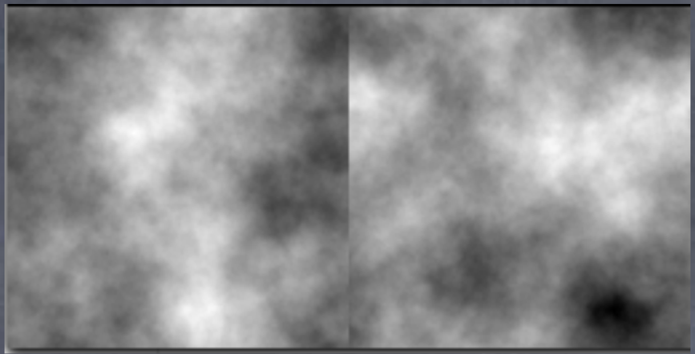
Images & turbulence - 17

```
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>
```



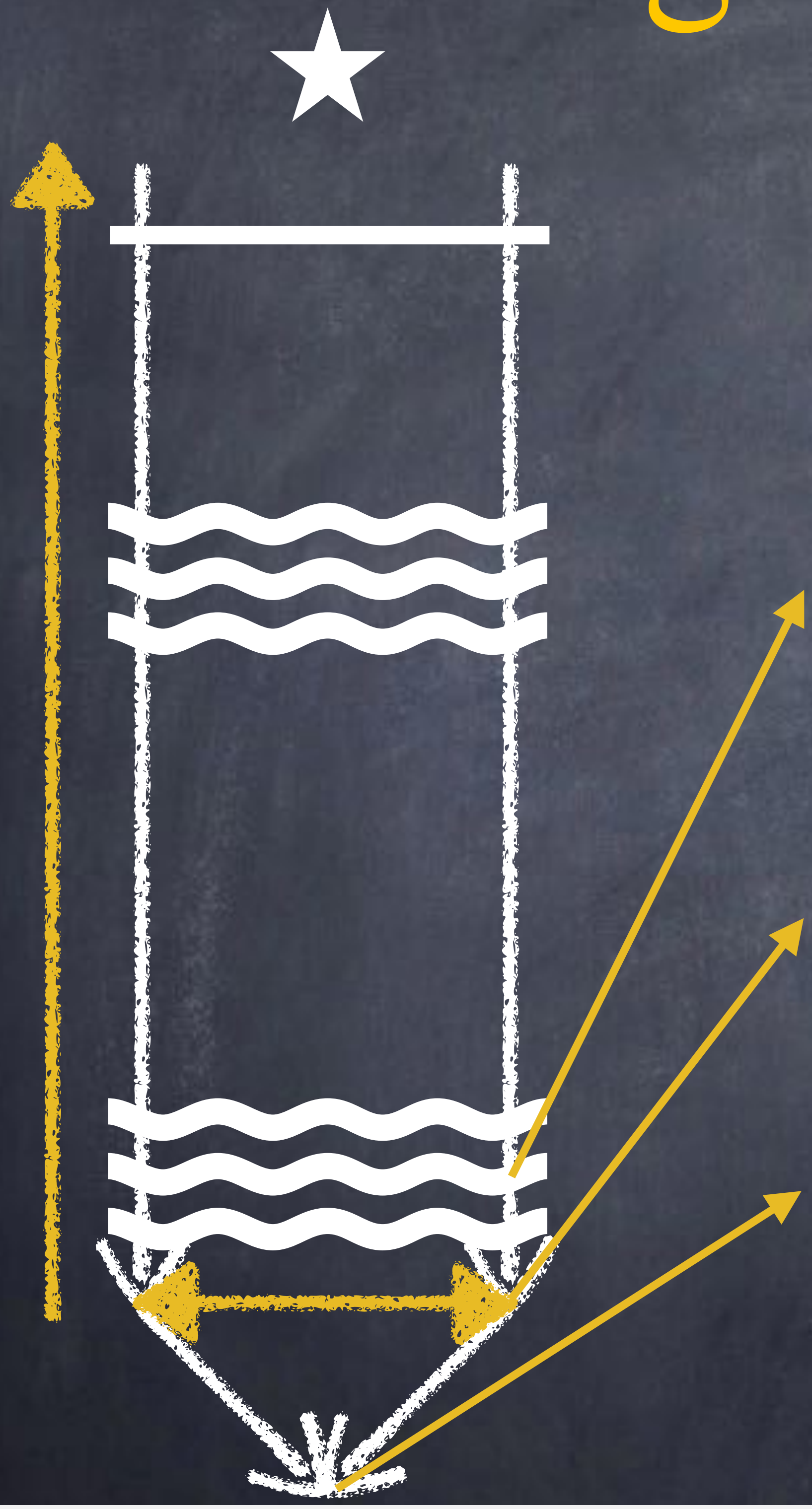
```
function wfcube, dim, length, L0, r0, lambda, n_wf
;
; use:
; dim       = 128L      ; [px] wf dimension
; length    = 2.        ; [m] wf physical dimension
; L0        = 27.       ; [m] outerscale
; r0        = .1        ; [m] Fried parameter
; lambda    = 500E-9    ; [m] r0 wavelength
; n_wf      = 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
```

```
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 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
```

-> Also read Carbillet & Riccardi (introduction of Sec. 2)...

Images & turbulence - 18



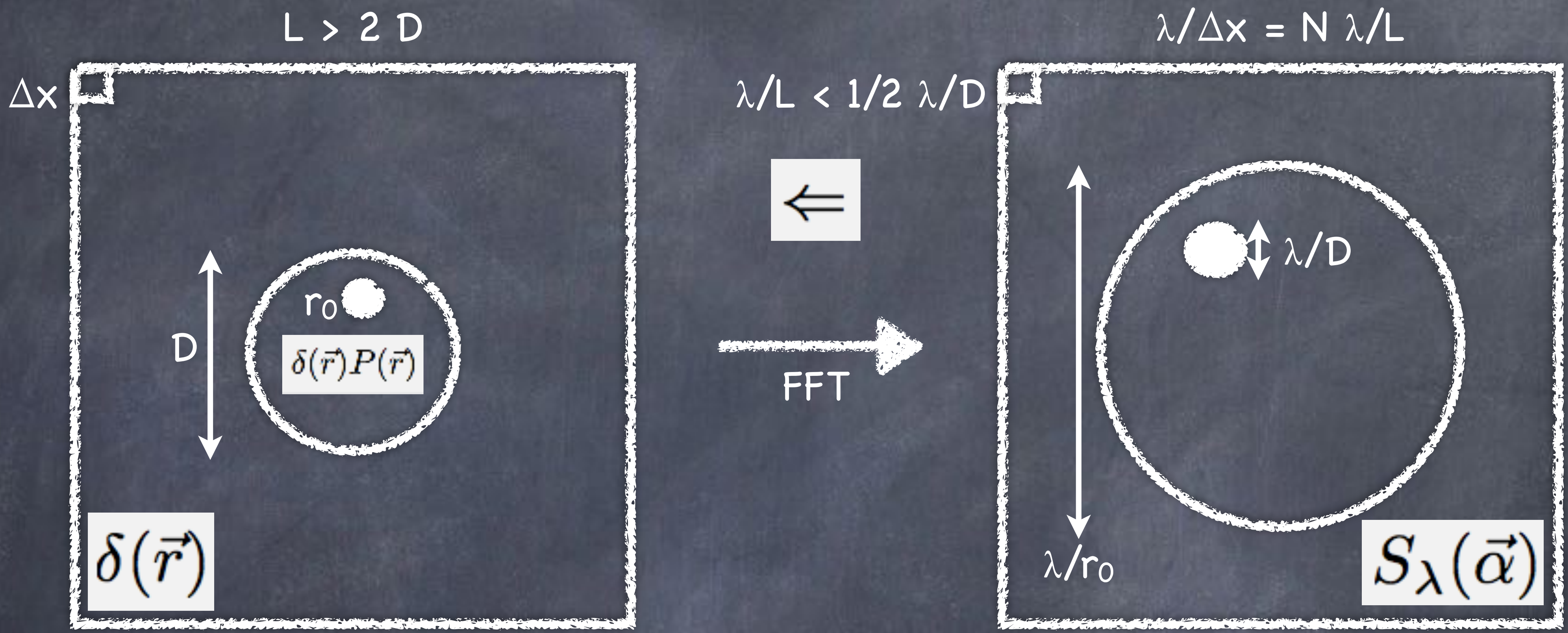
$$\Psi(\vec{r}) = A\ \exp\left(\imath\Phi(\vec{r})\right)$$

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

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

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

Images & turbulence - 19



Shannon (=Nyquist) criterium

=> the image pixel λ/L must be at most half the resolution element (resel!) λ/D

(in other words : one must has 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 $\lambda/\Delta x$ (=> N large enough)

Images & turbulence - 20

```
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 (2 wavelengths)

```
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), February 2013.
;
; Last modification: Feb. 2014
;
dim = (size(wf))[1]
img = (abs(fft(pup*exp(complex(0,1)*2*!PI/lambda*wf*pup))))^2
img = shift(temporary(img), dim/2, dim/2)

return, img
end
```

Images & turbulence - 20

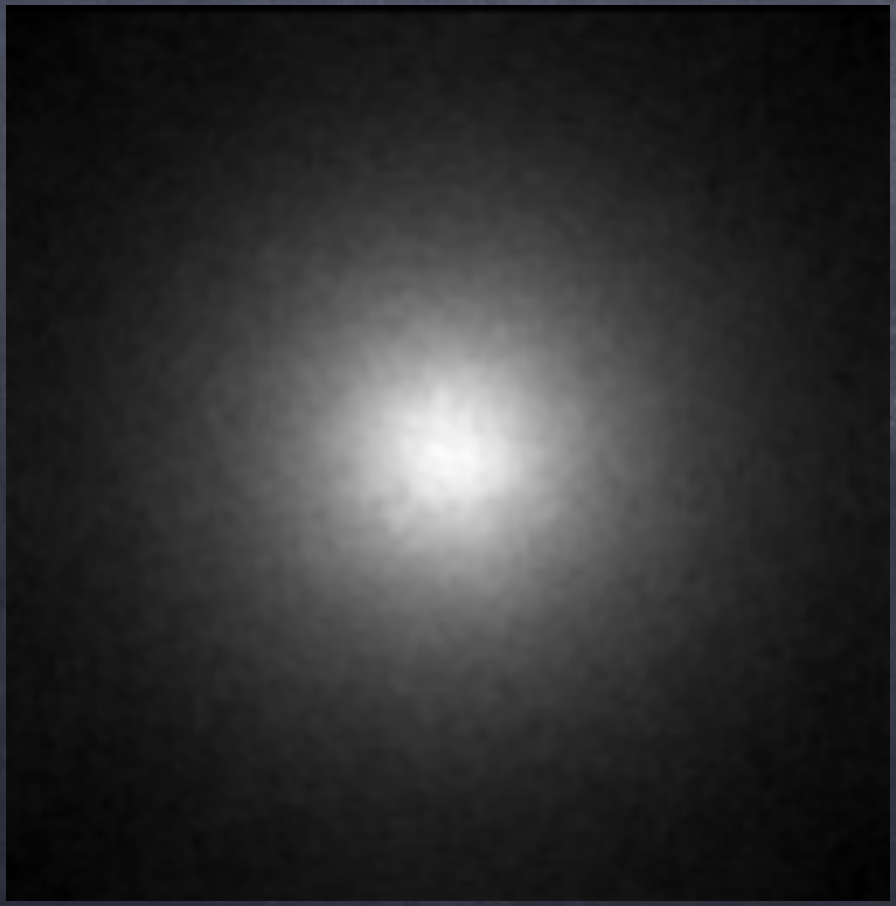
```
IDL> restore, 'cube.sav'
IDL> help
% At $MAIN$
CUBE              FLOAT     = Array[128, 128, 100]
Compiled Procedures:
    $MAIN$

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
```

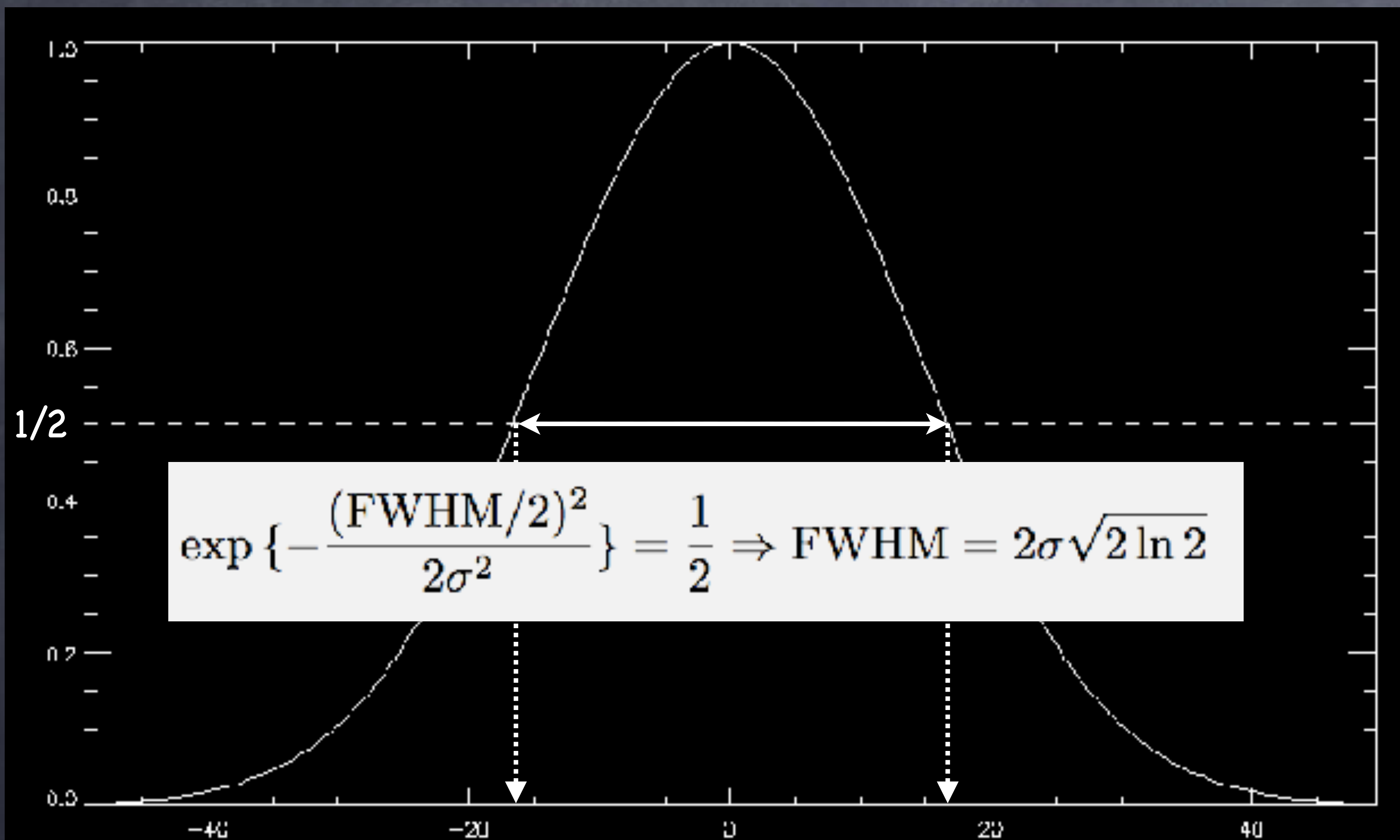
<u>image formation:</u>

1- cube of instantaneous PSFs (2 wavelengths)

2- long-exposure PSF



Images & turbulence - 21



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

<u>image formation:</u>
1- cube of instantaneous PSFs (2 wavelengths)
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
```