

# Documentation Boucle Néo-Narval

Anthony NOLL

## Introduction

Afin de relier les différentes parties de la réduction de données entre elles, mais aussi de simuler une nuit à l'observatoire, la mise en place d'une boucle est nécessaire. Celle-ci a été programmée en bash afin de gérer au mieux les fichiers.

La documentation suivante aura pour but d'expliquer le fonctionnement du code afin de saisir plus facilement sa structure. Celui-ci est aussi abondamment commenté, ce qui permet de saisir plus facilement son déroulement.

## Déroulement global

Pour chaque nuit, la boucle va d'abord chercher à produire toutes les matrices A nécessaires à la réduction de données. Pour cela, elle a besoin des fichiers suivants :

- Flat (généralement 10)
- Bias (généralement 1)
- Fabry-Pérot (généralement 1)

Une fois ces images obtenues, les matrices A sont calculées. Une fois celles-ci entièrement calculées, le spectre du flat est émis. Puis, après cela, le système de réduction de données prend en entrée chaque fichier étoile ou thorium-argon pour le traiter indépendamment.

## Calcul des matrices A

Le calcul des matrices A se fait via le script `arthur2.sh`. Celui-ci est prévu pour être lancé chaque nuit avant la toute première image de pré-traitement, donc en début d'après-midi. Il prend les options suivantes :

- `-v [NBR_VOIES]` afin de gérer le nombre de voies (2 pour Narval, 3 pour Néo-Narval)
- `-f [NBR_FLAT]` afin de gérer le nombre de flat nécessaires au prétraitement
- `-b [NBR_BIA]` afin de gérer le nombre de bias nécessaires au prétraitement
- `-F [NBR_FP]` afin de gérer le nombre de Fabry-Pérot

Ensuite, le script va vérifier l'existence ou non d'un fichier lock, afin d'éviter le lancement de plusieurs scripts en même temps.

## Prétraitement

Pour détecter l'arrivée des nouveaux fichiers dans le dossier `Brut`, on utilise `inotifywait` qui permet de scruter l'état de celui-ci. Cette fonction bloque l'avancée du script jusqu'à ce qu'une modification du type précisé (ici la création ou le déplacement de fichier dans le dossier) arrive. Alors, on regarde la dénomination du fichier pour incrémenter le bon compteur et ainsi lancer les prétraitements seulement quand tous les fichiers sont réunis.

Une fois les fichiers réunis, on lance dans l'ordre :

- `Search_bias` qui recherche les bias dans Brut fait la médiane entre eux et crée un fichier `10b` dans `SRC/TEMP`
- `Search_flat` qui recherche les flat dans Brut, fait la médiane entre eux, retire le bias et crée ainsi un fichier `10f` dans `SRC/TEMP`
- `Search_fp` qui recherche les Fabry-Pérot, les somme et retire le bias, et crée ainsi un fichier `fp1` dans `SRC/TEMP`
- `envelope` qui émet les fichiers contenant les enveloppes des voies ainsi que leur épaisseur, et les mets dans `SRC/TEMP`.

— `find_fp_slits` qui permet de trouver les fentes du Fabry-Pérot.

## Création des matrices A1

Ensuite, pour chaque voie, la matrice A1 correspondante va être calculée, à l'aide de `create_A_matrix`. Le script s'occupe aussi de modifier le fichier `SRC/DATA/Amatrix_DATA.txt` afin de configurer le python. Ces matrices sont de nomenclature `Amatrix_v1_OR[num_ordre]_LA[num_voie].npz`.

## Création des matrices A2

Les matrices A2 sont des matrices créées à partir d'un fichier de Fabry-Pérot auquel les cosmiques ont été enlevés. Pour cela, on va avec la matrice A1 créer le spectre du fabry-pérot de niveau 1 (`fp1`). Ensuite, on crée le spectre d'une image entièrement composée de 1, afin de pouvoir normaliser le Fabry-Pérot sans perdre les caractéristiques propres à l'instrument : en effet, seuls les défauts causés par la méthode de la matrice A seront éliminés de cette façon.

Après, une image virtuelle du Fabry-Pérot est calculée, à partir du spectre. Alors, `cosmic_auto` est utilisé pour traiter cette image et donc produire un `fp2` sans cosmiques. Alors, on peut créer les matrices A2 à partir de ces nouvelles images, matrices de meilleures qualités car dénuées de cosmiques.

## Gestion des étoiles et Thorium-Argon

La gestion des étoiles et des ThA (Thorium-Argon) tout au long de la nuit se fait grâce au script `stars2.sh`. Celui-ci va tout d'abord vérifier si le prétraitement a réussi (réussite attestée par le fichier `flag`) puis vérifier l'existence d'un lock afin d'éviter le lancement en parallèle de deux scripts. Ce script est prévu pour être lancé après la création des matrices A, et doit tourner durant toute la nuit.

Tout comme `arthur2.sh`, ce script va utiliser `inotifywait` pour scruter les fichiers Brut, et les traiter selon leur type.

## Traitement des étoiles et Thorium-Argon

Le traitement des fichiers étoiles et Th-A est assuré par `stars3.sh`. Il est lancé automatiquement à partir de `stars2.sh`. Afin d'éviter une modification concurrente des fichiers DATA, il doit être évité à tout prix que plusieurs script `stars3.sh` se lancent en même temps. Pour cela, un système de plusieurs fichiers lock a été mis en place. Ainsi, le  $n^{\text{ème}}$  script étoile lancé va créer le fichier lock `.star[n]` et va attendre la suppression du fichier `.star[n-1]` (le script sait quel est le fichier  $n-1$  en prenant le dernier fichier de type `.star*` crée.) Grâce à ce système, plusieurs étoiles peuvent être lancées les unes après les autres, sans qu'il y ait exécution en parallèle.

Le script effectue ensuite le prétraitement du ou des fichiers d'entrée via les python `Search_star` ou `Search_thar`. On notera au passage que ce même script est utilisé pour les fichiers de type flat.

Après, les spectres sont calculés en utilisant la matrice A2, calculée à la fin du prétraitement de début de nuit. Ensuite, de façon analogue à ce que l'on a fait pour les Fabry-Pérot, on calcule une image virtuelle qui permettra d'enlever les cosmiques et ainsi d'obtenir un spectre de meilleure qualité.

Enfin, une fois que tous les spectres sont calculés, `concatenate_toFITS` est appelé pour réunir tous les spectres en un unique sous format fits. Celui-ci contient une table de deux colonnes pour chaque voie : la longueur d'onde et l'intensité correspondante. Si la conversion Arturo  $\rightarrow$  nanomètre est impossible, le spectre de sortie est en Arturo (soit un index pour la colonne de gauche).

## Script traitement global

Si une telle gestion de fichiers correspond à une utilisation lors d'une nuit au Pic, elle n'est pas compatible avec un traitement d'un ensemble de fichiers d'un coup. Pour cela, un autre script a été créé de façon à, à partir d'un ensemble de fichiers dans un dossier, émuler une ou plusieurs nuit en copiant les fichiers un par un. Cette copie "lente" des fichiers est nécessaire au bon fonctionnement de `inotifywait` qui ne gère pas correctement l'apparition de plusieurs fichiers simultanément.

Ce script regarde les fichiers dans le dossier `SRC/Brut_sas`. Si les fichiers sont de nomenclature `Narval_YYYYMMDD_HHmmSS_[tpe].fts` alors ils sont directement traités, ils sont renommés d'après l'heure du header fits sinon. Ensuite, chaque fichier est placé dans un dossier nommé selon la nuit `YYYYMMDD`. Plus précisément, tous

les fichiers séparés de moins 20h (paramètre réglable) du premier fichier de la nuit se retrouvent dans ce dossier : ainsi, même une image d'étoile prise le lendemain à 3h sera dans le dossier de la veille.

Une fois ce tri effectué, **arthur2.sh** est lancé et les fichiers de type flat, fabry-pérot et bias sont copiés dans **Brut** et donc traités. Une fois le traitement terminé, on lance **stars2.sh** et on traite les fichiers de type étoile ou Th-A.

Le script passe à la nuit prochaine quand le dernier fichier lock star est supprimé.

## Rôle de chaque fichiers

### Dossiers

- **SRC** Contient l'ensemble des fichiers de la réduction de données
- **Brut** Contient les fichiers bruts sortant de l'instrument, en attente d'être traités
- **Brut\_sas** Dossier dans lequel sont déposés les fichiers pour un traitement via **script.sh**
- **TEMP** Contient tous les fichiers temporaires du traitement
- **DATA** Contient tous les fichiers de paramétrages de la réduction de données
- **SRC** Contient tous les fichiers de code source

### Fichiers dans SRC

- **arthur2.sh** : Permet d'effectuer le prétraitement
- **bias\_creator.py** : Créer un bias artificiel
- **bias\_generator.py** : Créer un bias en copiant réel
- **calcul\_bord.py** : Outil pour calculer les bords
- **Compare\_thar.py** : Outil pour comparer les spectres de ThA
- **concatenate\_toFITS.py** : Concatène les spectres vers un fits
- **cosmic\_auto.py** : Enlève les cosmiques d'une image
- **create\_A\_matrix.py** : Crée une matrice A
- **Create\_ref\_th.py** : Crée un spectre de thorium avec valeurs en absolue
- **envelope.py** : Calcule les enveloppes des ordres
- **find\_fp\_slits.py** : Trouve les fentes du FP
- **functions.sh** : Contient les fonctions des scripts bash
- **generate\_lane\_spectrum.py** : Génère le spectre d'une voie
- **methods.py** : Contient les fonctions du simulateur
- **narval\_simu2.py** : Simulateur de Narval
- **narval\_simu3.py** : Simulateur de Néo-Narval
- **normalization.py** : Normalise les spectres
- **philippe.sh** : (Obsolète) Simule une nuit
- **renamator.py** : Permet de renommer un fichier vers la nouvelle nomenclature
- **script.sh** : Permet de traiter un ensemble de fichier dans **Brut\_sas**
- **Search\_bias.py** : Permet de chercher et prétraiter les bias
- **Search\_flat.py** : Permet de chercher et prétraiter les flat
- **Search\_fp.py** : Permet de chercher et prétraiter les FP
- **Search\_star.py** : Permet de chercher et prétraiter les étoiles
- **Search\_thar.py** : Permet de chercher et prétraiter les ThA
- **settings.py** : Contient une fonction pour le simulateur
- **sp\_visu.py** : Visualisateur de spectre fits et pickle
- **star\_generator.sh** : Permet de générer une étoile à partir d'un fichier
- **stars2.sh** : Script pour gérer les étoiles lors d'une nuit
- **stars3.sh** : Script pour traiter une étoile, ThA ou flat
- **thar3.py** : Création d'une référence de conversion arturo/mètre
- **utils.py** : Contient plusieurs fonctions utiles, ainsi que la classe **config**
- **virtual\_CCD.py** : Permet de générer une image artificielle à partir d'un spectre