

PyExoCross: a Python program for generating spectra and cross sections from molecular line lists

Jingxin Zhang^{*}, Jonathan Tennyson^{*} and Sergei N. Yurchenko

Department of Physics and Astronomy, University College London, Gower Street, WC1E 6BT London, United Kingdom

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

PyExoCross is a Python adaptation of the ExoCross Fortran application (Yurchenko, *A&A*, **614**, A131 (2018)). PyExoCross is designed for postprocessing the huge molecular line lists generated by the ExoMol project and other similar initiatives such as the HITRAN and HITEMP databases. PyExoCross generates absorption and emission stick spectra, cross sections and other properties (partition functions, specific heats, cooling functions, lifetimes and oscillator strengths) based on molecular line lists. PyExoCross calculates cross sections with four line profiles: Doppler, Gaussian, Lorentzian and Voigt profiles in both sampling and binned methods; a number of options are available for computing Voigt profiles which we test for speed and accuracy. PyExoCross supports importing and exporting line lists in the ExoMol and HITRAN/HITEMP formats. PyExoCross also provides conversion between the ExoMol and HITRAN data format. In addition, PyExoCross has extra code for users to automate the batch download of line list files from the ExoMol database.

Key words: software; molecular data; exoplanets; line profiles; cross sections; ExoMol

1 INTRODUCTION

A major motivation of modern astronomy is the desire to characterise and model the atmospheres of the many extrasolar planets discovered over the previous three decades (Tinetti et al. 2013; Madhusudhan 2019; Tennyson & Yurchenko 2022). Doing this requires significant quantities of spectroscopic data with hot objects requiring billions of lines to accurately reproduce spectroscopic features of their atmospheres (Yurchenko et al. 2014). The ExoMol project (Tennyson & Yurchenko 2012), is systematically constructing comprehensive molecule line lists to aid the study of the atmospheres of exoplanets and other of hot objects which contain molecules such as brown dwarfs, cool stars and photon dominated regions of the interstellar medium. ExoMol systematically releases its data in a series of papers which are summarised in data release papers (Tennyson et al. 2016b, 2020). Similar endeavours include several well-known molecular spectroscopy databases used by astronomers such as HITRAN (Gordon et al. 2022), HITEMP (Rothman et al. 2010; Hargreaves et al. 2019), GEISA (Jacquinet-Husson et al. 2011), CDMS (Müller et al. 2013; Endres et al. 2016), the JPL spectroscopic catalog (Pickett et al. 1998), TheoReTS (Rey et al. 2016), and NASA Ames (Huang et al. 2021). A particular characteristic of ExoMol, which is shared by HITEMP, TheoReTS and NASA Ames, is that as the line lists are designed for studies involving hot molecules. The line lists are often huge; the ExoMol database contains in excess of 10^{12} transitions. This means for many purposes it is desirable to have a means to rapidly and flexibly post-process these transition data to provide them in forms, such as cross sections, suitable for use in astronomical models and spectroscopic studies.

The ExoMol database is very widely used by scientists studying exoplanets and has, for example, been used to create opacities for a variety of exoplanet codes, see for example Gandhi et al. (2020); Chubb et al. (2021); Grimm et al. (2021). ExoMol data has been instrumental in many molecular detections in exoplanet atmospheres such as the recent detection of CO₂ and SO₂ in the atmosphere of WASP-39b's atmosphere by Rustamkulov et al. (2023) using the James Webb Space Telescope. ExoMol is increasingly being used for studies of cool stars and brown dwarfs, as well as a variety of non-astronomical applications. Yurchenko et al. (2018a) provided a post-processing code ExoCross capable of treating data in both ExoMol and HITRAN/HITEMP formats. ExoCross is written in Fortran which makes it efficient for processing large datasets but somewhat inflexible. For example, it is hard to take part of the current ExoCross code and include as a utility in some other application; PyExoCross was developed in response to this issue.

The main practical challenge that confronts us is processing huge amounts of data because of the extremely large datasets for polyatomic molecular hot line lists. In particular, many molecules have more than one transition file in the ExoMol database, such as the parent isotopologue ²³Na¹⁶O¹H of the molecule NaOH, its OYT5 dataset due to Owens et al. (2021) containing an excess of 400 billion transitions stored in 90 transitions files, giving a total file size of 463 Gb. To overcome this problem, automation and batch data processing are important for improving

^{*} E-mail: jingxin.zhang.19@ucl.ac.uk (JZ); j.tennyson@ucl.ac.uk (JT)

efficiency and reducing errors caused by operational faults. In the PyExoCross program, the approach to working with tens of billion lines and optimising the program for high throughput involves the use of efficient parallelization and vectorization.

PyExoCross is a general Python adaptation of the ExoCross Fortran application (Yurchenko et al. 2018a) which aims to provide the same functionalities. Like ExoCross, PyExoCross is also designed to manage huge molecular line lists generated by the ExoMol project. PyExoCross calculates spectra and spectral properties on the basis of the underlying spectroscopic data including temperature- and pressure-dependent absorption and emission cross sections, temperature-dependent cooling functions, state-dependent radiative lifetimes, oscillator strengths and thermodynamic properties such as partition functions and specific heats. PyExoCross is designed to post-process data from the ExoMol database with the ExoMol, HITRAN and HITEMP databases all serving as input databases for PyExoCross. PyExoCross exports the results in the ExoMol format which can also be read by effective Hamiltonian spectral simulation program PGOPHER (Western 2017). The ExoMol and HITRAN/HITEMP formats are the line lists formats that are initially offered; these databases are characterised by very extensive wavelength coverage and well-developed data structures.

There are several similar programs available for post-processing line lists including the HITRAN interface HAPI (Kochanov et al. 2016, 2018), SPECTRAPLOT.COM (Goldenstein et al. 2017) and SPECTRA (Tennyson & Miller 1989; Tennyson et al. 1993). However these programs would struggle to process the huge line lists required for simulations of atmospheres at elevated temperatures. We note that RADIS (van den Bekerom & Pannier 2021), a fast line-by-line code for infrared emission and absorption spectra at equilibrium and non-LTE which works with HITRAN/HITEMP and ExoMol data, is designed for this purpose, as is the GPU-based opacity code of Grimm et al. (2021). PyExoCross imports the input line lists in both ExoMol (Tennyson & Yurchenko 2012; Tennyson et al. 2016b) and HITRAN (Rothman et al. 2005) formats and returns data in the ExoMol format. Future development will support more database formats in PyExoCross.

PyExoCross performs a variety of functions. It can produce molecular absorption and emission cross sections on a grid at a given set of temperatures and pressures (see Section 2.8); PyExoCross can apply a range of line profiles namely Gaussian, Lorentzian, and Voigt profiles (see Section 3). PyExoCross also implements the computations of other useful functions including partition functions (Section 2.2), specific heats (Section 2.3), cooling functions (Section 2.4), radiative lifetimes (Section 2.5), oscillator strengths (Section 2.6) and line intensities in the form of spectra (Section 2.7). PyExoCross supports converting data formats between the ExoMol and HITRAN databases (Section 2.11). PyExoCross provides some additional functionalities according to the ExoMol database including automating the batch download of the ExoMol line list files, finding the uncertainty available datasets and comparing updates of the ExoMol database (Section 3.8). The paper is organised into six sections as follows. The main functionality of PyExoCross is described in Section 2. Section 3 discusses the line profile for calculating the cross sections implemented in PyExoCross. Section 4 gives the details of the calculation steps. The units used in PyExoCross are specified in Section 5.2. Finally, Section 6 presents some conclusions. The PyExoCross manual (<https://pyexocross.readthedocs.io/>), which is given to the supplementary data and in the PyExoCross GitHub repository (<https://github.com/ExoMol/PyExoCross.git>), provides comprehensive working instructions and use case examples.

2 MAIN FUNCTIONALITY

This section introduces the main functionalities of the PyExoCross program, including partition functions, specific heats, cooling functions, radiative lifetimes, oscillator strengths, intensities, stick spectra and cross sections. Some extra functionalities include filters, data format conversion, the batch download of line list files, extracting the uncertainty available datasets and checking updates of the ExoMol database. The line profiles for calculating the cross sections will be illustrated in Section 3.

2.1 ExoMol format

ExoMol format (Tennyson et al. 2013) assumes that a line list consists of a States file (.states) and Transition files (.trans). A State file contains the state IDs, energy term values, total degeneracies, total angular momenta, uncertainties, lifetimes, Lande- g factors and any other quantum state descriptions. As an example, an excerpt from the recently updated States file for $^{27}\text{Al}^1\text{H}$ (Yurchenko et al. 2018b) is shown in Table 1. Transitions files consist of the upper IDs, lower IDs, the Einstein A -coefficient and (optionally) reference transition wavenumbers. An extract of a transition file for $^{27}\text{Al}^1\text{H}$ is shown in Table 2.

2.2 Partition functions

Although crucial for hot molecular models, partition functions are not always straightforward to determine (Neale & Tennyson 1995; Sousa-Silva et al. 2014). Irwin (1981), Sauval & Tatum (1984), Barklem & Collet (2016) and Gamache et al. (2017) present compilations of partition functions for key astrophysical species. The partition function $Q(T)$ can be defined as the sum over states:

$$Q(T) = \sum_n g_n^{\text{tot}} e^{-c_2 \tilde{E}_n / T}, \quad (1)$$

where $g_n^{\text{tot}} = g_n^{\text{ns}}(2J_n + 1)$ is the total degeneracy. g_n^{ns} is the nuclear-spin statistical weight factor; note that ExoMol and HITRAN follow the “physicist’s” convention and use the full nuclear spin degeneracy for g_n^{ns} , see Pavlenko et al. (2020) for a discussion of this. J_n is the corresponding total angular momentum. $c_2 = hc/k_B$ is the second radiation constant (cm·K) where h is the Planck constant (erg·s), c is the

Table 1. Extract from the .states states file of the $^{27}\text{Al}^1\text{H}$ line list (Yurchenko et al. 2018b).

i	\bar{E} (cm $^{-1}$)	g	J	unc (cm $^{-1}$)	τ (s)	+/-	e/f	State	v	$ \Lambda $	$ \Sigma $	$ \Omega $	Ecalc (cm $^{-1}$)	Source Label
1	0.000000	12	0	0.000001	inf	+	e	X(1Sigma+)	0	0	0	0	0.000000	Ma
2	1625.061501	12	0	0.000597	4.9288E-03	+	e	X(1Sigma+)	1	0	0	0	1625.069321	Ma
3	3194.213550	12	0	0.000365	2.6227E-03	+	e	X(1Sigma+)	2	0	0	0	3194.213685	Ma
4	4708.808866	12	0	0.000196	1.8633E-03	+	e	X(1Sigma+)	3	0	0	0	4708.817022	Ma
5	6170.189966	12	0	0.029900	1.4910E-03	+	e	X(1Sigma+)	4	0	0	0	6170.193041	Ma
6	7579.564189	12	0	0.250000	1.2740E-03	+	e	X(1Sigma+)	5	0	0	0	7579.564189	Ca
7	8938.046805	12	0	0.300000	1.1350E-03	+	e	X(1Sigma+)	6	0	0	0	8938.046805	Ca
8	10246.644027	12	0	0.350000	1.0410E-03	+	e	X(1Sigma+)	7	0	0	0	10246.644027	Ca
9	11506.245734	12	0	0.400000	9.7520E-04	+	e	X(1Sigma+)	8	0	0	0	11506.245734	Ca
10	12717.633797	12	0	0.450000	9.2834E-04	+	e	X(1Sigma+)	9	0	0	0	12717.633797	Ca

i : State counting number;

\bar{E} : Term value (in cm $^{-1}$);

g_{tot} : Total state degeneracy;

J : Total angular momentum quantum number;

unc: Estimated uncertainty of energy level (in cm $^{-1}$);

τ : Lifetime (in s $^{-1}$);

+/-: Total parity;

e/f: Rotationless parity;

State: Electronic term value;

v : Vibrational quantum number;

$|\Lambda|$: Absolute value of the projection of electronic angular momentum;

$|\Sigma|$: Absolute value of the projection of the electronic spin;

$|\Omega|$: Absolute value of the projection of the total angular momentum;

Ecalc: Calculated state energy (in cm $^{-1}$);

Source Label: method used to generate term value.

Table 2. Extract from the .trans transitions file of the $^{27}\text{Al}^1\text{H}$ line lists (Yurchenko et al. 2018b).

f	i	A_{fi} (s $^{-1}$)	$\bar{\nu}_{fi}$ (cm $^{-1}$)
594	598	9.1644e-14	3.731850
60	28	1.7183e-05	4.209495
59	27	2.1667e-05	4.325081
58	26	3.2749e-05	4.605786
57	25	4.9130e-05	4.917133
56	24	7.0449e-05	5.228273
54	23	9.6527e-05	5.537220
53	22	1.2674e-04	5.844347
51	21	1.5996e-04	6.150075

f : Upper state counting number;

i : Lower state counting number;

A_{fi} : Einstein A-coefficient in s $^{-1}$;

$\bar{\nu}_{fi}$: Transition wavenumber in cm $^{-1}$. (This column is optional).

velocity of light (cm/s) and k_B is the Boltzmann constant (erg/K). $\bar{E}_n = E_n/hc$ is the energy term value (cm $^{-1}$). T is the temperature (K). PyExoCross provides the functionality to compute molecular a partition function with Eq. (1) using the energy term values stored in the corresponding States file of a line list in question on a grid of temperatures.

In spectroscopic applications, where a partition function $Q(T)$ is required, it can be read directly from an ExoMol partition function (.pf) file; this is recommended as in some cases these partition functions have improved accuracy compared to the simple sum of over the states in the states file. If the .pf file is not available, $Q(T)$ for a given temperature can be computed using Eq. (1) as part of the spectral simulations.

Table 3 specifies the partition function file format and Table 4 gives an example of the partition function file. Figure 1 displays examples of partition functions computed using the ExoMol line lists with temperatures from 0 K to 5000 K. Here and elsewhere examples are based on use of the following line lists: H $_2$ O POKAZATEL (Polyansky et al. 2018), $^{12}\text{C}^{16}\text{O}_2$ UCL-4000 (Yurchenko et al. 2020), $^{32}\text{S}^{16}\text{O}_2$ ExoAmes (Tobias et al. 2018), $^1\text{H}^{12}\text{C}^{14}\text{N}$ (Barber et al. 2014), $^{12}\text{C}^{16}\text{O}$ Li2015 (Li et al. 2015; Somogyi et al. 2021) and NO XABC (Qu et al. 2021).

PyExoCross provides various ways to obtain the partition function: reading the local .pf file or the ExoMol website directly, computing the partition function from the ExoMol .states or by reading it from the HITRAN website. We note that recent analysis by Wang et al. (2023) suggests that the maximum temperature quoted by HITRAN for a converged partition function is, in many cases, too high.

Table 3. Specification of the .pf partition function file format.

Field	Fortran Format	C Format	Description
T	F8.1	%8.1f	Temperature in K
$Q(T)$	F15.4	%15.4f	Partition function (dimensionless)

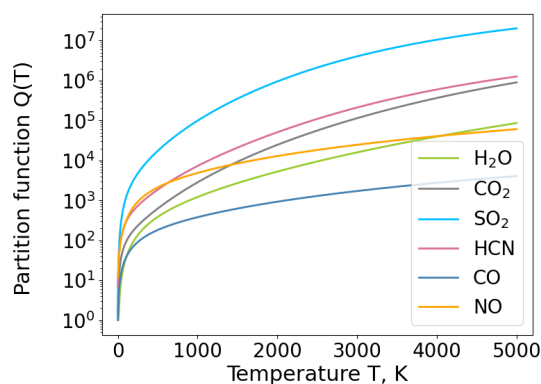
Fortran format: (F8.1, 1x, F15.4).

Table 4. Extract from the .pf partition function file generated from the $^{27}\text{Al}^1\text{H}$ line lists (Yurchenko et al. 2018b).

T	$Q(T)$
1.0	12.0000
2.0	12.0042
3.0	12.0858
4.0	12.3885
5.0	12.9621
6.0	13.7647
7.0	14.7312
8.0	15.8065

T : Temperature in K;

$Q(T)$: Partition function (dimensionless).

**Figure 1.** Partition functions of $^1\text{H}_2^1\text{O}$, $^{12}\text{C}^{16}\text{O}_2$, $^{32}\text{S}^{16}\text{O}_2$, $^1\text{H}^{12}\text{C}^{14}\text{N}$, $^{12}\text{C}^{16}\text{O}$ and $^{14}\text{N}^{16}\text{O}$ with temperatures from 0 K to 5000 K from the ExoMol database.

2.3 Specific heats

The molar specific heat ($\text{J K}^{-1} \text{mol}^{-1}$) at constant pressure C_p for a given temperature using the molecule partition function as given by Martin et al. (1992): is defined calculated on a grid of

$$C_p(T) = R \left[\frac{Q''}{Q} - \left(\frac{Q'}{Q} \right)^2 \right] + \frac{5R}{2}. \quad (2)$$

where R is the molar gas constant ($= 8.31446261815324 \text{ J K}^{-1} \text{mol}^{-1}$). The partition function $Q(T)$ is defined in Eq. (1). The first two moments of the partition function can be evaluated directly from the energy term values :

$$Q'(T) = T \frac{dQ}{dT} = \sum_n g_n^{\text{tot}} \left(\frac{c_2 \tilde{E}_n}{T} \right) \exp \left(-\frac{c_2 \tilde{E}_n}{T} \right), \quad (3)$$

$$Q''(T) = T^2 \frac{d^2 Q}{dT^2} + 2Q' = \sum_n g_n^{\text{tot}} \left(\frac{c_2 \tilde{E}_n}{T} \right)^2 \exp \left(-\frac{c_2 \tilde{E}_n}{T} \right).$$

PyExoCross uses these equation in connection with the molecular States file to compute specific heats on a grid of temperatures. Table 5 lists the specific heat file format and Table 6 gives an example of the specific heat file of AlH. Figure 2 shows the specific heats of NO, MgH and AlH at $T \in [0, 5000]$ K.

We note that following the work of Wang et al. (2023) a near complete set of specific heats are now available on the ExoMol website.

Table 5. Specification of the .cp specific heat file format.

Field	Fortran Format	C Format	Description
T	F8.1	%8.1f	Temperature in K
$C_p(T)$	F15.4	%15.4f	Specific heat (dimensionless)

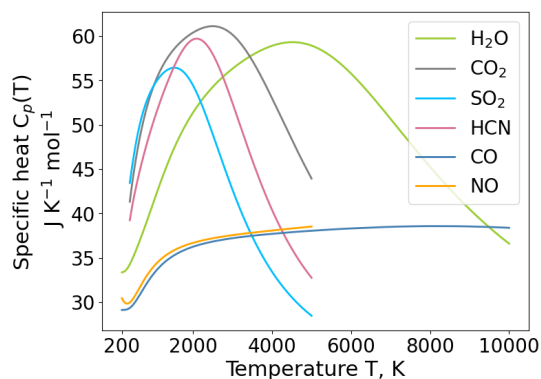
Fortran format: (F8.1, 1x, F15.4).

Table 6. Extract from the .cp specific heat file generated from the $^{27}\text{Al}^1\text{H}$ line lists (Yurchenko et al. 2018b).

T	$C_p(T)$
200.0	29.1538
201.0	29.1545
202.0	29.1553
203.0	29.1560
204.0	29.1567
205.0	29.1575
206.0	29.1583
207.0	29.1591
208.0	29.1955
209.0	29.1973

T : Temperature in K;

$C_p(T)$: Specific heat ($\text{J K}^{-1} \text{mol}^{-1}$).

**Figure 2.** Specific heats of $^1\text{H}^16\text{O}$, $^{12}\text{C}^{16}\text{O}_2$, $^{32}\text{S}^{16}\text{O}_2$, $^1\text{H}^{12}\text{C}^{14}\text{N}$, $^{12}\text{C}^{16}\text{O}$ and $^{14}\text{N}^{16}\text{O}$ with temperatures from 200 K to 10000 K or T_{max} for the given line list, using the ExoMol line lists.

2.4 Cooling functions

Cooling functions play an important role in a whole range of astronomical environments ranging from the early Universe to the interstellar medium, star formation and issues with planetary stability (Wang et al. 2014; Gérard et al. 2023; Nguyen et al. 2022; Coppola et al. 2011; Miller et al. 2013). The temperature-dependent cooling function $W(T)$ (units $\text{erg (s molecule str)}^{-1} = 10^{-7} \text{ Watts (molecule str)}^{-1}$) is the total energy per unit solid angle (in steradians) emitted by a molecule at temperature T (Melnikov et al. 2016) produced by the emissivity (Neale et al. 1996):

$$W(T) = \frac{1}{4\pi Q(T)} \sum_{f,i} A_{fi} hc \bar{\nu}_{fi} g'_f e^{-c_2 \tilde{E}'_f / T}. \quad (4)$$

In Eq. (4), $Q(T)$ is the partition function at temperature T ; g'_f is the upper state degeneracy and \tilde{E}'_f is the upper state energy which can be read from the .states states file; A_{fi} and $\bar{\nu}_{fi}$ are the Einstein A -coefficient (s^{-1}) and transition wavenumber (cm^{-1}) in the .trans transitions file (see Table 9 and Table 2). PyExoCross requires the States files, Transitions files and partition function file to compute $W(T)$ on a grid of temperatures. Although some smaller transition files provide the wavenumbers as column 4 (see below), it is important that they evaluated using the upper and lower state term values from the States files as given by

$$\bar{\nu}_{fi} = \tilde{E}'_f - \tilde{E}''_i. \quad (5)$$

Table 7 and Table 8 give the file format and an example of the cooling function file. Figure 3 shows the temperature-dependent cooling function of NO, MgH and AlH from the ExoMol database.

Table 7. Specification of the .cf cooling function file format.

Field	Fortran Format	C Format	Description
T	F8.1	%8.1f	Temperature in K
$W(T)$	ES12.4	%12.4E	Cooling function in $\text{erg (s molecule str)}^{-1}$

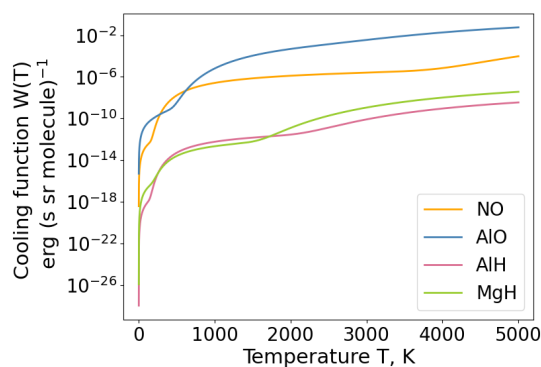
Fortran format: (F8.1, 1x, ES12.4).

Table 8. Extract from the .cf cooling function file generated from the $^{27}\text{Al}^1\text{H}$ line lists (Yurchenko et al. 2018b).

T	$W(T)$
1.0	1.04219598E-28
2.0	8.95038739E-25
3.0	1.82111455E-23
4.0	8.06932723E-23
5.0	1.94455237E-22
6.0	3.52589825E-22
7.0	5.56853398E-22
8.0	8.18964607E-22
9.0	1.15275411E-21
10.0	1.56910825E-21

T : Temperature in K;

$W(T)$: Cooling function in $\text{erg (s molecule str)}^{-1}$.

**Figure 3.** The temperature-dependent cooling functions of $^{14}\text{N}^{16}\text{O}$, $^{27}\text{Al}^{16}\text{O}$, $^{27}\text{Al}^1\text{H}$ and $^{24}\text{Mg}^{16}\text{O}$ with temperatures from 0 K to 5000 K using the ExoMol line lists.

2.5 Radiative lifetimes

Radiative life times are important for determining the critical density and as they represent a competing process to collisional thermalisation in environments including plasmas such as the interstellar medium. The radiative lifetime τ_i (s) of state i can be calculated by inverting the sum of Einstein A -coefficients A_{fi} (s^{-1}) with $\tilde{E}_f < \tilde{E}_i$ from the (.trans) transitions file(s) (see Table 2) (Tennyson et al. 2016a):

$$\tau_i = \frac{1}{\sum_f A_{fi}}. \quad (6)$$

To calculate the state-dependent lifetime, the process has been divided into three steps. Initially, transitions are reorganised with different upper state categories by `pandas.DataFrame.groupby` (McKinney et al. 2010). Each upper state f corresponds to one or more than one Einstein A -coefficient A_{fi} . Secondly, the sum over of the Einstein A -coefficients per upper state is computed. Finally, the inverse of each sum is calculated. In principle, τ_i can be calculated for each state i in the states file. In practice, however, the .trans files are not always complete. In this case the lifetime is considered to be undefined with NaN. Some lowest states, particularly for high symmetry molecules such methane or H_3^+ , have infinite lifetimes, for which 'inf' is used. The inclusion of lifetimes is being made a standard part of the (recommended) states files in ExoMol as part of the generalisation of the ExoMol data format to allow for predissociation lifetimes (Tennyson et al. 2023).

PyExoCross adds the lifetimes to the states file in the ExoMol data format. PyExoCross also provides a user option to save either an uncompressed .states file or a compressed .states.bz2 file. Figure 4 (lefthand panel) shows states-dependent lifetimes for different molecules while the righthand panel shows the lifetimes computed from AIH line list. These lifetimes are fully state specific; recently Owens et al. (2023) used ExoMol data to construct a lifetime database (LiDB) for plasma modelling which stores lifetimes and decay routes for vibronic states by lumping them over rotational and fine structure states.

Table 9. Specification of the .trans transitions file format.

Field	Fortran Format	C Format	Description
f	I12	%12d	Upper state ID
i	I12	%12d	Lower state ID
A	ES10.4	%10.4E	Einstein A -coefficient in s^{-1}
$\tilde{\nu}_{fi}$	F15.6	%15.6f	Transition wavenumber in cm^{-1} (optional)

Fortran format: (I12, 1x, I12, 1x, ES10.4, 1x, F15.6)

Table 10. Specification of the mandatory part of the .states states file with extra data options unc, τ and g .

Field	Fortran Format	C Format	Description
i	I12	%12d	State ID
E	F12.6	%12.6f	State energy in cm^{-1}
g_{tot}	I6	%6d	State degeneracy
J	I7/F7.1	%7d/%7.1f	J -quantum number (integer/half-integer)
(unc)	F12.6	%12.6f	Uncertainty in state energy in cm^{-1} (optional)
(τ)	ES12.4	%12.4E	Lifetime in s (optional)
(g)	F10.6	%10.6f	Landé g -factor (optional)
(Extra)			Extra quantum numbers, any format (optional)

ID: State identifier: a non-negative integer index, starting at 1;

J : Total angular momentum quantum, excluding nuclear spin;

Fortran format, J integer: (I12, 1x, F12.6, 1x, I6, I7, 1x, ES12.4, 1x, F10.6)

or J half-integer: (I12, 1x, F12.6, 1x, I6, F7.1, 1x, ES12.4, 1x, F10.6).

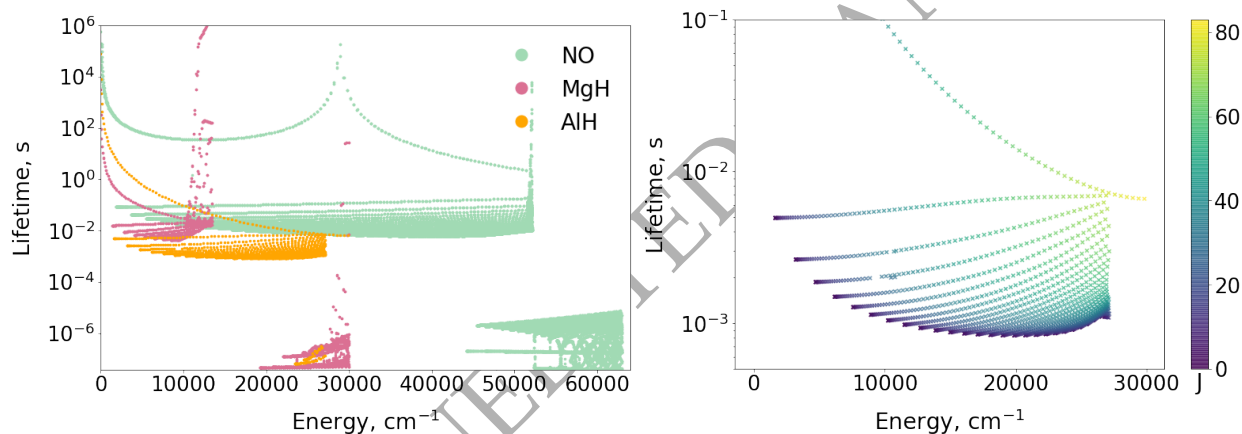


Figure 4. Left panel: Lifetime of $^{14}N^{16}O$, $^{24}Mg^1H$ and $^{27}Al^1H$ using the ExoMol line lists; Right panel: State-dependent lifetime in range of 10^{-4} s - 10^{-1} s of $^{27}Al^1H$ from the ExoMol database. The colors range from purple ($J = 0$) to yellow ($J = 80$).

2.6 Oscillator strengths

The oscillator strength is a dimensionless quantity used in spectroscopy to indicate the probability of electromagnetic radiation being emitted or absorbed in transitions between energy levels of an atom or molecule (Demtröder 1981). The fractional amount of effective electrons participating in the transition is represented by the oscillator strength f which is related to line intensity (Orloski & Trigueiros 2001). The weighted oscillator strength gf is defined as:

$$gf = \frac{g'_{tot} A_{fi}}{(c\tilde{\nu}_{fi})^2}, \quad (7)$$

where $g'_{tot} = g_{ns}(2J' + 1)$ is the degeneracy of the upper states with upper angular momentum J' ; A_{fi} and $\tilde{\nu}_{fi}$ are the Einstein A -coefficient (s^{-1}) and transition wavenumber (cm^{-1}). c is the velocity of light (cm/s).

The actual oscillator strength is defined by $f = gf/g''_{tot}$, where g''_{tot} is the degeneracy of the lower state. Table 11 and Table 12 give the file format and an example of the oscillator strength file. Figure 5 shows the oscillator strength of NO, MgH and AlH from the ExoMol database.

Table 11. Specification of the .os oscillator strength file format.

Field	Fortran Format	C Format	Description
f	I12	%12d	Upper state degeneracy
i	I12	%12d	Lower state degeneracy
gf	ES10.4	%10.4E	Oscillator strength
$\tilde{\nu}_i$	F15.6	%15.6f	Central bin wavenumber, cm^{-1} (optional)

Fortran format: (I12, 1x, I12, 1x, ES10.4, 1x, F15.6)

Table 12. Extract from the .os oscillator strength file generated from the $^{14}\text{N}^{16}\text{O}$ line lists (Qu et al. 2021).

f	i	gf	$\tilde{\nu}_i$
2237	2604	2.3938E-29	32406.776335
16945	16990	2.6529E-30	32406.778880
11519	11563	9.3397E-30	32406.843732
15241	14758	4.6102E-26	32406.846700
15986	15778	1.0011E-25	32406.866875
13921	13689	9.0563E-31	32406.888211
7437	7184	1.1891E-30	32406.897702
11504	11847	1.0124E-28	32406.925712
16602	16639	4.0684E-30	32406.926676
16353	16394	5.5179E-31	32406.942107

f : Upper state ID;

i : Lower state ID;

gf : Oscillator strength;

$\tilde{\nu}_i$: Central bin wavenumber, cm^{-1} (optional).

2.7 Line intensities and stick spectra

2.7.1 Absorption coefficient

The absorption coefficient, also known as the absorption line intensity $I_{f \leftarrow i}$ (cm molecule^{-1}), can be calculated at a general temperature T using (Jørgensen et al. 1996; Yurchenko et al. 2018a):

$$I_{f \leftarrow i} = \frac{g'_f A_{fi} e^{-c_2 \tilde{E}_i''/T} (1 - e^{-c_2 \tilde{\nu}_{fi}/T})}{8\pi c \tilde{\nu}_{fi}^2 Q(T)}, \quad (8)$$

where g'_f is the upper state degeneracy, \tilde{E}_i'' is the lower state energy, A_{fi} is the Einstein A -coefficient (s^{-1}) and $\tilde{\nu}_{fi}$ is the transition wavenumber (cm^{-1}). $c_2 = hc/k_B$ is the second radiation constant ($\text{cm}\cdot\text{K}$), where h is the Planck constant ($\text{erg}\cdot\text{s}$), c is the speed of light (cm/s) and k_B is the Boltzmann constant (erg/K), T is the temperature in K. The temperature-dependent partition function $Q(T)$ can be accessed from the partition function file (.pf). When computing $I_{f \leftarrow i}$ with PyExoCross, the energy term values are read from the ExoMol States (.states) file, which are also used to evaluate $\tilde{\nu}_{fi}^2$, the Einstein A -coefficient are read from the ExoMol Transitions (.trans) file and the partition function is either taken from the Partition function file or calculated on the fly using the energy term values and Eq. (1), as in the case for the cooling functions.

We note that the definitions of line intensities given in Eqs. (8, 9) are for 100% abundance and, unlike HITRAN and HITEMP, are not scaled to allow for isotopic fractional abundances. The results should therefore be multiplied by the appropriate fractional abundance for many applications.

2.7.2 Emission coefficient

The emission coefficient is also called the emission line intensity and emissivity ε (units $\text{erg (s molecule str)}^{-1} = 10^{-7} \text{ Watts (molecule str)}^{-1}$) (Yurchenko et al. 2018a).

$$\varepsilon(i \leftarrow f) = \frac{g'_f hc A_{fi} \tilde{\nu}_{fi} e^{-c_2 \tilde{E}_f'/T}}{4\pi Q(T)}, \quad (9)$$

where g'_f is the upper state degeneracy and \tilde{E}_f' is the upper state energy term value. As in the case of the absorption coefficients, the energy term values \tilde{E}_f' are read from the ExoMol States file, which are also used for $\tilde{\nu}_{fi}$, A_{fi} are taken from the Transition files and $Q(T)$ is from the Partition function file or computed using Eq. (1). When calculating the emission coefficient for the HITRAN database, the HITRAN format line lists file (.par) only has the lower state energy E_{lower} (E_i''). Then the upper state energy must be evaluated using Eq. (5) and the partition function is taken from the Partition function file.

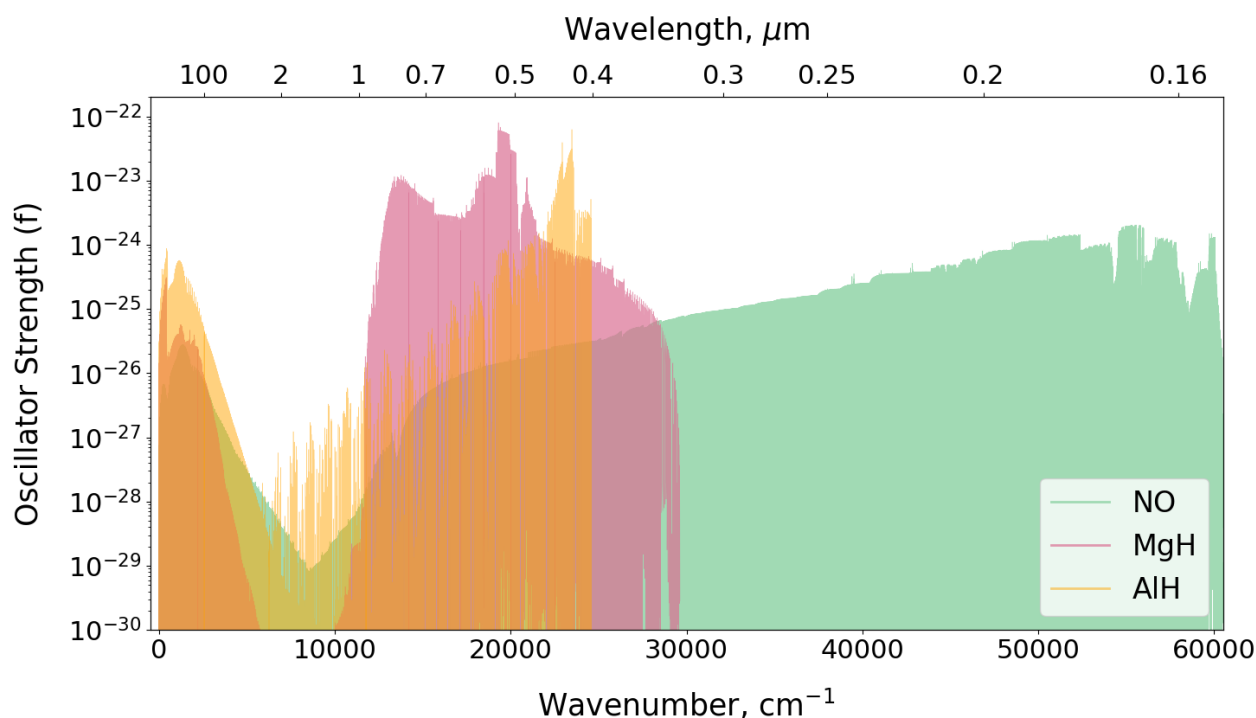


Figure 5. Oscillator strength of $^{14}\text{N}^{16}\text{O}$, $^{24}\text{Mg}^{16}\text{O}$ and $^{27}\text{Al}^1\text{H}$ from the ExoMol line lists.

Table 13. Specification of the mandatory part of the `.stick` stick spectra file with the extra columns for the optional quantum numbers.

Field	Fortran Format	C Format	Description
$\tilde{\nu}$	F12.6	%12.6f	Frequency (wavenumber) in cm^{-1}
I	ES14.8	%14.8E	Intensity in cm^{-1}
J'	I7/F7.1	%7d/%7.1f	J -quantum number (integer/half-integer) for upper state
E'	F12.4	%12.4f	Upper state energy in cm^{-1}
J''	I7/F7.1	%7d/%7.1f	J -quantum number (integer/half-integer) for lower state
E''	F12.4	%12.4f	lower state energy in cm^{-1}
(QN')			Extra quantum numbers for upper state, any format (optional)
(QN'')			Extra quantum numbers for lower state, any format (optional)

J : Total angular momentum quantum, excluding nuclear spin;

Fortran format, J integer: (I12, 1x, F12.6, 1x, I6, I7, 1x, ES12.4, 1x, F10.6)

or J half-integer: (I12, 1x, F12.6, 1x, I6, F7.1, 1x, ES12.4, 1x, F10.6).

2.7.3 Stick spectra

Stick spectra are lists of frequencies and line intensities, absorption or emission, calculated using Eqs. (8) and (9), respectively, along with their descriptions (required quantum numbers) for the upper and lower states (Yurchenko et al. 2018a). PyExoCross can apply filters of uncertainties, thresholds and quantum numbers when producing stick spectra. Table 13 gives the mandatory part of the stick (`.stick`) spectra file. Table 14 shows an example of a stick spectrum output. Lines are plotted as “sticks”, with their height given by their intensity, see Figure 6.

2.7.4 Intensity thresholds

PyExoCross provides an intensity threshold filter to skip weak lines. Using an intensity threshold can speed up the cross-section calculations or reduce the output of stick spectra calculations. In PyExoCross, the default value of the intensity threshold is $10^{-30} \text{ cm} \cdot \text{molecule}^{-1}$ but this value can be changed in the input file.

2.8 Cross sections

PyExoCross computes the cross sections (σ_{ab} or σ_{em}) for the given wavenumber range, temperature and pressure with line intensity (absorption coefficient I_{fi} in Eq. (8) or emission coefficient ϵ_{if} in Eq. (9)) and line profile ($f_{\tilde{\nu}}(\tilde{\nu})$, more details in Section 3). PyExoCross provides

Table 14. Extract from the .stick stick spectra file generated from the absorption $^{24}\text{Mg}^1\text{H}$ line list (Owens et al. 2022).

$\tilde{\nu}_{fi}$	I_{fi}	J'	\tilde{E}'	J''	\tilde{E}''	par'	e/f'	State'	v'	$ \Lambda '$	$ \Sigma '$	$ \Omega '$	par''	e/f''	State''	v''	$ \Lambda ''$	$ \Sigma ''$	$ \Omega ''$
1.242966	7.42574603E-49	20.5	10749.7776	20.5	10748.5346	-	f	X(2Sigma+)	8	0	-0.5	-0.5	+	e	X(2Sigma+)	9	0	0.5	0.5
1.252056	7.53243618E-46	20.5	10749.7776	19.5	10748.5256	-	f	X(2Sigma+)	8	0	-0.5	-0.5	+	f	X(2Sigma+)	9	0	0.5	0.5
1.337756	9.08537842E-46	21.5	10749.8724	20.5	10748.5346	-	e	X(2Sigma+)	8	0	-0.5	-0.5	+	e	X(2Sigma+)	9	0	0.5	0.5
1.569618	2.65779920E-52	36.5	12150.5019	36.5	12148.9323	-	f	X(2Sigma+)	7	0	-0.5	-0.5	+	e	X(2Sigma+)	8	0	0.5	0.5
1.597018	1.38115517E-48	36.5	12150.5019	35.5	12148.9049	-	f	X(2Sigma+)	7	0	-0.5	-0.5	+	f	X(2Sigma+)	8	0	0.5	0.5
1.816673	1.86393524E-48	37.5	12150.7490	36.5	12148.9323	-	e	X(2Sigma+)	7	0	-0.5	-0.5	+	e	X(2Sigma+)	8	0	0.5	0.5
2.859016	2.89098064E-45	0.5	10353.9808	0.5	10352.2738	-	f	X(2Sigma+)	11	0	-0.5	-0.5	+	e	X(2Sigma+)	11	0	0.5	0.5
2.862028	5.84278951E-45	1.5	10354.0094	0.5	10352.2738	-	e	X(2Sigma+)	11	0	-0.5	-0.5	+	e	X(2Sigma+)	11	0	0.5	0.5
3.589218	8.68228140E-47	33.5	11929.6911	34.5	11926.1019	-	e	X(2Sigma+)	9	0	-0.5	-0.5	+	e	X(2Sigma+)	8	0	0.5	0.5
3.606156	8.58388055E-47	32.5	11929.6121	33.5	11926.0060	-	f	X(2Sigma+)	9	0	-0.5	-0.5	+	f	X(2Sigma+)	8	0	0.5	0.5

$\tilde{\nu}_{fi}$: Absorption frequency (wavenumber) (in cm^{-1});

I_{fi} : Absorption intensity (in cm molecule^{-1});

J' and J'' : Total angular momentum quantum number for upper and lower state;

\tilde{E}' and \tilde{E}'' : Term value (in cm^{-1}) for upper and lower state;

par' and par'': Total parity for upper and lower state;

e/f' and e/f'': Rotationless parity for upper and lower state;

State' and State'': Electronic term value for upper and lower state;

v' and v'' : Vibrational quantum number for upper and lower state;

$|\Lambda|'$ and $|\Lambda|''$: Absolute value of the projection of electronic angular momentum for upper and lower state;

$|\Sigma|'$ and $|\Sigma|''$: Absolute value of the projection of the electronic spin for upper and lower state;

$|\Omega|'$ and $|\Omega|''$: Absolute value of the projection of the total angular momentum for upper and lower state;

Table 15. Specification of the .cross cross-section file format.

Field	Fortran format	C format	Description
$\tilde{\nu}_i$	F12.6	%12.6f	Central bin wavenumber, cm^{-1}
σ_i	ES14.8	%14.8E	Absorption cross section, $\text{cm}^2 \text{ molecule}^{-1}$

Fortran format: (F12.6,1x,ES14.8).

Table 16. Extract from the .cross cross-section file generated from the $^1\text{H}_2^{16}\text{O}$ POKAZATEL ExoMol line lists (Polyansky et al. 2018).

$\tilde{\nu}$ (cm^{-1})	σ_i ($\text{cm}^2 \text{ molecule}^{-1}$)
5000.000000	8.03246668E-25
5000.100000	1.84420150E-24
5000.200000	2.00318098E-23
5000.300000	3.84642766E-24
5000.400000	8.77939177E-25
5000.500000	4.33350935E-25
5000.600000	2.91607795E-25
5000.700000	2.30487176E-25
5000.800000	2.01726401E-25
5000.900000	2.32892097E-25

$\tilde{\nu}$: Central bin wavenumber, cm^{-1} ;

σ_i : Absorption cross section, $\text{cm}^2 \text{ molecule}^{-1}$.

uncertainty, threshold and quantum number filters for calculating cross sections (see Section 2.9). Figures 7 and 8 show examples of cross sections of MgH and H_2O at $P = 1$ bar and $T = 10, 300, 1000, 3000$ and 5000 K simulated using the ExoMol line list of XAB (Owens et al. 2022) and POKAZATEL (Polyansky et al. 2018). Figure 9 compares cross sections of H_2O (Polyansky et al. 2018), CO_2 (Yurchenko et al. 2020) and SO_2 (Underwood et al. 2016) at $T = 300$ and 3000 K. Table 15 shows the cross-section file format and Table 16 specifies a part of cross-section file.

2.8.1 Wavenumber grids

PyExoCross uses uniform wavenumber grids when calculating cross sections. The grids can be defined using two methods for a given wavenumber range $[\tilde{\nu}_{\min}, \tilde{\nu}_{\max}]$: (i) using the number of grid points N_{points} with $N_{\text{points}} - 1$ as the number of grid intervals or (ii) the size of

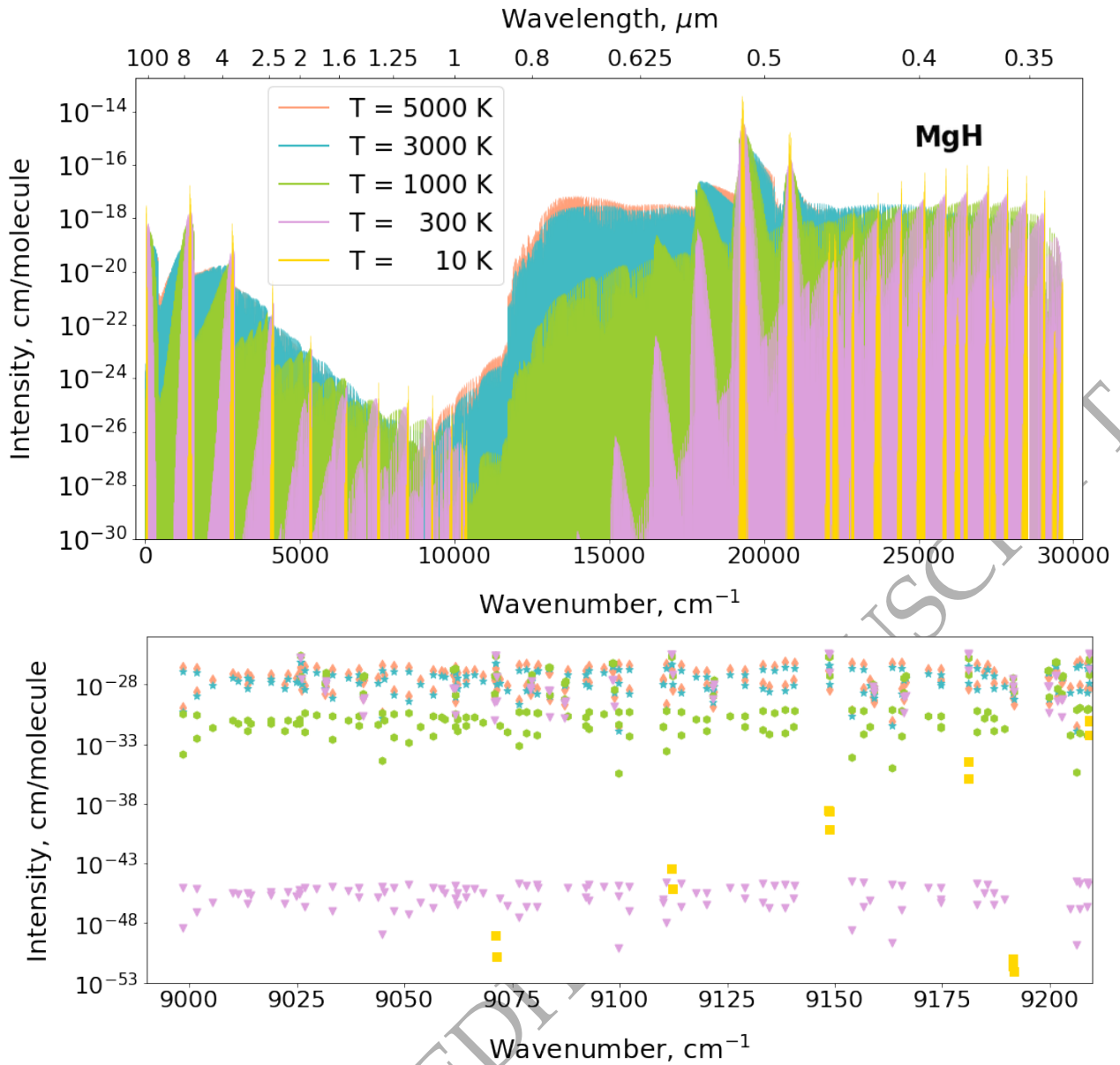


Figure 6. Stick spectra: Absorption intensities of $^{24}\text{Mg}^1\text{H}$ at $T = 10, 300, 1000, 3000$ and 5000 K simulated using the ExoMol line list of dataset XAB (Owens et al. 2022). Upper panel: Intensities in wavenumber range $[0, 30000]$ cm^{-1} ; Lower panel: Intensities zoom at $[9000, 9200]$ cm^{-1} .

the grid bin $\Delta\tilde{\nu}$. The wavenumber step size is defined as:

$$\Delta\tilde{\nu} = \frac{\tilde{\nu}_{\max} - \tilde{\nu}_{\min}}{N_{\text{points}} - 1}. \quad (10)$$

In order to let $\Delta\tilde{\nu}$ be a ‘round’ value, the number of grid points is usually set to an odd number (Yurchenko et al. 2018a).

2.8.2 Absorption and emission cross-sections

The absorption cross-section σ_{ab} at wavenumber $\tilde{\nu}$ for a transition line $f \leftarrow i$ with intensity I_{fi} (Eq. (8)) and line profile ($f_{\tilde{\nu}_{fi}}$) is given by:

$$I_{fi} = \int_{-\infty}^{\infty} \sigma_{fi}(\tilde{\nu}) d\tilde{\nu} \Rightarrow \sigma_{\text{ab}}(\tilde{\nu}) = I_{fi} f_{\tilde{\nu}_{fi}}(\tilde{\nu}). \quad (11)$$

Similarly, the emission cross-section σ_{em} at wavenumber $\tilde{\nu}$ for a transition line $i \rightarrow f$ is calculated with emission coefficient ϵ_{if} (Eq. (9)) and line profile ($f_{\tilde{\nu}_{fi}}$):

$$\epsilon_{if} = \int_{-\infty}^{\infty} \epsilon_{if}(\tilde{\nu}) d\tilde{\nu} \Rightarrow \sigma_{\text{em}}(\tilde{\nu}) = \epsilon_{if} f_{\tilde{\nu}_{if}}(\tilde{\nu}). \quad (12)$$

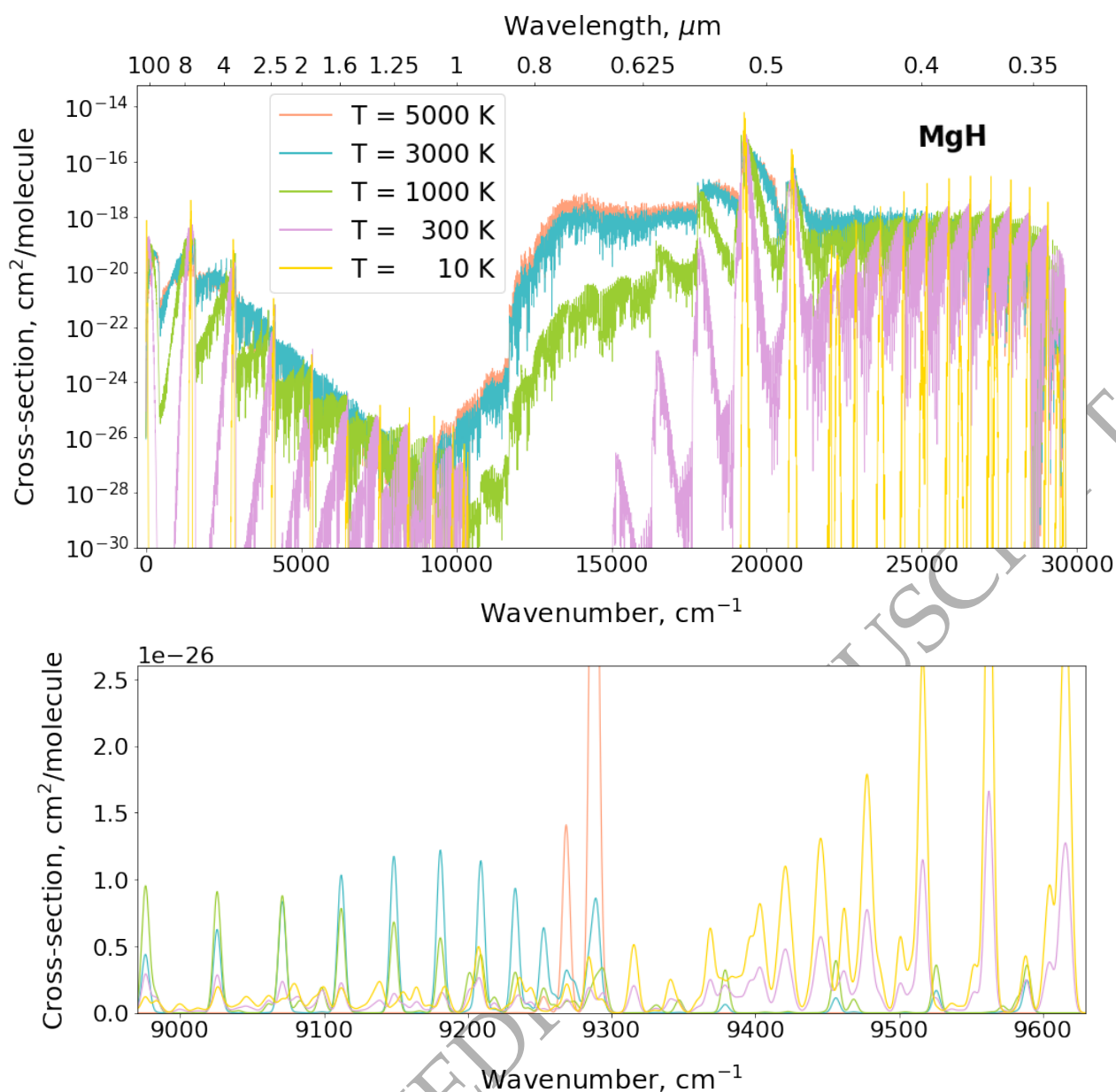


Figure 7. Cross section: Absorption spectrum of $^{24}\text{Mg-H}$ at $T = 10, 300, 1000, 3000$ and 5000 K simulated using the ExoMol line list of dataset XAB (Owens et al. 2022) broadened using a Gaussian line profile with its HWHM $\alpha_D = 3 \text{ cm}^{-1}$. Upper panel: Cross sections in wavenumber range $[0, 30000] \text{ cm}^{-1}$; Lower panel: Cross sections zoom for $[9000, 9600] \text{ cm}^{-1}$.

2.9 Filters

PyExoCross can apply filters including uncertainty, threshold, quantum number labels, formats and values based on the information given in the .states file if users require the filters.

2.9.1 Uncertainty filter

An uncertainty filter can be applied when computing stick spectra, cross sections or performing format conversion. The uncertainty refers to the accuracy of each individual transition wavenumbers. PyExoCross applies an uncertainty filter with a given value which is usually taken to be the upper bound of the selected uncertainties. Laboratory spectroscopy uses uncertainties in cm^{-1} and the uncertainties less than 0.01 cm^{-1} are usually assumed to be accurate, but even uncertainties of 0.05 cm^{-1} are probably good. Astronomers use the resolving power $R = \frac{\lambda}{\Delta\lambda}$ in wavelength as a measure of the spectral resolution of an instrument and this is approximately the same as $R = \frac{\nu}{\Delta\nu}$ in wavenumbers. $R > 100\,000$ indicates high resolution, but for some astronomy fields, $R > 50\,000$ could be seen as high enough resolution as well.

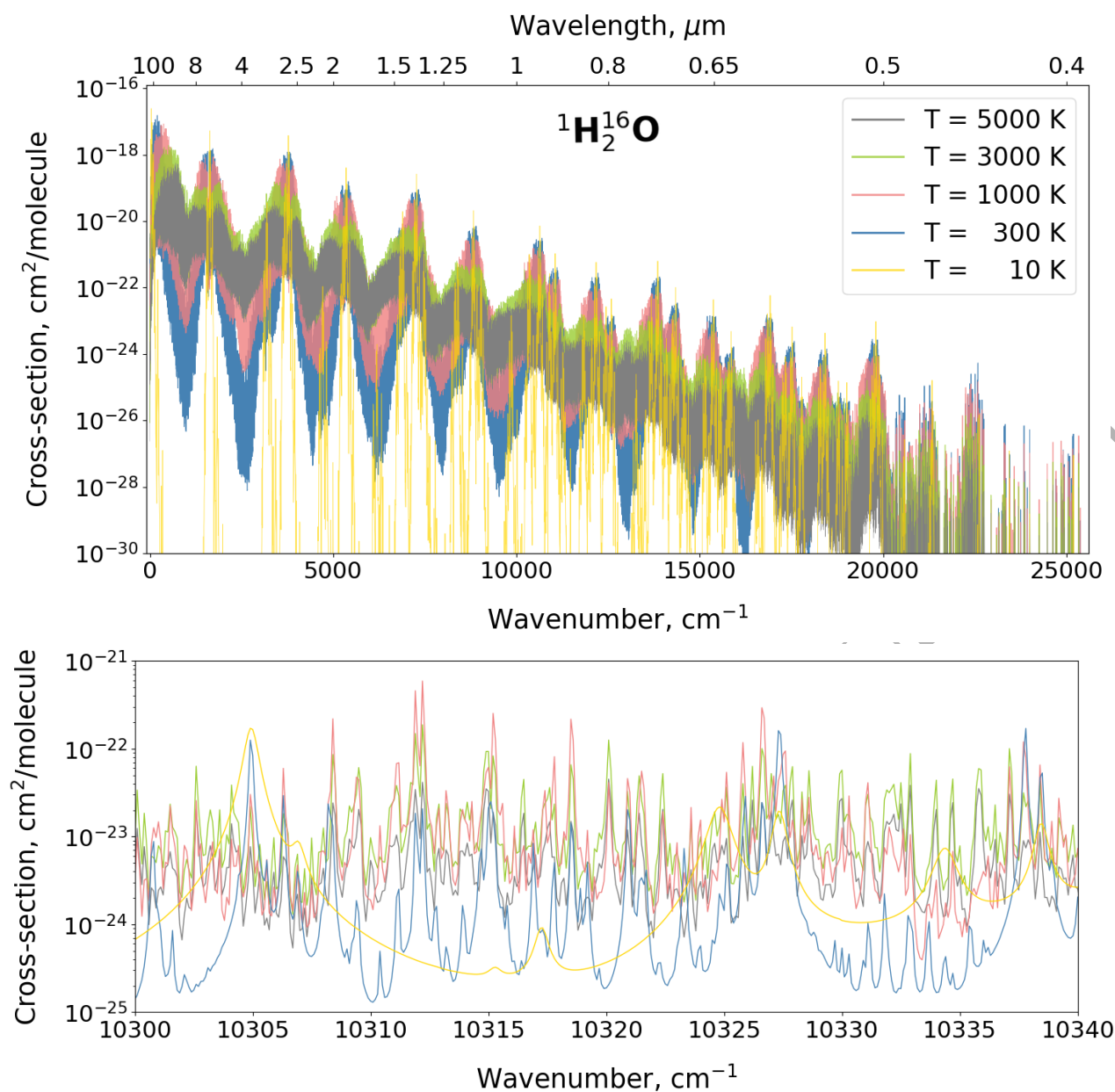


Figure 8. Cross section: Absorption spectrum of $^1\text{H}_2^{16}\text{O}$ at $T = 10, 300, 1000, 3000$ and 5000 K simulated using the ExoMol line list of dataset POKAZATEL (Polyansky et al. 2018) broadened using a SciPy Voigt line profile with the bin size 0.1 cm^{-1} at $P = 1$ bar, HWHM mixed with H_2 and He broadeners and also applied uncertainty and threshold filters (0.01 cm^{-1} and $10^{-30} \text{ cm}^2/\text{molecule}$). Upper panel: Cross sections in wavenumber range $[0, 26000] \text{ cm}^{-1}$; Lower panel: Cross sections zoom for $[10300, 10340] \text{ cm}^{-1}$.

2.9.2 Intensity threshold filter

An intensity threshold filter can be applied when computing stick spectra, cross sections or performing format conversion. A large intensity filter means only stronger intensities are processed in the task in question. PyExoCross extracts and process lines, stronger than intensity threshold requested. The default intensity threshold value is $I > 10^{-30} \text{ cm}^2/\text{molecule}$. Users are recommended to use the threshold filter when processing the molecules with large datasets to reduce run times; however, we should note that including weak lines with $I < 10^{-30} \text{ cm}^2/\text{molecule}$ can significantly alter the results (Yurchenko et al. 2014).

2.9.3 Quantum number filter

A quantum number filter can be applied in calculations of stick spectra and cross sections. This filter can be used to select transitions for a given set of the upper/lower quantum numbers. The user can also select the quantum numbers to be printed out as well as the printing format. Figure 10 shows the stick spectra of MgH using the electronic states as the quantum number filter.

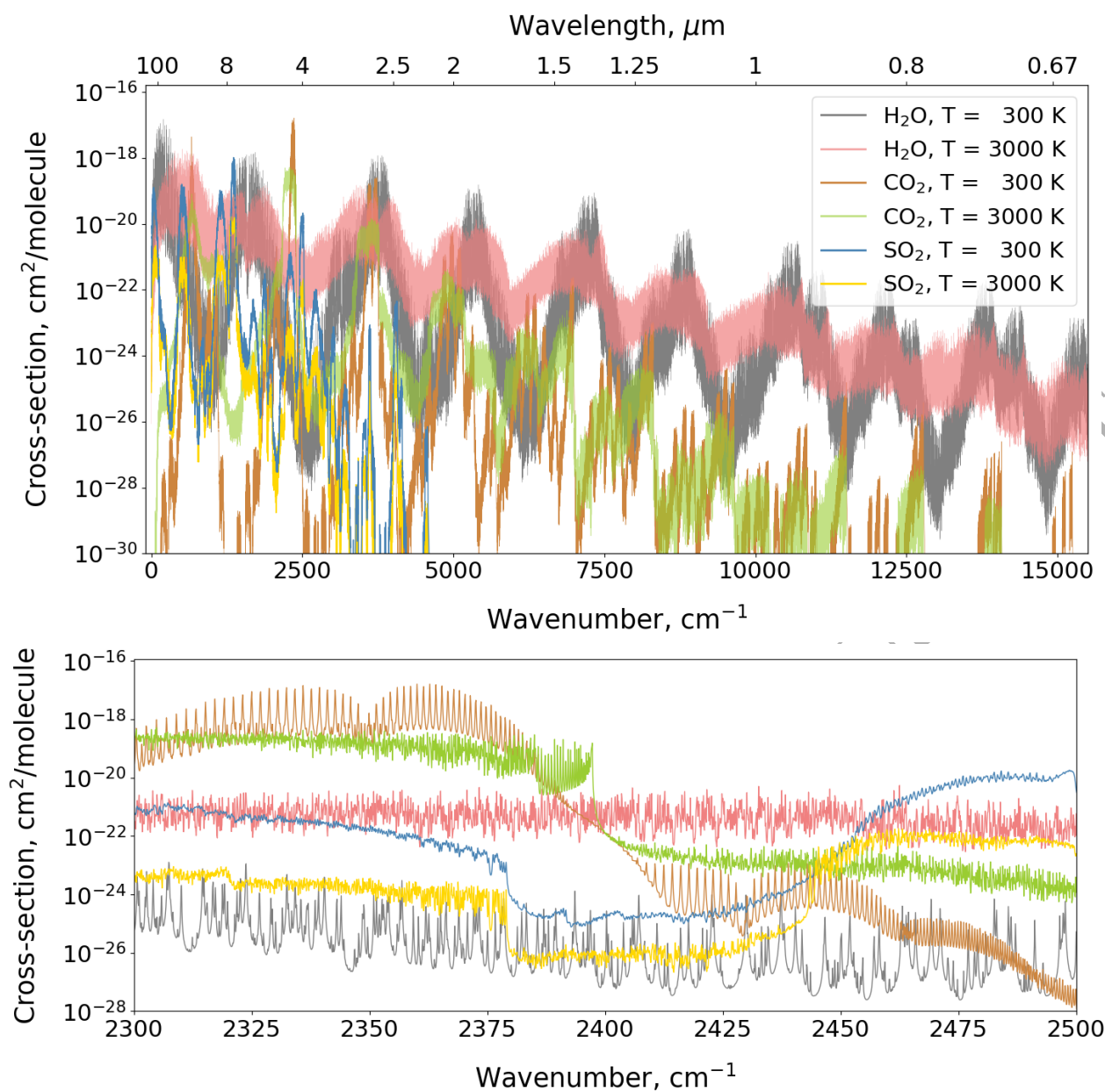


Figure 9. Cross section: Absorption spectrum of $^1\text{H}_2^{16}\text{O}$, $^{12}\text{C}^{16}\text{O}_2$ and $^{32}\text{S}^{16}\text{O}_2$, at $T = 300$ and 3000 K simulated using the ExoMol line list of dataset POKAZATEL (Polynsky et al. 2018), UCL-4000 (Yurchenko et al. 2020) and ExoAmes (Underwood et al. 2016) broadened using a SciPy Voigt line profile with bin size 0.1 cm^{-1} at $P = 1$ bar, HWHM mixed with H_2 and He broadeners and also applied uncertainty and threshold filters (0.01 cm^{-1} and $10^{-30} \text{ cm}^2/\text{molecule}$). Upper panel: Cross sections in wavenumber range $[0, 15500] \text{ cm}^{-1}$; Lower panel: Cross sections zoom for $[2300, 2500] \text{ cm}^{-1}$.

2.10 Plotting options

PyExoCross provides the functionality to optionally plot figures for stick spectra and cross sections. Users also can set the bottom y-axis view limits of data coordinates on plotting stick spectra and cross sections. The default value of the bottom y-axis limit is $10^{-30} \text{ cm}^2/\text{molecule}$.

2.11 Format conversion

2.11.1 Converting data format from ExoMol to HITRAN

PyExoCross can convert data from the ExoMol format to the HITRAN format (see Tables 17, 18). This involves the following steps: (i) reading the States file (.states) from the ExoMol database, (ii) reading the ExoMol Transition files (.trans) in chunks, (iii) reading the ExoMol partition function file, (iv) for each line, calculating the line intensity and applying the uncertainty and intensity thresholds, (v) finally, printing out the transition descriptors in the HITRAN format, with the corresponding error codes for the given uncertainties etc.

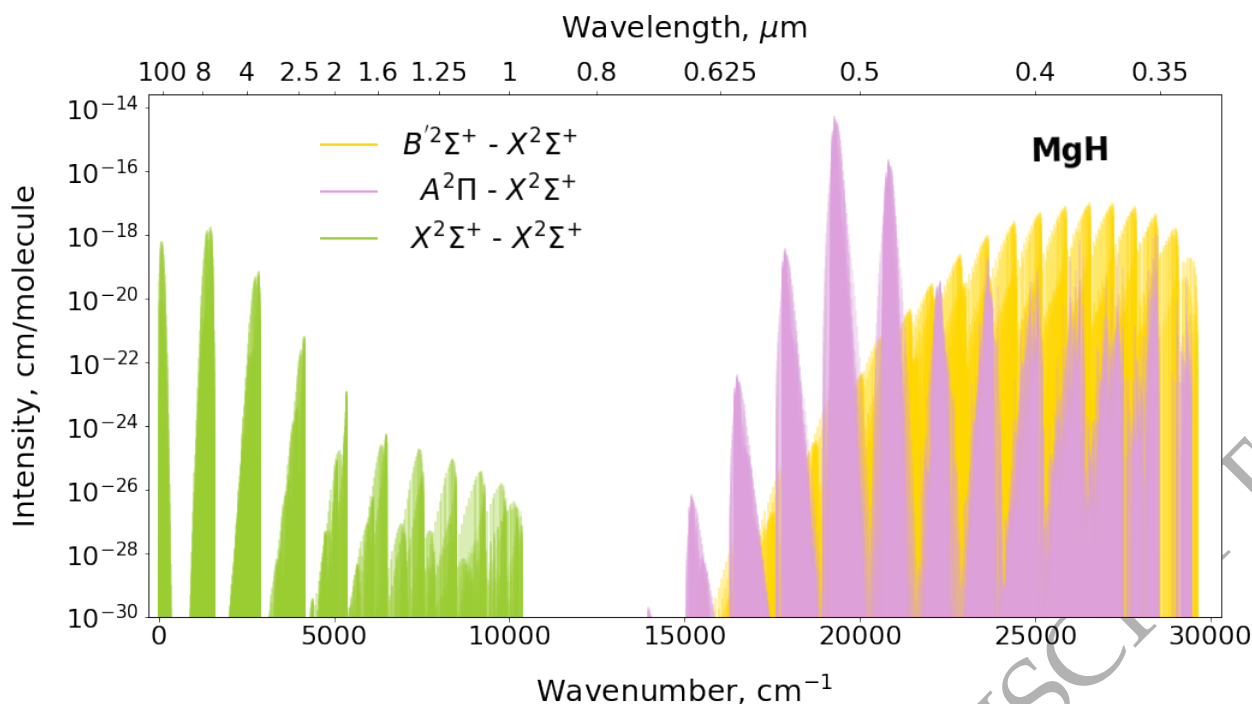


Figure 10. Absolute absorption line intensities of different electronic states X-X, A-X, and B-X bands of the ExoMol $^{24}\text{Mg}^1\text{H}$ simulated at $T = 300$ K.

For the format conversion, the user has to provide the HITRAN molecule and isotopologue IDs. If the molecule is included in the HITRAN database, PyExoCross finds the necessary isotopic abundance from HITRANonline, otherwise the programme uses 1 as the abundance to calculate the intensities. PyExoCross can extract Einstein A-coefficients from the ExoMol Transition files, lower state energy; upper and lower total degeneracy from the ExoMol States file. PyExoCross also supports extracting γ_{self} and n_{air} from .self and .air broadening files separately, if they exist. Otherwise, the default values $\gamma_{\text{self}} = 0.07 \text{ cm}^{-1}/\text{atm}$ and $n_{\text{air}} = 0.5 \text{ cm}^{-1}/\text{atm}$ are used. PyExoCross only considers the first two HITRAN error codes which are error codes of the line position and intensity. The remaining four integers are set to zero. The line position uncertainty error codes are described in Table 19. For the intensity error codes, the record “0” is used for all states since they are undefined in the ExoMol database. Table 20 gives some information on how the quantum number formats are converted.

PyExoCross provides more flexibility to help users choose which quantum number labels they prefer to save in the HITRAN format as well as their formats. Users can provide global and local quantum number labels and their corresponding formats in the input file so that they can arbitrarily change the quantum number formats to satisfy the 60-character limit of the HITRAN database format. If the total number of characters provided is less than 60, PyExoCross will fix them to be exactly 60 characters.

PyExoCross currently cannot provide Air pressure-induced line shifts (δ_{air}), reference indices (Iref) and an associated uncertainty flag.

2.11.2 Converting data format from HITRAN to ExoMol

In principle PyExoCross can convert HITRAN .par files into the ExoMol format (.states, .trans, .pf), but this is not straightforward since the HITRAN data model does not ensure that the tabulated lower state energies and transitions are self-consistent. The main challenge is therefore to construct the ExoMol States file and define the associated state IDs. To this end, PyExoCross the following steps are used.

(A) Construct the ExoMol States file:

(i) read HITRAN line list .par file, (ii) for each line, calculate the upper state energy \tilde{E}'_i with the wavenumber $\tilde{\nu}_j$ and lower state energy \tilde{E}''_j by Eq. (5), (iii) convert the uncertainty error code into the upper bound of the absolute uncertainty range, (iv) process quantum numbers and extract angular momentum F' and F'' from the HITRAN quantum numbers, (v) split HITRAN data into the upper and lower datasets, called dataframes in Python, which include the Einstein A-coefficients (A), the statistical weights (g), uncertainties, state energies \tilde{E} , global quantum number and local quantum numbers, and then concatenate these two data frames with the same columns, (vi) create a new dataframe which merges levels with the same statistical weights g_{tot} , state energies \tilde{E} , uncertainties and quantum numbers; in this process, we extract the lines with the maximum uncertainty when they have the same quantum numbers and statistical weights, (vii) sort with increasing state energies and use the continuum-increasing integers as the state IDs, (viii) create the ExoMol States file in the ExoMol data format with the state ID i , state energy \tilde{E} , total degeneracy g_{tot} , angular momentum J or F , uncertainties unc and quantum numbers, see Table 10.

(B) Construct the ExoMol Transitions file:

(i) For each line in the HITRAN dataframe, reconstruct the upper and lower state IDs. (ii) calculate new transitions frequencies with upper

Table 17. HITRAN line-by-line format in 2004 Edition (Rothman et al. 2005) (160-character record).

Parameter	M	I	v	S	A	γ_{air}	γ_{self}	E''	n_{air}	δ_{air}	V'	V''	Q'	Q''	Ierr	Iref	*	g'	g''
Field length	2	1	12	10	10	5	5	10	4	8	15	15	15	15	6	12	1	7	7
Fortran Format	I2	I1	F12.6	E10.3	E10.3	F5.4	F5.4	F10.4	F4.2	F8.6	A15	A15	A15	A15	6I1	6I2	A1	F7.1	F7.1
C Format	%2d	%1d	%12.6f	%10.3e	%10.3e	%5.4f	%5.4f	%10.4f	%4.2f	%8.6f	%15s	%15s	%15s	%15s	%6d	%12d	%1s	%7.1f	%7.1f

Table 18. Description of the parameters presented in the 160-character records in the line-by-line section of the HITRAN database (Rothman et al. 2005).

Parameter	Meaning	Field length	Type
M	Molecule number	2	Integer
I	Isotopologue number	1	Integer
v	Vacuum wavenumber	12	Real
S	Intensity	10	Real
A	Einstein A-coefficient	10	Real
γ_{air}	Air-broadened half-width	5	Real
γ_{self}	Self-broadened half-width	5	Real
E''	Lower-state energy	10	Real
n_{air}	Temperature-dependent exponent for γ_{air}	4	Real
δ_{air}	Air pressure-induced line shift	8	Real
V'	Upper-state global "quanta"	15	Hollerith
V''	Lower-state global "quanta"	15	Hollerith
Q'	Upper-state local "quanta"	15	Hollerith
Q''	Upper-state local "quanta"	15	Hollerith
Ierr	Uncertainty indices	6	Integer
Iref	Reference indices	12	Integer
*	Flag	1	Character
g'	The statistical weight of the upper state	7	Real
g''	The statistical weight of the lower state	7	Real

Table 19. The uncertainty codes used by the HITRAN database (Gordon et al. 2022).

Code	Absolute uncertainty (unc) range
0	$1 \leq \text{unc}$ or Unreported
1	$0.1 \leq \text{unc} < 1$
2	$0.01 \leq \text{unc} < 0.1$
3	$0.001 \leq \text{unc} < 0.01$
4	$0.0001 \leq \text{unc} < 0.001$
5	$0.00001 \leq \text{unc} < 0.0001$
6	$0.000001 \leq \text{unc} < 0.00001$
7	$0.0000001 \leq \text{unc} < 0.000001$
8	$0.00000001 \leq \text{unc} < 0.0000001$
9	$0.000000001 \leq \text{unc} < 0.00000001$

and lower state energies, (iii) write an ExoMol Transitions file with the upper ID, lower state ID, Einstein A-coefficient A and frequency wavenumber $\tilde{\nu}$. see Table 9;

(C) Construct an ExoMol air-broadening file (.air):

(i) concatenate the initial HITRAN dataframe with the γ_{air} , n_{air} and J'' columns, and then drop duplicated rows, (ii) fill 'a0' in the first column called 'code' and save code, γ_{air} , n_{air} and J'' to the ExoMol air broadening file with the ExoMol data format, see Table 21; note this procedure assumes the relatively simple a0 J -diet for line broadening (Barton et al. 2017).

(D) Construct ExoMol self-broadening files (.self):

(i) concatenate the initial HITRAN dataframe with the γ_{self} , n_{air} and J'' columns, and then drop duplicated rows. (ii) fill 'a0' in the first column called 'code' and save code, γ_{air} , n_{air} and J'' to the ExoMol self broadening file with the ExoMol data format, see Table 21 (see below for more details on the line broadening codes).

PyExoCross can only convert the standard 160-character HITRAN format data to the ExoMol format and for reasons given above the results need to be review carefully. As the HITRAN database has added extra H_2 , He, CO_2 , H_2O broadening, users can replace air and self broadening columns with these extra broadening data. The quantum numbers are formatted following Table S1 and Table S2 in the HITRAN 2020 supplementary material (Karlovets et al. 2021). The quantum numbers are written using the same format as used in the HITRAN .par file, therefore there is no need for the format specification of the quantum numbers.

Table 20. Conversion of the quantum number values between the ExoMol and HITRAN database.

Quantum number value in ExoMol		Quantum value in HITRAN			Rotational parity (0 or 1)
Description Label	Total symmetry G_{tot}	Vibrational symmetry G_{vib}	Rotational Symmetry G_{rot}	Rotational parity (0 or 1) τ_i	
0					s
1	A1'	A1'	A1'	A1'	a
2	A2'	A2'	A2'	A2'	
3	E'	E'	E'	E'	
4	A1''	A1''	A1''	A1''	
5	A2''	A2''	A2''	A2''	
6	E''	E''	E''	E''	

Table 21. Specification of the mandatory part of the .broad broadening file format.

Field	Fortran format	C format	Description
code	A2	%2s	Code identifying quantum number set following J
γ_{ref}	F6.4	%6.4f	Lorentzian half-width at reference temperature and pressure in $\text{cm}^{-1}/\text{bar}$
n_L	F6.3	%6.3f	Temperature exponent
J''	I7/F7.1	%7d/%7.1f	Lower J -quantum number integer/half-integer

Fortran format: J integer: (A2,1x,F6.4,1x,F6.3,1x,I7) or J half-integer: (A2,1x,F6.4,1x,F6.3,1x,F7.1).

3 LINE BROADENING AND LINE PROFILES

An individual spectral line at high resolution is characterised by (i) its transition frequency, (ii) transition integrated intensity, and (iii) line profile. The line profile is governed by line broadening effects which are functions of temperature and pressures, and whose inclusion is essential for most practical applications. The temperature-dependent Doppler broadening is represented by a Doppler line profile in the form of a Gaussian function. The pressure broadening can be modelled by a Lorentzian line profile, while the Voigt line profiles provide a convolution of the Doppler and Lorentzian profiles (Schreier 2017). It may also be necessary to include lifetime effects but these also produce a Lorentzian so do not alter the discussion below (Tennyson et al. 2023).

Line profiles $f_{\tilde{\nu}}(\tilde{\nu})$ are integrable functions whose areas are normalised to unity:

$$\int_{-\infty}^{\infty} f_{\tilde{\nu}}(\tilde{\nu}) d\tilde{\nu} = 1. \quad (13)$$

PyExoCross offers a choice of four profiles namely Doppler, generalised Gaussian, Lorentzian and Voigt. Of course the Doppler profile is a Gaussian but with the line width defined based on the mass of the species concerned, temperature and the line position, while Gaussian allows the user to specify the half-width at half-maximum, α_G . Section 3.5.1 below explores seven different methods of calculating Voigt profiles.

3.1 Wing cut-off

As line profiles have in principle infinite spread, frequency (or wavelength) cutoffs are often used in practical calculations to limit the calculation region to the line centre only and to allow for the fact that Voigt/Lorentzian functions do not generally give a good representation of the far wings (Yurchenko et al. 2018a). Aside from influencing computation time and the accuracy of the cross sections, a wing cut-off is also necessary if functions representing, for example, the water continuum are to be used, as was recently advocated for exoplanet studies by Anisman et al. (2022). Gharib-Nezhad et al. (2024) discuss this issue in detail. PyExoCross uses a default cut-off of 25 cm^{-1} in line with HITRAN, HITEMP and most representations of the water continuum. This choice is a matter of convention and can be changed by the user; however, we note that the default value is in line with the recent recommendations of Gharib-Nezhad et al. (2024).

3.2 Broadening parameters

Pressure broadening in the ExoMol database is currently based on the ExoMol diet (Barton et al. 2017); the database stores line broadening parameters in .broad broadening files; broadeners available for a given species are specified in the .def definition file. Users can provide as many broadening files with appropriate file suffixes, e.g. .H2O and .CO2. PyExoCross can then use suffixes to read the corresponding broadening files and, if necessary, process mixtures of broadeners, see Section 3.4.2. PyExoCross can process user-supplied broadening files provided they use the ExoMol .broad file format. The format of the broadening file is expressed in Table 21 and the specification of the broadening file is given in Table 22. Currently, the ExoMol database contains .broad files for air, Ar, CH₄, CO, CO₂, H₂, H₂O, He, N₂, NH₃, NO, O₂ and the rapidly increasing number of broadeners being included in the ExoMol database. The HITRAN database has air and self broadening for all species as well as parameters for the broadening by He, H₂, H₂O and CO₂ (Ngo et al. 2013; Wilzewski et al. 2016; Tan et al. 2019; Wcislo et al. 2021; Tan et al. 2022) for a small number of key species.

Table 22. Extract from the `.broad` broadening file of the H₂ broadening file `14N-1H3__H2.broad` (Barton et al. 2017).

Code	γ_{ref}	n_L	J''
a0	0.0908	0.583	0
a0	0.0935	0.559	1
a0	0.0926	0.527	2
a0	0.0850	0.508	3
a0	0.0800	0.529	4
a0	0.0768	0.506	5
a0	0.0728	0.501	6
a0	0.0698	0.506	7
a0	0.0671	0.514	8
a0	0.0655	0.500	9

Code: Code identifying quantum number set following J ;

γ_{ref} : Lorentzian half-width at reference temperature and pressure in $\text{cm}^{-1}/\text{bar}$;

n_L : Temperature exponent;

J'' : Lower J -quantum number integer/half-integer.

3.3 Doppler and Gaussian profile

The Gaussian line profile is defined as

$$f_{\tilde{\nu}_{fi}, \alpha_G}(\tilde{\nu}) = \sqrt{\frac{\ln 2}{\pi}} \frac{1}{\alpha_G} \exp\left(-\frac{(\tilde{\nu} - \tilde{\nu}_{fi})^2 \ln 2}{\alpha_G^2}\right), \quad (14)$$

where α_G is the half-width at half-maximum (HWHM) and $\tilde{\nu}_{fi}$ is the line centre. α_G is related to the Gaussian standard deviation σ_G as (Squires 2001):

$$\sigma_G = \frac{\alpha_G}{\sqrt{2 \ln 2}} \quad (15)$$

and can be specified in the PyExoCross input.

The Doppler profile is based on the Gaussian shape specified in Eq. (14) with HWHM α_D given by (Squires 2001)

$$\alpha_D = \sqrt{\frac{2N_A k_B T \ln 2}{M} \frac{\tilde{\nu}_{fi}}{c}}, \quad (16)$$

which depends on the temperature, mass of the molecule and the line position. Here, k_B (erg·K) is the Boltzmann constant, c (cm/s) is the speed of light, T (K) is the temperature. M (Da) ($\text{g}\cdot\text{mol}^{-1}$) is the molar mass of the isotopologue in grams (Ngo et al. 2013) which can be taken from the HITRANOnline website or from the `.def` definition file of the ExoMol database. N_A (mol^{-1}) is Avogadro's number. N_A/M can be replaced by $1/1000m$ where isotopologue mass m (kg) is available from the ExoMol definition file. For Doppler as well as for the generalised Gaussian, the HWHMs do not depend on the (upper/lower) molecular states.

3.4 Lorentzian profiles

The Lorentzian profile is the Cauchy distribution whose probability density function (PDF) is (Feller 1971; Johnson et al. 1994):

$$f_{\tilde{\nu}_{fi}, \gamma_L}^L(\tilde{\nu}) = \frac{1}{\pi} \frac{\gamma_L}{(\tilde{\nu} - \tilde{\nu}_{fi})^2 + \gamma_L^2}, \quad (17)$$

where $\tilde{\nu}_{fi}$ is the centroid in wavenumbers identifying the position of the peak of the distribution and γ_L is Lorentzian HWHM, which in PyExoCross conventionally assumes the following temperature and pressure dependent form

$$\gamma_L = \gamma_{\text{ref}} \left(\frac{T_{\text{ref}}}{T}\right)^{n_L} \frac{P}{P_{\text{ref}}}, \quad (18)$$

where $\tilde{\nu}_{fi}$ is the line centre. As in HITRAN, $T_{\text{ref}} = 296$ K is the reference temperature and $P_{\text{ref}} = 1$ bar is the reference pressure (Schreier & Hochstaffl 2021). γ_{ref} and n_L are the reference HWHM (see Sections 3.2, 3.4.1) and the temperature exponent, which are the broadening parameters for the given broadeners in the ExoMol database. In the HITRAN database, γ_{air} and γ_{self} are used to calculate γ_{ref} with mixing ratios provided by users, see Section 3.4.2. n_L is as n_{air} in the HITRAN line lists. In general, γ_{ref} and n_L are (upper/lower) state-dependent. The specification of the broadeners and the broadener mixture ratio must be set in the PyExoCross `.inp` input file.

Users can choose to input the HWHM (γ_L) of the Lorentzian profile directly as a constant for calculations instead of calculating with the parameters from the line-broadening files of the ExoMol database.

3.4.1 Extracting broadening parameters from .broad

Calculations of the Lorentzian/Voigt broadening requires information from the States (.states) file and the parameters from the broadening file (.broad). If the value of J'' in the states file is larger than J''_{\max} in the broadening file, then the broadening parameters γ_L and n_L corresponding to J''_{\max} are assumed. If J'' is half-integer, the parameters corresponding to the index value $J'' - \frac{1}{2}$ in the broadening file are used. This process of extracting values based on their indices is especially efficient in Python. A part of the Python code for extracting broadening parameters from the .broad file is shown in Listing 1.

```
def extract_broad(broad_df, states_trans_df):
    max_broad_J = max(broad_df['Jpp'])
    Jpp = states_trans_df['J'''].values
    Jpp[Jpp > max_broad_J] = max_broad_J
    id_broad = (states_trans_df['J''']-0.1).round(0).astype('int32').values
    gamma_L = broad_df['gamma_L'][id_broad]
    n_air = broad_df['n_air'][id_broad]
    return(gamma_L, n_air)
```

Listing 1: A sample of the Python code for extracting broadeners from the .broad broadening file. PyExoCross reads the .broad broadening file and saves the data as a DataFrame broad_df. states_trans_df is a DataFrame storing the line lists of the .states states file merged by upper and lower states IDs from the .trans transitions file. Jpp and J'' means lower J and id_broad indicates the index of the broad DataFrame. gamma_L and n_air are γ_{ref} and n_{air} denote the reference HWHM and the temperature exponents.

3.4.2 Mixtures of broadeners

PyExoCross allows for inclusion of different broadeners or their mixtures. For mixtures, in the input file, users can specify the fractional abundance (X_i) of the i th broadener. Then, the overall value of γ_L is the weighted sum of the values of γ_i^L from each broadener (Yurchenko et al. 2018a):

$$\gamma_L = \sum_i \gamma_i^L X_i \quad \text{and} \quad \sum_i X_i = 1. \quad (19)$$

3.4.3 Lifetime broadening

In principle, the natural lifetime contributes to the line profile alongside the effects of the temperature-dependent Doppler broadening and the pressure-dependent collision broadening. In practice, the natural lifetime generally contributes very little to the line profile and is consequently ignored. However, for predissociating states, the effect of the lifetime broadening can be important, see Pavlenko et al. (2022) and Yurchenko et al. (2024) for example; these can easily be included in the Voigt profile (Tennyson et al. 2023). Like the pressure broadening, lifetime broadening γ_τ produces a Lorentzian line shape with the HWHM given by:

$$\gamma_\tau = \frac{\hbar}{2\tau' hc}, \quad (20)$$

where h is the Planck constant (erg·s) and $\hbar = h/(2\pi)$ is the reduced Planck constant. c is the speed of light (cm/s). τ' is the upper state lifetime (s). The sum of the pressure broadening and lifetime broadening $\gamma_{\text{tot}} = \gamma_p + \gamma_\tau$ can be then used as the total Lorentzian component of the line profile. Since a Voigt profile is already being utilised, this has no computational impact on a calculation, implying that routine use of γ_{tot} for the half-width on a routine basis would eliminate the need to worry about whether or not predissociation has to be considered.

3.5 Voigt profiles

The Voigt profile results from the convolution of the Gaussian and Lorentzian profiles and is given by (Armstrong 1967; Herbert 1974):

$$f_{\tilde{\nu}_{fi}, \sigma_G, \gamma_L}^V(\tilde{\nu}) = f_{\tilde{\nu}_{fi}, \sigma_G}^G(\tilde{\nu}) \otimes f_{\tilde{\nu}_{fi}, \gamma_L}^L(\tilde{\nu}) = \int_{-\infty}^{+\infty} d\tilde{\nu}' f^G(\tilde{\nu} - \tilde{\nu}', \sigma_G) \times f^L(\tilde{\nu}' - \tilde{\nu}_{fi}, \gamma_L) = \frac{y}{\pi^{3/2} \sqrt{2} \sigma_G} \int_{-\infty}^{+\infty} \frac{e^{-t^2}}{(x-t)^2 + y^2} dt, \quad (21)$$

where

$$x = \frac{\tilde{\nu} - \tilde{\nu}_{fi}}{\sqrt{2} \sigma_G} \quad \text{and} \quad y = \frac{\gamma_L}{\sqrt{2} \sigma_G}. \quad (22)$$

Here x gives the wavenumber scale in units of the Gaussian standard deviation, y is the ratio of Lorentzian widths to the Gaussian standard deviation, σ_G is the Gaussian standard deviation and γ_L is HWHM of the Lorentzian profile. The central wavenumber or frequency, $\tilde{\nu}_{fi}$ is the energy difference of the transition (Schreier 2018).

PyExoCross offers nine methods for calculating the Voigt profile with five of them as pseudo-Voigt profiles as described in the following.

3.5.1 *SciPy Voigt profile*

Use of the Python open-source library SciPy (Virtanen et al. 2020) gives the Voigt profile directly, see Listing 2.

```
from scipy.special import voigt_profile
def scipy_Voigt_profile(dv, sigma, gamma):
    return voigt_profile(dv, sigma, gamma)
```

Listing 2: Sample of the Python code used to calculate Voigt Profile by SciPy package (Virtanen et al. 2020). $dv = \tilde{v} - \tilde{v}_{fi}$ and \tilde{v}_{fi} is the line centre. σ is the standard deviation σ_G of the Gaussian profile. γ is the half-width for pressure broadening γ_L .

`scipy.special.voigt_profile` uses `scipy.signal.convolve` to compute the Voigt profile with the convolution of a 1-D Normal distribution (Gaussian profile) with the standard deviation σ_G and a 1-D Cauchy distribution (Lorentzian profile) with HWHM γ_L (Virtanen et al. 2020), see Eq. (21). The method of `scipy.signal.convolve` is given in Listing 3.

```
from scipy.signal import convolve
convolved=dx*convolve(Lorentzian_profile, Gaussian_profile, mode="same")
```

Listing 3: Method for computing Voigt profile by the convolution of the Gaussian and Lorentzian profile using the SciPy Python package as used by the `scipy.special.voigt_function`.

3.5.2 *SciPy wofz Voigt profile*

The Python open-source library SciPy can be used to calculate the Voigt profile, see Listing 4. The SciPy function `scipy.special.wofz` calculates the Voigt profile for σ_G and γ_L as given by

$$f_{\tilde{v}_{fi}, \sigma_G, \gamma_L}^V(\tilde{v}) = \frac{1}{\sqrt{2\pi}\sigma_G} K(x, y), \quad (23)$$

where x and y are given in Eq. (22), $K(x, y)$ is the Voigt function

$$K(x, y) = \frac{y}{\pi} \int_{-\infty}^{+\infty} \frac{e^{-t^2}}{(x-t)^2 + y^2} dt \quad (24)$$

satisfying $\int K dx = \sqrt{\pi}$.

$K(x, y)$ comprises the real part of the complex error function (Faddeeva function) $\omega(z)$ (Faddeeva et al. 1961)

$$\omega(z) \equiv K(x, y) + iL(x, y) = \frac{i}{\pi} \int_{-\infty}^{+\infty} \frac{e^{-t^2}}{z-t} dt. \quad (25)$$

where

$$z = x + iy = \frac{(v - \tilde{v}_{fi} + i\gamma_L)}{\sqrt{2}\sigma_G} \quad (26)$$

and

$$f_{\tilde{v}_{fi}, \sigma_G, \gamma_L}^V(\tilde{v}) = \frac{\text{Re}[\omega(z)]}{\sigma_G \sqrt{2\pi}}. \quad (27)$$

In Eq. (26), in the limiting cases of $\sigma_G = 0$ and $\gamma_L = 0$, Eq. (27) is simplified to the Lorentzian profile and Gaussian profile, respectively.

```
import numpy as np
from scipy.special import wofz
def scipy_wofz_Voigt_profile(dv, sigma, gamma):
    z = (dv + 1j * gamma) / sigma / np.sqrt(2)
    voigt = (np.real(wofz(z)) / sigma / np.sqrt(2 * np.pi))
    return voigt
```

Listing 4: A sample of the Python code used to calculate the Voigt profile (Virtanen et al. 2020; Hill 2020) with the real part of the Faddeeva function. `wofz` in `scipy.special.wofz` can be computed with error function $\text{erf}(-iz)$ given by Pierluissi et al. (1977): $\omega(z) = e^{-z^2}(1 - \text{erf}(-iz))$.

3.5.3 Humlíček Voigt profile

Humlíček's algorithm (Humlíček 1979) for approximating the Voigt profile function is used extensively. Humlíček's Voigt profile gives a method for computing the complex probability function $\omega(z)$ where $z = x + iy$ (see Eqs. (21, 25, 26)). $x = \frac{(\tilde{\nu} - \tilde{\nu}_{fi})\sqrt{\ln 2}}{\alpha_D}$ is the wavenumber scale in units of the Doppler HWHM α_D and $y = \frac{\gamma_L\sqrt{\ln 2}}{\alpha_D}$ is the ratio of the Lorentzian HWHM γ_L to the Doppler HWHM α_D . A rational approximation is used in Humlíček's w4 algorithm, in which the imaginary part on the real axis is minimized in order to minimize its relative error (Humlíček 1982). The following procedure can evaluate both the real and imaginary parts of $\omega(z)$ with a high degree of relative accuracy (Humlíček 1979, 1982; Kuntz 1997).

$$\omega(z) = \begin{cases} \frac{t}{(0.5+t^2)\sqrt{\pi}}, & \text{if } s \geq 15, \\ \frac{t(1.4104739589 + \frac{u}{\sqrt{\pi}})}{0.75+u(3+u)}, & \text{if } 5.5 \leq s < 15, \\ \frac{16.4955+t(20.20933+t(11.96482+t(3.778987+0.5642236t)))}{16.4955+t(38.82363+t(39.27121+t(21.69274+t(6.699398t))))}, & \text{if } s < 5.5 \text{ and } y \geq 0.195|x| - 0.176, \\ e^u - \frac{t(36183.31-u(3321.99-u(1540.787-u(219.031-u(35.7668-u(1.320522-u(0.56419))))))}{32066.6-u(24322.8-u(9022.23-u(2186.18-u(364.219-u(61.5704-u(1.84144-u))))))}, & \text{if } s < 5.5 \text{ and } y < 0.195|x| - 0.176, \end{cases} \quad (28)$$

where complex function $t = y - xi$, $s = |x| + y$ and $u = t^2$. Zaghoul (2017) and Hui et al. (1978) defined a simpler, more efficient, and relatively accurate approximation for the evaluation of the Faddeyeva function with six regions of the proposed partitioning.

The Humlíček Voigt line profile is then given by (Humlíček 1979, 1982)

$$f_{\tilde{\nu}_{fi}, \alpha_D, \gamma_L}^V(\tilde{\nu}) = \frac{\text{Re}[\omega(z)]\sqrt{\ln 2}}{\pi\alpha_D}. \quad (29)$$

3.5.4 Pseudo-Voigt profiles

PyExoCross provides five pseudo-Voigt line profiles. They are not the recommended algorithms and we do not consider pseudo-Voigt profiles in comparison Section 3.7 because they have lower accuracy and speed of run time than other Voigt profiles.

3.5.4.1 Thompson pseudo-Voigt profile

The pseudo-Voigt function $V(d\nu, \gamma_V)$ is frequently used for calculating experimental spectral line shapes (Ida et al. 2000). As an approximation to the Voigt profile $V(x)$, the pseudo-Voigt profile combines a Gaussian curve $G(x)$ and a Lorentzian curve $L(x)$ linearly, rather than by convolution (Ida et al. 2000). The following equation provides the mathematical definition of the normalised pseudo-Voigt profile $V(d\nu, \gamma_V)$ (Thompson et al. 1987; Sánchez-Bajo & Cumbreira 1997):

$$V(d\nu, \gamma_V) = \eta L(d\nu, \alpha_D) + (1 - \eta)G(d\nu, \gamma_L) \quad \text{with } 0 < \eta < 1, \quad (30)$$

where $d\nu = \tilde{\nu} - \tilde{\nu}_{fi}$; α_D and γ_L are the HWHM of the Gaussian and Lorentzian profiles.

The accuracy of the pseudo-Voigt profile depends on several possible choices for calculating the parameter η and HWHM of the Voigt profile γ_V . For the Thompson pseudo-Voigt profile η is given in Eq. (31) (Thompson et al. 1987).

$$\eta \approx 1.36603(\gamma_L/\gamma_V) - 0.47719(\gamma_L/\gamma_V)^2 + 0.11116(\gamma_L/\gamma_V)^3. \quad (31)$$

From the associated Gaussian and Lorentzian HWHMs, the value of γ_V of the Thompson pseudo-Voigt profile can be determined using (Thompson et al. 1987):

$$\gamma_V = \left[\alpha_D^5 + 2.69269\alpha_D^4\gamma_L + 2.42843\alpha_D^3\gamma_L^2 + 4.47163\alpha_D^2\gamma_L^3 + 0.07842\alpha_D\gamma_L^4 + \gamma_L^5 \right]^{1/5}. \quad (32)$$

3.5.4.2 Kielkopf pseudo-Voigt profile

Kielkopf (1973) proposed a weighted-sum method to approximate the value of γ_V of the pseudo-Voigt profile via Eqs. (30,31) as given by (see also Schreier (2011))

$$\gamma_V \approx \frac{1}{4}[(1 + 0.099 \ln 2)\gamma_L + \sqrt{(1 - 0.099 \ln 2)^2\gamma_L^2 + 4 \ln 2\alpha_D^2}] \approx 0.5346\gamma_L + \sqrt{0.2166f\gamma_L^2 + \alpha_D^2}. \quad (33)$$

3.5.4.3 Olivero pseudo-Voigt profile

Olivero & Longbothum (1977) provided another alternative to Eqs. (30, 31) with the value of HWHM γ_V approximated by

$$\gamma_V \approx [1 - 0.18121(1 - d^2) - (0.023665e^{0.6d} + 0.00418e^{-1.9d}) \sin(\pi d)](\alpha_D + \gamma_L), \quad (34)$$

where d is a dimensionless parameter related to α_D and γ_L (Olivero & Longbothum 1977) as

$$d = \frac{\gamma_L - \alpha_D}{\gamma_L + \alpha_D}. \quad (35)$$

3.5.4.4 *Liu-Lin pseudo-Voigt profile*

Liu et al. (2001) used d related to the Gaussian and Lorentzian half widths (α_D and γ_L) defined in Eq. (35) to provide a new conversion of η from Kielkopf (1973), see Eq. (31). The half-width of the Voigt profile is the same as Eq. (34). The Liu-Lin pseudo-Voigt profile is calculated from Eqs. (30, 34, 35, 36) using

$$\eta \approx 0.68188 + 0.61293d - 0.18384d^2 - 0.11568d^3. \quad (36)$$

3.5.4.5 *Rocco pseudo-Voigt profile*

Di Rocco & Cruzado (2012) provided a theoretically justified method for calculating γ_V and η in the pseudo-Voigt methodology and defined them as follows

$$\begin{aligned} \gamma_V &= \omega_G b_{1/2}(y), \\ \eta &\approx \frac{\sqrt{\pi} V_y(0) \gamma_V - \sqrt{\ln 2}}{\sqrt{\pi} V_y(0) \gamma_V (1 - \sqrt{\pi \ln 2})}, \end{aligned} \quad (37)$$

where

$$\omega_G = \frac{\alpha_D}{\sqrt{\ln 2}},$$

$V_y(0)$ is the Voigt profile in Eq. (21) at $x = 0$

$$V_y(0) = \frac{e^{y^2} (1 - \operatorname{erf}(y))}{\sqrt{\pi} \omega_G}.$$

Here,

$$y = \frac{\gamma_L}{\omega_G}$$

and

$$b_{1/2}(y) = y + \sqrt{\ln 2} e^{-0.6055y + 0.0718y^2 - 0.0049y^3 + 0.000136y^4}.$$

3.6 Binned line profiles

All the above methods use a sampling algorithm to evaluate the Voigt profile on a set of frequency wavenumbers $\tilde{\nu}_i$. The disadvantage of these methods is that they can lead to under-sampling of the opacity (absorption or emission) at low sampling resolution, see e.g. Hill et al. (2013). In order to avoid this problem and guarantee that the full opacity is preserved, the following binned approach can be used by defining the analytically integrated (averaged) cross sections within a bin $[\tilde{\nu}_k - \Delta\tilde{\nu}/2, \tilde{\nu}_k + \Delta\tilde{\nu}/2]$ as a sum over the contributions from each line (Hill et al. 2013) as given by

$$\begin{aligned} \bar{\sigma}_k^{\text{ab},fi} &= \frac{I_{fi}}{\Delta\tilde{\nu}} \int_{\tilde{\nu}_k - \Delta\tilde{\nu}/2}^{\tilde{\nu}_k + \Delta\tilde{\nu}/2} f_{\tilde{\nu}_{fi}}(\tilde{\nu}) d\tilde{\nu}, \\ \bar{\sigma}_k^{\text{em},if} &= \frac{\epsilon_{if}}{\Delta\tilde{\nu}} \int_{\tilde{\nu}_k - \Delta\tilde{\nu}/2}^{\tilde{\nu}_k + \Delta\tilde{\nu}/2} f_{\tilde{\nu}_{if}}(\tilde{\nu}) d\tilde{\nu}, \end{aligned} \quad (38)$$

where $\tilde{\nu}_{fi}$ is the line centre, I_{fi} and ϵ_{if} are the line intensities (absorption and emission) from Eqs. (8, 9).

The related functions for the binned cross sections for the Doppler, Gaussian, Lorentzian and Voigt profiles are specified below.

3.6.1 *Binned Doppler and Gaussian profile*

Both the Doppler and the Gaussian profiles use the Gaussian function. Taking advantage of the fact that the integral of the Gaussian function has an analytic solution, the error function erf can be used:

$$\int e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \operatorname{erf}(x). \quad (39)$$

Therefore, we apply the analytical integral to the Gaussian profile (Hill et al. 2013):

$$\bar{\sigma}_k^{fi} = \frac{I_{fi}}{\Delta\tilde{\nu}} \int_{\tilde{\nu}_k - \Delta\tilde{\nu}/2}^{\tilde{\nu}_k + \Delta\tilde{\nu}/2} f_{\tilde{\nu}_{fi}}^G(\tilde{\nu}) d\tilde{\nu} = \frac{I_{fi}}{2\Delta\tilde{\nu}} \left[\operatorname{erf}(x_{k,fi}^+) - \operatorname{erf}(x_{k,fi}^-) \right], \quad (40)$$

where $x_{k,fi}^\pm$ are the scaled limits of the frequency bin k centred on $\tilde{\nu}_k$ related to the line centre $\tilde{\nu}_{fi}$ given by Hill et al. (2013):

$$x_{k,fi}^\pm = \frac{\sqrt{\ln 2}}{\alpha_D} \left[\tilde{\nu}_k - \tilde{\nu}_{fi} \pm \frac{\Delta\tilde{\nu}}{2} \right]. \quad (41)$$

3.6.2 Binned Lorentzian profile

The integral of the Lorentzian function has an analytical solution in the format:

$$\int \frac{dx}{x^2 + \gamma^2} = \frac{1}{\gamma} \arctan\left(\frac{x}{\gamma}\right) + C. \quad (42)$$

If we calculate profiles in the range of the wings which have large distances from the line centre, then the normalisation correction factor is unity. To solve the far wings in calculating profiles and save an excessive amount of computing time, we truncate the profiles at some distance $\Delta\tilde{\nu}/2$ from the line centre (Sharp & Burrows 2007). Therefore, PyExoCross provides two methods for calculating binned Lorentzian and Voigt profiles with or without a cutoff filter for users.

The same approach of an analytical binned integral can be applied to the Lorentzian profile $f_{\tilde{\nu}_{fi}}^L(\tilde{\nu})$ as follows (Yurchenko et al. 2018a):

$$\bar{\sigma}_k^{fi} = \frac{I_{fi}}{\Delta\tilde{\nu}} \int_{\tilde{\nu}_k - \Delta\tilde{\nu}/2}^{\tilde{\nu}_k + \Delta\tilde{\nu}/2} f_{\tilde{\nu}_{fi}}^L(\tilde{\nu}) d\tilde{\nu} = \frac{I_{fi}}{\pi\Delta\tilde{\nu}} \left[\arctan\left(y_{k,fi}^+\right) - \arctan\left(y_{k,fi}^-\right) \right], \quad (43)$$

where $\bar{\sigma}_k^{fi}$ is an averaged cross section for the bin k centred at $\tilde{\nu}_k$ over the interval $\Delta\tilde{\nu}$, $\tilde{\nu}_{fi}$ is the centre of the line and $y_{k,fi}^\pm$ are given by

$$y_{k,fi}^\pm = \frac{\tilde{\nu}_k - \tilde{\nu}_{fi} \pm \Delta\tilde{\nu}/2}{\gamma_L}. \quad (44)$$

Because the line profiles are usually cut off at a distance $\Delta\tilde{\nu}/2$ from the line centre, it is common to preserve the total strength of the line by re-normalizing its area to unity (Sharp & Burrows 2007). Here we multiply Eq. (43) by the correction factor $b_{\text{corr}} \geq 1$ given by

$$b_{\text{corr}} = \frac{\pi}{\arctan(y_{\text{end},fi}) - \arctan(y_{\text{start},fi})}, \quad (45)$$

where $y_{\text{start},fi}$ and $y_{\text{end},fi}$ are defined as:

$$y_{\text{start},fi} = \frac{\tilde{\nu}_{\text{start}} - \tilde{\nu}_{fi}}{\gamma_L} \quad \text{and} \quad y_{\text{end},fi} = \frac{\tilde{\nu}_{\text{end}} - \tilde{\nu}_{fi}}{\gamma_L}. \quad (46)$$

3.6.3 Binned Voigt profile

Gauss-Hermite quadrature is a form of the Gaussian quadrature used in numerical analysis to approximate the value of integrals (Abramowitz & Stegun 1972):

$$\int_{-\infty}^{+\infty} e^{-t^2} f(t) dt \approx \sum_{m=1}^n \omega_m f(t_m), \quad (47)$$

where n is the number of the used sample points and t_m are the roots of the Hermite polynomial $H_n(t)$ ($m = 1, 2, \dots, n$) (physicists' version) and the Gauss-Hermite quadrature weights ω_m are given by Abramowitz & Stegun (1972).

$$\omega_m = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 [H_{n-1}(t_m)]^2}. \quad (48)$$

Combined with the Gaussian and Lorentzian profiles, we use Humlíček's algorithm (Humlíček 1979) as the Gauss-Hermite quadratures to create a formulation of a similar integral method for the Voigt profile (Yurchenko et al. 2018a) with HWHM of Doppler (and Gaussian) profile α_D and the HWHM of Lorentzian profile γ_L (see Eq. (21)).

$$f_{\tilde{\nu}_{fi}, \alpha_D, \gamma_L}^V(\tilde{\nu}) = \frac{y \ln 2}{\pi^{3/2} \alpha_D^2} \int_{-\infty}^{+\infty} \frac{e^{-t^2}}{(x-t)^2 + y^2} dt = \frac{y \ln 2}{\pi^{3/2} \alpha_D^2} \sum_{m=1}^{N_{\text{G-H}}} \frac{\omega_m^{\text{G-H}}}{(x-t_m)^2 + y^2}, \quad (49)$$

where t_m are the Gauss-Hermite quadrature points (roots) and $\omega_m^{\text{G-H}}$ are the Gauss-Hermite quadrature weights. Using Eq. (43) to form the area-conserved integrals, for the binned Voigt profile we obtain (Yurchenko et al. 2018a):

$$\bar{\sigma}_k^{fi} = \frac{I_{fi}}{\pi^{3/2} \Delta\tilde{\nu}} \sum_{m=1}^{N_{\text{G-H}}} \omega_m^{\text{G-H}} \left[\arctan\left(z_{k,fi,m}^+\right) - \arctan\left(z_{k,fi,m}^-\right) \right], \quad (50)$$

where the scaled limits $z_{k,fi,m}^\pm$ of the bin k centred at the grid point $\tilde{\nu}_k$ is given by

$$z_{k,fi,m}^\pm = \frac{\tilde{\nu}_k - \tilde{\nu}_{fi} - t_m \sigma_G \pm \Delta\tilde{\nu}/2}{\gamma_L}.$$

We commonly take $N_{\text{G-H}} = 20$ as the number of Gauss-Hermite points used.

This expression can be further corrected for the missing area due to the wing's cutoff similarly to the Lorentzian example above. Here, we multiply $\bar{\sigma}_k^{fi}$ in Eq. (50) with the normalisation correction factor b_{corr} given by

$$b_{\text{corr}} = \frac{\pi}{\arctan(z_{\text{end},fi,m}) - \arctan(z_{\text{start},fi,m})}, \quad (51)$$

where

$$z_{\text{start},fi,m} = \frac{\tilde{\nu}_{\text{start}} - \tilde{\nu}_{fi} - t_m \sigma_G}{\gamma_L} \quad \text{and} \quad z_{\text{end},fi,m} = \frac{\tilde{\nu}_{\text{end}} - \tilde{\nu}_{fi} - t_m \sigma_G}{\gamma_L}. \quad (52)$$

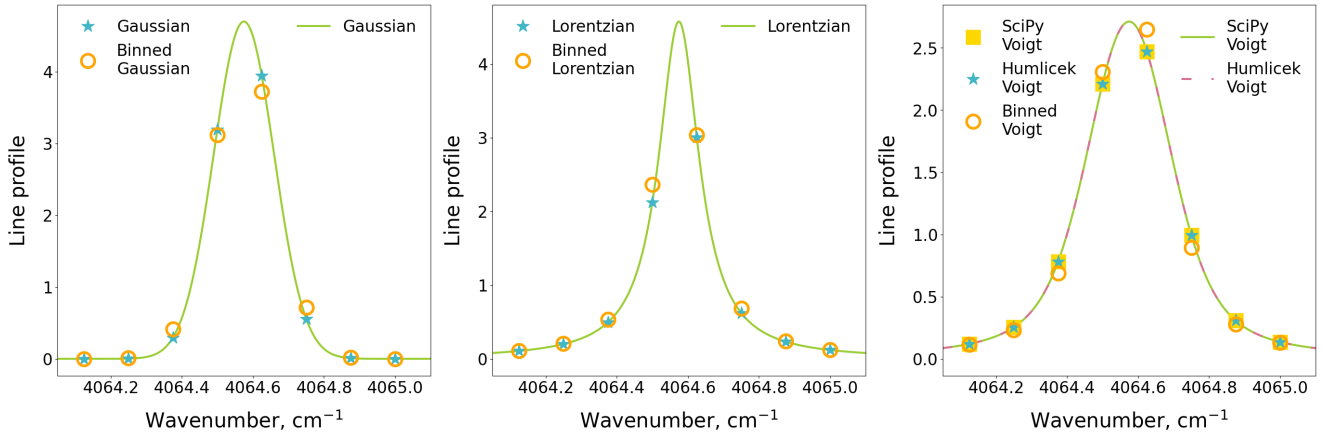


Figure 11. Comparison of the sampling and binned methods with different line profiles for a single normalised transition with a cutoff = 100 cm^{-1} and the bin size of 0.125 cm^{-1} (8 points per 1 cm^{-1}) at the same temperature $T = 300 \text{ K}$ and pressure $P = 1 \text{ bar}$. Left panel: Doppler profile; Middle panel: Lorentzian profile; Right panel: Voigt profile.

3.7 Line profile comparisons

3.7.1 Comparison of the sampling and binned line profile methods

When the cross sections resolution (i.e. the number of points per wavenumber interval) is low, the result of cross sections calculations becomes very sensitive to the way how profiles are evaluated, using sampling or binning. While the sampling line profile methods are constructed to accurately reproduce the cross sections at the given sampling points, the binned methods guarantee the integrated cross section within a given bin, regardless of the number of grid points or integration intervals. Accordingly, the binned methods are especially advantageous when the conservation of the total flux, absorption or emission coefficients, is required. The sampling method are preferred when the accuracy of the cross sections at the given frequencies is critical.

Figure 11 gives examples of the Doppler, Lorentzian and Voigt line profiles calculated using the sampling and binned methods for a single normalised transition in the wavenumber range $[4064, 4065] \text{ cm}^{-1}$ at the temperature $T = 300 \text{ K}$ and pressure $P = 1 \text{ bar}$ using the bin size of 0.125 cm^{-1} (8 points per cm^{-1}) with a cutoff of 100 cm^{-1} . The lines show the correct line profiles modelled, while points represent the corresponding calculated values. For the sampling method, the profile lines go exactly through the calculated cross section points, while for the binned method, the points are shifted away (up or down) from the curve in order to maintain the integrated cross section (unity in this case) as follows:

$$\int_{-\infty}^{\infty} f_{\tilde{\nu}_{fi}}(\tilde{\nu}) d\nu \equiv \sum_i f_{\tilde{\nu}_{fi}}(\tilde{\nu}_i) \Delta\nu, \quad (53)$$

where the rectangular integration rule is used.

The profile normalisation error given as the difference of the integrated area from unity of a (normalised) line profile is given by definition

$$\epsilon_{\text{area}} = 1 - \sum f_{\tilde{\nu}_{fi}}(\tilde{\nu}) \Delta\tilde{\nu}. \quad (54)$$

For the binned methods, ϵ_{area} is zero by construction, while for the sampling methods, it depends on the number of sampling points. Choosing an adequate number of sampling points is critical in radiative transfer applications (Chubb et al. 2021). Under-sampling can lead to underestimation of opacity, while too many points can make the calculations too slow. The profile sampling error can be defined a root-mean-squares (RMS) error as a deviation of the calculated from the exact profiles at the sampling points as given by

$$\epsilon_{\text{sampling}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (f(\tilde{\nu}_i) - f_{\text{calc}}(\tilde{\nu}_i))^2}, \quad (55)$$

where N is the number of sampling points. The sampling error $\epsilon_{\text{sampling}}$ should be zero for the sampling methods by construction (within the numerical accuracy).

Figure 12 gives a detailed analysis of the line profile errors ϵ_{area} and $\epsilon_{\text{sampling}}$ for a single normalised transition at $\tilde{\nu}_{if} = 4064.574702 \text{ cm}^{-1}$ for different temperatures T , pressures P , number of points N_{points} and bin sizes $\Delta\tilde{\nu}$ and a large wavenumber range of $[0, 8000] \text{ cm}^{-1}$ is used. All sampling methods have negligible $\epsilon_{\text{sampling}}$, while all binned methods should lead to perfect $\epsilon_{\text{sampling}}$, at least within the numerical error.

For the cross-comparisons, the Gaussian line profile is always the most accurate sampling line profile to preserve the profile area even with smaller number of points due to its compact shape. The SciPy Voigt and Humlíček Voigt profiles provide similar accuracy. For constant pressure and grid number, the line profiles uncertainty decreases with increasing temperature, see Figure 12 (first panel). In contrast, line profiles show a loss of accuracy as the pressure or grid number increase, see the second and third panels. The binned Lorentzian and Voigt profiles appear to degrade at higher pressures due to their non-negligible wings cutoff beyond the interval $[0, 8000] \text{ cm}^{-1}$.

The area errors and the sampling errors of different line profiles $\epsilon_{\text{sampling}}$ are shown in the fourth panel of Figure 12. While the sampling

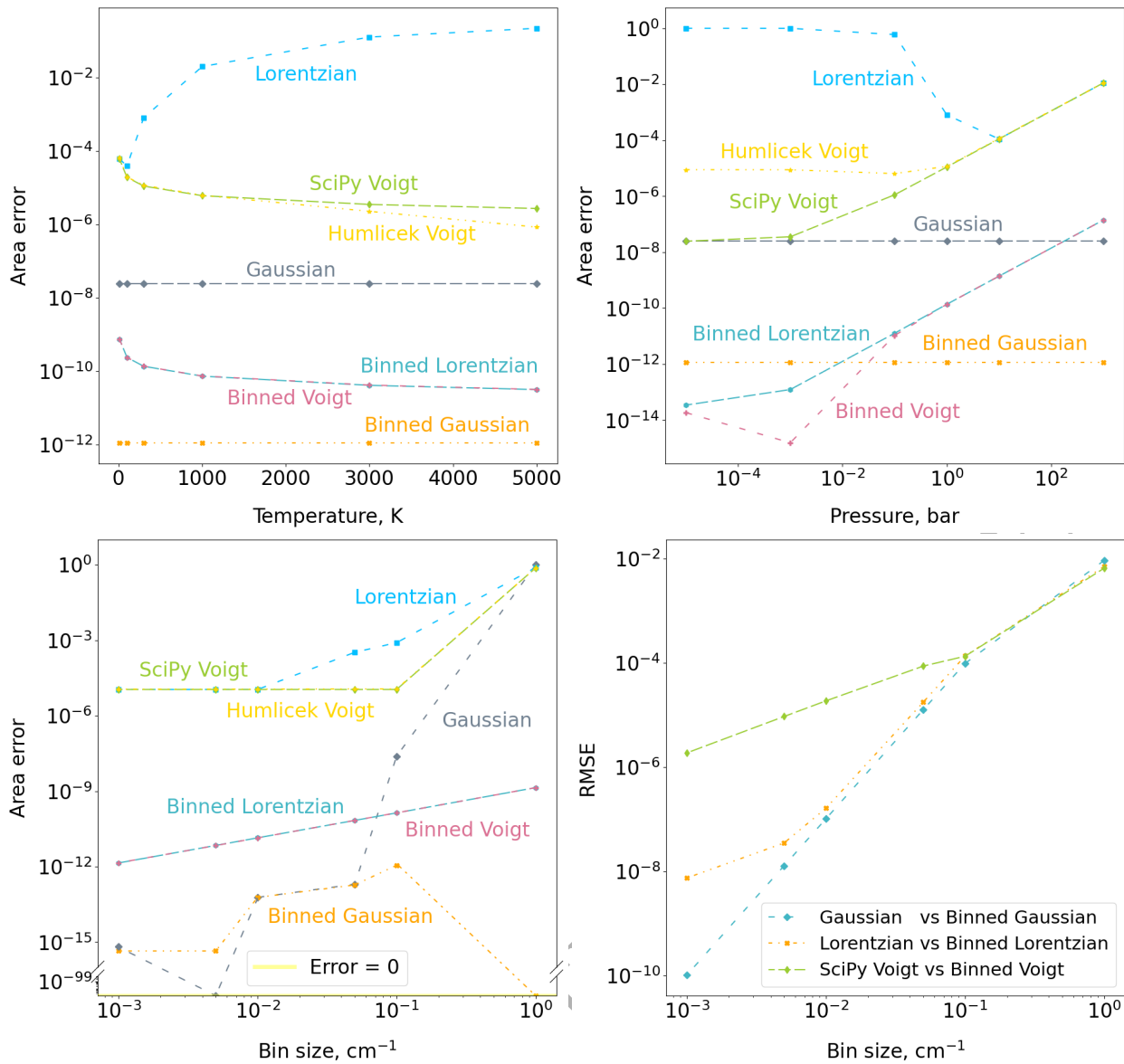


Figure 12. Line profile errors ϵ_{area} (Eq. (54)) and $\epsilon_{\text{sampling}}$ (Eq. (55)) for different line profiles representing the same single normalised transition ($\tilde{\nu}_{if} = 4064.574702 \text{ cm}^{-1}$), including Gaussian ($\alpha_G = 0.1$), Lorentzian (γ_L is estimated by $\gamma_{\text{ref}} = 0.07$ and $n_L = 0.5$) and Voigt, for the large wavenumber range of $[0, 8000] \text{ cm}^{-1}$ at different temperatures T , pressures P and bin sizes $\Delta\tilde{\nu}$. (top left): ϵ_{area} for different temperatures T at $P = 1$ bar, number of grid points $N_{\text{points}} = 80001$ (bin size $\Delta\tilde{\nu} = 0.1 \text{ cm}^{-1}$) without a cutoff; (top right): ϵ_{area} for different pressures P at $T = 300 \text{ K}$, $\Delta\tilde{\nu} = 0.1 \text{ cm}^{-1}$ without a cutoff; (bottom left): ϵ_{area} for different number of grid points N_{points} (bin sizes $\Delta\tilde{\nu}$) at $T = 300 \text{ K}$, $P = 1$ bar without a cutoff; (bottom right): $\epsilon_{\text{sampling}}$ for different number of grid points N_{points} (bin sizes $\Delta\tilde{\nu}$) at $T = 300 \text{ K}$, $P = 1$ bar and cutoff = 100 cm^{-1} .

profiles are almost perfect and perform better with more grid points, the binned profile can introduce a substantial error especially when the resolution is low. Generally, in radiative transfer applications direct sampling line profiles are preferred (Yurchenko et al. 2018a) especially for coarser grids.

We ignore the pseudo-Voigt profiles in this analysis as having too low accuracy.

3.7.2 Comparison of program run times with different line profiles

To compare the program running times for computing cross sections with different line profiles, in Table 23 we give examples of cross sections calculations for different numbers of transitions, bin sizes (numbers of grid points) and filters used.

From Table 23, we can find that the Lorentzian profile is the fastest followed by the Doppler, Gaussian and SciPy Voigt profiles. Pseudo-Voigt profiles process data with NumExpr (Cooke et al. 2009) which is a fast numerical expression evaluator for NumPy. As an approximation method commonly in use, the Humlíček Voigt profile runs rather slowly. Because of the increased number of steps in binned profile calculations, the program costs more time than the sampling methods. The SciPy wofz Voigt, Humlíček Voigt and Rocco pseudo-Voigt profiles are all have

Table 23. Example of the program run times on 32 cores, in s, for computing cross sections using different line profiles at temperatures $T = 300$ K, pressures $P = 1$ bar and cutoff = 25 cm^{-1} for the ExoMol H₂O POKAZATEL line list (Polyansky et al. 2018), for the wavenumber range [20000, 22000] cm^{-1} . The left panel compares the run times for different numbers of transitions 10^4 , 10^6 , 10^8 with bin size of 0.1 cm^{-1} (20001 grid points) without any filters. The middle panel compares the run times for different numbers of transitions 10^4 , 10^6 , 10^8 with bin size of 0.1 cm^{-1} (20001 grid points) with uncertainty and threshold filters. The right four columns compare the run times for different bin sizes ($0.1, 0.01, 0.005, 0.001 \text{ cm}^{-1}$) and the numbers of grid points (20001, 200001, 400001, 2000001) using 10^8 transitions with uncertainty and threshold filters.

Line profiles	Uncertainty filter: none Threshold filter: none Bin size $\Delta\tilde{\nu} = 0.1 \text{ cm}^{-1}$ $N_{\text{points}} = 20001$ $N_{\text{transitions}} =$			Uncertainty filter: 0.01 cm^{-1} Threshold filter: $10^{-30} \text{ cm/molecule}$ Bin size $\Delta\tilde{\nu} = 0.1 \text{ cm}^{-1}$ $N_{\text{points}} = 20001$ $N_{\text{transitions}} =$			Uncertainty filter: 0.01 cm^{-1} Threshold filter: $10^{-30} \text{ cm/molecule}$ $N_{\text{transitions}} = 10^8$ $N_{\text{points}} = 2000/\Delta\tilde{\nu} + 1$ Bin size (cm^{-1}) $\Delta\tilde{\nu} =$		
	10^4	10^6	10^8	10^4	10^6	10^8	0.01	0.005	0.001
Doppler	2.70	6.79	54.63	3.01	2.64	8.00	7.66	9.42	13.43
Binned Doppler	4.37	10.52	114.23	3.98	4.00	10.12	9.45	12.65	19.66
Gaussian	2.84	7.43	52.53	3.74	3.85	9.37	8.72	11.09	15.00
Binned Gaussian	4.42	10.85	119.77	3.85	3.96	10.08	9.39	12.28	20.03
Lorentzian	2.83	7.39	51.69	3.87	3.97	9.47	8.77	10.46	14.15
Binned Lorentzian	4.14	9.16	61.24	3.95	3.95	9.67	9.24	11.31	14.73
SciPy Voigt	2.99	9.77	83.19	3.87	3.91	9.34	8.85	10.83	13.21
SciPy Wofz Voigt	3.10	10.26	121.01	3.80	3.97	9.71	9.09	11.73	17.27
Humlíček Voigt	3.24	10.69	161.25	3.90	4.05	10.15	12.62	20.03	46.62
Thompson pseudo-Voigt	2.99	9.89	85.86	3.80	3.90	10.14	9.27	11.63	19.85
Kielkopf pseudo-Voigt	4.01	10.15	85.98	3.82	3.94	10.08	9.57	13.09	19.72
Olivero pseudo-Voigt	3.93	9.93	88.55	3.81	3.89	10.01	9.58	12.48	20.44
Liu-Lin pseudo-Voigt	3.98	10.03	85.20	3.88	3.99	10.18	9.40	14.02	19.63
Rocco pseudo-Voigt	3.97	10.09	85.47	3.97	3.98	9.83	9.56	13.08	20.16
Binned Voigt	6.19	13.80	657.68	3.95	4.35	10.32	22.38	38.78	107.06

similar accuracy but significantly slower than the SciPy Voigt profile. Binned Voigt profiles are more accurate than standard sampling Voigt profiles, but take much more computer time.

To summarise, considering its higher accuracy and relatively short run times, we recommend the SciPy Voigt profil for calculating cross sections.

3.7.3 Comparison of program run times with PyExoCross, ExoCross and RADIS

We tested run times of PyExoCross, ExoCross and RADIS programs on 32 cores. PyExoCross makes good use of the cores and cost 371 s on reading files, processing data, calculating absorption cross sections and saving results for H₂O at temperature $T = 3000$ K, pressure $P = 1$ bar, cutoff = 25 cm^{-1} , uncertainty $\leq 0.01 \text{ cm}^{-1}$, threshold $\geq 10^{-40} \text{ cm/molecule}$, wavenumber range from 0 cm^{-1} to 41200 cm^{-1} with SciPy Voigt line profile using the ExoMol ¹H₂¹⁶O line list of dataset POKAZATEL (Polyansky et al. 2018). Fortran program ExoCross cost 3600 s for whole processes with the same conditions; however, only about 300 s of this is actually processing data with the wall clock time dominated by reading and unpacking the files.

RADIS downloads ExoMol compressed bz2 files then converts them into HDF5 files and saves both original files and converted files to local for next time calculating. To compare with RADIS, PyExoCross cost 27 s (including reading compressed bz2 files, processing data, calculating cross sections and saving results) and RADIS cost 25 s (not including times for downloading, decompressing and converting files, includes reading HDF5 files, processing data and calculating cross sections) on calculating absorption cross sections of ExoMol ¹H₂¹⁶O line list of dataset POKAZATEL in the wavenumber range [20000,20500] cm^{-1} at temperature $T = 300$ K, pressure $P = 1$ bar, cutoff = 10^{-27} cm^{-1} , threshold $\geq 10^{-40} \text{ cm/molecule}$ with Voigt line profile.

3.8 Additional functionality

PyExoCross provides the following additional functionality for users: (i) locating line lists in the ExoMol database with uncertainties provided as part of the State file; (ii) checking for updated line lists compared with the last or new version of the master file, or the version the users saved last time; (iii) to downloading the States and Transition files via the ExoMol API instead of downloading files one by one.

4 CALCULATION PROTOCOL

The PyExoCross program processes line lists with the following steps (see Figure 13):

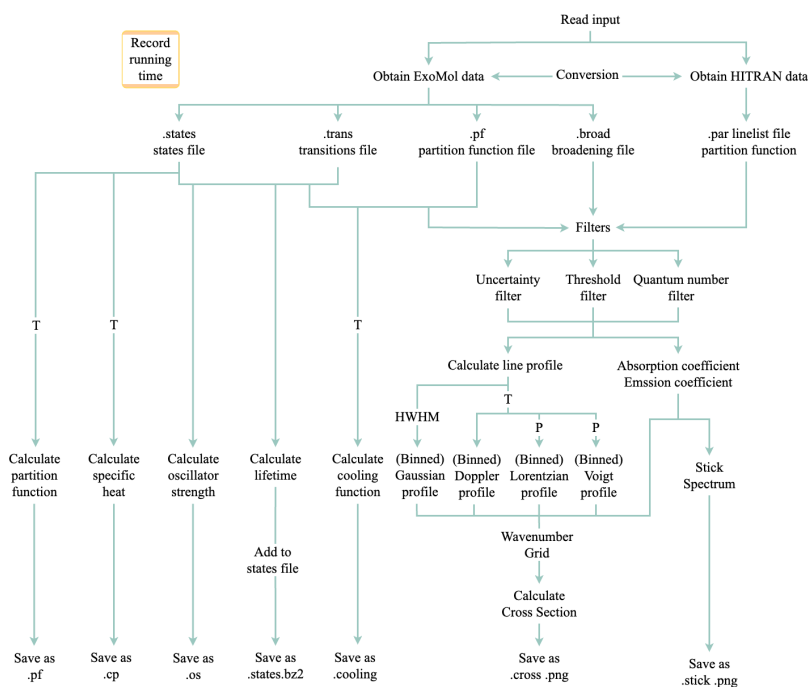


Figure 13. Calculation protocol of the PyExoCross program. *T* and *P* refer to temperature and pressure separately.

- Populate the input file with the necessary data and settings to be read by PyExoCross;
- Obtain (download) data from the ExoMol and/or HITRAN websites (if required)
- Read the ExoMol line lists files: (.states) states file, (.trans) transitions file(s), (.pf) partition function file and (.broad) broadening file or the HITRAN line lists (.par) files and partition function;
- Compute partition function (if required);
- Compute specific heat (if required);
- Compute cooling function (if required);
- Compute radiative lifetime (if required);
- Compute oscillator strength (if required);
- For intensities, apply the uncertainty or quantum numbers filters (if required);
- Compute a line intensity (absorption or emission coefficient, if required);
- Compute line profile (if required);
- Compute cross sections on a grid of wavenumbers (if required);
- Save the results of partition functions, specific heats, cooling functions, radiative lifetimes and cross sections into the separate files;
- Plot cross sections (if required);
- Report program running time.

5 DATA FORMAT

PyExoCross can use the input data in either ExoMol or HITRAN format. It can also convert between different formats including HITRAN or ExoMol. Table 24 gives all file types in the ExoMol database. PyExoCross available file types are described in the above sections.

5.1 HITRAN format

Compared to the ExoMol format line list files (.states and .trans), the HITRAN format line list file (.par) includes the parameters related to both upper-state and lower-state. Table 17 gives the format of the HITRAN line-by-line records. The description of the HITRAN parameters is specified in Table 18. PyExoCross processes the HITRAN format data to obtain the parameters for calculating cooling functions, oscillator strength, stick spectra and cross sections. The partition function, molar mass and fractional abundance are available from <https://hitran.org/docs/iso-meta/>. PyExoCross extracts these parameters from the webpage directly and automatically instead of entering these parameters by hand. The partition functions at different temperatures are recorded in the .txt file online which can be mined by PyExoCross.

Table 24. Specification of the ExoMol file types (contents in brackets are optional) (Tennyson et al. 2023).

File extension	PyExoCross available	N_{files}	File DSname	Contents
.all	Input	1	Master	Single file defining contents of the ExoMol database.
.def	Input	N_{tot}	Definition	Defines contents of other files for each isotopologue.
.states	Input/Output	N_{tot}	States	Energy levels, quantum numbers, lifetimes, (Landé g -factors, Uncertainties).
.trans	Input/Output	a	Transitions	Einstein A coefficients, (wavenumber).
.broad	Input/Output	N_{mol}	Broadening	Parameters for pressure-dependent line profiles.
.cross	Output	b	Cross sections	Temperature or temperature and pressure-dependent cross sections.
.kcoef	No	c	k -coefficients	Temperature and pressure-dependent k -coefficients.
.pf	Input/Output	N_{tot}	Partition function	Temperature-dependent partition function, (cooling function).
.cf	Output	N_{tot}	Cooling function	Temperature-dependent cooling function.
.cp	Output	N_{tot}	Specific heat	Temperature-dependent specific heat.
.dipoles	No	N_{tot}	Dipoles	Transition dipoles including phases.
.super	No	d	Super-lines	Temperature dependent super-lines (histograms) on a wavenumber grid.
.nm	No	e	VUV cross sections	Temperature and pressure dependent VUV cross-sections (wavelength, nm).
.fits, .h5, .kta	No	f	Opacities	Temperature and pressure dependent opacities for radiative-transfer applications.
.overview	No	N_{mol}	Overview	Overview of datasets available.
.readme	No	N_{iso}	Readme	Specifies data formats.
.model	No	N_{iso}	Model	Model specification.
.cont	No	$N_{\text{iso}}/0$	Continuum	Continuum contribution to the photoabsorption.
.pdef	No	N_{tot}	Definition	Defines contents of .photo files for each isotopologue.
.photo	No	N_{iso}	Photodissociation	Photodissociation cross sections for each lower state.

N_{files} total number of possible files;

N_{mol} Number of molecules in the database;

N_{tot} is the sum of N_{iso} for the N_{mol} molecules in the database;

N_{iso} Number of isotopologues considered for the given molecule.

^a There are N_{tot} sets of .trans files but for molecules with large numbers of transitions the .trans files are subdivided into wavenumber regions.

^b There are N_{cross} sets of .cross files for isotopologue.

^c There are N_{kcoef} sets of .kcoef files for each isotopologue.

^d There are N_T sets of T -dependent super-lines.

^e There are N_{VUV} sets of VUV cross sections.

^f Set of opacity files in the format native to specific radiative-transfer programs.

Table 25. Default units used in the PyExoCross program.

Quantity	Label	Units
Partition function	$Q(T)$	none
Specific heat	$C_p(T)$	$\text{J K}^{-1} \text{mol}^{-1}$
Cooling function	$W(T)$	$\text{erg (s molecule str)}^{-1}$
Radiative lifetime	τ_i	s
Oscillator strength	f	none
Molar mass	M	Dalton
Wavenumber		cm^{-1}
Wavelength		μm
Temperature	T	K
Pressure	P	bar
Absorption coefficient	I_{fi}	cm molecule^{-1}
Emission coefficient	ϵ_{if}	$\text{erg (s molecule str)}^{-1}$
Absorption cross sections	σ_{ab}	$\text{cm}^2 \text{molecule}^{-1}$
Emission cross sections	σ_{em}	$\text{erg cm (s molecule str)}^{-1}$

5.2 Units

The default labels and units used in the PyExoCross program are listed in Table 25. The physical and astronomical constants in calculations are shown in Listing 5. `astropy.constants` is a sub-package of `Astropy` where `Astropy` is a community Python package for astronomy (Robitaille et al. 2013). Some of the parameters used by PyExoCross are not in SI units but in cgs units (e.g. erg).

```

import astropy.constants as ac
Tref = 296.0 # Reference temperature is 296 K
Pref = 1.0 # Reference pressure is 1 bar
N_A = ac.N_A.value # Avogadro number (1/mol)
h = ac.h.to('erg s').value # Planck's constant (erg s)
c = ac.c.to('cm/s').value # Velocity of light (cm/s)
kB = ac.k_B.to('erg/K').value # Boltzmann's const (erg/K)
R = ac.R.to('J / (K mol)').value # Molar gas constant (J/(K mol))
c2 = h * c / kB # Second radiation constant (cm K)

```

Listing 5: Python code for calculating the physical and astronomical constants

6 CONCLUSIONS

We have developed PyExoCross, a python new program for calculating the spectroscopic properties of molecules with spectral line lists. PyExoCross is the Python version of the Fortran program ExoCross (Yurchenko et al. 2018a). Some CPU parallelization is available, at present there is no GPU support. The spectral line lists of the ExoMol database are used to generate partition functions, specific heats, cooling functions, radiative lifetimes, oscillator strengths, stick spectra and cross sections by PyExoCross. At present, PyExoCross can ingest data from the ExoMol, HITRAN, and HITEMP databases to provide parts of the functionalities supported by the ExoCross program. Other features will be developed later, such as other data formats while this project will also implement the processing of other databases in the future.

Future enhancements to PyExoCross include non-LTE (non-local thermodynamic equilibrium) calculations, generation of k-tables and opacity production. A library of fractional isotopic abundances will be built for the PyExoCross program to calculate the intensities for different distributions of isotopologues. PyExoCross will also support more molecular spectral databases in future versions, along with support for data type and data format conversions among different databases.

ACKNOWLEDGEMENTS

We thank Ahmed Al-Refaie for helpful discussions and suggestions on the methods of this work. This work was supported by the European Research Council under Advanced Investigator Project 883830.

DATA AVAILABILITY

The PyExoCross program is publicly accessible on GitHub (<https://github.com/ExoMol/PyExoCross.git>) and its program manual webpage is <https://pyexocross.readthedocs.io/>.

The ExoCross Fortran program is freely available from the GitHub (<https://github.com/ExoMol/ExoCross.git>). The ExoCross manual webpage is <https://exocross.readthedocs.io/>.

The molecular line lists discussed in this paper and many more are available from the ExoMol website (<https://www.exomol.com/>).

REFERENCES

- Abramowitz, M. & Stegun, I., 1972. Handbook of Mathematical Functions 10th Printing with Corrections, *Natl. Bur. Stand. Appl. Math. Ser.*, **55**.
- Anisman, L., Chubb, K. L., Elsey, J., Al-Refaie, A., Changeat, Q., Yurchenko, S. N., Tennyson, J., & Tinetti, G., 2022. Cross-sections for heavy atmospheres: H₂O continuum, *J. Quant. Spectrosc. Radiat. Transf.*, **278**, 108013.
- Armstrong, B., 1967. Spectrum line profiles: the Voigt function, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **7**(1), 61–88.
- Barber, R. J., Strange, J. K., Hill, C., Polyansky, O. L., Mellau, G. C., Yurchenko, S. N., & Tennyson, J., 2014. ExoMol line lists – III. An improved hot rotation-vibration line list for HCN and HNC, *MNRAS*, **437**, 1828–1835.
- Barklem, P. S. & Collet, R., 2016. Partition functions and equilibrium constants for diatomic molecules and atoms of astrophysical interest, *A&A*, **588**, A96.
- Barton, E. J., Hill, C., Czurylo, M., Li, H.-Y., Hyslop, A., Yurchenko, S. N., & Tennyson, J., 2017. The ExoMol diet: H₂ and He line-broadening parameters, *J. Quant. Spectrosc. Radiat. Transf.*, **203**, 490–495.
- Chubb, K. L., Rocchetto, M., Yurchenko, S. N., Min, M., Waldmann, I., Barstow, J. K., Molliere, P., Al-Refaie, A. F., Phillips, M., & Tennyson, J., 2021. The ExoMolOP Database: Cross-sections and K-tables for Molecules of Interest in High-Temperature Exoplanet Atmospheres, *A&A*, **646**, A21.
- Cooke, D., Hochberg, T., Alted, F., Vilata, I., Wiebe, M., de Menten, G., Valentino, A., & McLeod, R., 2009. NumExpr: Fast numerical expression evaluator for NumPy.
- Coppola, C. M., Lodi, L., & Tennyson, J., 2011. Radiative cooling functions for primordial molecules, *MNRAS*, **415**, 487–493.
- Demtröder, W., 1981. Laser spectroscopy: Basic concepts and instrumentation, *NASA STI/Recon Technical Report A*, **82**, 12273.
- Di Rocco, H. O. & Cruzado, A., 2012. The Voigt profile as a sum of a Gaussian and a Lorentzian functions, when the weight coefficient depends only on the widths ratio, *Acta Physica Polonica A*, **122**.
- Endres, C. P., Schlemmer, S., Schilke, P., Stutzki, J., & Müller, H. S. P., 2016. The Cologne Database for Molecular Spectroscopy, CDMS, in the Virtual Atomic and Molecular Data Centre, VAMDC, *J. Mol. Spectrosc.*, **327**, 95–104.

- Faddeeva, V. N., Terent'ev, N. M., Fok, V. A., Fry, D. G., & Hons, B. A., 1961. *Tables of values of the function $\omega(z) = \exp(-z^2)(1 + 2i/\sqrt{\pi} \int_0^z \exp(t^2) dt$ for complex argument*, Pergamon Press Oxford.
- Feller, W., 1971. *An introduction to probability theory and its applications, Volume 2*, p. 704, New York: John Wiley & Sons Inc., 2nd edn.
- Gamache, R. R., Roller, C., Lopes, E., Gordon, I. E., Rothman, L. S., Polyansky, O. L., Zobov, N. F., Kyuberis, A. A., Tennyson, J., Yurchenko, S. N., Császár, A. G., Furtenbacher, T., Huang, X., Schwenke, D. W., Lee, T. J., Drouin, B. J., Tashkun, S. A., Perevalov, V. I., & Kochanov, R. V., 2017. Total Internal Partition Sums for 167 isotopologues of 53 molecules important in planetary atmospheres: application to HITRAN2016 and beyond, *J. Quant. Spectrosc. Radiat. Transf.*, **203**, 70–87.
- Gandhi, S., Brogi, M., Yurchenko, S. N., Tennyson, J., Coles, P. A., Webb, R. K., Birkby, J. L., Guilluy, G., & Madhusudhan, N., 2020. Molecular Cross Sections for High Resolution Spectroscopy of super Earths, warm Neptunes and Hot Jupiters, *MNRAS*, **495**, 224–237.
- Gérard, J.-C., Gkouvelis, L., Bonfond, B., Gladstone, G. R., Mura, A., Adriani, A., Grodent, D., Hue, V., & Greathouse, T. K., 2023. H3+ cooling in the Jovian aurora: Juno remote sensing observations and modeling, *Icarus*, **389**, 115261.
- Gharib-Nezhad, E. S., Batalha, N. E., Chubb, K., Freedman, R., Gordon, I. E., Gamache, R. R., Hargreaves, R. J., Lewis, N. K., Tennyson, J., & Yurchenko, S. N., 2024. The impact of spectral line wing cut-off: Recommended standard method with application to MAESTRO opacity database, *RASTI*, **3**, 44–55.
- Goldenstein, C. S., Miller, V. A., Spearrin, R. M., & Strand, C. L., 2017. SpectraPlot.com: Integrated spectroscopic modeling of atomic and molecular gases, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **200**, 249–257.
- Gordon, I. E., Rothman, L. S., Hargreaves, R. J., Hashemi, R., Karlovets, E. V., Skinner, F. M., Conway, E. K., Hill, C., Kochanov, R. V., Tan, Y., Wcisło, P., Finenko, A. A., Nelson, K., Bernath, P. F., Birk, M., Boudon, V., Campargue, A., Chance, K. V., Coustenis, A., Drouin, B. J., Flaud, J., Gamache, R. R., Hodges, J. T., Jacquemart, D., Mlawer, E. J., Nikitin, A. V., Perevalov, V. I., Rotger, M., Tennyson, J., Toon, G. C., Tran, H., Tyuterev, V. G., Adkins, E. M., Baker, A., Barbe, A., Canè, E., Császár, A. G., Dudaryonok, A., Egorov, O., Fleisher, A. J., Fleurbaey, H., Foltynowicz, A., Furtenbacher, T., Harrison, J. J., Hartmann, J., Horneman, V., Huang, X., Karman, T., Karns, J., Kassi, S., Kleiner, I., Kofman, V., Kwabia-Tchana, F., Lavrentieva, N. N., Lee, T. J., Long, D. A., Lukashevskaya, A. A., Lyulin, O. M., Makhnev, Y. Y., Matt, W., Massie, S. T., Melosso, M., Mikhailenko, S. N., Mondelain, D., Müller, H. S. P., Naumenko, O. V., Perrin, A., Polyansky, O. L., Raddaoui, E., Raston, P. L., Reed, Z. D., Rey, M., Richard, C., Tóbiás, R., Sadiek, I., Schwenke, D. W., Starikova, E., Sung, K., Tamassia, F., Tashkun, S. A., Vander Auwera, J., Vasilenko, I. A., Viganin, A. A., Villanueva, G. L., Vispoel, B., Wagner, G., Yachmenev, A., & Yurchenko, S. N., 2022. The HITRAN2020 molecular spectroscopic database, *J. Quant. Spectrosc. Radiat. Transf.*, **277**, 107949.
- Grimm, S. L., Malik, M., Kitzmann, D., Guzmán-Mesa, A., Hoeijmakers, H. J., Fisher, C., ao M. Mendonça, J., Yurchenko, S. N., Tennyson, J., Kurucz, R. L., & Heng, K., 2021. HELIOS-K 2.0 and an Open-source Opacity Database for Exoplanetary Atmospheres, *ApJS*, **253**, 30.
- Hargreaves, R. J., Gordon, I. E., Rothman, L. S., Tashkun, S. A., Perevalov, V. I., Lukashevskaya, A. A., Yurchenko, S. N., Tennyson, J., & Müller, H. S. P., 2019. Spectroscopic line parameters of NO, NO₂, and N₂O for the HITEMP database, *J. Quant. Spectrosc. Radiat. Transf.*, **232**, 35–53.
- Herbert, F., 1974. Spectrum line profiles: a generalized Voigt function including collisional narrowing, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **14**(9), 943–951.
- Hill, C., 2020. *Learning scientific programming with Python*, Cambridge University Press.
- Hill, C., Yurchenko, S. N., & Tennyson, J., 2013. Temperature-dependent molecular absorption cross sections for exoplanets and other atmospheres, *Icarus*, **226**, 1673–1677.
- Huang, X., Schwenke, D. W., & Lee, T. J., 2021. What It Takes to Compute Highly Accurate Rovibrational Line Lists for Use in Astrochemistry, *Acc. Chem. Res.*, **54**, 1311–1321.
- Hui, A. K., Armstrong, B. H., & Wray, A. A., 1978. Rapid computation of the Voigt and complex error functions, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **19**(5), 509–516.
- Humlíček, J., 1979. An efficient method for evaluation of the complex probability function: the Voigt function and its derivatives, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **21**(4), 309–313.
- Humlíček, J., 1982. Optimized computation of the Voigt and complex probability functions, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **27**(4), 437–444.
- Ida, T., Ando, M., & Toraya, H., 2000. Extended pseudo-Voigt function for approximating the Voigt profile, *Journal of Applied Crystallography*, **33**(6), 1311–1316.
- Irwin, A. W., 1981. Polynomial partition-function approximations of 344 atomic species and molecular-species, *ApJS*, **45**(4), 621–633.
- Jacquinet-Husson, N., Crepeau, L., Armante, R., Boutammine, C., Chédin, A., Scott, N. A., Crevoisier, C., Capelle, V., Boone, C., Poulet-Crovisier, N., Barbe, A., Campargue, A., Benner, D. C., Benilan, Y., Bézard, B., Boudon, V., Brown, L. R., Coudert, L. H., Coustenis, A., Dana, V., Devi, V. M., Fally, S., Fayt, A., Flaud, J.-M., Goldman, A., Herman, M., Harris, G. J., Jacquemart, D., Jolly, A., Kleiner, I., Kleinböhl, A., Kwabia-Tchana, F., Lavrentieva, N., Lacombe, N., Xu, L.-H., Lyulin, O. M., Mandin, J.-Y., Maki, A., Mikhailenko, S., Müller, C. E., Mishina, T., Moazzen-Ahmadi, N., Müller, H. S. P., Nikitin, A., Orphal, J., Perevalov, V., Perrin, A., Petkie, D. T., Predoi-Cross, A., Rinsland, C. P., Remedios, J. J., Rotger, M., Smith, M. A. H., Sung, K., Tashkun, S., Tennyson, J., Toth, R. A., Vandaele, A.-C., & Vander Auwera, J., 2011. The 2009 edition of the GEISA spectroscopic database, *J. Quant. Spectrosc. Radiat. Transf.*, **112**, 2395–2445.
- Johnson, N. L., Kotz, S., & Balakrishnan, N., 1994. *Continuous univariate distributions, volume 1*, chap. 16, New York: John Wiley & Sons Inc.
- Jørgensen, U. G., Larsson, M., Iwamae, A., & Yu, B., 1996. Line intensities for CH and their application to stellar atmospheres, *A&A*, **315**, 204.
- Karlovets, E. V., Gordon, I. E., Rothman, L. S., Hashemi, R., Hargreaves, R. J., Toon, G., Campargue, A., Perevalov, V. I., Čermak, P., Birk, M., Wagner, G., Hodges, J. T., Tennyson, J., & Yurchenko, S. N., 2021. The update of the line positions and intensities in the line list of carbon dioxide for the HITRAN2020 spectroscopic database, *J. Quant. Spectrosc. Radiat. Transf.*, **276**, 107896.
- Kielkopf, J. F., 1973. New approximation to the Voigt function with applications to spectral-line profile analysis, *JOSA*, **63**(8), 987–995.
- Kochanov, R. V., Gordon, I., Rothman, L., Wcisło, P., Hill, C., & Wilzewski, J., 2016. HITRAN Application Programming Interface (HAPI): A comprehensive approach to working with spectroscopic data, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **177**, 15–30.
- Kochanov, R. V., Gordon, I. E., Rothman, L. S., Tan, Y., Karns, J., Matt, W., & Hill, C., 2018. Updates and Current Status of the HITRAN Application Programming Interface (HAPI), in *73rd International Symposium on Molecular Spectroscopy*, p. MI07.
- Kuntz, M., 1997. A new implementation of the Humlicek algorithm for the calculation of the Voigt profile function, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **57**(6), 819–824.
- Li, G., Gordon, I. E., Rothman, L. S., Tan, Y., Hu, S.-M., Kassi, S., Campargue, A., & Medvedev, E. S., 2015. Rovibrational line lists for nine isotopologues of the CO molecule in the X¹Σ⁺ ground electronic state, *ApJS*, **216**, 15.
- Liu, Y., Lin, J., Huang, G., Guo, Y., & Duan, C., 2001. Simple empirical analytical approximation to the Voigt profile, *JOSA B*, **18**(5), 666–672.
- Madhusudhan, N., 2019. Exoplanetary Atmospheres: Key Insights, Challenges, and Prospects, *ARA&A*, **57**, 617–663.
- Martin, J. M. L., François, J.-P., & Gijbels, R., 1992. First principles computation of thermochemical properties beyond the harmonic approximation. i. method

- and application to the water molecule and its isotopomers, *J. Chem. Phys.*, **96**, 7633.
- McKinney, W. et al., 2010. Data Structures for Statistical Computing in Python, in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX.
- Melnikov, V. V., Yurchenko, S. N., Tennyson, J., & Jensen, P., 2016. Radiative cooling of H_3O^+ and its deuterated isotopologues, *Phys. Chem. Chem. Phys.*, **18**, 26268–26274.
- Miller, S., Stallard, T., Tennyson, J., & Melin, H., 2013. Cooling by H_3^+ emission, *J. Phys. Chem. A*, **117**, 9770–9777.
- Müller, H. S., Endres, C., Stutzki, J., & Schlemmer, S., 2013. The CDMS view on molecular data needs of Herschel, SOFIA, and ALMA, in *AIP Conference Proceedings*, vol. 1545, pp. 96–109, American Institute of Physics.
- Neale, L. & Tennyson, J., 1995. A high temperature partition function for H_3^+ , *ApJ*, **454**, L169–L173.
- Neale, L., Miller, S., & Tennyson, J., 1996. Spectroscopic properties of the H_3^+ molecule: a new calculated line list, *The Astrophysical Journal*, **464**, 516.
- Ngo, N., Lisak, D., Tran, H., & Hartmann, J.-M., 2013. An isolated line-shape model to go beyond the Voigt profile in spectroscopic databases and radiative transfer codes, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **129**, 89–100.
- Nguyen, T. G., Cowan, N. B., Pierrehumbert, R. T., Lupu, R. E., & Moores, J. E., 2022. The impact of ultraviolet heating and cooling on the dynamics and observability of lava planet atmospheres, *Monthly Notices of the Royal Astronomical Society*, **513**(4), 6125–6133.
- Olivero, J. J. & Longbothum, R., 1977. Empirical fits to the Voigt line width: A brief review, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **17**(2), 233–236.
- Orloski, R. & Trigueiros, A., 2001. Weighted oscillator strengths and lifetimes for the Si VIII spectrum, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **69**(1), 13–21.
- Owens, A., Tennyson, J., & Yurchenko, S. N., 2021. ExoMol line lists – XLI. High-temperature molecular line lists for the alkali metal hydroxides KOH and NaOH, *MNRAS*, **502**, 1128–1135.
- Owens, A., Dooley, S., McLaughlin, L., Tan, B., Zhang, G., Yurchenko, S. N., & Tennyson, J., 2022. ExoMol line lists – XLV. Rovibronic molecular line lists of calcium monohydride (CaH) and magnesium monohydride (MgH), *MNRAS*, **511**, 5448–5461.
- Owens, A., He, T., Hanicinec, M., Hill, C., Mohr, S., & Tennyson, J., 2023. LiDB: Database of molecular vibronic state radiative lifetimes for plasma processes, *Plasma Sources Sci. Technol.*, **32**, 085015.
- Pavlenko, Y., Tennyson, J., Yurchenko, S. N., Schmidt, M. R., Jones, H. R. A., Lyubchik, Y., & Suárez Mascareño, A., 2022. AlH lines in the blue spectrum of Proxima Centauri, *MNRAS*, **516**, 5655–5673.
- Pavlenko, Y. V., Yurchenko, S. N., & Tennyson, J., 2020. Analysis of first overtone bands of isotopologues of CO and SiO in stellar spectra, *A&A*, **633**, A52.
- Pickett, H. M., Poynter, R. L., Cohen, E. A., Delitsky, M. L., Pearson, J. C., & Müller, H. S. P., 1998. Submillimeter, millimeter, and microwave spectral line catalog, *J. Quant. Spectrosc. Radiat. Transf.*, **60**, 883–890.
- Pierluissi, J. H., Vanderwood, P. C., & Gomez, R. B., 1977. Fast calculational algorithm for the Voigt profile, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **18**(5), 555–558.
- Polyansky, O. L., Kyuberis, A. A., Zobov, N. F., Tennyson, J., Yurchenko, S. N., & Lodi, L., 2018. ExoMol molecular line lists XXX: a complete high-accuracy line list for water, *MNRAS*, **480**, 2597–2608.
- Qu, Q., Yurchenko, S. N., & Tennyson, J., 2021. ExoMol molecular line lists – XLII: Rovibronic molecular line list for the low-lying states of NO, *MNRAS*, **504**, 5768–5777.
- Rey, M., Nikitin, A. V., Babikov, Y. L., & Tyuterev, V. G., 2016. TheoReTS—An information system for theoretical spectra based on variational predictions from molecular potential energy and dipole moment surfaces, *Journal of Molecular Spectroscopy*, **327**, 138–158.
- Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., et al., 2013. Astropy: A community Python package for astronomy, *Astronomy & Astrophysics*, **558**, A33.
- Rothman, L., Gordon, I., Barber, R., Dothe, H., Gamache, R., Goldman, A., Perevalov, V., Tashkun, S., & Tennyson, J., 2010. HITEMP, the high-temperature molecular spectroscopic database, *J. Quant. Spectrosc. Radiat. Transf.*, **111**, 2139–2150.
- Rothman, L. S., Jacquemart, D., Barbe, A., Benner, D. C., Birk, M., Brown, L., Carleer, M., Chackerian Jr, C., Chance, K., Coudert, L. e. a., et al., 2005. The HITRAN 2004 molecular spectroscopic database, *Journal of quantitative spectroscopy and radiative transfer*, **96**(2), 139–204.
- Rustamkulov, Z., Sing, D. K., Mukherjee, S., May, E. M., Kirk, J., Schlawin, E., Line, M. R., Piaulet, C., Carter, A. L., Batalha, N. E., Goyal, J. M., Lopez-Morales, M., Lotheringer, J. D., MacDonald, R. J., Moran, S. E., Stevenson, K. B., Wakeford, H. R., Espinoza, N., Bean, J. L., Batalha, N. M., Benneke, B., Berta-Thompson, Z. K., Crossfield, I. J. M., Gao, P., Kreidberg, L., Powell, D. K., Cubillos, P. E., Gibson, N. P., Leconte, J., Molaverdikhani, K., Nikolov, N. K., Parmentier, V., Roy, P., Taylor, J., Turner, J. D., Wheatley, P. J., Aggarwal, K., Ahrer, E., Alam, M. K., Alderson, L., Allen, N. H., Banerjee, A., Barat, S., Barrado, D., Barstow, J. K., Bell, T. J., Blecic, J., Brande, J., Casewell, S., Changeat, Q., Chubb, K. L., Crouzet, N., Daylan, T., Decin, L., Desert, J., Mikal-Evans, T., Feinstein, A. D., Flagg, L., Fortney, J. J., Harrington, J., Heng, K., Hong, Y., Hu, R., Iro, N., Kataria, T., Kempton, E. m. r., Krick, J., Lendl, M., Lillo-Box, J., Louca, A., Lustig-Yaeger, J., Mancini, L., Mansfield, M., Mayne, N. J., Miguel, Y., Morello, G., Ohno, K., Palle, E., Petit Dit de la Roche, D. J. M., Rackham, B. V., Radica, M., Ramos-Rosado, L., Redfield, S., Rogers, L. K., Shkolnik, E. L., Southworth, J., Teske, J., Tremblin, P., Tucker, G. S., Venot, O., Waalkes, W. C., Welbanks, L., Zhang, X., & Zieba, S., 2023. Early Release Science of the exoplanet WASP-39b with JWST NIRSpec PRISM, *Nature*, **614**, 659+.
- Sánchez-Bajo, F. & Cumbreira, F., 1997. The use of the pseudo-Voigt function in the variance method of X-ray line-broadening analysis, *Journal of Applied Crystallography*, **30**(4), 427–430.
- Sauval, A. J. & Tatum, J. B., 1984. A set of partition functions and equilibrium constants for 300 diatomic molecules of astrophysical interest, *ApJS*, **56**, 193–209.
- Schreier, F., 2011. Optimized implementations of rational approximations for the Voigt and complex error function, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **112**(6), 1010–1025.
- Schreier, F., 2017. Computational aspects of speed-dependent voigt profiles, *J. Quant. Spectrosc. Radiat. Transf.*, **187**, 44 – 53.
- Schreier, F., 2018. The Voigt and complex error function: Humlíček’s rational approximation generalized, *Monthly Notices of the Royal Astronomical Society*, **479**(3), 3068–3075.
- Schreier, F. & Hochstaff, P., 2021. Computational aspects of speed-dependent Voigt and Rautian profiles, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **258**, 107385.
- Sharp, C. M. & Burrows, A., 2007. Atomic and molecular opacities for brown dwarf and giant planet atmospheres, *ApJS*, **168**, 140–166.
- Somogyi, W., Yurchenko, S. N., & Yachmenev, A., 2021. Calculation of electric quadrupole linestrengths for diatomic molecules: Application to the H_2 , CO, HF, and O_2 molecules, *J. Chem. Phys.*, **155**(21), 214303.
- Sousa-Silva, C., Hesketh, N., Yurchenko, S. N., Hill, C., & Tennyson, J., 2014. High Temperature partition functions and thermodynamic data for ammonia and phosphine, *J. Quant. Spectrosc. Radiat. Transf.*, **142**, 66–74.

- Squires, G. L., 2001. *Practical physics*, Cambridge university press.
- Tan, Y., Kochanov, V. R., Rothman, L. S., & Gordon, I. E., 2019. Introduction of Water-Vapor Broadening Parameters and Their Temperature-Dependent Exponents Into the HITRAN Database: Part I-CO₂, N₂O, CO, CH₄, O₂, NH₃, and H₂S, *J. Quant. Spectrosc. Radiat. Transf.*, **124**, 11580–11594.
- Tan, Y., Skinner, F. M., Samuels, S., Hargreaves, R. J., Hashemi, R., & Gordon, I. E., 2022. H-2, he, and co2 pressure-induced parameters for the hitran database. ii. line lists of co₂, n₂o, co, so₂, oh, ocs, h₂co, hcn, ph₃, h₂s, and geh₄, *ApJS*, **262**(2), 40.
- Tennyson, J. & Miller, S., 1989. A program suite for the calculation of ro-vibrational spectra of triatomic molecules, *Comput. Phys. Commun.*, **55**, 149–175.
- Tennyson, J. & Yurchenko, S. N., 2012. ExoMol: molecular line lists for exoplanet and other atmospheres, *MNRAS*, **425**, 21–33.
- Tennyson, J. & Yurchenko, S. N., 2022. High Accuracy Molecular Line Lists for Studies of Exoplanets and Other Hot Atmospheres, *Frontiers Astron. Space Sci.*, **8**, 795040.
- Tennyson, J., Miller, S., & Le Sueur, C. R., 1993. TRIATOM: a suite for basis set calculations of ro-vibrational spectra of triatomic molecules, *Comput. Phys. Commun.*, **75**, 339–364.
- Tennyson, J., Hill, C., & Yurchenko, S. N., 2013. Data structures for ExoMol: Molecular line lists for exoplanet and other atmospheres, in *6th international conference on atomic and molecular data and their applications ICAMDATA-2012*, vol. 1545 of **AIP Conference Proceedings**, pp. 186–195, AIP, New York.
- Tennyson, J., Hulme, K., Naim, O. K., & Yurchenko, S. N., 2016a. Radiative lifetimes and cooling functions for astrophysically important molecules, *J. Phys. B: At. Mol. Opt. Phys.*, **49**, 044002.
- Tennyson, J., Yurchenko, S. N., Al-Refaie, A. F., Barton, E. J., Chubb, K. L., Coles, P. A., Diamantopoulou, S., Gorman, M. N., Hill, C., Lam, A. Z., Lodi, L., McKemmish, L. K., Na, Y., Owens, A., Polyansky, O. L., Rivlin, T., Sousa-Silva, C., Underwood, D. S., Yachmenev, A., & Zak, E., 2016b. The ExoMol database: molecular line lists for exoplanet and other hot atmospheres, *J. Mol. Spectrosc.*, **327**, 73–94.
- Tennyson, J., Yurchenko, S. N., Al-Refaie, A. F., Clark, V. H. J., Chubb, K. L., Conway, E. K., Dewan, A., Gorman, M. N., Hill, C., Lynas-Gray, A. E., Mellor, T., McKemmish, L. K., Owens, A., Polyansky, O. L., Semenov, M., Somogyi, W., Tinetti, G., Upadhyay, A., Waldmann, I., Wang, Y., Wright, S., & Yurchenko, O. P., 2020. The 2020 release of the ExoMol database: Molecular line lists for exoplanet and other hot atmospheres, *J. Quant. Spectrosc. Radiat. Transf.*, **255**, 107228.
- Tennyson, J., Pezzella, M., Zhang, J., & Yurchenko, S. N., 2023. Data structures for photoadsorption within the ExoMol project, *RASTI*, **2**, 231–237.
- Thompson, P., Cox, D., & Hastings, J., 1987. Rietveld refinement of Debye–Scherrer synchrotron X-ray data from Al₂O₃, *Journal of Applied Crystallography*, **20**(2), 79–83.
- Tinetti, G., Encrenaz, T., & Coustenis, A., 2013. Spectroscopy of planetary atmospheres in our galaxy, *A&A Rev*, **21**, 1–65.
- Tóbiás, R., Furtenbacher, T., Császár, A. G., Naumenko, O. V., Tennyson, J., Flaud, J.-M., Kumard, P., & Poirier, B., 2018. Critical Evaluation of Measured Rotational-Vibrational Transitions of Four Sulphur Isotopologues of S¹⁶O₂, *J. Quant. Spectrosc. Radiat. Transf.*, **208**, 152–163.
- Underwood, D. S., Tennyson, J., Yurchenko, S. N., Huang, X., Schwenke, D. W., Lee, T. J., Clausen, S., & Fateev, A., 2016. ExoMol line lists XIV: A line list for hot SO₂, *MNRAS*, **459**, 3890–3899.
- van den Bekerom, D. & Pannier, E., 2021. A discrete integral transform for rapid spectral synthesis, *J. Quant. Spectrosc. Radiat. Transf.*, **261**, 107476.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., & SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nature Methods*, **17**, 261–272.
- Wang, R., Balcunaite, U., Chen, J., Yuan, C., Owens, A., & Tennyson, J., 2023. NASA polynomial representation of molecular specific heats, *J. Quant. Spectrosc. Radiat. Transf.*, **306**, 108617.
- Wang, Y., Ferland, G. J., Lykins, M., Porter, R., van Hoof, P. A., & Williams, R., 2014. Radiative cooling II: effects of density and metallicity, *Monthly Notices of the Royal Astronomical Society*, **440**(4), 3100–3112.
- Wcislo, P., Thibault, F., Stolarczyk, N., Jozwiak, H., Slowinski, M., Gancewski, M., Stankiewicz, K., Konefal, M., Kassi, S., Campargue, A., Tan, Y., Wang, J., Patkowski, K., Ciurylo, R., Lisak, D., Kochanov, R., Rothman, L. S., & Gordon, I. E., 2021. The first comprehensive dataset of beyond-voigt line-shape parameters from ab initio quantum scattering calculations for the hitran database: He-perturbed h-2 case study, *J. Quant. Spectrosc. Radiat. Transf.*, **260**, 107477.
- Western, C. M., 2017. PGOPHER: A program for simulating rotational, vibrational and electronic spectra, *J. Quant. Spectrosc. Radiat. Transf.*, **186**, 221–242.
- Wilzowski, J. S., Gordon, I. E., Kochanov, R. V., Hill, C., & Rothman, L. S., 2016. H-2, he, and co2 line-broadening coefficients, pressure shifts and temperature-dependence exponents for the hitran database. part 1: So₂, nh₃, hf, hcl, ocs and c₂h₂, *J. Quant. Spectrosc. Radiat. Transf.*, **168**, 193–206.
- Yurchenko, S. N., Tennyson, J., Bailey, J., Hollis, M. D. J., & Tinetti, G., 2014. Spectrum of hot methane in astronomical objects using a comprehensive computed line list, *Proc. Nat. Acad. Sci.*, **111**, 9379–9383.
- Yurchenko, S. N., Al-Refaie, A. F., & Tennyson, J., 2018a. ExoCross: A general program for generating spectra from molecular line lists, *A&A*, **614**, A131.
- Yurchenko, S. N., Williams, H., Leyland, P. C., Lodi, L., & Tennyson, J., 2018b. ExoMol line lists XXVIII: The rovibronic spectrum of AlH, *MNRAS*, **479**, 1401–1411.
- Yurchenko, S. N., Mellor, T. M., Freedman, R. S., & Tennyson, J., 2020. ExoMol molecular line lists XXXIX: Ro-vibrational molecular line list for CO₂, *MNRAS*, **496**, 5282–5291.
- Yurchenko, S. N., Szajna, W., Hakalla, R., Semenov, M., Sokolov, A., Tennyson, J., Pavlenko, Y., & Schmidt, M. R., 2024. ExoMol line lists – LIV. Empirical line lists for AlH and AlD and Emission spectroscopy of AlD in A¹Π (ν = 0, 1, 2), *MNRAS*, **527**, 9736–9756.
- Zaghloul, M. R., 2017. Algorithm 985: Simple, Efficient, and Relatively Accurate Approximation for the Evaluation of the Faddeyeva Function, *ACM Trans. Math. Softw.*, **44**, 22.