
1.2	Geometric configuration	3
1.3	Physical hypothesis	3
1.4	Numerical method	4
1.5	Analysis formalism	5
1.6	Informatics	5
2	Installation and compilation	8
2.1	Notations and conventions	8
2.2	Program installation	8
2.3	Compiling the program	9
3	Input data	11
3.1	Configuration, geometry and hypothesis	11
3.2	The data file	11

4	Using the code	16
4.1	Settings	16
4.1.1	File “\$RAYDIR/data.in”	16
4.1.2	File “\$RAYDIR/options.in”	20
4.1.3	Variables in file “\$RAYDIR/includes/max.inc”	27
4.2	Running a computation	28
4.2.1	Launch the program	28
4.2.2	Messages displayed during a successfull run	29
4.2.3	Checking the computation progress	30
4.2.4	Tips	31
5	Results visualization	32
5.1	Data plots	32
5.2	Results plots	32
5.3	Tips and advices	33
6	Frequently asked questions	34

1 Introduction

This document is intended to be used by any new user of the code KARINE . The basic linux commands are supposed to be known (file system commands, environment variables use, basic shell scripts).

The user is free to use and modify the sources of the code. It has been written in fortran 77, which imposes some limitations, mostly for memory management.

Any question, remark, or improvement suggestion is welcome, and should be submitted by e-mail to the author (Vincent Eymet, eymet@energetique.ups-tlse.fr).

1.1 Purpose of the code

KARINE is a radiative transfer computing code that can be used to solve infrared radiative transfer problems in absorbing and scattering planetary atmospheres. This program is using a Net Exchange Formulation (NEF) [1, 2, 3, 4, 5, 6], different from traditional flux or radiative budget formulations and analysis. It is based on the Monte-Carlo method.

1.2 Geometric configuration

The program uses a plane-parallel geometric configuration (also referred as a “slab” configuration). In the case of planetary atmospheres, using such a geometric configuration will result in radiative transfer computations for a horizontally homogeneous atmosphere (radiative properties of the atmosphere can only vary with altitude). The effects of planetary curvature are not taken into account.

1.3 Physical hypothesis

In the general case, a vertically inhomogeneous atmosphere (variations of species concentrations, but also of temperature, with altitude) is divided into a number of semi-transparent media layers, within which optical properties are homogeneous. These atmospheric layers are surrounded by two parallel opaque planes (ground and space), characterized by a temperature and an hemispheric emissivity.

In order to represent the temperature field, temperatures of ground and space have to be specified. These values are independent of temperatures at the bottom and at the top of the atmosphere. For gas layers, values of temperature must be given at the center of each atmospheric layer, at the bottom and at the top of the atmosphere. Temperature is considered as linear between two consecutive data points.

The program is currently using a discrete k-distribution spectral model¹ with the possibility to use any quadrature order. The spectral model is also using the CK hypothesis for taking into consideration the effects of inhomogeneities. Every optical property (boundaries emissivities, gas and clouds absorption and scattering coefficients) have to be defined for each quadrature element, in each narrowband.

1.4 Numerical method

The program KARINE is based on the Monte-Carlo numerical method [7, 8, 9]. Its main advantage is the possibility to compute, in addition of each result, a statistical standard deviation (that can be interpreted in terms of numerical uncertainty) over each result. The method mainly consists in generating the optical path of a number of energy bundles (also denoted as “rays” in this text) from their emission to their absorption point. These optical paths in a scattering medium are represented by a succession of non-curved segments.²

Radiative transfer computations in the infrared part of the spectrum using the Monte-Carlo method traditionally encounters well known numerical difficulties: problems occur for frequencies at which the semi-transparent medium is highly absorbing (typically, for water or carbon dioxide absorption bands in the case of the terrestrial atmosphere). In a traditional Monte-Carlo algorithm, energy bundles that have been emitted at these frequencies will be absorbed in the immediate vicinity of their emission position; in order to compute radiative exchanges between distant locations with a reasonable accuracy (a low standard deviation), a great number of energy bundles must be simulated.

¹It would be fairly easy to use instead a continuous k-distribution spectral model, such as the k-distribution model corresponding to the Malkmus statistical model

²In particular, the effects of trajectories curvature associated with a variation of the refractive index (“mirrage” effect) are not taken into account.

In KARINE , this problem was solved by using in the one hand, a formulation of radiative transfer that is adapted to optically thick medium and quasi-isothermal medium, and in the other hand, statistical sampling laws that can adapt to any medium optical thickness. For more information, please refer to article [5].

A special mention must be made to the spectral sampling optimization. Each energy bundle is associated with a frequency and a quadrature index (this is due to the discrete k-distribution spectral model). The (also discrete) spectral probability density function used for frequencies and quadrature index choices is based on a preliminary estimation of the radiative transfer results, using a number of analytic expressions. Since the frequencies that are relevant for the computation of a given result (for instance fluxes) may be irrelevant for computing another result (for instance radiative budgets), the program must be specified which kind of result the spectral probability density function (pdf) must be computed for. This pdf is then used by the code for the Monte-Carlo computation. In most cases, one result will be computed with a greater accuracy than the others. For instance when the spectral pdf is based on the analytic expressions of radiative budgets, these budgets will be computed with a better numerical accuracy (lower standard deviation) than radiative fluxes.

1.5 Analysis formalism

The program is using a Net Exchange Formulation for Net Exchange Rates computations, but also a flux formulation for computing fluxes. The Net Exchange Rate (NER) between two geometric elements (boundary or gas layer) i and j is defined as the energy rate emitted by i and absorbed by j , minus the energy rate emitted by j and absorbed by i (see [6]). Using the NEF, it is possible to have a quick and clear view of all radiative NER (energetic redistributions within the system) and thus a better understanding of the dominant radiative transfer mechanisms.

1.6 Informatics

Compiling the program KARINE does not require external libraries: all source files are included in the package. However, data and results visualization requires the software **GRADS**. This software can be freely downloaded

at the following URL: <http://grads.iges.org/grads.html> ; warning: when installing binaries from the provided archive, you will have to create a symbolic link from the “gradsc” command to “grads” (into /usr/local/bin or wherever you install the executable files).

Data and results visualization will also require **ghostview**, accessible via the command “gv” (symbolic link or alias).

KARINE will use small java applets for printing messages on-screen. It is preferable to have the **javac** command (java compiler) installed on your system (it is available into the java-jdk toolkit) otherwise you will get errors at execution time (even if those errors will not prevent KARINE from running).

When using the program for the first time, it will ask if you accept to send an information email to the author each time a computation starts, for support (personalized compilation options) and debugging (whenever an unknown error occurs). If you accept to send emails, you will need the **mutt** or **mail** command-line tool properly installed on your system; **mutt** is preferred since it will smoothly send relevant files (options.in, data.in and atmos_data.txt files) as attachments.

In terms of CPU time, two separate stages must be considered: the preliminary stage, that mainly consists in building the spectral probability density function, and the Monte-Carlo computation by itself. It is also possible to pre-compute different spectral pdfs, for a number of atmospheric configurations, and to save them into a number of files. It is also possible to use a pre-computed spectral pdf in order to skip the preliminary stage of the computation. Orders of magnitude for CPU time and memory loads are given in the following chart, for a typical modern machine (64 bits processors), for a typical atmospheric configuration: $N_{max} = 90$ (maximum number of atmospheric layers), $N_{bmx} = 70$ (maximum number of spectral intervals) and $N_{qmx} = 16$ (maximum quadrature order) that are defined in the “/includes/max.inc” file. You may get different scores using different values of N_{max} , N_{bmx} and N_{qmx} (see section 4.1.3 for more information about this file and how to change the values of those variables).

Please note that for a typical atmospheric configuration, 1Gb of RAM should be enough for running both the system and KARINE itself.

Phase	Settings	CPU time	Memory load
Initialization phase	without computation of sensitivities (standard)	15 min	25 Mb
	with computation of $d(\text{NER})d(\text{T})$	12 hours	400 Mb
	with computation of both $d(\text{NER})d(\text{T})$ and $d(\text{NER})d(\text{X})$	24 hours	440 Mb
Monte-Carlo computation	10.000 realizations per layer	5 min	380 Mb
	1.000.000 realizations per layer	8 hours	380 Mb

2 Installation and compilation

2.1 Notations and conventions

Throughout this document, the following conventions will be used:

- Folder and file names will be noted bold and between quotation marks (ex: “**/home/user/**”)
- Linux commands will be noted in italic and between quotes (ex: ‘*cd /home/user*’).

Moreover, it is highly recommended for the user to define a single environment variable that points to the code main folder (folder that is created during the installation of the code). In this document, this environment variable will be referred as RAYDIR. As an example, variable RAYDIR can point to folder “**/home/user/Monte-Carlo/karine[X]**”, where **user** represents the user name and **X** is the code version number (the latest version is currently version 3.4.5, therefore we have **karine[X]=karine3.4.5**). The definition of this environment variable is usually done in file `.bashrc` (using command ‘*export RAYDIR=/home/user/Monte-Carlo/karine3.4.5*’) or in file `.cshrc` (using command ‘*setenv*’ instead of ‘*export*’). The code main folder will thus be referred as “**\$RAYDIR**”.

2.2 Program installation

The installation of KARINE requires to decompress an archive (probably named “**karine.tgz**”) using the command ‘*tar -zxvf karine.tgz*’. Folder “**\$RAYDIR**” that is created during the decompression will include the following sub-folders :

- “**\$RAYDIR/data/**”: folder where reside KARINE input data files. A number of example programs is provided in this folder -> see the README file.
- “**\$RAYDIR/Doc/**”: contains documentation files.
- “**\$RAYDIR/results/**”: this is the main code output folder.
- “**\$RAYDIR/plot/**”: this folder contains the “**plot**” script that is used for data and results visualization. See the README file.

- “**\$RAYDIR/source/**”: folder that contains the code source files (.for extension).
- “**\$RAYDIR/includes/**”: folder that contains the include files (.inc extension) used for the declaration of some variables.
- “**\$RAYDIR/objects/**”: folder that contains the object files (.o)
- “**\$RAYDIR/config/**”: folder that contains KARINE configuration file. See “**configure.txt**” that is created by KARINE on the first run.

Moreover, the folder “**\$RAYDIR**” will also include the following files :

- “**Makefile**”: is necessary to compile the executable file.
- “**data.in**” and “**options.in**”: needed to modify the Monte-Carlo algorithm behaviour, as well as the main program inputs/outputs. (cf. 4).

2.3 Compiling the program

Once the program sources are installed, the first compilation can be started using the command *'make clan all'* in folder “**\$RAYDIR**”. Compiling the program will first create object files (.o extension) in folder “**\$RAYDIR/objects/**” for each source file in folder “**\$RAYDIR/source/**”, and then will link these files to create the executable file “**KARINE** ”.

Every time a source file is modified, the executable must be recompiled, using the command *'make all'*. Re-compiling the code will overwrite the object file(s) corresponding to the source file(s) that has(have) been modified (and only these files) and finally generate a new executable file. This results in faster compilation.

Whenever a new subroutine must be created, it is advised to create a new source file (in folder “**\$RAYDIR/source/**”) for this subroutine. The name of this source file has then to be added to the source file list that resides in “**Makefile**”.

When an include file (in folder “**\$RAYDIR/includes/**”) must be modified ³, it is safe to remove every object file (using command *'make clean'*)

³Which is the case in particular when arrays sizes that are defined in “**\$RAYDIR/includes/max.inc**” become too small. This point will be detailed at chapter 4.

then to perform a fresh new compilation (using command *'make all'*). The full list of options that can be used along with the command *'make'* can be supplied using *'make help'*. Alternatively, the file “**Makefile**” can be edited.

Notice : editing “**\$RAYDIR/Makefile**” can be necessary in order to change the compiler name (*'g77'* by default), the main executable name (*'KARINE '* by default), or even to specify better compilation options, so that the program can run faster (options relative to the processor and various optimization, debugging, or parallelization options).

3 Input data

3.1 Configuration, geometry and hypothesis

Originally, KARINE has not been specifically written for atmospheric applications. It was designed for general infrared radiative transfer computations in absorbing and scattering, potentially inhomogeneous media, with the following constraints:

- A plane-parallel (slab) geometric configuration - atmospheric radiative transfer problems can be assimilated to a sub-class of slab problems.
- The participating media can be inhomogeneous in concentration and temperature; however, inhomogeneous media will be discretized into a succession of homogeneous gas layers.
- The spectral model must be a k-distribution model, along with the CK hypothesis in order to take inhomogeneities into account.
- In the case of planetary atmospheres, the temperature profile will be considered as linear between two consecutive gas layers centers; this results from the need to have a continuous and derivable temperature profile at layers interfaces (in order to make correct computations for optically thick media).

KARINE uses a raytracing algorithm for scattering media that is used in slab configurations, which is a valid approximation for numerous domains of application, among which combustion and medical diagnosis.

Notice 1 : the lower plane (ground) and upper plane (space) are denoted respectively layer 0 and layer $m + 1$. This is fairly important for data and results visualization.

3.2 The data file

The computing code needs an input data file: a description of the computation grid, temperature and species concentration data, optical properties for each component of the medium, for each layer, for each narrowband. In

order to simplify the data entry process, a single data file, with all necessary information for the program to run, must be created in folder “**\$RAYDIR/data/**” before running any computation. The name of this data file may be changed in file “**\$RAYDIR/data.in**”; however, its default name is “**atmos_data.txt**”.

The example program “**make_data.for**” provided in folder “**\$RAYDIR/data/**” can be used as a basis in order to create your own input data files. Command ‘*fo*’ will compile this program in order to create the executable file “**make_data.exe**”.

What contains the input data file :

- m is the number of gas layers, N_b is the number of narrowbands, and N_q is the k-distribution quadrature order.
- The m values of altitude (in meters) in array *alt* are altitudes for which we know the temperature; the computing grid (array *e*) altitudes will be obtained from these data points altitudes, as shown on figure 2: initially, we know m temperature values, along with the m corresponding altitudes $alt(i), i = 0, m - 1$ these temperatures are given at. We want to produce a m layers computing grid (array $e(i), i = 0, m$) with temperatures known at layers centers.⁴ The $m + 1$ altitudes of array *e* will be derived as follows: $e(0) = alt(0) = 0$, $e(i) = \frac{alt(i-1) + alt(i)}{2}$ for $i \in [1, m]$ and $e(m) = alt(m)$. The m temperature data points are respectively ground temperature, layer 2 center temperature, etc, layer $m - 1$ center temperature and space temperature. As KARINE only considers the infrared part of the spectrum, there is no downward radiation from space: space temperature will therefore be null.

Warning: in the configuration described by figure 2, it is not possible to specify center temperatures for layers index 1 and m . Which is a possible cause of discomfort for subsequent results analysis for atmospheric applications. It is however possible to visualize (see section 5) all “recombined” results that are computed by KARINE in addition to standard results. Standard NERs and radiative budgets are computed for the standard analysis grid “e” that is shown in figure 2. Recombined

⁴The temperature profile will therefore be considered as linear between two consecutive layers centers.

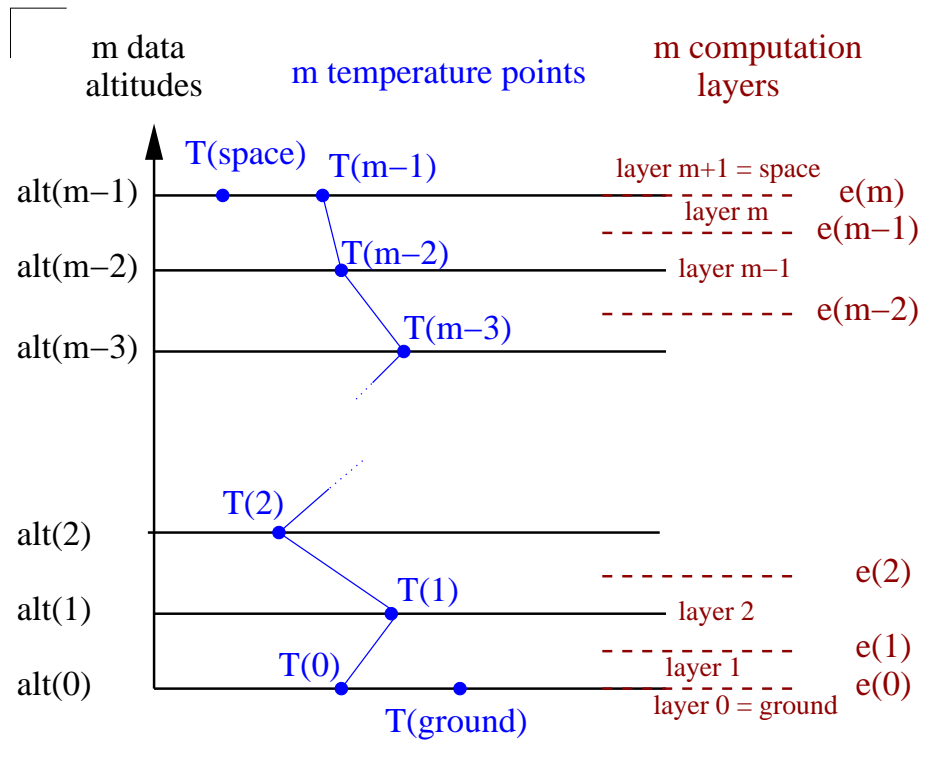


Figure 2: Meshing the atmosphere

NERs and radiative budgets are computed over a “recombined” mesh, where the first layer is located between $e(0)$ and $e(2)$, the second layer is between $e(2)$ and $e(3)$, etc, and the last (index $m-2$) layer is located between $e(m-2)$ and $e(m)$. In other words, the first layer of the recombined grid is constituted by layers 1 and 2 of the standard grid, and the last layer of the recombined grid is constituted by layers $m-1$ and m of the standard grid. In this recombined grid, $\text{temp}(1)$ is specified in the first layer, and $\text{temp}(m-2)$ is specified in the last layer. Recombined NERs and radiative budgets are therefore represented over a $(m-2)$ gas layers grid.

- The $m + 1$ altitudes of array e . $e(0)$ is the ground altitude (zero), $e(i), i = 1, m$ are the top altitudes (in meters) of the m atmospheric layers.

- The m values of pressure (in Pa), known for the m altitudes of array *alt*.
- The m values of gas temperature (in K), known for the m altitudes of array *alt*, followed by T_g the temperature of ground and T_s the temperature of space.
- The m values of volumic mass (in kg/m^3), known for the m altitudes of array *alt*.
- The m values of adiabatic lapse rate (in K/km) for each altitude of *alt*.
- The m values of gravity acceleration (in m/s^2) for each altitude of *alt*.
- The m values of non-ideality parameter γ (in law $P.V = n.R.T.\gamma$), dimensionless, for each altitude of *alt*.
- The m values of thermal capacity (in $J/kg/K$) for each altitude of *alt*.

Next, for each narrowband (label *band*), we have:

- The value of wavelenght (in μm) lower limit for the given narrowband, the value of wavelenght upper limit for the given narrowband, the value of lower plane (ground) emissivity in the given narrowband, the value of upper plane (space) emissivity in the given narrowband, the N_q values of the weighting coefficients for the N_q order quadrature, and finally values of the user-defined flux that must be emitted by the ground and by space in the current spectral interval (see file “options.in”, in section “spectral optimizations”).
- Then for $quad = 1$ to N_q :
 - The m values of total extinction coefficient (in m^{-1}) for the gas.
 - The m values of absorption coefficient (in m^{-1}) for the gas.
 - The m values of scattering coefficient (in m^{-1}) for the gas.
 - The m values of single-scattering albedo (dimensionless) for the gas.
 - The m values of phase function assymetry parameter (dimensionless) for the gas.

- The m values of total extinction coefficient (in m^{-1}) for the aerosol.
- The m values of absorption coefficient (in m^{-1}) for the aerosol.
- The m values of scattering coefficient (in m^{-1}) for the aerosol.
- The m values of single-scattering albedo (dimensionless) for the aerosol.
- The m values of phase function asymmetry parameter (dimensionless) for the aerosol.
- The m values of total extinction coefficient (in m^{-1}) for the clouds.
- The m values of absorption coefficient (in m^{-1}) for the clouds.
- The m values of scattering coefficient (in m^{-1}) for the clouds.
- The m values of single-scattering albedo (dimensionless) for the clouds.
- The m values of phase function asymmetry parameter (dimensionless) for the clouds.

It should be noticed that the information contained in file “**atmos_data.txt**” is redundant: for instance, if absorption and scattering coefficients are known, it should not be necessary to record also total extinction coefficient and single-scattering albedo values. It would also be possible to record only the values of total absorption and scattering coefficients for the media in its globality, without making such a clear distinction between the components (gas, aerosol, clouds). However, this data file is intended to include all the available data for a given atmospheric configuration. This file can be saved for further use, or can be used by some other program designed on the basis of the same physical hypothesis (linear temperature profile, optical properties constant throughout each gas layer, the use of a k-distribution spectral model, etc) but using total extinction coefficient and single scattering albedo instead of absorption and scattering coefficients.

4 Using the code

This chapter will describe the different steps that should be completed before running an actual atmospheric run, after the data file “**atmos_data.txt**” has been created, then it will precise what should be normally displayed on screen during an actual run.

4.1 Settings

Once successfully compiled, the program behaviour can be modified depending on what it actually has to compute, and also on a number of settings the user can modify. There are two important files that should be checked before running the program: “**\$RAYDIR/data.in**” and “**\$RAYDIR/options.in**”, that have a number of connections.

4.1.1 File “**\$RAYDIR/data.in**”

This file contains a number of values that should be set before running the computation code. The values that should be specified in “**data.in**” are the following:

Section “General data” :

- Emission angle at the boundaries: value of the emission angle for rays that are emitted at the boundaries. This value is used only when it has been chosen (using the appropriate option in file “**options.in**”) to start rays optical pathes at a given fixed angle. The value must be in degrees, and refers to a direction relative to the local normal. For instance, in order to launch rays with a fixed emission direction, perpendicularly to the lower and upper planes, this value should be set to “0.0”.
- Emission angle for the gas: value of the emission angle for rays that are emitted by the participating media, in the case that these rays must be emitted at a fixed angle (similarly to the previous value).
- Minimum / maximum wavelenght, in μm . This will set the radiative transfer integration wavelenght range. These values will be used in the case that it has been chosen (in file “**options.in**”) to make a radiative transfer computation for a fraction of the spectral range the spectroscopic data is available at. The specified values are used as follows:

the first (resp. last) narrowband used for spectral integration will be the one that includes the lowest (resp. greatest) specified wavelength. As such, limits of the spectral interval will not really be those who are specified, but rather the lowest wavelength of the first narrowband, and the greatest wavelength of the last narrowband, which are themselves found according to the specified values.

- Computation accuracy for NER (in W/m^3) for fluxes (in W/m^2) for net fluxes (in W/m^2): this is used to stop multiple scattering optical pathes whenever the energy bundle's contribution to Net Exchange Rates or fluxes becomes weaker than the specified value. In order to not introduce extra numerical uncertainty, this value will be set as small as possible.
- Number of angular sectors: is used to discretize the $[0 - \pi]$ interval, in order to draw integrated upward and downward fluxes as a function of the angle. Warning: this value has to be even ! The code will return an error if the specified value is odd. The code will generate the following results files:
 1. "Flux_asc_int_ang.txt" and "Flux_des_int_ang.txt": for each layer interface, there are four columns: the minimum and maximum values of the angular sector, the value of the upward (downward) flux in W/m^2 , and its uncertainty in W/m^2 . Angles are specified as follows: values between $\pi/2$ and $3\pi/2$ denoted upward directions. Values between $3\pi/2$ and $\pi/2$ denote downward directions (in other words, a null angle corresponds to the downward vertical).
 2. "angular_reflectivity.txt" and "angular_transmittivity.txt": for each layer interface, there are four columns: the minimum and maximum values of the angular sector, the value of reflectivity (transmittivity) (unitless) and its uncertainty (unitless). Reflectivity and transmittivity are the fractions of energy emitted by the ground that are transmitted or backscattered to each layer interface, in each angular sector. These results are only valid if only the ground emits radiation: temperatures within the gas and for space have to be null in order for these results to be meaningful.

- Scattering events max number: imposes a limit on the number of possible scattering events for every multiple scattering optical path. This parameter will only be taken into account when the associated option in file “**options.in**” is set so that optical paths are stopped after a number of scattering events. This option is intended at quantifying for instance the fraction of a given result (radiative budget, NER, flux) that is due to photons that have been scattered 0 times, 1 time, 2 times,... n times.
- Number of random events per layer: this is the number of ray bundles, or energy bundles, or multiple scattering optical paths, that will be emitted from each geometric element (plane or atmospheric layer). The bigger is this value, the more accurate the results will be (or the weaker the Monte-Carlo numerical uncertainty will be). Warning: in order to reduce by a factor 10 the numerical uncertainties attached to the results, the number of random events per layer (and therefore the CPU time) has to be increased by a factor 100.

The following parameters are the input and output file names. Even if there is no reason to change these, the user has the possibility to do so.

Section "Input data files" By default, these files will be looked for in the *data* folder.

- Name of the file that contains all optico-geometric information describing an atmospheric configuration. Default name: **"atmos_data.txt"**.
- Name of the file that contains all data relative to the clouds phase function. Phase function values must be specified for a given number of directions. This gives the possibility of use a phase function different from the default Henyey-Greenstein phase function that takes only the asymmetry parameter as an input. There is an option in file **"options.in"** for using a different phase function whose values are given in this file. Default value: **"phase_function.txt"**
- Name of the GCM pressure grid file: the user can specify to use a different pressure grid than the one provided in the input data file **atmos_data.txt**. See description of the **"options.in"** file for more detailed information about using a different pressure grid. Default value: **"meshGCM.txt"**
- Name of the file that contains sensitivities of absorption coefficients to temperature. Default value: **"dkadT_pos0.01.txt"**. This information is used when computation of NERs to temperature is required. An example program for creating such a file is provided in the *data* folder: **"write_dkadT"**. Note: sensitivities computations are disabled in the "light" version of KARINE .
- Name of the file that contains sensitivities of absorption coefficients to chemical species concentrations. Default value: **"dkadx.txt"**. This information is used when computation of NERs to concentrations is required. An example program for creating such a file is provided in the *data* folder: **"write_dkadx"**. Note: sensitivities computations are disabled in the "light" version of KARINE .

Section "Major output data files"

- File where the spectral pdf is stored. Default: **"freq_opti.txt"**. This file will be created within the *data* folder, and may be used by a new run.

- File containing interpolated optico-geometric data. Default: “`atmos_data_interp.txt`”. This file will be created within the `data` folder.
- Path to sensitivities output folder: indicate the path name to the folder where sensitivities results have to be stored. Sensitivities to temperature and concentration output files can easily weight more than a Gb. KARINE will compute and print the exact amount of disk space needed before the computation. Note: sensitivities computations are disabled in the “light” version of KARINE .

Section “Major output data files”

Better not to change anything here.

4.1.2 File “`$RAYDIR/options.in`”

In this file, all options relative to the Monte-Carlo algorithm behaviour can be tweaked, along with a number of options relative to the way the program will work:

Monte-Carlo core algorithm options

+ **Angular settings:**

- Emission angle fixed for boundaries: if this option is set to 1, every ray emitted at a boundary will leave it at a fixed angle, whose value should be specified in file “`data.in`”. If this option is set to 0, ray emitted at a boundary will leave it at random angles.
- Angular sampling follows Lambert law for boundaries: if this option is set to 1, rays emitted at a boundary will be emitted at random angles, with a lambertian probability density function. If this option is set to 0, rays emitted at a boundary will use a different angular pdf.
- Angular sampling follows isotropic law for boundaries: if this option is set to 1, rays emitted at a boundary will be emitted according to an isotropic pdf. If it is set to 0, another angular pdf will be used. It should be noticed that if no option is set to chose the angular pdf, emission angles will be chosen according to a special angular pdf that was optimized as a function of the medium optical thickness (see. [10, 6]).

- Specular or isotropic reflexion at the boundaries: if this option is set to 1, the fraction of energy that falls on a boundary and that will be reflected by it will be reflected using a specular angular pdf. If it is set to 2, reflexion will occur using an isotropic angular pdf (diffuse reflexion).
- Emission angle fixed for gas layers Angular sampling follows lambert law for gas layers Angular samling follows isotropic law for gas layers: identical to the same options for the boundaries.
- Optimized angular pdf: if this option is set to 1, and no other option relative to angular pdf choice is activated, the angular pdf used in the Monte-Carlo algorithm will be the one developpend by de Lataillade et al, 2002 [10]. It is optimized for purely absorbing media. This pdf has since been improved to take into account scattering (see. [6, 5]). This second pdf is used when this option is set to 2.

+ **Scattering algorithm:**

- Emission algorithm in gas layers choice: the value of this option will select the algorithm used for energy bundles emission within gas layers. Details concerning the two possible algorithms are given in V. Eymet PhD document [6]. In short, algorithm number 2 has a better behaviour (enables a faster numerical convergence) for scattering media. Algorithm number 1 is a “classical” emission algorithm. Both algorithms will have similar convergence speeds for (nearly) purely scattering media.
- Phase function: if this option is set to 1, each scattering event by a cloud of particles (aerosols, water clouds, etc) will use the Henyey-Greenstein phase function, whose asymetry parameter is given in the optico-geometric input data file. If it is set to 2, the program will look for a “user defined” phased function in file “**fonction.txt**”. This file should contain the value of the phase function for a given number of direction, for each narrowband (the format is has to be written at can be found in source files). It should be noticed that scattering events by gas molecules (Rayleigh scattering) uses the Rayleigh phase function.
- Stop optical path after a given number of scattering events: if this option is set to 1, each optical path will be stopped after it reaches a

given number of scattering events. This number has to be specified in file “**data.in**”. If it is set to 0, the number of scattering events for a given optical path can be infinite.

+ **Spectral optimizations:**

- Radiative transfer computation for a part or for the whole infrared spectra: if this option is set to 1, radiative transfer will be computed for the whole spectral range data is available for. If it is set to 2, it is possible to compute radiative transfer for only a part of the whole spectral range. In this case, the limits of the spectral range the code has to use should be specified in file “**data.in**” (see “Minimum / maximum wavelenght”).
- Monochromatic NER and fluxes computation: this option tells explicitly the code to compute the NER matrix for each narrowband. This will require a lot of memory (RAM) to run. This option was introduced because of special analyses requirements.
- Use Planck intensity for ground and space or a user-specified input flux: it is possible to use either the Planck intensity at specified ground and space temperatures for emission from ground and space, or a user-specified intensity profile (as a function of narrowband). The purpose of this option is to allow the use of an input intensity that is different from the Planck intensity. For instance, make computations in the visible part of the spectrum, or computations with a laser emission (monochromatic) in a cold media. The code will return the following results, that are valid only in the case of a cold media (i.e. if radiation is emitted only by the ground, gas and space temperature must be null):
 1. “total_transm_refl.txt”: there are four columns in this file: the value of the total (spectrally integrated) transmittivity, i.e. the fraction of energy emitted by the ground that reaches space, its uncertainty, the total reflectivity, i.e. the fraction of energy emitted by the ground that is backscattered at ground level, and its uncertainty.
 2. “spectral_transm_refl.txt”: idem, for each specified spectral interval. The two first colums are the minimum and maximum values of wavelenght (in μm) of each spectral interval.

- Planck intensity taken at narrowband center of integrated over narrowband: it is possible to use the value of the Planck intensity taken at the center of each narrowband spectral interval (1) or the very accurate value of the Planck intensity integrated over each narrowband interval (2). In common cases, it does not introduce any difference because narrowband intervals are defined so that the Planck intensity does not change a lot over each interval. This option is meaningful when spectral intervals are not longer “narrowband” intervals.
- Spectral sampling optimization for nothing/NER/flux/previous computation : this option will determine which spectral sampling optimization method to use. Setting it to 1 will make the code compute a spectral sampling probability density function (pdf) based on analytic NER (these analytic results can only be computed within the hypothesis of a purely absorbing participating medium). This will result on a Monte-Carlo computation optimized for NER. Fluxes will be computed too, but they will be computed with much worse accuracy than NER. Setting this option to 2 will make the code compute a spectral sampling pdf based on analytic fluxes. In this case, the Monte-Carlo algorithm will compute NER less accurately than fluxes. In the case this option is set at 0, the spectral sampling pdf by the Monte-Carlo algorithm will be a uniform pdf: both NER and fluxes will be computed with the same accuracy. If this option is set to 3, the spectral pdf computed by the previous run will be used (KARINE will check that the atmospheric configuration has not changed so that the previous pdf can be used).

+ **Data interpolation**

- Interpolate spectroscopic data over a new GCM grid: input data file (see “**data.in**” file) comes with a vertical grid for which altitude and pressure levels are given. However, it is possible to use a different pressure grid from the default one, by setting this option to 1. The reason is that the user may want to compute radiative transfer results over a simplified pressure grid (for instance the GCM may not require a very high number of atmospheric layers). Instead of interpolating results over the new grid, this option will save computation time by instead interpolating the input data. The interpolation module is quickly described below: The first step is to compute an altitude grid attached

to the new pressure grid. Altitudes are exponentially interpolated between two known pressure levels. If the new ground pressure is higher than the default ground pressure, the corresponding altitude level will be negative. Negative altitudes will make the Monte-Carlo algorithm crash. This is why, in this case, all altitude levels have to be increased so that the ground altitude is set to zero. Note that this GCM grid is used for the subsequent radiative transfer simulation: final NER and ksi matrixes will therefore be given for the GCM grid. The next step is to linearly interpolate temperature and spectroscopic data over the new altitude levels (this is equivalent to interpolate the data exponentially over pressure levels). In the case the ground pressure is higher than the default ground pressure, the temperature at the bottom atmosphere may become higher than the default ground temperature. The ground temperature is therefore taken equal to this value.

- Conservation of absorbant quantities: if interpolation over a new pressure grid is required, setting this option to 1 will ensure absorbant quantities will remain the same as over the default grid. This option should be used with extreme care: in particular, it should not be used if ground pressures are different in the default and the new pressure grids. Altitude levels will be artificially modified so that absorbant quantities stick to given values.

General options

- Monte-Carlo computation: when activated (1), this option will enable the Monte-Carlo computation. If it is not activated (0), only the analytic calculation will be performed. This preliminary analytic calculation will always be performed; it is intended at estimating spectral NER and radiative budgets or fluxes (using various hypothesis, refer to the “spectral sampling optimisation” section) in order to optimize the spectral sampling during the Monte-Carlo computation.
- Display computation’s state: if this option is set to 1, the program will display on the xterm windows it was launch in different messages that indicate the progression of the computation. In order to not display these messages, this option has to be set at 0.
- Display error messages: the program tries to identify the source of errors that may happen and cause the computation to fail, and can display a message in the case such an error happens. These messages are not always relevant, but may at least give clues on what the error is. Setting this option at 1 will cause error messages to be displayed. Setting it at 0 will disable error messages display.
- Save spectral results: if this option is set to 1, the program will record (see “**data.in**” for associated file names) spectral results such as radiative budgets and their different contributions, for each narrowband. If this option is set to 0, spectral results will not be recorded, and only the values of radiative quantities integrated over the spectra will be saved.
- Backup intermediate results: if this option is set to 1, the program will periodically backup the complete state of the computation and all intermediate results on the hard drive (the period can be adjusted using the “number of records per layer” option below) in files “**arrays1.sav**” et “**arrays2.sav**”, which is necessary to resume an unvoluntarily interrupted computation ⁵ If this option is set to 0, intermediate states will not be recorded. This can be usefull for increasing total running times in the case of short computations.

⁵Whether the cause of this interruption is a power shutdown or someone who rebooted the system, judging it’s not worth waiting your at least 24h computation comes to an end.

- Automatic resume in the case an interruption has been detected: the code automatically detects if the previous run was interrupted or if it was successfully finished. If an interruption has been detected, and if this option is set to 1, intermediate results files will be used to resume the previous run. If no interruption has been detected, or if this option is set to 0, the code will launch a new computation.
- Number of records per layer: if the backup option is activated, the value of this option will set the number of backups that must be performed during the process of rays emission and raytracing for each atmospheric layer.
- Pseudo-random number generator: if this option is set to 1, the code will use the standard simple-precision pseudo-random number generator “urand” that KARINE was initially based upon. A full double-precision generator was subsequently added in order to increase the random generation period. This double-precision generator can be used by setting this option to 2.

Analytic computations options

- Compute $d(\text{NER})/d(T)$ (1:yes/0:no). KARINE can compute analytic sensitivities of NERs to the temperature at the center of each gas layer when performing analytic computations. Computation of $d(\text{NER})/d(T)$ requires a valid $d(k_a)/d(T)$ input data file (see “**data.in**”). Note: sensitivities computations are disabled in the “light” version of KARINE .
- Save $d(\text{NER})/d(T)$ matrices and A,C,D,E,F matrices when performing analytic sensitivities $d(\text{NER})/d(T)$ computations. Note: sensitivities computations are disabled in the “light” version of KARINE .
- Compute $d(\text{NER})/d(x)$ (1:yes/0:no). KARINE can also compute analytic sensitivities of NERs to chemical species concentrations. This will require a valid $d(k_a)/d(x)$ file. Sensitivities of NERs to several chemical species concentrations can be performed simultaneously (see program “**data/write_dkadx**” for details). Note: sensitivities computations are disabled in the “light” version of KARINE .
- Save $d(\text{NER})/d(x)$ matrices. Note: sensitivities computations are disabled in the “light” version of KARINE .

4.1.3 Variables in file “\$RAYDIR/includes/max.inc”

The success of a computation also depend on the values given to different variables declared in file “\$RAYDIR/includes/max.inc”. It may be necessary to modify these values in the case some errors happen.

- Diffmax: the value of this variable is used in the declaration of the array that records the history of each optical path; during each scattering event, the altitude this scattering occurred at and the path length (in meters) between the previous scattering event and this one will both be saved in this array. This array is reinitialized when the optical path crosses a layer boundary. As a consequence, its size must be greater than the number of scattering events that may occur in a single atmospheric layer. Unfortunately, when the number of scattering events is greater than “Diffmax”, the code generates an error messages and the computation stops (this is probably not the best strategy though). The error message explicitly states that the value of “Diffmax” should be increased before trying a new run.
- Nmax: the maximum number of atmospheric layers m . If you get an error “m=# while Nmax=#”, you may consider increasing the value of Nmax.
- Pmax: is used for intermediate exchanges arrays declaration. An error message warns the user when Pmax should be increased.
- NbmX: the maximum number of narrowbands N_b . If you get an error “Nb=# while NbmX=#”, you may consider increasing the value of NbmX.
- Nqmx: the maximum quadrature order N_q . If you get an error “Nq=# while Nqmx=#”, you may consider increasing the value of NbmX.
- PhMax: maximum number of discretization points for the “user defined” phase function written in file “**fonction.txt**”.
- Na: maximum number of discretization intervals for computing spectral sampling acceleration arrays. A typical value of 10 gives a good compromise between the CPU time saved by this acceleration method and the time actually required to use the acceleration method.

- `kmin`: value that will be given to absorption and scattering coefficients that would be found equal to zero in the input data file. This is only a security, so that there is no division by zero, and for some other numerical bugs that have been encountered.

Warnings:

- You can change any value in the “`$RAYDIR/includes/max.inc`” include file. Keep in mind there are a lot of arrays defined using these variables. Therefore, you may encounter memory issues at run time if those values are too high.
- Whenever a value is changed in “`$RAYDIR/includes/max.inc`”, `KARINE` has to be entirely recompiled, using command *make clean all*. If you do not do this, your executable may not work any longer, since you changed an include file and did not recompile all object files that use this include.

4.2 Running a computation

When the input data file has been successfully created, when the various parameters of files “`data.in`” and “`options.in`” have been set, and if the program has been successfully compiled, the user will be able to run a radiative transfer computation.

4.2.1 Launch the program

In order to run the program, the name of the executable file *'karine'* has to be entered on the command line. Alternatively, the command *'make run'* can be used to run a computation, or resuming a previously interrupted computation. As long as the user remains logged on the system, the computation will not be interrupted.

Now, given that atmospheric computations may be rather time consuming, especially when a good accuracy is required over the results, it is recommended to run the program in the batch, using command *nohup*. In order to run the program as a batch process, the following commands may be used: *'nohup karine.exe &'* or *'nohup make run &'*. The user may then terminate its linux session.

Advice : before running at atmospheric radiative transfer computation, a number of points should be carefully checked:

- The input data file should have been created, with the right format, and it should contain no 'Inf' or 'NaN' values. Zeros are generally not very well managed. Using “plot” for visualizing data may help checking the input data file.
- The program should be properly compiled. The amount of RAM should be checked, as should be the fact that enough memory can be allocated to a single job (more especially under the **Cygwin** system).
- Every parameter in files “**data.in**” and “**options.in**”, and possibly in file “**max.inc**”, should have correct values. Moreover, they should not conflict with each other. The program is designed to handle such conflicts, or at least display an error message if one occurs.
- Check that enough space disk is free when performing sensitivities computations. KARINE will display the amount of disk needed for sensitivities result files record. Note: sensitivities computations are disabled in the “light” version of KARINE .

4.2.2 Messages displayed during a successful run

In the case everything runs correctly, a number of messages should be displayed to inform the user about the progress of the run:

Messages relative to the beginning of the run

- When the previous computation was successful, or when it was interrupted and the user decided to dump it, a message indicating the beginning of a new run should be displayed.
- If the previous run was interrupted, and it was specified to resume it, a message indicating the point it is resumed at should be displayed.

Data analysis

- The minimum and maximum wavelenght of the spectral interval the radiative transfer will be computed for are displayed.

- The number of narrowbands in this interval, and for which data is available, is displayed.

NER or fluxes analytic estimation

- In the case it was chosen to use a uniform spectral sampling pdf, (see “options.in” in section 4.1.2), a message displays this choice.
- In the case it was decided to use a non uniform spectral pdf, the program will display the rate the spectral optimisation is computed at, then two messages will indicate the calculation of spectral sampling acceleration arrays, and intermediate results record.
- In the case it was chosen to use a pre-computed spectral pdf, a message will remind this choice.

Looking for known errors: next, the program tries to identify different sources of potential errors. This mainly consists in checking that parameters in files **data.in** and **options.in** are correct and do not conflict.

Computation progress: during the run, the program will display the computation’s progress, before the first ray of each atmospheric layer is emitted. See section 4.2.3 below for more information about the computation progress.

Post-processing: when the Monte-Carlo algorithm has finished, the program will use its results to compute definitive radiative transfer quantities (NER, fluxes, radiative budgets) and record these results in the appropriate output files.

4.2.3 Checking the computation progress

A message is displayed before the first ray is emitted from any boundary or atmospheric layer. This lets the user know exactly what is the state of the program at any time. However, when the program is run as a batch job and thus progress messages are no longer displayed, or when it is necessary to check more thoroughly the progress of the code, some files may be visualized for more detailed information:

- “**\$RAYDIR/state**”: displays the layer the computation is running in, and what is the number of the last ray a backup was made for. These information are also available as a percentage of the computation load for the current layer, and as a percentage of the total computation load. In addition, this file shows the estimated date and time the end of the run has been envisaged at. This estimation is based on the CPU time that was required to achieve the computation until the last backup, and the total computation load that has to be achieved.
- “**\$RAYDIR/time**”: displays the time (in hours:minutes:seconds) elapsed between two successive backups, for all backups that have been made, and the estimated time until next backup.

4.2.4 Tips

- You can pause the execution of KARINE , using linux command-line tools. While running, find KARINE 's pid (using 'top' or 'ps -aux | grep karine'). #pid refers to the pid of the process. Now, you can pause the process using 'kill -19 #pid' and resume it using 'kill -18 #pid'. KARINE displays a message when it is paused and resumed.
- Easter egg: if you survived reading this documentation file, try using the following command while KARINE is running, provided that you know its pid: 'kill -10 #pid'

5 Results visualization

The 'plot' script that is provided in the “**plot**” folder can be used for data and results visualization. This script is used as a command-line tool, which means for instance: 'plot ka gas' will draw a map of the absorption coefficient values, as a function of gas layer index and narrowband spectral interval index. Similarly, 'plot ner' will display the integrated NER matrix. 'plot' use Grads software to displays color maps. Therefore, you must have Grads installed on your system before using 'plot'.

Plot needs some information before it can be used: you need to indicate where is the input data file that was used, and where are located the folder that contain the results you want to display. 'plot' can actually use two different folder path names, so that it can compute and display differences between two given results sets. Follow instructions in file “graphi” for path names specifications.

You will be able to display a lot of data and results using the 'plot' script. Please refer to the README file for available commands. Here is a quick description of what 'plot' is able to display:

5.1 Data plots

- Absorption, scattering, and total extinction coefficients.
- Asymetry parameters, single-scattering albedoes.
- Distinction between gas, aerosols and clouds parameters.
- Mean overlap parameter (for gas only); possibility to get it for individual spectral intervals.
- Pressure and temperature profiles.

5.2 Results plots

- The Monte-Carlo integrated NER matrix, its uncertainties, effects of scattering on the integrated NER matrix, and differences between two NER matrices.
- The analytic integrated NER matrix, and differences between two analytic NER matrices.

- ξ coefficients and their differences.
- The analytic NER matrix for every narrowband interval, or for a given number of narrowband intervals. Effects of scattering and differences.
- Radiative budgets, and their full decomposition: gas-space, gas-ground and gas-gas components. Effects of scattering and differences.
- Fluxes: integrated, for a given spectral interval, or for a number of narrowband intervals, or at a specified altitude. Effects of scattering, differences. Monte-Carlo and analytic.
- Standard deviations over every Monte-Carlo result.

5.3 Tips and advices

- 'plot' integrates and automatic detection of visualization results. Which means that in most cases, the user should not worry about scales, the number of colors in a map, or the numeric levels associated to these colors. However, it is possible to specify some options to the script: increase or decrease the number of colors (and associated visualization levels), increase or decrease levels themselves. Every color map can be displayed in black and white for a print-friendly result. NERs, radiative budgets and ξ coefficients can be visualized for both standard and recombined grids (see section 3). Please refer to the README file for details.
- The script will look for basic data (number of atmospheric layers, number of narrowbands and quadrature order, etc) in the specified input data file. Specified results sets must be compatible with this data file (at least one result set should have been obtained using the specified data file).
- Of course, any bug or suggestion can be submitted to the author: eymet@energetique.ups-tlse.fr

6 Frequently asked questions

This user guide is probably not complete as it is. Users who would not find answers to their questions in this section are welcomed to submit them to the author by e-mail (eymet@energetique.ups-tlse.fr). Answers will be added to this last section.

Numerical uncertainties over my results are too high. How can I reduce them ?

All you have to do is to increase the number of rays optical pathes random generations for the Monte-Carlo algorithm (see section 4.1.1). Remind that decreasing numerical uncertainties by a factor 10 requires to increase the number of random generations (and the CPU time) by a factor 100. As such, it is not always possible to decrease numerical uncertainties.

**When I use command "plot", I get an error of this kind:
/usr/bin/ld: unrecognized option '-stack'
/usr/bin/ld: use the -help option for usage information
collect2: ld returned 1 exit status**

Look for compilation scripts (most probably f0, f1,etc) where should reside the following compilation option: `-Wl,-stack,0x4000000`. All you need is to remove or to comment this option. It was added for compiling programs that require heavy memory loads under Cygwin.

The program starts, it does a spectral optimization, says no errors has been found and therefore the computation will start, the initial random seed is set, but it's definitely stuck there. What is going on ?

After setting the random seed, the program will initialize some arrays that are needed by the Monte-Carlo algorithm. If there is no error message, but the computation does not start at this stage (and you can see your process alive, but taking no CPU time at all), it is because you do not have enough RAM on your system: the initialization will take forever because the arrays have been set to SWAP memory (on the hard drive). You might decrease the values of parameters 'Nmax', 'NbmX' and 'Nqmx' (that deter-

mine arrays sizes) to the minimum required for your computation, and then recompile the whole program (using “make clean all”). Remember that for most configurations, you will need at least 512Mb of RAM.

You can alternatively increase the amount of RAM on your system OR use a machine with more RAM OR use a PhD student whose personal laptop has enough RAM. Finally, if all above solutions failed, you may try using the “light” version of KARINE , which has been optimized for low memory consumption. See details about the “light” version below.

When I run the program, I only get a message “process stopped”

This can happen on some systems. This is because the process (karine) requires more memory than available on the system (although the system should allow the process to run on swap, but you would end up with the previous case). Use above recommendations.

I get a message “segmentation fault”

This can happen for two reasons: once again, if you do not have enough RAM (use above advices) or because of a bad array declaration. In the latter case (or if you don’t know what is going on), please submit the bug to the author.

Why a “light” version of KARINE ?”

Using the full version of the code may require tremendous amounts of RAM (i.e. more than 4Gb) for heavy atmospheric configurations (more than 300 narrowbands and more than 100 gas layers). This is all because the declaration of huge arrays that are required for sensitivities computations.

This is the reason why a “light” version of KARINE is developed in parallel with the full version. The only difference between the full and the light versions is that sensitivities computations have been completely removed in the light version. All options relative to sensitivities are therefore irrelevant in the light version. The light version code source is automatically generated from the full version source code. Every code modification (in the full version, which is the only one that is developed) is therefore propagated in the light version.

If you encounter memory-related compilation issues with the full version of KARINE , you may try the light version.

References

- [1] M. Cherkaoui, J. L. Dufresne, R. Fournier, and J. Y. Grandpeix. Radiative net exchange formulation within 1D gaz enclosures with reflective surfaces. *ASME Journal of Heat Transfer*, 120:275–278, 1998.
- [2] M. Cherkaoui, J. L. Dufresne, R. Fournier, J. Y. Grandpeix, and A. Lahllec. Monte-Carlo simulation of radiation in gases with a narrow-band model and a net-exchange formulation. *ASME Journal of Heat Transfer*, 118:401–407, 1996.
- [3] J. L. Dufresne, R. Fournier, and J. Y. Grandpeix. Méthode de Monte-Carlo par échanges pour le calcul des bilans radiatifs au sein d’une cavité 2D remplie de gaz. *Compte-rendu de l’Académie des Sciences, Paris*, 326 Série II b:33–38, 1998.
- [4] L. Tessé, F. Dupoirieux, B. Zamuner, and J. Taine. Radiative transfer in real gases using reciprocal and forward Monte-Carlo Methods and a correlated-k approach. *International Journal of Heat and Mass Transfer*, 3, issue 13:2797–2814, 2002.
- [5] V. Eymet, J.L. Dufresne, R. Fournier, and S. Blanco. A boundary-based net exchange Monte-Carlo Method for absorbing and scattering thick medium. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 95:27–46, 2005.
- [6] V. Eymet. *Simulation Monte-Carlo et analyse en Puissances Nettes Echangées des transferts radiatifs infrarouges avec diffusion : vers une paramétrisation dans un modèle de Circulation Générale atmosphérique*. PhD thesis, Thèse de l’Université Paul Sabatier, 2003.
- [7] J.M. Hammersley and D.C. Handscomb. *Monte-Carlo methods*. John Wiley, New York, 1964.
- [8] J.R. Howell. The Monte-Carlo Method in radiative heat transfer. *Journal of Heat Transfer*, 120:547–560, 1998.
- [9] J.R. Howell. Application of Monte-Carlo to heat transfer problems. *Advances in Heat Transfer*, 5:1–54, 1969.

- [10] A De Lataillade, J. L. Dufresne, M. El Hafi, V. Eymet, and R. Fournier. A net exchange Monte-Carlo approach to radiation in optically thick systems. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 74:563–584, 2002.