NIRISS SOSS extraction algorithm

Antoine Darveau-Bernier & David Lafrenière

September 30, 2020

1 Concept

The idea is to create a model of the signal received in each pixel. To do so, we need a good model of the spatial point spread function and the throughput (spectral transmission) for each order. The wavelength solution for each order also needs to be given. The flux is not modeled here. Then the model of the detector and the observation are compared using the log L. By maximising it with respect to the flux, we obtain a system of linear equations which can be solved for the flux by inverting a matrix.

2 Nomenclature and assumptions

- *Detector geometry* Each pixel that will be used for the fit is labeled by the index i. We don't need the 2 dimensional information of the detector.
- *Diffraction orders* The order of diffraction are labeled by the index n.
- *Wavelength solution* Let's assume that an exact wavelength solution is known for every orders, such that at a pixel i the central wavelength of order n is λ_{ni}. Let the wavelength dispersion per pixel i be $\Delta\lambda_{ni}$ and λ^+_{ni}, λ^-_{ni}, the wavelengths at the borders of the pixel.
- *Spectral transmission* The throughput for each order must be given and is noted $T_n(\lambda)$. It only depends on the wavelength and not the pixel.
- *Spatial transmission* Let's assume that the point spread function is exactly known and described by P_{ni} for order n.
- *Target spectrum* Let the spectral flux density of the target incident upon the spectrograph be $f(\lambda)$. Let's define a grid where to project f labeled by the index k so that $f(\lambda_k) = f_k$ and $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$. Now, the flux density is seen by each order at a different resolution, so for an order n, we have $\tilde{f}_{n\tilde{k}} = \sum_k c_{n\tilde{k}k} f_k$ where $\tilde{f}$ is the convoluted flux and the $c_{n\tilde{k}k}$ are the coefficient of the convolution kernel. Finally, let's define N_k, the length of the discretized flux f, so that $1 \leq k \leq N_k$. Similarly, $N_{\tilde{k}}$ is the length of the convolved flux $\tilde{f}$, so that $1 \leq \tilde{k} \leq N_{\tilde{k}}$.

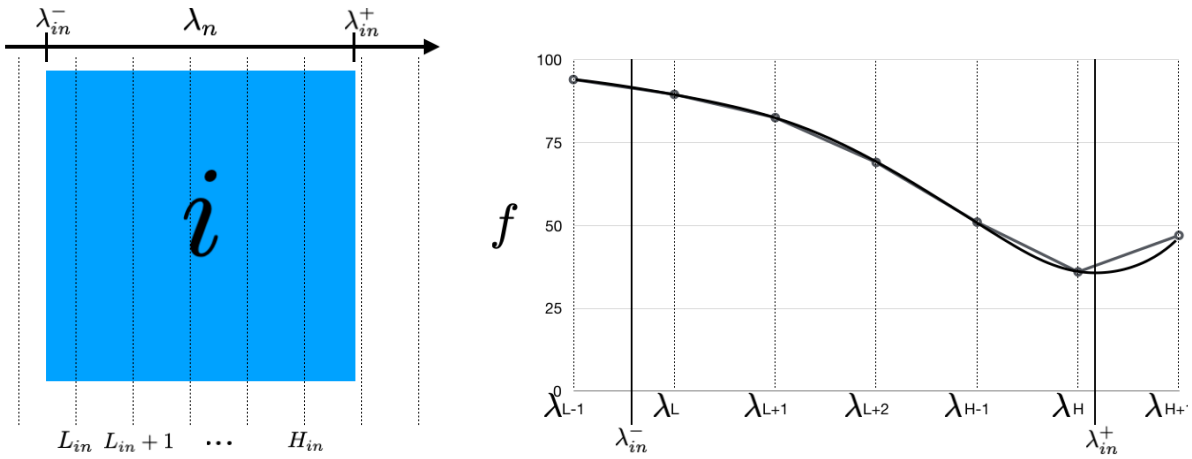

Figure 1: Model of a pixel i.

3 The model detector image

The number of photo-electrons detected by pixel i can be modelled, up to a multiplicative constant, as,

$$M_i = \sum_n \int_{\lambda_{ni}^-}^{\lambda_{ni}^+} P_{ni} T_n(\lambda) \tilde{f}_n(\lambda) \lambda d\lambda = \sum_n \int_{\lambda_{ni}^-}^{\lambda_{ni}^+} a_{ni}(\lambda) \tilde{f}_n(\lambda) d\lambda. \tag{1}$$

Where $a_{ni}(\lambda)$ accounts for all the coefficients that are not the flux. To use it in an algorithm, we need a numerical form of this integral. There are multiple ways to do so, but in this case, we will use the trapeze method on a specified grid. The details of this method are in the section ?. Independently of the chosen integration technique, the numerical form of the integral will look like

$$M_i = \sum_n \sum_{\tilde{k}} w_{in\tilde{k}} a_{in\tilde{k}} \tilde{f}_{n\tilde{k}}, \tag{2}$$

with the $w_{in\tilde{k}}$ given by the integration method. Now $\tilde{f}_n(\lambda)$ is different for each order. To link the diffraction orders, we want to write these equation according to the underlying flux $f(\lambda)$. In the numerical form,

$$\tilde{f}_{n\tilde{k}} = \sum_k c_{n\tilde{k}k} f_k \tag{3}$$

with $c_{n\tilde{k}k}$ being the coefficients of the convolution kernel at order n.

Finally, we have that

$$M_i = \sum_n \sum_{\tilde{k}} w_{in\tilde{k}}\, a_{in\tilde{k}} \sum_k c_{n\tilde{k}k} f_k \,. \tag{4}$$

This equation can be written in a more intuitive matrix form as

$$\Big(M\Big)_{N_i} = \sum_n \Big(w_n a_n\Big)_{N_i \times N_{\tilde{k}}} \Big(c_n\Big)_{N_{\tilde{k}} \times N_k} \Big(f\Big)_{N_k} . \tag{5}$$

To simplify the notation again, we can put all the coefficient in a single matrix $\mathbf{B}$ so that

$$\Big(M\Big)_{N_i} = \sum_n \Big(B_n\Big)_{N_i \times N_k} \Big(f\Big)_{N_k} = \Big(B\Big)_{N_i \times N_k} \Big(f\Big)_{N_k} . \tag{6}$$

SO finally, we have a model of each valid pixel that is given by

$$\boxed{\mathbf{M}_{N_i} = \mathbf{B}_{N_i \times N_k} \mathbf{f}_{N_k}}. \tag{7}$$

4 Solving for f

Now that we have a model of the intensity at each pixel, we can link the measured intensity at each pixel, I_i, with f_i by fitting directly the pixel model on the detector using the maximum log-likelihood. Given

$$\log L = \sum_i \frac{1}{2} \left(\frac{I_i - M_i}{\sigma_i} \right)^2 , \tag{8}$$

we want to solve the equation system

$$\frac{d \log L}{d f_k} = 0 \,. \tag{9}$$

The details can be found in 5.1 and result in the following equation

$$\sum_i \frac{I_i}{\sigma_i} \frac{B_{ik}}{\sigma_i} = \sum_i \sum_{k'} \frac{B_{ik}}{\sigma_i} \frac{B_{ik'}}{\sigma_i} f_{k'} \,, \tag{10}$$

which can be re-write in a matrix form as

$$\left(\frac{\mathbf{I}}{\sigma}\right)_{1 \times \mathrm{N_i}} \left(\frac{\mathbf{B}}{\sigma}\right)_{\mathrm{N_i} \times \mathrm{N_k}} = \left(\frac{\mathbf{B}}{\sigma}\right)^T_{\mathrm{N_k} \times \mathrm{N_i}} \left(\frac{\mathbf{B}}{\sigma}\right)_{\mathrm{N_i} \times \mathrm{N_k}} \mathbf{f}_{\mathrm{N_k}} \tag{11}$$

With

$$\left(\frac{\mathbf{B}}{\sigma}\right)_{\mathrm{N_i} \times \mathrm{N_k}} = \mathrm{diag}\left(\frac{1}{\sigma}\right)_{\mathrm{N_i} \times \mathrm{N_i}} \mathbf{B}_{\mathrm{N_i} \times \mathrm{N_k}} \tag{12}$$

This can now be solve for $\mathbf{f}$!

5 Annexe

5.1 Minimization of the $\log L$

From equation 8, we can compute the derivative of the $\log L$.

$$\begin{aligned}\frac{d\log L}{df_k} &= \frac{d}{df_k}\sum_i \frac{1}{2}\left(\frac{I_i - M_i}{\sigma_i}\right)^2 \\ &= \sum_i -\left(\frac{I_i - M_i}{\sigma_i}\right)\frac{d}{df_k}\frac{M_i}{\sigma_i} \\ &= \sum_i -\left(\frac{I_i - \sum_{k\prime} B_{ik\prime} f_{k\prime}}{\sigma_i}\right)\frac{d}{df_k}\frac{1}{\sigma_i}\sum_{k\prime\prime} B_{ik\prime\prime} f_{k\prime\prime}\end{aligned} \tag{13}$$

However, since the $B_{ik\prime\prime}$ are simple coefficients, we have that

$$\frac{d}{df_k}\sum_{k\prime\prime} B_{ik\prime\prime} f_{k\prime\prime} = \sum_{k\prime\prime} B_{ik\prime\prime}\frac{df_{k\prime\prime}}{df_k} = \sum_{k\prime\prime} B_{ik\prime\prime}\delta_{kk\prime\prime} = B_{ik} \tag{14}$$

to finally end up with

$$\frac{d\log L}{df_k} = \sum_i -\left(\frac{I_i - \sum_{k\prime} B_{ik\prime} f_{k\prime}}{\sigma_i}\right)\frac{B_{ik}}{\sigma_i}\,. \tag{15}$$

By setting this equation equal to zero, we have that

$$0 = \sum_i -\frac{I_i}{\sigma_i}\frac{B_{ik}}{\sigma_i} + \sum_i\sum_{k\prime}\frac{B_{ik}}{\sigma_i}\frac{B_{ik\prime}}{\sigma_i} f_{k\prime} \tag{16}$$

which leads to equation 10.

5.2 Integration methods

For to simplification, lets use the following integral,

$$M_i = \sum_n \int_{\lambda_{ni}^-}^{\lambda_{ni}^+} P_{ni} T_n(\lambda)\tilde{f}_n(\lambda)\lambda d\lambda = \sum_n \int_{\lambda_{ni}^-}^{\lambda_{ni}^+} \tilde{g}_{ni}(\lambda) d\lambda. \tag{17}$$

With g being the integrand.

5.2.1 Trapeze integration on a grid

We can approximate the integral in a discrete form with the trapeze method,

$$M_i \simeq \sum_n \left(\frac{\tilde{g}_{ni}(\lambda_{ni}^-) + \tilde{g}_{niL_{ni}}}{2}(\lambda_{L_{ni}} - \lambda_{ni}^-) + \sum_{\tilde{k}=L_{ni}+1}^{H_{ni}-1} \frac{\tilde{g}_{ni\tilde{k}} + \tilde{g}_{ni(\tilde{k}+1)}}{2}\Delta\lambda_{\tilde{k}} + \frac{\tilde{g}_{niH_{ni}} + \tilde{g}_{ni}(\lambda_{ni}^+)}{2}(\lambda_{ni}^+ - \lambda_{H_{ni}}) \right), \tag{18}$$

where L_{ni} and H_{ni} are respectively the lowest and highest $\tilde{k}$ located into the pixel, so that L_{ni} is the smallest $\tilde{k}$ for which $\lambda_{ni}^- \leq \lambda_{\tilde{k}}$ and H_{ni} is the biggest $\tilde{k}$ for which $\lambda_{\tilde{k}\leq\lambda_{ni}^+}$. Now, we need to express $\tilde{g}_{ni}(\lambda_{ni}^-)$ and $\tilde{g}_{ni}(\lambda_{ni}^+)$ with the $\tilde{g}_{ni\tilde{k}}$. To do so, we can use a linear interpolation to write

$$\begin{aligned}\tilde{g}(\lambda_i^-) &= \frac{\lambda_{Lni} - \lambda_i^-}{\Delta\lambda_{L_{ni}-1}}\tilde{g}_{L_{ni}-1} + \frac{\lambda_i^- - \lambda_{L_{ni}-1}}{\Delta\lambda_{L_{ni}-1}}\tilde{g}_{Lni} \text{ and} \\ \tilde{g}(\lambda_i^+) &= \frac{\lambda_{H_{ni}+1} - \lambda_i^+}{\Delta\lambda_{Hni}}\tilde{g}_{H_{ni}} + \frac{\lambda_i^+ - \lambda_{H_{ni}}}{\Delta\lambda_{H_{ni}}}\tilde{g}_{H_{ni}+1}.\end{aligned} \tag{19}$$

Thus, this approximation will be practicable for pixels where $\lambda_i^- \geq \lambda_{\tilde{k}=1}$ and $\lambda_i^+ \leq \lambda_{\tilde{k}=N_{\tilde{k}}}$. We can now replace

into the equation 18 and, some algebra later, we get that

$$
\begin{aligned}
M_i = \sum_n \frac{1}{2} \Bigg(& \frac{(\lambda_{Lni} - \lambda_i^-)^2}{\Delta\lambda_{L_{ni}-1}} \tilde{g}_{L_{ni}-1} \\
& + \left(\lambda_{L_{ni}+1} - \lambda_i^- + \frac{(\lambda_i^- - \lambda_{L_{ni}-1})(\lambda_{Lni} - \lambda_i^-)}{\Delta\lambda_{L_{ni}-1}} \right) \tilde{g}_{L_{ni}} \\
& + \sum_{\tilde{k}=L_{ni}+1}^{H_{ni}-1} \left(\Delta\lambda_{\tilde{k}} + \Delta\lambda_{\tilde{k}-1} \right) \tilde{g}_{\tilde{k}} \\
& + \left(\lambda_i^+ - \lambda_{H_{ni}-1} + \frac{(\lambda_i^+ - \lambda_{Hni})(\lambda_{H_{ni}+1} - \lambda_i^+)}{\Delta\lambda_{Hni}} \right) \tilde{g}_{Hni} \\
& + \frac{(\lambda_i^+ - \lambda_{Hni})^2}{\Delta\lambda_{Hni}} \tilde{g}_{H_{ni}+1} \Bigg) .
\end{aligned}
\tag{20}
$$

This has the form needed to compute the $w_{ni\tilde{k}}$ in the equation

$$
M_i = \sum_n \sum_{\tilde{k}=L_{ni}-1}^{H_{ni}+1} w_{ni\tilde{k}} \tilde{g}_{\tilde{k}} \tag{21}
$$