Documentation

Isabelle Sangha

May 2022

1 Overview

This document describes the code: `https://github.com/IzzyS1010/EmissivityEstimation.git` to estimate the emissivity of a planet using the molecular cross-section data from HITRAN.

2 Data

**Pre-downloaded data can be found and downloaded from : `https://drive.google.com/drive/folders/1NUDNvPA0yExsYPsw9d_ffHxZCHYZ5t4x?usp=sharing`

The data used in the code is molecular cross-section data downloaded from SPECTRA [1]. The data is downloaded as text files (.txt) with wavenumber in cm^{-1} and cross-section in cm^2/molecule. Each file is for a specific molecule at a specified temperature, pressure, wavenumber range and resolution, and half-width length for the Voigt profile. Pre-downloaded files are included in the folder linked above. The files are for cross-section data of H_2O, CO_2 and CH_4 at temperatures 288 K, 252 K, 240 K, 217K, and 1600K and pressures 1 atm, 0.5 atm, 1/e atm, and 1/e^2. The wavenumber range, resolution and half-width length are 500 - 11501 cm^{-1}, 500 and 0.01 cm^{-1} respectively. If the user would prefer to analyse other molecules or a different temperature and pressure combination than the ones already downloaded they can access and download the data via `https://spectra.iao.ru/`. Using the ‘Molecule’ tab the user can select the molecule, data source (HITRAN), simulation type (here choose absorption cross-section), the molecular isotope, and the parameters for the data. An example of the interface is shown in Figure 1.

3 Code

3.0.1 Constants

Here the constants used in calculations through the code are defined. The constants defined are the Gravitational constant (taken to be 9.8 m/s^2), Planck's constant, the speed of light, the Boltzmann constant, and the Stefan-Boltzmann constant.

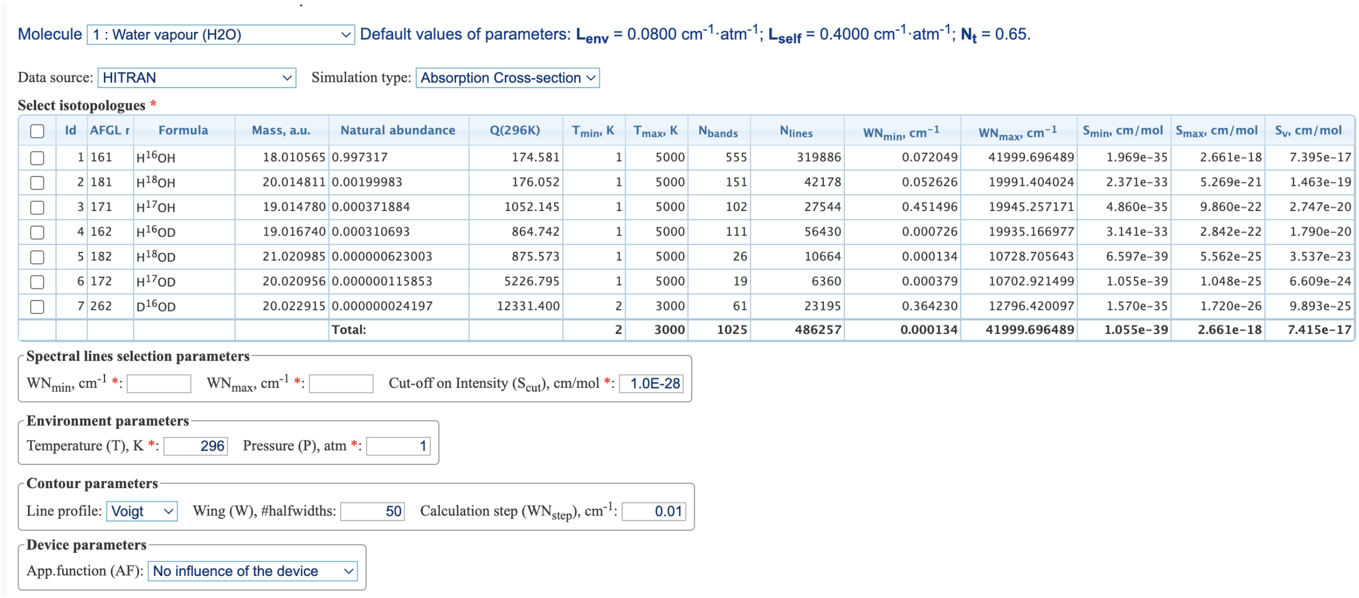
3.1 User Inputs

Here the user enters the temperature of the planet they want to estimate the emissivity of, the atmospheric pressure of the planet (in atm), and the concentrations of the longwave absorbing gases in the atmosphere. The concentration for gases that are evenly mixed (like CO_2 and CH_4) should be entered as volume mixing ratios. The concentration of water vapor should be entered as the precipitable water vapor in meters.

3.2 Reading data from files

Here the data from the folder is loaded and read. **Lines 41 and 44** define path and folder which the data is located. **Lines 47 - 49** define the gases whose data you want to access as strings for ease in labeling plots later on in the code. **Lines 54-55** take the temperature and pressure of the gases as strings (in order to format into the file name in the following lines). In **Lines 58 - 60** the name of the file for each gas at the desired temperature and pressure is created along with the directory (note that the files are named in a format : GAS_tK_patm.txt, were ‘GAS’ is the molecule, ‘t’ is the temperature, and ‘p’ is the pressure). If the user wants to load data other than the pre-downloaded files then they must ensure that they correctly call the name of the files and their location. **Lines 63 - 65** use the fopen command to define FID's for each file which are then used to read the data from the text files.

The first 6 lines of each text file contain information about the data (the temperature, pressure, resolution, wavenumber range, etc.). **Lines 78 - 80, 86-88, and 92-94** run through these lines so that later on fscanf is looking only at the data. If you want to check what data you are using you can print each line (1 - 6) of the text file to confirm it is the correct molecule, temperature, and pressure. **Lines 81, 89, and 95** use fscanf to read the data from the text file for each molecule. fscanf takes the FID and the



Molecule 1 : Water vapour (H2O) Default values of parameters: L_{env} = 0.0800 $cm^{-1}\cdot atm^{-1}$; L_{self} = 0.4000 $cm^{-1}\cdot atm^{-1}$; N_t = 0.65.

Data source: HITRAN Simulation type: Absorption Cross-section

Select isotopologues *

□	Id	AFGL r	Formula	Mass, a.u.	Natural abundance	Q(296K)	T_{min}, K	T_{max}, K	N_{bands}	N_{lines}	WN_{min}, cm^{-1}	WN_{max}, cm^{-1}	S_{min}, cm/mol	S_{max}, cm/mol	S_v, cm/mol
□	1	161	$H^{16}OH$	18.010565	0.997317	174.581	1	5000	555	319886	0.072049	41999.696489	1.969e-35	2.661e-18	7.395e-17
□	2	181	$H^{18}OH$	20.014811	0.00199983	176.052	1	5000	151	42178	0.052626	19991.404024	2.371e-33	5.269e-21	1.463e-19
□	3	171	$H^{17}OH$	19.014780	0.000371884	1052.145	1	5000	102	27544	0.451496	19945.257171	4.860e-35	9.860e-22	2.747e-20
□	4	162	$H^{16}OD$	19.016740	0.000310693	864.742	1	5000	111	56430	0.000726	19935.166977	3.141e-33	2.842e-22	1.790e-20
□	5	182	$H^{18}OD$	21.020985	0.000000623003	875.573	1	5000	26	10664	0.000134	10728.705643	6.597e-39	5.562e-25	3.537e-23
□	6	172	$H^{17}OD$	20.020956	0.000000115853	5226.795	1	5000	19	6360	0.000379	10702.921499	1.055e-39	1.048e-25	6.609e-24
□	7	262	$D^{16}OD$	20.022915	0.000000024197	12331.400	2	3000	61	23195	0.364230	12796.420097	1.570e-35	1.720e-26	9.893e-25
					Total:		**2**	**3000**	**1025**	**486257**	**0.000134**	**41999.696489**	**1.055e-39**	**2.661e-18**	**7.415e-17**

Spectral lines selection parameters

WN_{min}, cm^{-1} *: WN_{max}, cm^{-1} *: Cut-off on Intensity (S_{cut}), cm/mol *: 1.0E-28

Environment parameters

Temperature (T), K *: 296 Pressure (P), atm *: 1

Contour parameters

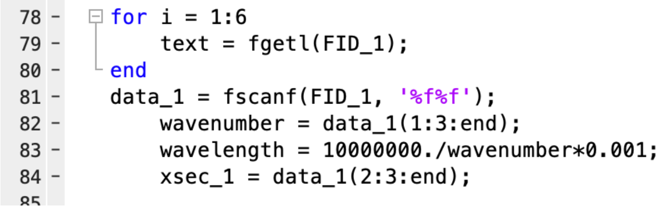
Line profile: Voigt Wing (W), #halfwidths: 50 Calculation step (WN_{step}), cm^{-1}: 0.01

Device parameters

App.function (AF): No influence of the device

Figure 1: Example of SPECTRA simulation from [1]

formatspec (format of the data) as inputs. The data is read and saved as a vector in such a way that the values alternate through wavenumber, cross-section value, and absorption. To pull out the wavenumber data (which is the same for every file if using the same range and resolution) and convert to wavelength, **Lines 82 - 83** take every third value of the vector with all the data and use the conversion that wavelength = 1/wavenumber to get a vector of wavelengths in micrometers. A similar process is done in **Lines 84, 90, and 96** to pull the cross-section data for each molecule as its own vector. An example of this code is shown in Figure 2.



```
78 -     for i = 1:6
79 -         text = fgetl(FID_1);
80 -     end
81 -     data_1 = fscanf(FID_1, '%f%f');
82 -         wavenumber = data_1(1:3:end);
83 -         wavelength = 10000000./wavenumber*0.001;
84 -         xsec_1 = data_1(2:3:end);
85
```

Figure 2: Section of code that runs through pulling the data from vector

3.3 Calculating absorption coefficient, optical depth, and transmissivity

With the cross-section data loaded and organized we can now calculate the absorption coefficient, optical depth and transmissivity of each gas and the atmosphere (accounting for all the absorbing gases). To get the absorption coefficient from the cross-section data we must use the molecular weight of each gas. First enter the molar mass of each gas in g/mol in **Lines 104 - 106**. These values are then used in **Lines 109 - 111** to calculate the molecular weight in kg/mol using Avogadro's number. The absorption coefficient can be found by dividing the cross-section data by the molecular weight (accounting for the fact that the cross-section data is in cm^2/molecule) and is done in **Lines 114 - 116**

To calculate the optical depth, transmittance, and absorbance of the atmosphere we have to weight each gas's contribution according to the concentration of the gas in the atmosphere. These 'weighting coefficients' are calculated in **Lines 124 - 126** (note the slightly different calculation for H_2O due to the use of precipitable water vapor instead of concentration).

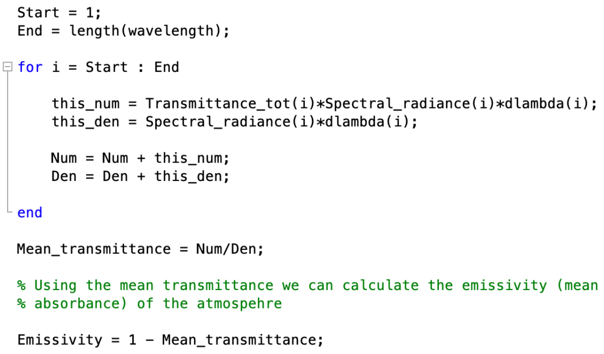
Lines 132 - 134, 137 - 139, and 142 - 144 calculate the optical depth, transmittance, and absorbance of each gas using the equations discussed in the paper . **Lines 151 - 154** use the same equations as for the individual gases, but the optical depth is the sum of each individual gas's optical depth.

3.4 Calculating Earth's emission spectrum

Here (**Line 179**) we calculate Earth's emission spectrum using Planck's law and the wavelength vector created earlier. Note that in the expression the wavelength vector is multiplied by 10^6 in order to get wavelength into meters from micrometers.

3.5 Calculating mean absorbance (emissivity)

The average absorbance of the atmosphere over all longwave wavelengths (emissivity) is done by calculating a weighted mean of the transmittance and then subtracting 1 by this value. Calculating the integral like a Riemann sum, we need dλ (incremental wavelength sections) for each transmittance value. Because of the relationship between wavenumber and wavelength, evenly spaced intervals in wavenumbers do not translate to evenly spaced intervals in wavelength. To determine the wavelength intervals you must square the wavelength vector (this is determined from taking the differential of the relationship between wavelength and wavenumber). This is done in **Line 183** where 'dlambda' is a vector that contains the width of each wavelength rectangle to be used in the Riemann sum. **Lines 207 - 215** run through a for loop that iterates through each wavelength in the range being analysed and determines the contribution of the transmittance at the given wavelength by taking the transmittance value and multiplying it by Earth's emission at that wavelength and by the width of the wavelength interval at that wavelength (dλ). These values are progressively summed through the for loop so that by the end 'Num' is the sum of the atmosphere's transmittance weighted by the planet's emission spectrum. To get the weighted average of transmittance this value is divided by the sum of the planet's emission spectrum over the same wavelength (the value 'Den' from the for loop). This is done in **Line 217**. Finally, in **Line 222** this value is subtracted from 1 to get the atmospheric emissivity. This code is shown in Figure 3

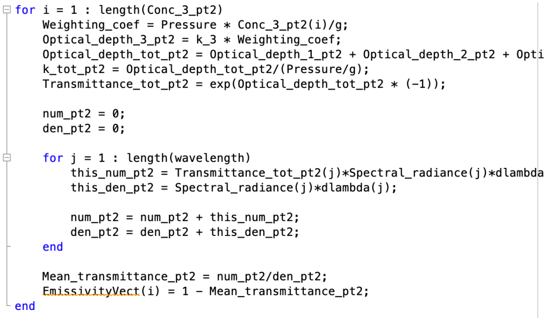


```
Start = 1;
End = length(wavelength);

for i = Start : End

    this_num = Transmittance_tot(i)*Spectral_radiance(i)*dlambda(i);
    this_den = Spectral_radiance(i)*dlambda(i);

    Num = Num + this_num;
    Den = Den + this_den;

end

Mean_transmittance = Num/Den;

% Using the mean transmittance we can calculate the emissivity (mean
% absorbance) of the atmospehre

Emissivity = 1 - Mean_transmittance;
```

Figure 3: Section of code that runs through the for loop to sum and average the absorbance

3.6 Emissivity as function of molecule concentration

Here we determine how emissivity changes as the concentration of one of the absorbing molecules changes. **Lines 263 - 272** run through the same process for determining the gas's optical depths as earlier in the code, but this time only for the two gases whose concentration is not changing through this simulation. The user can determine what concentration they want for the gases that are not changing on **Lines 263 and 264**. In **Lines 278 to 280** the user can input the range of concentration they want the gas that is changing to run through and define the interval (the step between concentrations) they want. **Line 298** then creates a vector that contains the concentrations ranging from the low bound to the high bound with steps equal to the defined interval. **Line 285** creates vector initialized to zeros (called 'EmissivityVect') of the same length as the concentration vector. **Lines 290 - 310** runs through a nested for loop. The first for loop runs through the concentrations of the gas (from the vector just defined). Each iteration calculates a new weighting coefficient and optical depth for the gas that is changing (**Line 291 - 292**) and from this calculates a new atmospheric optical depth spectrum, absorption coefficient spectrum, and atmospheric transmittance spectrum (**Lines 293 - 295**). The second loop (starting on **Line 200**) is identical to the for loop used in section 3.5 to calculate mean atmospheric transmittivity this time taking the transmittance value determined in the first loop. Each emissivity value for a given concentration of gas (calculated in the second for loop) is saved in the vector 'EmissivityVect' at the same index as corresponding concentration used to calculate it. The end result of the nested for loop is a vector of emissivity values that relates to the vector of gas concentration values. An example of the nested for loop is shown in Figure 4. These two can then be plotted to visualize the relationship.



```
for i = 1 : length(Conc_3_pt2)
    Weighting_coef = Pressure * Conc_3_pt2(i)/g;
    Optical_depth_3_pt2 = k_3 * Weighting_coef;
    Optical_depth_tot_pt2 = Optical_depth_1_pt2 + Optical_depth_2_pt2 + Optical_depth_3_pt2;
    k_tot_pt2 = Optical_depth_tot_pt2/(Pressure/g);
    Transmittance_tot_pt2 = exp(Optical_depth_tot_pt2 * (-1));

    num_pt2 = 0;
    den_pt2 = 0;

    for j = 1 : length(wavelength)
        this_num_pt2 = Transmittance_tot_pt2(j)*Spectral_radiance(j)*dlambda(j);
        this_den_pt2 = Spectral_radiance(j)*dlambda(j);

        num_pt2 = num_pt2 + this_num_pt2;
        den_pt2 = den_pt2 + this_den_pt2;
    end

    Mean_transmittance_pt2 = num_pt2/den_pt2;
    EmissivityVect(i) = 1 - Mean_transmittance_pt2;
end
```

Figure 4: Section of code that runs through the nested for loops to get vector of emissivity values

3.7 Figures

Figure 1: Plot of absorption cross-sections (Lines 156 - 175)

This figure plots the absorption cross-sections of each gas in a tiled format so all three spectra are visible. The number of tiles (gases) is determined in **Line 158** and can be adjusted from 3 to any other number depending on how many gases the user is modeling. An example of the figure produced is seen in Figure 5.

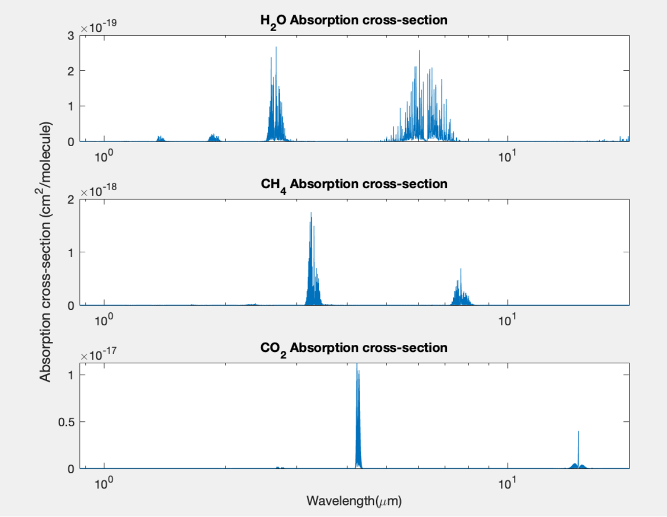


Figure 5: Example of figure 1

Figure 2: Plot of planetary Planck function and extinction coefficient (Lines 224 - 252)

This figure is similar to 1, but plots the extinction coefficient of each gas along with the normalized and re-scaled Planck function. An example of the figure produced can be seen in Figure 6.

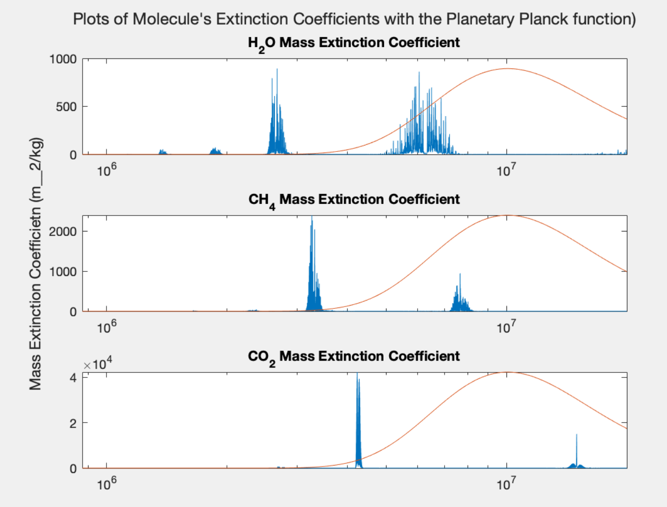


Figure 6: Example of figure 2

Figure 3: Plot of emissivity as a function of gas concentration (Lines 312 - 323)

This plots the last section of code that determines emissivity as a function of a gases concentration. It takes the concentration vector created in a earlier section as the x axis and the emissivity vector as the y input. There are lines of code that input vertical lines at pre-industrial and double pre-industrial values for CO_2 concentration, but these can be removed if not plotting against CO_2 concentration. An example of the figure produced can be seen in Figure 7.

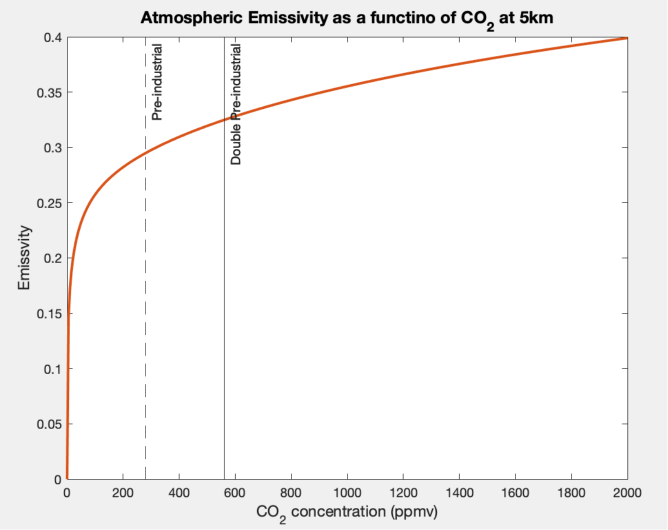


Figure 7: Example of figure 3

Figure 4: Plot of emissivity as a function of gas concentration on a loglog scale (Lines 324 - 334)

This figure plots the same data as Figure 3, just on a loglog scale (both y and x axis are log scale). An example of the figure produced can be seen in Figure 8.

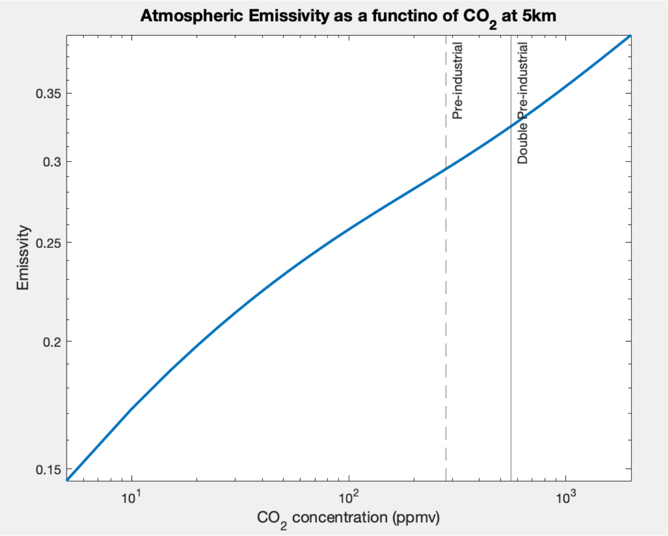


Figure 8: Example of figure 4

References

[1] C. Mikhailenko, Y. Babikov, and V. Golovko. "Information-calculating system Spectroscopy of Atmospheric Gases. The structure and main functions". In: *Atmospheric and Oceanic Optics* 18 (2005), pp. 685–695.