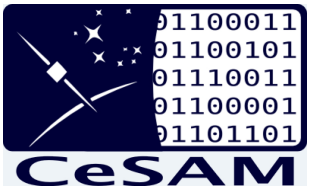

ETC-42 : Universal Exposure Time Calculator

Calculation Method

Laboratoire d'Astrophysique de Marseille
Centre donneeS Astrophysique de Marseille

Authors:
N. Apostolakos,
P.Y. Chabaud,
C. Surace

January 2012

Contents

1 Introduction

This document describes in detail the method used by the ETC for calculating the exposure time and the signal to noise ratio. Even though the document is focused on the software parts responsible for performing the calculation, this document can also be used as a reference by anyone interested in the formula the ETC uses for its calculations in a theoretical level.

The SNR calculation performed by the ETC-42 has been broken down to smaller calculations, depending the configuration of the simulation. Each of the following sections describes one of the smaller calculations, together with the different methods to perform the calculation according the given configuration.

2 Exposure Time

The exposure time calculator is responsible for calculating the exposure time for achieving a specified central pixel or total SNR. The different calculation methods are all based on the same equation:

$$SNR_{ref} = \frac{S_{(\lambda_{ref})} * T}{\sqrt{S_{(\lambda_{ref})} * T + BN_{(\lambda_{ref})} * T + N_{pix} * (D * T + n_{exp} * R^2)}} \quad (1)$$

where
T: is the exposure time to be calculated
SNR_{ref}: is the fixed signal to noise ratio (given by the user)
λ_{ref}: is the wavelength for which the SNR is given, expressed in Å(given by the user)
$S_{(\lambda_{ref})} = S_{sim(\lambda_{ref})} + S_{extra(\lambda_{ref})}$
S_{sim}: is the simulated signal (see section 7)
S_{extra}: is the extra signal (see section 6)
$BN_{(\lambda_{ref})} = BN_{sim(\lambda_{ref})} + BN_{extra(\lambda_{ref})}$
BN_{sim}: is the simulated background noise (see section 5)
BN_{extra}: is the extra background noise (see section 4)
N_{pix}: is the number of CCD pixels affected by the target light (see section 22)
D: is the dark current expressed in $e^-/s/pixel$ (given by the user)
n_{exp}: is the number of exposures (given by the user)
R: is the readout noise expressed in $e^-/pixel$ (given by the user)

2.1 DIT - Number of exposures

The user can select one of two options, give the number of exposures directly or give the detector integration time (DIT). If the user gives the detector integration time (DIT), then the n_{exp} on equation 1 is replaced with:

$$n_{exp} = \frac{T}{DIT} \quad (2)$$

2.2 Imaging

In the case of imager simulation, the SNR_{ref} given by the user is the signal to noise ratio for the total filter wavelength range. In this case, the $S_{(\lambda_{ref})}$ and $BN_{(\lambda_{ref})}$ of the equation 1 are calculated by integrating the signal (see sections 7 and 6) and the background noise (see sections 5 and 4) on the filter wavelength range. These values are then replaced in the equation 1 and it is solved for T.

2.3 Spectroscopy

In the case of spectrograph simulation, the SNR_{ref} given by the user is the signal to noise ratio for the spectral element around the λ_{ref}. Note that the signal and the background noise are considered constant inside the spectral element, so the $S_{(\lambda_{ref})}$ and $BN_{(\lambda_{ref})}$ of the equation 1 are calculated as:

$$\begin{aligned} S &= (S_{sim(\lambda_{ref})} + S_{extra(\lambda_{ref})}) * \Delta\lambda \\ BN &= (BN_{sim(\lambda_{ref})} + BN_{extra(\lambda_{ref})}) * \Delta\lambda \end{aligned} \tag{3}$$

3 Signal to Noise Ratio (SNR)

The SNR (Signal to Noise Ratio) calculator is responsible for calculating the central pixel and total SNR for a fixed exposure time. The different calculation methods are all based on the same equation:

$$SNR = \frac{(S_{sim} + S_{extra}) * T}{\sqrt{(S_{sim} + S_{extra}) * T + (BN_{sim} + BN_{extra}) * T + N_{pix} * (D * T + n_{exp} * R^2)}} \tag{4}$$

where
SNR: is the signal to noise ratio to be calculated
T: is the exposure time expressed in s (given by the user)
S_{sim}: is the simulated signal (see section 7)
S_{extra}: is the extra signal (see section 6)
BN_{sim}: is the simulated background noise (see section 5)
BN_{extra}: is the extra background noise (see section 4)
N_{pix}: is the number of CCD pixels affected by the target light (see section 22)
D: is the dark current expressed in $e^-/s/pixel$ (given by the user)
n_{exp}: is the number of exposures (given by the user)
R: is the readout noise expressed in $e^-/pixel$ (given by the user)

Note that for the central pixel SNR calculation the N_{pix} is always 1. Also the extra signal and background functions (see sections 6 and 4) are returning values for the total pixel coverage, so their values are divided with the N_{pix} before being used for the central pixel SNR calculation.

3.1 DIT - Number of exposures

The user can select one of two options, give the number of exposures directly or give the detector integration time (DIT). If the user gives the detector integration time (DIT), then the n_{exp} on equation 4 is replaced with:

$$n_{exp} = \left\lceil \frac{T}{DIT} \right\rceil \tag{5}$$

3.2 Imaging

In the case of imager simulation, the ETC-42 gives as result the SNR for the total filter wavelength range (both for the central pixel and for the affected by the source area). In this case, the S and BN of the equation 4 are calculated by integrating the signal (see sections 7 and 6) and the background noise (see sections 5 and 4) on the filter wavelength range. These values are then replaced in the equation 4 and is calculated the SNR for all the filter range.

3.3 Spectroscopy

In the case of spectrograph simulation, the ETC-42 gives as result an array of SNR values, one for each wavelength step of $\Delta\lambda$ (see section 8) covering the total wavelength range (see section 28). Note that the signal and the background noise are considered constant for each step, so the S and BN of the equation 4 are calculated as:

$$\begin{aligned} S &= (S_{sim(\lambda_c)} + S_{extra(\lambda_c)}) * \Delta\lambda \\ BN &= (BN_{sim(\lambda_c)} + BN_{extra(\lambda_c)}) * \Delta\lambda \end{aligned} \tag{6}$$

where λ_c is the central wavelength of the simulation step.

4 Extra BackgroundNoise

The extra background noise calculator is responsible for computing a function representing the number of electrons ($S_{extra(\lambda)}$) per second and for a specific wavelength, to be added to the simulated background noise (see section 5). The calculator does not require any input and its result is a function which can be used for calculating the extra background noise for given wavelength values (expressed in $e^-/s/\text{Å}$). Note that this calculator is useful only for spectrograph simulations and it should not be used for imager simulations.

4.1 No Extra Background Noise

This method is used in the case the user has selected the option to use only the simulated background noise and to ignore the provided extra one. In this case the extra background noise is always zero.

4.2 Extra Background Noise Profile

This method uses a profile given by the user for the extra background noise. In this case the values are calculated either using linear interpolation or by treating the profile as a template, based on the selection of the user. This has as result the following restrictions:

- The x values of the profile must be expressed in Å
- The y values of the profile must be expressed in $e^-/s/$Å
- The method cannot calculate the extra background noise for wavelengths outside the range defined from the profile

5 Simulated Background Noise

The simulated background noise calculator is responsible for computing the number of the detected by the instrument electrons ($BN_{sim(\lambda)}$) from the simulated background, per second and for a specific wavelength, expressed in $e^-/s/$Å, for the total spatial binning (see section 25).

5.1 No Simulated Background Noise

This method is used in the case the user has selected the option to use only the extra background noise provided and to ignore the simulated one. In this case the simulated background noise is always zero.

5.2 Background Noise from Flux

This method uses the simulated background flux (see section 12) and it takes in consideration the size of the primary mirror and any losses before the light reach the CCD. The equation used is:

$$BN_{sim(\lambda)} = \phi_{(\lambda)} * \varepsilon_{(\lambda)} * Filter_{(\lambda)} * A_{tel} * A_{sky(\lambda)} * \frac{\lambda}{h * c} \tag{7}$$

where:
$BN_{sim(\lambda)}$: is the calculated background noise expressed in $e^-/s/$Å
$\phi_{(\lambda)}$: is the background flux expressed in $erg/s/cm^2/$Å$/arcsec^2$ (see section 12)
$\varepsilon_{(\lambda)}$: is the total system efficiency, excluding the filter (see section 29)
$Filter_{(\lambda)}$: is the filter response (see section 30)
A_{tel}: is the total effective primary mirror area (see section 11)
$A_{sky(\lambda)}$: is the area of the sky contributing to the background (see section 16)
λ: is the wavelength the signal is calculated for, expressed in Å
h: is the Planck constant expressed in $erg * s$
c: is the speed of light expressed in Å$/sec$

5.3 Slitless Spectrograph

For a slitless spectrograph all the background spectrum range contributes to every point on the CCD. This results in a constant background noise value (expressed in $e^-/s/arcsec^2$), which can be used to calculate the background noise based on the area the covered by the affected pixels. The background noise is calculated as:

$$BN_{sim(\lambda)} = Factor * \int_{min}^{max} \phi_{(\lambda)} * \varepsilon_{(\lambda)} * Filter_{(\lambda)} * A_{tel} * \frac{\lambda}{h * c} \, \mathrm{d}\lambda \tag{8}$$

where:
$BN_{sim(\lambda)}$: is the calculated background noise expressed in $e^-/s/$Å
$\phi_{(\lambda)}$: is the background flux expressed in $erg/s/cm^2/$Å$/arcsec^2$ (see section 12)
$\varepsilon_{(\lambda)}$: is the total system efficiency, excluding the filter (see section 29)
$Filter_{(\lambda)}$: is the filter response (see section 30)
$[min, max]$: is the range where $\varepsilon_{(\lambda)}$ and $Filter_{(\lambda)}$ have non zero values
A_{tel}: is the total effective primary mirror area (see section 11)
λ: is the wavelength the signal is calculated for, expressed in Å
h: is the Planck constant expressed in $erg * s$
c: is the speed of light expressed in Å$/sec$
$Factor$: is a factor containing the area information and the convertion to per Å

$$Factor = \begin{cases} \dfrac{N_{pix(\lambda)} * p^2}{\Delta\lambda_{(\lambda)}} & \text{, for the total spatial binning} \\ p^2 * \dfrac{R_{pix(\lambda)}}{\lambda} & \text{, for the central pixel} \end{cases} \tag{9}$$

where:
$N_{pix(\lambda)}$: is the number of affected pixels (see section 22)
p: is the pixel scale expressed in $arcsec/pixel$ (given by the user)
$\Delta\lambda_{(\lambda)}$: is the wavelength step of the calculation (see section 8)
$R_{pix(\lambda)}$: is the spectral resolution per pixel (see section 10)

6 Extra Signal

The extra signal calculator is responsible for computing a function representing the number of electrons ($S_{extra(\lambda)}$) per second and for a specific wavelength, to be added to the simulated signal (see section 7). The calculator does not require any input and its result is a function which can be used for calculating the extra signal for given wavelength values (expressed in $e^-/s/$Å). Note that this calculator is useful only for spectrograph simulations and it should not be used for imager simulations.

6.1 No Extra Signal

This method is used in the case the user has selected the option to use only the simulated signal and to ignore the provided extra one. In this case the extra signal is always zero.

6.2 Extra Signal Profile

This method uses a profile given by the user for the extra signal. In this case the values are calculated either using linear interpolation or by treating the profile as a template, based on the selection of the user. This has as result the following restrictions:

- The x values of the profile must be expressed in Å
- The y values of the profile must be expressed in $e^-/s/$Å
- The method cannot calculate the extra signal for wavelengths outside the range defined from the profile

7 Simulated Signal

The simulated signal calculator is responsible for computing the number of the detected by the instrument electrons ($S_{sim(\lambda)}$) from the simulated observed object, per second and for a specific wavelength, expressed in $e^-/s/$Å. It return two values, one for the central pixel and one for the total area covered by the source.

7.1 No Simulated Signal

This method is used in the case the user has selected the option to use only the extra signal provided and to ignore the simulated one. In this case the simulated signal is always zero.

7.2 Signal from Flux

This method uses the simulated source flux (see section 18) and it takes in consideration the size of the primary mirror and any losses before the light reach the CCD. The equation used is:

$$S_{sim(\lambda)} = F_{(\lambda)} * C_{(\lambda)} * \xi_{(\lambda)} * \varepsilon_{(\lambda)} * Filter_{(\lambda)} * A_{tel} * \frac{\lambda}{h * c} \tag{10}$$

where:
$S_{sim(\lambda)}$: is the calculated signal expressed in $e^-/s/$Å
$F_{(\lambda)}$: is the source flux expressed in $erg/s/cm^2/$Å(see section 18)
$C_{(\lambda)}$: is the normalization factor (see section 19)
$\xi_{(\lambda)}$: is the atmospheric transmission (see section 17)
$\varepsilon_{(\lambda)}$: is the total system efficiency, excluding the filter (see section 29)
$Filter_{(\lambda)}$: is the filter response (see section 30)

A_{tel}: is the total effective primary mirror area (see section 11)
λ: is the wavelength the signal is calculated for, expressed in Å
h: is the Planck constant expressed in $erg * s$
c: is the speed of light expressed in $Å/sec$

Note that this equation is used for both electrons on the central pixel and on the total area covered by the source, by using the different values returned from the normalization factor calculator.

8 Wavelength step $\Delta\lambda$

The $\Delta\lambda$ calculator is responsible for computing the wavelength steps size (expressed in Å) for a spectrograph simulation.

8.1 Imaging

This method is used in the case of imager simulation. In this case the $\Delta\lambda$ is always considered to be 1 Å.

8.2 Spectroscopy

This method is used in the case of spectrograph simulation. In this case the $\Delta\lambda$ is calculated by the equation:

$$\Delta\lambda_{(\lambda)} = \frac{\lambda}{R_{pix(\lambda)}} * Pixels \tag{11}$$

where:
$\Delta\lambda_{(\lambda)}$: is the size of the wavelength step of the simulation (expressed in Å)
λ: is the wavelength expressed in Å
$R_{pix(\lambda)}$: is the spectral resolution per pixel (see section 10)
$Pixels$: is the number of pixels used for determining the size of the wavelength step

$$Pixels = \begin{cases} 1 & \text{, for } sq = \text{Spectral Pixel} \\ SpectralBinning_{(\lambda)} & \text{, for } sq = \text{Spectral Resolution Element} \end{cases} \tag{12}$$

where:
$SpectralBinning_{(\lambda)}$: is the spectral binning (see section 24)
sq: is the Spectral Quantum (given by the user)

9 Spectral Convolution Kernel Size

The spectral convolution kernel size calculator is responsible for computing the size of the kernel (expressed in Å) used to perform the convolution of all the spectra used.

9.1 Imaging

This method is used in the case of imager simulation. In this case the size of the convolution is always considered to be 0 Å.

9.2 Spectroscopy

This method is used in the case of spectrograph simulation. In this case the convolution kernel size is calculated by the equation:

$$CK_{size(\lambda)} = \frac{\lambda}{R_{pix(\lambda)}} * Pixels \tag{13}$$

where:
$CK_{size(\lambda)}$: is the size of the kernel
λ: is the wavelength expressed in Å
$R_{pix(\lambda)}$: is the spectral resolution per pixel (see section 10)
$Pixels$: is the number of pixels used for determining the size of the kernel

$$Pixels = \begin{cases} 1 & \text{, for } sq = \text{Spectral Pixel} \\ SpectralBinning_{(\lambda)} & \text{, for } sq = \text{Spectral Resolution Element} \end{cases} \tag{14}$$

where:
$SpectralBinning_{(\lambda)}$: is the spectral binning (see section 24)
sq: is the Spectral Quantum (given by the user)

Note that the $Pixels$ can be forced to always take the value of the spectral binning by setting the "Force convolution per spectral resolution element" option in the calculator options. By default this configuration is enabeld as the convolution should not be depended by the spectral quantum of the calculation. This is the only difference between this calculator and the $\Delta\lambda$ calculator (see section 8) for spectrograph simulations.

10 Spectral Resolution per Pixel

The spectral resolution per pixel calculator is responsible for producing functions for computing the spectral resolution per pixel $R_{pix(\lambda)}$. The calculator does not require any input and its output is a function which can be used for calculating the spectral resolution (expressed in Å) for given wavelength values (expressed in Å).

10.1 Fixed $\Delta\lambda_{/pixel}$

This method is used when the user provides a fixed $\Delta\lambda_{/pixel}$. In this case the spectral resolution per pixel is:

$$R_{pix(\lambda)} = \frac{\lambda}{\Delta\lambda_{/pixel}} \tag{15}$$

where:
$R_{pix(\lambda)}$: is the spectral resolution per pixel
λ: is the wavelength expressed in Å
$\Delta\lambda_{/pixel}$: is the fixed value expressed in Å(given by the user)

10.2 Spectral Resolution Profile

This method is used when the user gives a profile for the spectral resolution $R_{pix(\lambda)}$. In this case the values are calculated using linear interpolation. This has as result the following restrictions:

- The x values of the template must be expressed in Å
- The y values of the template must be expressed in Å
- The method cannot calculate the spectral resolution for wavelengths outside the range defined from the profile

11 Telescope Area

The telescope area calculator is responsible for calculating the total area of the telescope mirror which reflects light, expressed in cm^2.

11.1 Circular Primary Mirror / Area Obstruction

This method is used in the case of a circular primary mirror. In this case the telescope area is calculated based on the diameter of the mirror, accounting also any obstruction of the light, by using the equation:

$$A_{tel} = \pi * \left(\frac{D_1}{2}\right)^2 * (1 - ob) \tag{16}$$

where:
D_1: is the diameter of the primary mirror expressed in centimeters (cm) (given by the user)
ob: is the total light obstruction (a number in the range [0,1])

NOTE: The obstruction is stored internally as a number in the range [0,1], but a user of the GUI should use % values (in the range [0,100]).

12 Background Flux

the background flux calculator is responsible for producing a function representing the total background flux $\phi_{(\lambda)}$. The calculator does not require any input and its result is a function which can be used for calculating the background flux (expressed in $erg/s/cm^2/Å/arcsec^2$) for given wavelength values (expressed in Å).

12.1 Ground

This method is used when a ground based telescope simulation is performed. In this case the total background noise is equal with the sky noise (see section 13):

$$\phi_{(\lambda)} = \phi_{sky(\lambda)} \tag{17}$$

where:
$\phi_{(\lambda)}$: is the total background flux
$\phi_{sky(\lambda)}$: is the sky flux (see section 13)

12.2 Space

This method is used when a space based telescope simulation is performed. In this case the total background noise is equal with the summary of the zodiacal and galactic noises:

$$\phi_{(\lambda)} = \phi_{zod(\lambda)} + \phi_{gal(\lambda)} \tag{18}$$

where:
$\phi_{(\lambda)}$: is the total background flux
$\phi_{zod(\lambda)}$: is the zodiacal flux (see section 14)
$\phi_{gal(\lambda)}$: is the galactic flux (see section 15)

13 Sky Flux

The sky flux calculator is responsible for producing a function representing the flux of the sky $\phi_{sky(\lambda)}$. The calculator does not require any input and its result is a function which can be used for calculating the sky flux (expressed in $erg/s/cm^2/\text{Å}/arcsec^2$) for given wavelength values (expressed in Å).

13.1 No Sky Noise

This method is used for simulations of space based telescopes. In this case the sky noise contribution is always zero.

13.2 Sky Template

This method is used when the user gives a template for the sky flux, with different options for the brightness of the night sky. In this case the flux is calculated according the values of the template representing the correct night sky. This has as result the following restrictions:

- The x values of the template must be expressed in Å
- The y values of the template must be expressed in $erg/s/cm^2/\text{Å}/arcsec^2$

- The method cannot calculate the flux for wavelengths outside the range defined from the template

14 Zodiacal Flux

The zodiacal flux calculator is responsible for producing a function representing the zodiacal flux $\phi_{zod(\lambda)}$ (the flux of the atmosphere). The calculator does not require any input and its result is a function which can be used for calculating the zodiacal flux (expressed in $erg/s/cm^2/\text{Å}/arcsec^2$) for given wavelength values (expressed in Å).

14.1 Flux Template

This method is used when the user gives a template for the zodiacal flux. In this case the flux is calculated according the values of the template. This has as result the following restrictions:

- The x values of the template must be expressed in Å
- The y values of the template must be expressed in $erg/s/cm^2/\text{Å}/arcsec^2$
- The method cannot calculate the flux for wavelengths outside the range defined from the template

14.2 No Zodiacal Noise

This method is used when the zodiacal noise is not contributing to the background noise. It always returns zero.

15 Galactic Flux

The galactic flux calculator is responsible for producing a function representing the galactic flux $\phi_{gal(\lambda)}$. The calculator does not require any input and its result is a function which can be used for calculating the galactic flux (expressed in $erg/s/cm^2/\text{Å}/arcsec^2$) for given wavelength values (expressed in Å).

15.1 Flux Template

This method is used when the user gives a template for the galactic flux. In this case the flux is calculated according the values of the template. This has as result the following restrictions:

- The x values of the template must be expressed in Å
- The y values of the template must be expressed in $erg/s/cm^2/\text{Å}/arcsec^2$

- The method cannot calculate the flux for wavelengths outside the range defined from the template

15.2 No Galactic Noise

This method is used when the galactic noise is not contributing to the background noise. It always returns zero.

16 Sky Area

The sky area calculator is responsible for calculating the area of the sky (in $arcsec^2$) that contributes to the background noise. The calculation depends on the wavelength λ, so the calculator gets as input the wavelength expressed in Åand returns two values, the sky area contributing to the central pixel and the total sky area (contributing to all the pixels affected by the source flux).

16.1 Imaging

This method is used in the case of imager simulation. In this case the background flux is directed directly to the CCD, so the sky area can be calculated as:

$$A_{sky(\lambda)} = N_{pix} * p^2, \;\; A_{sky/cp(\lambda)} = p^2 \tag{19}$$

where:
$A_{sky(\lambda)}$: is the total sky area
$A_{sky/cp(\lambda)}$: is the sky area for the central pixel
N_{pix}: is the number of pixels (see section 22)
p: is the pixel scale of the instrument expressed in $arcsec/pixel$

16.2 Slit

This method is used in the case of slit spectrograph simulation. In this case the calculator takes into consideration that any background flux not passing through the slit is lost. The equation for calculating the sky area is:

$$\begin{aligned} A_{sky(\lambda)} &= min(p * \lceil SpatialBinning_{(\lambda)} \rceil, l) * w \\ A_{sky/cp(\lambda)} &= min(p, l) * w \end{aligned} \tag{20}$$

where:
$A_{sky(\lambda)}$: is the total sky area
$A_{sky/cp(\lambda)}$: is the sky area for the central pixel
$SpatialBinning_{(\lambda)}$: is the spatial binning (see section 25)
p: is the pixel scale of the instrument expressed in $arcsec/pixel$ (given by the user)

w: is the width of the slit expressed in *arcsec* (given by the user)
l: is the length of the slit expressed in *arcsec* (given by the user)

16.3 Fiber

This method is used in the case of fiber spectrograph simulation. In this case the calculator takes into consideration that any background flux not passing through the fiber is lost. The equation for calculating the sky area is:

$$A_{sky(\lambda)} = \pi * \left(\frac{\phi_{fiber}}{2}\right)^2 \tag{21}$$

where:
$A_{sky(\lambda)}$: is the total sky area
ϕ_{fiber}: is the fiber diameter expressed in *arcsec* (given by the user)

The central pixel sky area is calculated using the ratio of the fiber area which falls in the central pixel.

17 Atmospheric Transmission

The atmospheric transmission calculator is responsible for producing functions which represent the total atmospheric transmission, $\xi_{(\lambda)}$. The atmospheric transmission is a number in the range [0, 1], which shows how much of the light is transmitted through the atmosphere.

17.1 Space

This method is used when the user selects the **Space** option for the **Site location**. In this case there is no atmosphere, so the atmospheric transmission is always one.

17.2 Absorption and Extinction Profiles

This method is used when the user selects the **Ground** option for the **Site location**. This method uses the profiles for the atmospheric absorption and for the atmospheric extinction, given by the user. The atmospheric transmission is calculated with the equation:

$$\xi_{(\lambda)} = \xi_{abs(\lambda)} * 10^{-0.4*X*k_{(\lambda)}} \tag{22}$$

where:
λ: is the wavelength the atmospheric transmission is calculated for, expressed in Å
$\xi_{(\lambda)}$: is the atmospheric transmission
$\xi_{abs(\lambda)}$: is the atmospheric absorption calculated with interpolation (given by the user)
X: is the air mass (given by the user)
$k_{(\lambda)}$: is the atmospheric extinction calculated with interpolation (given by the user)

The usage of the two profiles has a result the following restrictions:

- For both profiles the x values must be expressed in Å
- The method cannot make calculations outside the overlap of the two profiles

18 Source Flux

The source flux calculator is responsible for producing a function representing the flux of the source, $F_{(\lambda)}$. The calculator does not require any input and its result is a function which can be used for calculating the source flux for given wavelength values (expressed in Å).

Here it must be noted that the meaning of the source flux differs between point and extended sources. For point sources it represents the total flux of the target, when for extended sources it represents the flux value on the center of the surface brightness profile. The AB magnitude (m_{AB}) given by the user changes meaning accordingly.

18.1 Rest Frame Template

This method is used when the user gives a **template** for the **spectral distribution** of the source flux. In this case the spectral distribution is redshifted according the redshift given by the user and normalized by the flux calculated by the magnitude AB. This is done with the following steps:

- Convert the magnitude AB to flux by using the equation:

$$F_x = 10^{-\frac{m_{AB}+48.6}{2.5}} * \frac{c}{\lambda_x^2} \tag{23}$$

  where:
  m_{AB}: is the AB magnitude (given by the user)
  c: is the speed of light expressed in $Å/s$
  λ_x: is the reference wavelength of the AB magnitude selected by the user, expressed in Å

- Redshift the template given by the user. This has the restriction that the x values of the template must be expressed in Å. The y values can be expressed in anything, as they are going to be normalized by the calculated flux. The equation to calculate the wavelengths of the redshifted template is:

$$\lambda_{(z)} = \lambda_{(0)} * (1 + z) \tag{24}$$

  where:
  $\lambda_{(z)}$: is the redshifted wavelength (expressed in Å)
  $\lambda_{(0)}$: is the restframe wavelength (expressed in Å)
  z: is the redshift (given by the user)

- Normalize the redshifted template by the flux calculated earlier. This is done in two steps:
  - Calculate the value of the redshifted template for the central wavelength of the filter selected by the user (F_{0x}).

– Calculate the normalized values. For this the following equation can be used:

$$F_z = F_0 * \frac{F_x}{F_{0x}} \tag{25}$$

where:
F_z: is the normalized value
F_0: is the value of the redshifted template before the normalization
F_x: is the Flux at the central wavelength of the filter as calculated earlier
F_{0x}: is the value calculated on the previous step

18.2 Flat Photon Flux

This method is used when the user selects the **Flat Photon Flux** option for the **spectral distribution** of the sources flux. In this case the flux is constant over the wavelength and is calculated as:

$$F_{(\lambda)} = 10^{-\frac{m_{AB}+48.6}{2.5}} * \frac{c}{\lambda_x^2} \tag{26}$$

where:
m_{AB}: is the AB magnitude (given by the user)
c: is the speed of light expressed in $\text{Å}/s$
λ_x: is the reference wavelength of the AB magnitude selected by the user, expressed in Å

18.3 Flat Energy Flux

This method is used when the user selects the **Flat Energy Flux** option for the **spectral distribution** of the sources flux. In this case the flux is constant over the frequency and is calculated as:

$$F_{(\lambda)} = 10^{-\frac{m_{AB}+48.6}{2.5}} * \frac{c}{\lambda^2} \tag{27}$$

where:
m_{AB}: is the AB magnitude (given by the user)
c: is the speed of light expressed in $\text{Å}/s$
λ: is the wavelength expressed in Å

18.4 Emission Line

This method is used when the user selects the **Emission Line** option for the **spectral distribution** of the sources flux. In this case the flux is represented by a single emission line and is calculated as:

$$F_{(\lambda)} = F_t G_{(\lambda+\lambda_0;\sigma)} \tag{28}$$

where:
F_t: is the total flux of the emission line expressed in $erg/cm^2/s$ (given by the user)
G: is the Gaussian function

λ: is the wavelength expressed in Å
$\lambda_0 = \lambda_{em}(1+z)$:
λ_{em}: is the wavelength of the emission line at rest frame expressed in Å(given by the user)
z: is the redshift (given by the user)
$\sigma = FWHM_{em}(1+z)/2.3548$
$FWHM_{em}$: is the FWHM of the emission line at rest frame expressed in Å(given by the user)

18.5 Black Body

This method is used when the user selects the **Black Body** option for the **spectral distribution** of the sources flux. In this case the flux is represented by a redshifted, normalized black body. The black body radiation per unit projected area is calculated by using the equation:

$$I_{(\lambda)} = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k T_z}} - 1} \tag{29}$$

where:
$I_{(\lambda)}$: is the black body radiation
λ: is the wavelength expressed in Å
h: is the Planck constant expressed in $erg * s$
c: is the speed of light expressed in $Å/s$
k: is the Boltzmann constant expressed in erg/K
$T_z = \frac{T}{1+z}$: is the black body redshifted temperature
T: is the temperature of the black body at restframe expressed in K (given by the user)
z: is the redshift

This radiation is then normalized by using the flux calculated from the AB magnitude:

$$F = 10^{-\frac{m_{AB}+48.6}{2.5}} * \frac{c}{\lambda_x^2} \tag{30}$$

where:
F: is the flux on which the radiation will be normalized
m_{AB}: is the AB magnitude (given by the user)
c: is the speed of light expressed in $Å/s$
λ_x: is the reference wavelength of the AB magnitude selected by the user, expressed in Å

19 Normalization Factor

The normalization factor calculator is responsible for calculating the total normalization factor $C_{t(\lambda)}$ and the central pixel normalization factor $C_{cp(\lambda)}$. The normalization factor has different meaning for point sources and for extended sources.

For point sources, the magnitude AB given by the user corresponds to the total flux of the target. In this case the total normalization factor is a number in the range [0,1], which shows what percentage of the targets flux is passing through an existing slit or fiber (in any other case

it is one) and the central pixel normalization factor is a number in the range [0,1], which shows the percentage of the flux falling on the central pixel.

For extended sources, the magnitude AB given by the user corresponds to the maximum flux of the profile. In this case the total normalization factor is a number with which the maximum flux needs to be multiplied to calculate the total flux, taking in consideration any loses because of a slit or fiber and the central pixel normalization factor is a number with which the maximum flux needs to be multiplied to calculate the flux falling on the central pixel. In this case the normalization factors can be bigger than one.

19.1 Point Source Imaging

This method is used when the user gives the following setup:

- The instrument is an **imager**
- The **Source** is set to **Point Source**

For this method the total normalization factor is always one:

$$C_{t(\lambda)} = 1 \tag{31}$$

The central pixel normalization factor is calculated with the integral:

$$C_{cp(\lambda)} = \int_{-p/2}^{p/2} \int_{-p/2}^{p/2} PSF_{(x,y;\lambda)} dxdy \tag{32}$$

where:
p: is the pixel scale of the instrument expressed in $arcsec/pixel$
$PSF_{(x,y;\lambda)}$: is the PSF function (see section 27)

19.2 Point Source Slit

This method is used when the user gives the following setup:

- The instrument is a **slit spectrograph**
- The **Source** is set to **Point Source**

In this case the normalization factor takes in consideration the losses because of the slit. The total normalization factor is calculated with the integral:

$$C_{t(\lambda)} = \int_{y_1}^{y_2} \int_{x_1}^{x_2} PSF_{(x,y;\lambda)} dxdy \tag{33}$$

where:
$C_{t(\lambda)}$: is the total normalization factor
$PSF_{(x,y;\lambda)}$: is the PSF function (see section 27)

$x_2 = min(\frac{PSF_{size}}{2}, \frac{w}{2})$
$x_1 = -x_2$
$y_2 = min(\frac{PSF_{size}}{2}, \frac{l}{2})$
$y_1 = -y_2$
PSF_{size}: is the size of the PSF (see section 26)
w: is the width of the slit expressed in *arcsec* (given by the user)
l: is the length of the slit expressed in *arcsec* (given by the user)

The central pixel normalization factor is calculated with the integral:

$$C_{cp(\lambda)} = \int_{y_1}^{y_2} \int_{x_1}^{x_2} PSF_{(x,y;\lambda)} dxdy \tag{34}$$

where:
$C_{cp(\lambda)}$: is the central pixel normalization factor
$PSF_{(x,y;\lambda)}$: is the PSF function (see section 27)
$x_2 = min(\frac{PSF_{size}}{2}, \frac{w}{2})$
$x_1 = -x_2$
$y_2 = min(\frac{PSF_{size}}{2}, \frac{l}{2}, \frac{p}{2})$
$y_1 = -y_2$
PSF_{size}: is the size of the PSF (see section 26)
w: is the width of the slit expressed in *arcsec* (given by the user)
l: is the length of the slit expressed in *arcsec* (given by the user)
p: is the pixel scale expressed in *arcsec/pixel* (given by the user)

19.3 Point Source Fiber

This method is used when the user gives the following setup:

- The instrument is a **fiber spectrograph**
- The **Source** is set to **Point Source**

In this case the normalization factor takes in consideration the losses because of the fiber. The total normalization factor is calculated with the integral:

$$C_{t(\lambda)} = \iint_D PSF_{(x,y;\lambda)} dxdy \tag{35}$$

where:
$C_{t(\lambda)}$: is the total normalization factor
$PSF_{(x,y;\lambda)}$: is the PSF function (see section 27)
D: is a circle with diameter ϕ and center the center of the PSF
ϕ: is the fiber diameter expressed in *arcsec* (given by the user)

The central pixel normalization factor is calculated with the equation:

$$C_{cp(\lambda)} = \begin{cases} C_{t(\lambda)} & \text{, for } \phi \leq p \\ C_{t(\lambda)} * \frac{OverlappingArea}{FiberArea} & \text{, for } \phi > p \end{cases} \tag{36}$$

where:
$C_{cp(\lambda)}$: is the central pixel normalization factor
$C_{t(\lambda)}$: is the total normalization factor
ϕ: is the fiber diameter expressed in *arcsec* (given by the user)
p: is the pixel scale expressed in *arcsec/pixel* (given by the user)
$FiberArea$: is the area of the fiber
$OverlappingArea$: is the area of the fiber overlapping with the central pixel row

NOTE: Currently only circularly symmetric PSF functions are supported.

19.4 Extended Source Imaging

This method is used when the user gives the following setup:

- The instrument is an **imager**
- The **Source** is set to **Extended Source**

In this case the total normalization factor is calculated with the integral:

$$C_{t(\lambda)} = \iint_D I_{c(x,y;\lambda)} dxdy = \int_0^{2\pi} \int_0^{r_{max}} I_{c(r;\lambda)} r dr d\theta = 2\pi * \int_0^{r_{max}} I_{c(r;\lambda)} r dr \tag{37}$$

where:
$C_{t(\lambda)}$: is the total normalization factor
$I_{c(x,y;\lambda)}$: is the surface brightness profile convolved with the PSF (see section 20)
D: is a circle with radius r_{max} and center the center of the PSF
$r_{max} = \sqrt{r_{obj}^2 + \left(\frac{PSF_{size}}{2}\right)^2}$
r_{obj}: is the radius of the extended object expressed in *arcsec* (given by the user)
PSF_{size}: is the size of the PSF (see section 26)

The central pixel normalization factor is calculated with the equation:

$$C_{cp(\lambda)} = \begin{cases} C_{t(\lambda)} & \text{, for } r_{max} \leq p/2 \\ \int_{-p/2}^{p/2} \int_{-p/2}^{p/2} I_{c(x,y;\lambda)} dxdy & \text{, for } r_{max} > p/2 \end{cases} \tag{38}$$

where:
$C_{cp(\lambda)}$: is the central pixel normalization factor
$C_{t(\lambda)}$: is the total normalization factor
$I_{c(x,y;\lambda)}$: is the surface brightness profile convolved with the PSF (see section 20)
$r_{max} = \sqrt{r_{obj}^2 + \left(\frac{PSF_{size}}{2}\right)^2}$
r_{obj}: is the radius of the extended object expressed in *arcsec* (given by the user)
PSF_{size}: is the size of the PSF (see section 26)
p: is the pixel scale expressed in *arcsec/pixel* (given by the user)

19.5 Extended Source Slit

This method is used when the user gives the following setup:

- The instrument is a **slit spectrograph**
- The **Source** is set to **Extended Source**

In this case the normalization factor takes in consideration the losses because of the slit. The total normalization factor is calculated with the integral:

$$C_{t(\lambda)} = \int_{y_1}^{y_2} \int_{x_1}^{x_2} I_{c(x,y;\lambda)} dxdy \tag{39}$$

where:
$C_{t(\lambda)}$: is the total normalization factor
$I_{c(x,y;\lambda)}$: is the surface brightness profile convolved with the PSF (see section 20)
$x_2 = min(r_{max}, \frac{w}{2})$
$x_1 = -x_2$
$y_2 = min(r_{max}, \frac{l}{2})$
$y_1 = -y_2$
$r_{max} = \sqrt{r_{obj}^2 + \left(\frac{PSF_{size}}{2}\right)^2}$
r_{obj}: is the radius of the extended object expressed in *arcsec* (given by the user)
PSF_{size}: is the size of the PSF (see section 26)
w: is the width of the slit expressed in *arcsec* (given by the user)
l: is the length of the slit expressed in *arcsec* (given by the user)

The central pixel normalization factor is calculated with the integral:

$$C_{cp(\lambda)} = \int_{y_1}^{y_2} \int_{x_1}^{x_2} I_{c(x,y;\lambda)} dxdy \tag{40}$$

where:
$C_{cp(\lambda)}$: is the central pixel normalization factor
$I_{c(x,y;\lambda)}$: is the surface brightness profile convolved with the PSF (see section 20)
$x_2 = min(r_{max}, \frac{w}{2})$
$x_1 = -x_2$
$y_2 = min(r_{max}, \frac{l}{2}, \frac{p}{2})$
$y_1 = -y_2$
$r_{max} = \sqrt{r_{obj}^2 + \left(\frac{PSF_{size}}{2}\right)^2}$
r_{obj}: is the radius of the extended object expressed in *arcsec* (given by the user)
PSF_{size}: is the size of the PSF (see section 26)
w: is the width of the slit expressed in *arcsec* (given by the user)
l: is the length of the slit expressed in *arcsec* (given by the user)
p: is the pixel scale expressed in *arcsec/pixel* (given by the user)

19.6 Extended Source Fiber

This method is used when the user gives the following setup:

- The instrument is a **fiber spectrograph**
- The **Source** is set to **Extended Source**

In this case the normalization factor takes in consideration the losses because of the fiber. The total normalization factor is calculated with the integral:

$$C_{t(\lambda)} = \iint_D I_{c(x,y;\lambda)} dxdy \tag{41}$$

where:
$C_{t(\lambda)}$: is the total normalization factor
$I_{c(x,y;\lambda)}$: is the surface brightness profile convolved with the PSF (see section 20)
D: is a circle with diameter ϕ and center the center of the PSF
ϕ: is the fiber diameter expressed in *arcsec* (given by the user)

The central pixel normalization factor is calculated with the equation:

$$C_{cp(\lambda)} = \begin{cases} C_{t(\lambda)} & \text{, for } \phi \leq p \\ C_{t(\lambda)} * \frac{OverlappingArea}{FiberArea} & \text{, for } \phi > p \end{cases} \tag{42}$$

where:
$C_{cp(\lambda)}$: is the central pixel normalization factor
$C_{t(\lambda)}$: is the total normalization factor
ϕ: is the fiber diameter expressed in *arcsec* (given by the user)
p: is the pixel scale expressed in *arcsec/pixel* (given by the user)
$FiberArea$: is the area of the fiber
$OverlappingArea$: is the area of the fiber overlapping with the central pixel row

20 Convolution of Surface Brightness Profile with PSF

This calculator is responsible for providing functions representing the convolution of the surface brightness profile (see section 21) with the PSF (see section 27).

$$I_{c(r;\lambda)} = I_{(r)} * PSF_{(r;\lambda)} \tag{43}$$

where:
$I_{c(r;\lambda)}$: is the result of the convolution
$I_{(r)}$: is the surface brightness profile (see section 21)
$PSF_{(r;\lambda)}$: is the PSF (see section 27)

Because the PSF depends on the wavelength λ, this calculator gets as input a value representing the wavelength (expressed in Å). Its output is a bivariate function which (for the specific wavelength) can be used for calculating the convolution result for given distances (x,y) (expressed in *arcsec*) from the point of its maximum value.

20.1 Circularly Symmetric

This method is used when both the PSF and the surface brightness profile are circularly symmetric functions. It uses this fact for accelerating the convolution computation. Note that currently there is no support for non circularly symmetric PSF or surface brightness profiles.

Note: This method has been replaced with the faster FFT method. It still exists for testing reasons.

20.2 FFT Convolution

This method uses the Fast Fourier Transformation Convolution algorithm. This algorithm performs the following steps:

1. Both PSF and surface brightness profile functions are sampled in 2D
2. The sampled data are transformed using the FFT
3. The transformed data are multiplied element by element
4. The result of the multiplication is transformed using the inverse FFT

21 Surface Brightness Profile

The surface brightness profile calculator is responsible for providing functions representing the surface brightness profiles $I_{(r)}$.

21.1 Uniform

For this method the uniform spatial profile is used. This profile follows the distribution:

$$I_{(r)} = \begin{cases} 1 & , r \leq r_{obj} \\ 0 & , r > r_{obj} \end{cases} \tag{44}$$

21.2 De Vaucouleurs

For this method the De Vaucouleurs spatial profile is used. This profile follows the distribution:

$$I_{(r)} = e^{-7.67*\left(\frac{r}{r_{obj}}\right)^{1/4}} \tag{45}$$

21.3 Exponential

For this method the exponential spatial profile is used. This profile follows the distribution:

$$I_{(r)} = e^{-1.68*\left(\frac{r}{r_{obj}}\right)} \tag{46}$$

22 Number of Pixels

The number of pixels calculator is responsible for calculating the total number of pixels of the detector, which are affected by the target. This calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value, representing the total number of pixels affected by the target.

22.1 Photometry

This method is used in the case of imager simulation. In this case there is no obstacles on the light path (like slit, fiber, etc) and there is no diffraction medium (like grism, prism, etc), so the source pixel coverage image (see section 23) is a representation of the source on the sensor. What this method does is to just count the non zero pixels of this image.

22.2 Spectroscopy

This method is used in the case of spectrograph simulation. In this case the number of pixels can be calculated with the equation:

$$N_{pix} = \begin{cases} \lceil SpatialBinning \rceil & \text{, for per spectral pixel simulation} \\ \lceil SpatialBinning \rceil * \lceil SpectralBinning \rceil & \text{, for per spectral resolution element simulation} \end{cases} \quad (47)$$

where:
$SpatialBinning$: is the number of pixels affected in the spatial direction (see section 25)
$SpectralBinning$: is the number of pixels affected in the spectral direction (see section 24)
$\lceil\ \rceil$: is the ceiling function (rounding up)

23 Source Pixel Coverage

The source pixel coverage calculator is producing a map of the detector pixels which are affected by the source light, without taking into consideration any slit or fiber losses, or any grism or prism diffraction. The output of the calculator is a two dimensional image for each wavelength λ, which represents with 1 the pixels which are affected by the source light and with 0 the ones that are not.

Note: This calculator is currently used only for imaging simulations. Further investigation is necessary before it is used for spectrograph simulations.

23.1 Symmetric

This method is currently used for all the types of simulation. Firstly, the radius of the source (in *pixel*) is calculated, by using the equation:

$$R_{(\lambda)} = \frac{SpatialBinning_{(\lambda)}}{2} \tag{48}$$

where:
λ: is the wavelength expressed in angstrom (Å)
$SpatialBinning_{(\lambda)}$: is the spatial binning (see section 25)

By using this radius the side length of the result image (which is a square) is calculated (in *pixels*), by using the ceiling function:

$$Side_{(\lambda)} = \lceil 2 * R_{(\lambda)} \rceil \tag{49}$$

where:
R: is the radius calculated by equation 48 (in *arcsec*)
$\lceil\ \rceil$: is the ceiling function (rounding up)

Finally, the result image is constructed with dimensions $Side * Side$. The value of each pixel is set by checking the distance between the center of the image and the pixel corner closer to it. If this distance is bigger than the calculated radius R (equation 48) the pixel gets the value 0. If it is smaller (so there is some light from the source falling on the pixel), it gets the value 1.

24 Spectral Binning

The spectral binning calculator is responsible for calculating the number of pixels of the detector affected by light coming at a specific wavelength, in the spectral direction. The spectral binning has meaning only for spectroscopy and it should not be used for any other case.

24.1 Fiber

This method is used when the instrument is a **fiber spectrograph**. For this method, the spectral binning is calculated based on the diameter of the fiber, by using the equation:

$$SpectralBinning_{(\lambda)} = \frac{\phi_{fiber}}{p} \tag{50}$$

where:
λ: is the wavelength expressed in angstrom (Å)
ϕ_{fiber}: is the diameter of the fiber expressed in *arcsec* (given by the user)
p: is the pixel scale of the instrument expressed in *arcsec/pixel* (given by the user)

24.2 Slit

This method is used when the instrument is a **slit spectrograph**. For this method, the spectral binning is calculated based on the width of the slit, by using the equation:

$$SpectralBinning = \frac{w}{p} \tag{51}$$

where:
w: is the width of one slit expressed in *arcsec* (given by the user)
p: is the pixel scale of the instrument expressed in *arcsec/pixel* (given by the user)

24.3 Slitess

This method is used when the instrument is a **fiber spectrograph**. For this method, the spectral binning is considered to be the same with the spatial binning (see section 25).

25 Spatial Binning

The spatial binning calculator is responsible for calculating the number of pixels of the CCD affected by the light coming from the observed source, in the spatial direction. Because the spatial binning depends on the PSF size (see section 26), it also depends on the wavelength λ. This means that this calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value, representing the number of pixels in the spatial direction.

25.1 Fiber

This method is used when the instrument is a **fiber spectrograph** (for both point and extended sources). For this method, the spatial binning is calculated based on the diameter of the fiber, by using the equation:

$$SpatialBinning_{(\lambda)} = \frac{\phi_{fiber}}{p} \tag{52}$$

where:
λ: is the wavelength expressed in angstrom (Å)
ϕ_{fiber}: is the diameter of the fiber expressed in *arcsec* (given by the user)
p: is the pixel scale of the instrument, expressed in *arcsec/pixel* (given by the user)

25.2 Point Source

This method is used when the user gives the following setup:

- The **Source** is set to **Point Source**
- The instrument is **not** a **fiber** or **slit spectrograph**

For this method, the spatial binning is calculated based on the size of the PSF, by using the equation:

$$SpatialBinning_{(\lambda)} = \frac{PSF_{size(\lambda)}}{p} \tag{53}$$

where:
λ: is the wavelength expressed in angstrom (Å)
$PSF_{size(\lambda)}$: is the size of the PSF expressed in *arcsec* (see section 26)
p: is the pixel scale of the instrument, expressed in *arcsec/pixel* (given by the user)

25.3 Extended Source

This method is used when the user gives the following setup:

- The **Source** is set to **Extended Source**
- The instrument is **not** a **fiber** or **slit spectrograph**

For this method the spectral binning is calculated based on the effective radius of the target **and** the size of the PSF. It can be calculated by the following equation:

$$SpatialBinning_{(\lambda)} = \frac{\sqrt{(2 * r_{obj})^2 + PSF^2_{size(\lambda)}}}{p} \tag{54}$$

where:
λ: is the wavelength expressed in angstrom (Å)
r_{obj}: is the effective radius of the target expressed in *arcsec* (given by the user)
$PSF_{size(\lambda)}$: is the size of the PSF expressed in *arcsec* (see section 26)
p: is the pixel scale of the instrument, expressed in *arcsec/pixel* (given by the user)

25.4 Slit Point Source

This method is used when the user gives the following setup:

- The **Source** is set to **Point Source**
- The instrument is a **slit spectrograph**

For this method, the spatial binning is calculated based on the size of the PSF and limited by the slit length, by using the equation:

$$SpatialBinning_{(\lambda)} = \frac{min(l, PSF_{size(\lambda)})}{p} \tag{55}$$

where:
λ: is the wavelength expressed in angstrom (Å)
min: is the minimum function
l: is the length of the slit expressed in *arcsec* (given by the user)
$PSF_{size(\lambda)}$: is the size of the PSF expressed in *arcsec* (see section 26)
p: is the pixel scale of the instrument, expressed in *arcsec/pixel* (given by the user)

25.5 Slit Extended Source

This method is used when the user gives the following setup:

- The **Source** is set to **Extended Source**
- The instrument is a **slit spectrograph**

For this method the spectral binning is calculated based on the effective radius of the target **and** the size of the PSF, and it is limited by the slit length. It can be calculated by the following equation:

$$SpatialBinning_{(\lambda)} = \frac{min\left(l, \sqrt{(2 * r_{obj})^2 + PSF^2_{size(\lambda)}}\right)}{p} \tag{56}$$

where:
λ: is the wavelength expressed in angstrom (Å)
r_{obj}: is the effective radius of the target expressed in *arcsec* (given by the user)
min: is the minimum function
l: is the length of the slit expressed in *arcsec* (given by the user)
$PSF_{size(\lambda)}$: is the size of the PSF expressed in *arcsec* (see section 26)
p: is the pixel scale of the instrument, expressed in *arcsec/pixel* (given by the user)

26 PSF Size

The PSF size calculator is responsible for calculating the size of the point spread function (PSF), expressed in *arcsec*. The PSF depends on the wavelength λ, so this calculator get as input a value representing the wavelength (expressed in Å).

26.1 Symmetric Gaussian

This method is used when the PSF is described as a bivariate Gaussian with equal standard deviations for both axes (see section 27). This is true when the user has set the instrument **PSF type** either to **Auto** or to **Gaussian FWHM profile**. In this case the size of the PSF is considered to be twice as big as the FWHM of the Gaussian:

$$PSF_{size} = 2 * FWHM = 2 * 2.3548 * \sigma \tag{57}$$

where:
PSF_{size}: is the size of the PSF expressed in *arcsec*
$FWHM$: is the full with half maximum of the Gaussian expressed in *arcsec*
σ: is the standard deviation of the Gaussian
2.3548: converts the standard deviation σ to the FWHM

26.2 Symmetric Double Gaussian

This method is used when the PSF is described as a double Gaussian consisting of Gaussians with equal standard deviations for both axes (see section 27). This is true when the user has set the instrument **PSF type** either to **Adaptive Optics** or to **Double Gaussian**. In this case the size of the PSF is considered to be twice as big as the FWHM of the widest Gaussian:

$$PSF_{size} = 2 * FWHM_{max} = 2 * 2.3548 * \sigma_{max} \tag{58}$$

where:
PSF_{size}: is the size of the PSF expressed in *arcsec*
$FWHM_{max}$: is the full with half maximum of the widest Gaussian expressed in *arcsec*
σ_{max}: is the standard deviation of the widest Gaussian
2.3548: converts the standard deviation σ to the FWHM

27 PSF

The PSF calculator is responsible for calculating the point spread function (PSF). The PSF depends on the wavelength λ, so this calculator get as input a value representing the wavelength (expressed in Å). Its output is a bivariate function which (for the specific wavelength) can be used for calculating the PSF for given distances (x,y) (expressed in *arcsec*) from the point of its maximum value.

27.1 Diffraction Limited

This method is used when the user gives the following setup:

- The instrument **PSF type** is set to **Automatic**
- The observation is diffraction limited. This is valid for the following two cases:
  The **Site location** is set to **Space**
  The **Site location** is set to **Ground** AND the option **Seeing Limited** is **not** selected

This method calculates the theoretical PSF of the instrument as a Gaussian function, based on the diameter of the primary mirror and the input wavelength, by calculating the FWHM of the Gaussian as:

$$FWHM_{(\lambda)} = 1.22 * \frac{\lambda * 10^{-8}}{D_1} * 206265 \tag{59}$$

where:
λ: is the wavelength expressed in angstrom (Å)
D_1: is the diameter of the primary mirror expressed in centimeters (cm) (given by the user)
10^{-8}: converts Å to cm
206265: converts *radians* to *arcsec*

27.2 Gaussian Instrument PSF

This method is used when the user gives the following setup:

- The instrument **PSF type** is set to **Gaussian**
- The observation is diffraction limited. This is valid for the following two cases:
  The **Site location** is set to **Space**
  The **Site location** is set to **Ground** AND the option **Seeing Limited** is **not** selected

This method calculates the theoretical PSF of the instrument as a Gaussian function using a FWHM template given by the user. Linear interpolation is used for the values between the nodes of the template. The use of the template has as result the following restrictions:

- The values (x, y) of the profile must be expressed in (Å, *arcsec*)
- The method cannot calculate the PSF for wavelengths outside of the range defined from the profile

27.3 Double Gaussian Instrument PSF

This method is used when the user gives the following setup:

- The instrument **PSF type** is set to **Double Gaussian**
- The observation is diffraction limited. This is valid for the following two cases:
  The **Site location** is set to **Space**
  The **Site location** is set to **Ground** AND the option **Seeing Limited** is **not** selected

This method calculates the theoretical PSF of the instrument as a Double Gaussian function using two FWHM templates (one for each Gaussian) and a contribution factor given by the user. The contribution factor is a number between 0 and 1 which shows the contribution of the first Gaussian. The second Gaussian contribution is $1 - first\_contribution$. Linear interpolation is used for the values between the nodes of the templates. The use of the templates has as result the following resSWtrictions:

- The values (x, y) of the profiles must be expressed in (Å, *arcsec*)
- The method cannot calculate the PSF for wavelengths outside of the range defined from the profiles

27.4 Seeing

This method is used when the user gives the following setup:

- The instrument **PSF type** is set to **Automatic**
- The observation is seeing limited. This is valid when the **Site location** is set to **Ground** AND the option **Seeing Limited** is selected

This method calculates the PSF of the atmosphere as a Gaussian function, based on the Seeing and the input wavelength, by calculating the FWHM of the Gaussian as:

$$FWHM_{(\lambda)} = seeing_{(\lambda)} = seeing_{(Zenith\ in\ V-band)} * AirMass^{0.6} * \left(\frac{\lambda}{5000}\right)^{-0.2} \quad (60)$$

where:
λ: is the wavelength expressed in angstrom (Å)
$seeing_{(ZenithinV-band)}$: is the seeing at zenith, for V-band, expressed in *arcsec* (given by the user)
$AirMass$: is the optical path length through earths atmosphere, relative to that at the zenith (AirMass at zenith is 1). AirMass has no units. (given by the user)

27.5 Instruments PSF + Seeing

This method is used when neither the PSF of the instrument or the PSF of the atmosphere are negligible (for ground telescopes with big instrumental PSF). This is true when the user gives the following setup:

- The instrument **PSF type** is set to **Gaussian** or **Double Gaussian**
- The observation is seeing limited. This is valid when the **Site location** is set to **Ground** AND the option **Seeing Limited** is selected

This method calculates the total PSF as a convolution of the Instrument and Seeing PSFs described earlier.

27.6 Adaptive Optics

This method is used when instrument instrument **PSF type** is set to **Adaptive Optics**.

This method calculates the total PSF as a double Gaussian function which consists of the diffraction limited and seeing PSFs, as described earlier. More precisely, the PSF is calculated by using the equation:

$$PSF_{AO(\lambda)} = SR_{(\lambda)} * PSF_{dl(\lambda)} + (1 - SR_{(\lambda)}) * PSF_{s(\lambda)} \quad (61)$$

where:
$PSF_{AO(\lambda)}$: is the final adaptive optics PSF
$PSF_{dl(\lambda)}$: is the diffraction limited PSF (see section 27.1)
$PSF_{s(\lambda)}$: is the seeing limited PSF (see section 27.4)
$SR_{(\lambda)}$: is the strehl ratio calculated with the equation:

$$SR_{(\lambda)} = SR_{ref}^{\left(\frac{\lambda_{ref}}{\lambda}\right)^2} \quad (62)$$

where:
SR_{ref}: is the reference strehl ratio (given by the user)
λ_{ref}: is the wavelength of the reference strehl ratio expressed in Å(given by the user)
λ: is the wavelength expressed in Angstrom (Å)

28 Wavelength Range

The wavelength range calculator is responsible for calculating the range of the spectral axis for which the simulation is performed. It does not require any input and its result consists of two values, λ_{min} and λ_{max} (expressed in Å), which are the minimum and maximum wavelengths of the simulation respectively.

28.1 Spectroscopy

This method is used when the user runs a spectrograph simulation. In this case the minimum and maximum wavelengths of the calculation (expressed in Å) are given by the user.

28.2 Imaging

This method is used when the user runs an imager simulation. In this case the minimum and maximum wavelengths of the calculation depend on the system efficiency (see section 29) and the filter response (see section 30). More precisely, the minimum and maximum wavelengths of the calculation (expressed in Å) are the bounds for which both the system efficiency calculator and the filter response calculator can calculate non zero values. This has as result the restriction that the non zero ranges of these two calculators must overlap.

29 System Efficiency

The system efficiency calculator is responsible for calculating the total efficiency of the system, $\varepsilon_{(\lambda)}$. The efficiency is a number in the range [0, 1], which shows how efficiently the total system (telescope, optics, CCD QE, etc) transfers the light. Note that the filter is excluded as its efficiency is calculated separately. The calculator does not require any input and its result is a function which can be used for calculating the filter response for given wavelength values (expressed in Å).

29.1 Total Efficiency Profile

This method is used when the user gives a profile for the total system efficiency. In this case the values are calculated using linear interpolation. This has as result the following restrictions:

- The x values of the template must be expressed in Å
- The method cannot calculate the efficiency for wavelengths outside the range defined from the profile

30 Filter Response

The filter response calculator is responsible for calculating a function for the efficiency of the filter in use, $Filter_{(\lambda)}$. This efficiency is a number in the range [0, 1], which shows how efficiently the filter in use transfers the light. The calculator does not require any input and its result is a function which can be used for calculating the filter response for given wavelength values (expressed in Å).

30.1 NoFilter

This method is used when there is no filter used, so the filter response is always one.

30.2 Filter Transmission Profile

This method is used when the user gives a profile for the filter transmission. In this case the values are calculated using linear interpolation. This has as result the following restrictions:

- The x values of the template must be expressed in Å
- The method cannot calculate the efficiency for wavelengths outside the range defined from the profile