

(introduction à l')

Optique Adaptative

Marcel Carbillet

[marcel.carbillet@unice.fr]

[<https://lagrange.oca.eu/carbillet/enseignement/M2-Optique>]

Menu

- Imagerie astronomique à haute résolution angulaire (HRA) & turbulence atmosphérique
- Introduction à la *Speckle Interferometry*
- Introduction à l'optique adaptative (OA)
- Budget d'erreur post-OA & morphologie de la PSF
- Le côté *hard* de l'OA (ASO, MD, contrôle)
- Divers apartés...

+ démos et TP sur machines : fronts d'onde perturbés, formation d'image, bruits de détection, imagerie court-temps de pause, étude de l'erreur d'anisoplanétisme, évaluation de performance d'un système d'OA.

Au préalable...

- (1) Lire la charte informatique de l'OCA
<https://www.oca.eu/fr/charte-informatique>
- (2) Signer l'approbation et me la rendre !
- (3) Se connecter (compte Guest) et lancer un terminal
- (4) Se connecter sur le serveur de calcul de l'OCA srv-etudiant avec vos identifiants (voir liste plus bas) :
>> ssh -Y srv-etudiant.oca.eu -l nom_login
- (5) Pour changer votre mot de passe (dès la 1ère connexion) :
<https://dsi.oca.eu/spip.php?article600>

1	Pavel	Abajyan	pabajyan	9b-56UArh;T}
2	Rayan	Abdellah	rabdellah	xP\$A/2b.7q4T
3	Adam	Bellity	abellity	Kk?762!RxS:y
4	Varvara	Chiliaeva	vchiliaeva	ak_2SzLZ9-=5
5	Samba	Gueye	sgueye	5pX@h[WTx;37
6	Alek	Lagarrigue	alagarrigue	/uC56LDz]5/u
7	Mattis	Reisner	mreisner	>Jgg8]N[B9m6

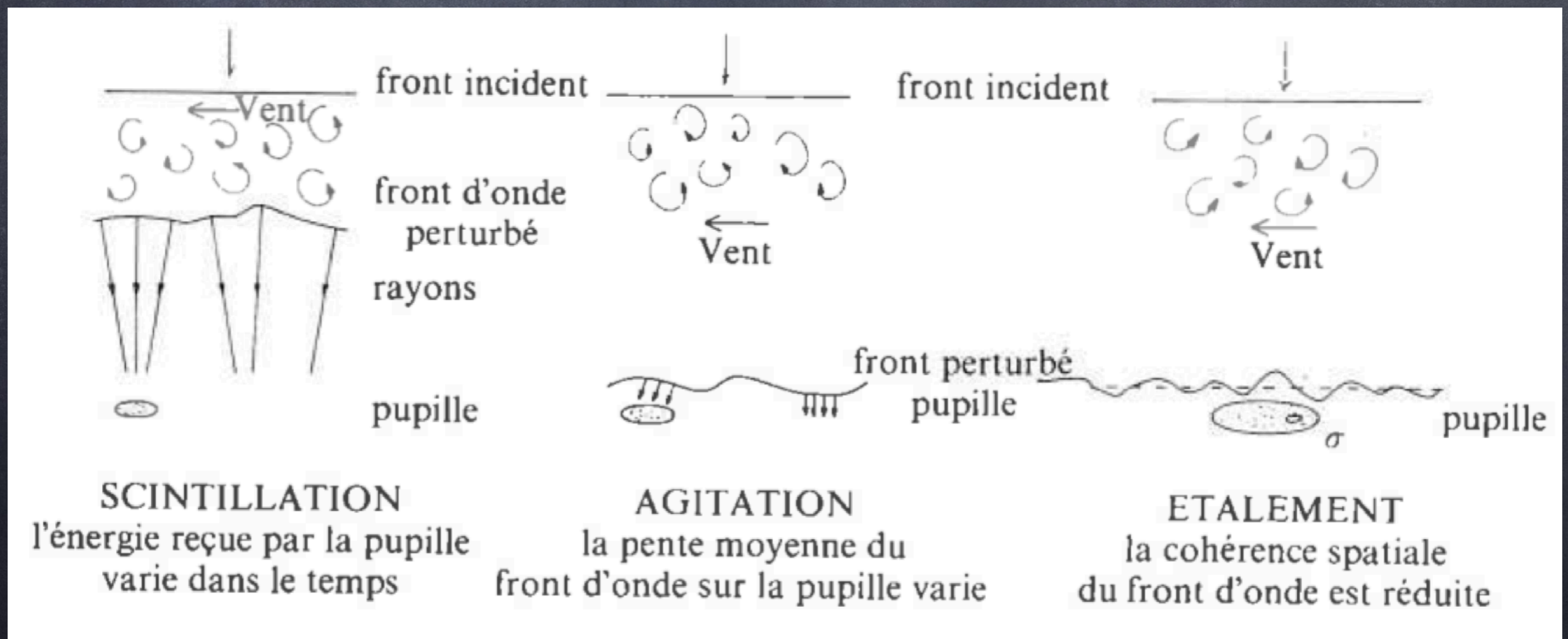
Images et turbulence - 1

L'image formée à travers l'atmosphère turbulente (au sens optique) est dégradée :

- Scintillation = fluctuation de l'éclat.
- Agitation = variation du photocentre.
- Étalement = perte de cohérence spatiale.

Images et turbulence - 1+

L'image formée à travers l'atmosphère turbulente (au sens optique) est dégradée :



Images et turbulence - 2

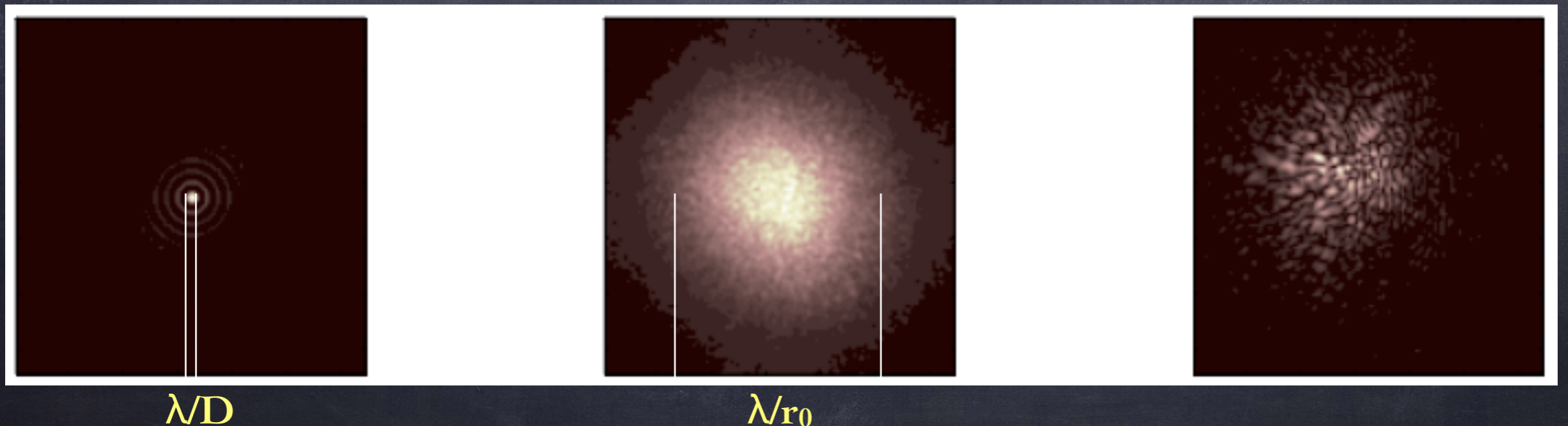
La relation objet-image qui lie l'éclairement $I(\alpha)$ dans le plan image (i.e. au foyer du télescope) à la luminance $O(\alpha)$ de l'objet (dans le ciel) est une relation de convolution impliquant la réponse impulsionnelle $S(\alpha)$ de l'ensemble télescope+atmosphère, où α est la direction de visée :

$$I(\vec{\alpha}) = O(\vec{\alpha}) * S(\vec{\alpha})$$

Images et turbulence - 3

$$I(\vec{\alpha}) = O(\vec{\alpha}) * S(\vec{\alpha})$$

Relation valable notamment à la condition que le système soit invariant par translation (tout se passe à l'intérieur du domaine d'isoplanétisme)...

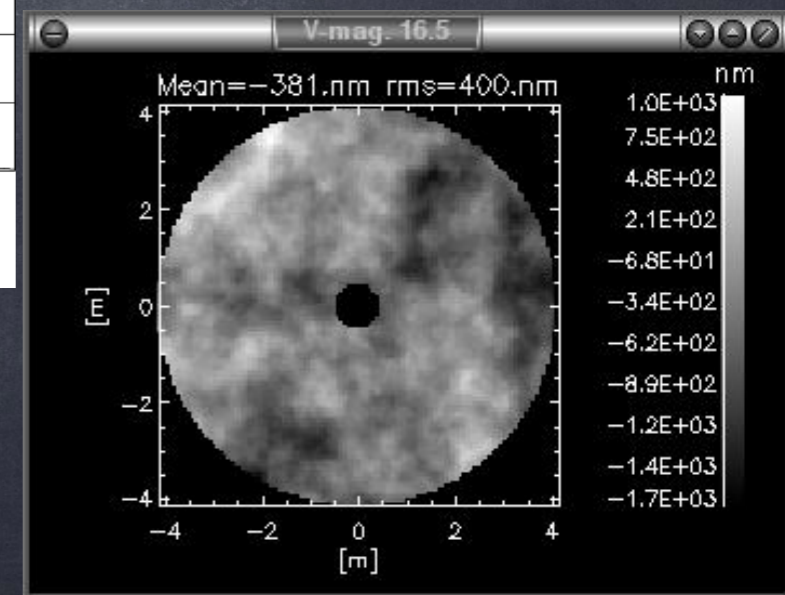
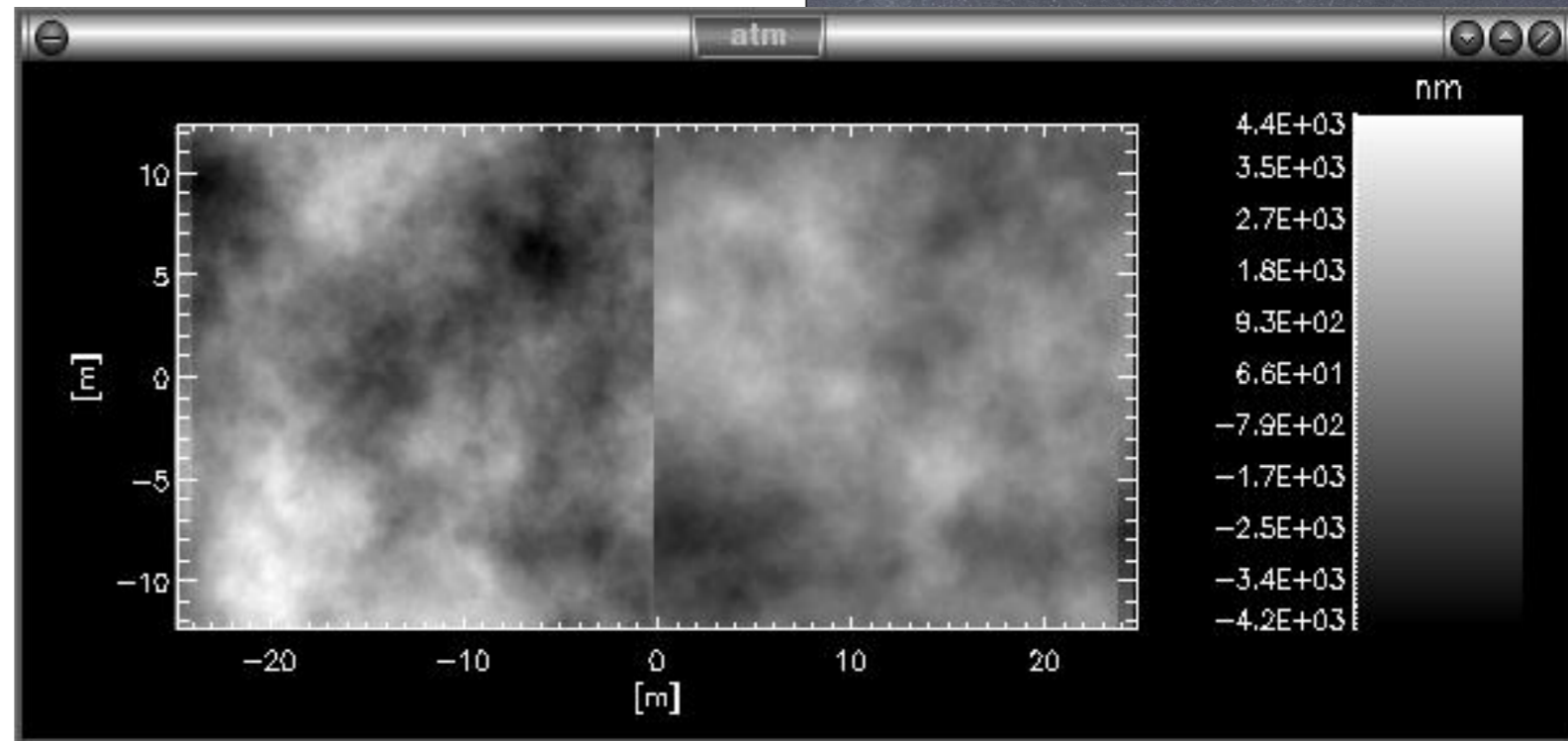
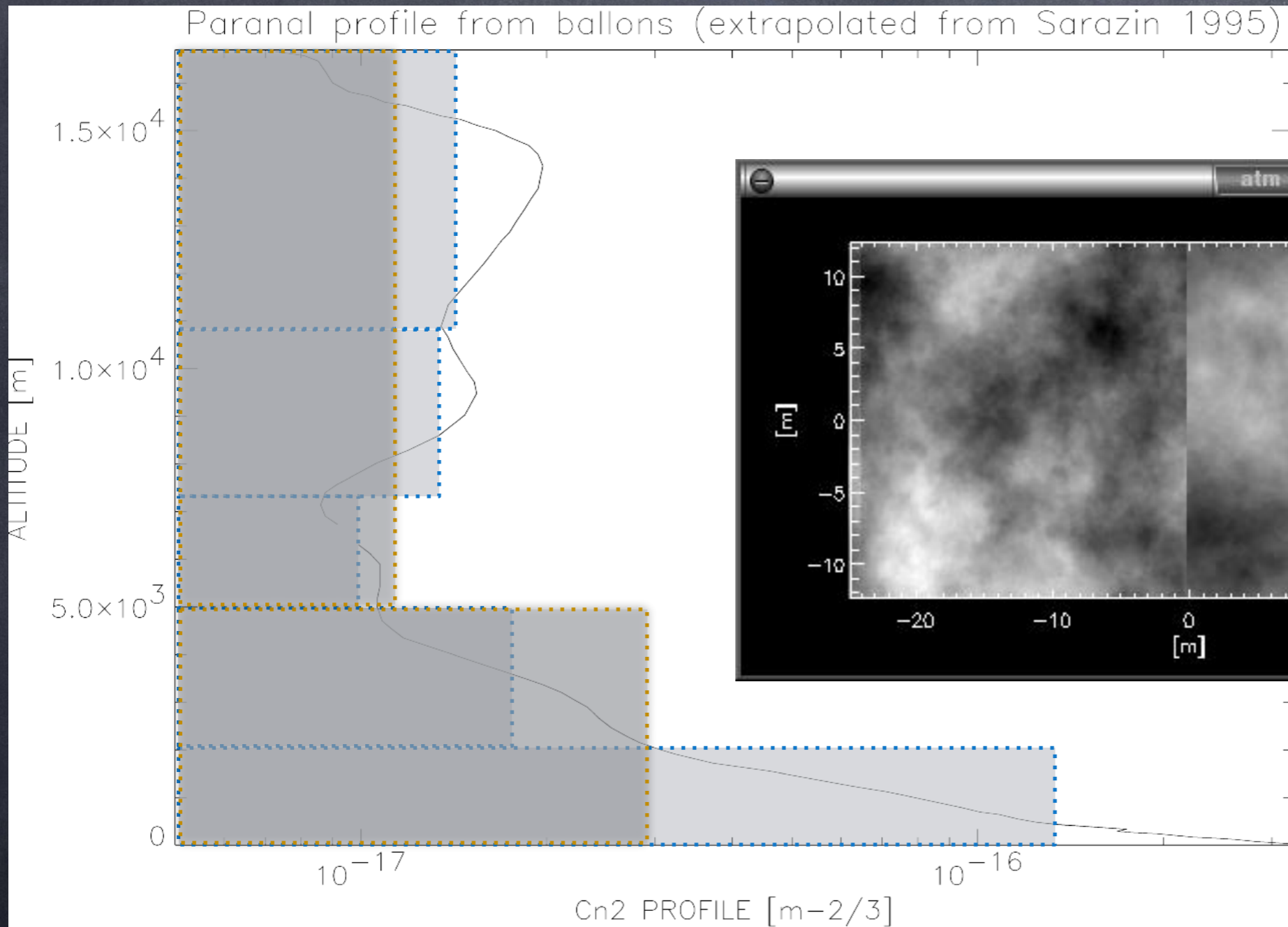


Images et turbulence - 4

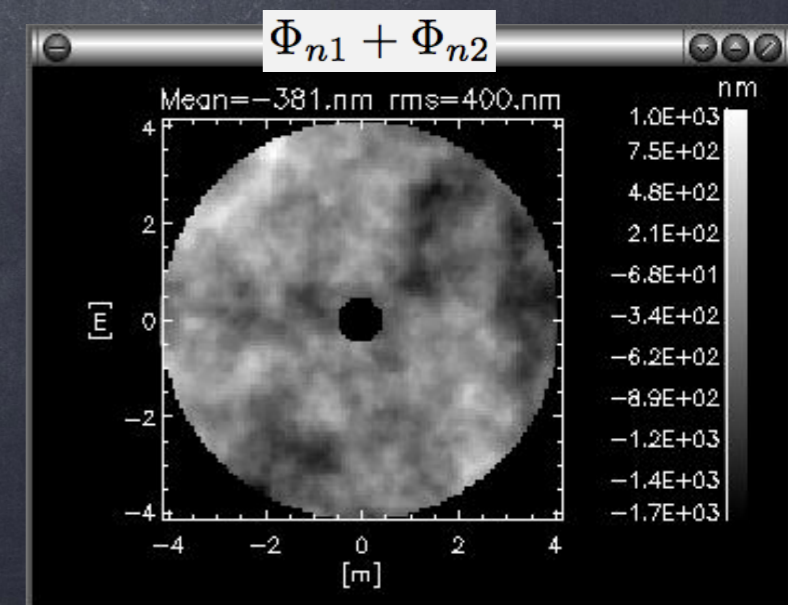
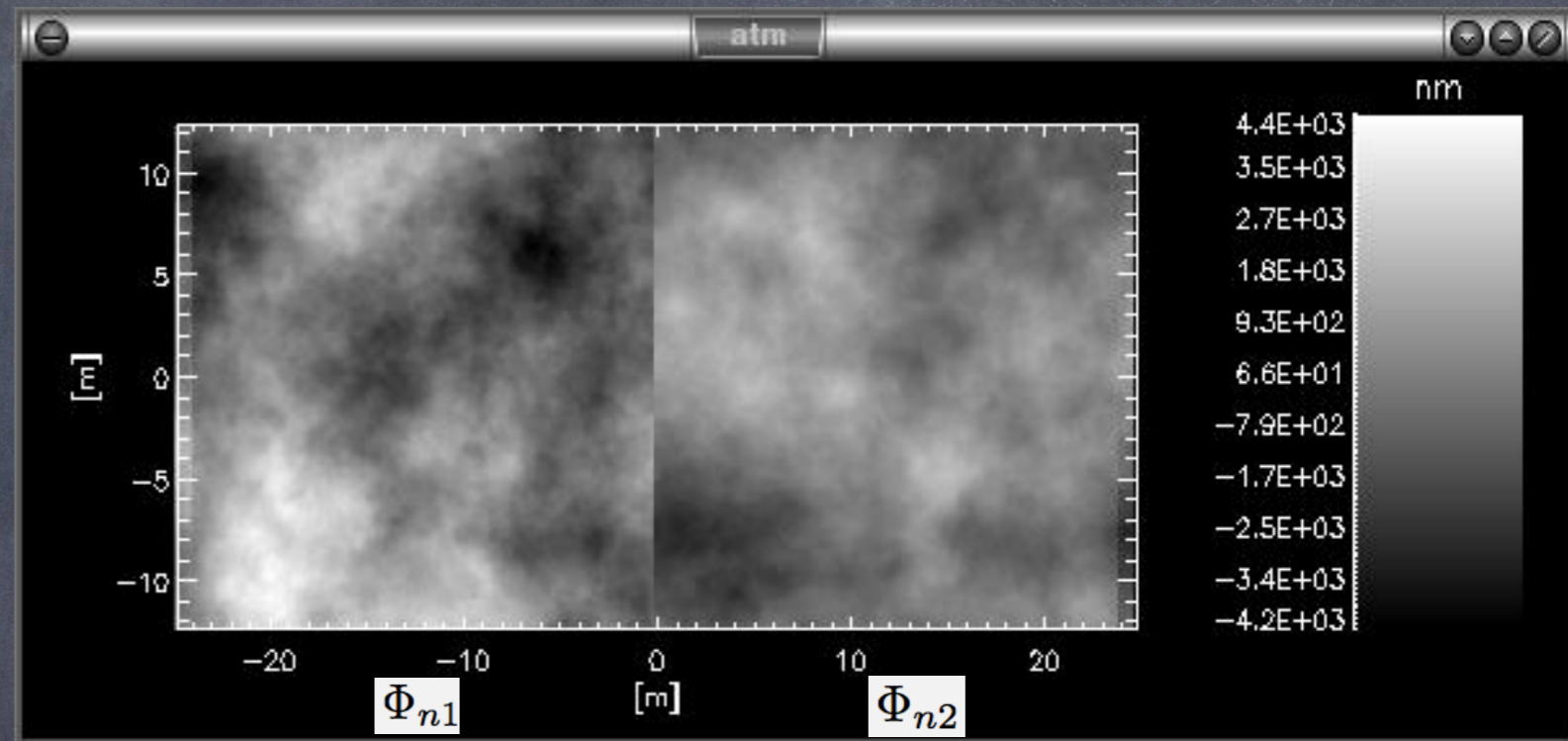
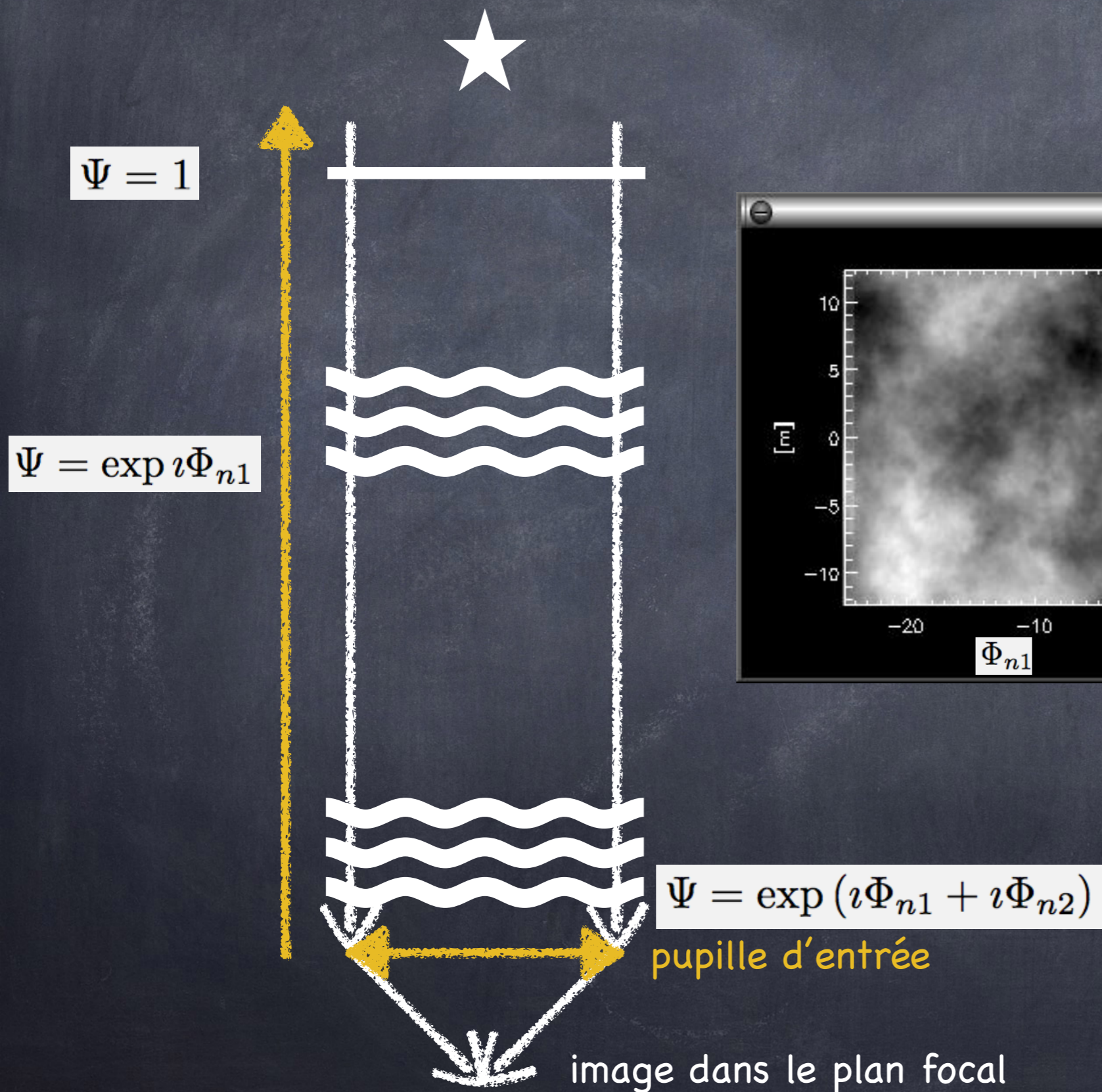
Quelques ordres de grandeur concernant l'atmosphère turbulente :

	$\lambda = 500 \text{ nm}$	$\lambda = 2.2 \mu\text{m}$
paramètre de Fried r_0	$\rightarrow 10 \text{ cm}$	60 cm
vitesse des couches turbulentes (v)	$\rightarrow 10 \text{ m/s}$	id.
=> FWHM d'une image ($\epsilon \approx \lambda / r_0$)	$\rightarrow 1''$	$\sim 1''$
=> temps caractéristique ($t_0 \propto r_0 / v$)	$\rightarrow 3 \text{ ms}$	18 ms

Images & turbulence - 5



Images & turbulence - 5+



Images et turbulence - 5++

pupille d'entrée



image dans le plan focal



Images et turbulence - 6

Le front d'onde est, à $\lambda/2\pi$ près, la phase $\phi(\mathbf{r})$ de l'onde $\psi(\mathbf{r})$ qui a traversé l'atmosphère turbulente avant de parvenir au télescope :

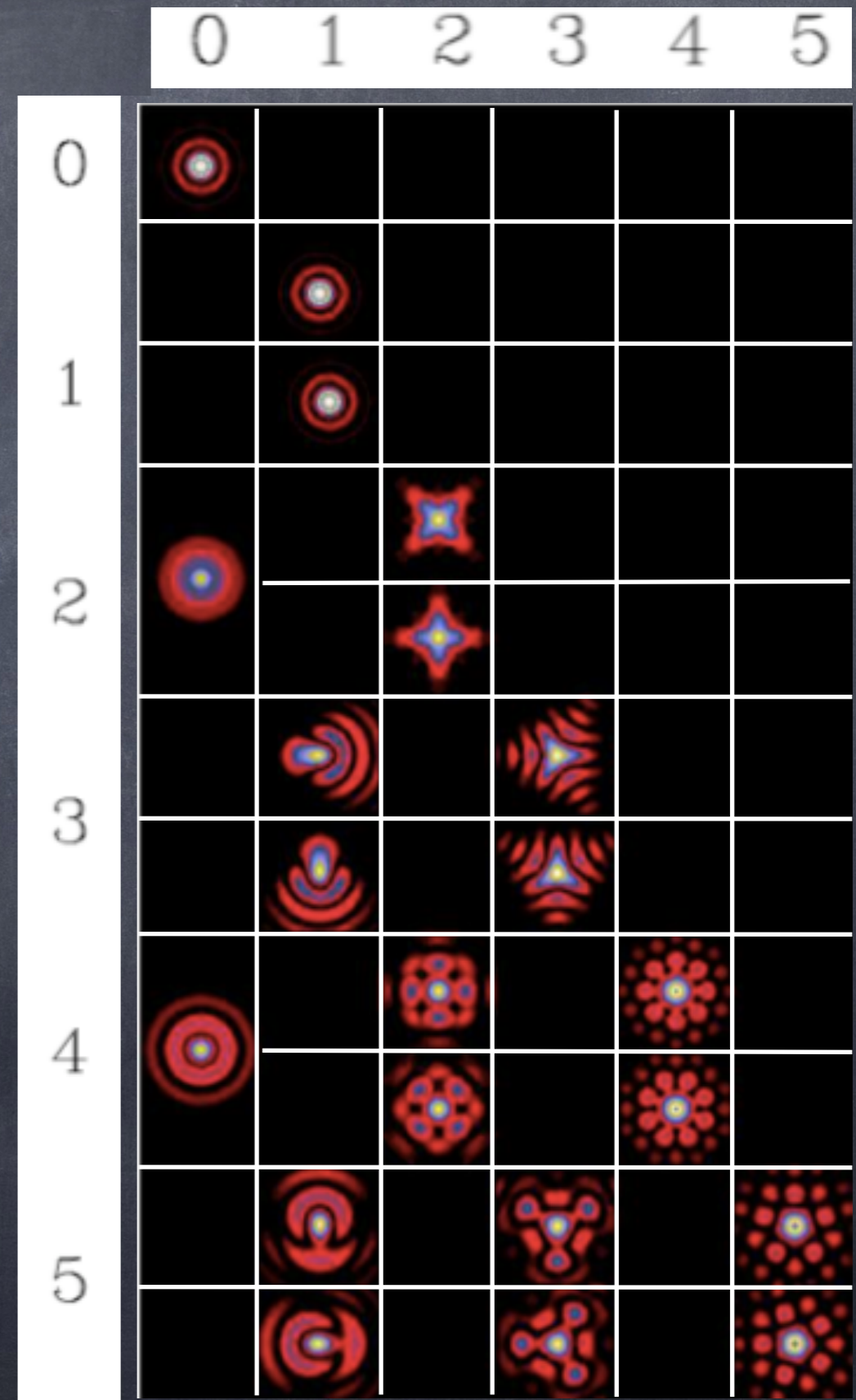
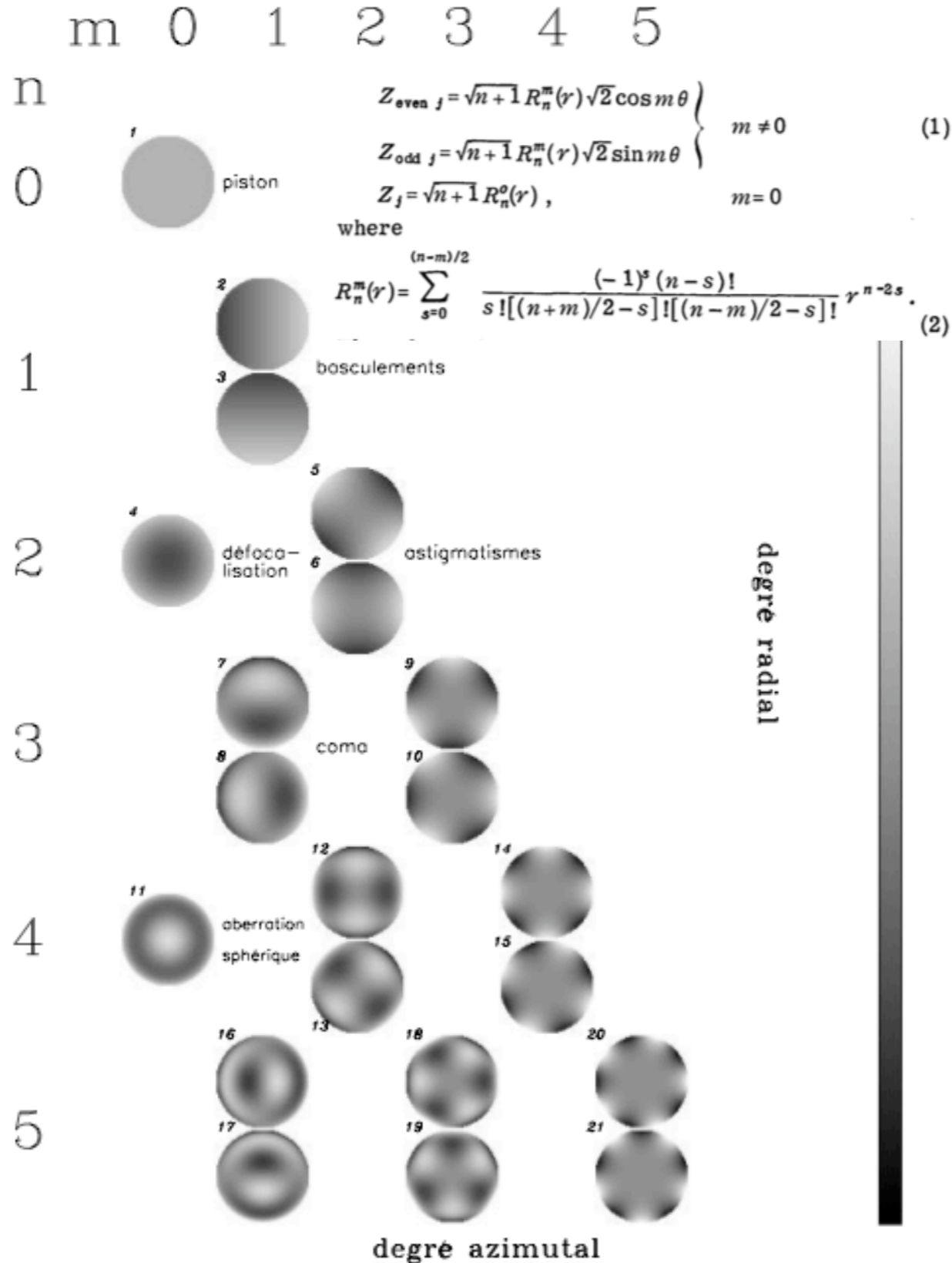
$$\Psi(\vec{r}) = A(\vec{r}) \exp\{i\Phi(\vec{r})\}$$

Cette phase peut se décomposer selon une base de polynômes, par exemple ceux de Zernike :

$$\Phi(\vec{r}) = \sum_i a_i Z_i(\vec{r})$$

Images & turbulence - 6+

polynômes de Zernike



pupille d'entrée

Images & turbulence - 7

intensité de la turbulence [$m^{1/3}$]

$$r_0 = 0.185 \lambda^{5/6} \cos(\gamma)^{3/5} \left[\int_0^\infty C_n^2(z) dz \right]^{-3/5}$$

r_0 en bande H connaissant r_0 à 500nm (10cm) ?...

$$\tau = 0.36 \frac{r_0}{v}$$

$$\epsilon = 0.98 \frac{\lambda}{r_0}$$

$$\theta_0 = 0.314 \frac{r_0}{\bar{h}}$$

$$\bar{v} = \left(\frac{\int C_n^2(h) v(h)^{5/3} dh}{\int C_n^2(h) dh} \right)^{3/5}$$

$$N_s \simeq 0.34 \left(\frac{D}{r_0} \right)^2$$

$$\bar{h} = \left(\frac{\int C_n^2(h) h^{5/3} dh}{\int C_n^2(h) dh} \right)^{3/5}$$

$$V_0 = c \tau_0 r_0^2$$

volume de cohérence

$$G_0 = r_0^2 \tau_0 \theta_0^2$$

"étendue" de la cohérence

Nombre de speckles pour $r_0=10\text{cm}$ et $D=1\text{m}$?...

Images et turbulence - 7+

r_0 en bande H sachant r_0 à 500nm ?...

$$r_0 = 0.185 \lambda^{6/5} \cos(\gamma)^{3/5} \left[\int_0^\infty C_n^2(z) dz \right]^{-5/3}$$

$$r_0^{H=1.65 \mu\text{m}} = r_0^{500 \text{ nm}} \left(\frac{1.65}{0.5} \right)^{6/5} \simeq 0.42$$

Nombre de speckles pour $r_0=10\text{cm}$ et $D=1\text{m}$?...

$$N_S^{500 \text{ nm}} \simeq 0.34 \left(\frac{1.0}{0.1} \right)^2 \simeq 34$$

$$N_S^H \simeq 0.34 \left(\frac{1.0}{0.42} \right)^2 \simeq 2$$

Images et turbulence - 8

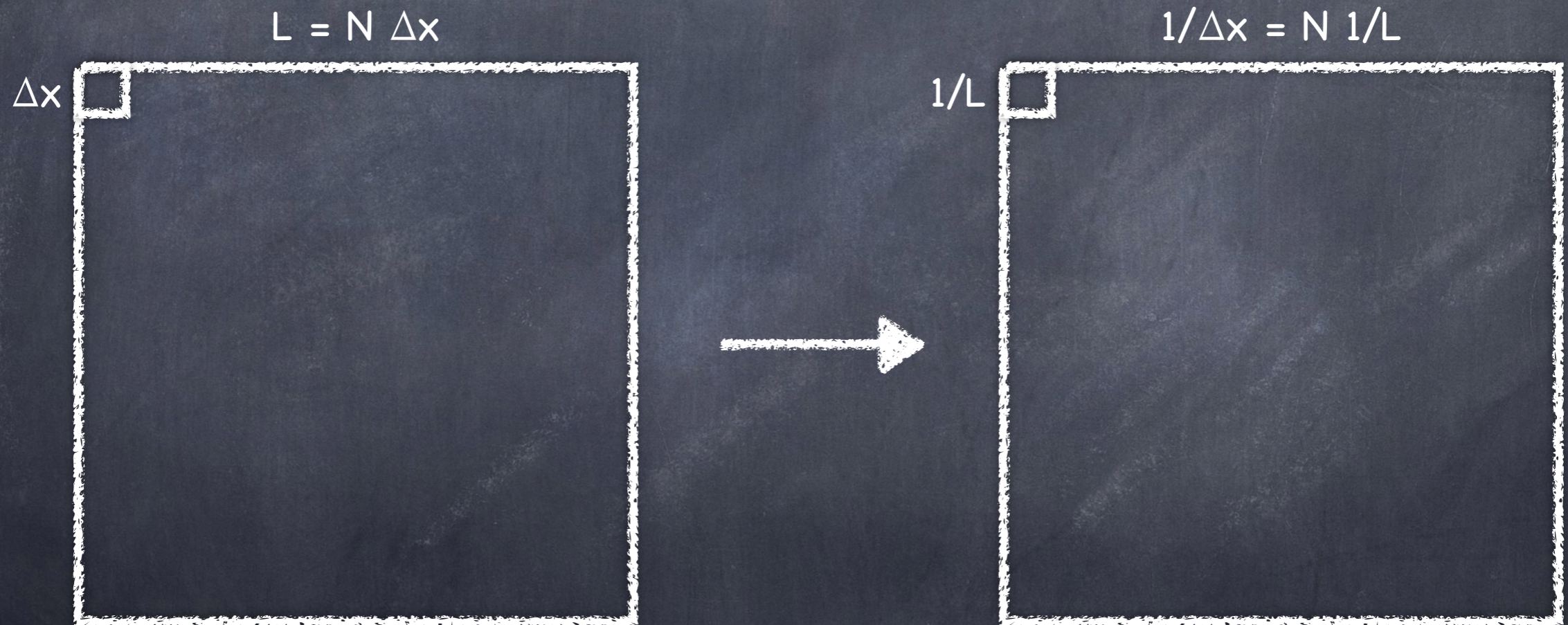
$$\Phi_{\varphi}(\vec{\nu}) = 0.0228 r_0^{-\frac{5}{3}} \left(\nu^2 + \frac{1}{\mathcal{L}_0^2} \right)^{-\frac{11}{6}}$$

Densité Spectrale de Puissance (DSP) de la phase, fonction de la fréquence spatiale

Modèle de Kolmogorov/von Kármán

- Kolmogorov : échelle externe \mathcal{L}_0 infinie.
- On peut affiner en considérant aussi l'échelle interne ℓ_0 .
- \exists d'autres modèles avec \mathcal{L}_0 finie et ℓ_0 non-nulle.

(Rappel sur la Transformée de Fourier Discrète...)



Images et turbulence - 8+

$$\Phi_{\varphi}(\vec{\nu}) = 0.0228 r_0^{-\frac{5}{3}} \left(\nu^2 + \frac{1}{\mathcal{L}_0^2} \right)^{-\frac{11}{6}}$$

Numériquement, et en considérant des fronts d'onde de côté « dim » pixels correspondant à « L » mètres, ceci s'écrit (en "dé-dimensionnalisant" avec $L_0=L_0 L/L$ et $\nu=\nu L/L\dots$) :

```
freq = findgen(dim)
dsp   = .0228*(L/r0)^(5/3.)*L^2*(freq^2+(L/L0)^2)^(-11./6)
```

Ce qui, avec la bonne échelle en fréquences, peut se tracer :

```
plot_oo, 1./L*findgen(dim), dsp, XR=[1/L/1.2,dim*1/L*1.2], /XS
```

=> faire une routine qui calcule et trace la DSP(r_0, L_0, dim, L) pour différents $[r_0, L_0]\dots$ [avec, par ex., $\text{dim}=1000, L=100., r_0=0.1, L_0=100.,10.,1.$]

-> Lire aussi Aime (Sec. 1 & Sec. 2) et Maire (Chap.1)...

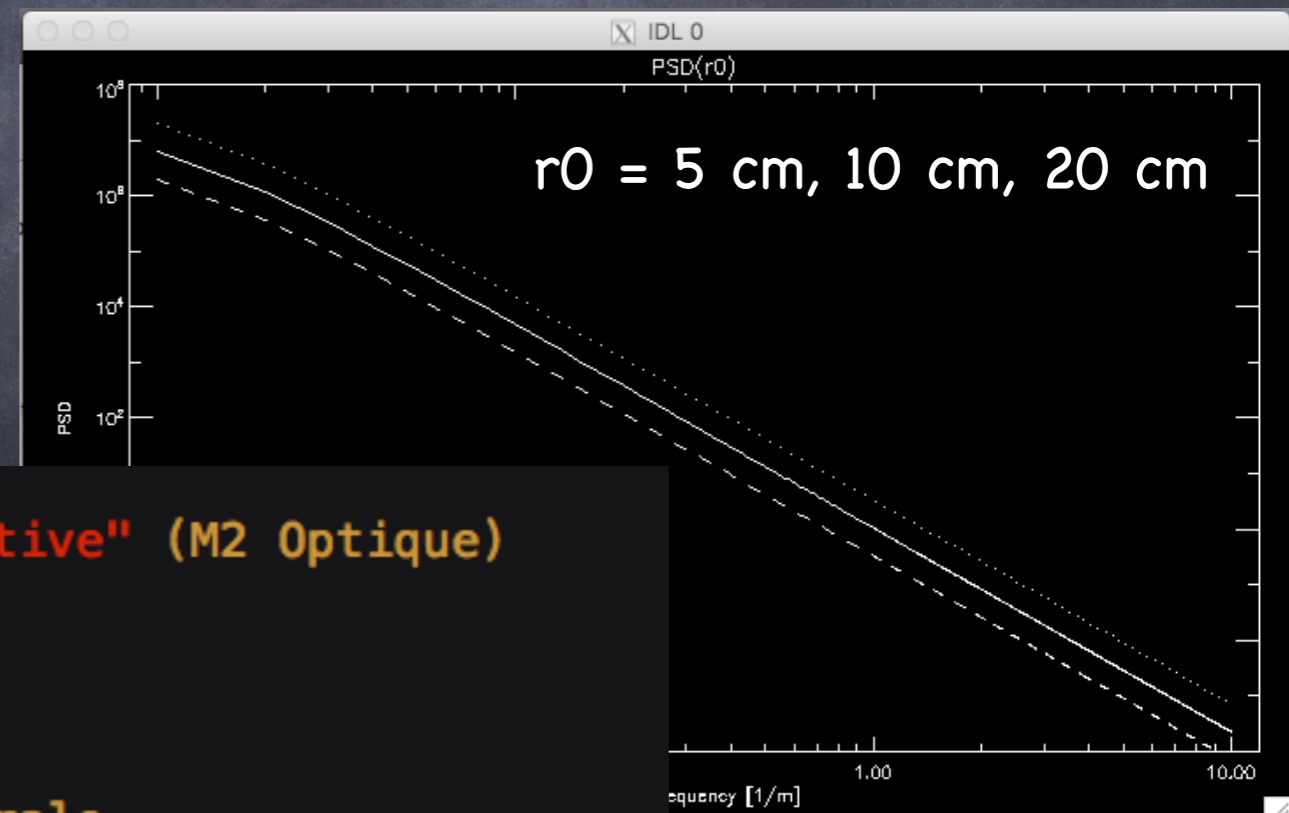
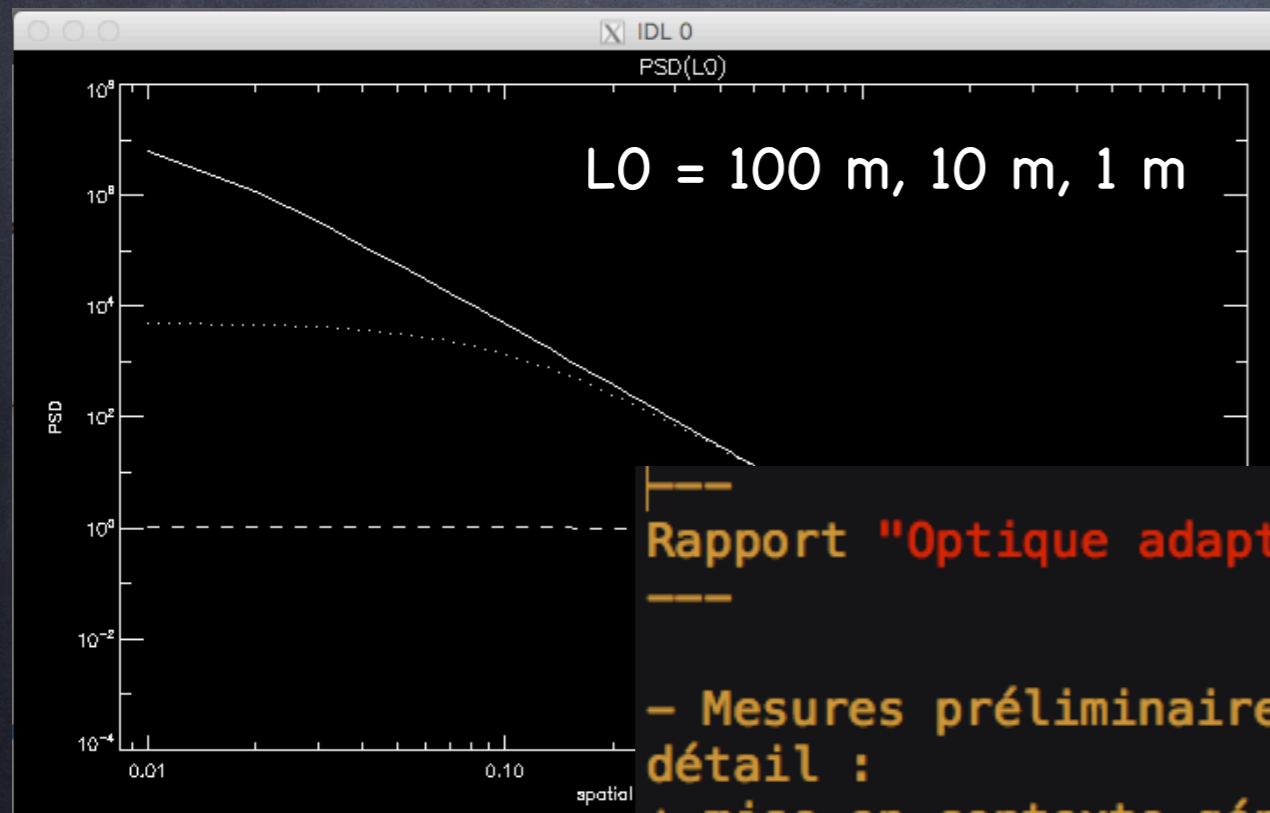
Images & turbulence - 9

```
[marcel@MBA-de-Marcel -> ssh -Y srv-etudiant.oca.eu
```

```
[marcel@srv-etudiant.oca.eu:~]$  
Warning: No xauth data;  
Last login: Wed Feb 21 09:00:00 2013  
[[marcel@srv-etudiant ~]$  
IDL Version 8.5.1 (linux64)  
Copyright 2008, H. V. Vetter, Inc., a subsidiary of  
H. V. Vetter, Inc., a subsidiary of  
Installation number: 3588  
Licensed for use by: Observatoire de la Côte d'Azur
```

```
[IDL> plot, [1,2], [0,4]
```

```
1 function dsp_theo, dim, L, r0, L0  
2  
3 freq = findgen(dim)  
4 dsp = .0228*(L/r0)^(5/3.)*L^2*(freq^2+(L/L0)^2)^(-11./6)  
5  
6 ;to be plotted afterwards with:  
7 ;plot_oo, 1./L*findgen(dim), dsp, XR=[1/L/1.2,dim*1/L*1.2], /XS, $  
8 ;   TIT='PSD(L0)', XTIT='spatial frequency [1/m]', YTIT='PSD'  
9 ;oplot , 1./L*findgen(dim), dsp, LINE=1  
10 ;playing, e.g., with L0=100.,10.,1., or r0=.05, .1, .2  
11  
12 return, dsp  
13 end
```



Rapport "Optique adaptative" (M2 Optique)

- Mesures préliminaires

détail :

+ mise en contexte générale

+ plot de $DSP(r0, L0)$

+ => ccl sur l'influence de $r0$ et celle de $L0$

The well-known FFT method allows us to generate phase screens $\varphi(\vec{r})$, where \vec{r} is the two-dimensional position within the phase screen, assuming usually either a Kolmogorov or a von Karman spectrum $\Phi_\varphi(\vec{\nu})$, where $\vec{\nu}$ is the two-dimensional spatial frequency, from which is computed the modulus of $\tilde{\varphi}(\vec{\nu})$, the Fourier transform of $\varphi(\vec{r})$. Assuming the near-field approximation and small phase perturbation [3], the von Karman/Kolmogorov spectrum is given by

$$\Phi_\varphi(\vec{\nu}) = 0.0229 r_0^{-\frac{5}{3}} \left(\nu^2 + \frac{1}{\mathcal{L}_0^2} \right)^{-\frac{11}{6}}, \quad (1)$$

where r_0 is the Fried parameter and \mathcal{L}_0 is the wavefront outer scale (infinite for the Kolmogorov model). Within the framework of the classical FFT-based technique, a turbulent phase screen $\varphi_L(\vec{r})$ of physical length L is obtained by taking the inverse FFT of $\tilde{\varphi}_L(\vec{\nu})$, the modulus of which is obtained from Eq. (1) by applying the definition of the power spectrum, which is

$$\begin{aligned} \Phi_\varphi(\vec{\nu}) &= \lim_{L \rightarrow \infty} \left(\frac{\langle |\tilde{\varphi}_L(\nu)|^2 \rangle}{L^2} \right) \\ \Rightarrow |\tilde{\varphi}_L(\nu)| &\simeq L r_0^{-\frac{5}{6}} \sqrt{0.0228} \left(\nu^2 + \frac{1}{\mathcal{L}_0^2} \right)^{-\frac{11}{12}}, \quad (2) \end{aligned}$$

and which phase is random and uniformly distributed.

-> Génération de front d'onde perturbés :

(la même manipulation que précédemment est ici appliquée afin d'obtenir la formulation numérique ci-dessous.)

-> Carbillet & Riccardi (sec. 2)...

The obtained phase screen is thus numerically written

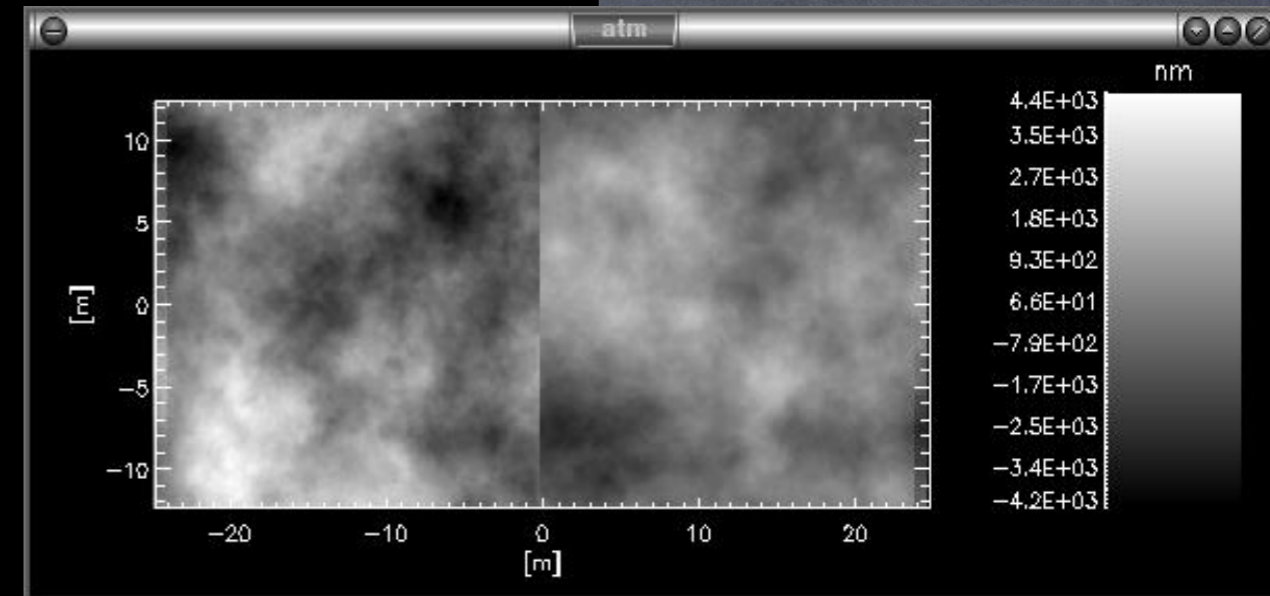
$$\begin{aligned} \varphi_L(i,j) &= \sqrt{2} \sqrt{0.0228} \left(\frac{L}{r_0} \right)^{\frac{5}{6}} \left\{ \text{FFT}^{-1} \left[\left(k^2 + l^2 \right. \right. \right. \\ &\quad \left. \left. \left. + \left(\frac{L}{\mathcal{L}_0} \right)^2 \right)^{-\frac{11}{12}} \exp\{i\theta(k,l)\} \right] \right\}, \quad (3) \end{aligned}$$

where i and j are the indices in the direct space, k and l are the indices in the FFT space, $\{ \}$ stands for either *real part of* or *imaginary part of*, i is the imaginary unit, and θ is the random uniformly distributed phase (between $-\pi$ and π). The factor $\sqrt{2}$ comes from the fact that here we use both the real and imaginary parts of the original complex generated FFT phase screens, which are independent one from the other [4]. This kind of phase screen suffers, however, from the lack of spatial frequencies lower than the inverse of the necessarily finite length L of the simulated array.

```

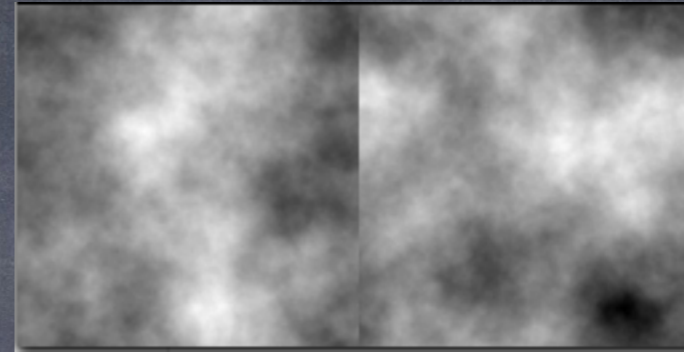
1 function wfgeneration, dim, length, L0, r0, lambda, SEED=seed
2 ;
3 ; wave-front (wf) generation following Kolmogorov or von Karman model
4 ;
5 ; dim      = wf linear dimension [px],
6 ; length  = wf physical length [m],
7 ; L0      = wf outer-scale [m],
8 ; seed    = random generation seed (OPTIONAL),
9 ; r0      = Fried parameter at wavelength 'lambda' [m],
10 ; lambda  = wavelength at which r0 is defined.
11 ;
12 ; Marcel Carillet [marcel.carillet@unice.fr],
13 ; UMR 7293 Lagrange (UNS/CNRS/OCA), February 2013.
14 ;
15 ; Last modification: Feb. 2014
16 ;
17 phase = (randomu(seed,dim,dim)-.5) * 2*!PI ; rnd uniformly distributed phase
18 ; (between -PI and +PI)
19 rr = dist(dim)
20 if L0 eq !VALUES.F_INFINITY then rr[0,0] = 1.; avoid 1/infinity afterwards
21 ; for Kolmogorov model
22 modul = (rr^2+(length/L0)^2)^(-11/12.) ; vonKarman/Kolmogorov model
23 if L0 eq !VALUES.F_INFINITY then modul[0,0] = 0.
24 ; put again modulus[0,0]=0
25 ; for Kolmogorov model
26 screen = fft(modul*exp(complex(0,1)*phase), /INVERSE)
27 ; compute wf
28 screen *= sqrt(2)*sqrt(.0228)*(length/r0)^(5/6.)*lambda/(2*!PI)
29 ; proper normalization of wf
30 ; (considering an r0 at lambda)
31 screen -= mean(screen)
32
33 return, screen ; deliver 2 independent wf:
34 ; float(screen) & imaginary(screen)
35 end

```



« wf generation »
générer un cube
de fronts d'onde
statistiquement
indépendants
(une centaine)...
=> calculer
l'écart-type
moyen pour
différents $[r_0, L_0]$

Images & turbulence - 10



```
[IDL> .r wfgeneration
% Compiled module: WFGENERATION.
[IDL> wf=wfgeneration(128,2.,27.,.1,500E-9,SEED=seed)
% Compiled module: DIST.
[IDL> wf1=float(wf)
[IDL> wf2=imaginary(wf)
[IDL> tvscl, [wf1,wf2]
[IDL> wf=wfgeneration(128,2.,27.,.1,500E-9,SEED=seed)
[IDL> wf1=float(wf)
[IDL> wf2=imaginary(wf)
[IDL> tvscl, [wf1,wf2]
IDL> █
```

```
[IDL> .rn wfcube
% Compiled module: WFCUBE.
[IDL> print, wfcube(128L,2.,27.,.1,500E-9,100L)*1E9
% Compiled module: COMPUTE_RMS.
367.668
% Program caused arithmetic error: Floating underflow
IDL> █
```

```
function wfcube, dim, length, L0, r0, lambda, n_wf
;
; use:
; dim      = 128L      ; [px] wf dimension
; length   = 2.        ; [m] wf physical dimension
; L0       = 27.       ; [m] outerscale
; r0       = .1        ; [m] Fried parameter
; lambda   = 500E-9    ; [m] r0 wavelength
; n_wf     = 100L     ; nb of generated wf
;
; print, wfcube(dim,length,L0,r0,lambda,n_wf,filename,SEED=seed)
; -> prints the rms value
;
; sub-routines needed:
; wfgeneration.pro, calcul_rms.pro
;
; Marcel Carbillet [marcel.carbillet@unice.fr],
; lab. Lagrange (UCA, OCA, CNRS), Feb. 2018.
;
; Last modification: Feb. 2018
;
cube = fltarr(dim, dim, n_wf)

for i=0, n_wf/2-1 do begin
    wf = wfgeneration(dim, length, L0, r0, lambda, SEED=seed)
    cube[:,*,2*i] = float(wf)
    cube[:,*,2*i+1] = imaginary(wf)
endfor

rms = compute_rms(cube)

return, rms
end
```

```
function compute_rms, cube
; cube: cube of wavefronts (square wf, no pupil!)

n_wf = (size(cube))[3]
rms = fltarr(n_wf)

for i=0,n_wf-1 do begin
    toto = moment(cube[:,*,i], SDEV=dummy)
    rms[i] = dummy
endfor

rms_moy = mean(rms)

return, rms_moy
end
```

+ plot ou tableau de rms(r0,L0)
+ => ccl sur l'influence de r0 et celle de L0