The Astrophysical Journal Supplement Series, 253:30 (28pp), 2021 March https://doi.org/10.3847/1538-4365/abd773

HELIOS-K 2.0 Opacity Calculator and Open-source Opacity Database for Exoplanetary Atmospheres

Simon L. Grimm[1], Matej Malik[2], Daniel Kitzmann[1], Andrea Guzmán-Mesa[1], H. Jens Hoeijmakers[1], Chloe Fisher[1,8], João M. Mendonça[3], Sergey N. Yurchenko[4], Jonathan Tennyson[4], Fabien Alesina[5], Nicolas Buchschacher[5], Julien Burnier[5], Damien Segransan[5], Robert L. Kurucz[6], and Kevin Heng[1,7]

[1] University of Bern, Center for Space and Habitability, Gesellschaftsstrasse 6, CH-3012, Bern, Switzerland; simon.grimm@csh.unibe.ch, kevin.heng@csh.unibe.ch
[2] Department of Astronomy, University of Maryland, College Park, MD 20742, USA
[3] National Space Institute, Technical University of Denmark, Elektrovej, DK-2800, Kgs. Lyngby, Denmark
[4] Department of Physics and Astronomy, University College London, London WC1E 6BT, UK
[5] University of Geneva, Department of Astronomy, CH-1290 Versoix, Switzerland
[6] Center for Astrophysics |Harvard & Smithsonian, Cambridge, MA 02138, USA
[7] University of Warwick, Department of Physics, Astronomy and Astrophysics Group, Coventry CV4 7AL, UK
[8] University of Bern International 2021 PhD Fellowship, Switzerland

Received 2020 October 6; revised 2020 December 14; accepted 2020 December 28; published 2021 March 12

Abstract

Computing and using opacities is a key part of modeling and interpreting data of exoplanetary atmospheres. Since the underlying spectroscopic line lists are constantly expanding and currently include up to $\sim 10^{10}$–10^{11} transition lines, the opacity calculator codes need to become more powerful. Here we present major upgrades to the HELIOS-K GPU-accelerated opacity calculator and describe the necessary steps to process large line lists within a reasonable amount of time. Besides performance improvements, we include more capabilities and present a toolbox for handling different atomic and molecular data sets, from downloading and preprocessing the data to performing the opacity calculations in a user-friendly way. HELIOS-K supports line lists from ExoMol, HITRAN, HITEMP, NIST, Kurucz, and VALD3. By matching the resolution of 0.1 cm^{-1} and cutting length of 25 cm^{-1} used by the ExoCross code for timing performance (251 s excluding data read-in time), HELIOS-K can process the ExoMol BT2 water line list in 12.5 s. Using a resolution of 0.01 cm^{-1}, it takes 45 s, equivalent to about 10^7 lines s^{-1}. As a wavenumber resolution of 0.01 cm^{-1} suffices for most exoplanetary atmosphere spectroscopic calculations, we adopt this resolution in calculating opacity functions for several hundred atomic and molecular species and make them freely available on the open-access DACE database. For the opacity calculations of the database, we use a cutting length of 100 cm^{-1} for molecules and no cutting length for atoms. Our opacities are available for downloading from https://dace.unige.ch/opacityDatabase and may be visualized using https://dace.unige.ch/opacity.

Unified Astronomy Thesaurus concepts: Exoplanet atmospheres (487)

1. Introduction

Opacities[9] are the cross sections per unit mass of atoms, ions, and molecules. They may be measured in the laboratory or calculated from first principles (by solving the Schrödinger equation) and are indispensable in the analysis of spectra across multiple subfields of astronomy and astrophysics. In the study of exoplanetary atmospheres, opacities are used in a number of fundamental ways: in one-dimensional radiative transfer calculations, three-dimensional general circulation models, and atmospheric retrievals.

Generally, the opacity of an atom, ion, or molecule depends on temperature, pressure, and wavenumber or wavelength. The number of spectral lines increases exponentially with temperature. For example, one only needs to compute $\sim 10^5$ lines of water at room temperature (because the weaker lines are not activated) but $\gtrsim 10^9$ lines at $\sim$1000 K. Furthermore, since models typically require hundreds, if not thousands, of combinations of temperature and pressure—as well as multiple chemical species—this problem rapidly becomes formidable.

Solving this computational problem requires the judicious design of software and its careful interfacing with hardware, including efficient memory management. In Grimm & Heng (2015), we took the first major steps toward producing an ultrafast, open-source opacity calculator named HELIOS-K (https://github.com/exoclime/HELIOS-K), which is accelerated by the use of graphics processing units (GPUs). In the current study, we report major upgrades to HELIOS-K that allow for further order-of-magnitude increases in computational speed.

As a service to the community and to promote scientific reproducibility and transparency, the computed opacities are publicly available on https://dace.unige.ch/opacityDatabase and https://chaldene.unibe.ch/. The former offers the possibility of obtaining customized opacities across user-specified arrays of temperature and pressure. The opacities may be visualized on the Swiss-run DACE database via https://dace.unige.ch/opacity, as we will describe. We note that the ExoMolOP database provides publicly available opacities as well (Chubb et al. 2021).

2. Methodology

The basic methodology of the HELIOS-K opacity calculator was published in Grimm & Heng (2015). The updates on the core methodology include a mostly new parallelization scheme and faster memory handling, as well as new features of the code. We summarize the supported line profiles in Section 2.1

[9] Specifically, this term is used to describe specific forms of the cross section per unit mass involving photons/light/radiation and not electron or heavy-particle collisions.

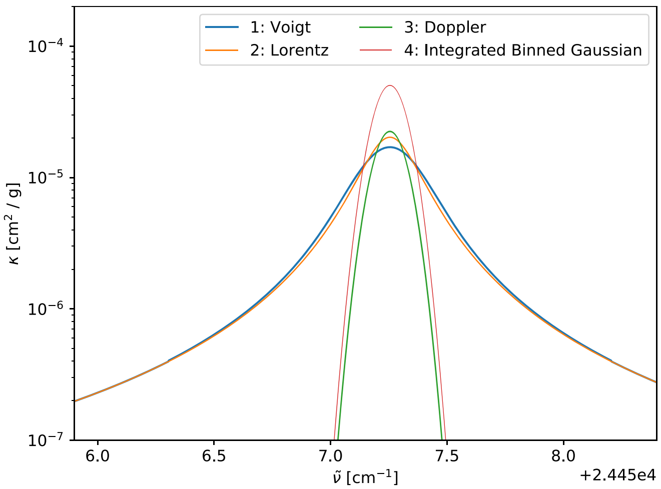


Figure 1. Available line profiles from HELIOS-K: Voigt, Lorentzian, Doppler, and the binned Gaussian integrated cross section. The Voigt profile is numerically the most expensive to calculate.

and the background theory in Section 2.2, describe the difficult aspects of the usage of GPUs in Section 2.3, and report the improvements and updates of HELIOS-K in Section 2.4. In Section 2.5, we describe in detail the handling of the different line list databases.

2.1. Line Profiles

In most applications, molecular or atomic transition lines are approximated by a Voigt[10] profile, which is a convolution of a Doppler and a Lorentzian profile. This convolution consists of an infinite integral, which is numerically expensive and requires optimized numerical methods to solve in a reasonable time. Besides the Voigt profile, sometimes pure Doppler and Lorentzian profiles are also used, and for comparison reasons, we also implemented the “binned Gaussian integrated cross section” used by the ExoCross code (Yurchenko et al. 2018a). HELIOS-K allows one to choose between these four profiles. Figure 1 shows an example of the available line profiles.

For some species, there are non-Voigt profiles available, e.g., Wcislo et al. (2016), but those are not implemented in the current version of HELIOS-K. However, it is possible to replace individual transition lines with precalculated tables as necessary for certain alkali resonant lines.

2.1.1. Removing the Plinth

HELIOS-K supports the option of removing the plinth (also sometimes called the “base”) from transition lines. The height of the plinth is defined as the opacity value from a transition line at the cutting length position and depends on the line intensity and broadening parameters (e.g., Clough et al. 1989; Paynter & Ramaswamy 2011; Ptashnik et al. 2012). Removing the plinth is necessary when opacities are used in combination with a background opacity that already includes the described plinth for each truncated transition line. This is the case, for example, when using the CKD (Clough et al. 1989), MT_CKD (Mlawer et al. 2012), or BPS (Paynter & Ramaswamy 2011) water vapor continuum as an opacity background. Not removing the plinth would lead to double counting of it. An example of the plinth and how it is removed from a transition line is shown in Figure 2.

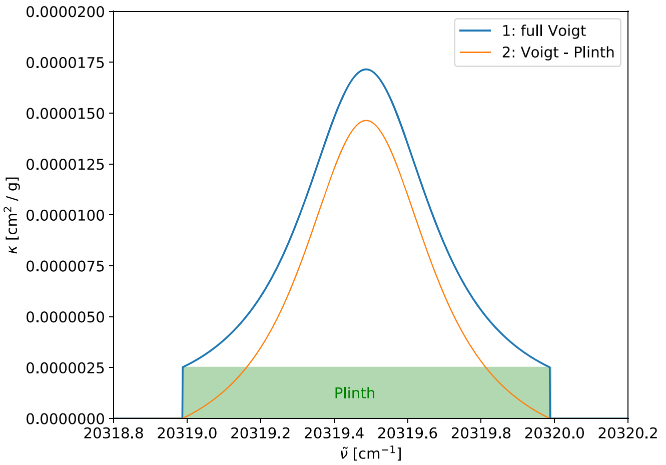


Figure 2. When truncated line wing opacities are used in combination with a far wing background, it may be necessary to remove the plinth from the opacities. The height of the plinth is defined as the opacity value at the cutting length position. Shown in blue is a full Voigt profile transition line, and shown in orange is the transition line with the removed plinth. The plinth itself is shown in green.

2.2. Background Theory

We follow the definitions of the parameters described in Grimm & Heng (2015) and Heng (2017) and repeat only some of them here for a better overview. Molecular or atomic opacities can be approximated by Voigt line profiles, which can be described by two parameters,[11] a and u. These two parameters for the Voigt line profiles are defined as

$$a = \sqrt{\ln 2}\,\frac{\Gamma_L}{\Gamma_D} \tag{1}$$

and

$$u = \sqrt{\ln 2}\,\frac{\tilde{\nu} - \tilde{\nu}_0}{\Gamma_D}, \tag{2}$$

where $\tilde{\nu}$ is the wavenumber, $\tilde{\nu}_0$ is the line center in wavenumber in cm^{-1}, and Γ_L and Γ_D are the Lorentz and Doppler half-widths in cm^{-1} , respectively. The Doppler half-width is given as

$$\Gamma_D = \frac{\tilde{\nu}_0}{c}\sqrt{\frac{2\ln 2 k_B T}{m}}, \tag{3}$$

where m is the mass in grams (molar mass in g mol^{-1} divided by the Avogadro constant), $k_B = 1.3806489 \times 10^{-16}\,\mathrm{cm^2\,g\,s^{-2}\,K}$ is the Boltzmann constant, and T is the temperature in kelvin.

The Lorentz half-width for HITRAN data is given as[12]

$$\Gamma_L = \frac{A}{4\pi c} + \left(\frac{T}{T_{\mathrm{ref}}}\right)^{-n}\left[\frac{\alpha_{\mathrm{air}}(P - P_{\mathrm{self}})}{P_{\mathrm{ref}}} + \frac{\alpha_{\mathrm{self}} P_{\mathrm{self}}}{P_{\mathrm{ref}}}\right], \tag{4}$$

[10] It should be noted that name of W. Voigt is usually mispronounced. The correct pronunciation is [fo:kt]; i.e., the vowels are articulated “oo,” rather than the commonly used “oi.”

[11] In the literature, these parameters are sometimes also called x and y.

[12] https://hitran.org/docs/definitions-and-units/

which is designed for studies of the terrestrial atmosphere, so broadening is considered in two parts: air broadening and self-broadening. For ExoMol data, the corresponding definition is (Tennyson et al. 2016)

$$\Gamma_L = \frac{A}{4\pi c} + \alpha_{\mathrm{ref}}\left(\frac{T_{\mathrm{ref}}}{T}\right)^n\left(\frac{P}{P_{\mathrm{ref}}}\right), \tag{5}$$

where α_{ref} gives the line broadening for the given broadening gas under consideration. In both cases, we have included the first term with the Einstein A coefficient (which has physical units of s^{-1}), but it is generally negligible.

For atomic opacities, the Lorentz half-width consists mainly of a combination of the natural broadening coefficient and pressure broadening due to collisions with electrons (Stark broadening) and other atomic and molecular species (van der Waals broadening). Stark broadening depends directly on the abundance of free electrons and is, therefore, only important in hot stellar atmospheres. In this work, we only consider natural broadening for atomic opacities.

2.2.1. Natural Broadening

The natural broadening of atomic transition lines is a consequence of the uncertainty principle $\Delta E \Delta t \geqslant \hbar/2$ (Kunze 2009; Draine 2011) and applies to both the upper and lower transition states. The line profile of the natural broadening effect can be described by a Lorentzian line profile with a half-width at half-maximum,

$$\Gamma_L = \frac{\Gamma_{\mathrm{nat}}}{4\pi}, \tag{6}$$

where Γ_{nat} is the sum over all Einstein A coefficients of transitions to the upper or lower energy levels (Draine 2011),

$$\Gamma_{\mathrm{nat},lu} = \sum_{E_j<E_u} A_{uj} + \sum_{E_j<E_l} A_{lj}. \tag{7}$$

For example,

$$\Gamma_{\mathrm{nat},31} = A_{31} + A_{32}$$

or

$$\Gamma_{\mathrm{nat},32} = A_{31} + A_{32} + A_{21}.$$

For Kurucz data, we use

$$\Gamma_L = \Gamma_{\mathrm{nat}} = \frac{10^{\Gamma_R}}{4\pi c}. \tag{8}$$

The quantity Γ_R is tabulated in the Kurucz database.

2.2.2. Line Intensity

Besides the Voigt line profile, the intensity S of the transition line is also needed. It is defined in units of $\mathrm{cm\,g^{-1}}$ as (e.g., Hill et al. 2013 or Chapter 5.3 of Heng 2017)

$$S = \frac{A}{8\pi c m}\frac{g'}{\tilde{\nu}_0^2 Q(T)} e^{\left(-\frac{hc}{k_B T}E''\right)}\left(1 - e^{\left(-\frac{hc}{k_B T}\tilde{\nu}_0\right)}\right), \tag{9}$$

where E'' is the lower-state energy in cm^{-1}, g' is the upper-state statistical weight, and $c = 2.99792458 \times 10^{10}\ \mathrm{cm\,s^{-1}}$ is the speed of light. Note that there is a typographical error in the first exponent of Equation (17) of Grimm & Heng (2015), which we correct here.

Table 1
The *a–u* Plane Is Split into Different Regions, Where Different Approximations on the Voigt Profile Calculations Are Applied

a–u Region	Limits	Method
AL	$10^6 < u^2 + a^2$ $u < 0$	First-order Gauss–Hermite quadrature
AR	$10^6 < u^2 + a^2$ $u > 0$	First-order Gauss–Hermite quadrature
B	$100 < u^2 + a^2 < 10^6$	Third-order Gauss–Hermite quadrature
C	$u^2 + a^2 < 100$	Algorithm 916 (Zaghloul & Ali 2012)

Note. We find this splitting to be very efficient for GPU calculations, where branch divergences should be avoided, and memory access should be minimized.

The partition function $Q(T)$ is in general available in tabulated form, together with the line list data. If the partition function is not available, it may be computed using (e.g., Sharp & Burrows 2007)

$$Q(T) = \sum_{i=1}^{n} g_i e^{-hcE_i/k_B T}, \tag{10}$$

where g_i and E_i are the statistical weight and the energy of the level i, respectively. Note that, as Sharp & Burrows (2007) wrote, this summation diverges formally for high pressure values. For the pressure values that we are typically interested in, the sum is still accurate enough. For molecular species, the partition function would also include electronic, vibrational, and rotational terms (Sharp & Burrows 2007).

2.2.3. Decompose the a–u Voigt Parameter Plane

Following Letchworth & Benner (2007) and as already described in Grimm & Heng (2015), we split the a and u parameter space into three different regions, called A, B, and C, and use different methods and approximations for the Voigt profile calculations. For the A region, we use a first-order Gauss–Hermite quadrature approximation. For the B region, we use a third-order approximation. For the C region, we solve the Voigt profile up to machine precision with an iterative method (Zaghloul & Ali 2012). In practice, we split the A region into a left (AL) and a right (AR) part in order to avoid many zero calculations at the line center. We do not split the B region in the same way, because most of the time the C region is very narrow. Splitting B would introduce too many unnecessary memory accesses, especially when the *a–u* regions are applied to thousands of neighboring transition lines, and the B and C regions can overlap. The used *a–u* regions are listed in Table 1 and illustrated in Figure 3. Figure 4 shows the fractional difference between the `HELIOS-K` Voigt calculation and the `SciPy` function `wofz` (`scipy.special.wofz`). The `wofz` function is similar to the algorithm in our region C. The largest error appears at the inner boundary of region B, where the Voigt function is approximated by a third-order Gauss–Hermite quadrature. The precision of `HELIOS-K` could be improved if necessary by expanding region C to larger values on the *u*-axis.

Note that Letchworth & Benner (2007) split the C region into different subregions with higher-order quadrature methods or interpolation of precalculated tables. But we find that by

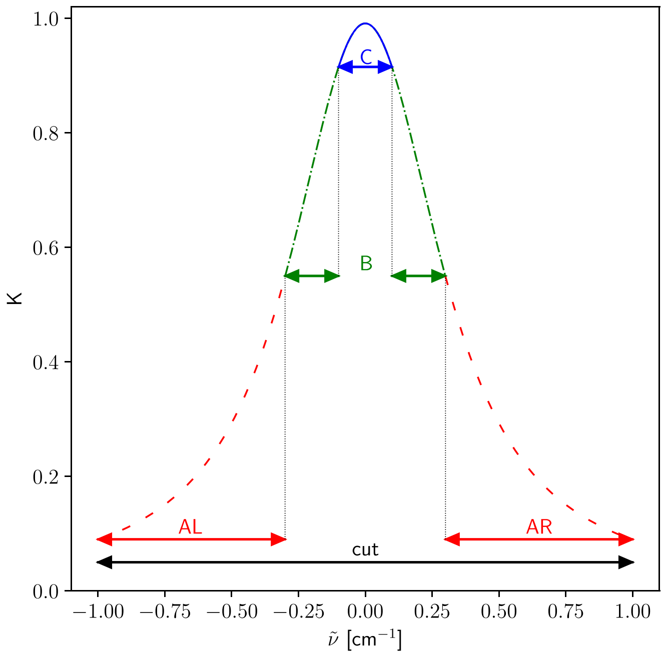


Figure 3. Example of the *a*–*u* regions, applied on a single transition line, with a cutting length of 1 cm^{-1}. In practice, thousands of neighboring transition lines are grouped together, and the regions, especially B and C, can overlap.

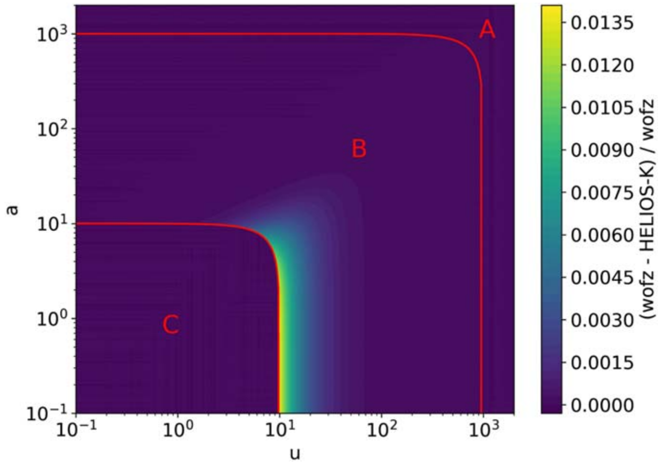


Figure 4. Fractional difference of the Voigt profile computed by `HELIOS-K` and using the `SciPy` function `wofz` (`scipy.special.wofz`). The boundaries between regions A, B, and C from Table 1 are shown in red. The algorithm in region C is similar to the `wofz` function, while regions B and A are approximations to speed up the calculation.

using GPUs, the three regions from Table 1 provide a good balance between the efficiency of the method, parallel applicability, and resource limitations on the GPU.

2.3. GPU Implementation Considerations

Using GPUs for numerical calculations is a very powerful approach. Modern GPUs can consist of more than 5000 CUDA cores, which allow us to parallelize the numerical problem at a high level. However, there are some limitations. First, not all CUDA threads are independent, and threads in the same warp (a collection of 32 threads) must perform the same operation but can use individual data (single instruction, multiple threads, SIMT). This means that thread divergences must be avoided. In our problem, thread divergences could occur when neighboring transition lines lie in different *a*–*u* regions and require a different algorithm to calculate the Voigt profile. To avoid those divergences, we have to sort the data and organize them in a well-suited form.

A second limitation of the GPUs is the memory transfer bottleneck. Before data can be used on the GPU, it must be transferred to the GPU through the peripheral component interconnect (PCI) express, which can cause large computing overhead time. However, the data transfer time can be hidden by using asynchronous and interleaved memory transfer. This is possible because the GPUs have separated copy and compute engines,[13] and both can work simultaneously. Besides the memory transfer, CUDA also allows us to overlap kernel execution with calculations performed on the CPU. In our problem, we can therefore hide data reading and transfer behind the opacity calculations on the GPU. But that only works if the GPU part takes long enough. That is typically the case when using high wavenumber resolution calculations ($\Delta\tilde{\nu} \approx 0.01\,\mathrm{cm}^{-1}$). If this is not the case, then the full computing time is dominated by pure data reading.

A further limitation is the slow global memory access. Nvidia GPUs have different levels of memory, mainly global memory, shared memory, and registers. Most of the time, memory access goes through global memory first and is then stored temporarily in shared memory or registers. Global memory access is slow and should be avoided. This can be achieved by designing the parallelization scheme in such a way that data can be reused multiple times from registers or shared memory.

2.4. Improvements and Upgrades to the `HELIOS-K` Opacity Calculator

This section describes details of the parallelization and new implementations of `HELIOS-K`. Readers who are not interested in these coding details can skip this section and continue with the different line list databases in Section 2.5.

2.4.1. Reading the Line List Data

Before the opacity calculation can be started, all of the line list data need to be read from the CPU and transferred to the GPU. We use a binary data format to store preprocessed line lists to speed up this reading process. How the line lists are preprocessed is described later in Section 2.5. To read the data from the hard disk sounds simple, but actually, one has to be very careful not to lose too much performance. In fact, when using low-to-moderate resolution in wavenumber, reading the line list can take more time than the opacity calculation itself. When the code is called many times, e.g., for different values of temperature and pressure, reading and writing of data can become the bottleneck. We use the following strategy to improve the reading and data transfer process.

1. Avoid reading data line by line. Reading the line list, e.g., by an *fscanf* command line by line, would not generate enough work to hide the overhead time, and the entire memory bandwidth would not be used. Instead, we read a bigger block of data, consisting of a few thousand

[13] Nvidia GeForce cards typically use the same copy engine for data transfer for host-to-device and device-to-host. Tesla cards have separate copy engines for host-to-device and device-to-host and can work simultaneously.

transition lines, at once by using the *fread* command and a memory buffer.

2. Use asynchronous and interleaved memory copy to transfer the data to the GPU. In this way, one block of data is transferred to the GPU at the same time as the CPU reads in the next block.
3. Overlap the reading and memory transfer with the opacity calculation of the previous block of data. Here we use the fact that the compute engine can work simultaneously with the copy engine and that the CPU can work in parallel to the GPU (see Section 2.3). In order to make this work, a clever scheduler of the opacity kernel calls and memory transfer commands is needed. This is described in detail in Section 2.4.6 and illustrated in Figure 6.

2.4.2. Code Parameters for the Voigt Profile Calculation

In practice, we do not use the parameters in Equations (1)–(4) in their stated form but instead cancel out the $\sqrt{\ln 2}$ factor in the numerator and denominator and directly store the inverse of the Doppler half-width, because the calculation of a product is faster than a division. The parameters used are then

$$x = (\tilde{\nu} - \tilde{\nu}_0) \cdot \alpha_D^{-1} \tag{11}$$

and

$$y = \Gamma_L \cdot \alpha_D^{-1}, \tag{12}$$

with

$$\alpha_D^{-1} = \frac{c}{\tilde{\nu}_0}\sqrt{\frac{m}{2k_B T}}. \tag{13}$$

The parameters x, y, and α_D^{-1} correspond to the parameters u, a, and Γ_D from Equations (1)–(3). We introduce here a new notation for the parameters that closely follows the implementation in the code. Since an ideal code implementation reduces the number of operations in each equation to the minimum, it is sometimes necessary to redefine parameters to be slightly different from their theoretical derivations.

When a constant resolution $\Delta\tilde{\nu}$ is used, the parameter x can be calculated very fast with a single fused-multiply-add operation as

$$x = d + i \cdot b, \tag{14}$$

with a thread index i, and the variables $d = (\tilde{\nu}_{\min} - \tilde{\nu}_0) \cdot \alpha_D^{-1}$ and $b = \Delta\tilde{\nu} \cdot \alpha_D^{-1}$. The quantity $\tilde{\nu}_{\min}$ is the starting point in wavenumber.

Assuming a constant resolution in wavenumber is a natural choice, because the wavenumber points nearly follow the positions of the transition lines. In `HELIOS-K`, it is possible to use a different wavenumber resolution for different bands. In that case, the variable x in Equation (14) is stored in a precalculated array.

In summary, the calculation of the Voigt profile needs a total of five parameters, which depend on the transition line and the wavenumber: the line intensity S; the quantities b, d, and y as defined above; and an additional quantity $\tilde{\nu}_{\mathrm{cut}}$ that handles the cutting length. The result of the opacity calculation is written into an array K. The major difficulty is how to handle these five parameters efficiently within GPU memory and use them in a parallel way.

2.4.3. Calculating the Line Parameters

As described in Section 2.4.2, we need five line parameters, S, b, d, y, and $\tilde{\nu}_{\mathrm{cut}}$. The calculation of these parameters is split into four different parts.

1. The line intensity (Equation (9)) may be written as
$$S = \frac{S'}{Q(T)} e^{\left(-\frac{hc}{k_B T}E''\right)} \left(1 - e^{\left(-\frac{hc}{k_B T}\tilde{\nu}_0\right)}\right), \tag{15}$$
where the term S' does not depend on the temperature or pressure,
$$S' = \frac{g' A}{8\pi c \tilde{\nu}^2 m}. \tag{16}$$
This quantity S' can be calculated beforehand and is part of the preprocessing routine, which is described below in Section 2.5. In this step, the partition function $Q(T)$ has to be determined by reading and interpolating the tabulated values provided by the line list databases.
2. The next step takes place after a block of line list data is transferred into a memory buffer on the GPU. The parameters S and y and, temporarily, the inverse Doppler half-width α_D^{-1} are then calculated on the GPU. We use one thread per transition line for these calculations, and the quantities are stored in arrays in global memory on the GPU.
3. The line list data obtained are mostly sorted at this point but can be slightly out of order due to pressure shift effects. For that reason, we sort the quantities along $\tilde{\nu}$ using the CUDA `Thrust` library.[14]
4. In the last step, the quantities b, d, and $\tilde{\boldsymbol{\nu}}_{\mathrm{cut}}$ are calculated, where $\tilde{\nu}_{\mathrm{cut}}$ depends on how the cutting length is defined. Options include a fixed cutting length, a fraction of the Doppler half-width, or a fraction of the Lorentzian half-width.

With this way of splitting the calculation of the parameters, we can reduce the sorting step to a minimum.

2.4.4. Parallelizing the Opacity Calculations

The opacity calculation can be parallelized in two dimensions: the wavenumber $\tilde{\nu}$ and the transition line index *id*. Parallelizing both dimensions simultaneously would not be efficient in terms of global memory data access. It is better to load a part from the data into registers and read it multiple times from there. Therefore, we divide the $\tilde{\nu}$–*id* plane into data tiles and only parallelize the line index dimension in the first step. For the wavenumber dimension, we use a serial loop to access all points. In a second step, the serial loop can be split further into multiple threads. How many threads to use for this process is dependent not only on the parameters but also on the GPU type and is therefore hard to estimate beforehand. For that reason, we use a self-tuning routine, which periodically checks for the best number of threads to use and adapts the CUDA-kernel calls automatically.

The top panel of Figure 5 gives a schematic of the parallelization scheme for an example with four transition lines and four points in

[14] https://developer.nvidia.com/thrust

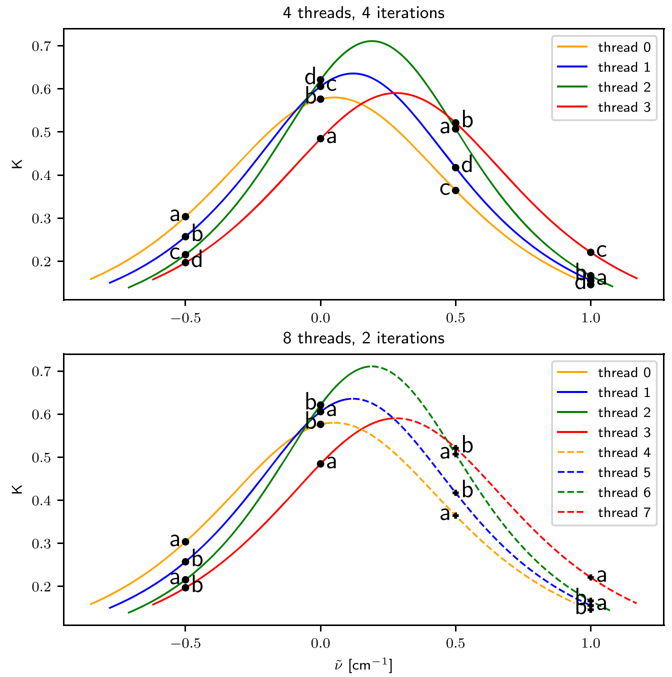


Figure 5. Schematic for parallelizing the opacity calculation for an example with four transition lines and four points in wavenumber. In the top panel, four threads are used, and each thread has to iterate through points a, b, c, and d. In the bottom panel, eight threads are used, and each thread has to iterate through points a and b. In the second case, the results have to be collected in two different arrays to avoid race conditions. The transition lines are cut at a distance from the line center of 0.9 cm^{-1}.

wavenumber (−0.5, 0.0, 0.5, and 1.0). When we use four threads in this example, each thread is assigned to a different transition line (different color in the figure) and has to iterate four times around the points in wavenumber. It is important that each thread calculates a different wavenumber point during the same iteration, because otherwise it would generate a race condition, which would lead to incorrect results. To prevent that, we can shift the starting point by the line index and take the modulo operator as

$$\tilde{\nu}\_\text{index} = (\text{line\_index} + \text{iteration\_index}) \\ \%(\text{number of threads}). \quad (17)$$

In the example from Figure 5, thread 2 would then calculate the opacities in the order (0.5, 1.0, −0.5, and 0.0). It is also possible to use more than four threads for the example above, as shown for eight threads in the bottom panel of Figure 5. But then we have to use separate arrays to collect the results of the different threads and merge the results later because, e.g., iteration b of thread 1 happens at the same time as iteration b of thread 3, and both threads write to the same point in wavenumber. Using eight threads instead of four increases the amount of parallel work performed by the GPU but also increases the fraction of reading versus computing, which means global memory access cannot be hidden behind the calculation time. Which version is the fastest depends on the individual case. Our algorithm self-tunes to find the best option.

Listing 1 shows a pseudocode for a GPU kernel that reads the transition line parameters and calls the Voigt profile function. The parameter NBx sets the number of iterations each thread has to perform. The parameter nl is the number of transition lines in a block of data. The parameters il1 and nstart define the starting points of the transition line index and wavenumber index of the current block of data. The kernel is launched on a two-dimensional grid. The *x*-dimension of that grid is defined as the maximum range of wavenumber points of the current data block divided by NBx. The *y*-dimension is used to launch multiple blocks of data together in order to reduce the kernel launch overhead time. The *y*-dimension is on the order of 16 or 32.

In lines 12–16 of Listing 1, the five transition line parameters from Section 2.4.2 are loaded from global memory into registers. The values in those registers can be reused NBx times. The result of the opacity calculation is stored in a two-dimensional shared memory array, K_s. At the end of the calculation, in lines 30–35, the results from multiple threads are merged into a single array, which can then be written back into global memory and at the very end copied back to the CPU.

Listing 1. Pseudocode of the Opacity Calculation Kernel

```
Line_ kernel(){
int idx = threadIdx.x;
int idy = blockIdx.x * NBx;

int il = il1 + blockIdx.y * nl;

for(int iil = 0;iil < nl;iil + = blockDim.x){

int iL = iil + il + idx;

S = S_ d[iL];
y = y_ d[iL];
d = d_ d[iL];
b = b_ d[iL];
cut2 = cut2_ d[iL];

for(int i = idx; i < NBx; ++i){
ii = i
iii = nstart + idy + ii;
x = d + iii * b;
if(cut2 < x * x + y * y){
K_ s[i
 }
__ syncthreads(); }
}
__ syncthreads();

if(idx < NBx){
for(int j = 1;j < blockDim.x / NBx;++j){
K_ s[idx] + = K_ s[idx + j*NBx];
__ syncthreads();
 }
 }
}
```

2.4.5. Calculating the $\tilde{\nu}$ Limits

As described before, the transition lines are grouped together into blocks of data, where each block consists of a few thousand lines. In order to call the Voigt calculation kernel, we have to determine the range of wavenumbers of each block. This range is defined as either the maximum or the minimum of the cutting range or the limits of the *a*–*u* regions described in Table 1. The limits $\Delta\tilde{\nu}$ can then be calculated with the equation

$$a^2 + u^2 = l^2,$$

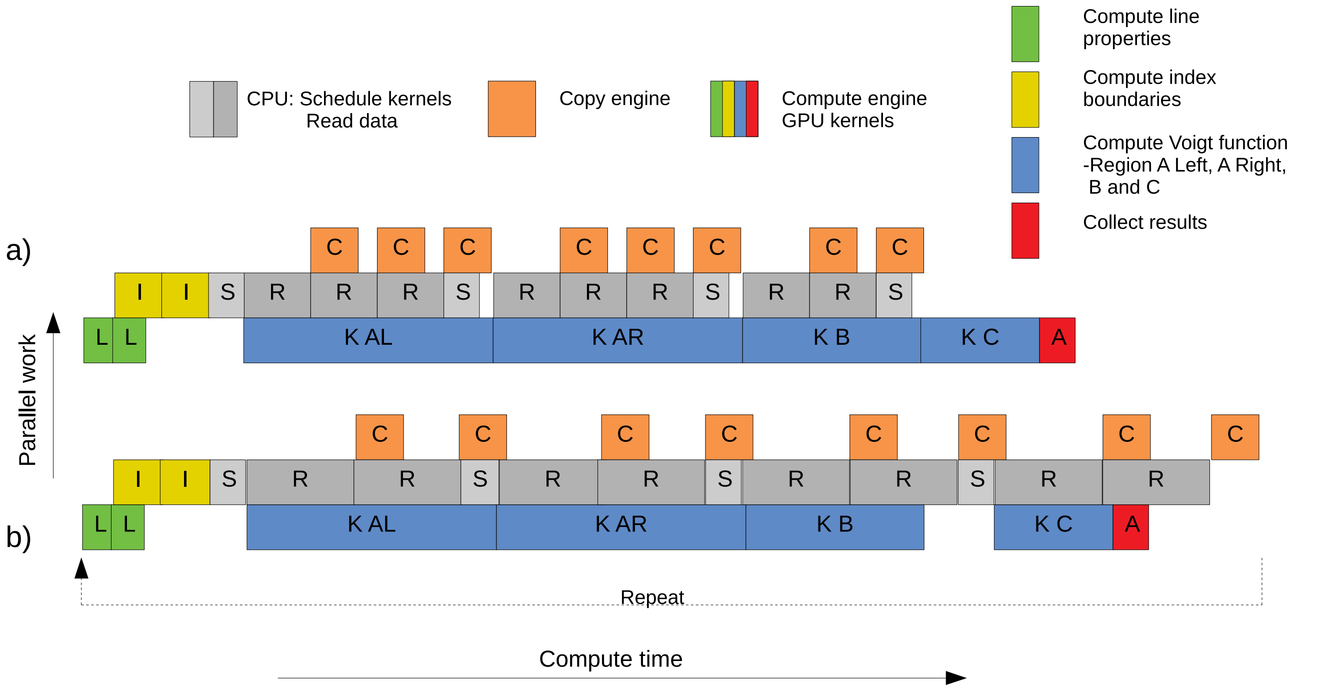


Figure 6. Simplified workflow of the code for two cases: (a) fast memory access and (b) slow memory access. The boxes indicate the compute time for all involved operations on the CPU (light and dark gray) and GPU (colors). When boxes are stacked on top of each other, the two operations are executed in parallel at the same time. In case (a), the compute time is dominated by the opacity calculation, and in case (b), it is dominated by reading the data from the hard disk. Case (b) should be avoided.

where l is either 10 or 1000, for regions B and C. Inserting Equations (1), (2), and (13) leads to

$$\frac{\Gamma_L^2}{\alpha_D^2} + \frac{(\tilde{\nu} - \tilde{\nu}_0)^2}{\alpha_D^2} = l^2$$

and

$$\Delta\tilde{\nu} = (\tilde{\nu} - \tilde{\nu}_0) = \sqrt{l^2 \cdot \alpha_D^2 - \Gamma_L^2}. \tag{18}$$

Therefore, the limits of the B and C regions for an individual transition line are $\tilde{\nu}_0 \pm \Delta\tilde{\nu}$. The limit of the A region is simply $\tilde{\nu}_0 \pm \tilde{\nu}_{\mathrm{cut}}$, where $\tilde{\nu}_{\mathrm{cut}}$ can depend on the Doppler or Lorentz half-width. In order to know the $\tilde{\nu}$ limits of an entire block of transition lines, we use a parallel reduction method on the GPU by using warp shuffle instructions.

2.4.6. The Core Workflow

Typically, the calculation of the Voigt function is the longest part of the GPU calculation, and we want to use that execution time to simultaneously read the next block of data from the hard disk and transfer it to the GPU. This is possible because different GPU engines can work at the same time and GPU kernel calls are asynchronous, meaning that the workflow goes back to the CPU immediately after the kernel is called and does not wait for completion. (See Section 2.3.) The resulting workflow is shown in Figure 6 and can be described as follows.

1. First, the line properties have to be calculated. This operation scales with the number of transition lines and is split into several different kernels, but it can be described mainly with two operations: (1) calculate the line intensity ("L"; green boxes in Figure 6) and (2) organization of the tiles described in Section 2.4.4 ("I"; yellow boxes). The "I" operation has to wait until "L" is finished, but by using multiple streams, parts from "L" and "I" can overlap on the GPU. This is indicated in Figure 6 by stacking boxes on top of each other.
2. The calculation of the Voigt profile (blue boxes) is split into four different parts (AL, AR, B, and C; Table 1). Before the calculation starts, the CPU determines the total number of threads for the next block of data by using the information from the "I" operation. This part is indicated in the figure as "S." Immediately after "S," the GPU kernel starts, and the CPU can be used for the next part. At the end of the four "K" parts, temporary results are collected ("A"; red boxes).
3. While the GPU is busy, we use the CPU to read future data blocks from the hard disk. We have to interrupt this operation as soon as the "K" kernel on the GPU is finished. In practice, we use a CUDA event query for this check.
4. The new data are transferred asynchronously to the GPU ("C"; orange boxes), and the execution time is hidden almost completely behind the reading operation or GPU execution time.
5. During the kernel execution, the self-tuning routine is called periodically to optimize the performance of the GPU, not shown in Figure 6.

The overall performance highly depends on whether the data reading is faster (case (a)) or slower (case (b)) than the Voigt profile calculation in the GPU (Figure 6). Case (b) can happen when the resolution in wavenumber $\Delta\tilde{\nu}$ is small, the GPU is very fast, or the data access from the hard disk is limited. Data access speed can generally be improved by moving the data closer to the calculation unit and ideally keeping data files in memory. If this is not possible, then opacities for multiple points in pressure should be calculated at once in order to reduce memory access. Some performance results using different data storage locations are listed in Section 3.5.

2.5. Preparation of Spectroscopic Line List Data for Computation

For the opacity calculations, we use line lists from several databases, which all use a different data format and support a different way of downloading the data. In order to use HELIOS-K in the most efficient way, we define a uniform binary data format for the line lists and provide scripts for automatically downloading the data and converting them to this format. Besides the data format, we also define a parameter file for each species, which contains the metadata of the species.

Our procedure is to process all transition lines from the given line lists. We do not use a truncation of weak lines. The line intensities are calculated in double precision floating point numbers. However, the final opacity function is reduced to single precision to reduce storage space, which truncates the opacity function to about 10^{-50}–10^{-40} cm^2 g^{-1}. If a truncation of weak lines is required to reduce the total number of transition lines, it could easily be included in the HELIOS-K preparation scripts.

In this section, we describe the file structures that we use for HELIOS-K and outline how the data from different databases can be preprocessed and used.

2.5.1. Parameter Files for Species

For each species (atom or molecule), we define a parameter file, which contains all necessary metadata and file structure information. These parameter files can be produced automatically and allow HELIOS-K to process the given data in a simple way, especially when the data from a database is split into many different subfiles. An example of a species-parameter file for HITRAN H_2O is given in Table 2. The file lists all isotopologues or isotopes of the species with the corresponding abundances, molecular masses, and partition function file names.

Table 2
Example Species-parameter File for HITRAN H_2O

```
Database = 0
Molecule number = 1
Name = hit16
Number of isotopologues = 7
#ID Abundance Q(296 K) g Molar Mass(g) Partition File :
11 0.997317 174.58 1 18.010565 q1.txt
12 0.002000 176.05 1 20.014811 q2.txt
13 3.718840E-04 1052.14 6 19.01478 q3.txt
14 3.106930E-04 864.74 6 19.01674 q4.txt
15 6.230030E-07 875.57 6 21.020985 q5.txt
16 1.158530E-07 5226.79 36 20.020956 q6.txt
17 2.419700E-07 1027.80 1 20.022915 q129.txt
Number of columns in partition file = 2
Number of line/transition files = 1
Number of lines per file :
304225
Line file limits :
0
30000
#ExoMol :
Number of states = 0
Number of columns in transition files = 0
Default value of Lorentzian half-width for all lines = 0
Default value of temperature exponent for all lines = 0
Version = 0
```

Note. The HELIOS-K species-parameter files contain all relevant metaparameters necessary for the opacity calculation. The files can be generated automatically with the provided scripts.

2.5.2. Binary File Structure

The different databases provide their data files in different structures and with a different number of parameters. We define a binary data format for the atomic or molecular transition lines. Depending on the database, these binary files contain different values, as indicated in Table 3. The possible values are described in Table 4. In the binary files, we preprocess the data in such a way that the final opacity calculation can be performed as fast as possible. One important part is the precalculation of the temperature- and pressure-independent part of the line intensity S', defined in Equation (16).

Table 3
Values Contained in the HELIOS-K Binary Data Files, Depending on the Database Used

Database	Parameters
HITRAN, HITEMP	ID, $\tilde{\nu}$, S', E_L, A, δ, γ_{Air}, γ_{Self}, n
ExoMol	$\tilde{\nu}$, S', E_L, A
Kurucz, VALD3	$\tilde{\nu}$, S', E_L, $\Gamma_{\rm nat}$
NIST	$\tilde{\nu}$, S', E_L, A

Note. The parameters are described in Table 4.

Table 4
Possible Values in the HELIOS-K Binary Data Files

Value	Description	Unit[a]	Type
ID	Species identity		char4
$\tilde{\nu}$	Center of the line	cm^{-1}	F64
S'	Equation (16)	cm g^{-1}	F64
E_L	Lower energy level	cm^{-1}	F64
A	Einstein A coefficient	s^{-1}	F64
δ_{Air}	Pressure-dependent line shift	cm^{-1} atm^{-1}	F64
γ_{Air}	Air-broadening coefficient	cm^{-1} atm^{-1}	F64
γ_{Self}	Self-broadening coefficient	cm^{-1} atm^{-1}	F64
n	Exponent of the temperature -dependent Lorentz half-width		F64
$\Gamma_{\rm nat}$	Natural broadening coefficient	s^{-1}	F64

Note.
[a] More details can be found at https://hitran.org/docs/definitions-and-units/.

2.5.3. ExoMol

The ExoMol line lists (Tennyson et al. 2020) can be accessed from www.exomol.com and are organized by molecules, isotopologues, and line list names. HELIOS-K provides a Python script to scan the webpage and produce a list of all available species. The species from ExoMol can be identified by their full species name, which includes the isotopologue information and the line list name, e.g., “1H2-16O__BT2”. Each species has a “.def” file, which contains properties like the isotopologue mass, default values of Lorentzian half-widths, default values of temperature exponents, or number of data files. It is recommended to check the number of files and maximal wavenumber of the line list. The data themselves are separated into a ".states" file and one or multiple ".trans" files. This data format is very compact to store on a server, but to use it with HELIOS-K, we need to produce a new line list, which contains [$\tilde{\nu}$, S', E_L, A]. This list is stored compactly as one or multiple binary files, which can be read by HELIOS-K. We provide a script that does the download and file conversion automatically. Some molecules contain the wavenumber of the molecular lines as a fourth column in the ".trans" files. However, it is not recommended to use that but instead to calculate it through the difference in energy levels. Partition functions are given in ".pf" files.

For some species, ExoMol also provides precalculated cross sections (xsec). When available, we compare our calculations with those data to verify our calculation (See Section 3.2).

2.5.4. ExoMol Superlines

For some molecules, ExoMol provides superlines (Yurchenko et al. 2017a; Tennyson et al. 2020). Superlines are temperature-dependent collections of line intensities integrated over small wavenumber bins. By using superlines, the total number of Voigt profile calculations can be reduced dramatically. The drawback is that individual transition lines are not represented anymore, and all lines are calculated with the same default pressure-broadening values. Practically, it means that the quantities S and S' from Equations (15) and (16) do not need to be calculated anymore but can be read in directly. Only the mass term needs to be included in the superline intensities. For example, the use of superlines reduces the ExoMol POKAZATEL water line list from 5.7×10^9 transition lines to 8.3×10^6 superlines.

2.5.5. HITRAN and HITEMP

The HITRAN database[15] (Rothman et al. 2010; Gordon et al. 2017) provides line lists in standard formatted text files (*.par files). These files can be downloaded for a single isotopologue or a mix of different molecules and isotopologues. In addition to the line list, HITRAN provides partition functions and molecular metadata in a webpage table. HITRAN supports a programming interface, HAPI (Kochanov et al. 2016), that allows data access through a Python script. However, several tests showed that the data in HAPI were not always updated; therefore, we opt to use the content of the webpage table directly. For usage with HELIOS-K, we extract only the data needed from the line lists and store them in more compact binary files that contain [ID, $\tilde{\nu}$, S', E_L, A, δ, γ_{Air}, γ_{Self}, n].

2.5.6. Kurucz

The Kurucz database[16] (Kurucz 2018, 2017) provides line lists and partition functions for a large number of neutral and ionized atoms. In this work, we use the GFNEW lines in wavenumber (gfallwn08oct17.dat), which includes isotope fractions and hyperfine splittings for some species. HELIOS-K provides a Python script to download the data file and available partition functions and extract the relevant data for the opacity calculations. The Kurucz database includes natural broadening (radiation dampening), as well as Stark and van der Waals broadening coefficients. The Stark broadening coefficient in the Kurucz database is given at a temperature of 10,000 K in units of the electron number density. The van der Waals coefficients, on the other hand, are also stated for a temperature of 10,000 K but in units of the atomic hydrogen number density. Van der Waals broadening in the Kurucz database is based on calculations employing a modified version of Unsöld's approximation (Unsöld 1938; Aller 1963). The tabulated coefficients can be adapted to collisions with He and H_2 by scaling them with a factor of 0.42 or 0.85, respectively.

Table 5
Example of a Potassium Transition Line with a Wrong Isotope Fraction in the Kurucz Database

$\tilde{\nu}$ (cm^{-1})	Isotope	Hyperfine Fraction	Isotope Fraction
5554.11	39	0.031	0.933
5554.11	39	0.437	0.933
5554.11	39	0.062	0.933
5554.11	39	0.156	0.933
5554.11	39	0.156	0.933
5554.11	0	1.0	**1.0 (wrong)**
5554.11	39	0.156	0.933

Note. The missing fraction must be calculated to get correct opacities. The bold number highlights the wrong entry in the line list; the isotope fraction should be 0.067117. Similar issues occur for several species and various transition lines. The numbers are rounded and taken from the file gfallwn08oct17.dat from http://kurucz.harvard.edu.

In the current version, we only include natural broadening in the opacity calculation. The resulting atomic opacities are therefore suitable for environments where collisional broadening is negligible, such as atomic absorption lines originating from upper atmospheres.

Partition functions. For most species, partition functions are available from the Kurucz database. If this is not the case, then we calculate the partition functions according to Equation (10) and the energy levels from the NIST database (see Section 2.5.7).

Isotope fraction correction. During the data processing, we noted an issue in the isotope fractions with the file gfallwn08oct17.dat. When a transition line consists of multiple isotopes but only one isotope is split up into its hyperfine structure, then the file contains an incorrect isotope fraction of the remaining isotopes. We show an example of a potassium transition line in Table 5. Potassium consists of three natural isotopes: ^{39}K (93.3%), ^{40}K (0.0117%), and ^{41}K (6.7%). In the example, isotope ^{39}K is split in its hyperfine structure, while isotopes ^{40}K and ^{41}K are combined into an entry with an isotope fraction of 1.0, which is not correct. The combined isotope fraction should instead be 6.7117%. If this factor is not corrected, then the resulting opacity can be significantly overestimated. Our HELIOS-K data preparation script scans all transition lines for these types of missing factors and corrects them.

Natural broadening coefficients. Natural broadening coefficients are provided for many species, but not for all. We compute the missing natural broadening coefficients according to Equation (7). The natural broadening coefficient $\Gamma_{\mathrm{nat},ij}$ between an upper state i and a lower state j includes the Einstein A coefficients from all transitions between two other levels, k and l, with either

$$\begin{cases} \langle E \rangle_i = \langle E \rangle_k \\ g_i = g_k \end{cases} \tag{19}$$

[15] www.hitran.org

[16] http://kurucz.harvard.edu

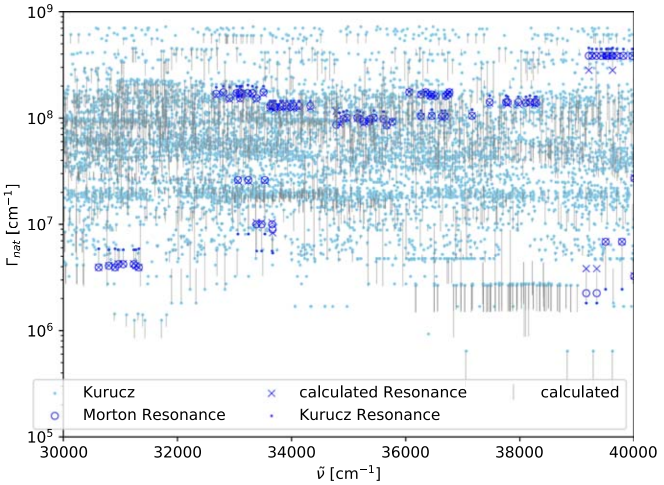


Figure 7. Comparison of the natural broadening coefficients $\Gamma_{\rm nat}$ of Fe between tabulated values from Kurucz (light and dark blue dots), calculated values using the Einstein A coefficients (gray error bars and crosses), and published data (Morton 2003, 2008) for resonance lines (blue circles).

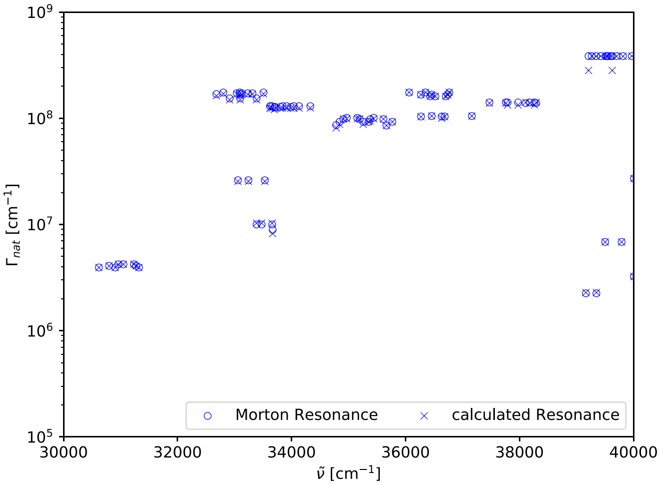


Figure 8. Comparison of the natural broadening coefficients $\Gamma_{\rm nat}$ of Fe between calculated values from NIST data and published values from Morton (2003). Only resonance lines are shown. Our calculated values agree well for most transition lines.

or

$$\begin{cases}\langle E\rangle_j = \langle E\rangle_k \\ g_j = g_k,\end{cases} \tag{20}$$

where $\langle E\rangle$ are the energy states averaged over the hyperfine sublevels and $g_{j,k}$ are the statistical weights from the Kurucz database. We compare our calculations to the tabulated values from Kurucz and published values from resonance lines (Morton 2003, 2008), as shown in Figure 7 for Fe. For most lines, our values correspond well with those from Morton (2003, 2008), but for a few transition lines, there are differences. Our calculated values are close to the ones from Kurucz, but for many lines, we cannot exactly reproduce them. The reason could be that we do not have full access to all relevant Einstein A coefficients or line configurations. The described comparison allows us to estimate the quality of calculated natural broadening coefficients for species where there are no tabulated values available.

2.5.7. NIST

The NIST database[17] (Kramida et al. 2019) provides line lists and energy levels for a large number of neutral and ionized atoms. The database can be accessed through a web interface by selecting the desired species and additional configuration options. The web access is intuitive and simple, but it is not very practical for accessing in an automated way, and it can be very time-consuming to download multiple line lists by hand. Therefore, we provide a Python script that navigates the webpage automatically and fills in the necessary web forms to download the line lists and energy level files. The script also extracts the needed quantities from the files and converts them into the standard HELIOS-K format.

A potential problem of the NIST database is that energy levels and transition lines can be listed multiple times. We noted this especially for the hydrogen atom, where energy levels are listed for individual angular momentum quantum numbers, besides the total averaged energy levels. Simply processing the entire line list would lead to duplicated transition lines; thus, it is necessary to filter the line lists for such duplicated entries.

Partition functions. We calculate the partition functions by the available energy levels according to Equation (10). As Sharp & Burrows (2007) noted, this summation diverges formally for high pressure values. For the pressure values that we are typically interested in, the sum is still accurate enough.

Natural broadening coefficients. The natural broadening coefficients are not provided by the NIST database; therefore, we use Equation (7) to calculate them by summing over all relevant Einstein A coefficients. The natural broadening coefficient $\Gamma_{{\rm nat},ij}$ between an upper state i and a lower state j includes the Einstein A coefficients from all transitions between two other levels, k and l, with either

$$\begin{cases}\text{configuration}_i = \text{configuration}_k \\ \text{term}_i = \text{term}_k \\ J_i = J_k\end{cases} \tag{21}$$

or

$$\begin{cases}\text{configuration}_j = \text{configuration}_k \\ \text{term}_j = \text{term}_k \\ J_j = J_k,\end{cases} \tag{22}$$

where “configuration,” “term,” and J are given in the NIST database.

We compare our values with published values for resonance lines due to Morton (2003, 2008) as shown in Figure 8 for Fe. Differences between the NIST data and Morton (2003, 2008) could occur because the underlying transition line data could be different.

The preprocessing of the NIST data requires the following steps.

1. Download the energy levels for all species. This can be done either manually through the web interface or by a script, which navigates the webpage automatically.
2. Calculate the partition function. Alternatively, the partition function would be available through the web form but must be queried for each temperature individually.

[17] https://www.nist.gov/pml/atomic-spectra-database

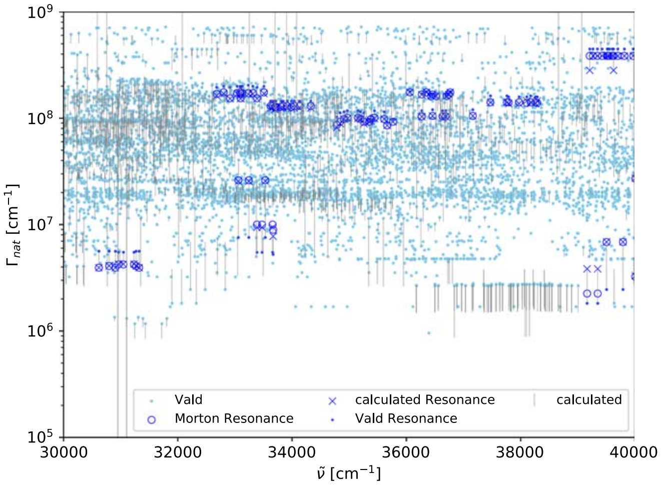


Figure 9. Comparison of the natural broadening coefficients Γ_{nat} of Fe between tabulated values from VALD3 (light and dark blue dots), calculated values using the Einstein A coefficients (gray error bars and crosses), and published data (Morton 2003, 2008) for resonance lines (blue circles). In most cases, our calculated values correspond well with those from Morton, but for a few lines, there are differences.

3. Download the line lists for all species. Similar to the energy levels, this can be done either manually through the webpage or by automated script.
4. Calculate the natural broadening coefficients and generate the HELIOS-K binary files.

2.5.8. VALD3

The VALD3 database[18] (Ryabchikova et al. 2015; Pakhomov et al. 2019) provides line lists for a large number of neutral and ionized atoms. After undergoing a registration process on the website, the database can be accessed through a web interface. Similar to our process for the NIST database, we provide a script to navigate the website automatically. For every data request, VALD sends an email with a link to download the data. By requesting multiple species, this process can result in a large number of emails and data links. To simplify the download, our script can be used without opening all the sent emails, and it converts the line lists to the standard HELIOS-K format.

Partition functions. Partition functions are not available in the VALD database. Therefore, we use the partition functions calculated from the NIST database.

Natural broadening coefficients. Natural broadening coefficients are provided for many species, but not for all. We compute the missing natural broadening coefficients as for the Kurucz database according to Equation (7). The natural broadening coefficient $\Gamma_{\text{nat},ij}$ between an upper state i and a lower state j includes the Einstein A coefficients from all transitions between two other levels, k and l, with either

$$\begin{cases} E_i = E_k \\ g_i = g_k \end{cases} \tag{23}$$

or

$$\begin{cases} E_j = E_k \\ g_j = g_k, \end{cases} \tag{24}$$

where the energy E and statistical weight g are given by the VALD database. We compare our calculations to the tabulated values from VALD3 and published values from resonance lines (Morton 2003, 2008) as shown in Figure 9 for Fe. For most lines, our values correspond well with those from Morton, but for a few transition lines, there are differences. Our calculated values are close to the ones from VALD3, but for many lines, we cannot exactly reproduce them. The reason could be that we do not have full access to all relevant Einstein A coefficients or line configurations. The comparison described allows us to estimate the quality of calculated natural broadening coefficients for species where there are no tabulated values available.

The preprocessing of the VALD3 data requires the following steps.

1. Request the data download for all species. This can be done either manually through the web interface or by a script, which navigates the webpage automatically.
2. Use the partition functions from the NIST database.
3. Calculate the natural broadening coefficients and generate the HELIOS-K binary files.

2.5.9. Alkali Resonance Lines

The wings of the resonance lines of the alkali metals sodium and potassium are known to deviate from the usual Voigt profile. Especially their far-wing line profiles exhibit strong non-Lorentzian behavior due to collisions with other molecules, such as, in particular, H_2 and He. Various descriptions of these line profiles have been provided in the past to characterize these non-Lorentzian line wings (see, e.g., Tsuji et al. 1999; Burrows et al. 2000; Burrows & Volobuyev 2003; Allard et al. 2012). The most recent theoretical calculations of the resonance line wings broadened by collisions with H_2 were published by Allard et al. (2019) for Na and Allard et al. (2016) for K. These calculations are valid for perturber densities of up to 10^{21} cm^{-3}.

Figure 10 shows an example of the non-Lorentzian behavior of the sodium resonance line wings at a temperature of 800 K and an H_2 partial pressure of 10 bars. We use the Kurucz line list to generate the cross sections, including the pressure-dependent van der Waals broadening by H_2. To convert the corresponding broadening parameters listed in the Kurucz line list from H to H_2, we use a scaling factor of 0.85. For the calculation involving the Allard et al. (2019) line wing description, we remove the two resonance lines from the line list, calculate the cross sections without them, and then add the two broadened resonance lines that have been constructed using the data from Allard et al. (2019) to the results.

The results shown in Figure 10 clearly suggest that the resonance lines have a very strong non-Lorentzian behavior and are distinctively asymmetric. They are super-Lorentzian close to the line center and become sub-Lorentzian in the far wings. Using a Voigt line profile for these two lines thus results in overestimating the line absorption in the far wings by orders of magnitude while strongly underestimating it near the line centers.

3. Results

3.1. Comparison of Atomic Opacities

We compare the atomic opacities between the databases NIST, Kurucz, and VALD3. As mentioned in Section 2.5.6, the

[18] http://vald.astro.uu.se/

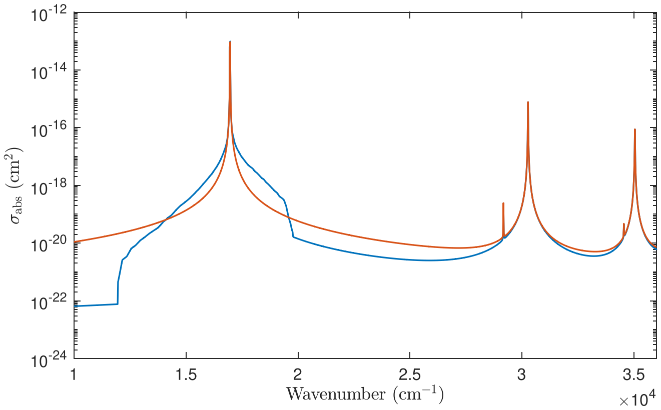


Figure 10. Comparison of the absorption cross-coefficients of Na between pure Voigt profiles (red line) and the non-Lorentzian line wings for the resonance lines based on Allard et al. (2019; blue line). The cross sections are calculated from the Kurucz line list data, including van der Waals broadening, and shown for a temperature of 800 K and an H_2 partial pressure of 10 bars.

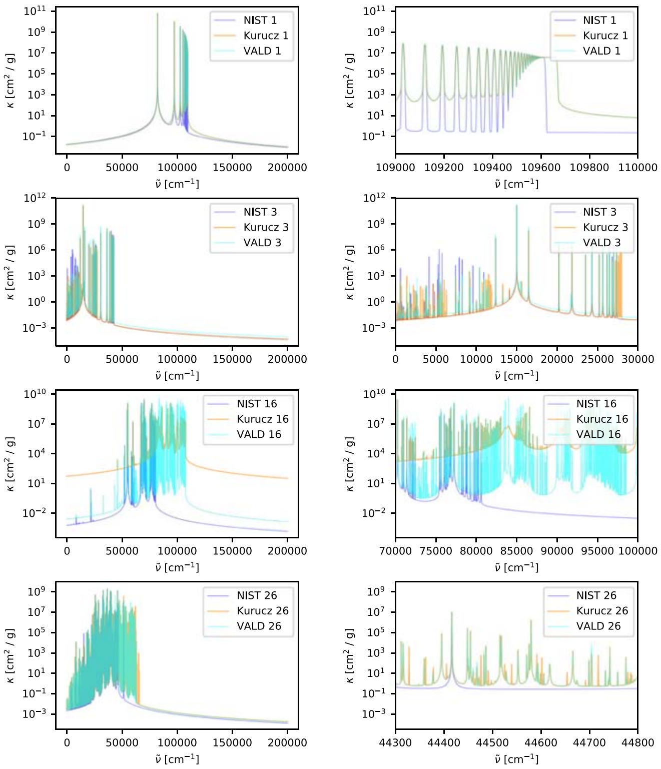


Figure 11. Opacity comparison of four atoms (H, Li, S, and Fe) between the databases NIST, Kurucz, and VALD3. The panels in the left column show the entire wavenumber range, and the panels in the right column show a more narrow band. We use a temperature of 3000 K and no cutting length. The three databases contain a different number of transition lines, leading to differences in wavenumber coverage of the opacities. Figures of all other species may be found in the HELIOS-K documentation.

calculations neglect pressure broadening. Therefore, the resulting opacities are suitable only for low-pressure environments with negligible collisional broadening. An example of four species (H, Li, S, and Fe) is shown in Figure 11. Figures with more species are shown in Appendix A (Figures A1–A4) and also in the HELIOS-K documentation.[19] The 1H example of Figure 11 shows a good agreement between all three databases. The only difference is in the natural broadening coefficients from NIST. In the example of 3Li, the agreement is not as good as for 1H, and some of the smaller transition lines are different in all three databases. The more intensive transition lines agree better with the exception of the natural broadening coefficient of the largest line. The example of S shows that NIST has far fewer lines than Kurucz and VALD3, and between Kurucz and VALD3, there is a large difference in the natural broadening coefficients around $\tilde{\nu} = 84{,}000$ cm^{-1}. The overall agreement for Fe is better, but again, NIST has far fewer lines, and Kurucz has more lines than VALD3. In general, NIST has significantly fewer lines than Kurucz and VALD3. Kurucz and VALD3 are similar for many species, but they are still not identical. For some species, VALD3 has fewer lines than Kurucz. The greatest impact on the opacities comes from the natural broadening coefficients from large resonance lines.

It should be noted that use of a Voigt profile for atomic lines is not always suitable. Autoionization lines (Auger transitions) are a particular example that is not well described by a Voigt profile. The wings of such a line are more appropriately described by a so-called Fano profile (Fano 1961), which decreases much faster than a Voigt profile. The Fano profile, however, depends on a parameter q that cannot be easily calculated (see Merts & Magee 1972 for details). Therefore, using a Voigt profile for these lines will result in an overestimate of the absorption, creating an artificial continuum. This can be seen, for example, in Figure 11 for sulfur (atomic number 16). The opacities based on the Kurucz line list show an elevated continuum due to autoionization lines located in the far-UV wavelength region. Since these lines are not included in the VALD3 and NIST databases, their opacities are not affected by the overestimated line wings.

Other problematic lines are resonance lines, where using a normal Voigt profile can also lead to over- or underestimated absorption. In particular, the Lyα wings of atomic hydrogen shown in the top left panel of Figure 11 are again overestimated by using a Voigt profile. Instead of following a Voigt profile, the actual line profile converges toward the Rayleigh scattering opacity far away from the line center (Dijkstra 2019).

3.2. Validation against ExoMol Opacities

For many species, ExoMol provides precalculated binned Gaussian cross sections, which we use to validate our calculations. Even if the validation is not done on a full Voigt profile, it confirms that our line list preprocessing and line intensity calculations are done correctly. In Figure 12, we show an example comparison of the H_2O POKAZATEL line list (Polyansky et al. 2018), and Figure 13 shows the same for CO (Li et al. 2015). Individual points in wavenumber are not reproduced exactly, as shown in the bottom panel. One reason is that every point in wavenumber can consist of millions of individual transition lines, for which the intensity can vary by several orders of magnitude. If the order of summation over all transitions involved is changed, it can affect the differences.

In Appendix B (Figures B1–B3), we show a comparison for 45 molecules between binned Gaussian cross sections from ExoMol and computed with HELIOS-K. The fractional difference $(\kappa_{\rm HELIOS-K} - \kappa_{\rm ExoMol})/\kappa_{\rm HELIOS-K}$ typically reaches values between 10^{-7} and 10^{-1}, which shows a good agreement between the two calculations. The difference can also reach

[19] https://helios-k.readthedocs.io/en/latest/index.html

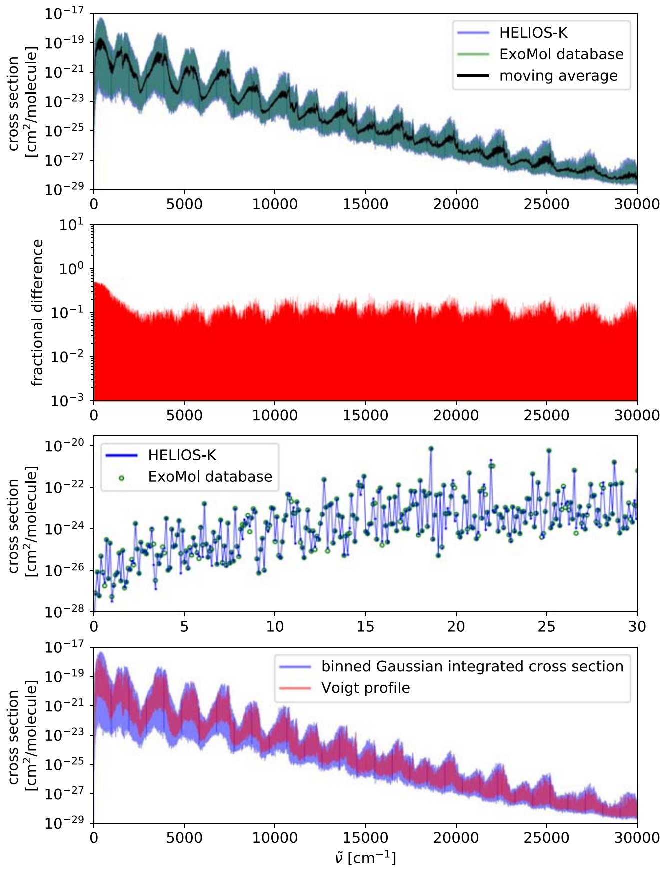


Figure 12. First panel: comparison of our calculated cross sections of the POKAZATEL line list with cross sections provided from the ExoMol database, showing a good agreement. Second panel: relative error between our work and ExoMol cross sections. Third panel: zoom comparison between our work and cross sections from ExoMol. Individual points in wavenumber are not reproduced exactly due to different orders in summation processes. Fourth panel: comparison between the binned Gaussian integrated cross section and the full Voigt profile. Calculations are done with a resolution of 0.01 cm^{-1} and a cutting length of 100 cm^{-1}.

higher values, which is most likely caused by division of very small numbers and limited numerical data precision.

Examples of molecular opacities (full Voigt profile) for a temperature of 1500 K (500 K for SO_3), pressure of 0.001 bar, and cutting length of 100 cm^{-1} are shown in Figures C1–C4. The figures include TiO (McKemmish et al. 2019), VO (McKemmish et al. 2016), AlO (Patrascu et al. 2015), SiO (Barton et al. 2013), CN (Brooke et al. 2014b), CH (Masseron et al. 2014), CP (Ram et al. 2014), CS (Paulose et al. 2015), H_2O (Polyansky et al. 2018), CO_2 (Rothman et al. 2010), CO (Rothman et al. 2010; Li et al. 2015), PN (Yorke et al. 2014), PO (Prajapat et al. 2017), PS (Prajapat et al. 2017), CH_4 (Yurchenko & Tennyson 2014), NH_3 (Yurchenko et al. 2011), C_2H_2(Gordon et al. 2017), HCN (Barber et al. 2013; Harris et al. 2006), SO_2 (Underwood et al. 2016a), SO_3 (Underwood et al. 2016b), SH (Yurchenko et al. 2018b), H_2S (Azzam et al. 2016), AlH (Yurchenko et al. 2018c), SiH_4 (Owens et al. 2017), SiH (Yurchenko et al. 2017b), SiS (Upadhyay et al. 2018), NS (Yurchenko et al. 2018b), NH (Brooke et al. 2014a), NO (Wong et al. 2017), and NO_2 (Rothman et al. 2010; Hargreaves et al. 2019).

During our analysis, we also spotted several issues in the ExoMol cross sections or data files, which have since been corrected.

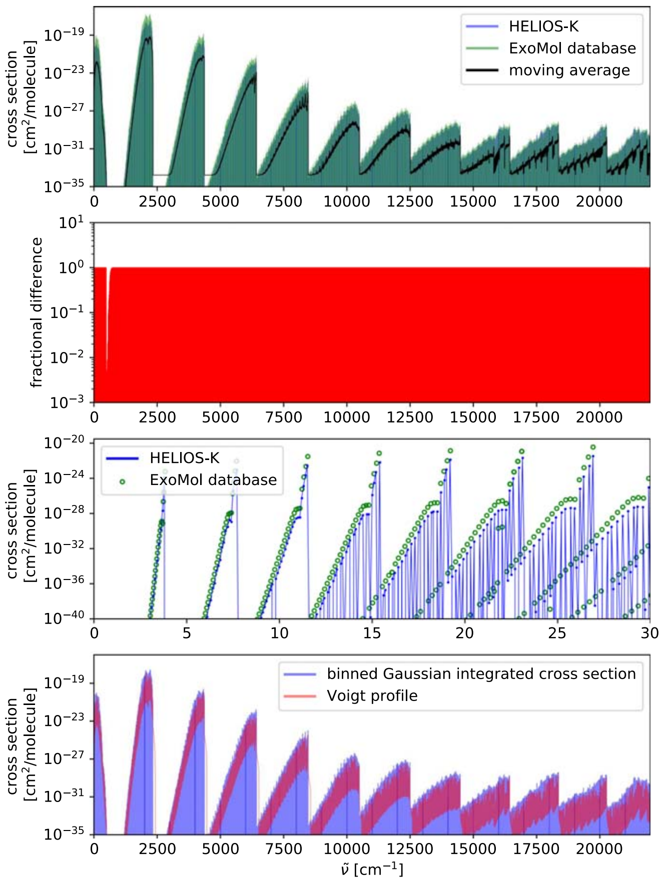


Figure 13. Same as Figure 12 but for CO.

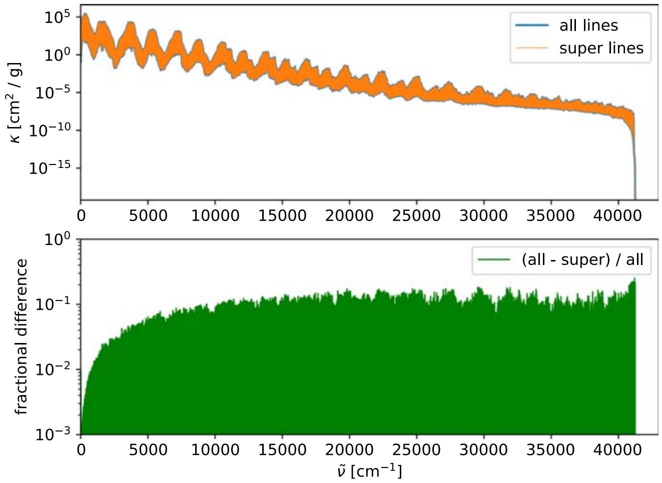


Figure 14. Comparison of the ExoMol POKAZATEL water line list with ExoMol superlines for $T = 1000$ K, $p = 1$ bar, and $\Delta\nu = 0.01$ cm^{-1}. The fractional difference between the opacity from all lines and the superlines is between 10% and 20% for most of the wavenumber range.

3.3. Comparison of ExoMol Superline Opacities to Full Opacities

We compare the opacities computed with superlines to opacities from the full ExoMol line lists. Figure 14 shows an example of the POKAZATEL water line list. While individual transition lines are not represented in the superlines, the overall opacities agree well, with a fractional difference of less than

20%. By using superlines, the computational speed of the example in Figure 14 is increased by a factor of 720.

3.4. The Effects of Line Lists on Atmospheric Calculations

In the following section, we estimate the impact on atmospheric modeling when using different line lists. Herein, we compare two different line lists for water: the older BT2 water line list (Barber et al. 2006) and the newer POKAZATEL water line list (Polyansky et al. 2018). Using those, we run models predicting atmospheric temperatures in radiative–convective equilibrium and corresponding planetary emission spectra. We choose two reference cases: (i) a typical hot Jupiter with a primordial H_2/He envelope with solar elemental abundances and (ii) a super-Earth with a water-steam atmosphere.

The atmospheric radiative transfer is calculated with the radiative–convective code HELIOS (Malik et al. 2017, 2019b). Details of the opacity and chemistry calculations are given in Malik et al. (2019a). The planetary parameters for HD 189733b and GJ 1132b are chosen as in Malik et al. (2017, 2019a). For simplicity, the host star is modeled as a blackbody in both cases.

The top row of Figure 15 shows the results for the hot Jupiter test case in the form of a vertical temperature–pressure (TP) profile, planetary emission spectrum, and relative difference in emitted flux. The latter is calculated as $|F_{\mathrm{POK}} - F_{\mathrm{BT2}}|/F_{\mathrm{POK}}$, with F_{POK} and F_{BT2} being the spectral fluxes from the POKAZATEL and BT2 models at a given wavelength. The spectra are calculated at the native resolution $R = 3000$ and also shown downsampled to $R = 30$. Choosing different water line lists causes a negligible effect on the vertical TP profile, with a temperature difference of <2 K for any layer. However, due to the unequal presence and strength of spectral lines between the two line lists, the difference in the planetary emission spectrum is significant. The relative difference is higher than 100% (up to 10%) for $R = 3000$ (for $R = 30$) at certain wavelengths in the near-infrared. The large dependency of this result on the spectral resolution is not surprising considering that the impact of individual lines decreases with decreasing spectral resolution.

The water world case is shown in the bottom row of Figure 15. Here the temperatures in the bottom atmosphere deviate between the two models by around 15 K. Consequently, and also because water is the only atmospheric absorber, the relative difference in the spectrum is larger than in the hot Jupiter case, namely, over 30% for some wavelengths at $R = 30$. Since POKAZATEL includes more spectral lines than BT2, the emission of the POKAZATEL model is often smaller by the relative amount shown.

In terms of observational consequences, the absolute difference associated with the secondary eclipse depth is up to ~40 ppm for the hot Jupiter case and up to ~2 ppm for the water world case in the near-infrared range of wavelengths.

3.5. Performance

An important part of the opacity calculation is the reading of the line list files. The performance therefore depends highly on the memory access speed. To quantify this dependency, we first test how long it takes to read the preprocessed BT2 water line list files (Barber et al. 2006; 15 Gbyte, $\tilde{\nu}$ range 0–30,000 cm^{-1}) from memory. We test three scenarios: reading the file over a network from a different server, reading from the hard disk, and reading from RAM. We also test three different systems.

1. A desktop machine (3.6 GHz) with a GeForce GTX 980 GPU and 10 Gbyte of free RAM. This machine does not have enough RAM to store the entire BT2 line list.
2. A GPU server (1.7 GHz) with four GeForce GTX 1080 GPUs and 32 Gbyte of RAM.
3. A cluster node (2.2 GHz) on the Ubelix supercomputer with eight GeForce RTX 2080ti GPUs and 256 Gbyte of RAM.

For all tests, we use only a single GPU.

Shown in Table 6 is the time needed to read the entire BT2 water line list from different memory types. It is clear that only reading the data files can take a substantial amount of time, and it is important to carefully plan memory usage before doing large opacity calculations. Table 6 indicates that there can be very large differences associated with the memory speed, depending on the system configuration.

Shown in Table 7 is the time needed for the entire calculation to complete (including memory access) for different computer systems. Again, the measured times clearly show that the performance depends highly on the memory speed. If the line list fits into the system RAM, then one can achieve a significantly higher performance than reading from a hard disk or even over a network cable.

A way to further increase the performance is to combine opacity calculations from different pressure values into a single run. In this way, the number of data reads can be reduced because the data remain on the GPU for the different pressure point calculations. This effect is shown in Table 8. By using 10 pressure points in a single run, the overall performance of low-resolution runs can be improved by about 60%–70%. In high-resolution runs, the effect is not as strong.

Shown in Figure 16 are timing measurements for different parts of HELIOS-K for the example of the BT2 water line list. One can see that for low-resolution opacity calculations, the reading part of the line list dominates the entire calculation. The Voigt calculation depends on the resolution and cutting length, while the other parts depend only on the number of transition lines in the line list database. Also shown is the time needed to download the line lists from the ExoMol website and preprocess the line list into the HELIOS-K binary file format. These two timings depend on the internet connection speed and the CPU clock rate of the machine.

HELIOS-K can perform a typical opacity calculation with a resolution of $\Delta\tilde{\nu} = 0.01$ with 10^7 lines in about 1 s.

For comparison, Yurchenko et al. (2018a) listed the timing of the ExoCross code as 251 s, excluding the data reading and by using approximations for the line wings. HELIOS-K performs the entire calculation in 12.5 s, including the data reading time. The traditional Humlicek approach takes 2775.6 s (excluding the data reading time).

4. Discussion

We present an improved and more powerful version of the GPU opacity calculator HELIOS-K. For certain species, the code achieves a speed-up of nearly 2 orders of magnitude since the first version presented in Grimm & Heng (2015). The code also supports more features and allows a simple use of different

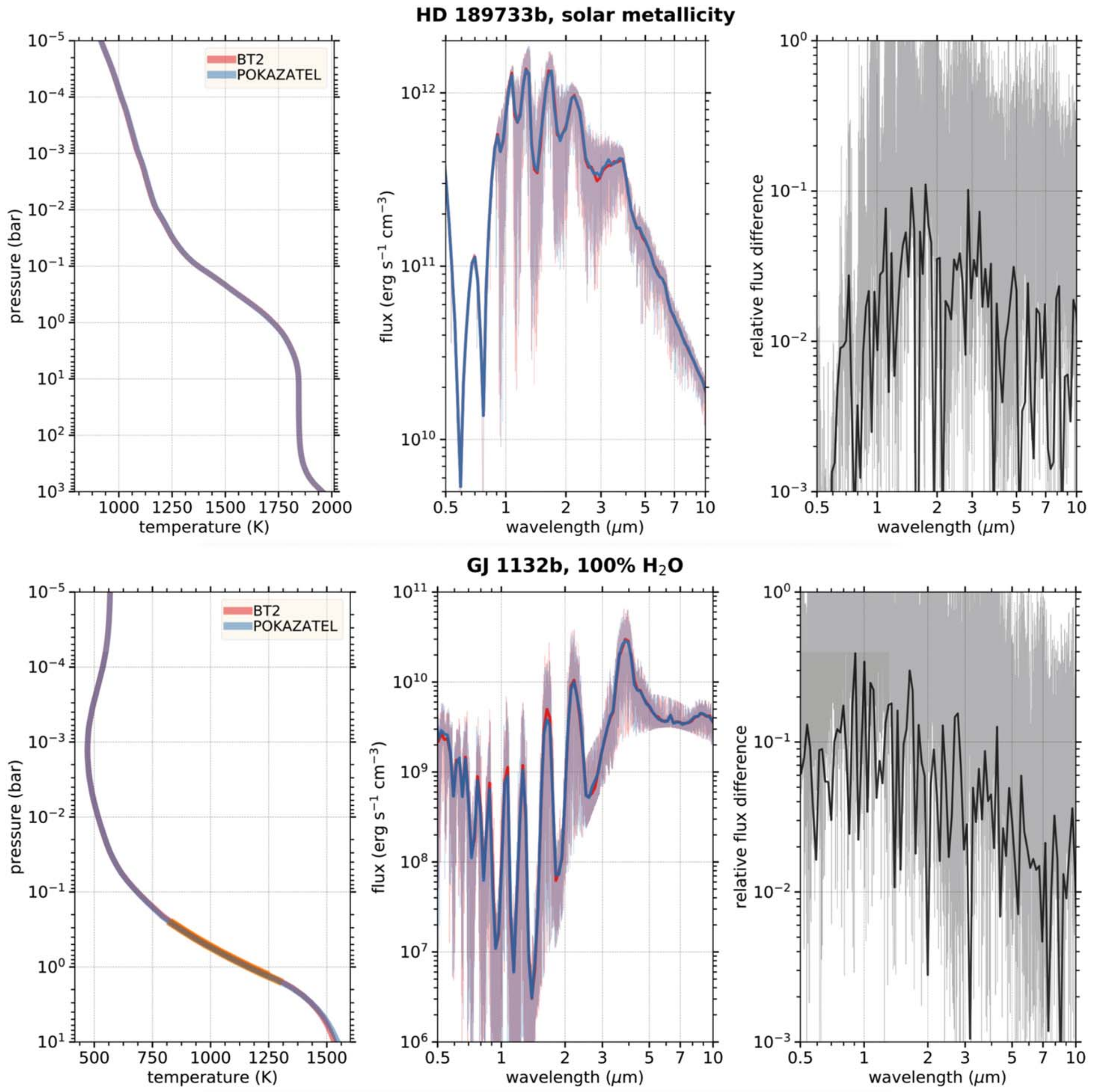


Figure 15. The TP profiles and emission spectra for a hot Jupiter with a solar abundance atmosphere (top) and a super-Earth with a water-steam atmosphere (bottom), calculated using the BT2 and POKAZATEL line lists for the water opacities. The right panels show the relative difference in emitted flux between the BT2 and POKAZATEL models. We find relative differences of up to 10%–40% ($R = 30$) and >100% ($R = 3000$) for certain wavelengths in the near-infrared. These translate to an absolute difference of ~30 ppm (~2 ppm) in the secondary eclipse signal of HD 189733b (GJ 1132b) at certain wavelengths in the near-infrared.

line list databases. During our work, we encountered several issues with multiple databases, which we reported to the maintainers. Our new code is fast enough to process the current largest molecular line lists from ExoMol. While the progress in the speed of modern GPUs helped to reduce the computing time, the largest increase in speed comes from the new parallelization algorithm, especially the usage of asynchronous and simultaneous data read and computation techniques, which turn out to be very powerful. It is not only the GPU itself that makes the speed-up but also the efficient interaction between CPU, GPU, and memory as a full system. But it is clear that when even bigger molecular line lists get published, new challenges could occur in both computing time and storage capacity. Atomic opacities are not as challenging in terms of computing power, but there we are faced with the incompleteness of theory. When atomic opacities are approximated by Voigt profiles with Lorentzian or sub-Lorentzian line wings, it can introduce significant errors in the opacity function. This is

Table 6
Time to Read the Entire BT2 Water Line List for Different Machines and Memory Types

System	GPU	$\tilde{\nu}$ Range in cm^{-1}	Data Location	Time
Desktop	GTX 980	0–30,000	Network	252 s
Desktop	GTX 980	0–30,000	Hard disk	30.0 s
Desktop	GTX 980	0–30,000	Partially RAM	12.5 s
Server	GTX 1080	0–30,000	Hard disk	40.5 s
Server	GTX 1080	0–30,000	RAM	1.9 s
Cluster	RTX 2080ti	0–30,000	Hard disk	55.5 s
Cluster	RTX 2080ti	0–30,000	RAM	4.8 s

Note. The GPUs are not involved in this test but are listed for a better overview of the system types. The measured times indicate that the memory type used can highly affect the performance.

Table 7
Performance of BT2 Water Line List for $\Delta\tilde{\nu} = 0.1$ cm^{-1} and a Cutting Length of 25 cm^{-1}

GPU	$\tilde{\nu}$ Range in cm^{-1}	Data Location	Time
GTX 980	0–5000	Over network	172.6 s
GTX 980	0–5000	Hard disk	20.9 s
GTX 1080	0–5000	Hard disk	26.3 s
GTX 980	0–5000	RAM	13.9 s
GTX 1080	0–5000	RAM	**8.8 s**
GTX 980	0–30,000	Partially in RAM	30.7 s
GTX 1080	0–30,000	RAM	**16.5 s**

Note. An essential factor of the overall performance is the memory access of the line lists. For the best performance, the data must remain fully in RAM; the bold numbers indicate the best scenario.

Table 8
Performance of BT2 Water Line List for $\tilde{\nu} = 0$–5000 cm^{-1}, Cutting Length 25 cm^{-1}

GPU	nP	$\tilde{\nu}$ Range in cm^{-1}	$\Delta\tilde{\nu}$ in cm^{-1}	Time
GTX 1080	1	0–30,000	10.0	10.1 s
GTX 1080	1	0–30,000	1.0	10.4 s
GTX 1080	1	0–30,000	0.1	16.5 s
GTX 1080	1	0–30,000	0.01	99.2 s
RTX 2080ti	1	0–30,000	10.0	10.7 s
RTX 2080ti	1	0–30,000	1.0	10.8 s
RTX 2080ti	1	0–30,000	0.1	12.5 s
RTX 2080ti	1	0–30,000	0.01	44.7 s
GTX 1080	10	0–30,000	10.0	66 s
GTX 1080	10	0–30,000	1.0	76 s
GTX 1080	10	0–30,000	0.1	165 s
GTX 1080	10	0–30,000	0.01	1044 s
RTX 2080ti	10	0–30,000	10.0	57 s
RTX 2080ti	10	0–30,000	1.0	66 s
RTX 2080ti	10	0–30,000	0.1	99 s
RTX 2080ti	10	0–30,000	0.01	475 s

Note. Here nP indicates the number of pressure points in the calculation. At low resolution, using multiple pressure points can improve the performance.

especially true for resonance and autoionization lines, where broadening mechanisms are not known well enough.

We computed opacity functions for several hundred species, which we store in our database and share with the community via the website www.opacity.world and our data server https://chaldene.unibe.ch or via https://dace.unige.ch/opacity and https://dace.unige.ch/opacityDatabase. It is evident that opacity calculations are a complex, tedious undertaking. In order to ensure reproducibility, the computed opacities have been made freely available on the Swiss PlanetS platform DACE (https://dace.unige.ch). To aid the user in developing intuition, the opacities may be visualized via a graphical user interface (https://dace.unige.ch/opacity). Opacities may be downloaded at the full spectral resolution (0.01 cm^{-1}) in binary format (https://dace.unige.ch/opacityDatabase). Alternatively, the user may specify customized arrays of temperature and pressure, as well as the range of wavenumbers or wavelengths required, and interpolated opacities will be produced in the HDF5 format. In the future, we envision a "versioning" capability, where different implementations using the same generation of spectroscopic line lists (e.g., by different researchers) or implementations using different generations of line lists may be archived, so that differences in the subsequent models or simulations may be diagnosed. It is our belief that these capabilities are critical for the exoplanet community as we move into the era of precision spectroscopy of exoplanetary atmospheres with the James Webb Space Telescope and the next generation of large ground-based telescopes.

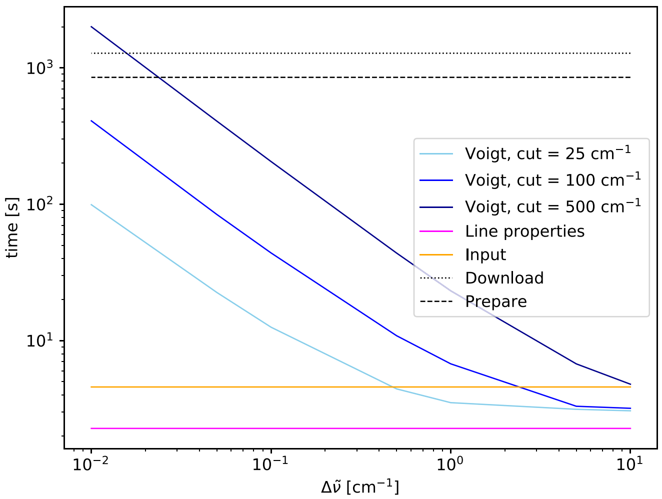


Figure 16. Timing of different parts of HELIOS-K for the example of the BT2 water line list (0–30,000 cm^{-1}). The Voigt function calculations are in dark blue, medium blue, and light blue for cutting lengths of 500, 100, and 25 cm^{-1}. Reading the line list (from RAM) is in orange, and preparing the transition line properties is in magenta. Also shown is the time needed to download and unpack the line list from the ExoMol servers and preprocess it into the HELIOS-K binary format. For low-resolution calculations, the reading part of the line list dominates the opacity calculation. Timings are measured with an Nvidia GTX 1080 GPU.

Calculations were performed on UBELIX (http://www.id.unibe.ch/hpc), the HPC cluster at the University of Bern. This work has made use of the VALD database, operated at Uppsala University, the Institute of Astronomy RAS in Moscow, and the University of Vienna. We acknowledge partial financial support from the Swiss National Science Foundation, the European Research Council (via a Consolidator Grant to K.H.; grant No. 771620), the PlanetS National Center of Competence in Research (NCCR), the Center for Space and Habitability (CSH), and the Swiss-based MERAC Foundation.

Appendix A
Atomic Opacities

The following Figures A1–A4, together with Figure 11 in the main text, give an overview of 48 atomic opacities, for the NIST, Kurucz, and VALD3 databases.

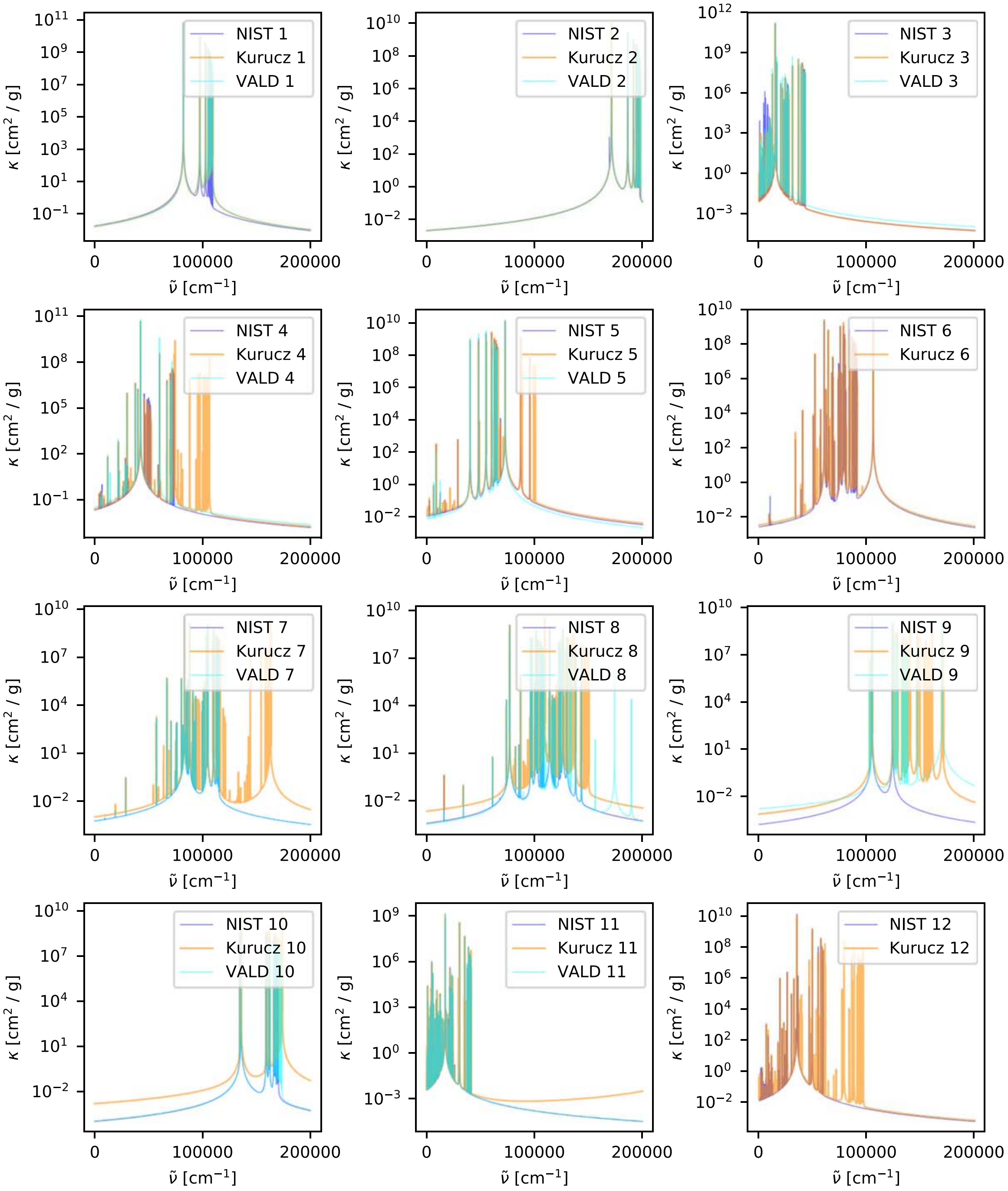


Figure A1. Comparison of atomic opacities between the NIST, Kurucz, and VALD3 databases, elements 1–12. We use a temperature of 3000 K and no cutting length.

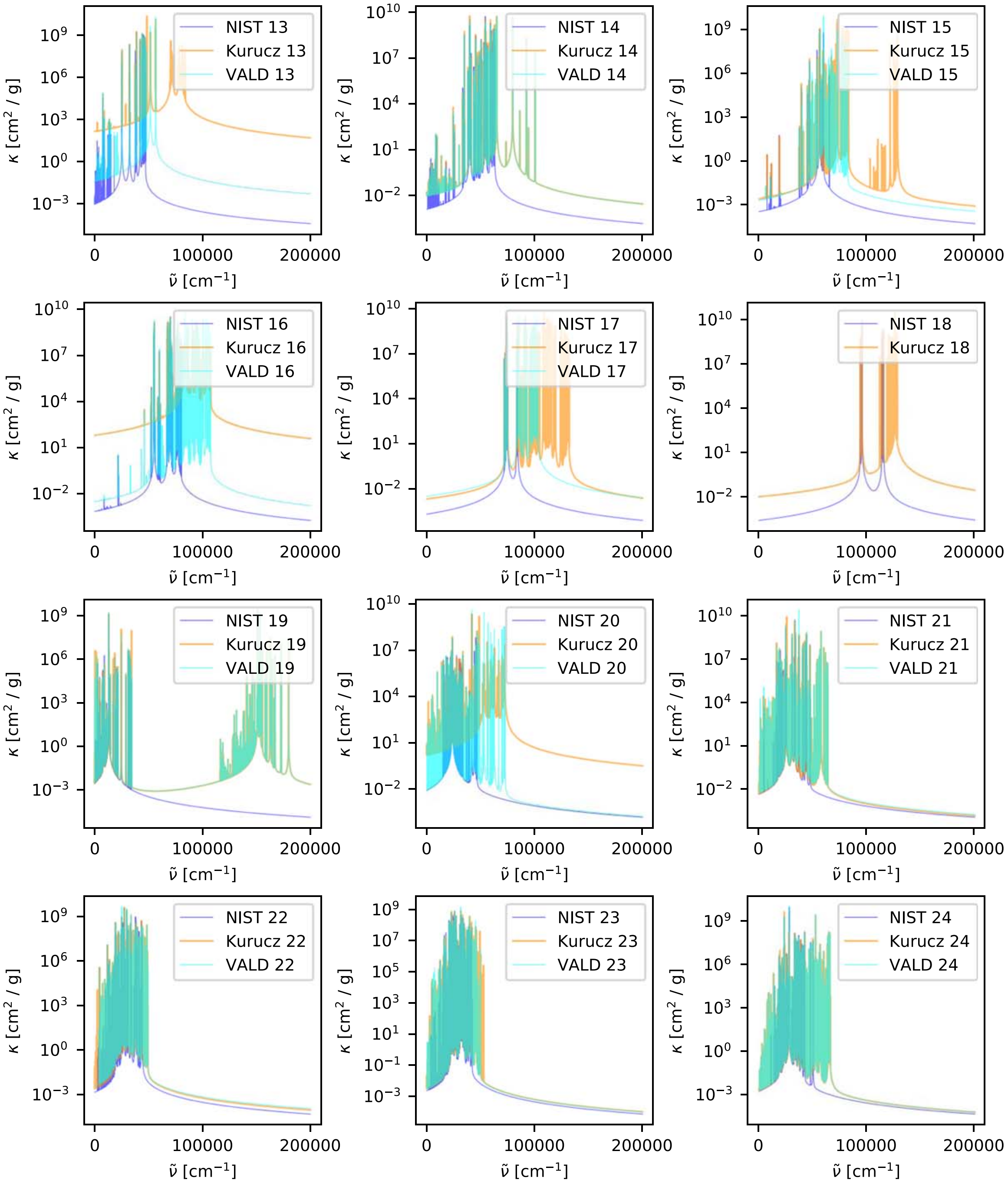


Figure A2. Comparison of atomic opacities between the NIST, Kurucz, and VALD3 databases, elements 13–24. We use a temperature of 3000 K and no cutting length.

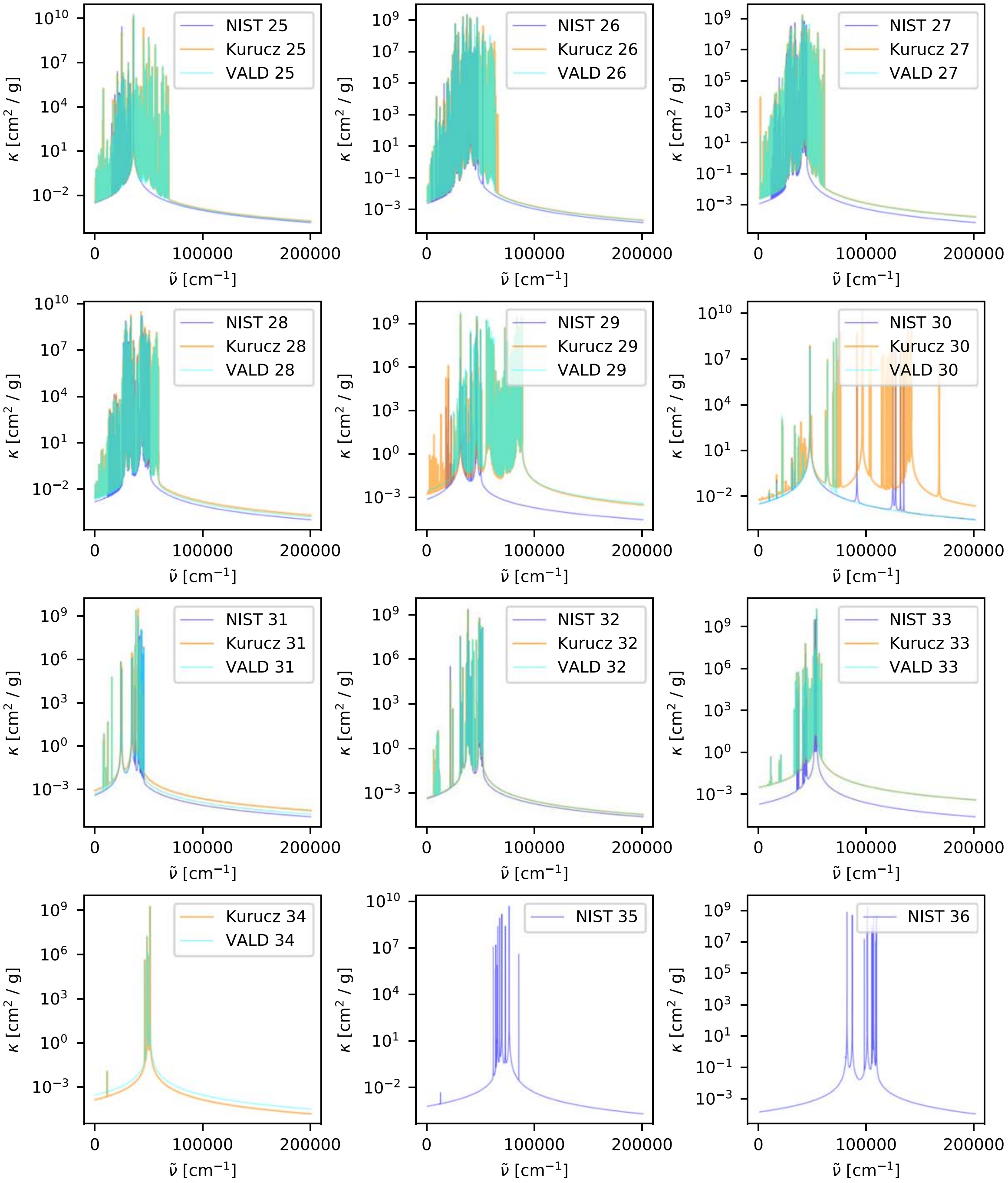


Figure A3. Comparison of atomic opacities between the NIST, Kurucz, and VALD3 databases, elements 25–36. We use a temperature of 3000 K and no cutting length.

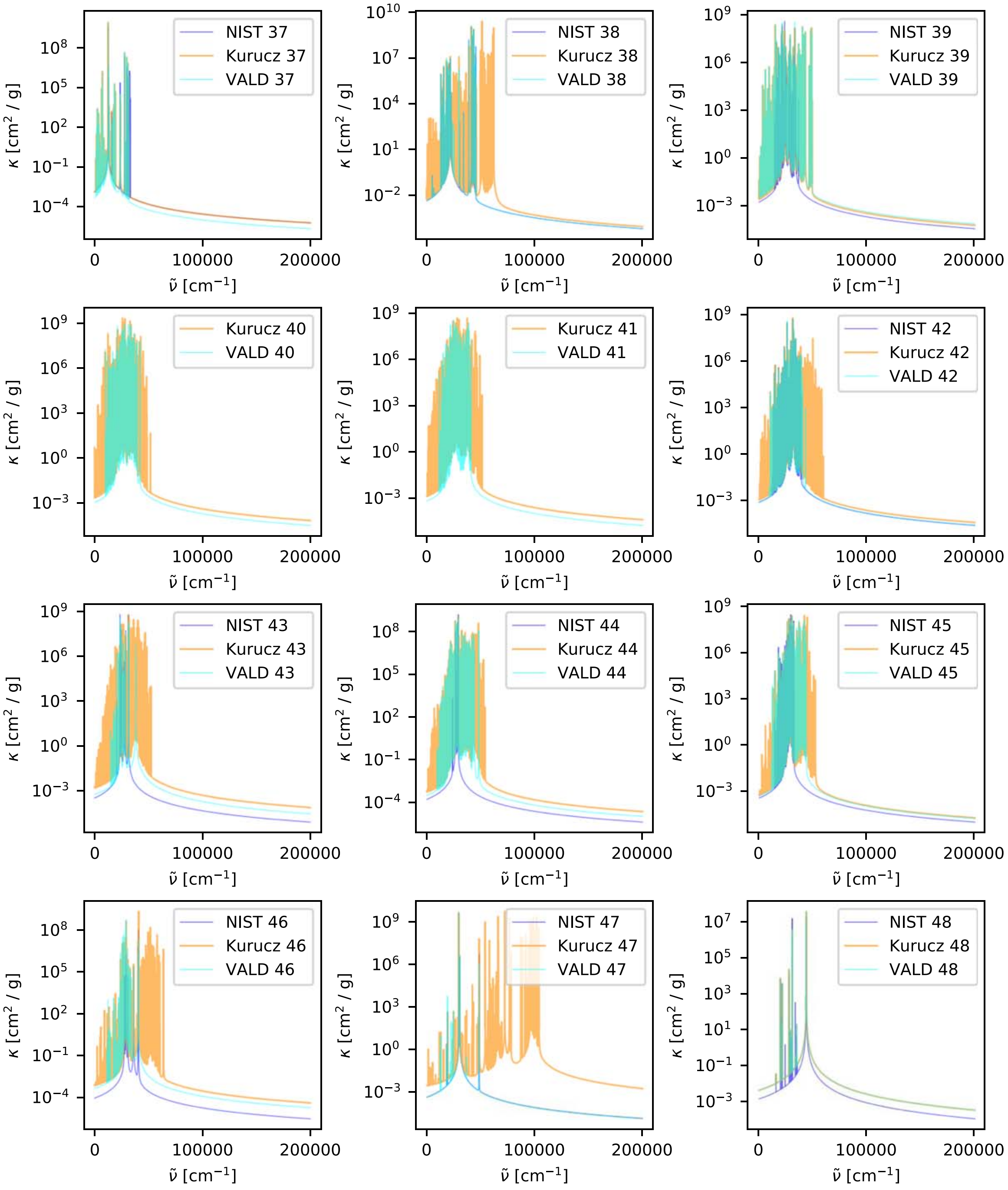


Figure A4. Comparison of atomic opacities between the NIST, Kurucz, and VALD3 databases, elements 37–48. We use a temperature of 3000 K and no cutting length.

Appendix B Molecular Opacity Comparison

The following Figures B1–B3 show a comparison between HELIOS-K cross sections and data from the ExoMol website.

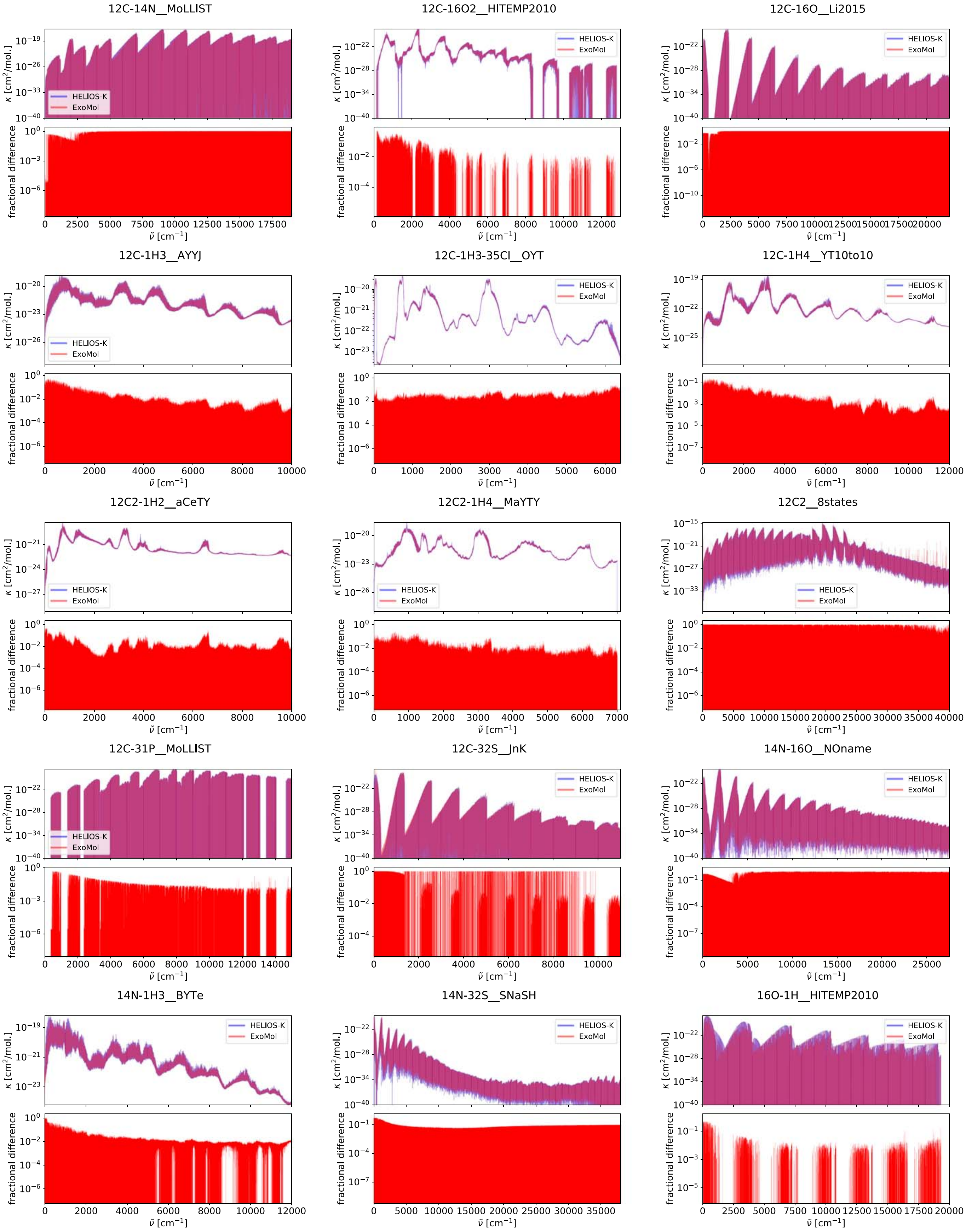


Figure B1. Comparison between cross sections calculated with HELIOS-K and from the ExoMol website. We use a wavenumber resolution of $\Delta\tilde{\nu} = 0.1$ cm^{-1} and $T = 1500$ K (1000 K for 12C2-1H4__MaYTY).

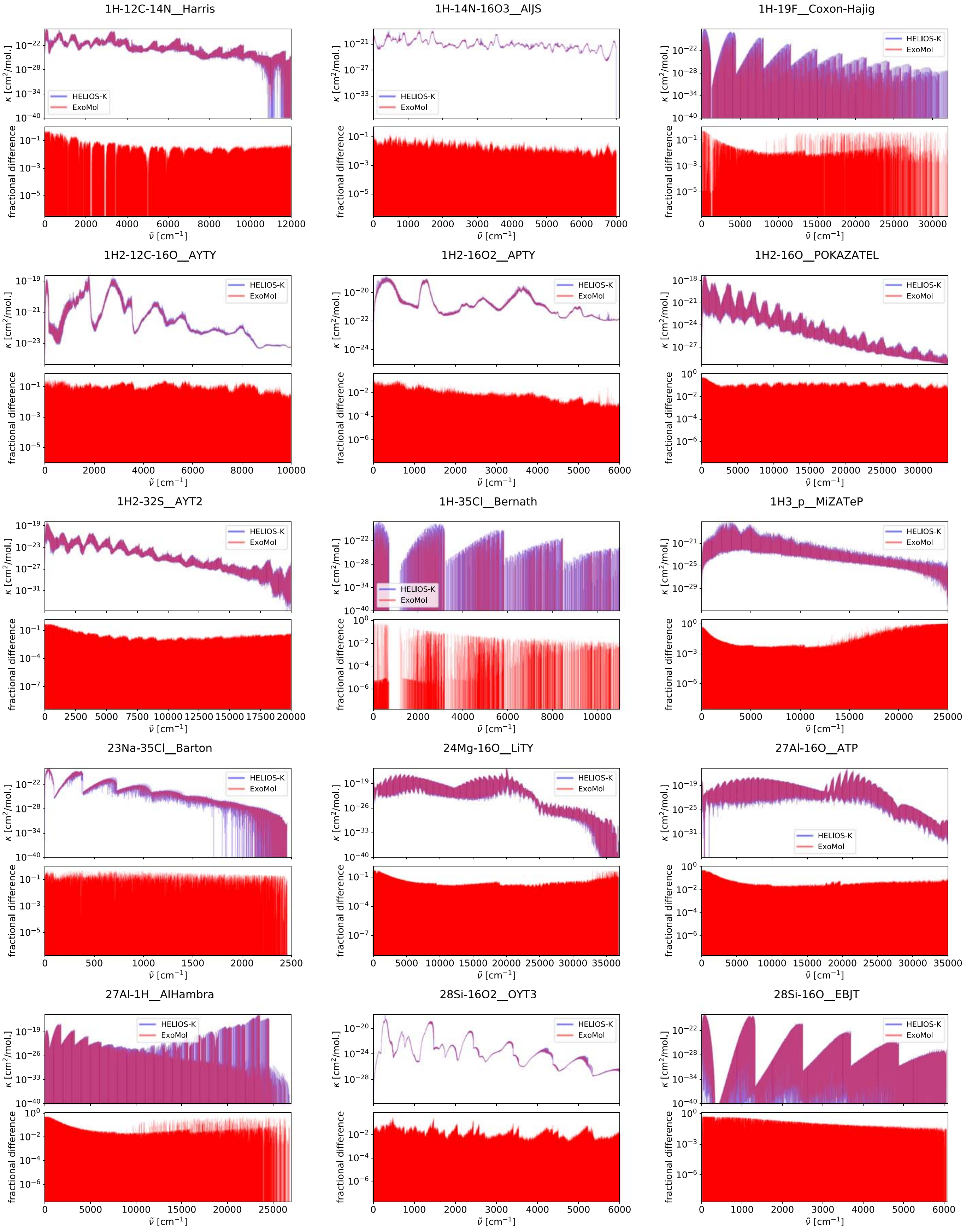


Figure B2. Comparison between cross sections calculated with HELIOS-K and from the ExoMol website. We use a wavenumber resolution of $\Delta\tilde{\nu} = 0.1$ cm^{-1} and $T = 1500$ K (500 K for 1H-14N-16O3__AIJS).

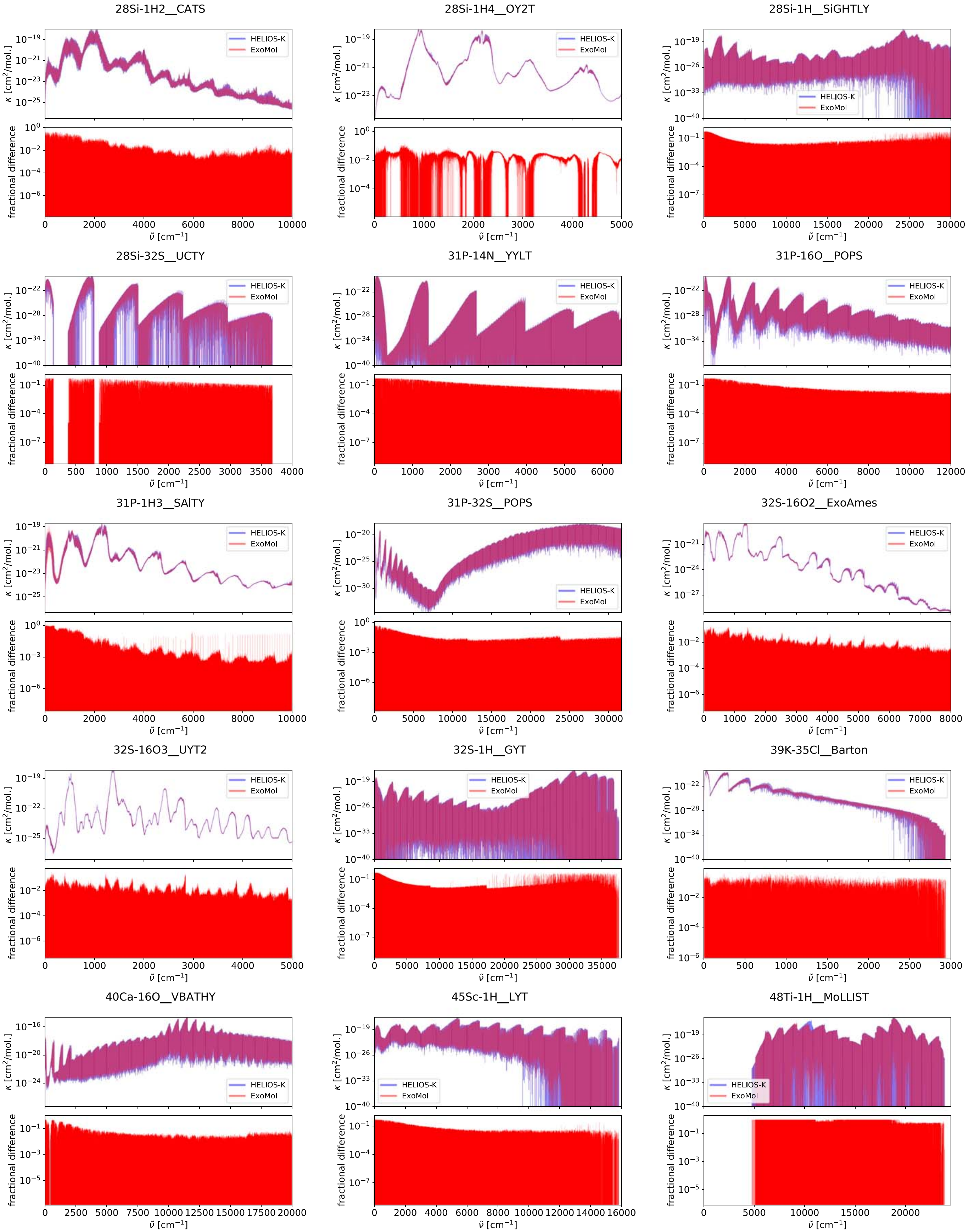
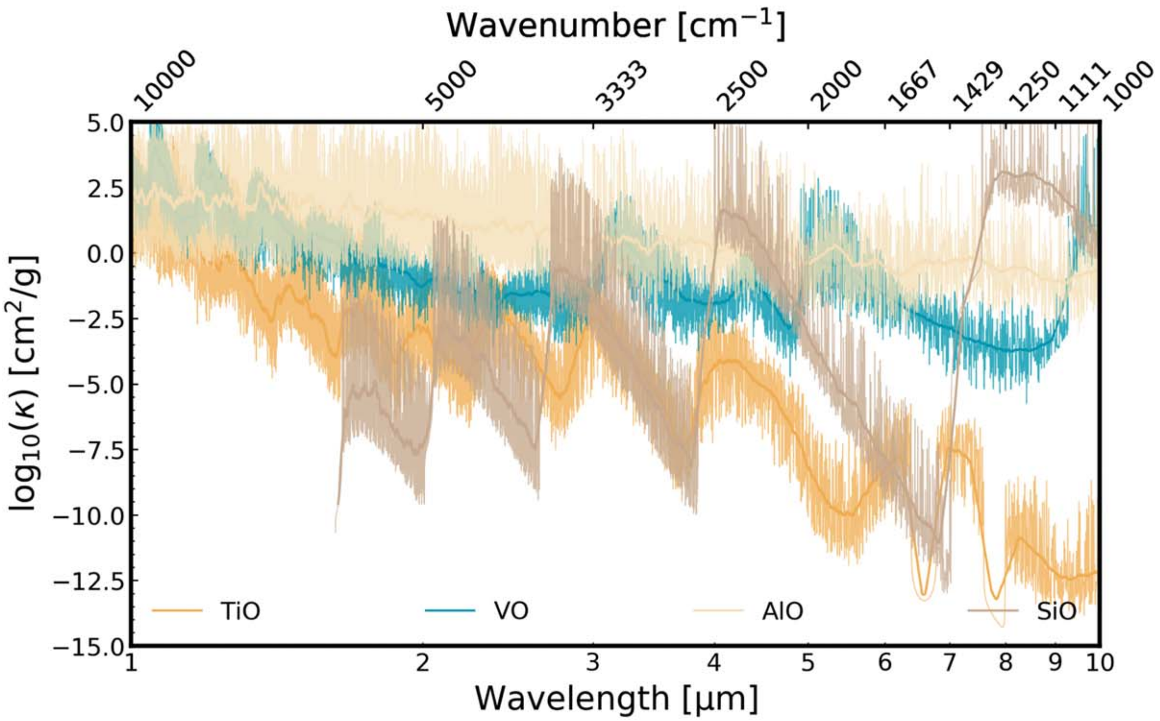


Figure B3. Comparison between cross sections calculated with HELIOS-K and from the ExoMol website. We use a wavenumber resolution of $\Delta\tilde{\nu} = 0.1$ cm^{-1} and $T = 1500$ K (1000 K for 32S-16O3__UYT2).

Appendix C
Molecular Opacity Overview

The following Figures C1–C4 show opacities for 30 commonly used molecules.



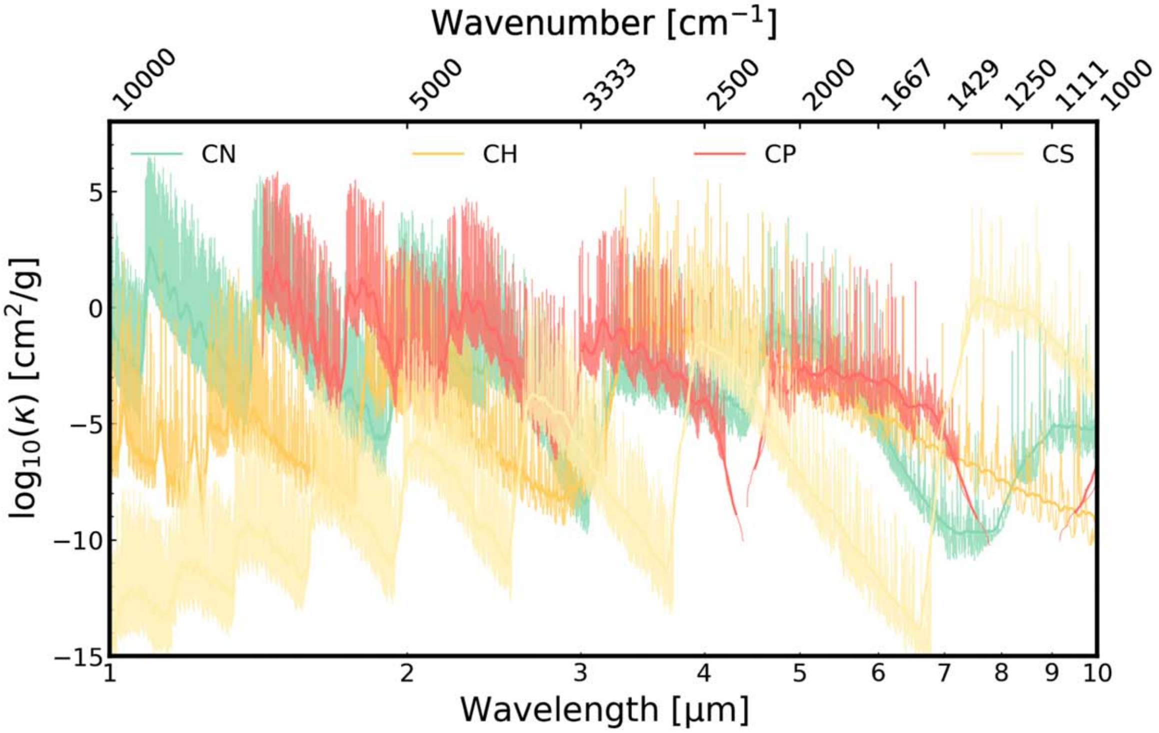


Figure C1. Examples of molecular opacities: TiO (McKemmish et al. 2019), VO (McKemmish et al. 2016), AlO (Patrascu et al. 2015), SiO (Barton et al. 2013), CN (Brooke et al. 2014b), CH (Masseron et al. 2014), CP (Ram et al. 2014), and CS (Paulose et al. 2015). We use a temperature of 1500 K, pressure of 0.001 bar, and cutting length of 100 cm^{-1}.

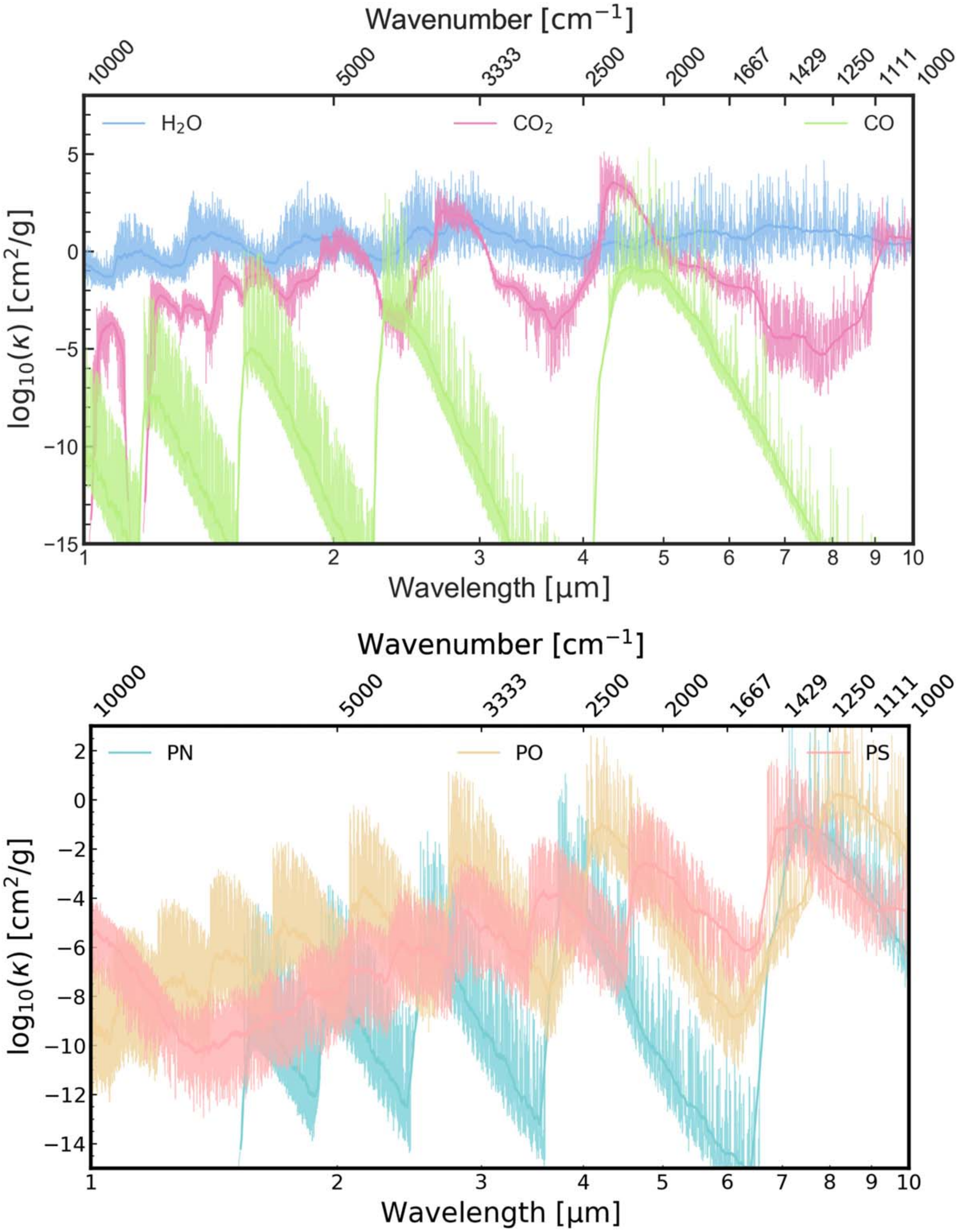


Figure C2. Examples of molecular opacities: H_2O (Polyansky et al. 2018), CO_2 (Rothman et al. 2010), CO (Li et al. 2015), PN (Yorke et al. 2014), PO (Prajapat et al. 2017), and PS (Prajapat et al. 2017). We use a temperature of 1500 K, pressure of 0.001 bar, and cutting length of 100 cm^{-1}.

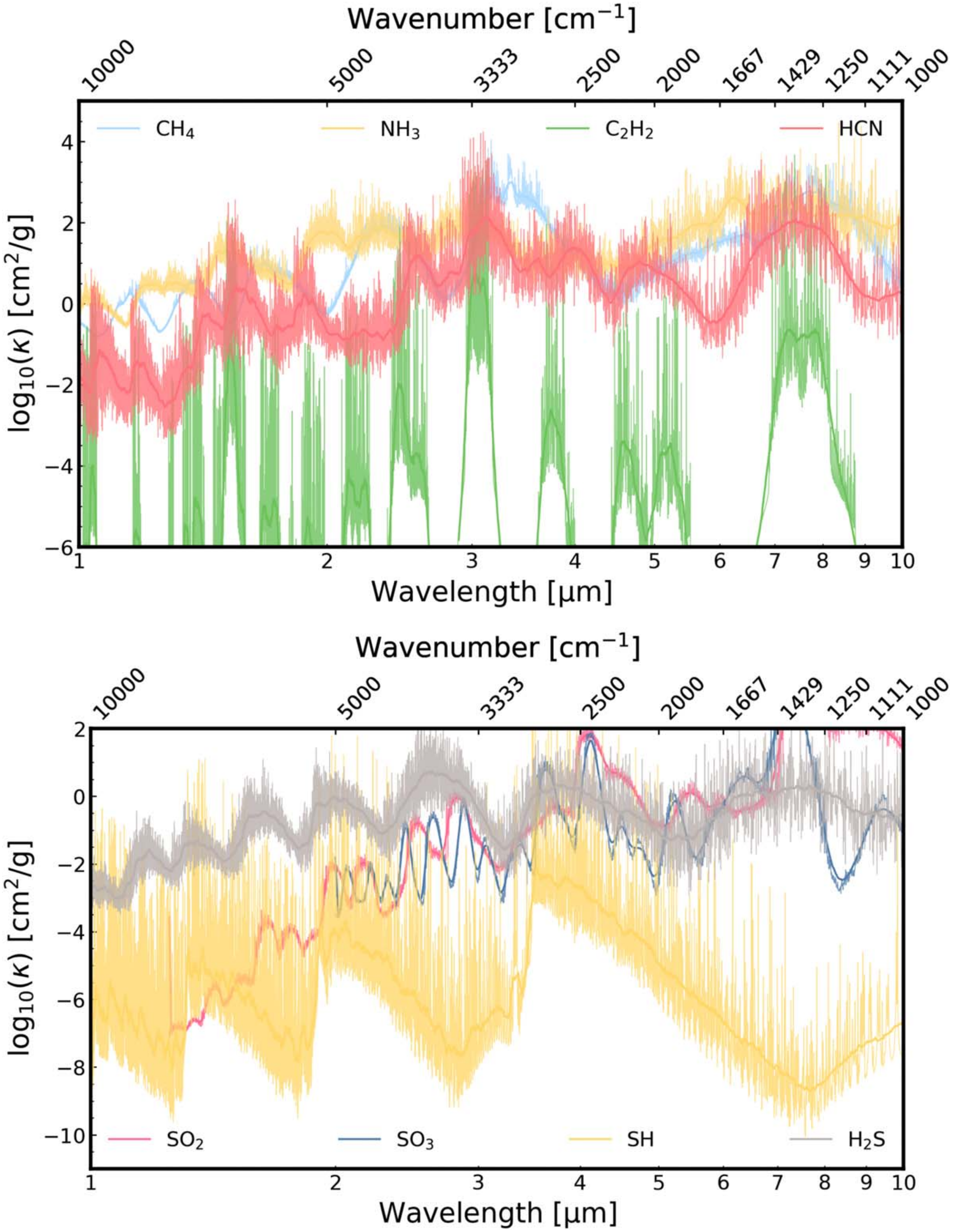


Figure C3. Examples of molecular opacities: CH_4 (Yurchenko & Tennyson 2014), NH_3 (Yurchenko et al. 2011), C_2H_2 (Chubb et al. 2020), HCN (Barber et al. 2013; Harris et al. 2006), SO_2 (Underwood et al. 2016a), SO_3 (Underwood et al. 2016b), SH (Yurchenko et al. 2018b), and H_2S (Azzam et al. 2016). We use a temperature of 1500 K (500 K for SO_3), pressure of 0.001 bar, and cutting length of 100 cm^{-1}.

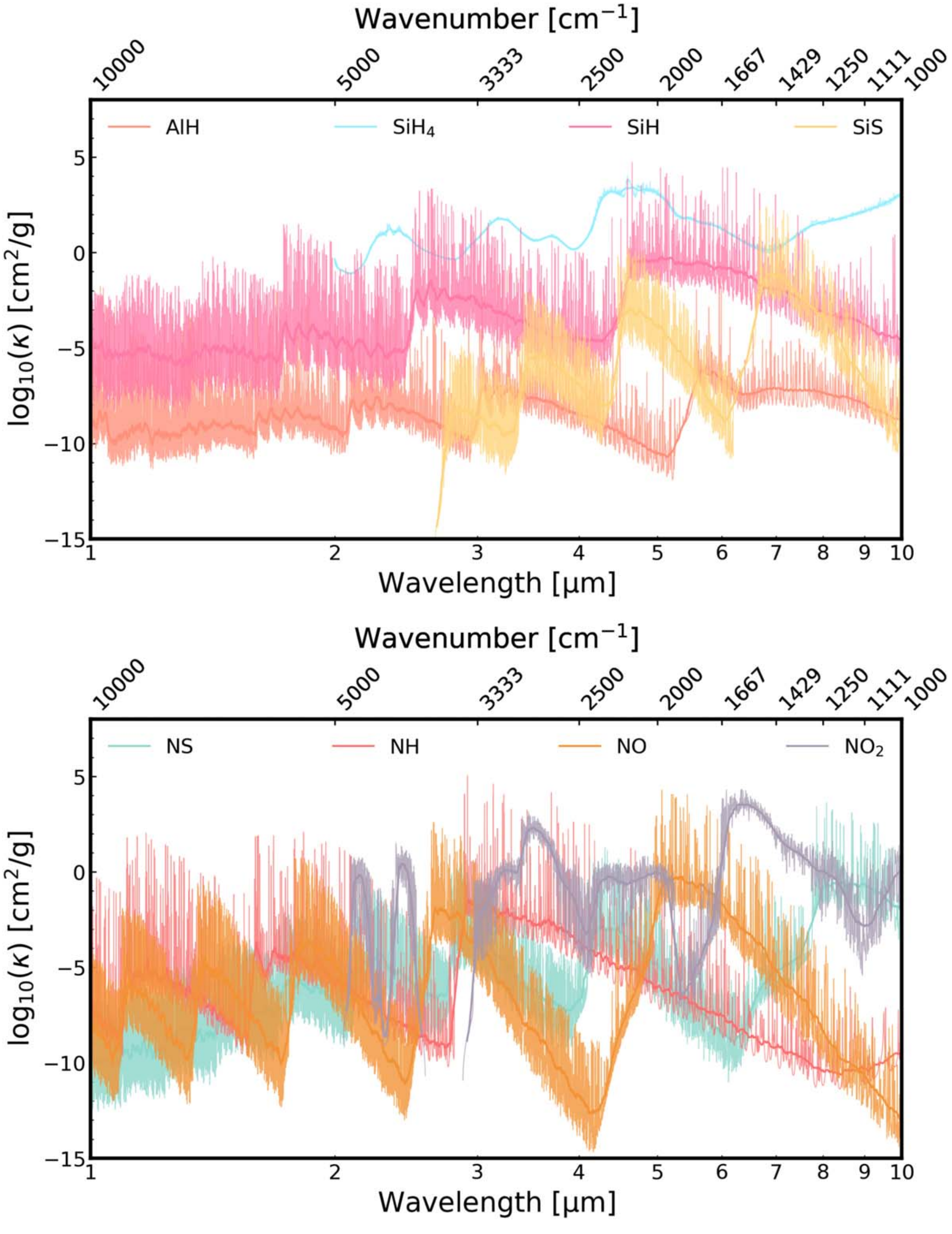


Figure C4. Examples of molecular opacities: AlH (Yurchenko et al. 2018c), SiH_4 (Owens et al. 2017), SiH (Yurchenko et al. 2017b), SiS (Upadhyay et al. 2018), NS (Yurchenko et al. 2018b), NH (Brooke et al. 2014a), NO (Wong et al. 2017), and NO_2 (Rothman et al. 2010; Hargreaves et al. 2019). We use a temperature of 1500 K, pressure of 0.001 bar, and cutting length of 100 cm^{-1}.

ORCID iDs

Simon L. Grimm https://orcid.org/0000-0002-0632-4407
Matej Malik https://orcid.org/0000-0002-2110-6694
Daniel Kitzmann https://orcid.org/0000-0003-4269-3311
Andrea Guzmán-Mesa https://orcid.org/0000-0001-5762-0276
Chloe Fisher https://orcid.org/0000-0003-0652-2902
João M. Mendonça https://orcid.org/0000-0002-6907-4476
Sergey N. Yurchenko https://orcid.org/0000-0001-9286-9501
Jonathan Tennyson https://orcid.org/0000-0002-4994-5238
Nicolas Buchschacher https://orcid.org/0000-0002-3697-1541
Kevin Heng https://orcid.org/0000-0003-1907-5910

References

Allard, N. F., Kielkopf, J. F., Spiegelman, F., Tinetti, G., & Beaulieu, J. P. 2012, A&A, 543, A159

Allard, N. F., Spiegelman, F., & Kielkopf, J. F. 2016, A&A, 589, A21

Allard, N. F., Spiegelman, F., Leininger, T., & Molliere, P. 2019, A&A, 628, A120

Aller, L. H. 1963, Astrophysics. The Atmospheres of the Sun and Stars (New York: Ronald Press)

Azzam, A. A. A., Tennyson, J., Yurchenko, S. N., & Naumenko, O. V. 2016, MNRAS, 460, 4063

Barber, R. J., Strange, J. K., Hill, C., et al. 2013, MNRAS, 437, 1828

Barber, R. J., Tennyson, J., Harris, G. J., & Tolchenov, R. N. 2006, MNRAS, 368, 1087

Barton, E. J., Yurchenko, S. N., & Tennyson, J. 2013, MNRAS, 434, 1469

Brooke, J. S. A., Bernath, P. F., Western, C. M., van Hemert, M. C., & Groenenboom, G. C. 2014a, JChPh, 141, 054310

Brooke, J. S. A., Ram, R. S., Western, C. M., et al. 2014b, ApJS, 210, 23

Burrows, A., Marley, M. S., & Sharp, C. M. 2000, ApJ, 531, 438

Burrows, A., & Volobuyev, M. 2003, ApJ, 583, 985

Chubb, K. L., Rocchetto, M., Yurchenko, S. N., et al. 2021, A&A, 646, A21

Chubb, K. L., Tennyson, J., & Yurchenko, S. N. 2020, MNRAS, 493, 1531

Clough, S., Kneizys, F., & Davies, R. 1989, AtmRe, 23, 229

Dijkstra, M. 2019, Lyman-alpha as an Astrophysical and Cosmological Tool (Berlin: Springer), 1

Draine, B. T. 2011, Physics of the Interstellar and Intergalactic Medium (Princeton, NJ: Princeton Univ. Press), 57

Fano, U. 1961, PhRv, 124, 1866

Gordon, I. E., Rothman, L. S., Hill, C., et al. 2017, JQSRT, 203, 3

Grimm, S. L., & Heng, K. 2015, ApJ, 808, 182

Hargreaves, R. J., Gordon, I. E., Rothman, L. S., et al. 2019, JQSRT, 232, 35

Harris, G. J., Tennyson, J., Kaminsky, B. M., Pavlenko, Y. V., & Jones, H. R. A. 2006, MNRAS, 367, 400

Heng, K. 2017, Exoplanetary Atmospheres: Theoretical Concepts and Foundations (Princeton, NJ: Princeton Univ. Press)

Hill, C., Yurchenko, S. N., & Tennyson, J. 2013, Icar, 226, 1673

Kochanov, R., Gordon, I., Rothman, L., et al. 2016, JQSRT, 177, 15

Kramida, A., Ralchenko, Yu., Reader, J. & NIST ASD Team 2019, NIST Atomic Spectra Database (v5.7.1) (Gaithersburg, MD: National Institute of Standards and Technology), https://physics.nist.gov/asd

Kunze, H.-J. 2009, Introduction to Plasma Spectroscopy (Berlin: Springer), 156

Kurucz, R. L. 2017, CaJPh, 95, 825

Kurucz, R. L. 2018, in ASP Conf. Ser., 515, Workshop on Astrophysical Opacities (San Francisco, CA: ASP), 47

Letchworth, K. L., & Benner, D. C. 2007, JQSRT, 107, 173

Li, G., Gordon, I. E., Rothman, L. S., et al. 2015, ApJS, 216, 15

Malik, M., Grosheintz, L., Mendonça, J. M., et al. 2017, AJ, 153, 56

Malik, M., Kempton, E. M.-R., Koll, D. D. B., et al. 2019a, ApJ, 886, 142

Malik, M., Kitzmann, D., Mendonça, J. M., et al. 2019b, AJ, 157, 170

Masseron, T., Plez, B., Van Eck, S., et al. 2014, A&A, 571, A47

McKemmish, L. K., Masseron, T., Hoeijmakers, H. J., et al. 2019, MNRAS, 488, 2836

McKemmish, L. K., Yurchenko, S. N., & Tennyson, J. 2016, MNRAS, 463, 771

Merts, A. L., & Magee, N. H. J. 1972, ApJ, 177, 137

Mlawer, E. J., Payne, V. H., Moncet, J.-L., et al. 2012, RSPTA, 370, 2520

Morton, D. 2008, ApJS, 130, 403

Morton, D. 2003, ApJS, 149, 205

Owens, A., Yachmenev, A., Thiel, W., Tennyson, J., & Yurchenko, S. N. 2017, MNRAS, 471, 5025

Pakhomov, Y. V., Ryabchikova, T. A., & Piskunov, N. E. 2019, ARep, 63, 1010

Patrascu, A. T., Yurchenko, S. N., & Tennyson, J. 2015, MNRAS, 449, 3613

Paulose, G., Barton, E. J., Yurchenko, S. N., & Tennyson, J. 2015, MNRAS, 454, 1931

Paynter, D. J., & Ramaswamy, V. 2011, JGRD, 116, D20302

Polyansky, O. L., Kyuberis, A. A., Zobov, N. F., et al. 2018, MNRAS, 480, 2597

Prajapat, L., Jagoda, P., Lodi, L., et al. 2017, MNRAS, 472, 3648

Ptashnik, I., McPheat, R., Shine, K., Smith, K., & Williams, R. 2012, RSPTA, 370, 2557

Ram, R., Brooke, J., Western, C., & Bernath, P. 2014, JQSRT, 138, 107

Rothman, L., Gordon, I., Barber, R., et al. 2010, JQSRT, 111, 2139

Ryabchikova, T., Piskunov, N., Kurucz, R. L., et al. 2015, PhyS, 90, 054005

Sharp, C. M., & Burrows, A. 2007, ApJS, 168, 140

Tennyson, J., Yurchenko, S. N., Al-Refaie, A. F., et al. 2016, JMoSp, 327, 73

Tennyson, J., Yurchenko, S. N., Al-Refaie, A. F., et al. 2020, JQSRT, 255, 107228

Tsuji, T., Ohnaka, K., & Aoki, W. 1999, ApJL, 520, L119

Underwood, D. S., Tennyson, J., Yurchenko, S. N., et al. 2016a, MNRAS, 459, 3890

Underwood, D. S., Yurchenko, S. N., Tennyson, J., et al. 2016b, MNRAS, 462, 4300

Unsöld, A. 1938, Physik der Sternatmosphären: mit besonderer Berücksichtigung der Sonne (Verlag von Julius: Springer)

Upadhyay, A., Conway, E. K., Tennyson, J., & Yurchenko, S. N. 2018, MNRAS, 477, 1520

Wcislo, P., Gordon, I., Tran, H., et al. 2016, JQSRT, 177, 75

Wong, A., Yurchenko, S. N., Bernath, P., et al. 2017, MNRAS, 470, 882

Yorke, L., Yurchenko, S. N., Lodi, L., & Tennyson, J. 2014, MNRAS, 445, 1383

Yurchenko, S., Amundsen, D., Tennyson, J., & Waldmann, I. 2017a, A&A, 605, A95

Yurchenko, S. N., Al-Refaie, A. F., & Tennyson, J. 2018a, A&A, 614, A131

Yurchenko, S. N., Barber, R. J., & Tennyson, J. 2011, MNRAS, 413, 1828

Yurchenko, S. N., Bond, W., Gorman, M. N., et al. 2018b, MNRAS, 478, 270

Yurchenko, S. N., Sinden, F., Lodi, L., et al. 2017b, MNRAS, 473, 5324

Yurchenko, S. N., & Tennyson, J. 2014, MNRAS, 440, 1649

Yurchenko, S. N., Williams, H., Leyland, P. C., Lodi, L., & Tennyson, J. 2018c, MNRAS, 479, 1401

Zaghloul, M. R., & Ali, A. N. 2012, ACM Trans. Math. Softw., 38, 2