Welcome to the Atmociad documentation!

Thank you for your interest in using the Atmociad database. The following guide will help you get started in using our database for your project.

The documentation is structured as follows:

- Quick Start Guide
- Function List
- Reference for Attribution
- Contact Us

We hope that our database is useful for your particular scientific project.

Quick Start Guide

This Quick Start document will get you started on understanding the database and plotting basic plots of each cross section contained in the database.

Data Structure Basics

The data contained in the Atmociad database is in an easily readable XML format. Contained in each XML data tree are all of the important information about the cross sections of each species including:

- Process type and energy threshold
- Reference for the cross section data
- Estimated or measured uncertainties

All of this information can easily be extracted from the xml file using many common programming languages, such as C, C++, Java, and Python. Additionally, all of our basic plotting functions described below automatically extract the data from the xml files.

If you would like more information about the specifics of the data contained in the xml files, refer to the Atmociad_detail.pdf document contained in the help directory.

Creating the First Plot

Contained in this package are Python functions that can help you plot the cross section values contained in the xml files quickly. To create a plot, simply run the following command in Python:

```
python <path to plotting function> <path to xml file>
```

Example

We ran the following code:

```
python angread.py $ROOT/electron/H2/RecommendedH2.xml
```

where $ROOT is the path to the directory where the database is located. The resulting plot is below.

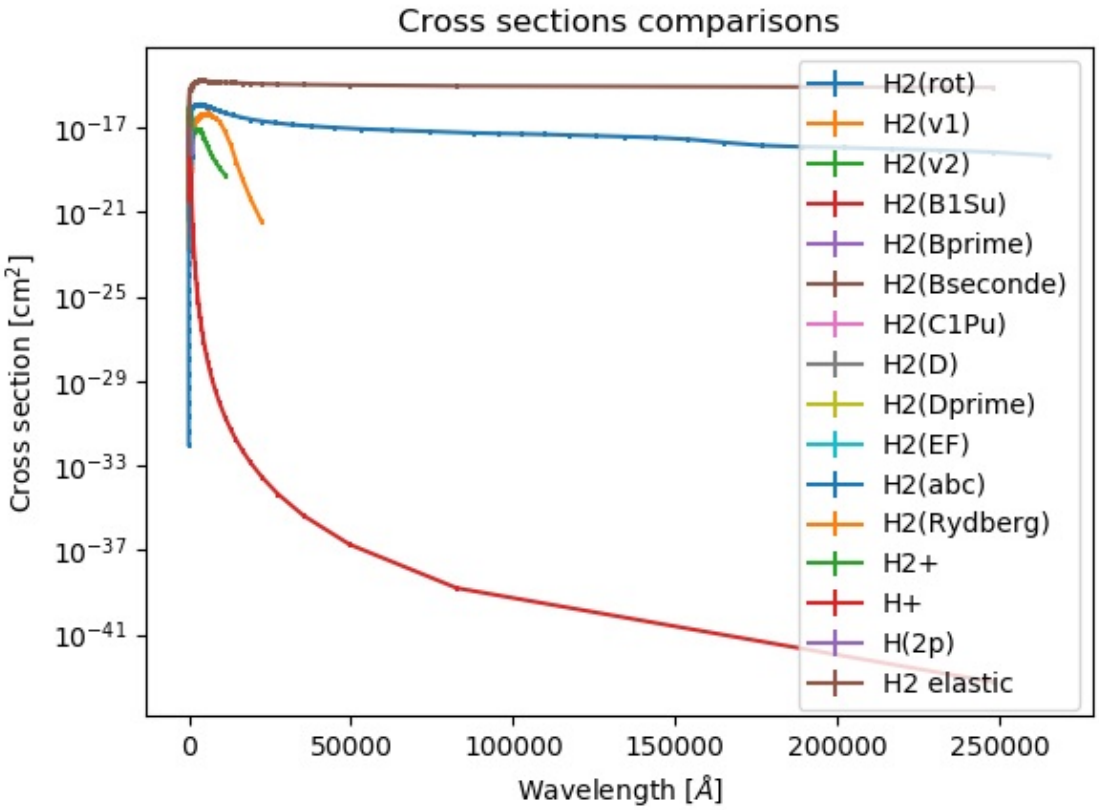

Figure 1: *Cross section plot example created by running the angread.py code.*

X-axis: Wavelength in Angstroms or Energy is electron volts (*eV*)
Y-axis: Cross section (*cm^2*)

All cross sections found in the xml data file are automatically plotted.

Summary of Plot Options Available

Extrapolation: Extrapolate beyond measured values using a power law function down to a minimum threshold set in the plot code.

Run either of the following in Python:

```
python energyextrapole.py <name of xml file>
#energy
```

or:

```
python angextrapole.py <name of xml file>
#wavelength
```

Plotting as function of energy or wavelength:

Options exist to plot the cross sections as either a function of photon energy (in electron volts), or as a function of wavelength (in Angstroms). The option is controlled by the name of the .py file used.

ang* in the file name: Plot as a function of wavelength
energy* in the file name: Plot as a function of energy

Function List

This is a complete list of plotting functions contained in the database. Links to each entry are in the table of contents or below.

angread

Creates plot of different cross-sections contained in xml file. Plotted as a function of wavelength in Angstroms.:

```
python angread.py filename
```

Arguments

filename : *string*

- The file location of the xml file that contains the cross-sections

Returns

None

Notes

If an uncertainty estimate is included in the xml files, then error bars are placed on the line plots. See comments in code for details about each function contained within. You can see the help for each function by running the following on the command line:

```
pydoc ./angread.py
```

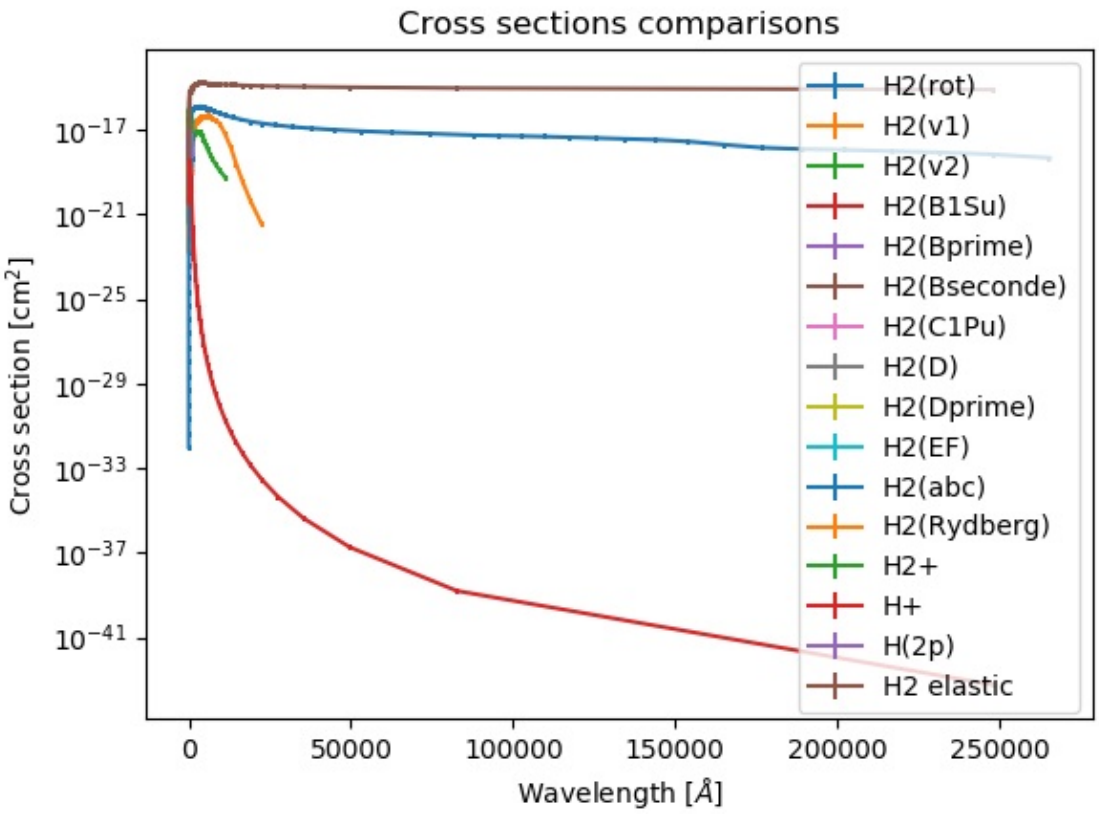

Figure 2. Output plot that is created when running angread.py with the xml file RecommendedH2.xml [in the electron subdirectory].

angextrapole

Creates plot of different cross-sections contained in xml file. Plotted as a function of wavelength in Angstroms.

The extrapolation of each cross-section uses a power law function to create the extrapolation. See the documentation for powerlaw.py for details on the extrapolation function. The extrapolation is done until the cross section values fall below a threshold value that is defined in the xml file.:

```
python angextrapole.py filename
```

Arguments

filename : *string*

- The file location of the xml file that contains the cross-sections

Returns

None

Notes

If an uncertainty estimate is included in the xml files, then error bars are placed on the line plots. See comments in code for details about each function contained within. You can see the help for each function by running the following on the command line:

```
pydoc ./angreadextrapole.py
```

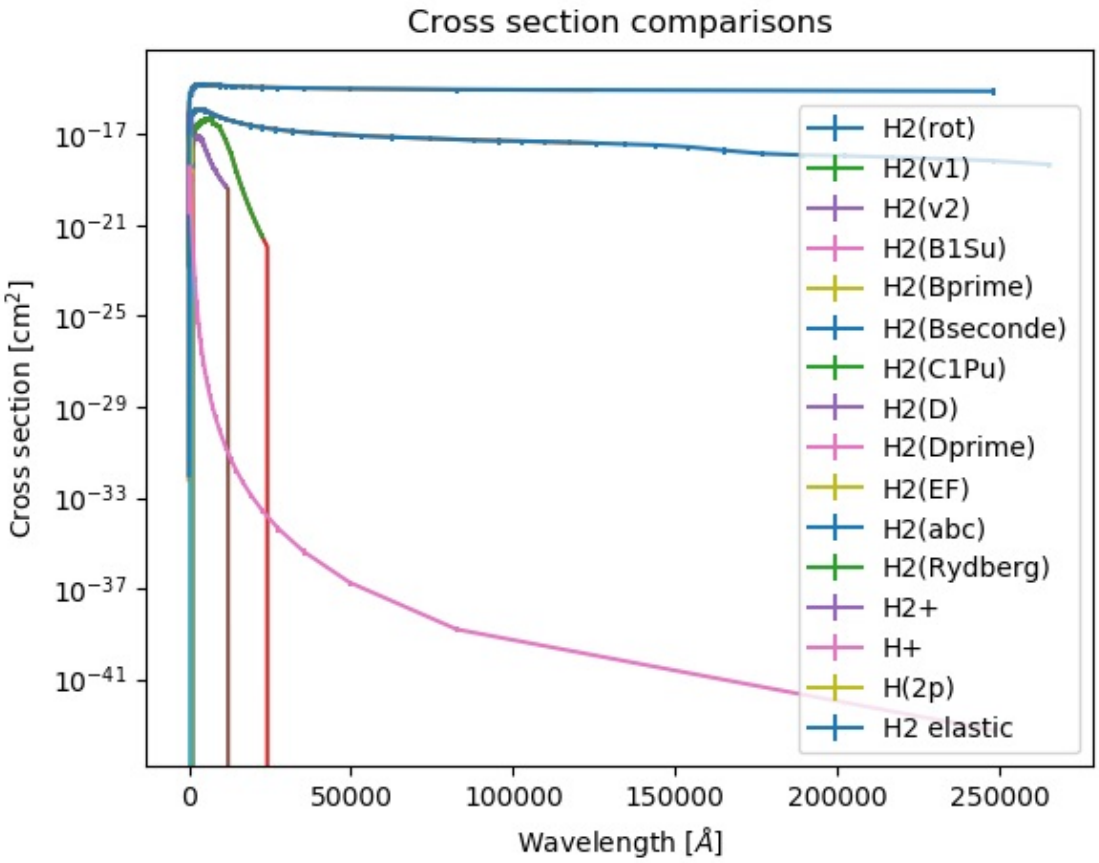

Figure 3. Output plot that is created when running angextrapole.py with the xml file RecommendedH2.xml [in the electron subdirectory].

energyread

Creates plot of different cross-sections contained in xml file. Unlike angread.py, plots cross sections as a function of energy (instead of wavelength).:

```
python energyread.py filename
```

Arguments

filename : *string*

- The file location of the xml file that contains the cross-sections

Returns

None

Notes

If an uncertainty estimate is included in the xml files, then error bars are placed on the line plots. See comments in code for details about each function contained within. You can see the help for each function by running the following on the command line:

```
pydoc ./energyread.py
```

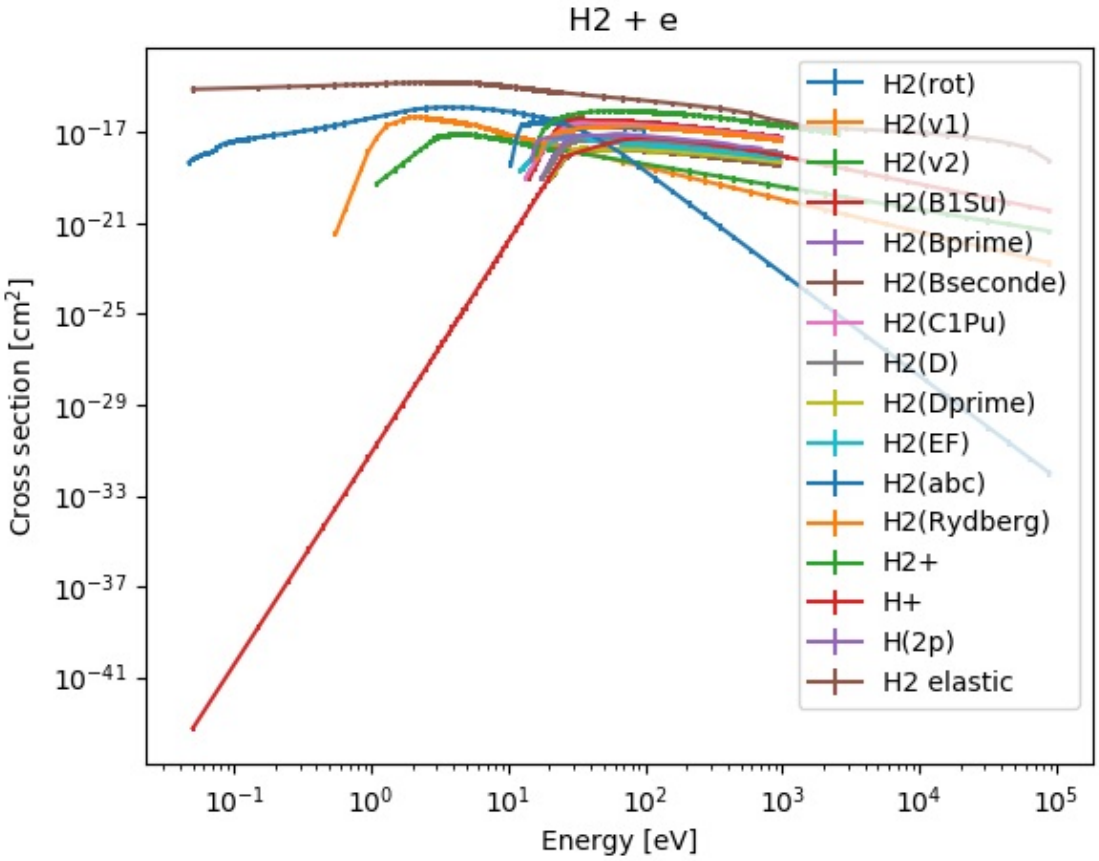

Figure 4. Output plot that is created when running energyread.py with the xml file RecommendedH2.xml [in the electron subdirectory].

energyextrapole

Creates plot of different cross-sections contained in xml file. Unlike angextrapole.py, plots cross sections as a function of energy (instead of wavelength). Extrapolation done using power law extrapolation as a function of energies.:

```
python energyextrapole.py filename
```

Arguments:

filename : *string*

- The file location of the xml file that contains the cross-sections

Returns:

None

Notes:

If an uncertainty estimate is included in the xml files, then error bars are placed on the line plots. See comments in code for details about each function contained within. You can see the help for each function by running the following on the command line:

```
pydoc ./energyextrapol.py
```

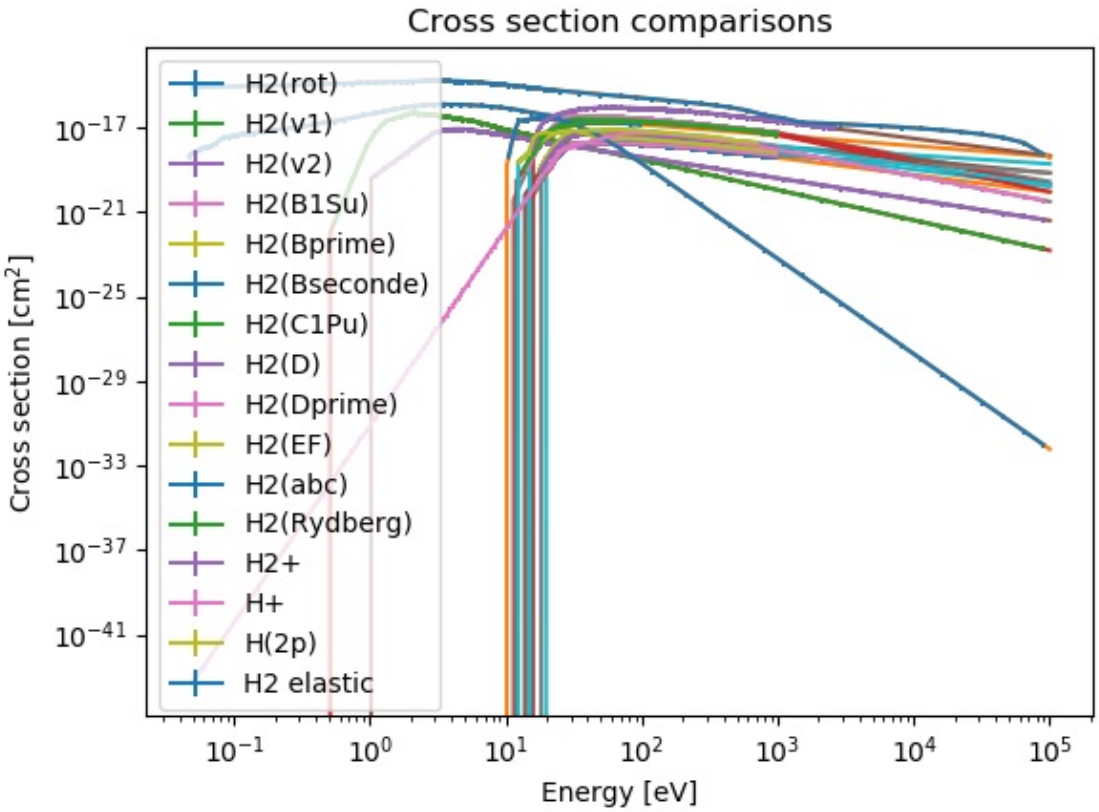

Figure 5. Output plot that is created when running energyextrapole.py with the xml file RecommendedH2.xml [in the electron subdirectory].

Reference for Attribution

For the database source, please cite:

Contact Us

Guillaume Gronoff - guillaume.p.gronoff@nasa.gov Brad Hegyi - bradley.m.hegyi@nasa.gov