

KDISTRIBUTION

version 1.1.1

User Manual

26 septembre 2016

V. Eymet, Laboratoire PLAsmas et Conversion de l'Energie, (LAPLACE), UMR 5213, bat. 3R1,
Université Paul Sabatier, 31062 Toulouse, France

Table des matières

1	Description of the code	3
1.1	Production of k-distribution data sets	3
1.2	Hypothesis	4
1.3	Parallel computing	4
2	Installation and prerequisites	5
2.1	First thing to do	5
2.2	Tweaking the “Makefile”	5
2.3	Compilation	5
2.4	Installation of MPICH	5
3	Using the code	7
3.1	User input files	7
3.1.1	The “data.in” file	7
3.1.2	The “options.in” file	7
3.2	High-resolution spectrum input files	7
3.2.1	Narrowband intervals	8
3.3	Running KDISTRIBUTION using MPICH	8
4	Results	9
4.1	Results files	9
4.2	More output files	9
5	The fun stuff	10
5.1	Code license	10
5.2	Questions and Answers	10

1 Description of the code

1.1 Production of k-distribution data sets

KDISTRIBUTION is a computer code that will produce k-distribution data sets from high-resolution spectra. A high-resolution spectrum consists in a list of absorption coefficients (preferably in m^{-1}) and their associated values of wavenumber (in cm^{-1}).

The idea of using k-distribution data sets is to consider the statistical distribution of the absorption coefficient within a narrowband spectral interval. Within an homogeneous media, only the absorption coefficient will depend on frequency within a narrowband interval [1]. It is therefore possible, within each narrowband interval $[\nu_1, \nu_2]$, to replace the dependency in frequency (or wavenumber ν) by a dependence in absorption coefficient $k_{a,\nu}$:

$$\frac{1}{\nu_2 - \nu_1} \int_{\nu_1}^{\nu_2} A(\nu) d\nu = \int_0^\infty A(k_a) f(k_a) dk_a \quad (1)$$

with A any radiative quantity, and f the distribution function of absorption coefficients. Integrals over frequency can be reformulated as integrals over k_a . Let us take the example of the average transmittivity over a narrowband interval $[\nu_1, \nu_2]$:

$$\bar{\tau}(l) = \frac{1}{\nu_2 - \nu_1} \int_{\nu_1}^{\nu_2} \exp(-k_{a,\nu} l) d\nu = \int_0^\infty f(k_a) \exp(-k_a l) dk_a \quad (2)$$

In other words, $\bar{\tau}(l)$ is the Laplace transform of function f :

$$\bar{\tau}(l) = \mathcal{L}(f(k_a)) \quad (3)$$

One way of choosing function f is to use a statistical model of average transmittivity. Domoto used the Malkmus model [2] that uses a two-parameters statistical model of the average transmittivity. Function $f(k_a) = \mathcal{L}^{-1}(\bar{\tau}(l))$, also called the inverse transmittivity function, can then be formulated analytically when using the Malkmus model. It was shown [3] that in this case, function $f(k_a)$ is an inverse Gaussian function, which allowed an optimized sampling of absorption coefficients in a Monte-Carlo algorithm.¹

Another way to choosing function f is to define the cumulated function g as :

$$g(k_a) = \int_0^{k_a} f(k) dk \quad (4)$$

This function is defined over the $[0, 1]$ interval, and it is monotonous and increasing. It is then possible to inverse it. The integral of any quantity A can then be reformulated as :

$$\frac{1}{\nu_2 - \nu_1} \int_{\nu_1}^{\nu_2} A(\nu) d\nu = \int_0^\infty A(k_a) f(k_a) dk_a = \int_0^1 A(k_a(g)) dg \quad (5)$$

Because function $k_a(g)$ is an increasing function, it is possible to use a quadrature to approximate the result of any frequential integral :

$$\frac{1}{\nu_2 - \nu_1} \int_{\nu_1}^{\nu_2} A(\nu) d\nu = \int_0^1 A(k_a(g)) dg \approx \sum_{i=0}^{N_q} A(k(g_i)) \omega_i \quad (6)$$

where N_q is the quadrature order. Gauss quadratures are generally used, since they represent well the distribution function $k_a(g)$ in the limit $g \rightarrow 1$.

The k-distribution spectral model (formulated in terms of discrete values of k) is therefore giving a possibility to compute any integral over frequency as the pondered sum of N_q gray computation results.

The purpose of the present code is to compute the values of $k(g_i)$ and pondering weights ω_i , for a given value of the quadrature order N_q .

1. It is then also possible to obtain an analytic expression of the cumulative of $f(k_a)$; considering this cumulative is also the inverse Laplace transform of the transmittivity, it is then possible to obtain k-distribution data provided that transmittivities are known : this is what can be achieved using the EM2C SNB model that provides transmittivities of CO/CO₂/H₂O mixtures over the infrared spectral range (there is a code for that, contact me).

1.2 Hypothesis

KDISTRIBUTION will take as input the discretized absorption spectra of a gas mixture : it consists in N values of wavenumber ν_i and absorption coefficient $k_a(\nu_i)$, as shown in figure 1.

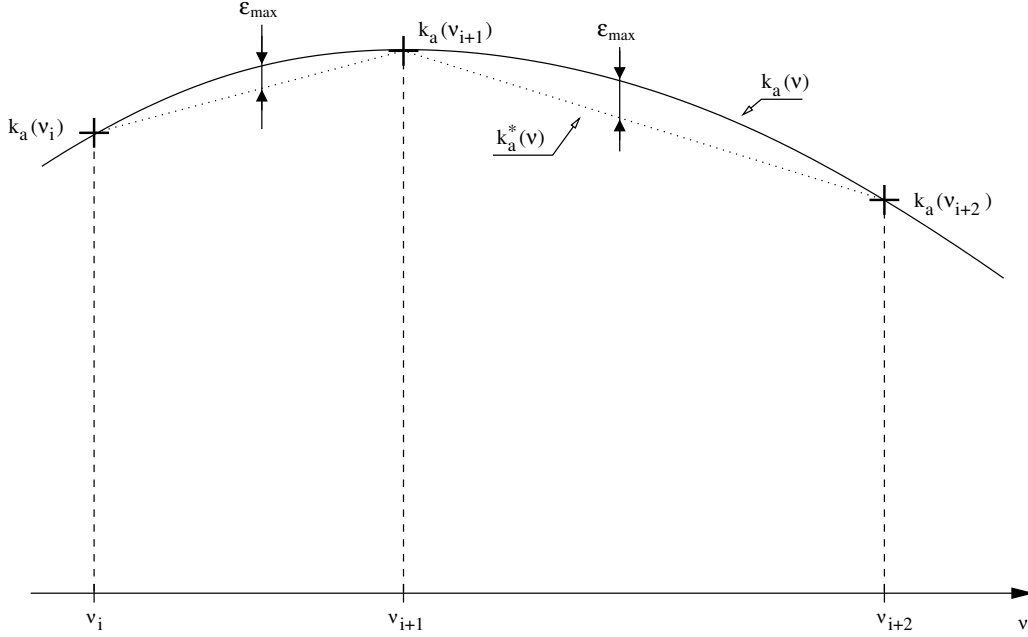


FIGURE 1 – Schematic representation of the input $k_a(\nu_i)$ spectrum

$k_a(\nu)$ spectrum may be computed, for instance, by a tool such as the KSPECTRUM computer code. See relevant documentation file for more information.

The main hypothesis that is performed by KDISTRIBUTION is that the absorption coefficient is linear between two consecutive values of $k_a(\nu_i)$ (see fig. 1). KSPECTRUM ensures that the relative difference between the real profile $k_a(\nu)$ and the linearly interpolated profile $k_a^*(\nu)$ never exceeds a user-defined value ϵ_2 . KSPECTRUM also ensures that the relative error over each value of $k_a(\nu_i)$ never exceeds a user-defined value ϵ_1 . Values of ϵ_1 and ϵ_2 are used by KDISTRIBUTION in order to compute uncertainties over the values of $k(g_i)$.

1.3 Parallel computing

KDISTRIBUTION can compute full k-distribution data sets. This means the computation of $k(g_i)$ and ω_i coefficients, for every narrowband spectral interval of the IR range, for several HR spectra simultaneously. Although this is feasible on any average PC, the computation is faster on a cluster of PCs.

This code uses MPICH instructions for running on multiprocessor machines / cluster of machines. The total computation time will be divided by the number of physical processors the code is running on.

You will need MPICH installed before using this code. Refer to section 2.4 for instructions about installing MPICH and creating a cluster of machines.

2 Installation and prerequisites

2.1 First thing to do

is to use script “install.bash” that is provided with the archive. This script should be placed in the same directory than the provided zipped archive. Use command “./install.bash” in order to run it. This script will :

- Untar the archive, if the corresponding version of KDISTRIBUTION is not already installed
- Go into the “data” directory, and make links to KSPECTRUM input files “composition.in” and “narrowbands.in” and also a link to KSPECTRUM output directory as “data/hires_spectrum” (see section 3.2).
- Erase the zipped archive, and erase itself.

Then you can go into the newly created directory (named “kdistributionX.X.X” with “X.X.X” the version number) to continue the installation.

2.2 Tweaking the “Makefile”

Before compiling, you will have to find out what compilation options are right for your compiler, and your machine. Open the “Makefile” file, and look at variables “FOR”, “ARCH” and “OPT”. Variable “FOR” is used to specify your fortran 77 compiler. As KDISTRIBUTION uses MPICH, you will most likely use the “mpif77” compilation command, that has been installed along with MPICH.

Variable “ARCH” is used to specify machine architecture. “-m486” is probably a good choice for a PC running a 32bits linux. Use the documentation of your fortran compiler to find out what architecture option you can use.

Variable “OPT” is used to specify code optimization options. The default options should be enough. Please note that you definitely must use option “-Wno-globals” for compiling parallel code. Please note also that the results may depend on optimization options that are used. You will have to check the validity of your results (see section 4).

You might also want to set variable “DEBUG” (look for its definition in the file). You can expect faster execution times if you leave it empty.

2.3 Compilation

Once you checked compilation options and array size definitions, you can use the following command in order to compile the executable file :

```
> make all
```

If compilation fails, use the compiler error message to determine what went wrong. The most probable error causes are : a bad definition of architecture compilation option, or an inappropriate value in code optimization options.

If you ever need to modify the source files (in directory “source”), you can quickly recompile the code using “make all” again. This will only recompile the modified source files, and link objects files in order to produce the new executable file.

If you have to modify the value of any variable defined in includes files (directory “includes”), you will have to recompile the whole code from scratch. Use the following command to erase all objects files, and then recompile them properly :

```
> make clean all
```

Odd errors may happen if you change an include file and then recompile using only the “make all” command (old value of the modified variable will remain in the unchanged object files).

2.4 Installation of MPICH

If you do not already have MPICH installed on the machine / group of machines you want to run KDISTRIBUTION on, you will first have to download mpich2 from <http://www.mcs.anl.gov/research/projects/mpich2> make sure you download version 1.0.7. Next, untar the downloaded archive, and install it on every system that will be part of your cluster :

```
> ./configure --prefix=/path/to/installation/directory
> make
> make install
```

Before running the MPD daemon, you must create a “.mpd.conf” file in your home folder :

```
> echo secretword=[secretword] » /.mpd.conf
> chmod 600 .mpd.conf
```

using any “secretword”.

Next, you will need to be able to connect via ssh to every other machine of your cluster, with no password request. For this, you must first create a DSA key :

```
> ssh-keygen -t dsa
```

leaving all fields blank (use the “enter” key to answer each question). Then you will have to add this DSA key to the list of authorized keys :

```
> cd .ssh
> cat id_dsa.pub » authorized_keys
```

Finally, create the list of machines that belong to your cluster. This list must reside within the “mpd.host” file on your home folder. Each line must contain the name of the machine, by order of availability :

```
[host1].[domain]
[host2].[domain]
[host3].[domain]
etc
```

You can then try to run the MPD daemon :

```
> mpdboot -n [#]
```

with [#] the number of hosts you want to run MPD on (typically, the number of machines in your cluster). If you encounter no error, you can use command “mpdtrace” to check the number of hosts the MPD daemon is running on. This should give you the list of machines in your cluster.

3 Using the code

3.1 User input files

There are two files in the main KDISTRIBUTION folder : “data.in” and “options.in”. The “options.in” file is where the user can chose between several options, for each algorithm. The “data.in” file is where the user can specify the numerical values of a number of variables the code will use. Below is a description of these two files.

3.1.1 The “data.in” file

The user has to specify indexes of the first and the last HR spectrum file the k-distribution data set has to be computed for. Input data files have to be found in the “data/hires_spectra” folder (see full description of the contents of folder “data/hires_spectra” in section 3.2).

3.1.2 The “options.in” file

- Quadrature type : the user should specify what quadrature will be used in order to generate the k-distribution data set. The default value is a Legendre quadrature. Gaussian quadratures (Legendre, Legendre-Radau and Legendre-Lobatto) only have been tested, and may therefore be used.
- Quadrature order : common quadrature orders are 8 and 16-points quadratures. But you can use any other value.
- First g abscissa imposed : in the case a Radau or Lobatto quadrature is required, the user has to provide the value of g_1 , the first g_i abscissa. It is commonly a value very close to zero.
- Second g abscissa : in the case a Lobatto quadrature is required, the last abscissa g_{N_q} has to be provided. It is commonly a value very close to 1.
- Number of intervals for acceleration arrays : an acceleration method is used for inversion of cumulative function $g(k_a)$. The size of arrays used by this method should be specified. A typical value of 1000 should be fine.
- Maximum relative error over g : the user can chose the maximum relative error over values of g_i ; this error is induced by the inversion procedure of the cumulative function. The default value is 10^{-3} (0.1%).
- Type of spline used for interpolation : the user must choose whether a linear or a cubic spline interpolation will be performed between discrete values of $k(\nu)$. Both will provide the same k-distribution data sets when the spectral discretization is very accurate. Using a cubic spline interpolation gives the possibility to use coarser spectral grids, but remember errors may happen, since performing a cubic spline interpolation requires to solve a tridiagonal linear system, and then inverting each spline function (third-degree polynomia, which is not error-proof).

3.2 High-resolution spectrum input files

The program will take as input the high-resolution absorption spectra produced by KSPECTRUM. These files have to be found under the “data/hires_spectra” directory. A symbolic link to any directory that contains result files from KSPECTRUM, and named “hires_spectra” under the “data” directory is enough.

From version 1.10, KDISTRIBUTION does not longer need a file that describes the number of HR spectrum files and the thermodynamic conditions (pressure, temperature, number of species and their concentrations) for each spectrum. The code is able to identify by itself HR spectrum files that reside within the “hires_spectra” folder. It is mandatory that these HR spectrum files have been produced by the KSPECTRUM code², since information given in file headers is used to identify the thermodynamic conditions used for each HR spectrum.

2. KSPECTRUM output files use the following format : first, a header that provides full information about the thermodynamic conditions used for computing the HR spectrum (total pressure and temperature, number of molecular species, their name and concentration) ; the remaining of the file contains 3-columns : ν in cm^{-1} , σ in $cm^2/molecule$ and k_a in m^{-1}).

Pay attention to the fact that KDISTRIBUTION will use every HR spectrum file found in the “hires_spectra” folder : it is therefore important to carefully check the contents of this folder. Any file that is not a KSPECTRUM output file will be ignored. Of course, symbolic links can be used instead of manually duplicating each HR spectrum file required for the k-distribution computation.

KDISTRIBUTION will also look for file “calculation_info.txt” file in directory “data/hires_spectra”. This is an output file from KSPECTRUM, that contains values of ϵ_1 and ϵ_2 (used for producing uncertainties over values of k). If this file is not found, uncertainties will be set to zero.

3.2.1 Narrowband intervals

have to be provided in the “data/narrowbands.in” file. This file must contain the number of narrowband intervals, and, for each interval, its lower and upper limits, in terms of wavenumber (in cm^{-1}). A link to the “narrowbands.in” file of KSPECTRUM into the “data” directory is fine.

3.3 Running KDISTRIBUTION using MPICH

Once everything is installed and the executable file “kspectrum.exe” file has been compiled, you can try to run a computation.

Use the following command to run the code :

```
> mpirun -np [#] kdistribution.exe
```

with [#] the number of processes that have to run.

Because communication times are small compared to computation times in KDISTRIBUTION , it is a good idea to chose a number of processes equal to the number of (physical) processors of your cluster, plus one. One process, the master process, is dispatching computational loads to every other processes (slave processes), and gathering results from them. It does not require any significant CPU time, therefore it is OK to have a number of slave processes equal to the number of processors, so that each slave process can use a processor (or each processor will have only one slave process running on it).

In practice, if your cluster is composed of n processors, you can use :

```
> mpirun -np n+1 kspectrum.exe
```


4 Results

4.1 Results files

KDISTRIBUTION will produce results files in the “results” directory. There are two main output files, named “kdist.txt” and “delta_kdist.txt”.

- “results/kdist.txt” contains the actual k-distribution data set. The file first provides the thermodynamic conditions of each HR spectrum file that was used to compute k-distribution data. Next, quadrature weights ω_i are given. Then, for each HR spectrum, the quadrature points $k(g_i)$ are given for each narrowband spectral interval, along with the relative error between the transmittivity of a 1 meter gas column, computed using the input high-resolution spectrum and using the output k-distribution $k(g_i)$ and ω_i coefficients.
- “results/kdist_***.txt” are similar to the main “kdist.txt” file, but the information that is contained in these files is provided for each HR spectrum used in the computation.
- “results/delta_kdist.txt” contains the values of the N_q quadrature weights ω_i , and for each HR spectrum, the absolute uncertainties $\Delta k(g_i)$ over the N_q $k(g_i)$ coefficients, for each narrowband interval.
- “results/delta_kdist_***.txt” : these are similar to the “delta_kdist.txt” file, for each HR spectrum used.

4.2 More output files

Additionally, the user can check the information gathered by the code into file “results/quadrature_results.txt”. It contains values of g_i abscissas and ω_i pondering weights. The sum of pondering weights is also given : $\sum_{i=1}^{N_q} \omega_i = 1$. If it is not the case, something went wrong. If pondering weights are not symmetrical (i.e. if $\omega_1 \neq \omega_{N_q}$, $\omega_2 \neq \omega_{N_q-1}$, etc. then the k-distribution results are not coherent. This is probably due to wrong optimization options at compilation time (see section 2). Finally, this file contains values of $\sum_{i=1}^{N_q} \omega_i g_i^n$ for $n \in [1, 2 * N_q]$. This sum should be equal to $\frac{1}{1+n}$. Values of the sum and of $\frac{1}{1+n}$ are given, along with the relative difference between those values. Relative differences should be small ($< 10^{-6}$), otherwise the k-distribution data is not valid.

5 The fun stuff

5.1 Code license

KDISTRIBUTION is distributed under the General Public License. See the COPYING file within the main directory. This means that you can use / redistribute part of all the source code in your own source code, provided that your source code is also distributed under the GPL. However, since KDISTRIBUTION is a very basic code that everybody is able to write, I will not sue you if you manage to compile it under windows and sell it.

5.2 Questions and Answers

I can't manage to compile the source code / I don't understand compilation error messages.

There should be not problems using the gfortran compiler and mpich2 (or openmpi). Send me an email with a description of the tools you are using, on what hardware/system, and what error messages you get : v.eymet@gmail.com

Compilation was successful, but the code crashes.

A real crash? It does not run at all? Are you sure you did not modify the source code or include files? Well, then maybe I did an error. Probably compilation options are wrong for your system. Tell me what error message you get.

The sum of pondering weights is not 1

That should not occur. This is interesting. Please send me an email to let me know.

Quadrature results are incoherent when I look into the “results/quadrature_results.txt” file.

While the sum of quadrature weights is 1, these weights can be non-symmetrical. This is most probably due to bad compilation options (optimization options). Tell me what options you used, on what system, and maybe I will be able to help.

Références

- [1] C.F. Bohren and D.R. Huffman. *Absorption and scattering of light by small particles*. Wiley, New York, 1983.
- [2] G.A. Domoto. Frequency integration for radiative transfer problems involving homogeneous non-gray gases : the inverse transmission function. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 14 :935–942, 1973.
- [3] J.-L. Dufresne, R. Fournier, and J.-Y. Grandpeix. Inverse gaussian k-distributions. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 61 n4 :433–441, 1999.