Images & turbulence — 14

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

Which, numerically written, and by considering wavefronts made of 'dim' pixels corresponding to 'L' meters, becomes:
(re-writing - ``de-dimensionalizing" - the equation with $L_0=L_0$ L/L and $\nu=\nu$ L/L…)

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

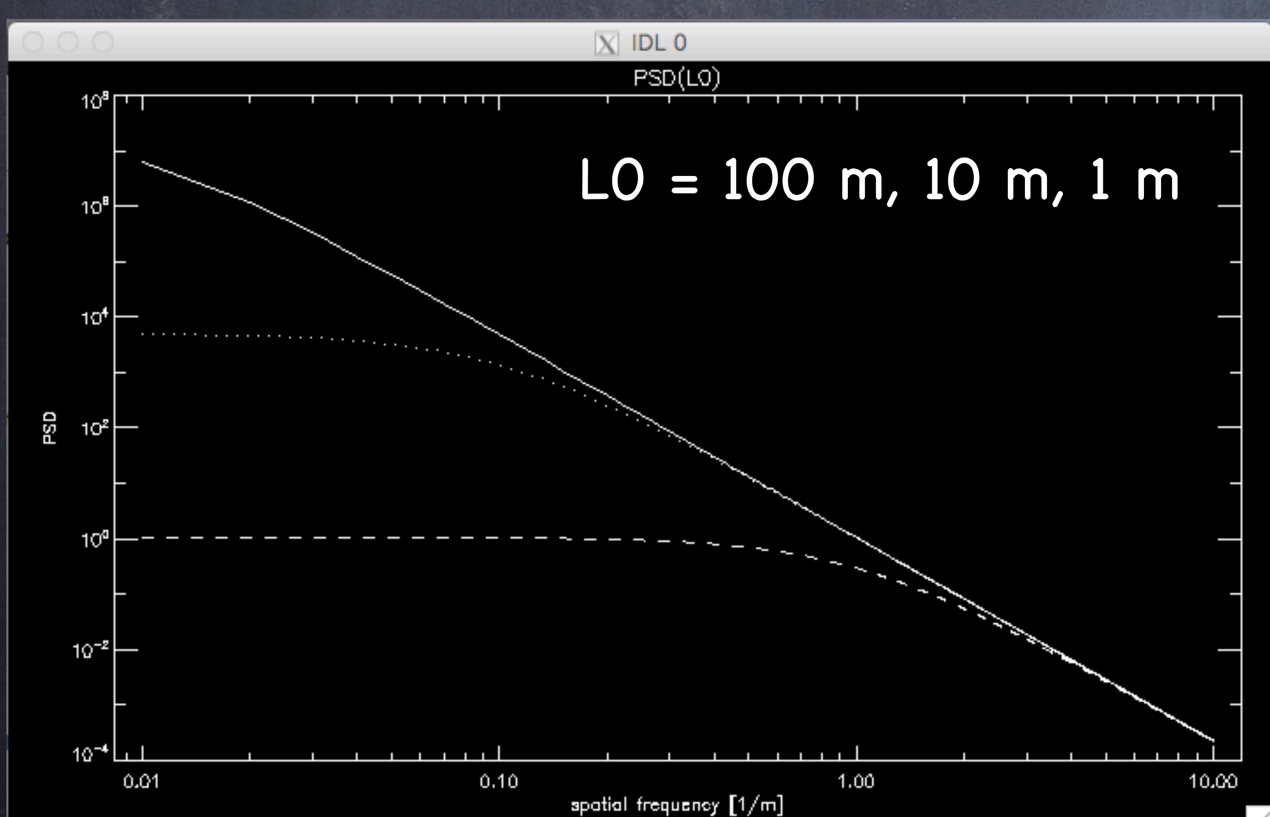
And which (with the right frequency scale) can be plot with:

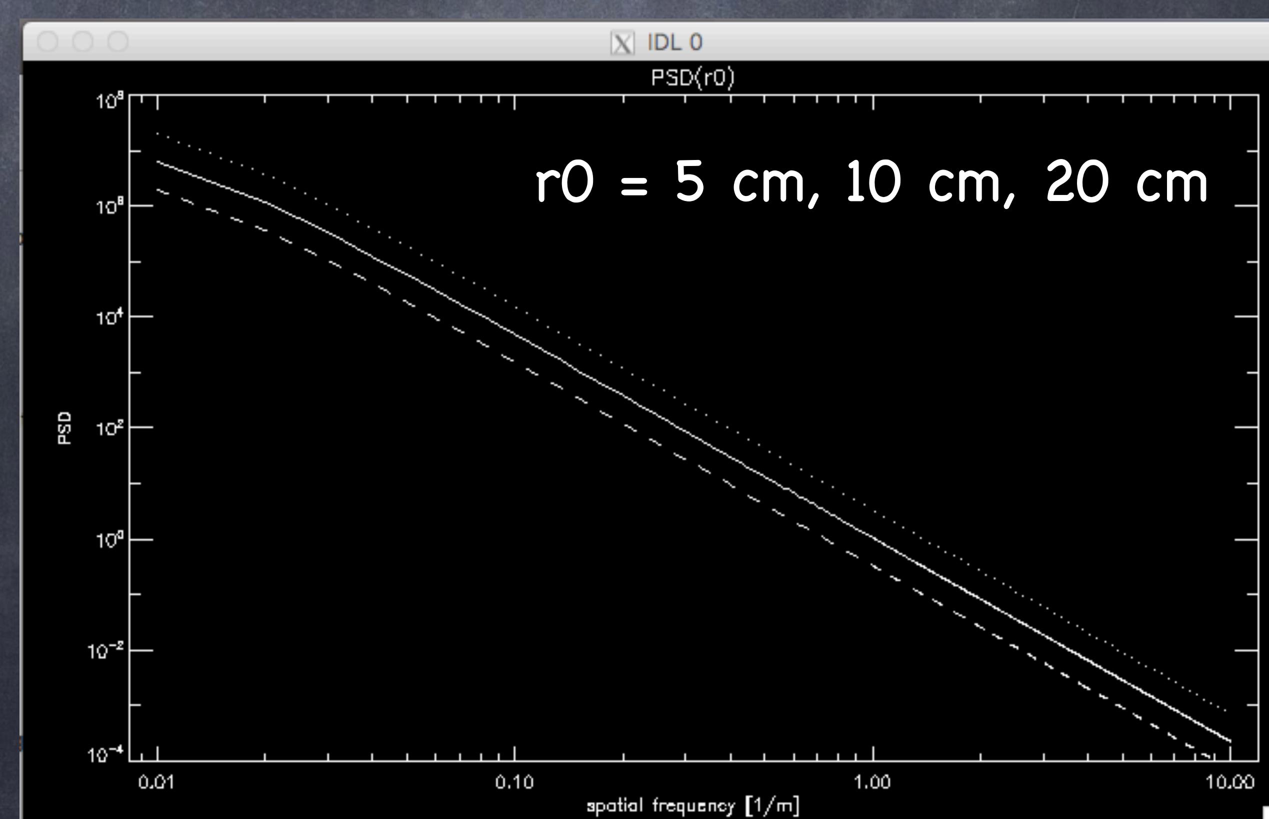
```
plot_oo, 1./L*findgen(dim), dsp, XR=[1/L/1.2,dim*1/L*1.2], /XS
```

=> make a function that computes PSD(L_0, r_0, dim, L) and plot it for different [r_0, L_0]… [with, for example: dim=1000, L=100., r0=0.1, L0=100.,10.,1.]

Images & turbulence — 15

```
function dsp_theo, dim, L, r0, L0

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

;to be plotted afterwards with:
;plot_oo, 1./L*findgen(dim), dsp, XR=[1/L/1.2,dim*1/L*1.2], /XS, $
;         TIT='PSD(L0)', XTIT='spatial frequency [1/m]', YTIT='PSD'
;oplot  , 1./L*findgen(dim), dsp, LINE=1
;playing, e.g., with L0=100.,10.,1., or r0=.05, .1, .2

return, dsp
end
```





```
REPORT

- Preliminary measures
+ introduction
+ PSD(r0, L0) plot
+ ccl on influence of r0 and L0
+ (more to come...)
```

Images & turbulence — 16

-> For next time: read Aime (Sec. 1 & Sec. 2) and Maire (Chap.1)...

Chapitre 1

Introduction

Jérôme Maire, PhD thesis (in French), chap.1

Long Telescopes may cause Objects to appear brighter and larger than short ones can do, but they cannot be so formed as to take away the confusion of the Rays which arises from the Tremors of the Atmosphere.
I. Newton, *1717 Optics, Sec. Ed., Book I, Part I, Prop.VIII*

INSTITUTE OF PHYSICS PUBLISHING EUROPEAN JOURNAL OF PHYSICS

Eur. J. Phys. **22** (2001) 169–184 www.iop.org/Journals/ej PII: S0143-0807(01)14580-0

Teaching astronomical speckle techniques

Claude Aime

UMR 6525 Astrophysique, Faculté des Sciences de l'Université de Nice Sophia-Antipolis, Parc Valrose-06108, Nice Cedex 2, France

Received 7 June 2000, in final form 6 December 2000

Abstract
This paper gives an introduction to speckle techniques developed for high angular-resolution imagery in astronomy. The presentation is focussed on fundamental aspects of the techniques of Labeyrie and Weigelt. The formalism used is that of Fourier optics and statistical optics, and corresponds to graduate level. Several new approaches of known results are presented. An operator formalism is used to identify similar regions of the bispectrum. The relationship between the bispectrum and the phase closure technique is presented in an original way. Effects of photodetection are treated using simple Poisson statistics. Realistic simulations of astronomical speckle patterns illustrate the presentation.

also have a look here [A. Tokovinin tutorial on atm'c turbulence]:

https://www.ctio.noirlab.edu/~atokovin/tutorial/part1/turb.html

Images & turbulence — 17

-> Perturbed wavefront generation

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

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

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

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

and which phase is random and uniformly distributed.

(From Carbillet & Riccardi, sec. 2: read it as well…)

(the same manipulation as before is applied here in order to obtain the numerical formulation here below.)

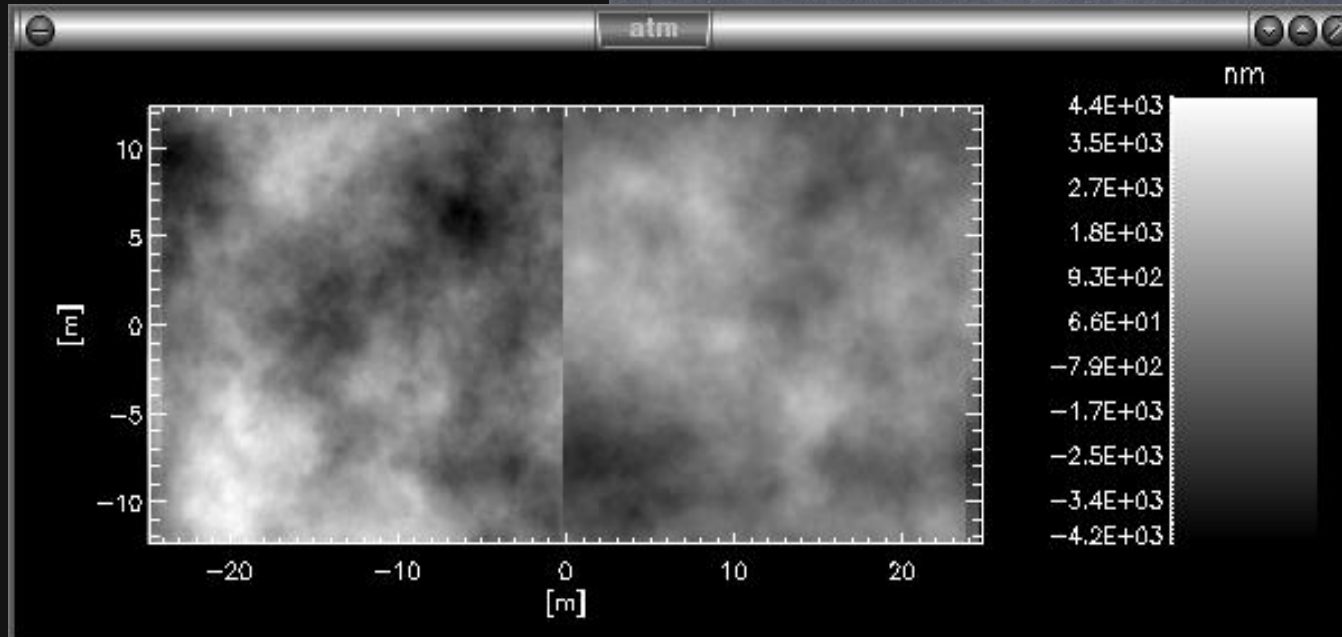
The obtained phase screen is thus numerically written

$$\varphi_L(i,j) = \sqrt{2}\sqrt{0.0228}\left(\frac{L}{r_0}\right)^{\frac{5}{6}} \left\{ \mathrm{FFT}^{-1}\left[\left(k^2 + l^2 + \left(\frac{L}{\mathcal{L}_0}\right)^2\right)^{-\frac{11}{12}} \exp\{\imath\theta(k,l)\}\right]\right\}, \qquad (3)$$

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*, $\imath$ 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.

Images & turbulence — 18

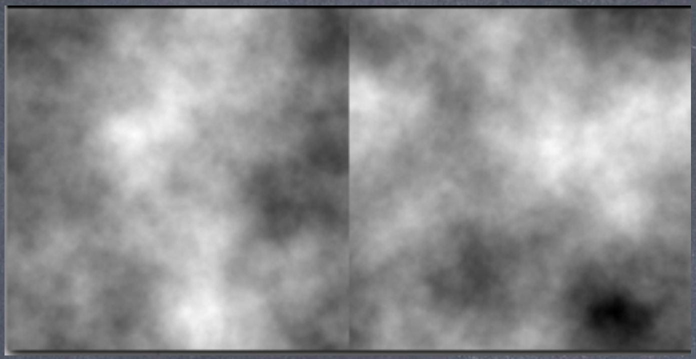
```
function wfgeneration, dim, length, L0, r0, lambda, SEED=seed
;
; wave-front (wf) generation following von Karman model
; (infinite L0 -Kolmogorov model- not allowed here).
;
; dim    = wf linear dimension [px],
; length = wf physical length [m],
; L0     = wf outer-scale [m],
; seed   = random generation seed (OPTIONAL),
; r0     = Fried parameter at wavelength 'lambda' [m],
; lambda = wavelength at which r0 is defined.
;
; Marcel Carbillet [marcel.carbillet@unice.fr],
; lab. Lagrange (UCA, OCA, CNRS), Feb. 2013.
;
; Last modification: Feb. 2018.
;
phase = (randomu(seed,dim,dim)-.5) * 2*!PI    ; rnd uniformly distributed phase
                                              ; (between -PI and +PI)
rr = dist(dim)
modul = (rr^2+(length/L0)^2)^(-11/12.)        ; von Karman model

screen = fft(modul*exp(complex(0,1)*phase), /INVERSE)
                                              ; compute wf
screen *= sqrt(2)*sqrt(.0228)*(length/r0)^(5/6.)*lambda/(2*!PI)
                                              ; proper normalization of wf
screen -= mean(screen)                        ; force mean to zero

return, screen                                ; deliver 2 independent wf:
                                              ; float(screen) & imaginary(screen)
end
```



<u>wf generation:</u>
generate a cube of statistically independent wf (typically 100)...
=> compute mean *rms* for different $[r_0, L_0]$

Images & turbulence — 19

```
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>
```

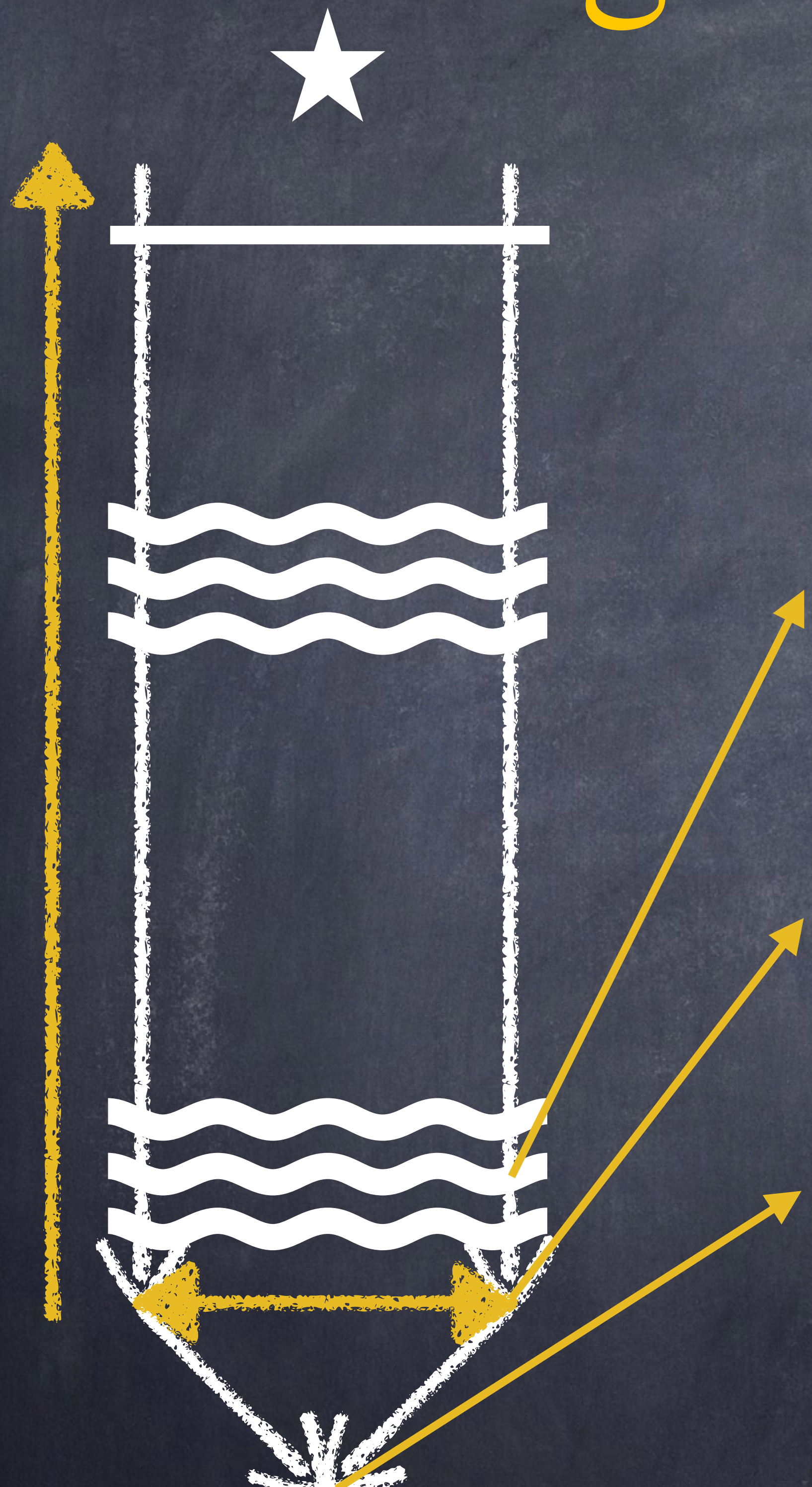


```
IDL> .rn wfcube
% Compiled module: WFCUBE.
IDL> print, wfcube(128L,2.,27.,.1,500E-9,100L)*1E9
% Compiled module: COMPUTE_RMS.
      367.668
% Program caused arithmetic error: Floating underflow
IDL>
```

```
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
```

Images & turbulence — 20



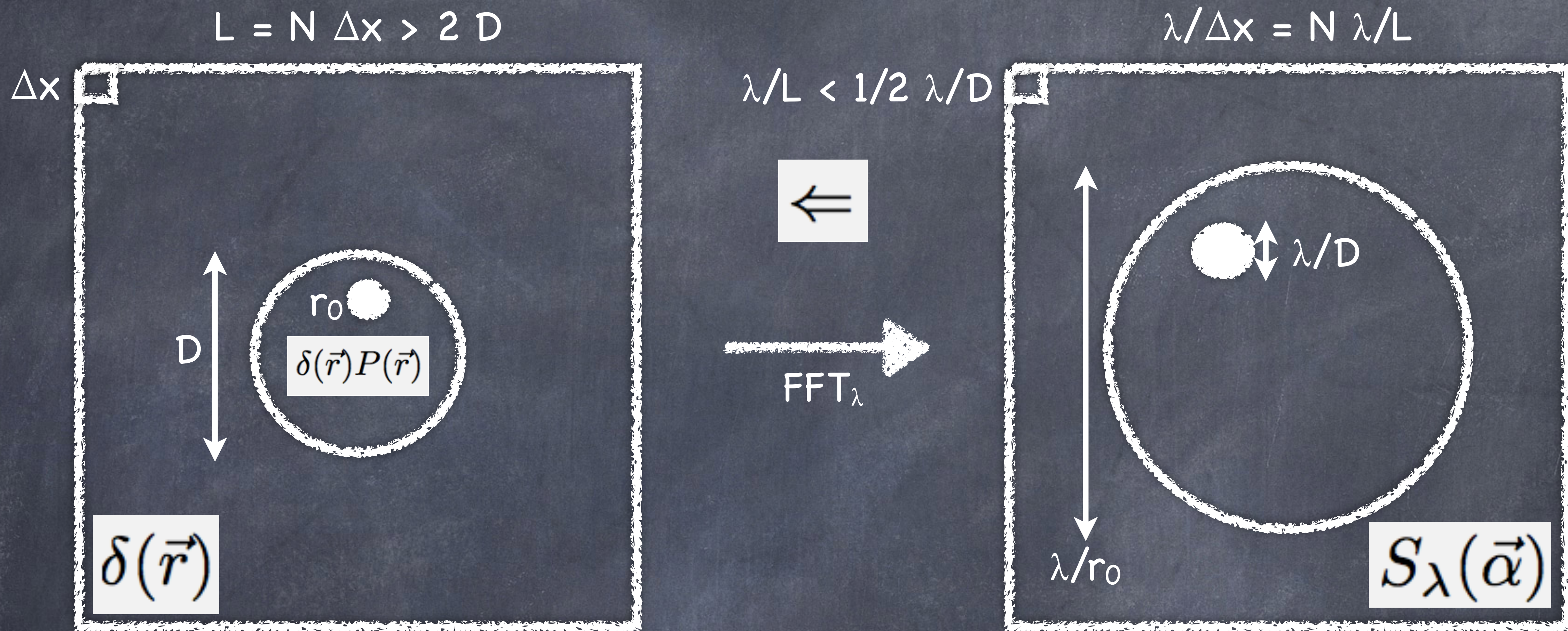
$$\Psi(\vec{r}) = A \ \exp\left(\imath\Phi(\vec{r})\right)$$

$$P(\vec{r}) \Rightarrow A \ P(\vec{r}) \ \exp\left(\imath\Phi(\vec{r})P(\vec{r})\right)$$

$$S_\lambda(\vec{\alpha}) \propto \|FT\{A \ P(\vec{r}) \ \exp\left(\imath\Phi(\vec{r})P(\vec{r})\right)\}\|^2$$

$$A = 1 \text{ and } \Phi(\vec{r}) = \frac{2\pi}{\lambda}\delta(\vec{r}) \Rightarrow S_\lambda(\vec{\alpha}) \propto \|FT\{P(\vec{r}) \ \exp\left(\imath\frac{2\pi}{\lambda}\delta(\vec{r})P(\vec{r})\right)\}\|^2$$

Images & turbulence — 21



Shannon (=Nyquist) criterium

=> the image pixel λ/L must be at most half the resolution element (resel!) λ/D
(in other words : one must have AT LEAST 2 image pixels per λ/D)

=> the simulated wavefronts must be at least twice the telescope diameter (L>2D)

In addition

- λ/r_0 should be smaller than λ/Δx (=> N large enough)

Images & turbulence — 22

```
function wfimg, dim, length, L0, r0, lambda_r0, obs, diam, lambda_psf, n_psf, filename
;
; use:
; dim        = 128L      ; [px] wf dimension
; length     = 2.        ; [m] wf physical dimension
; L0         = 27.       ; [m] outerscale
; r0         = .1        ; [m] Fried parameter
; lambda_r0 = 500E-9     ; [m] r0 wavelength
; obs        = 0. [0-1]  ; (linear) obscuration ratio
; diam       = dim/2     ; [px] telescope pupil dimension
; lambda_psf= 500E-9     ; [m] PSF wavelength
; n_psf      = 100L      ; nb of generated statistically independent PSFs
; filename   = 'cube.sav'; cube of PSFs filename
;
; print, wfimg(dim,length,L0,r0,lambda_r0,obs,diam,lambda_psf,n_psf,filename)
;
; sub-routines needed: image.pro, wfgeneration.pro, makepup.pro
;
; Marcel Carbillet [marcel.carbillet@unice.fr], Lagrange (UCA, OCA, CNRS), Feb. 2018.
;
cube = fltarr(dim,dim,n_psf)

for i=0, n_psf/2-1L do begin
  dummy = wfgeneration(dim,length,L0,r0,lambda_r0,SEED=seed)
  wf1   = float(dummy)
  wf2   = imaginary(dummy)
  dummy = makepup(dim,diam,obs)
  img1  = image(dummy,wf1,lambda_psf)
  img2  = image(dummy,wf2,lambda_psf)
  cube[*,*,2*i]   = img1
  cube[*,*,2*i+1] = img2
endfor

save, cube, FILENAME=filename

return, 'Cube of PSFs '+filename+' saved on disk...'
end
```

image formation:

1- cube of instantaneous PSFs (500nm & H-band)

```
function image, pup, wf, lambda
;
; image computation from a wavefront
;
; pup    = pupil,
; wf     = wavefront [float],
; lambda = wavelength at which image is computed.
;
; Marcel Carbillet [marcel.carbillet@unice.fr],
; UMR 7293 Lagrange (UNS/CNRS/OCA), Feb. 2013.
;
; Last modification: Feb. 2019
;
dim = (size(wf))[1]
img = (abs(fft(pup*exp(complex(0,1)*2*!PI/lambda*wf*pup))))^2
; NB: (abs(fft(pup*exp(complex(0,1)*2*!PI/lambda*wf))))^2 would suffice
img = shift(temporary(img), dim/2, dim/2)
; NB: shift(img, dim/2, dim/2) OK too

return, img
end
```

```
IDL> .r wfimg
% Compiled module: WFIMG.
IDL> print, wfimg(128L,2.,27.,0.1,500E-9,0.,64L,500E-9,100L,'cube.sav')
Cube of PSFs cube.sav saved on disk...
```

Images & turbulence — 23

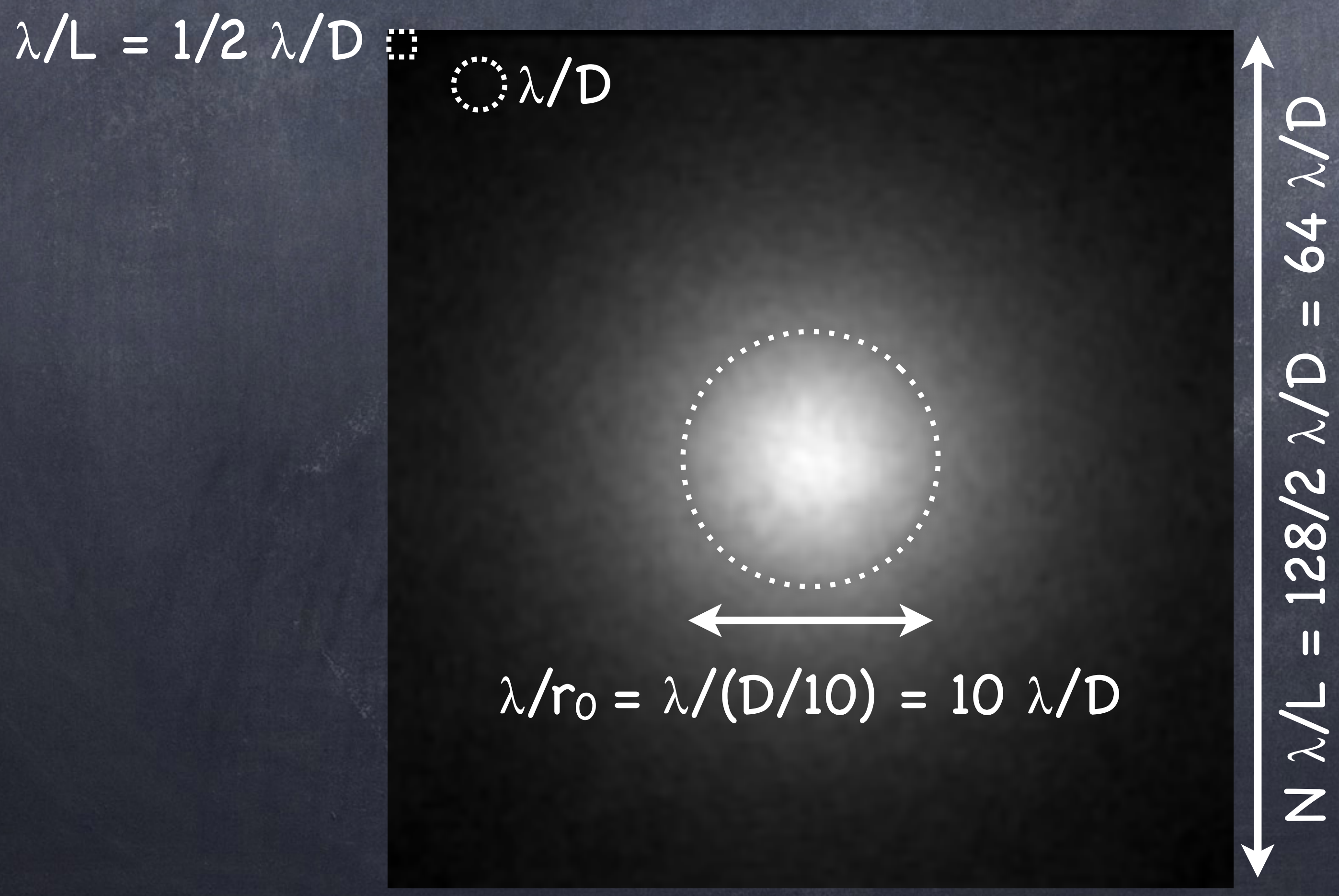
```
IDL> restore, 'cube.sav'
IDL> help
% At $MAIN$
CUBE            FLOAT     = Array[128, 128, 100]
Compiled Procedures:
    $MAIN$

Compiled Functions:
  COMPUTE_RMS DIST         IMAGE        MAKEPUP      WFCUBE
    WFGENERATION        WFIMG

IDL> for i=0,99 do tvscl, cube[*,*,i]
```

```
IDL> longexp = total(cube, 3)
IDL> tvscl, longexp^.1
```

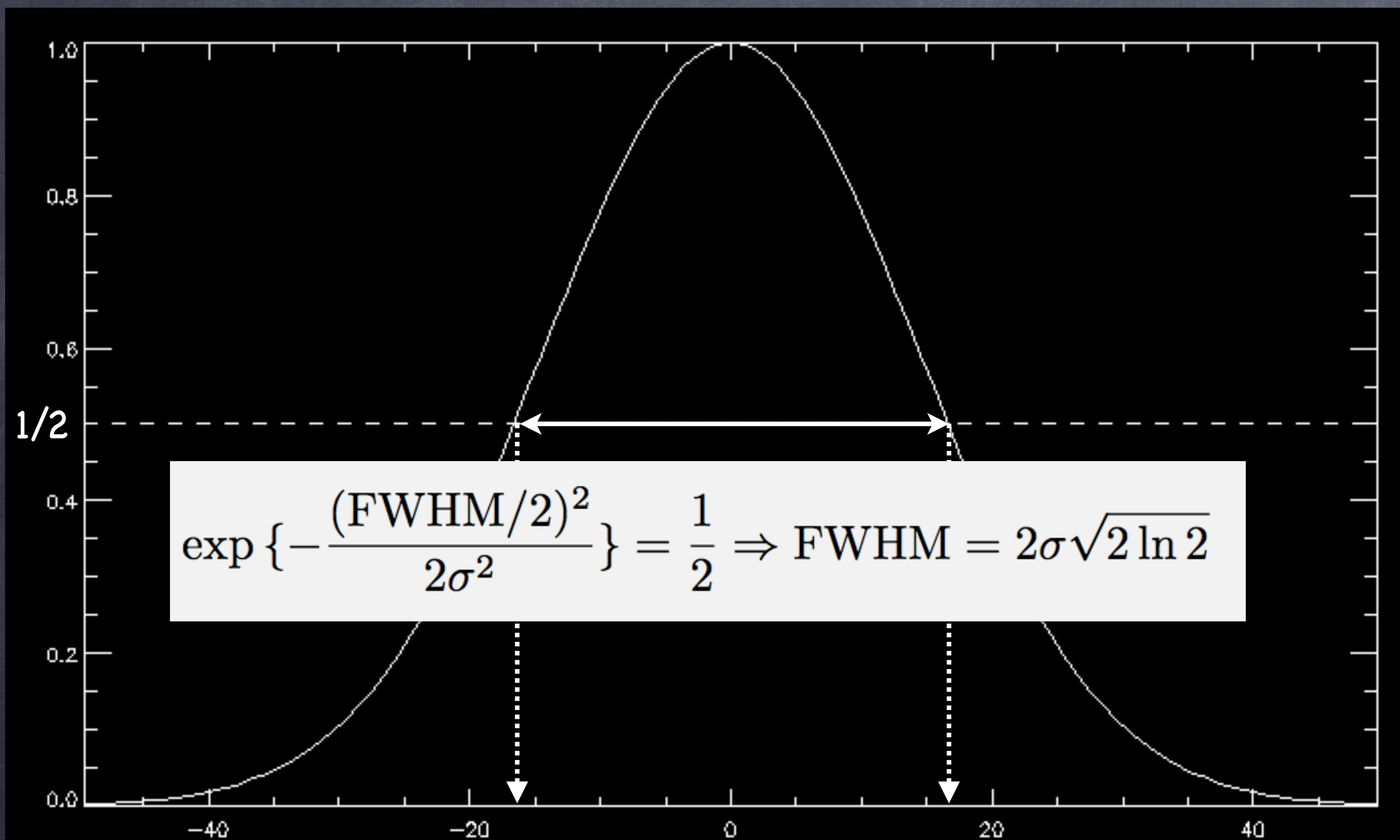
<u>image formation:</u>

1- cube of instantaneous PSFs (500nm & H-band)

2- long-exposure PSF



Images & turbulence — 24



$$\exp\{-\frac{(\mathrm{FWHM}/2)^2}{2\sigma^2}\} = \frac{1}{2} \Rightarrow \mathrm{FWHM} = 2\sigma\sqrt{2\ln 2}$$

<u>image formation:</u>

1- cube of instantaneous PSFs (500nm & H-band)

2- long-exposure PSFs

3- fit with gaussian and compare FWHM vs. λ/r_0 (seeing), also in function of the outerscale L_0.

-> Also read Martinez...

```
IDL> restore, 'cube.sav'
IDL> longexp=total(cube,3)
IDL> tvscl, longexp
IDL> res=gauss2dfit(longexp,a)
% Program caused arithmetic error: Floating underflow
IDL> print, 2*(a[2]+a[3])/2*sqrt(2*alog(2))
      15.5423
```

In this example, the FWHM is ~15.54px and, since we have here: 1px=(λ/D)/2, we have hence: FWHM≈7.77 (λ/D) [i.e. 7.77/10≈0.78 arcsec here (λ=500nm)]