

Stage de Master 2
Université Claude Bernard

Modélisation des observations de spectroscopie sans fente de l'Auxtel dans le cadre de LSST

Supervisé par Yannick Copin
Equipe Cosmos de l'IPNL

Introduction

Univers \approx énergie noire + matière noire
 $\approx 69\%$ $\approx 26\%$

Responsable de l'accélération
de l'expansion de l'univers

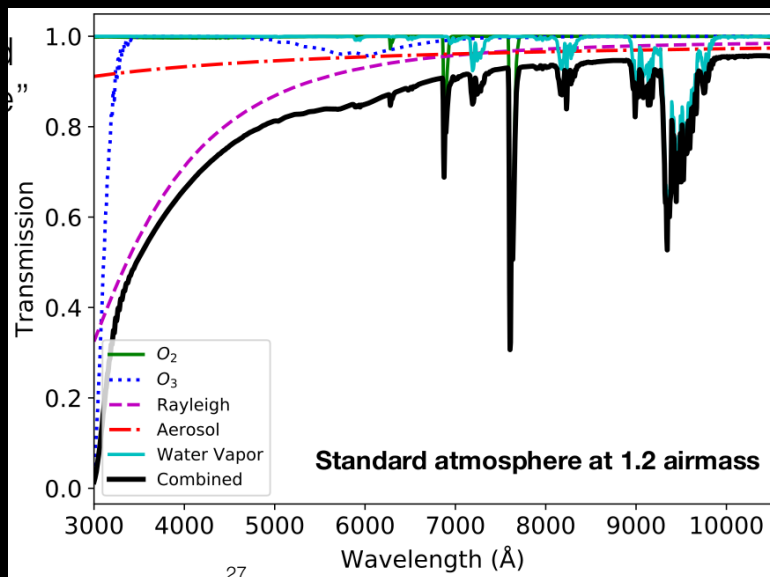


Supernovæ de type Ia
"chandelles standard"

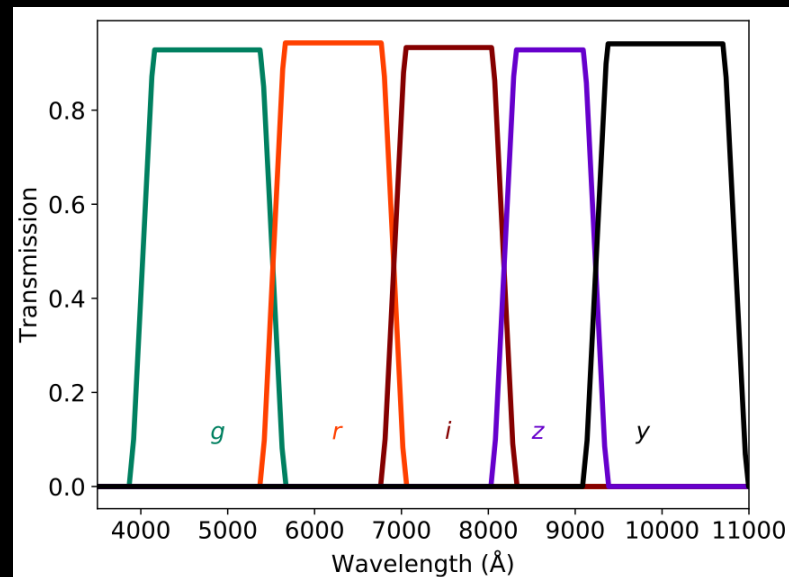
contraindre à 0,1% les
erreurs de calibration

Auxtel

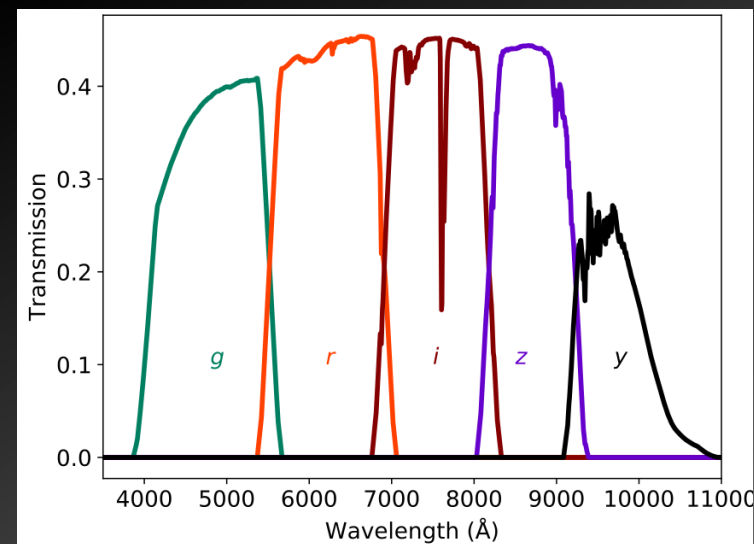
L'impact de l'atmosphère



×

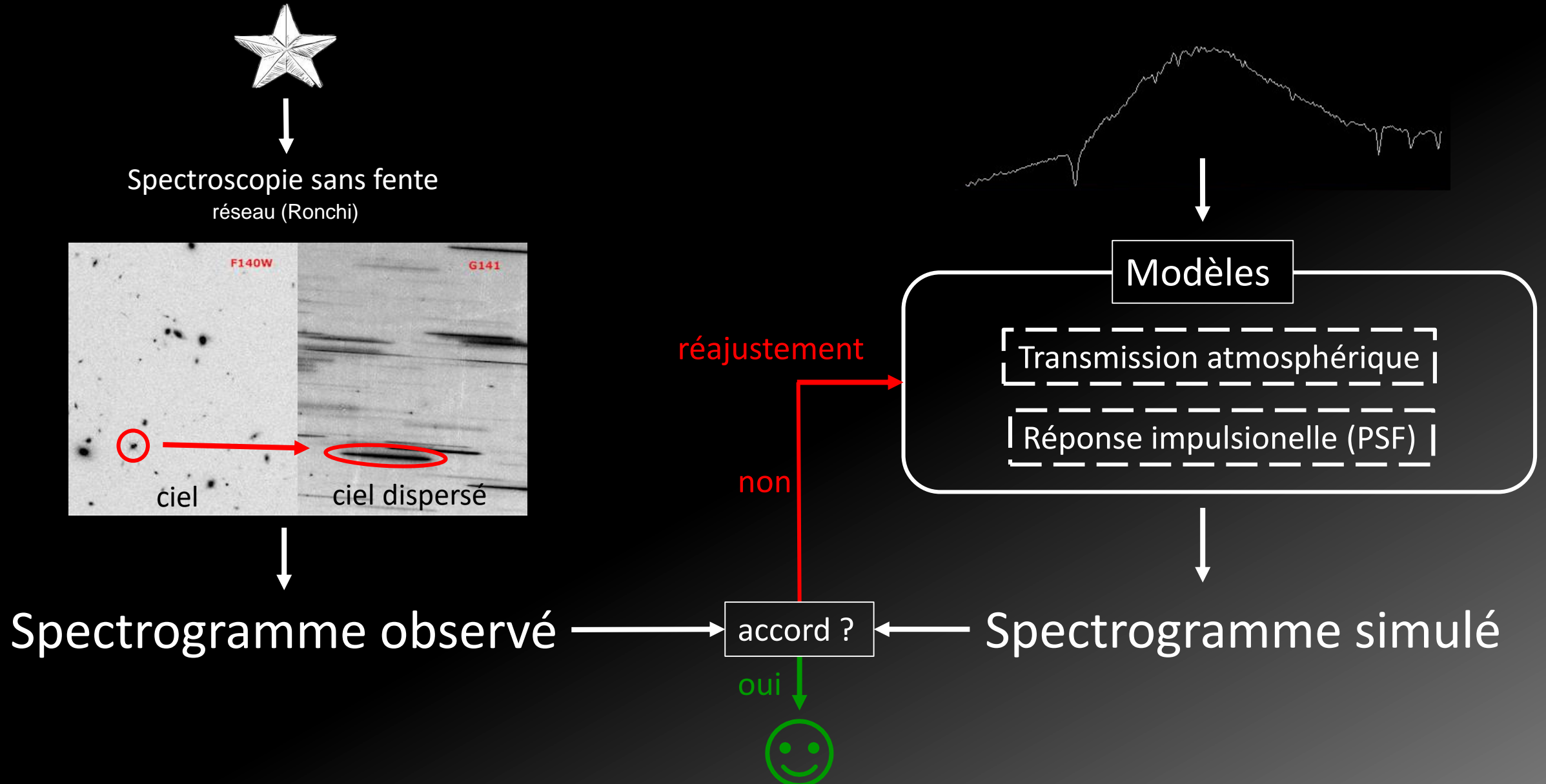


=



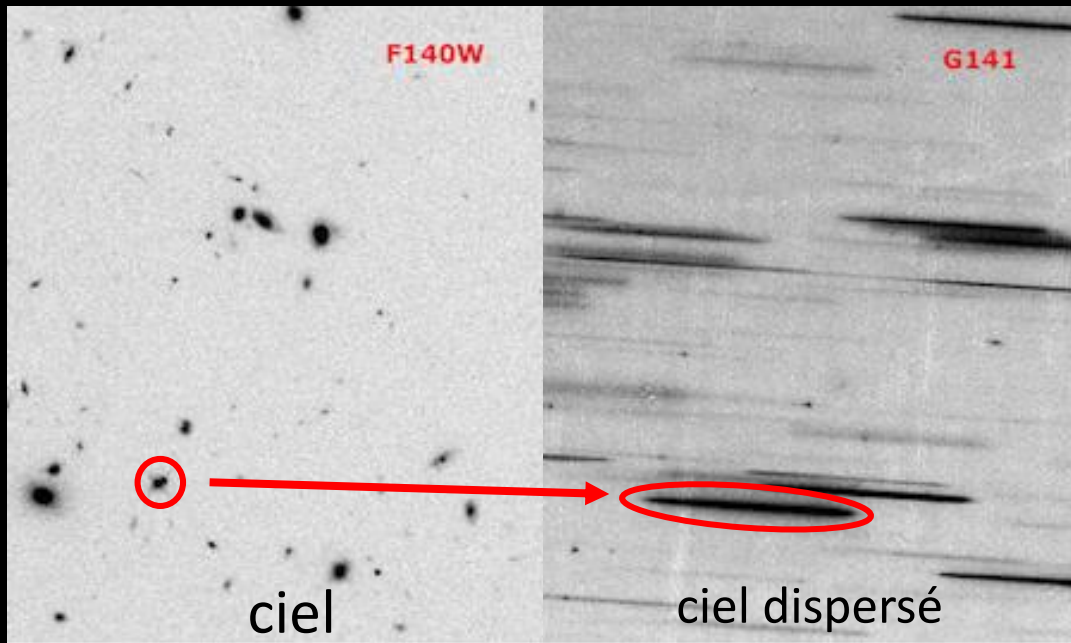
Diffusion rayleigh
Vapeur d'eau
Aérosols
Ozone

Modélisation end-to-end et forward



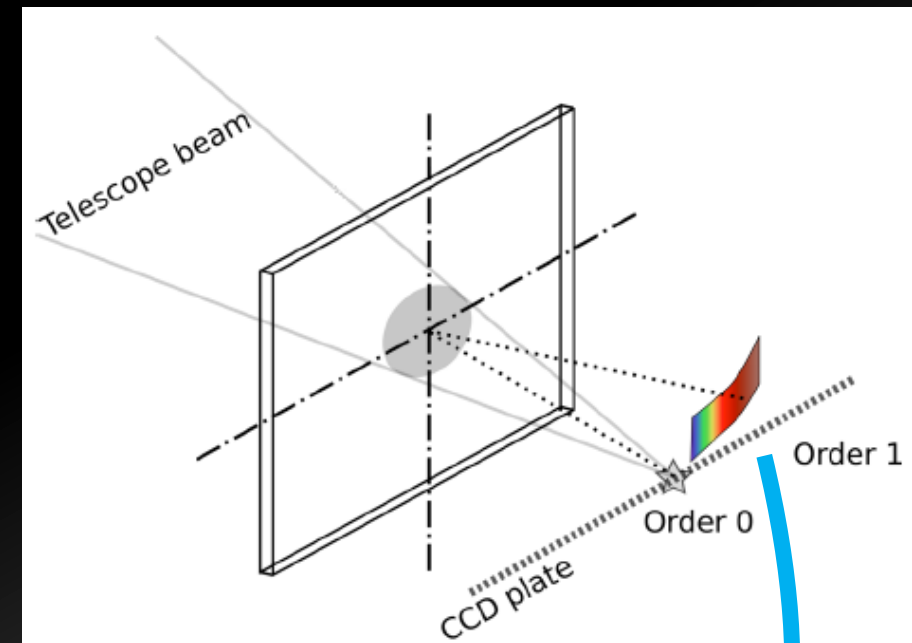
Obtention du spectre

Spectroscopie sans fente



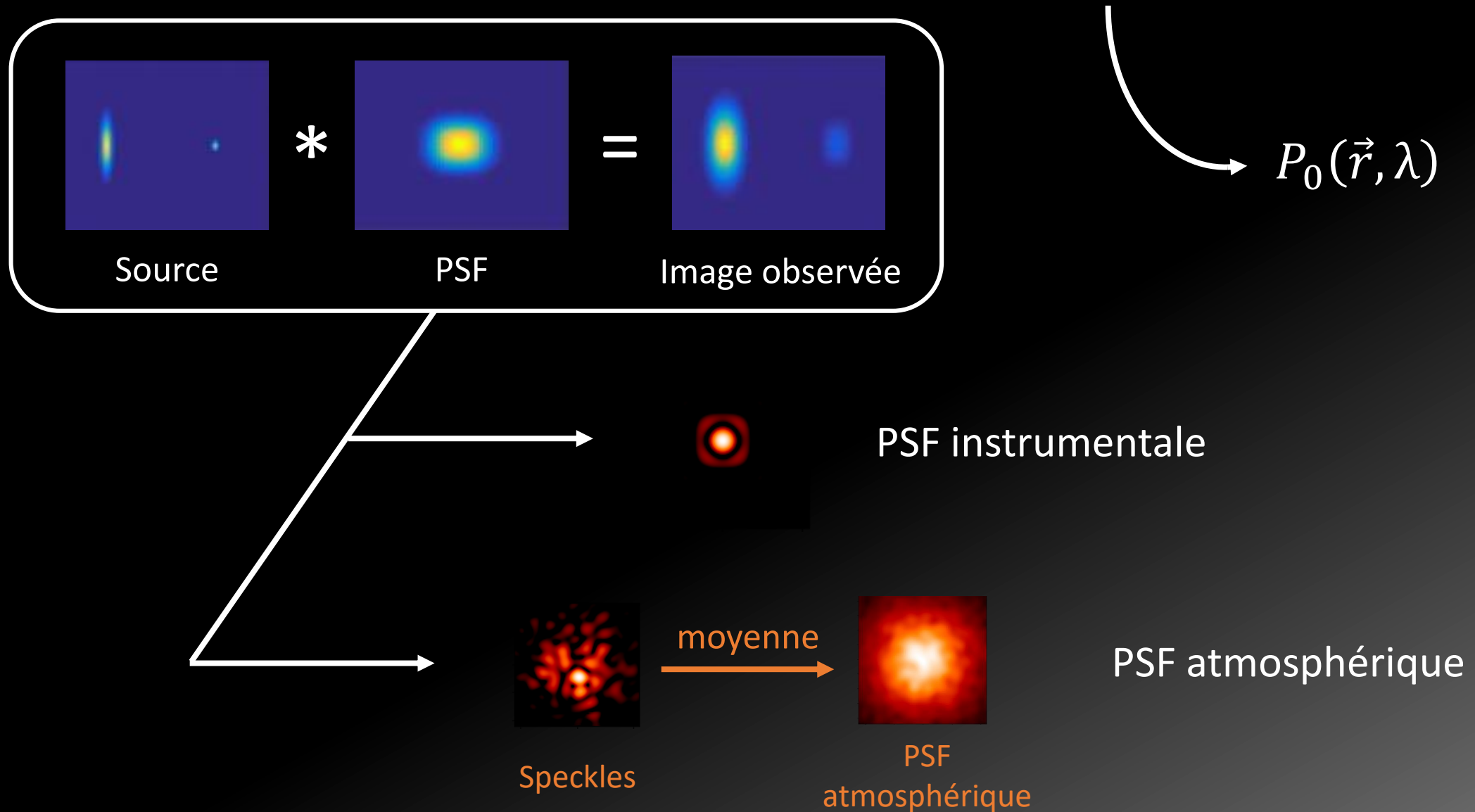
Loi de dispersion $\Delta(\lambda)$

Réseau (Ronchi)



défocalisation

Réponse impulsionnelle (PSF)

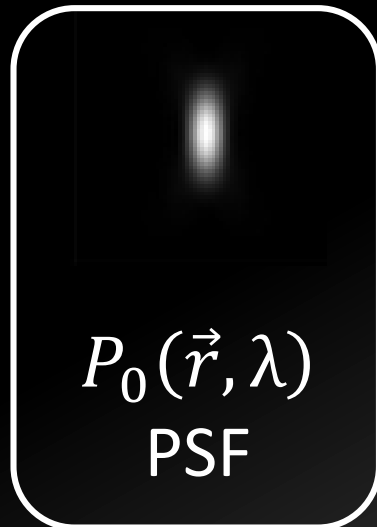


Modélisation mathématique

$$I(\vec{r}) \approx FT^{-1} \left(\int \widehat{P}_0(\vec{k}, \lambda) S(\lambda) e^{-2i\pi \vec{k} \vec{\Delta}(\lambda)} d\lambda \right)$$



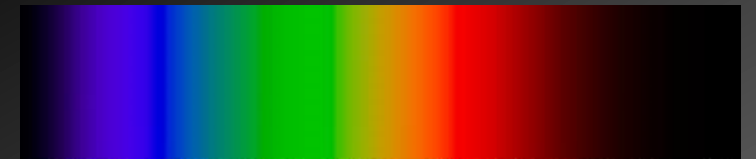
$I(\vec{r})$
Image observée



$P_0(\vec{r}, \lambda)$
PSF



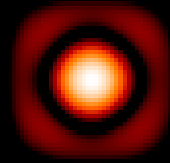
$S(\lambda)$
Spectre



$\Delta(\lambda)$
Loi de dispersion

Effets modélisés

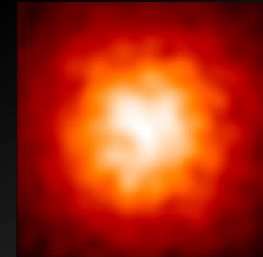
PSF instrumentale



Défocalisation



Turbulences atmosphériques



Réfraction différentielle atmosphérique

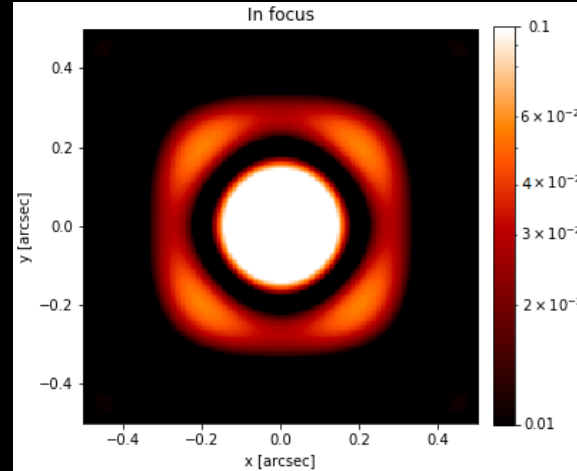
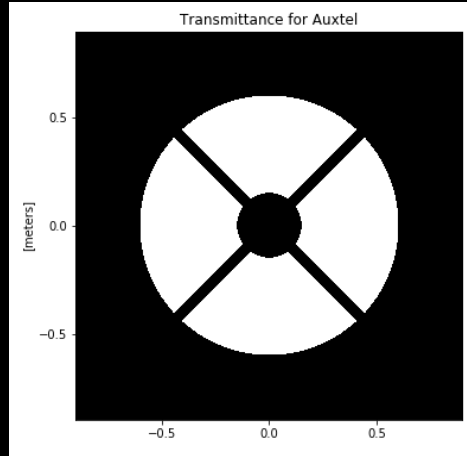
Libraries utilisées:

- HCIPy
- POPPy
- Soapy

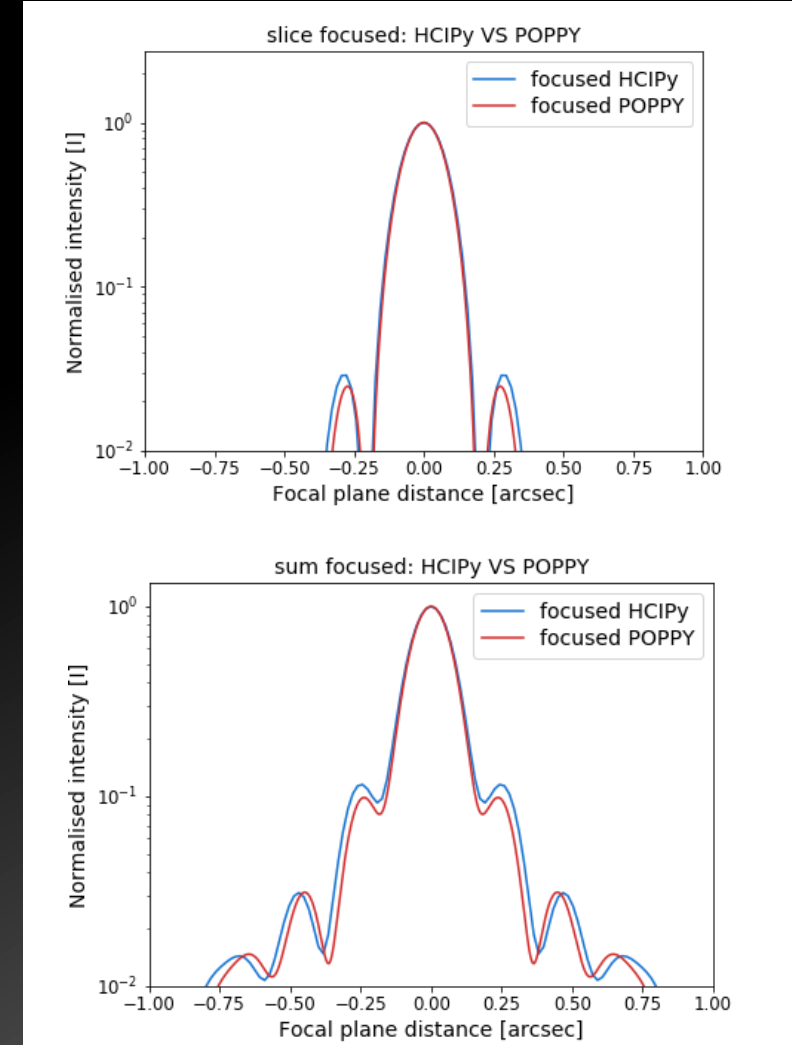
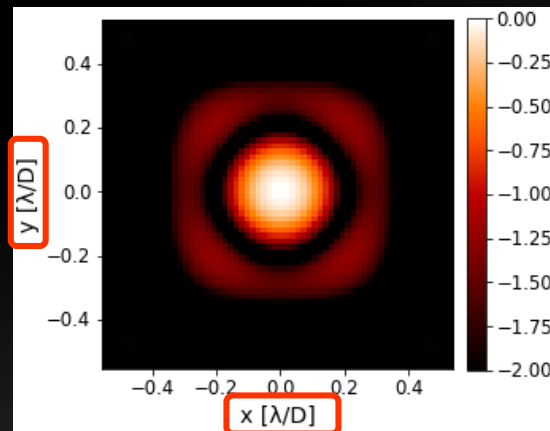
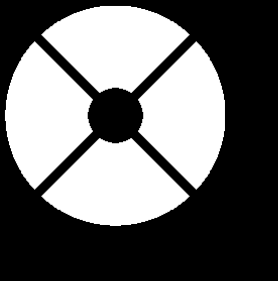


PSF instrumentale

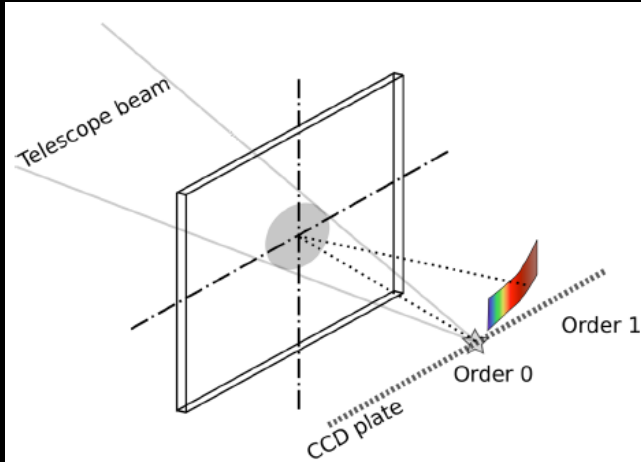
POPPy



HCIPy



Défocalisation



Polynômes de Zernike

- Extraire les valeurs du polynôme de Zernike Z_2^0
- Calculer la différence de phase Φ due à la défocalisation
- Renormaliser Z_2^0 par Φ

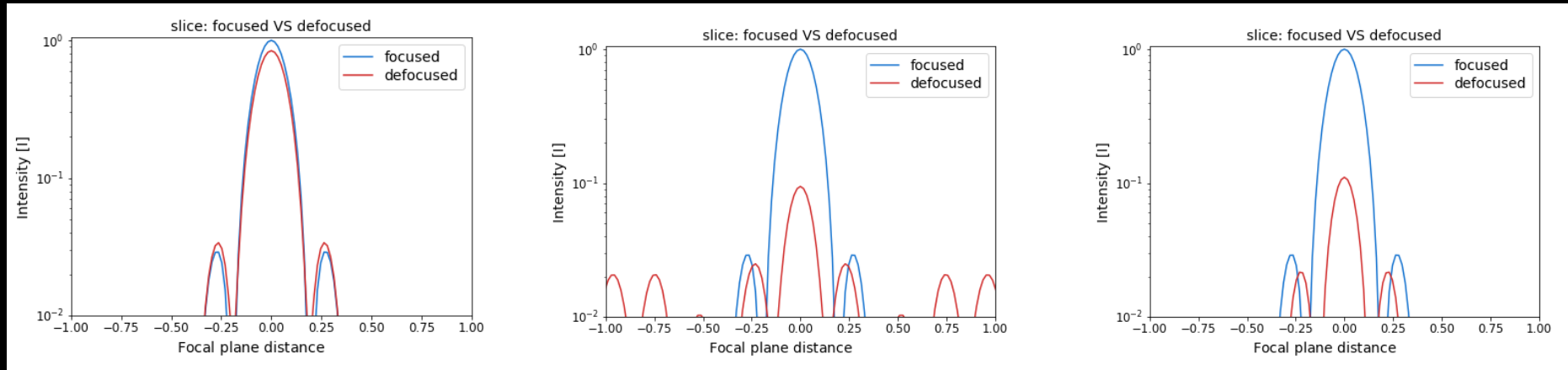
Simulation de l'effet d'une lentille convergente

- Extraire l'équation complexe du front d'onde
- La multiplier par la transmission d'une lentille convergente

$$T_0(f) = e^{i\pi \frac{x^2 + y^2}{f}}$$

Défocalisation

HCIPy

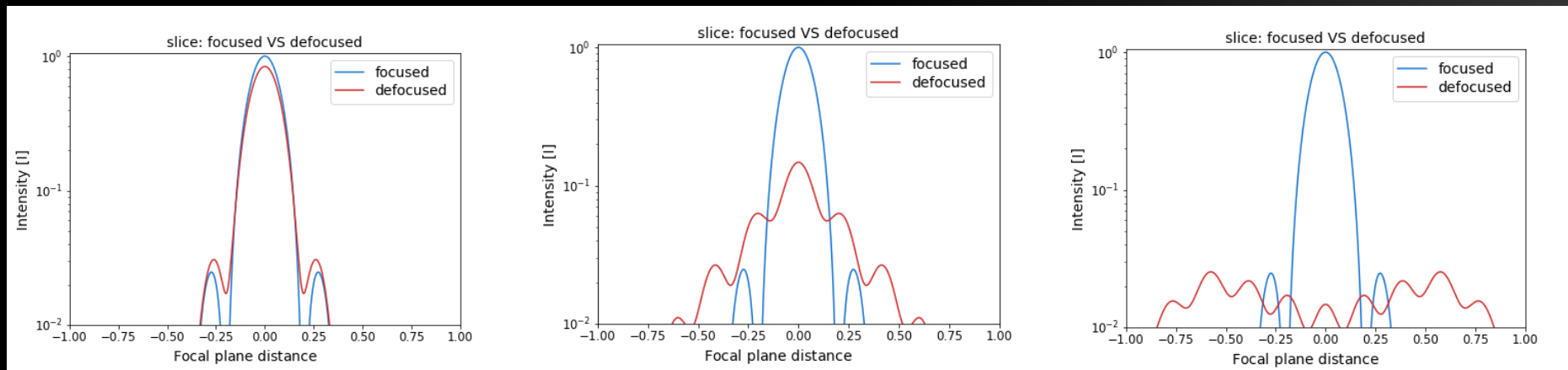


0.8 mm

3 mm

11 mm

POPPy



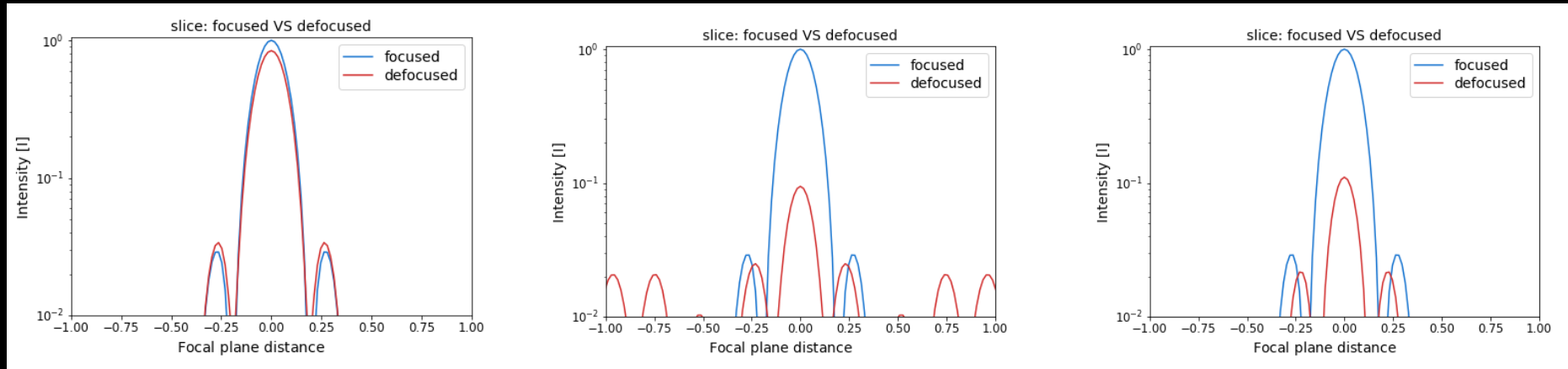
0.1 mm

0.3 mm

0.5 mm

Défocalisation

HCIPy

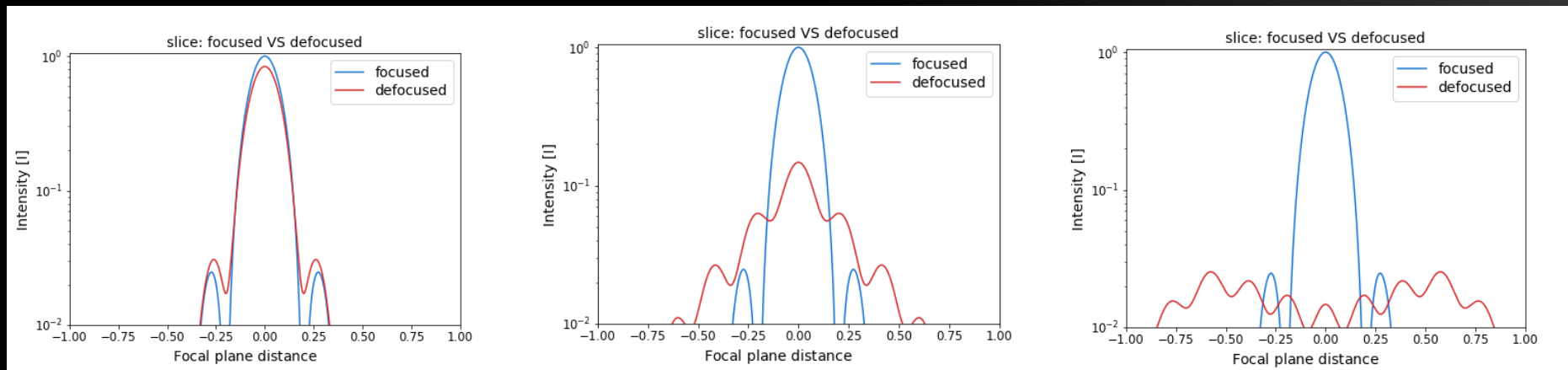


0.8 mm

3 mm

11 mm

POPPy



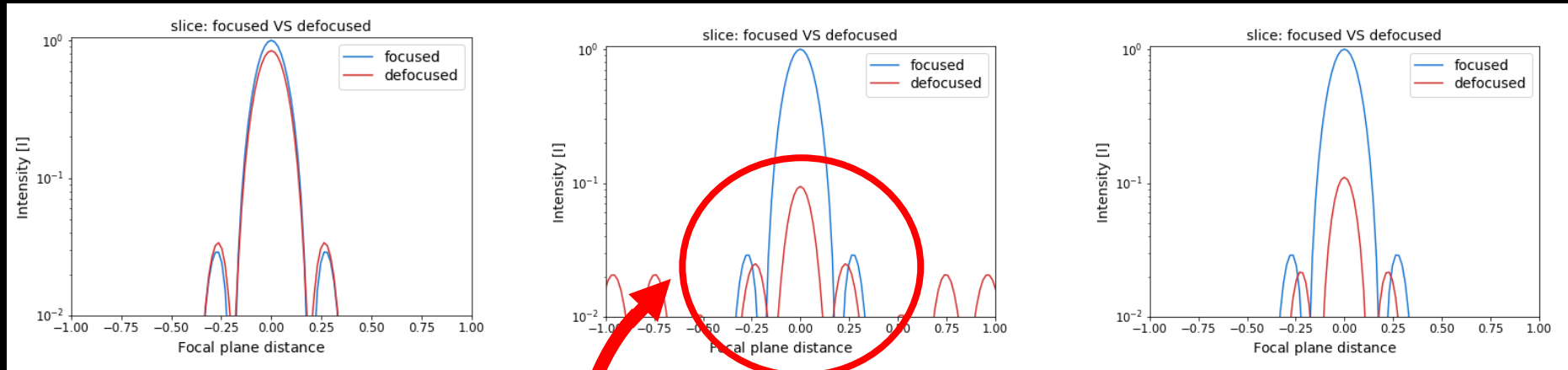
0.1 mm

0.3 mm

0.5 mm

Défocalisation

HCIPy

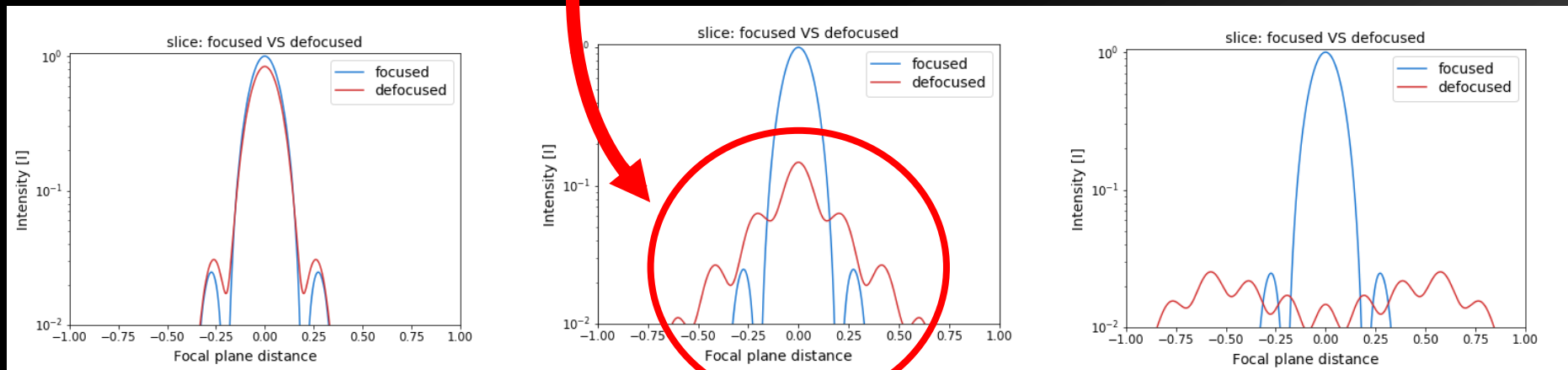


0.8 mm

3 mm

11 mm

POPPy



0.1 mm

0.3 mm

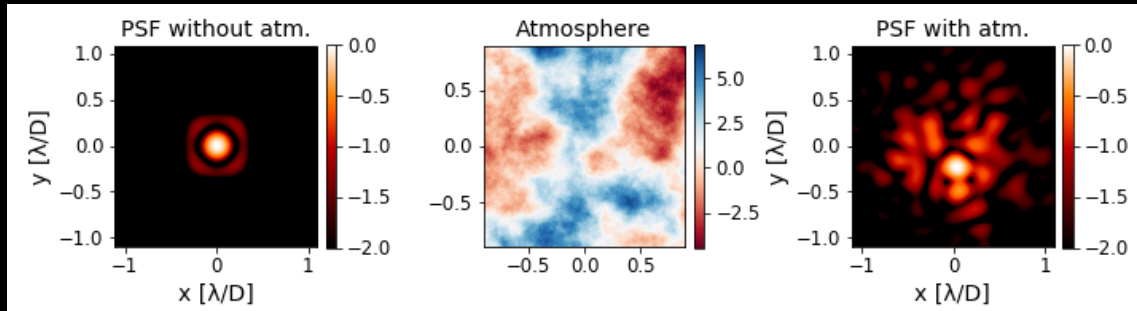
0.5 mm

Turbulences atmosphériques (seeing)

HCIPy

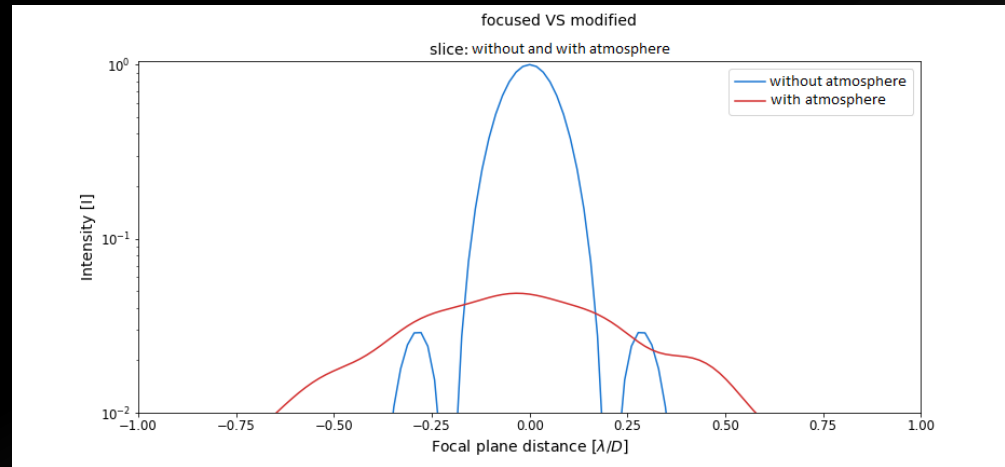
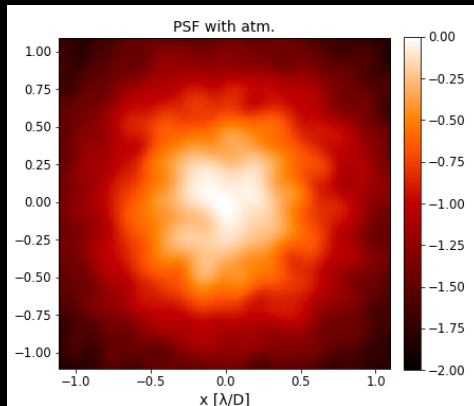
Moyenne

Une réalisation: speckles



$$\lambda = 5000 \text{ \AA}$$
$$r_0 = 10 \text{ cm}$$

seeing ≈ 1



Temps de calcul
15 minutes

Soapy

similaire

Turbulences atmosphériques (seeing)

Equation de structure de phase de Kolmogorov:

$$D_{\Phi}(\vec{r}) = 2 \left[\frac{24}{5} \Gamma\left(\frac{6}{5}\right) \right]^{\frac{5}{6}} \left(\frac{|\vec{r}|}{r_0} \right)^{\frac{5}{3}}$$

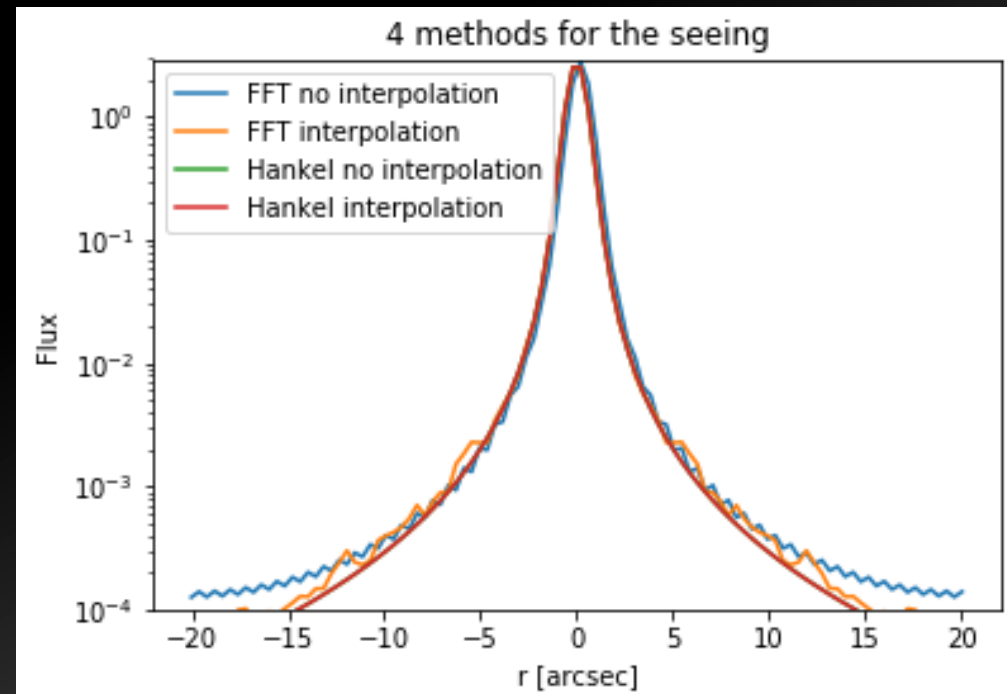
Paramètre de Fried

Fonction de transfert optique: $T_0(f) = e^{-\frac{1}{2}D_{\Phi}(\lambda f)}$

$$PSF = TF(T_0(f))$$

Si symétrie cylindrique:

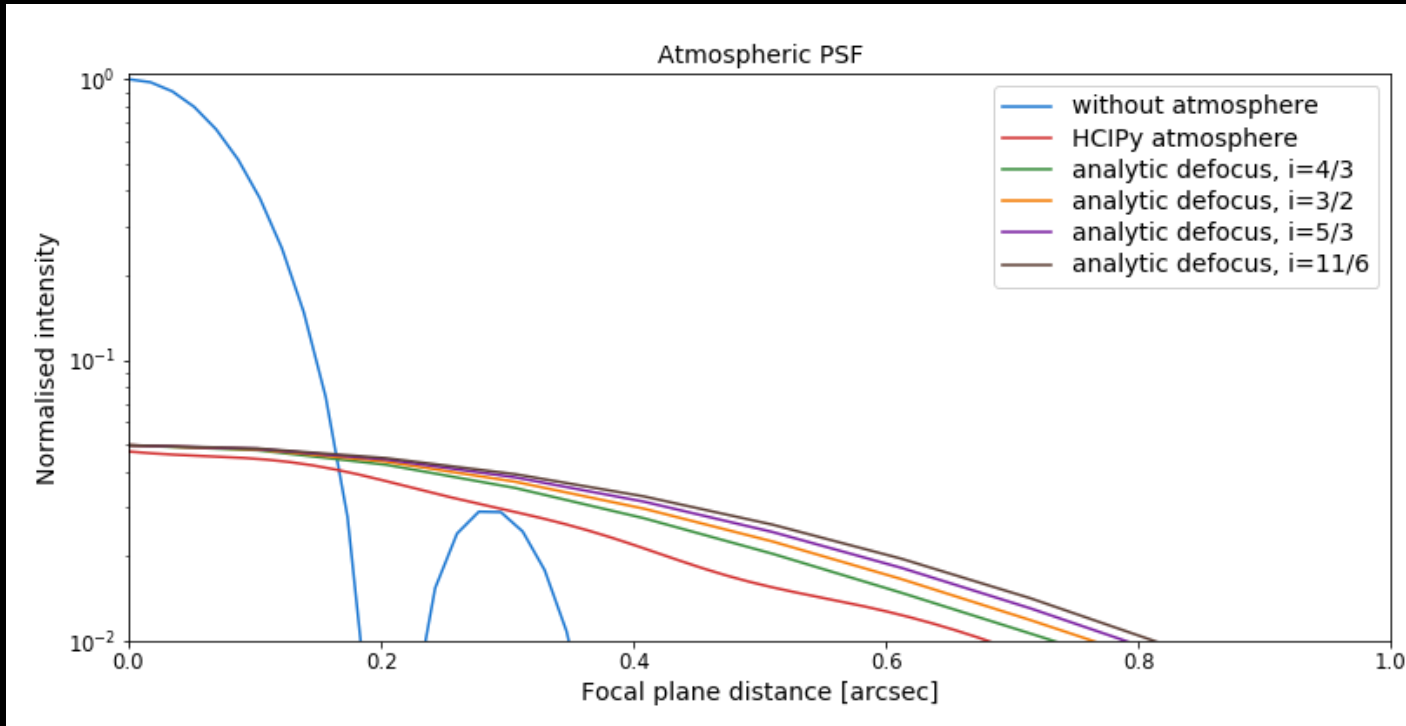
$$PSF = TH_0(T_0(f))$$



seeing ≈ 1

Turbulences atmosphériques (seeing)

Comparaison avec HCIPy



$parallactic = -60.02^\circ$
 $airmass = 1.917$

$$\log T_0(f) = - \left[\frac{8}{i} \Gamma \left(\frac{2}{i} \right) \right]^{\frac{i}{2}} \left(\frac{\vec{r}}{r_0} \right)^i$$

Courbes similaires mais écart notable

Causes potentielles:

- moyenne faite sur seulement 100 réalisations
- modèle plus développé utilisé dans HCIPy.

Réfraction différentielle atmosphérique (ADR)

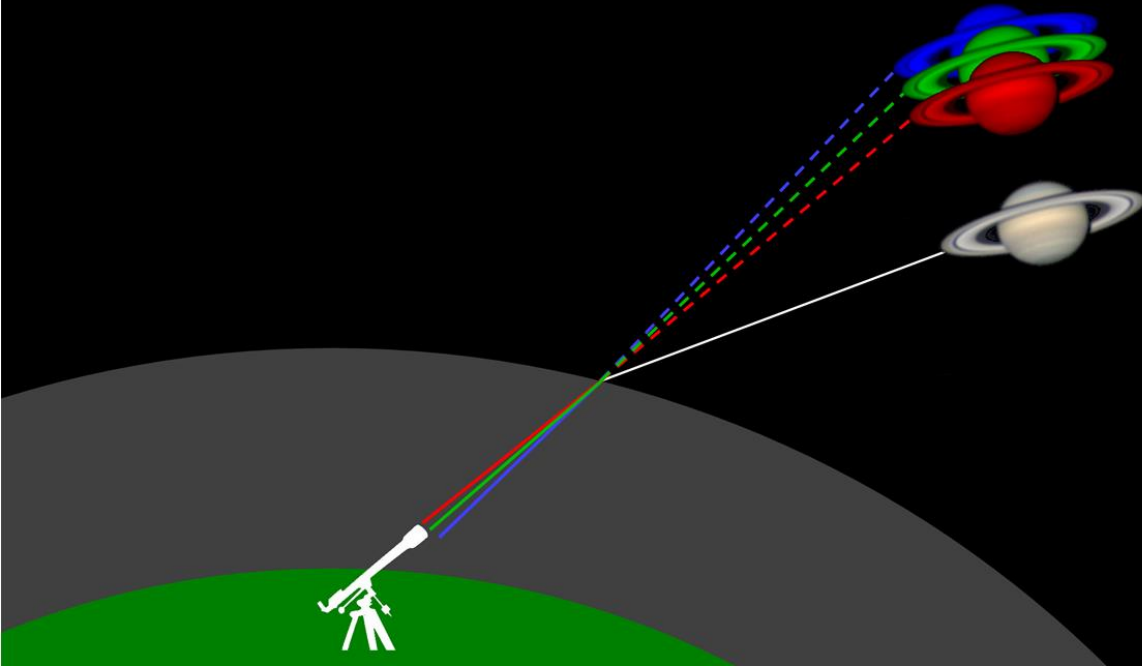
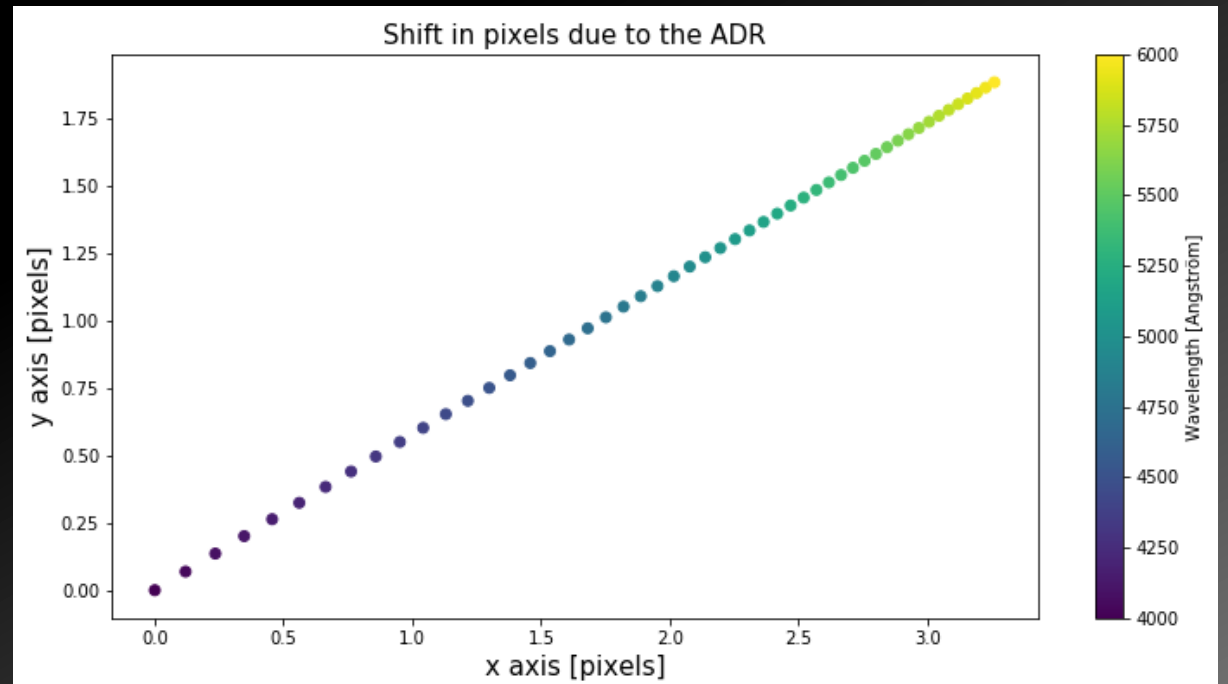
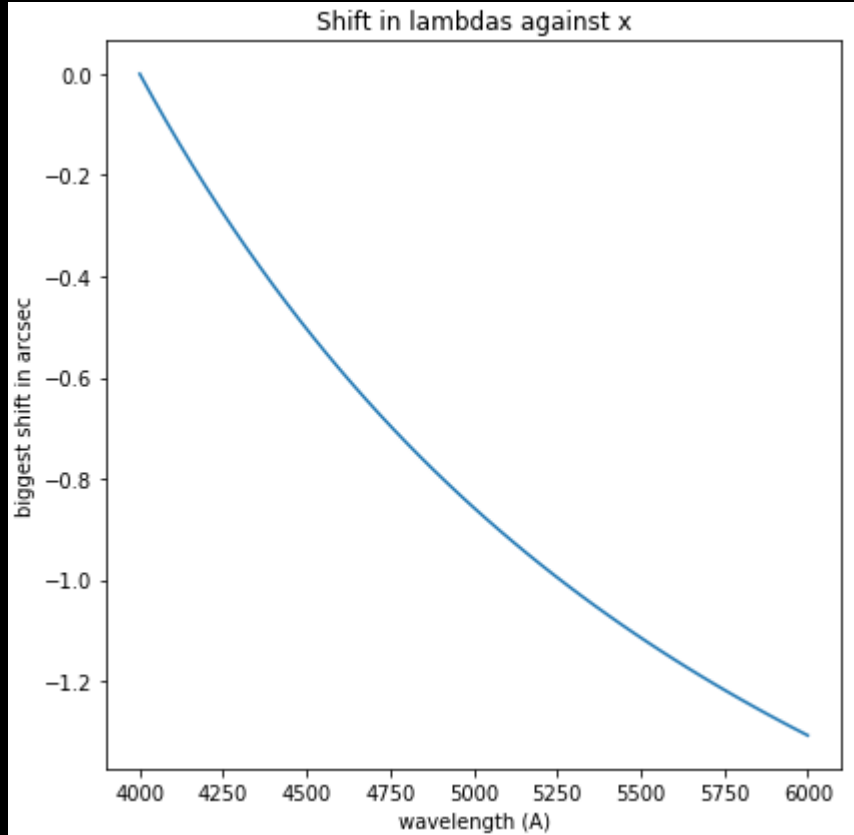


Illustration du phénomène

Effet sur le spectrogramme



Réfraction différentielle atmosphérique (ADR)



Effet testé sur des observations

Axis	x (pixels)	y (pixels)
Maximum	3.291	2.630
Moyenne	1.279	0.616
Médiane	1.082	0.633

1 pixel = 0.401 arcsecondes

Code développé
et documenté
pour intégrer cet
effet.

Effet similaire à la loi de dispersion.

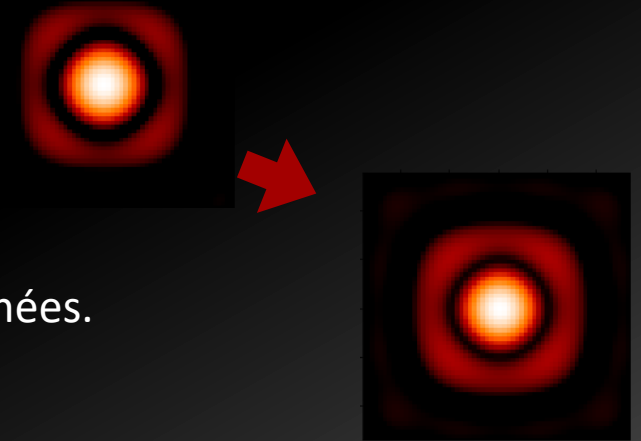
$$\rightarrow \Delta_{total} = \Delta_{ADR} + \Delta_{réseau}$$

En train d'être implémenté
dans Spectractor.

Résumé et conclusion

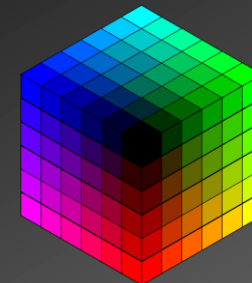
Défocalisation

- Trouver d'où vient la différence d'échelle (HCIPy en $\frac{\lambda}{D}$ et POPPy en arcseconde).
- Vérifier implémentation des polynômes de Zernike avec HCIPy et avec POPPy:
 - Erreur dans un calcul ?
 - Comparer les deux.
- Faire des tests avec une ouverture plus simple: sans obscurisation centrale ni araignées.
- Tenter de simuler l'effet d'une lentille convergente.



Turbulences

- Comparer plus en détail HCIPy et l'utilisation de la fonction de transfert optique:
 - Moyenner les couches d'atmosphère et pas les PSF résultantes
 - Moyenner sur un plus grand nombre de réalisation.
 - Voir comment HCIPy exploite le modèle utilisé.
 - Implémenter le modèle de Von-Karman.
- Globalement satisfaisant



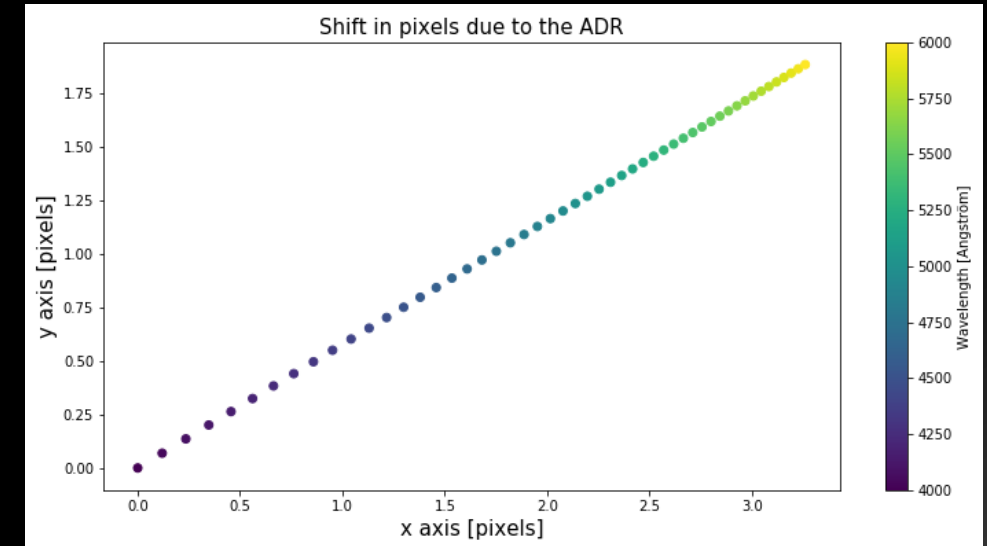
Résumé et conclusion

ADR

- Fonctionne.
- Testée et comparée à des précédents résultats.
- En implémentation

Développement personnel

- Code en Python:
 - Analyse de codes source, utilisation et adaptations de bibliothèques.
 - Développement de scripts documentés et robustes
 - Automatisation des scripts et de leur analyse
- Utilisation de Git:
 - “Versionnage” des codes
 - Signalement et corrections d’erreurs



```
193 def make_psf(final_ap, focal, wvlgh, rad_tel, pixsize, fov, shift=0, showimg=True, logscale=True, showsteps=False):
194
195
196     osys = pop.OpticalSystem()
197     osys.add_pupil(final_ap)
198     if shift!=0:
199         nbwaves = compute_nwaves(focal, wvlgh, rad_tel, shift) # Zernike sign convention inversed
200         osys.add_pupil(pop.ThinLens(nwaves=nbwaves, # Defocus in wvlgh
201                                   reference_wavelength=wvlgh, radius=rad_tel))
202     osys.add_detector(pixelscale=pixsize, fov_arcsec=fov) # Add detector to optical system
203
204     if showing:
205         if showsteps:
206             psf = osys.calc_psf(wvlgh, display_intermediates=True)
207         else:
208             psf = osys.calc_psf(wvlgh)
209             one_plot(psf, shift, logscale=logscale, normalised=True)
210     else:
211         psf = osys.calc_psf(wvlgh)
212
213     return psf
```