

Forward modeling to measure atmospheric transmission

LAL: Sylvie Dagoret-Campagne, Marc Moniez, Jérémie Neveu,
Olivier Perdereau

LPNHE: Sébastien Bongard, Augustin Guyonnet, Laurent Le
Guillou, Nicolas Regnault

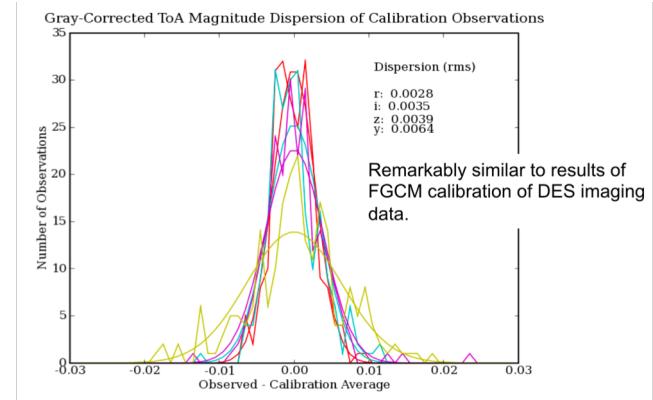
IPNL: Yannick Copin



Context



- LSST: millimag level for photometric uncertainties
- Auxtel: get spectra from photometric standards
 - to constrain atmospheric transmission
 - to propagate to LSST photometry



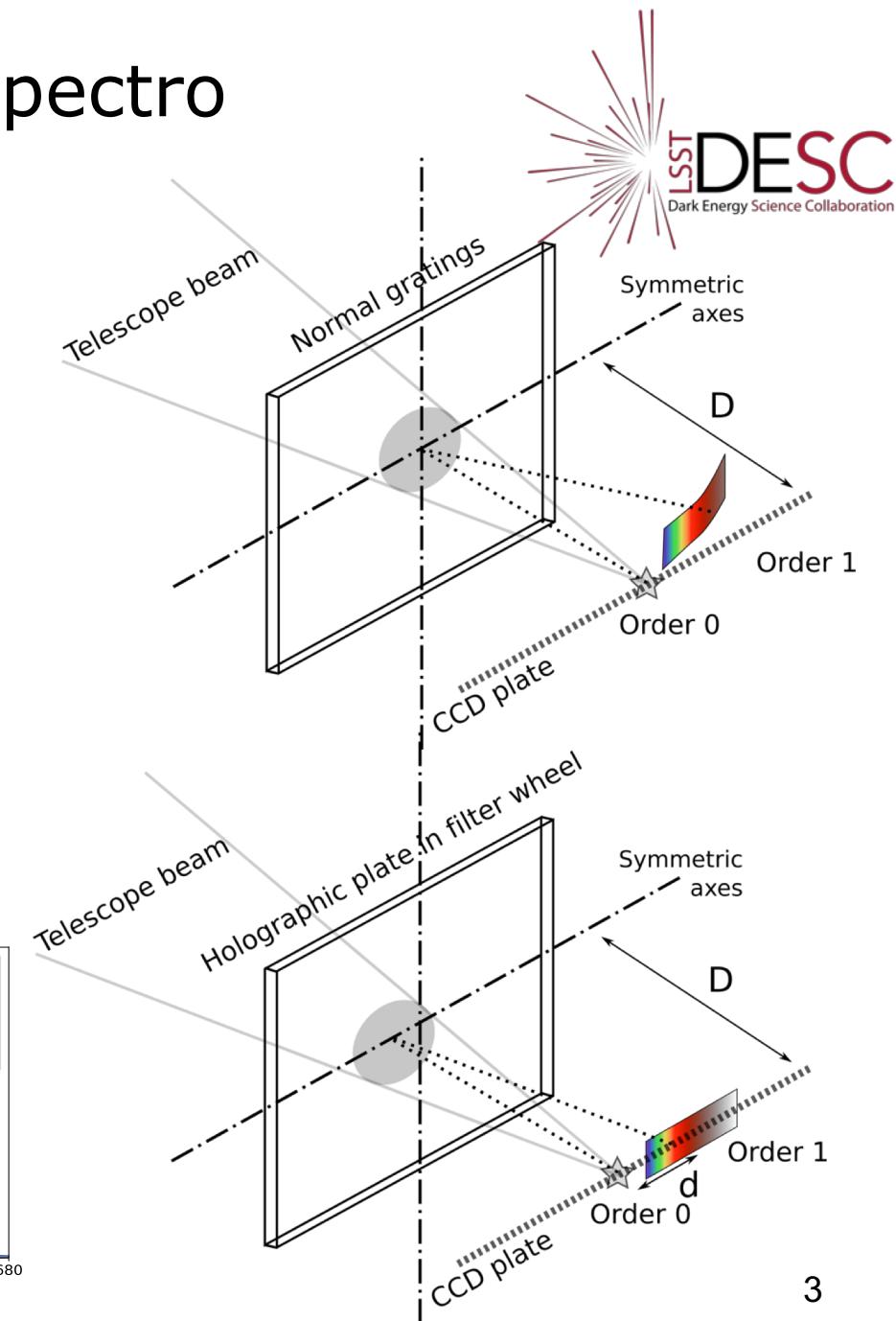
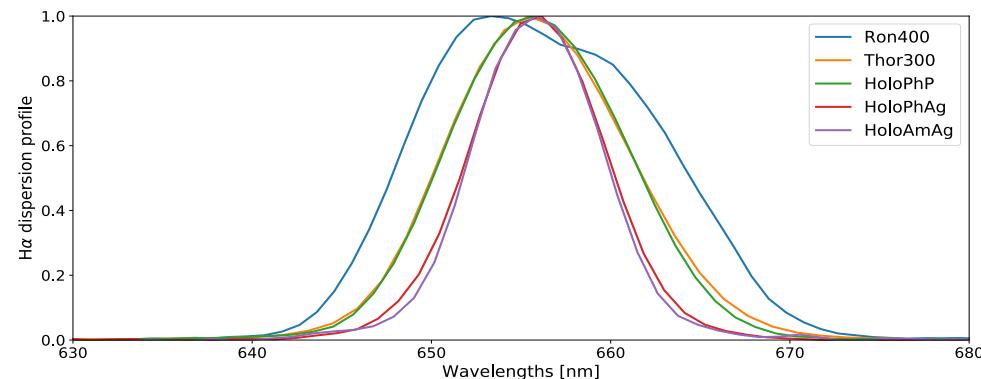
- But Auxtel Spectrograph is slitless and produces defocused spectra
 - What do we call a spectrum in this context ?
 - **Mixing of spatial and spectral information**
 - Y. Copin proposition: go to forward modelling

No spectrum extraction but direct atmospheric transmission measurement

Holograms for slitless spectro

- **Usual gratings:** all wavelengths not focused simultaneously on the focal plane because used with a convergent beam
- **Holograms:** forced focus on the focal plane at almost all wavelengths → hardware solution for the focus

H α filter profile



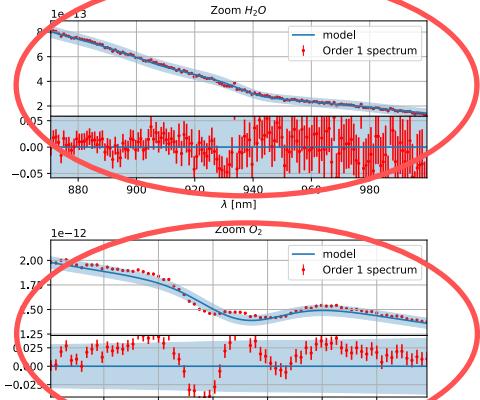
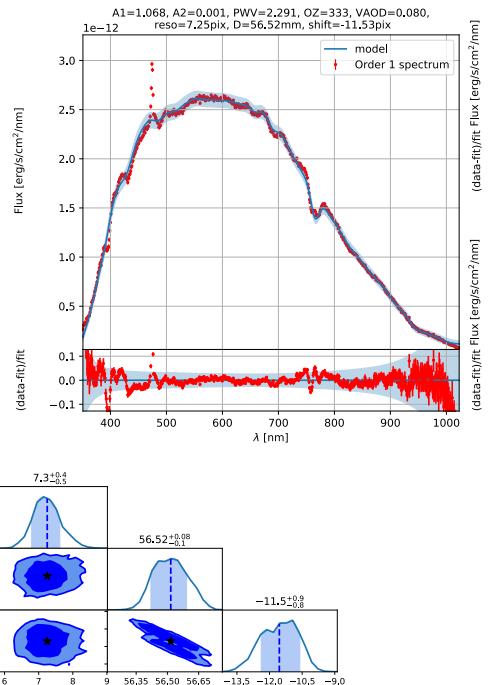
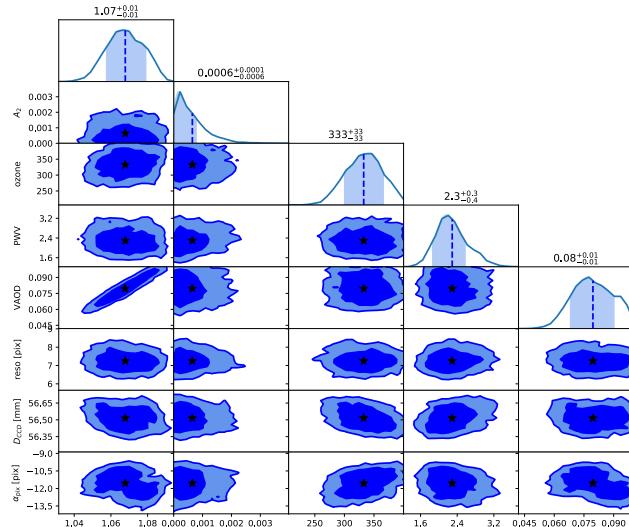
Traditional approach: cross spectrum



- Cross spectrum = sum across the transverse direction of the dispersion axis

Ronchi 400: #130 2017/05/31 02h45 UTC

Target:
HD111980



H₂O information
washed out

O₂ line
defocused

Dispersed imaging

- Slitless spectroscopy

- ◆ P_0 = Point/Line Spread Function
- ◆ $P_\Delta(\mathbf{r}, \lambda) = \delta(\mathbf{r} - \Delta(\lambda))$ where $\Delta(\lambda)$ is the dispersion law

- ◆ Dispersed image: $I(\mathbf{r}) = \int d\lambda (C \otimes P_0)(\mathbf{r} - \Delta(\lambda), \lambda)$
- ◆ In spatial Fourier domain:

$$\hat{I}(\mathbf{k}) = \int d\lambda \hat{C}(\mathbf{k}, \lambda) \hat{P}_0(\mathbf{k}, \lambda) e^{-i2\pi \mathbf{k} \cdot \Delta(\lambda)}$$

Direct approach
→ demixing

- Source: $C(\mathbf{r}, \lambda) = [T_{\text{instrument}}(\lambda) \times T_{\text{atm}}(\lambda|\theta) \times S_{\text{star}}(\lambda)] \times \delta(\mathbf{r} - \mathbf{r}_{\text{order } 0})$
- Chromatic PSF: $P_0(\lambda)$
- Disperser law: $\Delta(\lambda) = (x_{\text{order } 1}(\lambda), y_{\text{order } 1}(\lambda))$

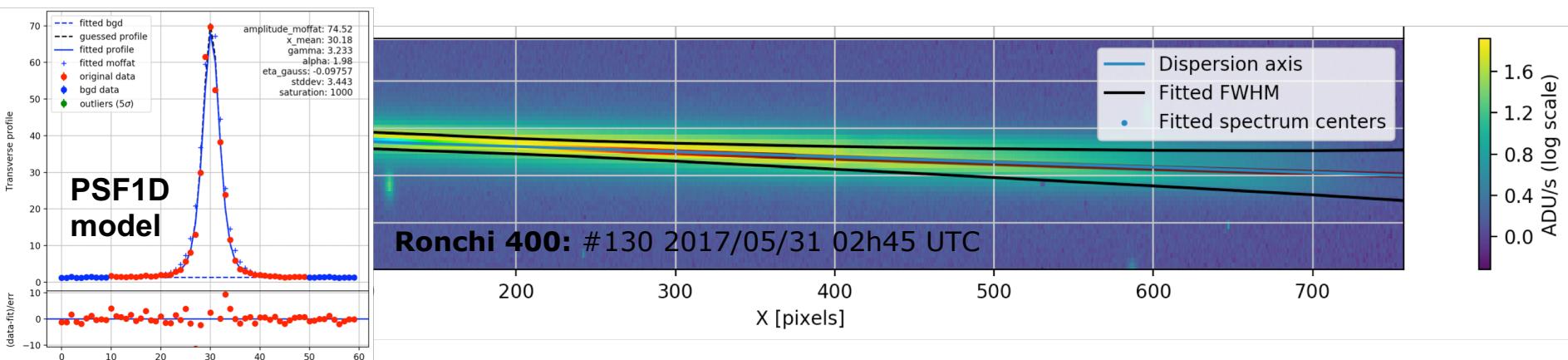
Application to CTIO images

- Determination of $P_0(\lambda)$ and $\Delta(\lambda)$ directly on the spectrograms:

- Find the order 0 and spectro lines to calibrate λ

- Rotate the image to fit spectrogram centers and derotate

$\Delta(\lambda)$



- Fit transverse empirical 1D $\text{PSF}(\lambda) = \text{Moffat}$

- Smooth polynomial evolution of the shape parameters

- To feed $P_0(\lambda)$ 2D PSF with same shape parameters (first guess) 6

Spectrogram simulation

- Use Direct or Fourier formalism to simulate order 1 and order 2

$$I(\mathbf{r}) = \sum_{p=1,2} \text{FFT}^{-1} \left[\int d\lambda C(\lambda) \hat{P}_0(\mathbf{k}, \lambda) e^{-i2\pi\mathbf{k}\cdot\Delta_p(\lambda)} \right]$$

$$I(\mathbf{r}) = \sum_{p=1,2} \int d\lambda C(\lambda) P_0(\mathbf{r} - \Delta_p(\lambda)) \quad C(\lambda) = T_{\text{atm}}(\lambda|\theta) \times T_{\text{inst}}(\lambda) \times S_{\text{star}}(\lambda)$$

- Simulation and fitting procedure implemented in Spectractor
<https://github.com/LSSTDESC/Spectractor>

- Fixed input: $T_{\text{instrument}}(\lambda) \times S_{\text{star}}(\lambda)$

- ~ 23 parameters to fit with $\sim 3e4$ pixels: $\chi^2 = \sum_{i=0}^{\text{all pixels}} \left(\frac{D(\mathbf{r}_i) - I(\mathbf{r}_i)}{\sigma(\mathbf{r}_i)} \right)^2$

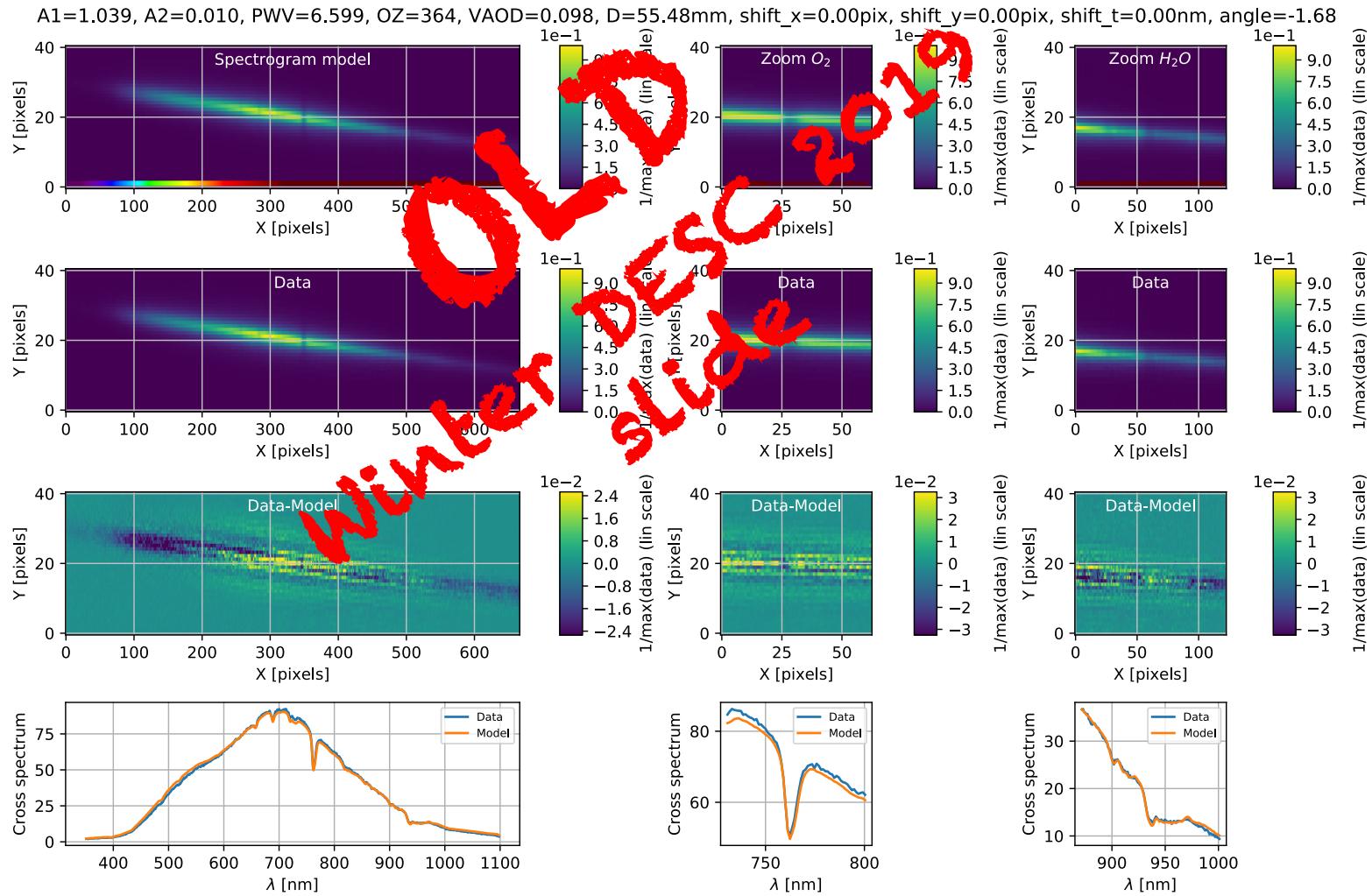
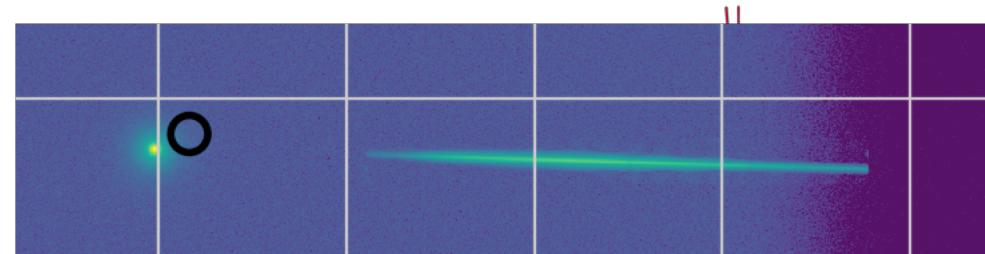
- **3 atmospheric parameters**

- A1: order 1 amplitude, A2: order 2 relative amplitude
 - $\Delta(\lambda)$ and $P_0(\lambda)$ parameters

Data here is a simulation

Fit: test model

NOT CONVERGED!
Minuit have not finished its job

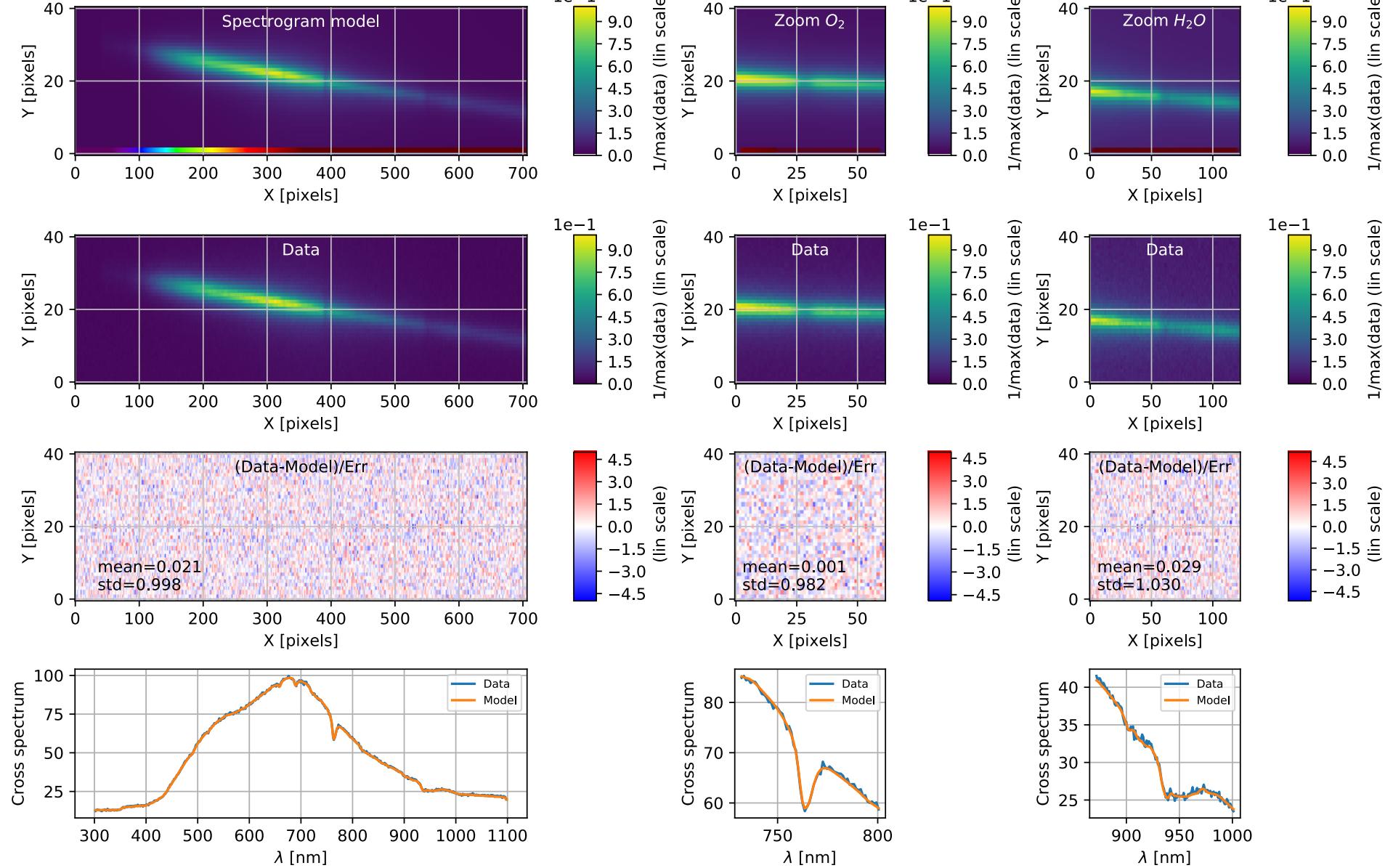


Improvements since February

- Use direct approach
 - Better background subtraction:
 - photutils library with Sextractor algorithm
 - background estimated farther from the spectrogram
 - Change the fitter:
 - minimization of χ^2 the vector (not summed)
 - implementation of a Newton-Raphson method to find the zeros of the gradient of the χ^2
 - Integrate transmissions in $d\lambda$ bins of pixel size
 - Some debugging...
 - Fitting algorithm tuned on several images: get to the rough minimum chisq in ~ 5 iterations and faster (no oversampling of the PSF), then refine in ~ 5 iterations
-
- **Total : 1 Fit in ~15 minutes**
 - Further possibilities: parallelize some computations

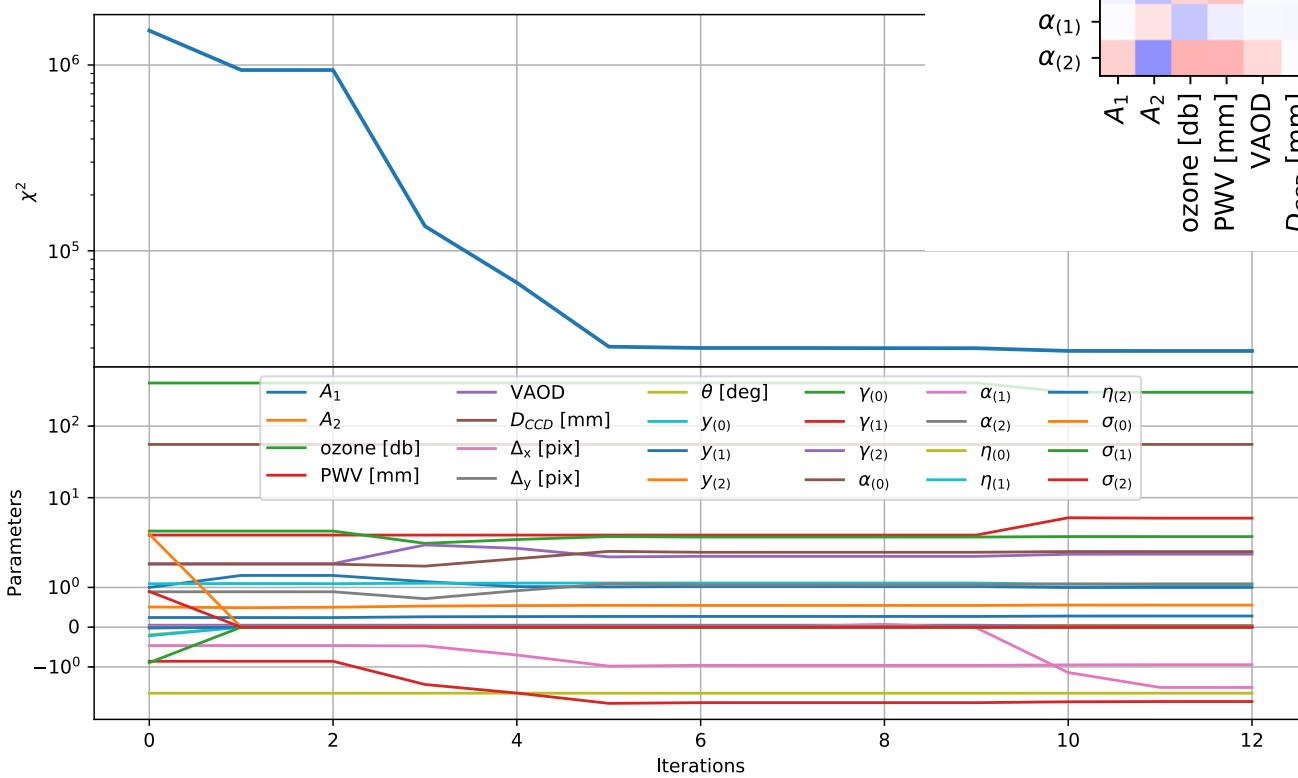
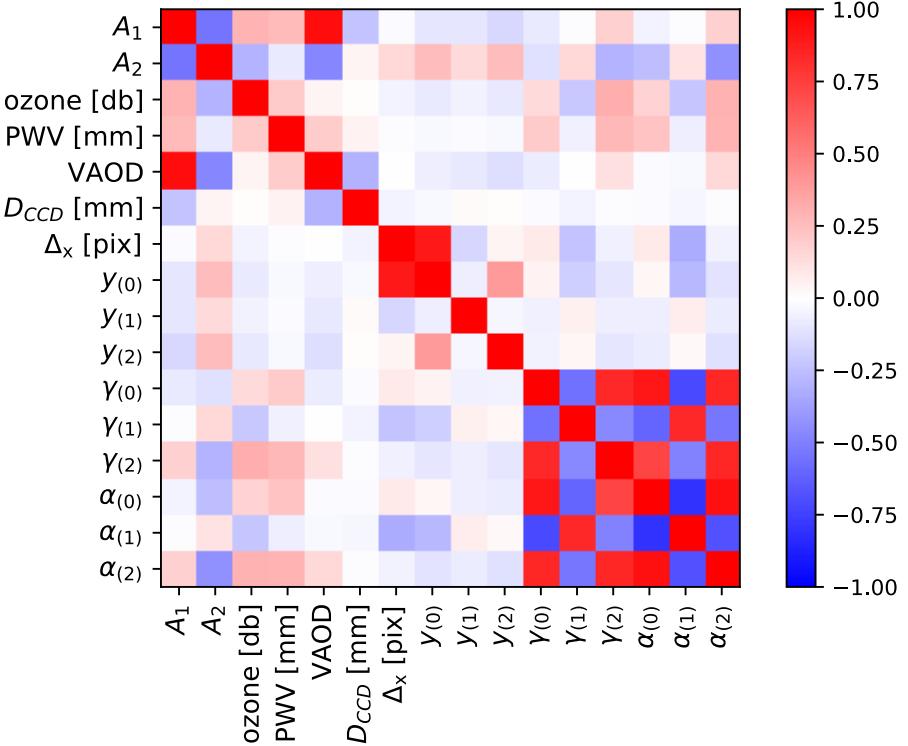
Data here is a simulation

$A_1=1.004, A_2=0.046, PWV=5.132, OZ=301, VAOD=0.035, D=55.46\text{mm}, \text{shift}_x=-1.52\text{pix}, \text{shift}_y=0.00\text{pix}$



Reduced $\chi^2 = 0.996$

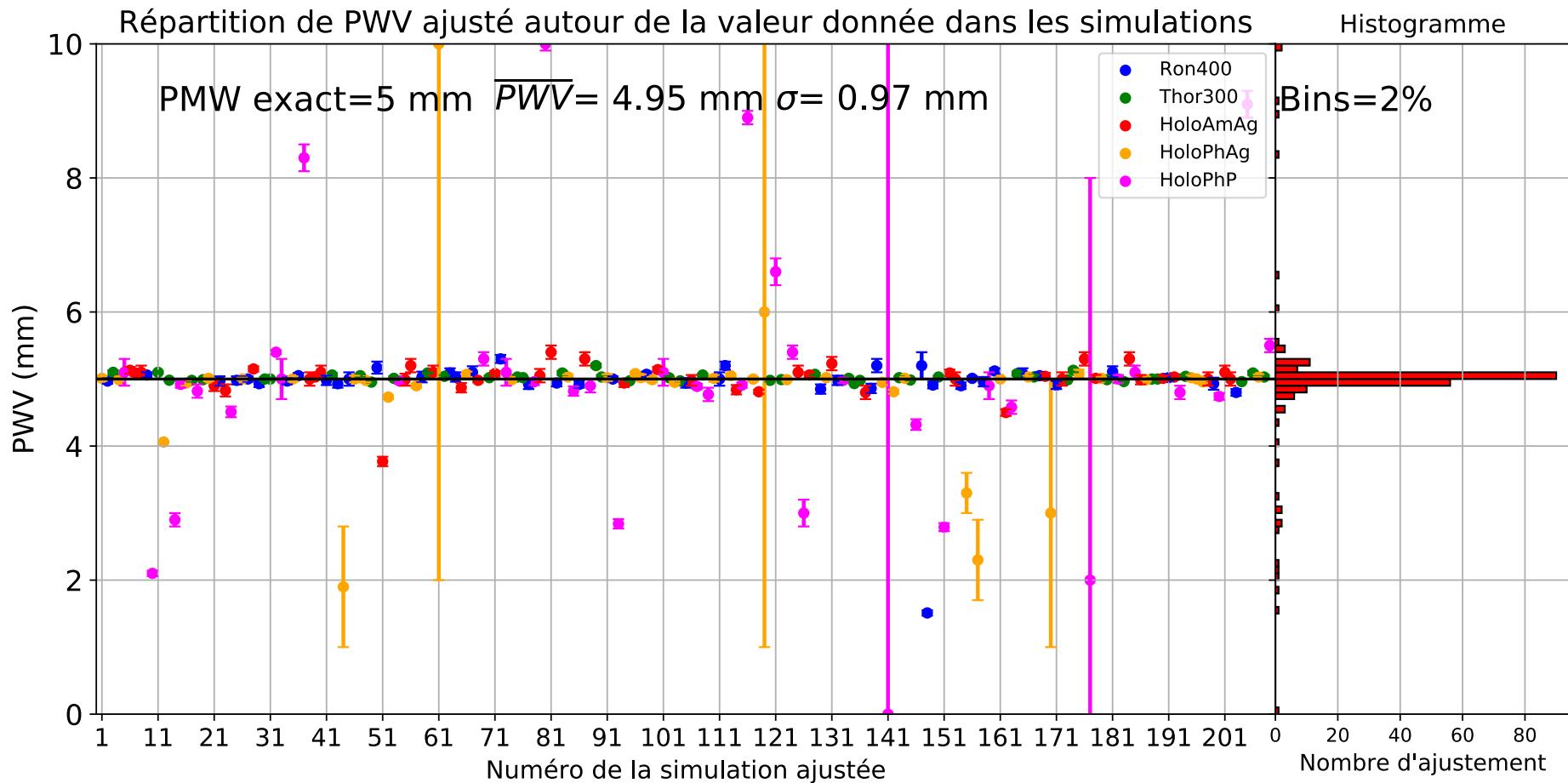
Correlation matrix



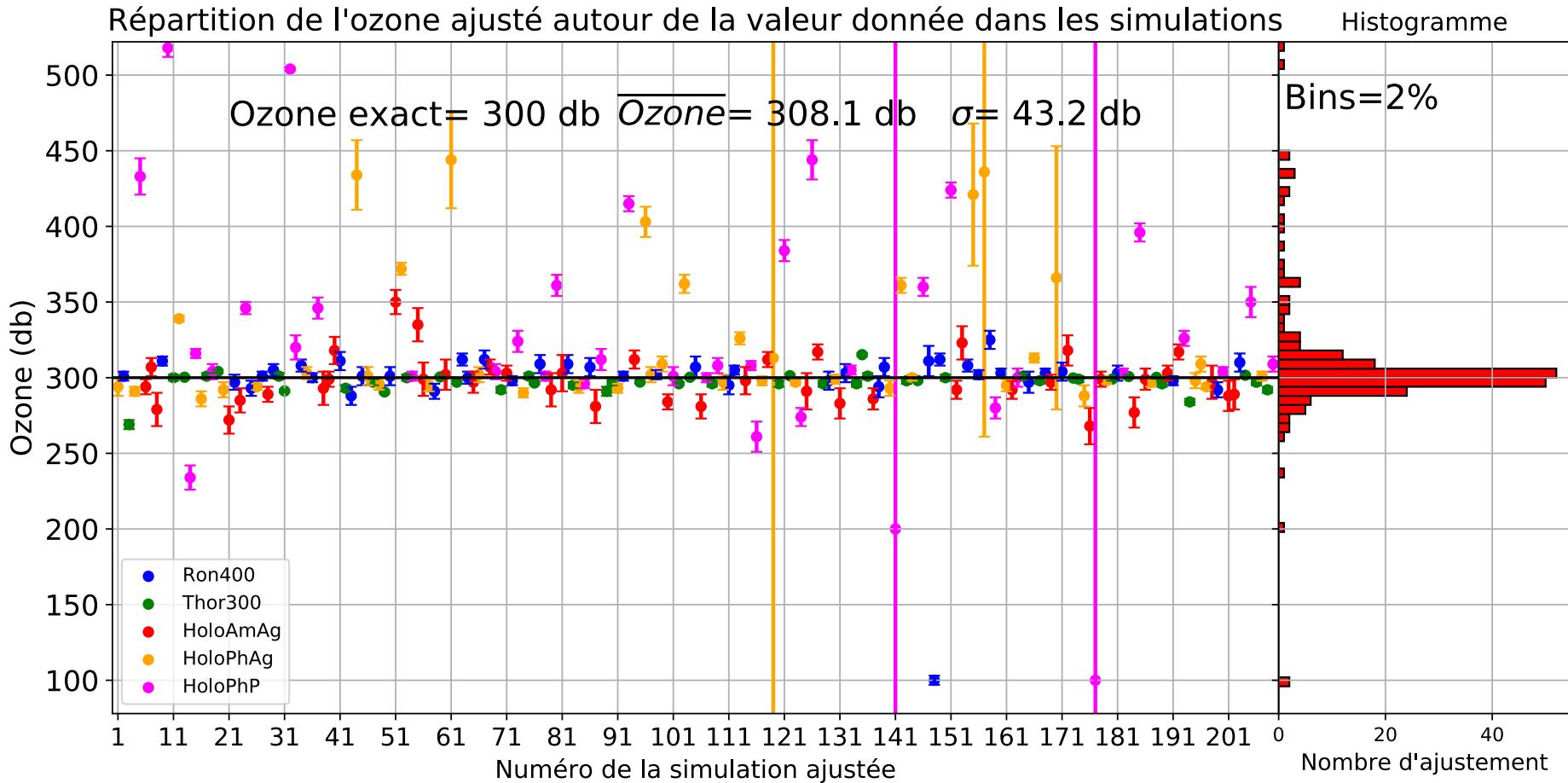
Tests on simulations

- 1 CTIO image → 1 spectrogram simulation with:
 - Same star, same background level
 - Same disperser
 - Same airmass, P, T
 - Same « PSF(λ) » (estimated on the data image)
- But with:
 - PWV = 5mm
 - Ozone = 300 db
 - VAOD = .03

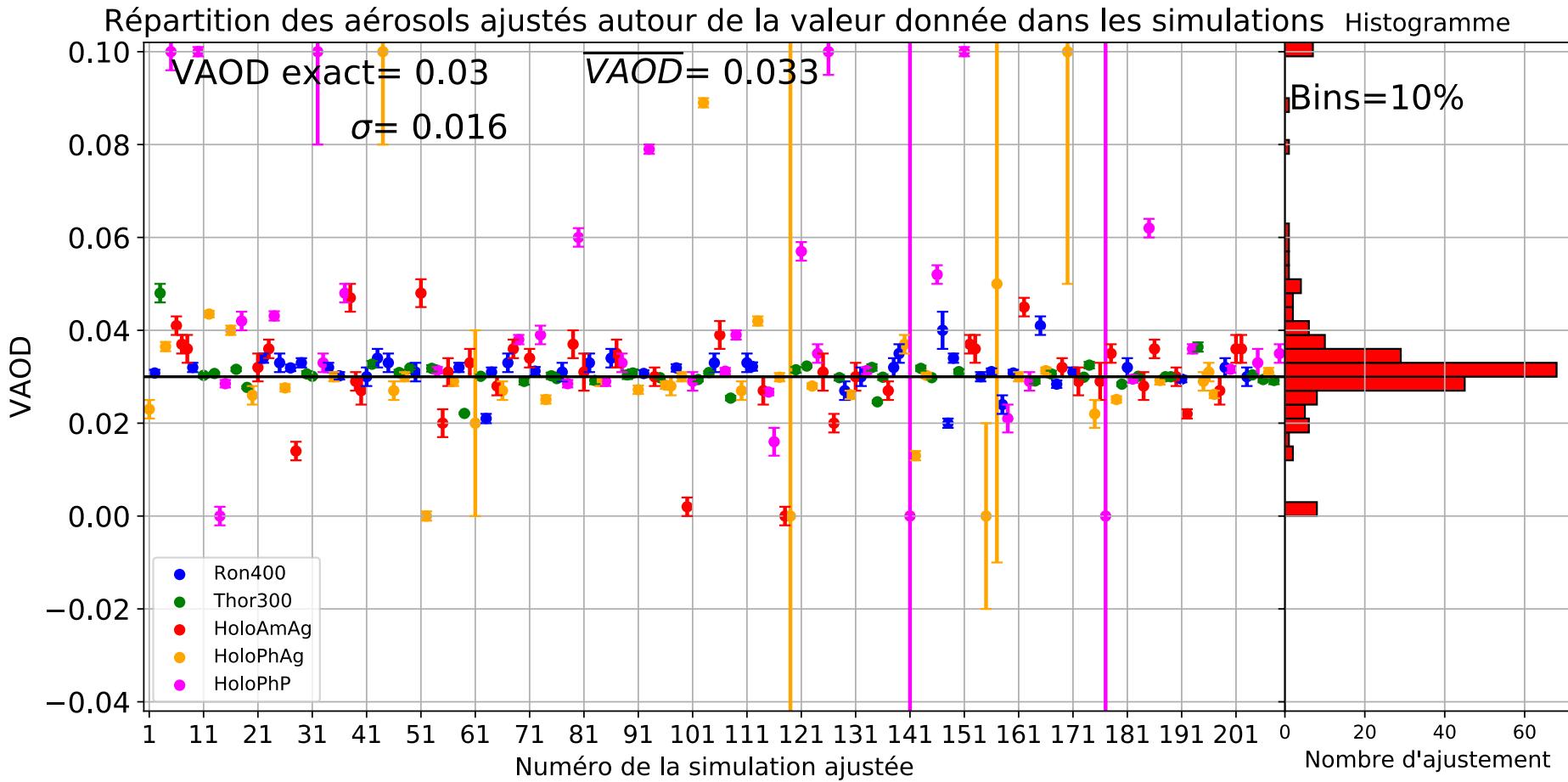
Tests on simulations



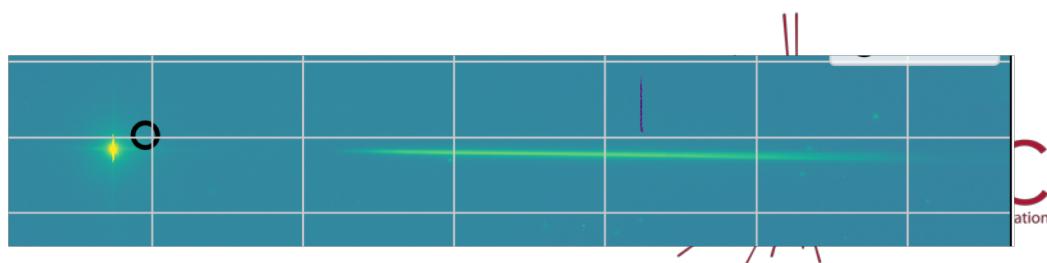
Tests on simulations



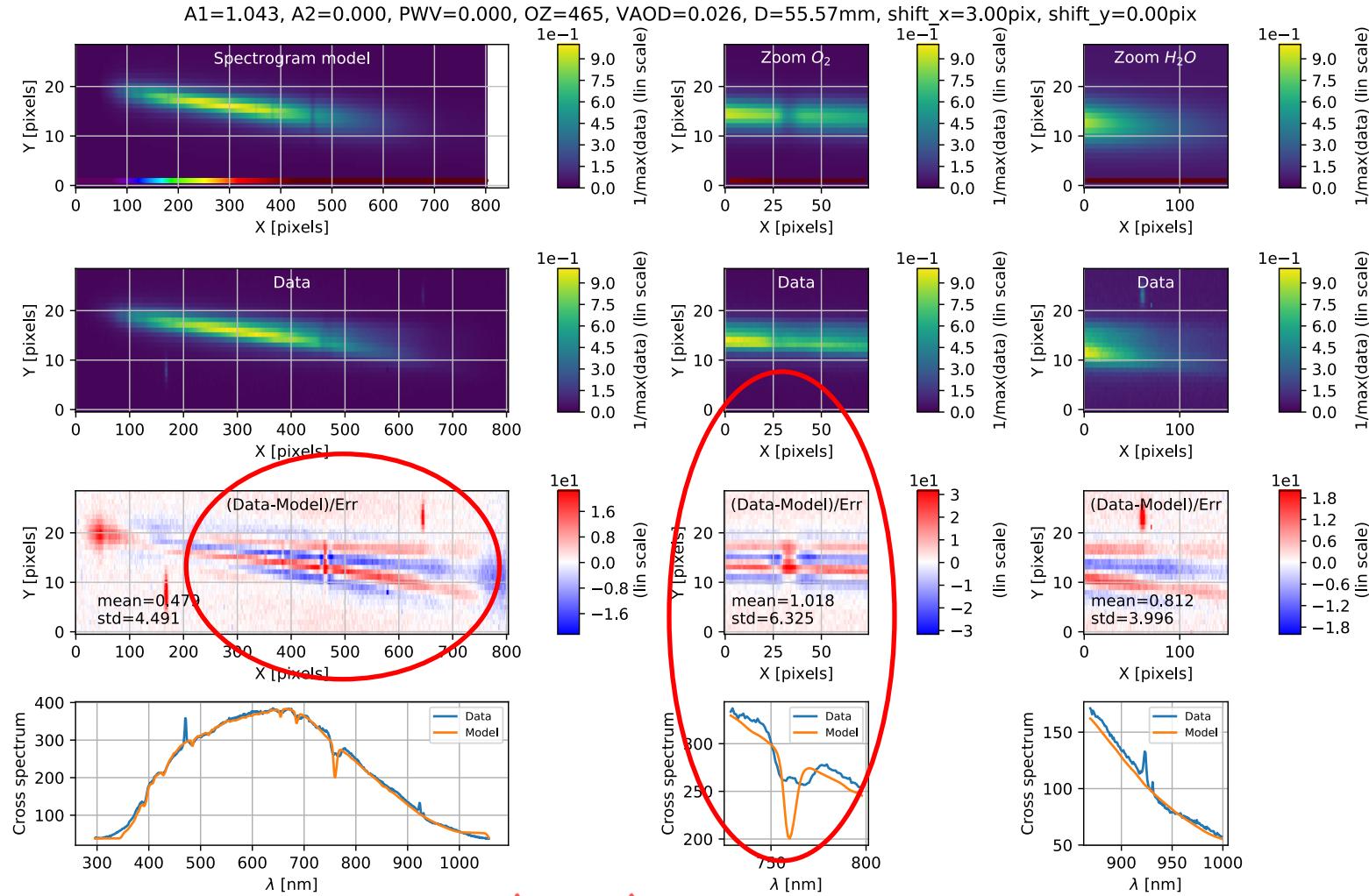
Tests on simulations



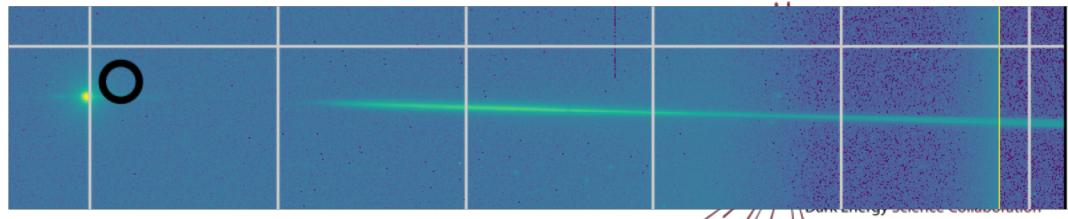
Fit: CTIO Ron#130



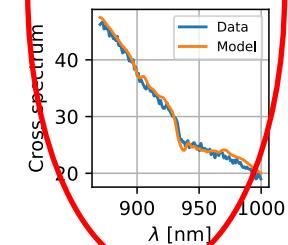
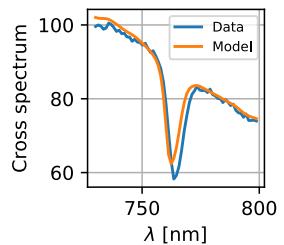
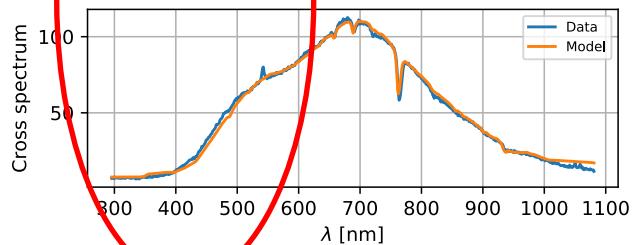
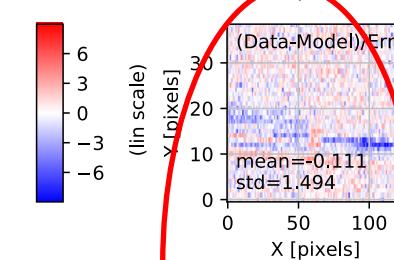
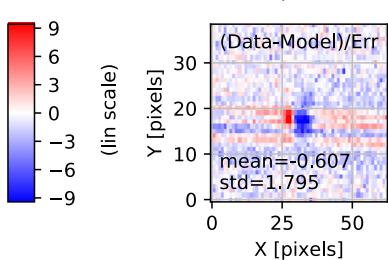
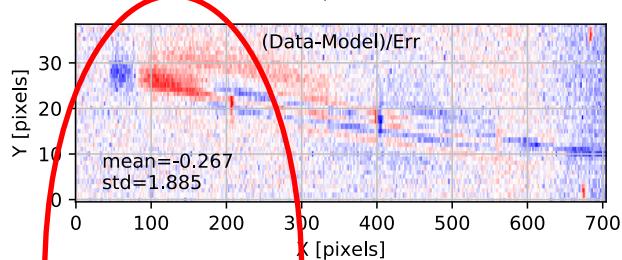
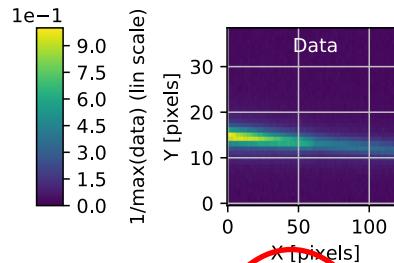
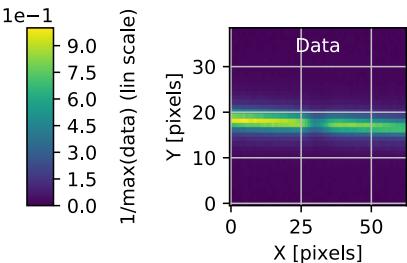
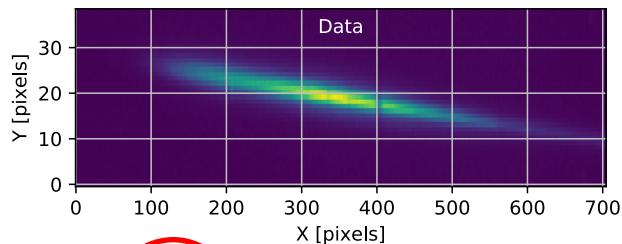
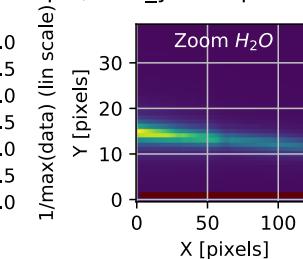
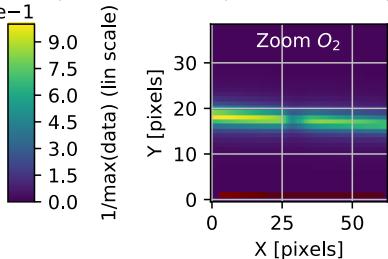
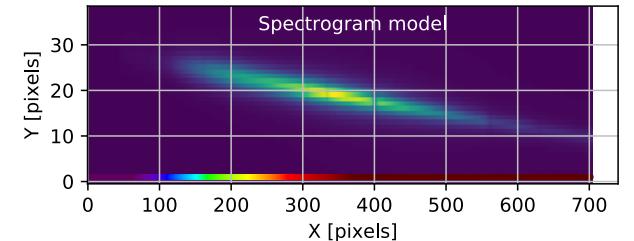
PSF model Moffat-Gauss not sufficient to model the defocus



Fit: CTIO holo#134



$A_1=1.090, A_2=0.000, PWV=2.912, OZ=453, VAOD=0.000, D=56.26\text{mm}, \text{shift}_x=0.15\text{pix}, \text{shift}_y=0.00\text{pix}$



$T_{\text{instrumental}}(\lambda)$
not correct?

Reduced $\chi^2 = 3.6$

Encouraging!

Conclusions and next steps

- The pipeline loop is closed (at least on simulations)
 - Forward model very efficient to recover the atmospheric parameters
- Working on reducing the last residuals/outliers
- Secure the background subtraction
- Implement ADR
- Ability to fit the instrumental transmission using airmass regression
- More realistic model of a (defocused) PSF

On the Moon Again !

- Celebration of the Apollo 11 landing 50 years ago
- Public observation of the Moon, and planets, stars, nebula...

Parc Montsouris
(RER B Cité Universitaire)
From 9pm to 1am



Back-up slides

Presented at Winter DESC meeting



Y. Copin's slide

Dispersed imaging

● Slitless spectroscopy

- ◆ P_0 = Point/Line Spread Function
- ◆ $P_\Delta(\mathbf{r}, \lambda) = \delta(\mathbf{r} - \Delta(\lambda))$ where $\Delta(\lambda)$ is the dispersion law
- ◆ Dispersed image: $I(\mathbf{r}) = \int d\lambda (C \otimes P_0)(\mathbf{r} - \Delta(\lambda), \lambda)$ Direct approach
- ◆ In spatial Fourier domain:
$$\hat{I}(\mathbf{k}) = \int d\lambda \hat{C}(\mathbf{k}, \lambda) \hat{P}_0(\mathbf{k}, \lambda) e^{-i2\pi \mathbf{k} \cdot \Delta(\lambda)}$$
 Fourier approach
→ FFT faster

- Source: $C(\mathbf{r}, \lambda) = [T_{\text{instrument}}(\lambda) \times T_{\text{atm}}(\lambda|\theta) \times S_{\text{star}}(\lambda)] \times \delta(\mathbf{r} - \mathbf{r}_{\text{order } 0})$
- Chromatic PSF: $P_0(\lambda)$
- Disperser law: $\Delta(\lambda) = (x_{\text{order } 1}(\lambda), y_{\text{order } 1}(\lambda))$

Presented at Winter DESC meeting



Spectrogram simulation

- Use Fourier formalism to simulate order 1 and order 2

$$I(\mathbf{r}) = \sum_{p=1,2} \text{FFT}^{-1} \left[\int d\lambda \hat{C}(\mathbf{k}, \lambda) \hat{P}_0(\mathbf{k}) e^{-i2\pi\mathbf{k}\cdot\Delta_p(\lambda)} \right]$$

- Simulation and fitting procedure implemented in [Spectractor](https://github.com/LSSTDESC/Spectractor)
<https://github.com/LSSTDESC/Spectractor>
- Fixed input: $T_{\text{instrument}}(\lambda) \times S_{\text{star}}(\lambda)$
- ~ 30 parameters to fit with $\sim 3e4$ pixels: $\chi^2 = \sum_{i=0}^{\text{all pixels}} \left(\frac{D(\mathbf{r}_i) - I(\mathbf{r}_i)}{\sigma(\mathbf{r}_i)} \right)^2$
 - Atmospheric parameters
 - A1: order 1 amplitude, A2: order 2 relative amplitude
 - $\Delta(\lambda)$ and $P_0(\lambda)$ parameters

Improvements: minimisation algo

- χ^2 function to minimize with 23 parameters
- IMINUIT too slow (or not well tuned...):
 - **1h20**
 - **$\chi^2_{\min}=37394$ (with 27429 pixels)**

```

start = time.time()
fit_workspace.simulation.fix_psf_cube = False
error = 0.1 * np.abs(guess) * np.ones_like(guess)
error[2:5] = 0.3 * np.abs(guess[2:5]) * np.ones_like(guess[2:5])
z = np.where(np.isclose(error, 0.0, 1e-6))
error[z] = 1.
fix_ = [False] * guess.size
# noinspection PyArgumentList
m = Minuit.from_array_func(fcn=nll, start=guess, error=error, errordef=1,
                           fix=fix_, print_level=2, limit=bounds)
m.tol = 10
m.migrad()
fit_workspace.p = m.np_values()
print(f"Minuit: total computation time: {time.time()-start}s")

```

Improvements: minimisation algo

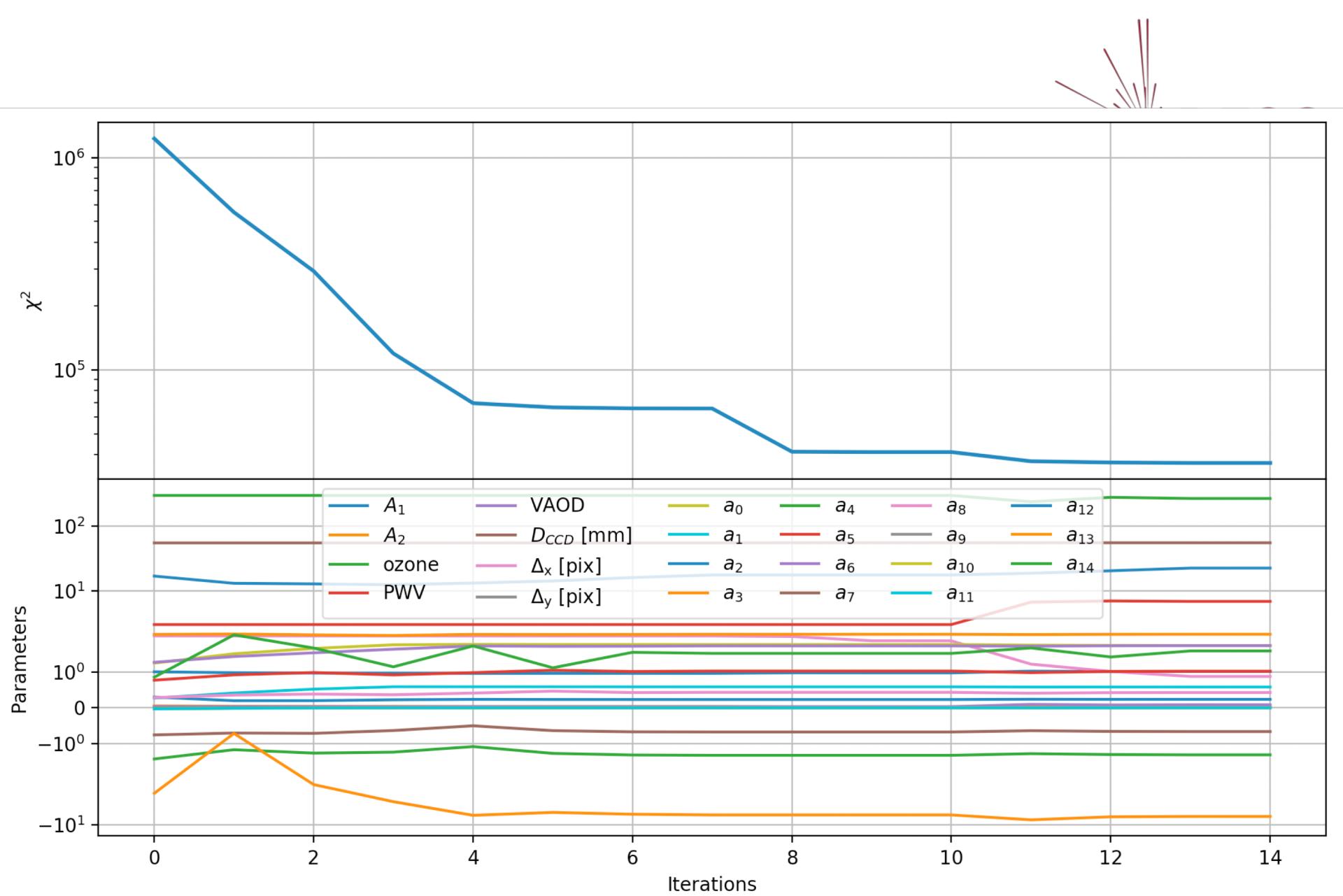
- Minimization of χ^2 the vector (not summed)
- Implementation of a Newton-Raphson method to find the zeros of the gradient of the χ^2

$$\chi^2(\theta) = \left(\vec{M}(\theta) - \vec{D} \right)^T W \left(\vec{M}(\theta) - \vec{D} \right) = \vec{R}^T(\theta) W \vec{R}(\theta)$$

$$\vec{\nabla}_\theta \chi^2 \approx 2J_0^T W \vec{R}_0 + 2J_0^T W J_0 \delta \vec{\theta} \text{ with } \delta \vec{\theta} = \vec{\theta} - \vec{\theta}_0 \quad \text{J: Jacobian}$$

$$\begin{aligned} \vec{\nabla}_\theta \chi^2 &= 0 \Rightarrow \vec{\theta} = \vec{\theta}_0 - (J_0^T W J_0)^{-1} J_0^T W \vec{R}_0 \\ &\Rightarrow \vec{\theta}_{k+1} = \vec{\theta}_k - \alpha (J_k^T W J_k)^{-1} J_k^T W \vec{R}_k \end{aligned} \quad \text{(J}^T W \text{J)}^{-1}: \text{covariance matrix}$$

- Minimization of α with a line search method
- $\chi^2_{\min}=36516$ (with 27429 pixels) in 6 minutes (14 steps)



Data here is a simulation

$A_1=1.010, A_2=0.026, PWV=6.856, OZ=269, VAOD=0.081, D=55.45\text{mm}, \text{shift}_x=0.87\text{pix}, \text{shift}_y=0.00\text{pix}$

