

Réduction de données - Néo-Narval

Création de la matrice A - `create_A_matrix.py`

Sylvain LUCAS

MàJ du 17 août 2017

1 Introduction

L'objectif de ce programme est de faire un premier lien entre les pixels de la caméra et les longueurs d'onde associées. L'idée est donc de créer un outil capable, à partir d'une image issue de la CCD, de fournir directement un spectre de l'objet observé. Grâce au même outil (la matrice A, voir les explications plus loin), on peut donc obtenir le spectre d'une image stellaire, tout comme une image de flat ou de thorium-argon. Ce rapport a ainsi pour vocation d'expliquer la méthode mise en place en détaillant la structure et la philosophie adoptées pour ce module. Notons que ce document ne se substitue en rien à la documentation interne au code, mais permet d'adopter un point de vue plus global vis-à-vis de ce dernier, en explicitant les choix effectués.

Enfin, on suppose que nous avons la configuration de dossiers suivantes : DATA, FILES, SRC et TEMP, le programme étant enregistré sous `./SRC/create_A_matrix.py`.

2 Philosophie

2.1 Idée générale

On considère qu'on a une image pré-traitée d'une voie d'un ordre. L'idée est de créer un outil permettant de facilement obtenir, à partir de l'image de départ, son spectre associé. Notons n_{pix} le nombre de pixels dans une voie (si la voie est d'épaisseur constante, n_{pix} est le simple produit de la longueur de la voie par son épaisseur en pixels) et n_{art} le nombre de longueurs d'onde considérées pour la voie (c'est-à-dire la résolution souhaitée du spectre). Pour simplifier les choses, on pourra tout d'abord considérer qu'un pixel correspond à une longueur d'onde.

En considérant que l'image de la voie que l'on a est mise sous la forme d'un vecteur colonne de longueur n_{pix} (que l'on appellera Y) et que le spectre recherché est un vecteur colonne de longueur n_{art} (appelé X), alors l'outil que l'on cherche est une matrice de taille (n_{art}, n_{pix}) (appelons la B) telle que $BY = X$. Nous verrons plus tard qu'il nous est plus aisé de construire la matrice A inverse de B , c'est-à-dire telle que $AX = Y$. La matrice A de taille (n_{pix}, n_{art}) nous permettra donc d'accéder directement au spectre (pour simplifier, on aura $X = A^{-1}Y$). Les explications relatives à ce processus se trouvent dans la documentation de la génération de spectre (du fichier `generate_lane_spectrum.py`).

2.2 Quelques précisions sur la construction de la matrice

Comme mentionné ci-dessus, il nous est plus facile de construire la matrice en passant du spectre à l'image que l'inverse. En effet, l'objectif de la matrice A (ou B) est de donner la participation de chaque pixel de la caméra à chacune des longueurs d'onde. Il nous faut donc connaître, pour chacune des longueurs d'onde, les pixels qui y participent. Seule l'image du Fabry-Perot nous donne cette information sans erreur : en effet, chaque raie correspondant à une longueur d'onde, on obtient l'information recherchée (grâce au spectre du Fabry-Perot, voir la figure 3). Ainsi, connaissant les longueurs d'onde (donc X), on peut partiellement remplir partiellement la matrice A . Il nous suffit donc ensuite de remplir le reste de la matrice en interpolant les résultats entre les raies du Fabry-Perot.

La matrice A a pour objectif de nous donner précisément la participation de chaque pixel de la caméra à chacune des longueurs d'onde. Elle se présente donc sous la forme donnée par la figure suivante.

$$A = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \dots & \lambda_{n_{art}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ pix_1 & pix_2 & \vdots & \vdots & pix_{n_{pix}} \end{pmatrix}$$

3 Les données

3.1 En entrée

Il est nécessaire de fournir en entrée les chemins vers certains fichiers ainsi que les informations suivantes :

- un fichier donnant les positions de l’enveloppe gauche (ou haute, suivant le point de vue) de chaque voie. C’est un tableau reçu au format .p (*./TEMP/polys_envelope.p* ouvert avec cPickle) donnant les polynômes de fit des enveloppes de chacune des voies.
- Un fichier donnant l’épaisseur de l’enveloppe pour chacune des voies. Il s’agit d’un simple tableau enregistré au format .p avec cPickle (*./TEMP/lane_thickness.p*).
- le fichier image du Fabry-Perot (format .fts), déjà prétraité. On adopte les même conventions d’axe que dans l’algorithme de recherche des centre des raies.
- Un fichier qui regroupe l’ensemble des positions des centres des raies de l’image du Fabry-Perot donnée. Matrice donnée au format .p ouvert avec cPickle (*./TEMP/fp_slit_position.p*)
- le numéro de l’ordre et de la voie que l’on considère (on construit en effet une matrice A par voie), en sachant que la numérotation des ordres commence à 0 (0 <-> ordre 21)
- le nombre de voies par ordre (1, 2 ou 3) ainsi que la version de la matrice A (1 ou 2, c’est-à-dire avant ou après traitement des cosmiques).

L’ensemble de ces données est fournis par un fichier texte *./DATA/A_matrix_DATA.txt*, lu par le programme afin d’ouvrir les bons fichiers en fonction des différents chemin d’accès. Ce fichier de données est d’ailleurs partagé avec l’algorithme de génération de spectre.

3.2 En sortie

Le programme renvoie tout d’abord la matrice A construite pour la voie considérée et l’enregistre au format suivant : *./TEMP/Amatrix_v(1)_OR(2)_LA(3).npz* avec (1) le numéro de la version de la matrice (1 si l’image utilisée n’a pas encore été traité pour les cosmiques, 2 sinon), (2) étant le numéro de l’ordre (en comptant à partir de 0 pour l’ordre 21) et (3) le numéro de la voie.

Par ailleurs, une entrée est créée (ou modifiée si elle existe déjà) dans le fichier *./DATA/A_matrix_DATA.txt* afin de fournir le nom de la matrice qui vient d’être enregistrée aux programmes suivants.

Attention /!\ : La matrice enregistrée est creuse (format CSR - Compressed Sparse Row) et nécessite d’être ouverte par la fonction *read_A_matrix* du module *./SRC/utils.py* avant d’être utilisée. En effet, un header est écrit dans la matrice A au moment de son enregistrement afin de conserver les informations suivantes :

- la date de l’image source (pour la retrouver éventuellement), au format YearMonthDayHourMinutes.
- le numéro de l’ordre considéré (/!\ ordre 0 <-> 0) ainsi que le numéro de la voie
- le nombre de pixels par arturo (voir plus loin pour l’explication)
- les indices de début et de fin de calcul de la matrice A sur la voie (correspondant au centre de la première et de la dernière raie du Fabry-Perot considérées).

La date peut-être lue grâce à la fonction *date_A_matrix* et le reste du header grâce à la fonction *header_A_matrix* du module *./SRC/utils.py*.

4 L’algorithme en bref

Après avoir ouvert les fichiers nécessaires, le programme crée une première matrice A remplie uniquement avec l’image du Fabry-Perot. On a ainsi de l’information pour 100 à 300 longueurs d’onde sur la voie. Une

fois cela fait, l'interpolation entre ces longueurs d'ondes connues est lancée (avec parallélisation du calcul). Il suffit ensuite de rassembler l'ensemble des informations de remplissage inter-raies et de former la matrice globale A.

5 L'algorithme en détail

5.1 Initialisation (`create_A_matrix`)

Le programme charge tout d'abord par le biais de cette fonction l'ensemble des fichiers et informations nécessaires à la construction de la matrice. C'est également cette fonction qui lance le calcul de la matrice et s'occupe d'enregistrer au bon format la matrice une fois qu'elle est calculée.

5.2 Coupure de la voie sur les bords (`cut_border_edge`)

Pour les ordres les plus élevés (ce cas ne se présente que pour les ordres 59 et 60 avec Narval), la courbure des ordres et la dimension du CCD font que l'ordre peut être coupé par le bas de la caméra. Lorsque l'on traite une voie, il ne faut donc pas l'analyser sur toute la longueur du CCD. Cette méthode permet donc d'identifier les indices de début et de fin de la voie.

5.3 Remplissage de la matrice A avec le Fabry-Perot (`fill_A_fp`)

Il a précédemment été fait mention d'arturos. Il s'agit d'une unité alternative utilisée pour décrire la résolution d'un spectre sans connaître son équivalent en mètres. Ainsi, avec l'équivalence d'un arturo par pixel, les spectres fournis auront 4612 points (dans le cas de NARVAL) et auront une résolution de 2,6 km/s. En revanche, si on souhaite avoir de la super-résolution, il faut augmenter le nombre d'arturos par pixel. Ainsi, à 1,5 arturo/pixel, on passe à 1,73km/s par pixel (et le spectre en sortie aura donc 6918 points pour NARVAL).

Une fois que le nombre d'arturos par pixel est choisi, on peut s'atteler à remplir la matrice. Pour ce faire, on utilise les coordonnées des centres des raies du Fabry-Perot qui ont été préalablement calculées afin d'isoler chacune des raies. On définit ensuite autour de chaque centre de raie une fenêtre (de hauteur l'épaisseur de la voie et de largeur variable suivant le Fabry-Perot utilisé *). On connaît précisément la longueur d'onde correspondant à la raie (en arturos, relativement à la voie du moins), ce qui nous permet de remplir la colonne correspondante dans la matrice A.

On constitue ainsi une matrice creuse A_{fp} , remplie uniquement sur sa diagonale et dans les colonnes correspondant aux raies du Fabry-Perot. On a ainsi un outils efficace pour nous donner les informations de l'images aux longueurs d'onde des raies du Fabry-Perot.

* : le Fabry-Perot actuel fournit des raies trop larges pour être utilisables directement dans la DRS. On choisit donc une fenêtre d'une largeur correspondant à une seule colonne dans ce cas, afin d'affiner artificiellement et grossièrement la fente. Dans le cas d'un meilleur Fabry-Perot, ce qui sera normalement le cas pour Néo-Narval, on pourra prendre une fenêtre plus large (11 pixels de large).

5.4 Interpolation entre deux raies du Fabry-Perot (`launch`)

Une fois que la matrice A_{fp} a été créée, on peut ensuite la remplir. Il faut donc pour chaque longueur d'onde restante, déterminer la participation individuelle de chaque pixel de la caméra. Pour cela, on utilise l'information du Fabry-Perot en interpolant entre deux raies successives. Pour ce faire, on met en place une interpolation quadratique qui prend en compte les décalage en y dû aux arrondis effectués. On préfère ainsi ce modèle à une interpolation linéaire qui ne tient compte que de la différence en longueur d'onde λ des raies du Fabry-Perot. On identifie alors pixel à pixel entre les zones de chaque raie et on interpole (en notant 0 et 1 deux raies successives du Fabry-Perot, et λ une longueur d'onde intermédiaire). D'où pour chaque paire de pixels :

$$I_{\lambda} = I_0 + \frac{I_1 - I_0}{\lambda_1 - \lambda_0} \lambda, \text{ avec } \lambda \in [\lambda_0; \lambda_1]$$

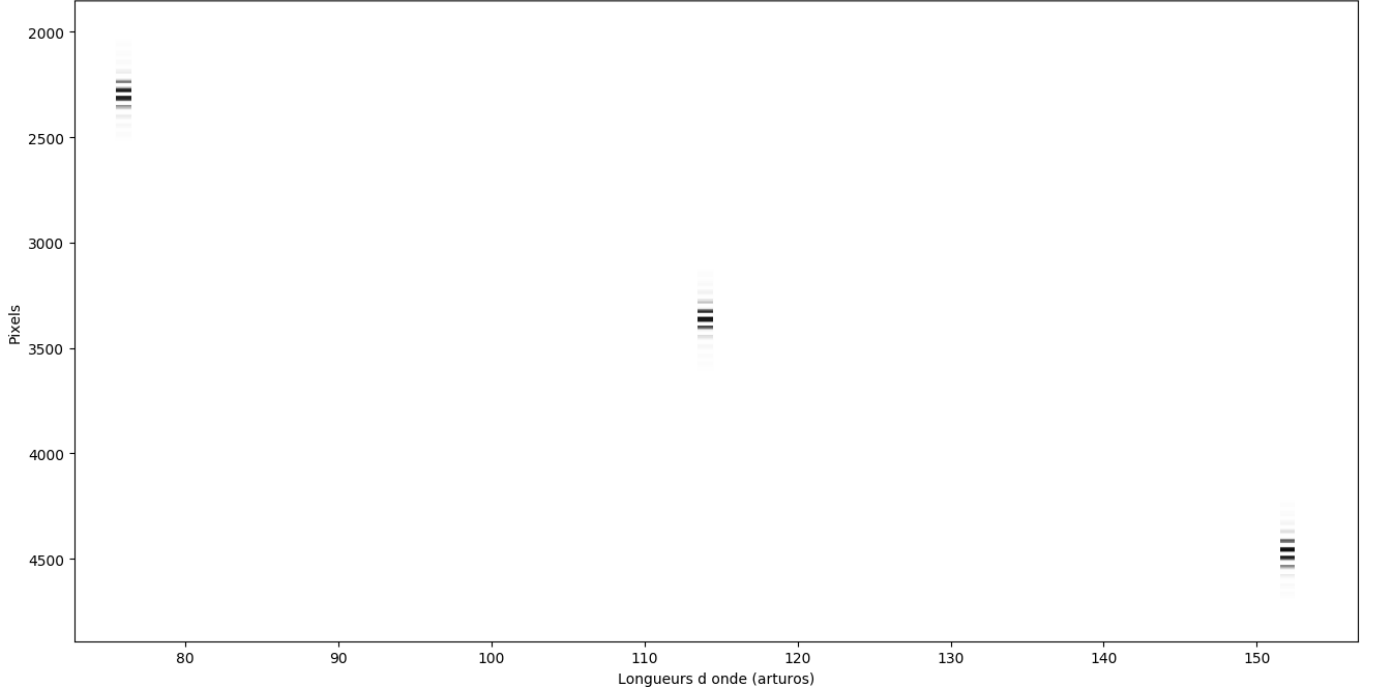


FIGURE 1 – Matrice A_{fp}



FIGURE 2 – Une longueur d'onde dans la matrice A_{fp}

Ce modèle est améliorable en incorporant le Δy dû à la courbure des ordres. Δy est le déplacement sur l'axe y entre les raies 0 et 1 (figure 4). Entre les deux, ce déplacement sera petit. On peut donc le linéariser $\Delta y(\lambda) = k\lambda$. L'interpolation sera donc une fonction de λ seulement.

On suppose le modèle $I = a\lambda^2 + b\lambda + c$. On a donc :

$$I_1 - I_0 = a(\lambda_1^2 - \lambda_0^2) + b(\lambda_1 - \lambda_0) (*)$$

Il existe deux cas particuliers qui nous conduisent à dire que $b \neq 0$, et que a est probablement également non nul :

- Si $b = \frac{I_1 - I_0}{\lambda_1 - \lambda_0}$ (modèle linéaire) alors on a $a = 0$
- Si $b = 0$, alors $a = \frac{I_1 - I_0}{\lambda_1^2 - \lambda_0^2}$ et donc toute dépendance en Δy disparaît.

Le déplacement Δy a comme conséquence qu'au pixel i , l'intensité mesurée I_{1_i} n'est pas l'intensité attendue \tilde{I}_{1_i} . Cette différence est due aux photons échangés avec les pixels $i + 1$ et $i - 1$. Comme Δy est strictement

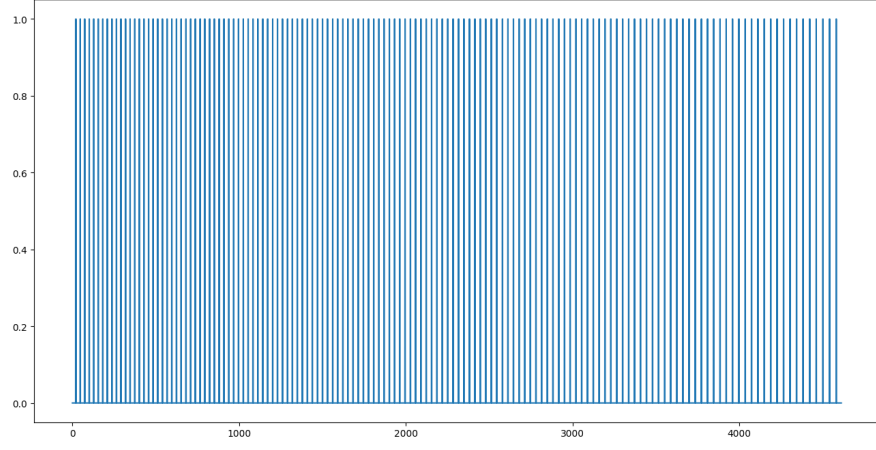


FIGURE 3 – Spectre du Fabry-Perot

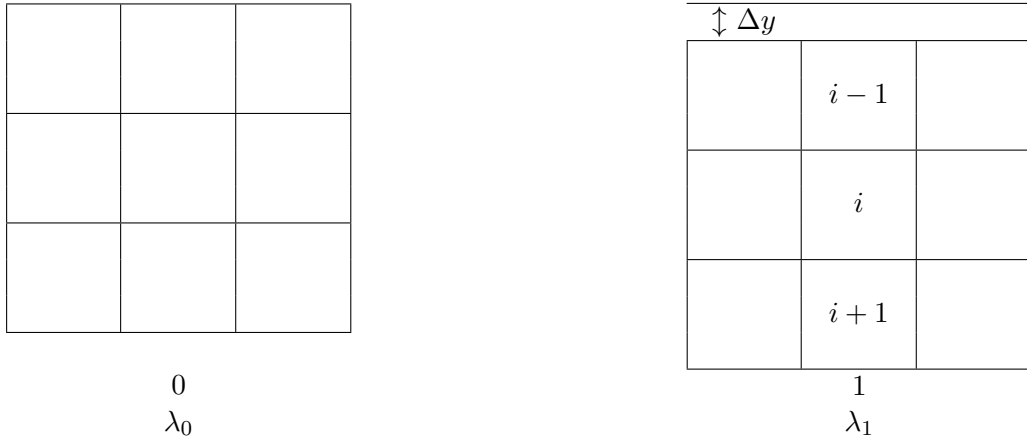


FIGURE 4 – Interpolation quadratique - déplacement en Y entre deux raies

vertical, seuls $i + 1$ et $i - 1$ contribuent à cette perturbation que l'on suppose linéaire :

$$I_{1_i} = \tilde{I}_{1_i} + \Delta y \tilde{I}_{1_{i+1}} - \Delta y \tilde{I}_{1_{i-1}}$$

En insérant cette nouvelle expression dans (*) avec b égal au modèle linéaire ($b = \frac{\tilde{I}_{1_i} - I_{0_i}}{\lambda_1 - \lambda_0}$), on trouve que $a = -\Delta y \frac{\tilde{I}_{1_{i+1}} - \tilde{I}_{1_{i-1}}}{(\lambda_1 - \lambda_0)(\lambda_1^2 - \lambda_0^2)}$.

On sait que $\Delta y < 0.5$ (par construction). Si l'approximation linéaire est bonne, alors Δy devrait être petit, on néglige donc Δy^3 et on se limite à un cas avec les plus proches voisins. On a donc :

$$\begin{aligned} \tilde{I}_{1_i} &= I_{1_i} + \Delta y (\tilde{I}_{1_{i-1}} - \tilde{I}_{1_{i+1}}) \\ \tilde{I}_{1_{i-1}} &= I_{1_{i-1}} - \Delta y \tilde{I}_{1_i} \\ \tilde{I}_{1_{i+1}} &= I_{1_{i+1}} + \Delta y \tilde{I}_{1_i} \end{aligned}$$

On peut alors résoudre :

$$\begin{aligned} \tilde{I}_{1_i} &= I_{1_i} + \Delta y (I_{1_{i-1}} - I_{1_{i+1}}) - 2\Delta y^2 \tilde{I}_{1_i} \\ \tilde{I}_{1_i} &= \frac{1}{1 + 2\Delta y^2} [I_{1_i} + \Delta y (I_{1_{i-1}} - I_{1_{i+1}})] \end{aligned}$$

Avec les valeurs de \tilde{I} introduites dans les expressions pour a et b , on définit donc une interpolation quadratique en fonction de λ .

On a donc l'interpolation pour toute longueur d'onde λ comprise entre λ_0 et λ_1 :

$$I_{\lambda_i} = I_{0_i} + \frac{\lambda - \lambda_0}{\lambda_1 - \lambda_0} ((a + \tilde{I}_{\lambda_i}) - I_{0_i})$$

$$\text{avec } a = -\Delta y \frac{\tilde{I}_{1_{i+1}} - \tilde{I}_{1_{i-1}}}{(\lambda_1 - \lambda_0)}$$

5.5 Construction de la matrice globale (**fill_A_matrix**)

Les interpolations entre les différentes raies du Fabry-Perot sont calculées en parallèle. Il suffit donc ensuite de rassembler l'ensemble des données afin de créer la matrice globale. L'étape précédente permet de créer une matrice creuse A_{interp} qui contient l'ensemble des données de l'interpolation, avec des trous au niveau des raies du fabry-Perot (données non interpolées). On a donc alors :

$$A_{global} = A_{fp} + A_{interp}$$

Avant de construire cette dernière matrice, on a préalablement pris soin de rajouter les informations du header dans le coin supérieur droit (vide de toute donnée). Ces informations sont :

- la date (YearMonthDayHourMinute) du fichier Fabry-Perot source
- l'ordre
- la voie au sein de l'ordre
- le nombre de pixels par arturo (résolution du spectre)
- les indices de début et de fin de calcul au sein de la voie