Code to Optimize Gaussian Filter for Photodissociation Cross Sections

I. Background:

A given photodissociation cross section, σ, straight out of ExoCross will have a structure such as that in fig. 1.

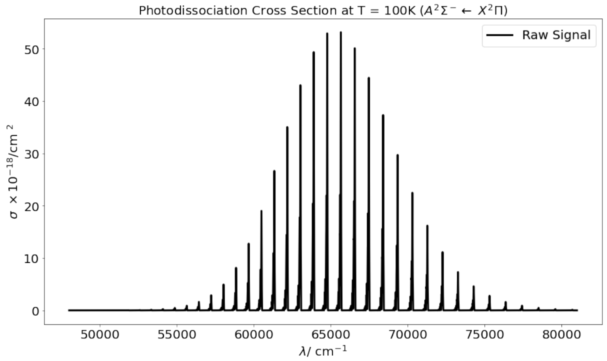

Fig. 1: Plot of photodissociation cross section, σ for $1^2\Sigma^- \leftarrow\ \mathrm{X}^2\Pi$ from ExoCross

This code creates a smooth, continuous cross section by applying a gaussian filter according to the method from Pezzella, Yurchenko, and Tennyson [1]. This is done by first computing the a photodissocation cross section with a Gaussian line profile and increasing the Gaussian half width at half maximum (α_D) to an appropriate level to return a continuous curve. The method detailed in Pezzella, Yurchenko, and Tennyson [1] works by further smoothing a cross section with Gaussian line profile extracted from ExoCross. The same has been done here. This new continuous curve should return an unchanged value for the photodissociation rate as the Gaussian line profile (in this case, Gaussian filter) is normalised to unity upon integration.

It was found in [1], that for Hydrogen Halides an $\alpha_D \in [2000, 3000]$ cm $^{-1}$is appropriate and in the case of high temperatures, $\alpha_D \approx 75$ cm $^{-1}$is appropriate for BeH$^+$. Indeed as temperature increases, the required smoothing would decrease as more excited states are populated (Note - I am unsure of the Physics here, but I think this is accurate).

The goal of this code is to find an $\alpha_D$such that the cross section is a smooth function and no more than that. Increasing alpha to an arbitrarily large value produces one of two problems, either the peak intensity is significantly decreased and the cross section is flattened out, or the output is garbage.

II. How it works:

In order to use this code, a requirement is to have a *.xsec formatted file out from ExoCross. This is produced with a Gaussian line profile with α_D= 10 cm $^{-1}$. This is used as the input for this code. One can, of course fine an optimal $\alpha_D$by simply adjusting the HWHM value in ExoCross, however then this requires more a more trial and error approach which can be frustrating.

The code works on the premise that for a cross section such as in fig. 1, we can see that there is one peak at $\sim 65,000$ cm $^{-1}$. Therefore, for a minimally smoothed cross section, the requirement is that there is only one turning point. Furthermore, I required that the function be continuous to its second derivate, as the first requirement is not always sufficient to ensure a smooth curve. The steps are as follows:

- Compute smoothed cross section for small HWHM ($\alpha_D \approx 100$ cm $^{-1}$).
- Differentiate and count the number of roots
- If > 1, increase α_D, else continue
- Differentiate again and count the number of roots
- If > 2, increase α_D and repeat steps 2-4, else break.
- Return last used α_D

.

The code is run by calling the function **AlphaFinder** and has the following arguments:

- **Alpha_Range:** range of Alphas to test - dtype = np.array
- **Lambda:** lambda values for input signal - dtype = np.array
- **Sigma:** sigma values for input signal - dtype = np.array
- **Num_Turn_Points:** Number of turning points in expected cross section, found by visual inspection - dtype = Float
- **sampling_frequency:** sampling frequency as a percentage (of input signal length [between 0 - 1]) - dtype = list (len = 2)
- **scalefactor_:** Linear multiplicative scale factor for resulting curve, default is 18 - dtype = float
- **Thresh_:** Threshold value for derivatives; if root located in region where the cross section < Thresh, roots in the derivatives are not counted.

The arguments **sampling_frequency:** and **Thresh_:** are further explained in section III

III. Finding Roots:

For a cross section σ, we can find $d_\lambda \sigma$ numerically. A pair of points are connected and the gradient is calculated between this gradient pair with the following equation:

$$d_x f(i) = \frac{f_{i+s} - f_i}{|x_{i+s} - x_i|} \tag{1}$$

Where f is the function to differentiate, i is the i'th term, s is the gradient pair distance (in terms of number of points), indicated by the **sampling_frequency** argument. For example for $i = 1, s = 2$ for $f = [1, 4, 9, 16, 25]$, $x = [1, 2, 3, 4, 5]$, then:

$$d_x f(2) = \frac{16 - 4}{4 - 2} = 6$$

If the step in x is constant then the denominator becomes $\Delta x = x_s - x_0$. The length of the array produced:

$$\textbf{len}(d_x f) = \textbf{len}(f) - s$$

So $s \gg 1$ is not suitable.

Furthermore, in the case of areas of flatness where noise is the predominant contributor to signal, a threshold parameter **Thresh_** is introduced. In regions where $d_x f = 0$ but $f <$ **Thresh_**, a root is not counted.

A. *Sampling_Frequency*, s

The motivation for the inclusion of the s term is due to noise, especially in the second derivative.

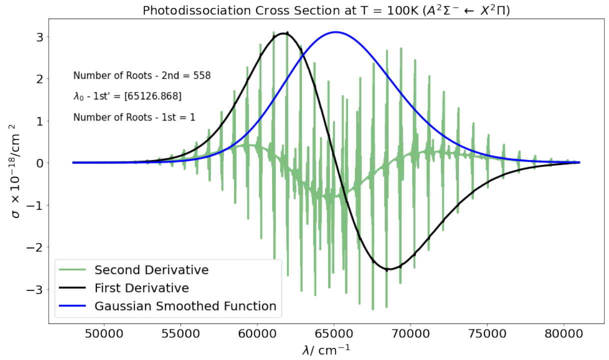

Fig. 2: Photodissociation Cross section of OH for $1^2\Sigma^- \leftarrow \ \mathrm{X}^2\Pi$; Gaussian line profile smoothed with $\alpha_D = 700$ and first and second differentials with $s_{1,2} = 1$ and **Thresh_**$_{1,2} = 0$.

In fig. **??** the first derivative meets the condition of a single root in the region where the cross section is at its maximum. There is, however, noise in the first derivative and subsequently, > 550 roots in the second derivative. We could further increase α_D, however this doesn't solve the problem, especially since there is a preliminary 10cm $^{-1}$Gaussian line profile applied to the input from ExoCross. Increasing $\alpha_D$further risks producing a garbled signal, see fig. 3.

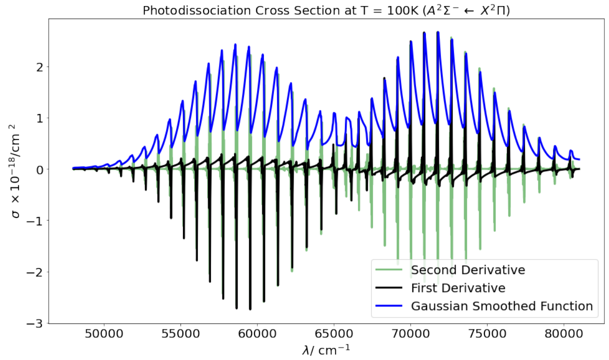

Fig. 3: Photodissociation Cross section of OH for $1^2\Sigma^- \leftarrow \ \mathrm{X}^2\Pi$; Gaussian line profile smoothed with $\alpha_D = 2000$ and first and second differentials with $s_{1,2} = 1$ and **Thresh_**$_{1,2} = 0$.

In fig. 2 we can see, however, that there appears to be an overall structure in the second derivative which appears to have two roots as required. We can remove some of the noise by ignoring some information from both signals by setting $s = P \times \mathbf{len}(f)$. Where P is some percentage of the length of the Gaussian array, increasing the distance between gradient pairs. P is queried as the **sampling_frequency** argument in the **AlphaFinder** function. From testing, a nominal value of $P = 1\%$ works for OH for the $1^2\Sigma^- \leftarrow \ \mathrm{X}^2\Pi$ and $1^2\Delta \leftarrow \ \mathrm{X}^2\Pi$ transitions. An example is shown in fig. 4.

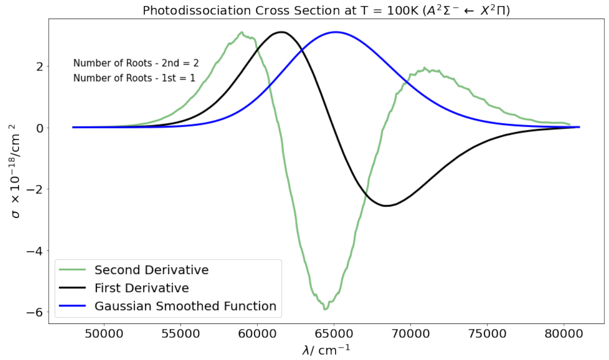

Fig. 4: Photodissociation Cross section of OH for $1^2\Sigma^- \leftarrow \ \mathrm{X}^2\Pi$; Gaussian line profile smoothed with $\alpha_D = 700$ and first and second differentials with $s_{1,2} = 2400$ and $_{1,2} = 0$**.**

B. *Thresh_*:

The **Thresh_** argument has also been included in this code due to noise and to avoid over smoothing. In the wing regions of the cross sections, where the curve is close to zero, bumps can arise causing a false turning points to be found. **Thresh_** seeks to remove this by setting a lower limit on cross section intensity where if $f <$ **Thresh_**, then a found turning point will not be found.

IV. Summary:

Fig. 5 explains how **AlphaFinder** takes input data and outputs an optimized α_D for a given ExoCross *.xsec input file.

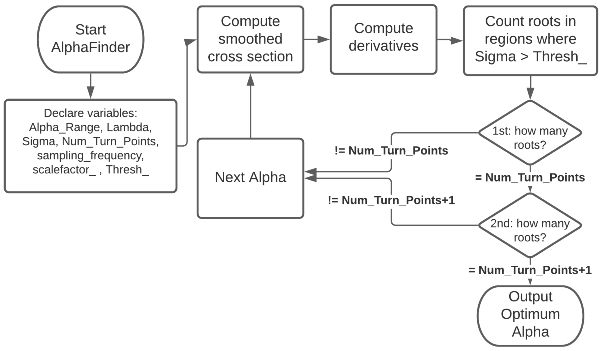
Start AlphaFinder
Declare variables: Alpha_Range, Lambda, Sigma, Num_Turn_Points, sampling_frequency, scalefactor_ , Thresh_
Compute smoothed cross section
Compute derivatives
Count roots in regions where Sigma > Thresh_
1st: how many roots?
!= Num_Turn_Points
Next Alpha
= Num_Turn_Points
!= Num_Turn_Points+1
2nd: how many roots?
= Num_Turn_Points+1
Output Optimum Alpha

Fig. 5: Flowchart explaining how **AlphaFinder** processes the input signal.

References

[1] Marco Pezzella, Sergei N. Yurchenko, and Jonathan Tennyson. "A method for calculating temperature-dependent photodissociation cross sections and rates". In: *Physical Chemistry Chemical Physics* 23.30 (July 2021), pp. 16390–16400. DOI: 10.1039/d1cp02162a.