

# 1

## Validation des mocks

---

Ce chapitre a pour vocation de présenter l’analyse menée sur les mocks, afin de valider leur construction et le choix des différents paramètres. Nous présentons d’abord le calcul des diverses fonctions de corrélations, puis l’analyse des 100 réalisations produites. Pour faire cette analyse, nous utilisons le code `picca` qui permet de calculer les champs  $\delta$ , les fonctions de corrélations, les matrices de distorsion, les matrices de covariance ainsi que de réaliser l’ajustement du modèle.

## 1 Les estimateurs

Nous présentons ici les estimateurs utilisés pour calculer la fonction d’auto-corrélation du  $\text{Ly}\alpha$ , la fonction de corrélation croisée  $\text{Ly}\alpha$ -QSO, ainsi que la fonction d’auto-corrélation d’objets ponctuels tels les quasars. Dans le cas des objets ponctuels, la seule information nécessaire au calcul de la fonction de corrélation est le catalogue fournissant pour chaque objet l’ascension droite, la déclinaison et le redshift. Pour le calcul des fonctions de corrélation impliquant le  $\text{Ly}\alpha$ , il est nécessaire de calculer au préalable le champ  $\delta_F$  (équation ??). Nous distinguons deux cas : le cas où nous analysons les *raw mocks* et le cas où nous analysons les *cooked mock* ou les données. Les raw mocks (les mocks bruts, non préparés) désignent les mocks avant d’avoir tourné le code `quickquasars`. Les fonctions de corrélation sont alors calculées en utilisant directement la fraction de flux transmise  $F$  donnée dans les fichiers de transmissions. Etant donné que pour chaque forêt, nous avons directement accès à  $F$ , le champ  $\delta_F$  est donné par

$$\delta_F = \frac{F}{\overline{F}} - 1. \quad (1.1)$$

Les cooked mocks quant à eux désignent les mocks après avoir appliqué `quickquasars` à chaque forêt. Dans ce cas, comme pour les données, la fraction de flux transmise  $F$  n’est pas accessible. Il faut alors utiliser la procédure d’ajustement du continuum décrite dans la section ?. Une fois le champ  $\delta_F$  calculé, nous pouvons calculer les différentes fonctions de corrélation.

### 1.1 L’auto-corrélation $\text{Ly}\alpha \times \text{Ly}\alpha$

Comme expliqué dans le chapitre d’introduction, la fonction d’auto-corrélation du champ d’absorption  $F$  du  $\text{Ly}\alpha$  est définie comme

$$\xi_{\text{Ly}\alpha}(\vec{r}) = \langle \delta_F(\vec{r}') \delta_F(\vec{r} + \vec{r}') \rangle. \quad (1.2)$$

Elle peut-être mise sous la forme

$$\xi(\vec{r}) = \langle \delta_i \delta_j \rangle, \quad (1.3)$$

où  $\langle . \rangle$  désigne la moyenne sur tous les pixels  $i$  et  $j$  qui vérifient  $\vec{r}_{ij} = \vec{r}$ . Afin de calculer  $\xi_{\text{Ly}\alpha}$ , nous créons une grille en  $(r_{\parallel}, r_{\perp})$ . Les bins mesurent  $4 h^{-1}$  Mpc dans chaque direction. Cette taille de bin permet à la fois de regrouper suffisamment de pixels pour avoir une bonne statistique dans chaque bin, mais aussi d’avoir suffisamment de bins pour résoudre correctement la région du pic BAO (#prov est-ce que c’est vrai ? toujours la même quantité d’info, donc ça améliore pas le fit ?). Une fois cette grille construite, nous transformons les coordonnées  $(\alpha, \delta, z)$  des pixels  $\delta_F$  en distances (#prov donner les transformations géométriques ?). Puis, pour chaque bin  $A$  de la grille en  $(r_{\parallel}, r_{\perp})$ , nous considérons toutes les paires de pixels dont la distance de séparation se trouve dans ce bin  $A$ . La fonction de corrélation dans ce bin est alors donnée par

$$\xi_A = \frac{\sum_{(i,j) \in A} w_i w_j \delta_i \delta_j}{\sum_{(i,j) \in A} w_i w_j}, \quad (1.4)$$

où  $w_i$  est le poids associé au pixel  $i$ . Ces poids sont définis dans l’équation ?. La correction de la dépendance en redshift du biais du  $\text{Ly}\alpha$  permet de donner plus de poids aux pixels à grand redshift,

où l'amplitude de la fonction de corrélation est plus importante. Le calcul de la fonction de corrélation est donc effectué à l'aide d'une double boucle sur les pixels. Afin de réduire le temps de calcul, nous calculons la fonction de corrélation uniquement pour les paires de pixels pour lesquelles  $r_{\parallel}$  et  $r_{\perp}$  sont inclus dans  $[0; 200]h^{-1}$  Mpc. La fonction de corrélation est donc calculée dans  $50 \times 50 = 2500$  bins. Les paires formées par des pixels provenant de la même forêt sont exclues du calcul afin d'éviter que les erreurs sur l'ajustement du continuum biaisent la mesure de la fonction de corrélation.

L'analyse Ly $\alpha$  des données complète d'eBOSS (?) utilise deux fonctions de corrélation distinctes : les fonctions de corrélation Ly $\alpha$ (Ly $\alpha$ ) $\times$ Ly $\alpha$ (Ly $\alpha$ ) et Ly $\alpha$ (Ly $\alpha$ ) $\times$ Ly $\alpha$ (Ly $\beta$ ). La région Ly $\beta$  ne fournit pas suffisamment de pixels pour pouvoir calculer la fonction de corrélation Ly $\alpha$ (Ly $\beta$ ) $\times$ Ly $\alpha$ (Ly $\beta$ ) et y détecter le pic BAO. L'utilisation de ces deux fonctions de corrélation permet de mesurer la position du pic BAO avec une plus grande précision. Dans ce manuscrit, nous nous intéressons à la mesure de  $b_{\text{Ly}\alpha}$  et  $\beta_{\text{Ly}\alpha}$  afin de construire correctement nos mocks. Pour limiter les potentielles systématiques, nous considérons donc uniquement la fonction de corrélation Ly $\alpha$ (Ly $\alpha$ ) $\times$ Ly $\alpha$ (Ly $\alpha$ ). Le graphique de droite de la figure ?? présente les distributions pondérées en redshift des paires Ly $\alpha$ (Ly $\alpha$ ) $\times$ Ly $\alpha$ (Ly $\alpha$ ) et Ly $\alpha$ (Ly $\alpha$ ) $\times$ Ly $\alpha$ (Ly $\beta$ ). L'analyse des données DR16 que nous présentons ici considère donc uniquement la distribution indiquée en orange.

## 1.2 La corrélation croisée Ly $\alpha$ $\times$ QSO

Nous donnons dans cette section l'estimateur de la fonction de corrélation croisée Ly $\alpha$  $\times$ QSO. De la même manière que précédemment, le calcul s'effectue dans des bins en  $(r_{\parallel}, r_{\perp})$ . La fonction de corrélation dans le bin  $A$  est donnée par

$$\xi_A = \frac{\sum_{(i,j) \in A} w_i w_j \delta_i}{\sum_{(i,j) \in A} w_i w_j}, \quad (1.5)$$

où l'indice  $i$  court sur les pixels des forêts et  $j$  sur les quasars pour lesquels la distance de séparation est comprise dans le bin  $A$ . Comme précédemment, les paires  $ij$  où le pixel  $i$  appartient à la forêt du quasar  $j$  sont rejetées. Les poids  $w_j$  associés aux quasars, similairement au Ly $\alpha$ , favorisent les quasars à plus grand redshift. Ces poids sont définis comme

$$w_j = \left( \frac{1 + z_j}{1 + 2.25} \right)^{\gamma_{QSO}}, \quad (1.6)$$

où  $\gamma_{QSO} = 1.44 \pm 0.08$  (?). Dans le calcul de la fonction de corrélation croisée Ly $\alpha$  $\times$ QSO, nous nous restreignons aux paires pour lesquelles  $r_{\perp} \in [0; 200]h^{-1}$  Mpc. Contrairement à l'auto-corrélation du Ly $\alpha$ , la fonction de corrélation croisée n'est pas symétrique en  $r_{\parallel}$ . Nous la calculons donc dans les bins pour lesquels  $r_{\parallel} \in [-200; 200]h^{-1}$  Mpc. La fonction de corrélation croisée est donc calculée dans  $50 \times 50 = 2500$  bins.

## 1.3 Le spectre de puissance à une dimension

Comme expliqué dans le chapitre précédent, nous ajustons le spectre de puissance  $P_{\text{miss}}$  appliqué à  $\delta_s$  de façon à obtenir un spectre de puissance à une dimension  $P^{1D}$  en accord avec les données. Nous présentons donc maintenant la mesure de ce spectre de puissance. Pour chaque forêt, nous appliquons une transformation de Fourier au champ  $\delta_F$  afin d'obtenir le champ  $\delta_k$ . Puis, le spectre de puissance de cette forêt est obtenu comme

$$P^{1D}(k) = |\delta_k^2|. \quad (1.7)$$

Nous répétons cette procédure pour toutes les forêts, puis le spectre de puissance total est obtenu comme la moyenne du spectre de puissance de chaque forêt.

En ce qui concerne les données, la mesure du spectre de puissance à une dimension est plus complexe. Un certain nombre d'effets liés à la mesure doivent être pris en compte. Le bruit, par exemple, doit être estimé afin d'être pris en compte pour ne pas fausser l'estimation du spectre de puissance. Cette analyse est détaillée dans ?.

#### 1.4 La fonction de corrélation à une dimension

Important / nécessaire ?

### 2 Les matrices de distorsion

L'ajustement du continuum nécessaire au calcul du champ  $\delta_F$  dans les données et les coocked mocks biaise le champ mesuré. Cependant, grâce à la transformation (équation ??) décrite dans la section ??), l'effet sur la fonction de corrélation peut-être pris en compte. L'idée est la suivante : modéliser la distorsion induite par l'ajustement du continuum et par la transformation ?? sur la fonction de corrélation, appliquer cette distorsion au modèle, puis ajuster le modèle "distordu" aux données. L'ajustement du continuum et la transformation ?? induisent des corrélation le long de la ligne de visée. Au premier ordre, nous pouvons considérer que chaque  $\delta_F$  d'une forêt est une combinaison linéaire de tous les  $\delta_F$  de cette forêt. La fonction de corrélation distordue dans le bin  $A$  peut alors être reliée à la vraie fonction de corrélation comme

$$\xi_{distortion}(A) = \sum_B D_{AB} \xi_{vraie}(B), \quad (1.8)$$

où  $D_{AB}$  est appelée la *matrice de distorsion*. Celle-ci s'exprime en fonction du projecteur  $\eta_{ij}^q$  défini dans l'équation ??. Pour l'auto-corrélation, elle est défini comme

$$D_{AB} = \frac{1}{W_A} \sum_{(i,j) \in A} w_i w_j \left( \sum_{(i',j') \in B} \eta_{ii'} \eta_{jj'} \right) \quad (1.9)$$

où  $W_A = \sum_{(i,j) \in A} w_i w_j$  est le poids du bin  $A$ . Pour la corrélation croisée, la matrice de distorsion est donnée par

$$D_{AB} = \frac{1}{W_A} \sum_{(i,j) \in A} w_i w_j \left( \sum_{(i',j') \in B} \eta_{ii'} \right). \quad (1.10)$$

Comme précédemment, les indices  $i$  correspondent aux pixels des forêts, et  $j$  aux quasars. A cause