Master thesis : Photon sieve based telescope for deployable CubeSat interferometer

Auteur : Mercenier, Adrien

Promoteur(s) : Loicq, Jerome

Faculté : Faculté des Sciences appliquées

Diplôme : Master en ingénieur civil en aérospatiale, à finalité spécialisée en "aerospace engineering"

Année académique : 2017-2018

URI/URL : http://hdl.handle.net/2268.2/4639

Photon sieve based telescope for deployable CubeSat interferometer

ACADEMIC SUPERVISOR:
JÉROME LOICQ

STUDENT:
ADRIEN MERCENIER

Thesis Presented for the Master Degree in
Aerospace Engineering

Academic year 2017-2018

Abstract

This Master thesis is about the study of the *photon sieves* aimed to make a *photon sieve based telescope for deployable CubeSat interferometer*. The *photon sieves* are diffracive lens made of polymide, a light material suitable in space environment.
Such a satellite would have a great interest in space observation, and more specifically, in the research of exoplanets thanks to the *nulling interferometry*. The contextual CubeSat is a 6U CubeSat studied to carry and deploy two telescopes.

To begin this work, a presentation of the concepts involved in the project (*interferometry*, *photon sieve* and CubeSat) is done. Then, in order to study the behavior of the *photon sieves*, a light propagator specific to the *photon sieves* is implemented in the *Matlab* environment. Finally, thanks to the propagator, the analysis and the sizing of the *photon sieves* is performed.

Contents

1 Introduction 1

2 Presentation of the project 2
2.1 Photon sieve 2
2.1.1 What is a photon sieve? 2
2.1.2 Advantages 2
2.2 Interferometry 3
2.2.1 Principles 3
2.2.2 Interferometry in space 5
2.3 CubeSat 8
2.3.1 Specifications 8
2.3.2 Deployable 6U CubeSat 10
2.4 Photon sieve based telescope for deployable CubeSat interferometer 13

3 Photon sieves: state-of-the-art 15
3.1 Theoretical reminders 15
3.1.1 Diffraction 15
3.1.2 Interference 18
3.1.3 Huygens-Fresnel principle 20
3.1.4 Coherence 20
3.2 Fresnel zone plate 25
3.3 Photon sieve behaviors 27
3.4 Photon sieve manufacturability 29
3.5 *FalconSAT-7* 29
3.5.1 Goals 29
3.5.2 Spacecraft 30
3.5.3 Deployment of the membrane 30
3.5.4 *Peregrine* 31

4 Nulling interferometry and optical fiber 34
4.1 Young experiment 34
4.2 Michelson stellar interferometer 37
4.3 Nulling interferometry 38
4.4 Optical fiber 39

5 Optical design 41
5.1 Requirements 41
5.1.1 FOV and resolution 41
5.1.2 Wavelength 42
5.1.3 Brightness gain 42
5.2 Optical tools 43
5.2.1 PSF 43
5.2.2 Encircled energy 44
5.2.3 MTF 44
5.3 Specifications 46

6 Development of a propagator 47
6.1 From Huygens-Fresnel to Fourier optics 48
6.2 Numerical discretization 53
6.2.1 Discretization and Shannon-Nyquist Sampling Theorem 53
6.2.2 DFT and FFT 55
6.2.3 Impact of the DFT 57
6.3 Propagator 60
6.3.1 Running 61
6.3.2 Files and hierarchy 62
6.4 Limitations 65
6.4.1 Hypothesis 65
6.4.2 Computer resources 66
6.5 Verifications 69
6.5.1 Circle 69
6.5.2 Photon sieve 72

7 Analyze and sizing of the photon sieves 75
7.1 Reminders of the specifications 75
7.2 Analysis and sizing in visible 75
7.2.1 Pattern parameters 75
7.2.2 Resolution in observation plan 77
7.2.3 Radiometry purposes 81
7.2.4 Bandwidth 85
7.2.5 Alternative patterns 88
7.2.6 Discussion 92

8 Conclusion and perspectives 94

Appendices 95

A **Matlab files of the propagator** 95
1.1 Photon_Sieve_Propagator.m 95
1.2 Propagation.m 96
1.3 Fraunhofer.m 97
1.4 Fresnel_TF.m 97
1.5 Single_Simulation.m 97
1.6 Optimization_Simulation.m 100
1.7 Display_Results_Optimization_C.m 108
1.8 Display_Results_Optimization_PS.m 116
1.9 Display_Results_Single_C.m 138
1.10 Display_Results_Single_PS.m 140
1.11 Photon_Sieve_Generator.m 142
1.12 Circle_Generator.m 143
1.13 Holes_Generator.m 144
1.14 Irradiance_Power.m 144
1.15 Encircle_Power.m 144
1.16 Time_Estimation.m 145

1.17 Diameter_Zones_Max.m 147

B **Multilayer holes** 148

2.1 Photon_Sieve_Generator_ML.m 148

List of Figures

2.1 Shapes of a *Fresnel zone plate* and of a *photon sieve*.[1] 2
2.2 *Photon sieve* of *FalconSAT-7*. [2] 3
2.3 Breakdown of the ELT's structure. [3] 4
2.4 *VLTI*: Effective diameter formed by the Auxiliary Telescopes.[4] 5
2.5 Atmospheric transparency windows.[5] 6
2.6 SPIRIT telescope.[6] 7
2.7 Mission Darwin constellation.[7] 8
2.8 Conception of *Oufti-1* at the *CSL*. [8] 9
2.9 CubeSat dispensers. [9] 9
2.10 Different configurations of Multiple Aperture Optical Telescope. [9] 10
2.11 6U configuration. [9] 11
2.12 Stowed configuration of the deployable 6U CubeSats. [9] 12
2.13 Deployed configuration. [9] 12
2.14 Illustration of the photon sieve based telescope for deployable CubeSat interferometer. 14
3.1 Diffraction effects on the water. [10] 15
3.2 Double slits experiment.[11] 16
3.3 Airy disk pattern.[11] 16
3.4 Illumination profile of the Airy disk. [11] 17
3.5 Constructive and destructive interferences.[12] 18
3.6 *Huygens-Fresnel* principle. [11] 20
3.7 Random jump of the phase. [13] 22
3.8 Jumps of two independents sources. [13] 23
3.9 Coherence in the two slits experiment. [13] 23
3.10 Beams crossing a *Fresnel zone plate*. [14] 25
3.11 Interference in the *Fresnel zone plate*. [14] 26
3.12 From a *Fresnel zone plate* to a *photon sieve*. [15] 27
3.13 Variation of the observation plan along the axis.[16] 27
3.14 Chromatic dispersion of the *photon sieve*. [17] 28
3.15 Dispersion corrector. [16] 28
3.16 Photolithography process to produce a *photon sieve*. [16] 29
3.17 CAD of *FalconSAT-7*. [18] 30
3.18 *FalconSat-7*: Deployment sequence of the membrane *photon sieve*. [16] 31
3.19 *Peregrine* in stowed configuration. [19] 32
3.20 Secondary optics of *FalconSat-7*. [18] 32
3.21 Secondary optics of *FalconSat-7* (exploded view). [18] 33
4.1 Young experiment with a quasi-monochromatic source. [13] 34
4.2 Young experiment with an extended source. [11] 35
4.3 Interference figures obtained by increasing the size of the extended source.[11] 36
4.4 Michelson stellar interferometer.[11] 37
4.5 Working of a stellar nulling interferometer.[20] 39
4.6 Optical fiber representation.[21] 40
4.7 Example of design for a stellar fiber interferometer.[22] 40
5.1 FOV for a simple lens.[23] 41

5.2 Spectrum of light sources. [24] 42
5.3 Brightness gain. [25] 42
5.4 Angular resolution with Rayleigh criterion. [26] 43
5.5 Encircled energy of the *Airy disk*. [27] 44
5.6 Modulation of intensity. [27] 45
5.7 Modulation transfer function with respect to the spatial frequency. [28] 45
6.1 Interface of Code V. [29] 47
6.2 Representation of the light as beams and wavefronts. [30] 48
6.3 Geometrical description of the propagation of the light between two parallel planes. [31] 50
6.4 Geometrical description of the propagation in the context of the *photon sieves*. . 51
6.5 Discretization in the *photon sieve* plane. 55
6.6 Squared signal. 58
6.7 Comparison between the results of the DFT and the FT. 58
6.8 Periodic extension of the squared signal. 58
6.9 Periodic extension of the response (M=20). 59
6.10 Periodic extension of the response (M=40). 60
6.11 Periodic effect obtained during a simulation. 60
6.12 Display single results. 62
6.13 Choice of the parameter to analyze. 62
6.14 Display optimization results. 62
6.15 Computation of the encircled energy thanks to the *irradiance* profile. 64
6.16 Hierarchy of the files. 65
6.17 Pixelization of the smallest holes with respect to **Speed**. 66
6.18 **Memory** command. 67
6.19 Maximum diameter possible. 68
6.20 Time estimation 69
6.21 Pattern of *irradiance* for a circular aperture. 70
6.22 Profile of *irradiance* for a circular aperture. 71
6.23 Theoretical encircled energy of the *Airy disk*. [32] 72
6.24 Encircled energy obtained with the propagator. 72
6.25 Generated photon sieves with different densities. 73
6.26 Radius of the first peak of *irradiance* when the distance of the observation plane varies. 74
7.1 Pattern parameters for a *photon sieve* with a focal length of 1m and λ=550nm. . 76
7.2 PSF obtained with a *photon sieve*. 78
7.3 Radius of the first peak of the PSF with different focal lengths. 80
7.4 Radius of the first peak of the PSF with different wavelengths. 81
7.5 Surface ratio of the *photon sieve*. 82
7.6 Encircled energy of a *photon sieve*. 82
7.7 Encircled energy (**Speed**=2). 83
7.8 Encircled energy (**Speed=3**). 84
7.9 Energy contains in the first peak for a *photon sieve*. 85
7.10 Chromatic dispersion in the observation plane for a *photon sieve*. 86
7.11 Chromatic dispersion with the profile of *intensity* for a *photon sieve*. 87
7.12 Radius of the first peak with respect to λ. 88

7.13 Multilayer holes pattern (Layer=2). 88
7.14 Photon sieve with 2 layers. 89
7.15 Photon sieve with 5 layers. 89
7.16 Impact introduces by the variation of the holes diameter ($\frac{d}{w}$) 91
7.17 Diameter of the smallest holes with respect to $\frac{d}{w}$. 92

List of Tables

2.1 Classification of the small satellite. [9] 8
2.2 Dimensions of the tubes. [9] 13
6.1 Maximum numbers of zones possible for a simulated *photon sieve* with respect to the RAM and **Speed** 67
7.1 Variation of the number of layers (density=80%). 90
7.2 Variation of the number of layers (density=90%). 90
7.3 Variation of the number of layers (density=95%). 90
7.4 Results for λ=400nm 92
7.5 Results for λ=550nm 93
7.6 Results for λ=700nm 93

Aknowledgement

I would like to thank Jerôme Loicq for many reasons. At first, to have offered me a subject in which I found a real interest. Then, for giving me the opportunity to perform my Master Thesis at the *Center Spatial of Liège*. I also thank him for his availability, time and many advice. The countless meetings we had were really helpful as well as his careful reading.

Thanks to Juliette Renard, who took from her time to correct my English.

Finally, I would like to thank Mathilde Renard, who supported me all along this work, who corrected my English and who did a really careful reading. I wish I could do the same for her.

1 Introduction

The *Hubble Space Telescope*, operational since 1990, allowed views and knowledge of the universe never obtained before. Indeed, by observing from space, the negative impacts of the atmosphere are no longer present and then, it highly improves the quality of the observations. But at what price?

Its cost was close to 1.5 billion of dollars in 1990. Indexed, it represents approximately 3 billion of dollars nowadays. Concerning its next and close successor, the *James Webb Space Telescope* or simply the *JWST*, the price could even exceed 10 billion of dollars. [29] [30]

But, this is not only about money, there is also the cost of time. The *JWST* began to be imagined after the launch of the *Hubble Space Telescope*, and, almost 30 years later, it is still on Earth. Its launch has already taken a decade of delay.

These examples show how space observation and more generally space can be expensive. For this purpose, an interferometer based on a deployable CubeSat with telescopes using *photon sieve* could be interesting. Indeed, *photon sieves* and *interferomertry* could lead to good optical performances without exceeding too large sizes and so, as they are linked, too large costs. The CubeSat standardization is also intended to reduce the costs and to accelerate the procedures. Moreover, a spatial interferometer would have a particular interest in the research of exoplanets thanks to the *nulling interferometry*. Obviously, the study of an interferometer is a vast oeuvre. This work is mainly focused on the *photon sieve*, its performances and sizing, and then, leads to a continuity of the research.

To begin, in Chapter 2, a description of the three concepts linked in this project is done (*photon sieve*, *interferometry* and CubeSat). Chapter 3 acts as a state of the art of the *photon sieve*. Its working is carefully demonstrated and its integration into the space telescope, *FalconSat-7*, is explained.

Chapter 4 is about the *nulling interferometry* and the *optical fiber* which has an interest in *interferometry*. In Chapter 5, a description of the optical design process is made to define the specifications for the study of the *photon sieves*.

In order to analyze the behavior of the *photon sieve*, a propagator has been implemented in the *Matlab* environment. The numerical discretization of the problem, based on the *Fourier Optics*, is explained in Chapter 6. Finally, in Chapter 7, the *photon sieve* is analyzed and sized by the help of the propagator.

2 Presentation of the project

2.1 Photon sieve

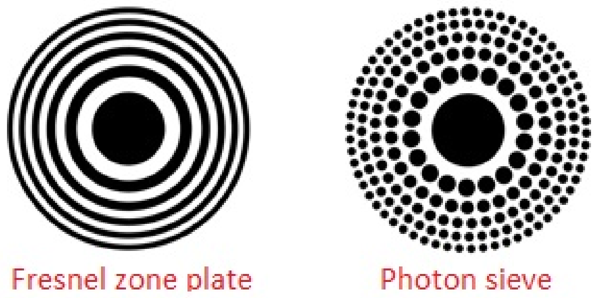


Figure 2.1: Shapes of a *Fresnel zone plate* and of a *photon sieve*.[1]

2.1.1 What is a photon sieve?

Optical elements of a telescopes use special properties of light : the reflection with the mirrors and the refraction with the classical lens. A *photon sieve* is a device that uses the diffraction principles and is then called a *diffractive lens*.

The *photon sieve* is made of a polymeric membrane with thickness approximately equal to 10 microns. The FIG.2.2, represents the tiny membrane of the CubeSat, *FalconSAT-7*. It is shaped with billions of holes, following the pattern of the *Fresnel zone plate* represented in FIG.2.1.

The theoretical content and features of the *photon sieve* will be exposed in the next section. The advantages that it could bring to space observation are listed here after. All of these are leading to a reduction of the weight and then, of the costs. They could also accelerate the manufacturing process.

2.1.2 Advantages

• Light weight and compact

Photon sieve are made of aluminum-coated polyimide films which are strong, pliable, and have a very light weight, which is a huge benefit knowing that one of the main concern of space application is weight. Indeed, nowadays, the price to launch one kilogram in space is close to 10 000$.[31]

For example, the primary mirror used for the *Hubble Space Telescope* (HST) had a specific mass of 180 kg/m^2. Close to 30 years later, the specific mass has been reduced to 25 kg/m^2 in the futur *James Webb Space Telescope* (JWST). However, the use of polyimide membranes could highly reduce this parameter to 0.25 kg/m^2.

An other issue imposed by large monolithic structures, such as primary mirrors, is the packaging that has to respect the limits of the launch vehicle. This problem could be avoided thanks to *photon sieves* which could be folded without losing their properties.[19]

• Surface tolerance

To keep high performances (diffraction-limited performances), the refractive or reflective optical elements have to maintain a quality on the surface close to 1/10 of the wavelength studied. Concerning the *photon sieve*, the constrain is much less significant. Indeed, it requires a flatness of about 10 times the wavelength. For instance, assuming a *photon sieve* dedicated to the visible frequencies, its defaults in flatness should not exceed 5 microns. [19]

• Drawback

One weakness of such a diffractive element, concerns the chromatic dispersion. The pattern of the *photon sieve* depends on the wavelength of interest, and the focal length varies linearly with the wavelength.[19]

This effect can be reduced by using secondary optics as it will be explained further.

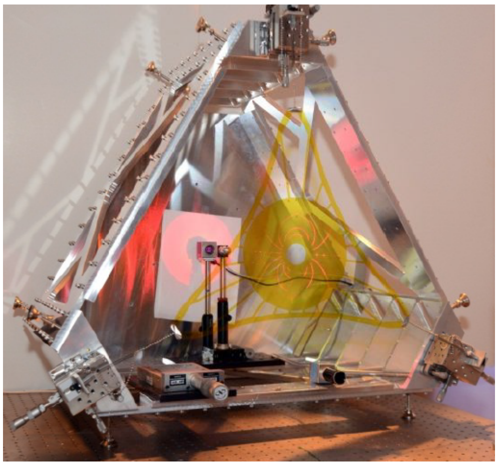


Figure 2.2: *FalconSAT-7*: The picture shows the *photon sieve*, which is the thin circular membrane. [2]

2.2 Interferometry

2.2.1 Principles

When one talks about the size of a telescope, it is about the size of the diameter of its primary mirror. The resolution, or in other words, the ability to catch fine details, is related to that diameter. The larger the diameter of the mirror; the higher the resolution.

However, even with the largest telescopes, some details can not be distinguished. For instance, if the scientists wanted to study the surface of close stars, they would need a telescope with a primary mirror of 1.5 kilometers in diameter. No need to say that it is impossible. [4]

Moreover, even if it was possible to build a mirror as large as that, it would deform due to its own weight. By using an active optics system below the mirror to compensate the flexures, the size of the telescopes can achieve a diameter of 8 to 10 meters. A recent solution used to reach larger diameters is the segmented mirrors. It can increase the size to a few tens of meters. The best example is the *ELT* (Extremely Large Telescope). It is a project of giant telescope under development by the *ESO* (European Southern Observatory) in the Cerro Armazones in Chile. By means of segmented mirrors, its primary mirror will have a diameter of 39 meters and will become the largest telescope in the world (FIG.2.3). [4]

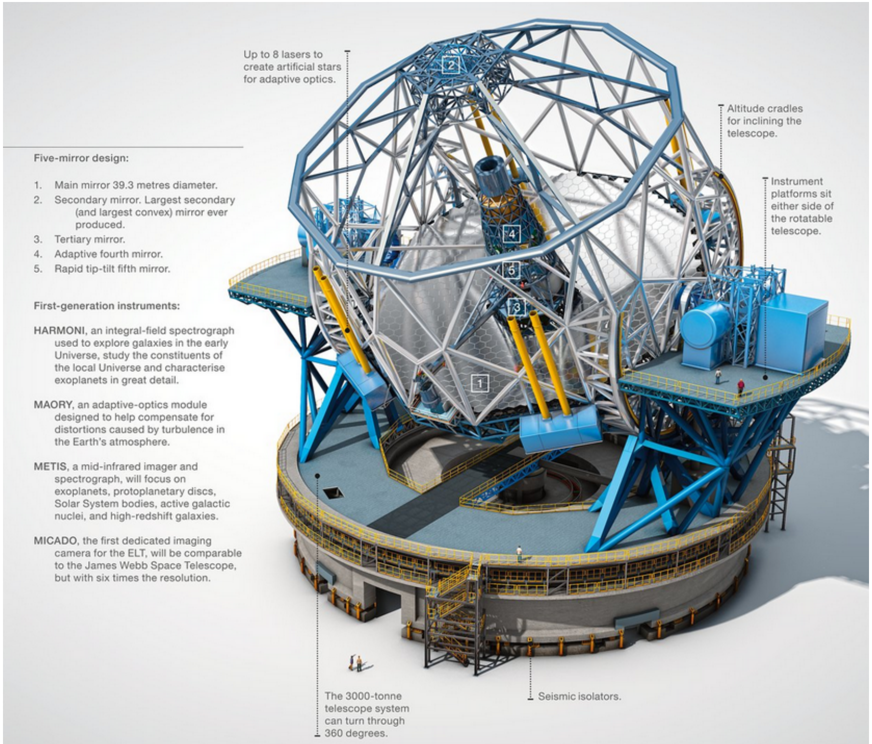


Figure 2.3: Breakdown of the ELT's structure. [3]

This is probably close to the limits of resolution that a single telescope can reach. However, it is possible to reach a better resolution by using more than one telescope with a technology called *inteferometry*.

An interferometer, combines light coming from several distant telescopes. The distance between two telescopes, also called the separation, forms a baseline. The longest baseline of the set determines the effective diameter of the interferometer. The interferometer can thus be seen as a virtual telescope of that diameter. So, it is possible to reach a resolution equivalent to the one of a telescope that has a diameter of hundreds of meters. The more baselines are used, in the length and in the orientation, the more information is acquired about the object studied.

One of the most famous observatory using *interferometry* is the *VLTI* (Very Large Telescope Interferometer), working in infrared and based in the Cerro Paranal in Chile. The FIG.2.4 shows the effective diameter formed by the 1.8-metre Auxiliary Telescopes (ATs), which is equal to 200 meters. In this case the angular resolution is 4 milliarcseconds. [4]

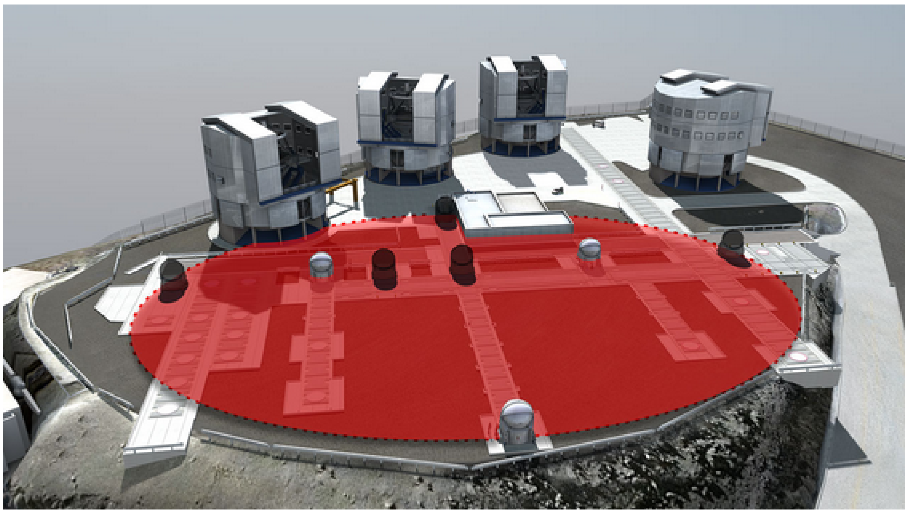


Figure 2.4: *VLTI*: Effective diameter formed by the Auxiliary Telescopes.[4]

The different telescopes are observing the same object, and the light received by each one is combined in a centralized detector. To be successfully combined, the light passing through the telescopes has to arrive at the same time in the detector. As the telescopes are positioned differently, the light does not reach them simultaneously. In order to compensate those differences, extra distances are added in the path of the telescopes which receive the light earlier. They are called *delay lines*. [4]

More careful explanations of the concepts will be done lately in this report.

2.2.2 Interferometry in space

Observation from space has several advantages compared to the observation on Earth. Indeed, the turbulences of the atmosphere decrease the resolution of an optical instrument (these effects can be reduce with adaptive optics). Furthermore, the atmosphere is not transparent to every frequencies as shown in FIG.2.5

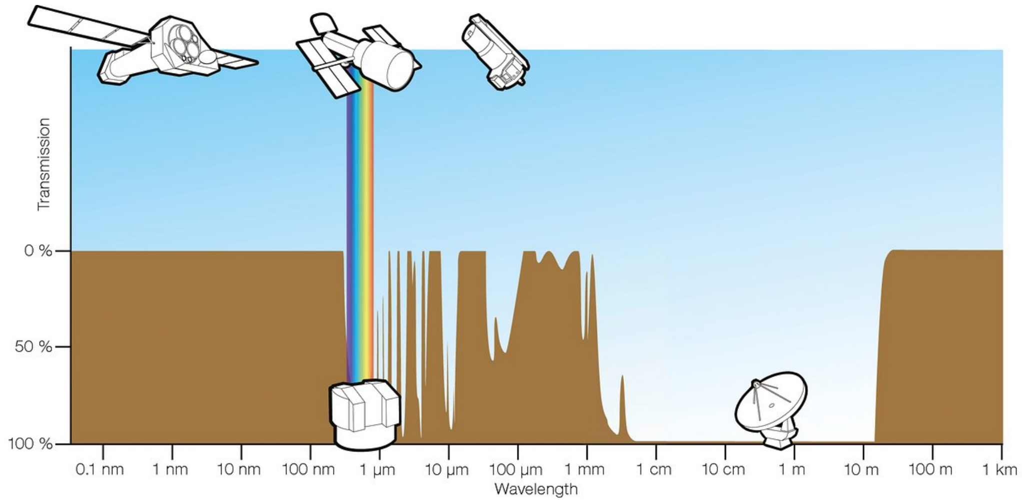


Figure 2.5: Atmospheric transparency windows.[5]

However, advantages never come without drawbacks. As already said, space is expensive and then space telescopes are much more expensive than ground-based telescopes. In this case, *interferometry* could be very promising in space. Indeed, two small telescopes instead of an equivalent bigger one would decrease the weight and size of an optical system.

Nowadays, *interferometry* in space is still at its beginning. A few projects have been made but some did not succeed, because of lack of funding, and others are still in the feasibility phase. Therefore, none of them went into space yet.

Two of them, *SPIRIT* and *Darwin mission*, show different ways to apply *interferometry* in the space environment. They are slightly described here after.

• SPIRIT

SPIRIT (Space Infrared Interferometric Telescope), is a project of the *NASA* whose feasibility study began in 2004. Its mission, is to provide views of planets, stars and galaxy formations by observing in the infrared frequencies (25-400μm).[6]

The *SPIRIT* satellite is composed of two telescopes placed oppositely on deployable rails (FIG.2.6). In order to change the base, they are able to move along the beam like a train on a railway and their information is combined using *interferometry* in the central part of the satellite. The maximum distance reached by the telescopes is 40 meters.

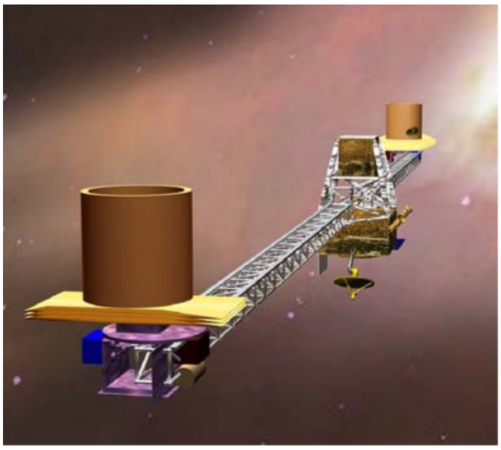


Figure 2.6: SPIRIT telescope.[6]

• Darwin mission

The *Darwin mission* was a project of the *ESA*, which ended in 2007 after the phase-A study (feasability study). The configuration studied was a constellation of five satellites working together (FIG.2.7) designed to search exoplanets (Earth-like planets around other stars) and to analyze their atmospheres seeking for chemical signatures of life. The frequencies studied are in the infrared spectrum (6-18μm) .[7]

Looking for planets orbiting around extrasolar stars is complex. Indeed, for a given optical wavelength, a star outshines a planet by a thousand million to one. The *Mission Darwin* proposed to design the mission in mid-infrared in order to reduce the contrast drop to a million to one (10^{-6}).

The technique used to perform the research is called the *nulling interferometry*. The telescopes are pointed in the direction of the central star. The light which reaches the telescopes has to be delayed before being combined in the hub satellite. The goal is to cancel out the light coming from the star (destructive interferences). Since the exoplanets are not located in the direction pointed by the telescopes, their emitted light is already delayed from one telescope to another. Thanks to this delay and the delay used to cancel out the star, the light coming from the planets interferes constructively, and thus shows the planet.

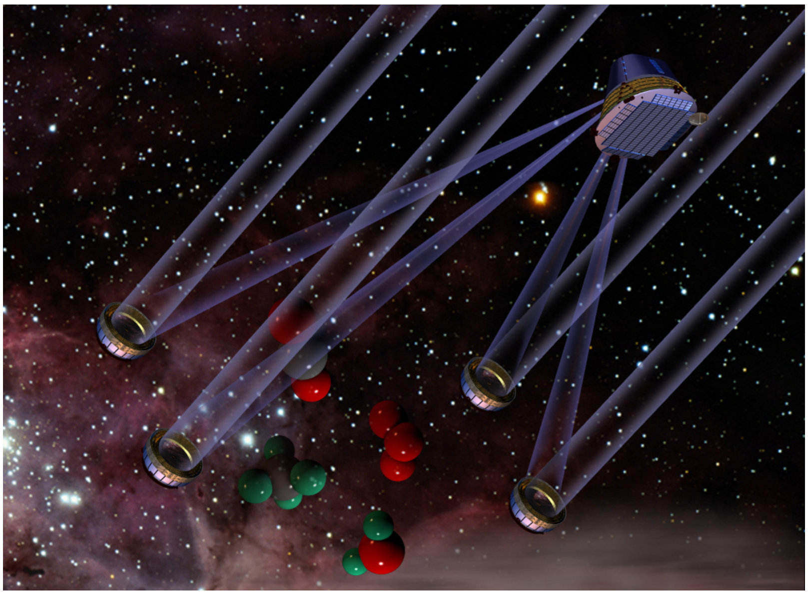


Figure 2.7: Mission Darwin constellation.[7]

2.3 CubeSat

2.3.1 Specifications

Satellites heavier than 500kg are considered as large satellites. Below that mass, the satellites are categorized from minisatellites to femtosatellites following the TABLE 2.1

Category	Mass range [kg]
Minisatellites	100-500
Microsatellites	10-100
Nanosatellites	1-10
Picosatellites	0.1-1
Femtosatellites	<0.1

Table 2.1: Classification of the small satellite. [9]

The CubeSats are standardized nanosatellites. That concept was first developed in the early 2000s by the *Stanford University* (California) in collaboration with the *California State Polytechnic University*. The goal was to broaden the access to space for small payloads and to reduce the development time with the standardized design. [32]

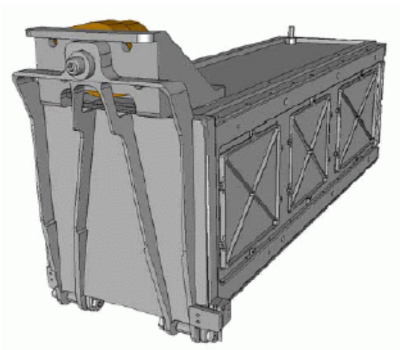
The development and design of a specific CubeSats is generally not made by huge companies as *NASA* or *ESA* but by students within a school or a university. For example, the Belgian nanosatellite *Oufti-1* has been made by students from the University of Liège (FIG.2.8).

Figure 2.8: Conception of *Oufti-1* at the *CSL*. [8]

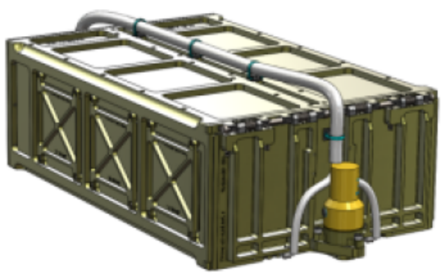
At the beginning, the CubeSat was only referring to a cube of dimensions approximately equal to 10x10x10 cm, also called 1U. Then, it has been extended to the 2U, 3U, 6U and even bigger. The dimensions and weights are the following: [9]

- 1U is a 10x10x11.35cm cube with a mass up to 1.33 kg,
- 2U is a 10x10x22.7cm cube with a mass up to 2.66 kg,
- 3U is a 10x10x34.05cm cube with a mass up to 4 kg,
- 6U is a 10x22.63x36.6cm cube with a mass up to 12 kg.

Usually, CubeSats are secondary payloads of a launch. The dispenser is a standardized system used to stack the CubeSat during the phase of launch. Once the orbital altitude is reached, the dispenser is able to eject the CubeSat. The FIG.2.9 shows two different dispensers.
In FIG.2.9(a) the P-POD (Poly-Picosatellite Orbital Deployer) which can contain three CubeSats 1U or one CubeSat 3U. In FIG.2.9(b), a dispenser made by *Tyvak* society, which is able to transport a CubeSat 6U.[9]



(a) P-POD (3U).



(b) Tyvak (6U).

Figure 2.9: CubeSat dispensers. [9]

2.3.2 Deployable 6U CubeSat

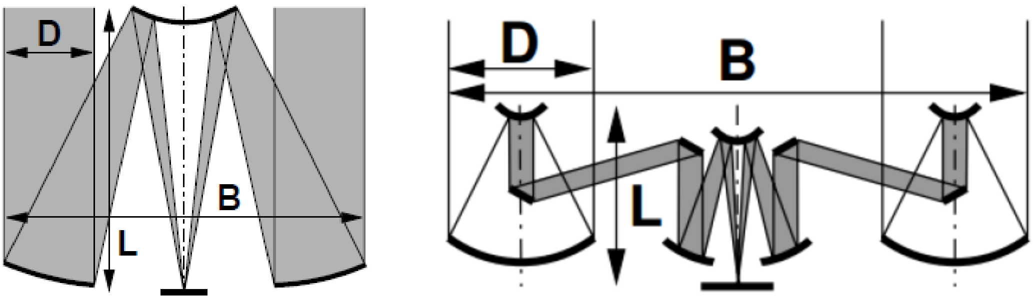
The study of a deployable 6U CubeSat in order to provide solution for *interferometry* has been performed by Sebastien Duijsens during the 2016-2017 academic year. A summary of this work is presented here after. [9]

• System

The idea was to visualize the 6U CubeSat as three 2U CubeSats. In this configuration, the two lateral parts contain a telescope and the central one is relative to the recombination of the signals received by the telescopes. In this case, the system is called a MAOT (Multiple Aperture Optical Telescope). Two configurations exist for a MAOT:

- *Fizeau* design: the aperture segments are portions of a unique primary mirror. [9]
- *Michelson* design: independent telescopes are combined by a dedicated telescope. [9]

They are shown in FIG.2.10.



(a) *Fizeau* configuration. (b) *Michelson* configuration.

Figure 2.10: Different configurations of Multiple Aperture Optical Telescope. [9]

The choice of the *Michelson* configuration, and then, of two independent telescopes has been kept by Sebastien. The FIG.2.11 shows the configuration of the optical system through the 6U CubeSat. In that Figure, the traditional recombination of telescopes thanks to the mirrors is represented. Approximately 1.25 U are dedicated to the optical system in each lateral part of the CubeSat.

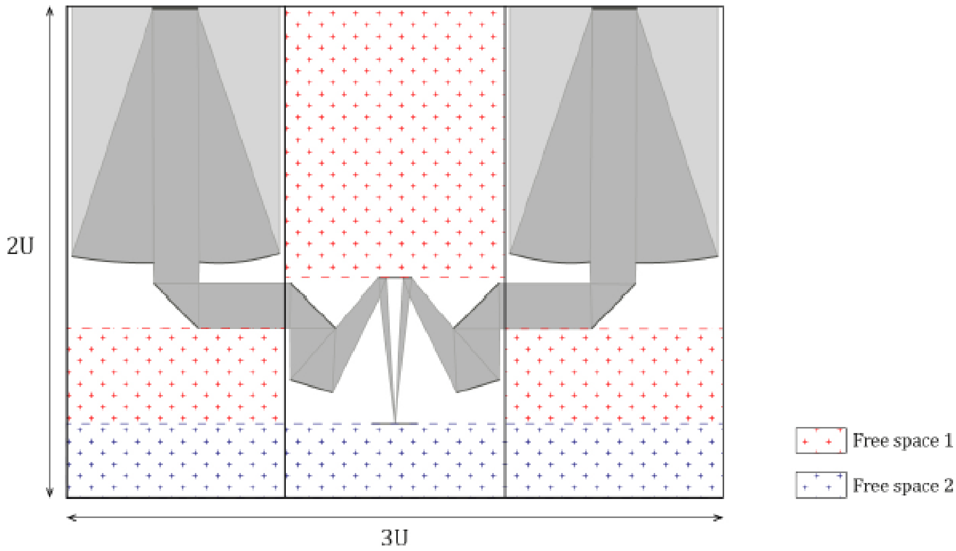


Figure 2.11: 6U configuration. [9]

The proposed concept for the configuration of the 6U CubeSat is illustrated in FIG.2.11. The free spaces shown are filled by other systems used to make the satellite operational. The free space 2 (in blue) is used by the deployment system while the free space 1 (in red) contains the system used for the good behavior of the satellite. Among these:

• Computer: The ISIS on board computer (iOBS) placed in one of the lateral 2U CubeSats.

• Communication: The ISIS TXS S-band transmitter, which is dedicated for nanosatellite in Low Eart Orbit, is placed in the same lateral 2U CubeSat than the computer.

• Attitude: The iADCS400, aimed for small satellite as the 6U. It contains a star tracker for the attitude determination and a reaction wheel for the control of the satellite. It is located in the above part of the central 2U CubeSat.

• Electrical power system: The NanoPower P60, especially designed for nanosatellites, is used to regulate the power and is placed in the second lateral 2U CubeSat.

• Solar panels: To power the satellite; deployable solar panels are added on the sides of the different 2U CubeSats.

• Thermal control: Because of the small remaining place, a passive method has been chosen, such as a coating of the structure.

Finally, a CAD of the stowed configuration is illustrated in FIG.2.12.

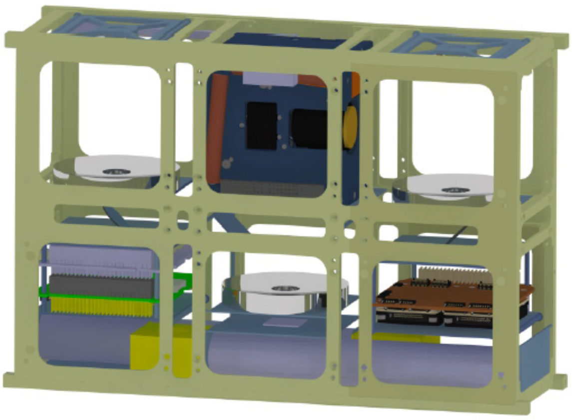


Figure 2.12: Stowed configuration of the deployable 6U CubeSats. [9]

• Proposed deployment mechanism

An enumeration of the different deployment mechanisms has been made by Sebastien. To be suitable with *interferometry*, the deployment mechanism has to respect some criteria. The deployment has to intend an accurate positioning of the telescopes, a good rigidity and a sufficient packaging ratio.

The retained option is the telescoping booms. Indeed, the mechanism is not too complex, is one of the most rigid and is suitable with the available space.

The FIG.2.13 represents the deployed configuration. One can see that guiding rails have been added to increase the rigidity of the satellite.

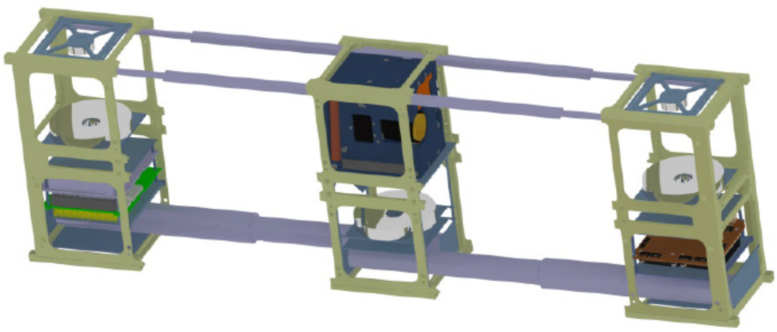


Figure 2.13: Deployed configuration. [9]

Two cases have been studied: a boom with two tubes and a boom with 3 tubes. In the first case, the structure is more rigid, while in the second case, the deployed length is longer and then, leads to an increase of the resolution. The dimensions of the tubes are the following:

	Diameter [mm]	Length [mm]
Tube 1	30	203.2
Tube 2	24	201.2
Tube 3	18	199.2

Table 2.2: Dimensions of the tubes. [9]

• **Modal and thermal analysis**

To perform *inteferometry*, stability is a critical requirement. Indeed, the optical paths, between the telescopes and the recombiner have to be stable with respect to the studied wavelength. Thus, knowing the behavior of the structure, some correctional tools are used.

To this end, a modal and thermal analysis have been studied. For more information; the work of Sebastien Duisjens should be consulted.[9]

2.4 Photon sieve based telescope for deployable CubeSat interferometer

The FIG.2.14 illustrates what a *Photon sieve based telescope for deployable CubeSat interferometer* might look like. Such a satellite has a particular interest in the research of exoplanets thanks to the *nulling interferometry*.

Instead of traditional telescopes made with mirrors, deployable *photon sieves* would be installed in the lateral CubeSats. The illustration shows the *photon sieves* when they are already deployed, the deployable structures are the one used by *FalconSat-7* which will be discussed in the following section. In the stowed configuration both *photon sieves* should be fold in half of the lateral CubeSats space.

This work is focused on the *photon sieves*. They will be studied and sized in the context of this 6U deployable CubeSat interferometer.

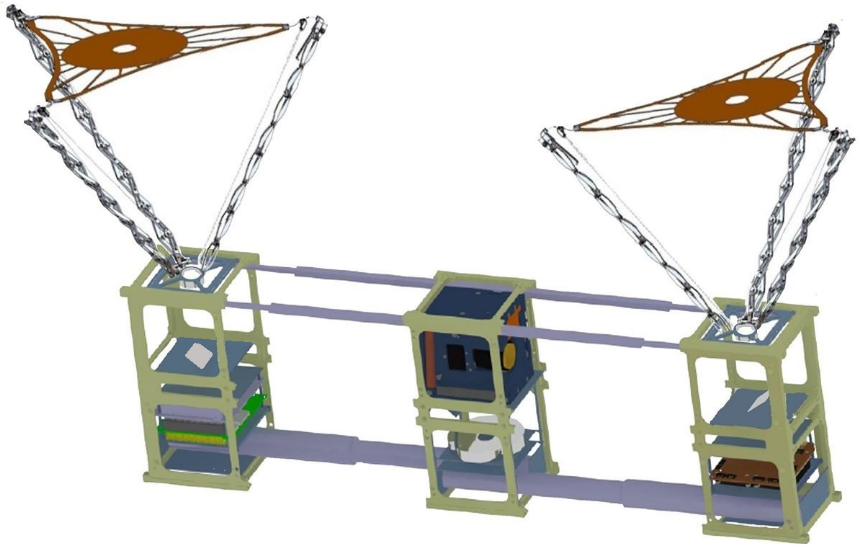


Figure 2.14: Illustration of the 6U CubeSat developed in this work. Both photon sieves are in the deployed configuration.

3 Photon sieves: state-of-the-art

The concept of *photon sieve* was firstly developed in the beginning of the 2000s. In comparison with the traditional telescope, which is a few centuries old, it can be said that the *photon sieve* is a young technology. Therefore, there is still a lot to understand about them and their limits. They could be applied to innovative design.

In this Chapter, the *photon sieve* is described: its mode of operation, the way it is manufactured and the project involving it. The best way to begin is to recall the theoretical principles of diffraction and interference coming from the wave behavior of light. After that, the *Fresnel zone plates* (FZP) will be introduced. [11]

3.1 Theoretical reminders

3.1.1 Diffraction

The diffraction is a characteristic behavior of a wave observed when it meets an obstacle which size is close to its wavelength. This is clearly highlighted in FIG.3.1. This Figure represents a water wave that crosses an opening which size is smaller and smaller. It can be seen that instead of keeping the direction, the waves are spreaded once the obstacle is passed. [11]

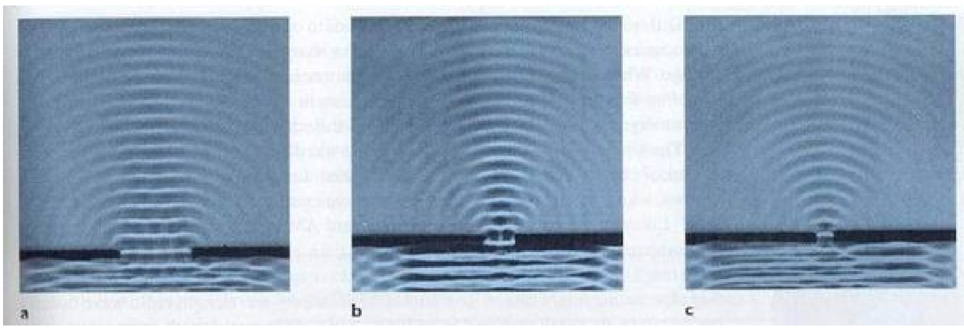


Figure 3.1: Diffraction effects on the water.
[10]

It is important to note that the effect also occurs with the biggest opening. The phenomenon is less marked but is nevertheless present on the edges.

The "double-slit experiment" was performed by *Thomas Young* in the beginning of the 19th century. The experiment consisted in illuminating an opaque plane constituted of two transparent thin slits and then by observing the pattern made by the light on a screen behind the slits. The experiment and the shadow pattern observed are represented in FIG.3.2. This unexpected pattern was explained by the phenomenon of diffraction by *Young* and marked the beginning of the understanding of light as a wave. Nowadays, the wave nature of light is well explained by the theory of the electromagnetic waves.

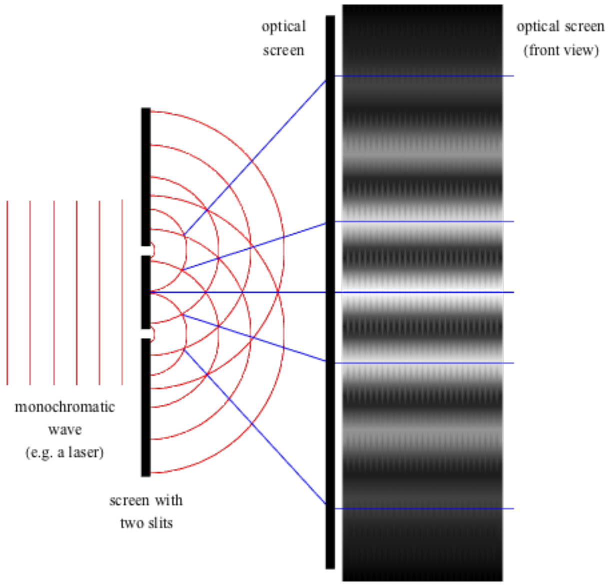


Figure 3.2: Double slits experiment.[11]

An interesting diffracting pattern, a fortiori in optics, is the *Airy disk* obtained by a small circular aperture and a screen located sufficiently far. It is represented in FIG.3.3. One can see that it is composed of concentric circles, alternatively bright and dark. In the bright zones, constructive interferences of the light occur while in the dark zones, destructive interferences occur.
The total amount of luminous energy crossing the circular aperture is received in the observation screen following the principle of conservation of energy. So, in the *Airy disk*, most of the energy is concentrated in the first disk, 84% to be accurate.

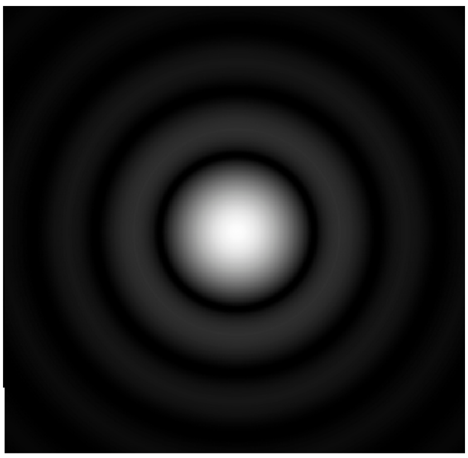


Figure 3.3: Airy disk pattern.[11]

The FIG.3.4, shows the illumination profile of the *Airy disk*. Its analytical expression is rigorously given by :

$$I(\theta) = I(0)[\frac{2J_1(k\frac{D}{2}\sin\theta)}{k\frac{D}{2}\sin\theta}]^2 \tag{3.1}$$

Where *D* is the diameter of the aperture, $k = \frac{2\pi}{\lambda}$ is the wave number and θ is the angle between the line linking a point on the screen to the center of the circular aperture and the optical axis, which is perpendicular to the aperture and crossing its center. J_1 is called a *Bessel function* and verifies the following expression with the value *n=1*

$$J_n(x) = \sum_{p=0}^{\infty} \frac{(-1)^p}{p!(n+p)!} (\frac{x}{2})^{2p+n} \tag{3.2}$$

The first value of θ for which $J_1 = 0$ verifies $\frac{D}{2} k \sin\theta = 3.83$. By noticing that $\sin\theta \approx \frac{r}{z}$, with *r* being the distance between the point in the screen and the center of the screen and z the distance between the aperture and the screen.
The following relation, of considerable interests, gives the radius of the first peak in the *Airy disk*

$$r_1 \approx 1.22 \frac{z\lambda}{D} \tag{3.3}$$

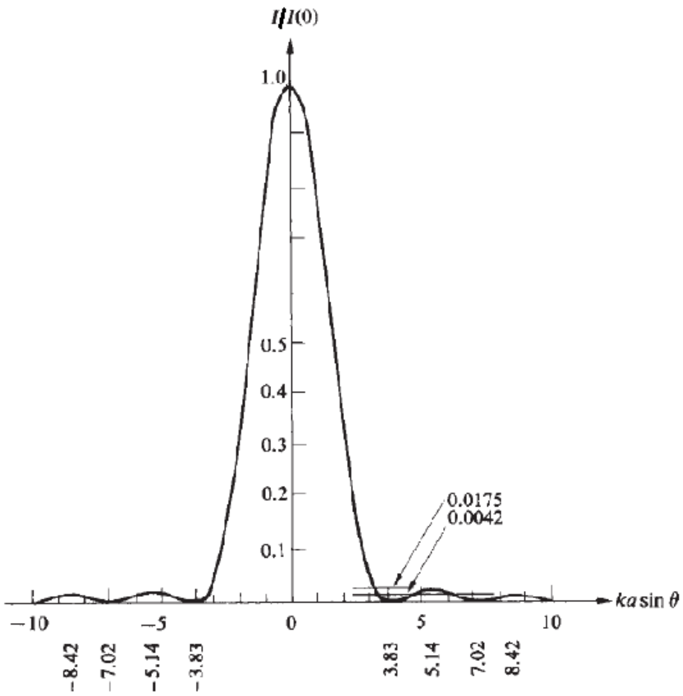


Figure 3.4: Illumination profile of the Airy disk. [11]

Without entering in the details, which will be more discussed in the next sections, the importance of the relation 3.4 in optics is relative to the resolution of an optical system.

An optical system, without any geometric aberration or other else, is said *diffraction-limited*, and is usually, an idealization of the reality. The best resolution of an optical system is obtained when it is diffracted limited. In that case the image of a point is a spot which radius, r_{spot}, is related to the one of the *Airy disk* and is given by the relation

$$r_{spot} = 1.22\frac{f\lambda}{D} \tag{3.4}$$

With *f*, the focal distance of the optical system and *D*, its diameter.

3.1.2 Interference

The interferences between waves are explained by the *superposition principle* which stands that for linear systems, the response caused by two or more sources is the complex sum of the individual responses. When the amplitude of the response is bigger than the amplitude of the individual perturbations, the interferences are said *constructive*. In the opposite situation, when the amplitude of the response is smaller, the interference are said *destructive*. Those effects are well represented in FIG.3.5. The *superposition principle* is applicable for every kind of waves: light wave, acoustic wave, water wave, and so on.

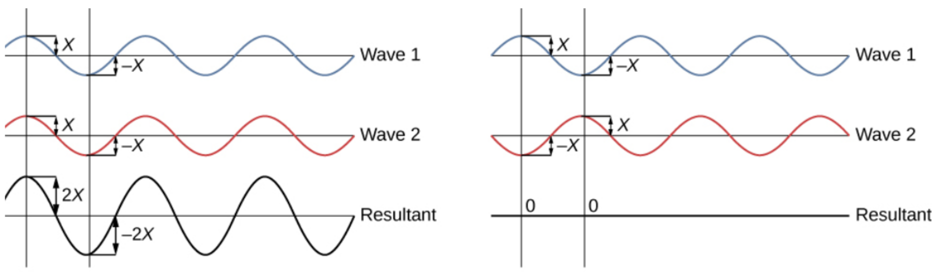


Figure 3.5: Constructive and destructive interferences.[12]

• Electomagnetic waves

The mathematical description of a wave can be made with a sinusoïdal form or a complex form. For a plane monochromatic wave, in the direction ***k***, the sinusoidal form is given by

$$A(\mathbf{r}, t) = A_0 \cos(\mathbf{kr} - \omega t + \phi) \tag{3.5}$$

With ***k*** ($|\boldsymbol{k}| = \frac{2\pi}{\lambda}$), the wave vector; ω, the angular frequency and ϕ the phase. The complex form, which is more convenient for manipulations is

$$U(\mathbf{r}, t) = A_0 e^{j(\mathbf{kr} - \omega t + \phi)} \tag{3.6}$$

And the relation between the two expressions is simply

$$A(\mathbf{r},t) = \mathbb{R}[U(\mathbf{r},t)] \tag{3.7}$$

The frequencies of the electromagnetic waves are generally very high. For example, the frequencies of the visible light are close to 10^{15} Hertz.
Nowadays, the sensors are not able to record information on an interval of time as short as the period of an electromagnetic waves. Instead of that, the sensors record an average parameter, called the *intensity*. This one, which unit is a power per unit of surface ($\frac{W}{m^2}$), is exactly given by

$$\begin{aligned} I(\mathbf{r}) &= \frac{n}{2}\langle c\epsilon_0 U(\mathbf{r})U^*(\mathbf{r})\rangle & (3.8)\\ &= \frac{1}{2\eta}|U(\mathbf{r})|^2 & (3.9)\\ &= \frac{1}{2\eta}A_0^2 & (3.10)\end{aligned}$$

With *c*, the speed of light; *n*, the refractive index; ϵ_0, the vacuum permittivity and η, the characteristic impedance of the medium (377 Ω for vacuum). The mathematical operator $\langle\rangle$ stands for the temporal average. So, assuming a high value of *T* (with respect to the period of the wave) it gives

$$\langle c\epsilon_0 U(\mathbf{r})U^*(\mathbf{r})\rangle = \frac{1}{T}\int_0^T c\epsilon_0 U(\mathbf{r})U^*(\mathbf{r})dt \tag{3.11}$$

It has to be noted that $c\epsilon_0 U(\mathbf{r})U^*(\mathbf{r})$ is actually the energy density of the wave. Generally, by convenience, the prefactor $\frac{1}{2\eta}$ is not taken into account in the expression of the *intensity*.

Assuming now that two monochromatic electromagnetic waves have the same frequencies, $U_1(\mathbf{r},t)$ and $U_2(\mathbf{r},t)$, their interference in ***r*** (the position vector) gives the response

$$\begin{aligned} U(\mathbf{r},t) &= U_1(\mathbf{r},t) + U_2(\mathbf{r},t) & (3.12)\\ &= A_1 e^{j(\phi_1(\mathbf{r})-\omega t)} + A_2 e^{j(\phi_2(\mathbf{r})-\omega t)} & (3.13)\end{aligned}$$

And, by making the mathematical development, the following expression for the *intensity* is found

$$
\begin{aligned}
I(\mathbf{r}) &= |U(\mathbf{r})|^2 && (3.14)\\
&= A_1^2 + A_1^2 + 2A_1A_2cos(\phi_1(\mathbf{r}) - \phi_2(\mathbf{r})) && (3.15)\\
&= I_1 + I_2 + 2\sqrt{I_1I_2}cos(\phi_1(\mathbf{r}) - \phi_2(\mathbf{r})) && (3.16)
\end{aligned}
$$

The Eq.3.16 shows that the resulting *intensity* depends on the phase difference between the waves. In this case, the phase depends on the position, $\boldsymbol{r}$, of the studied point. The phase is an important feature in the *interferometry* purposes.

3.1.3 Huygens-Fresnel principle

The principle of *Huygens-Fresnel* stands that every point of the space crossed by a wave becomes itself a source of a spherical wave having the same wavelength (FIG.3.6). Even if this principle is not rigorously true from the point of view of physics, coupled with the *superposition principle*, it gives good understandings and predictions of the wave propagation of light. Indeed, the mathematical expression given in Eq. 3.1 comes from this principle.[14]

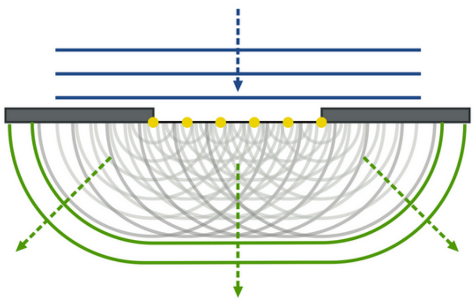


Figure 3.6: *Huygens-Fresnel* principle. [11]

It is important to keep in mind that diffraction and interference are two dependent phenomena and are thus inseparable.

3.1.4 Coherence

Coherence is a delicate topic, because out of intuition, but it is an important concept in interference and therefore in *interferometry*. The principles are presented without entering into the details.[13]

But before that, some identities of the operator $\langle\rangle$ are given ([13])

$$\langle f(x)\rangle = \frac{1}{T}\int_0^T f(x)dt \qquad (3.17)$$

$$\langle cst \rangle = cst \tag{3.18}$$

$$\langle cos(\omega t + \phi) \rangle = 0 \tag{3.19}$$

$$\langle cos^2(\omega t + \phi) \rangle = \frac{1}{2} \tag{3.20}$$

Assuming two monochromatic plane waves, $U_1(\mathbf{r}, t)$ of frequency ω_1 and $U_2(\mathbf{r}, t)$ of frequency ω_2, which are interfering in ***r***. The response is the same than in Eq.3.13 with the particular values for the frequencies. Concerning the *intensity*, its value is given by

$$I(\mathbf{r}) = \langle |A_1 e^{j(\phi_1(\mathbf{r}) - \omega_1 t)} + A_2 e^{j(\phi_2(\mathbf{r}) - \omega_2 t)}|^2 \rangle \tag{3.21}$$

Developing the expression, it yields to

$$I(\mathbf{r}) = I_1 + I_2 + 2\sqrt{I_1 I_2} \langle cos((\omega_1 - \omega_2)t + \phi_1(\mathbf{r}) - \phi_2(\mathbf{r})) \rangle \tag{3.22}$$

Following the identities 3.18, 3.19 and 3.20, two cases arise:

• $\omega_1 = \omega_2$. In this case, the expression of the Eq. 3.16 is still valid

$$I(\mathbf{r}) = I_1 + I_2 + 2\sqrt{I_1 I_2} cos(\phi_1(\mathbf{r}) - \phi_2(\mathbf{r})) \tag{3.23}$$

• $\omega_1 \neq \omega_2$. In this case, the expression becomes

$$I(\mathbf{r}) = I_1 + I_2 \tag{3.24}$$

It shows that in the case where the frequencies are different, the resulting *intensity* is simply the addition of the two *intensities*. Consequently, two waves which have different frequencies will not interfere together.

As it is known, monochromatic waves do not exist; they are an idealization of the reality. Indeed, a light source always emits polychromatic waves ($\Delta\omega \neq 0$). However, the broadband of frequencies can be quite small. For instance, the light emitted by *lasers* is close to be monochromatic. A more realistic model uses the *quasi-monochromatic waves*. Its mathematical description is

$$e^{j(\phi_1(\mathbf{r},t)-\omega_1 t)} \tag{3.25}$$

Compared to the previous model, in this one, the phase is a random function of time. This effect comes from the physical nature of light emissions. For instance, light can be emitted during the transition of an atom from an excited energy state to a lower energy state (spontaneous emission). This transition can occur at random moments and the phase of the emitted light is then a random function of time.

The FIG.3.7 represents this effect. The characteristic time or *coherence time* between two phase jumps is noted τ_c.

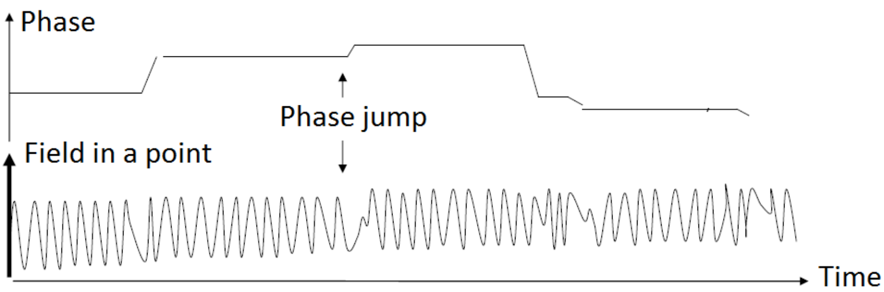


Figure 3.7: Random jump of the phase. [13]

Now, two *quasi-monochromatic waves* of same frequencies which are interfering are considered. By adapting the Eq. 3.22, the *intensity* of the response becomes

$$I(\mathbf{r}) = I_1 + I_2 + 2\sqrt{I_1 I_2}\langle cos(\phi_1(\mathbf{r},t) - \phi_2(\mathbf{r},t))\rangle \tag{3.26}$$

If the sources responsible of these two waves are independent, the phase jumps of the waves are independent too, as seen in FIG.3.8.

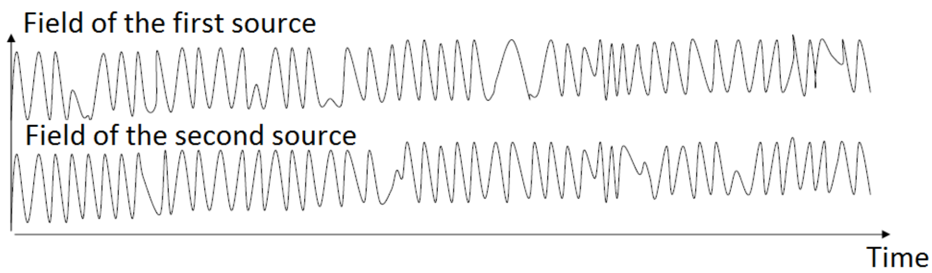


Figure 3.8: Jumps of two independents sources. [13]

In this case, the function $\phi_1(\boldsymbol{r}, t) - \phi_2(\boldsymbol{r}, t)$ becomes a random function of time, sometimes positive, sometimes negative, so

$$\langle cos(\phi_1(\mathbf{r}, t) - \phi_2(\mathbf{r}, t)) \rangle = 0 \tag{3.27}$$

Then

$$I(\mathbf{r}) = I_1 + I_2 \tag{3.28}$$

Two independent *quasi-monochromatic waves* sources cannot create interference. They are said mutually *incoherents*.

Let's go back to *Young*'s experiment, presented in FIG.3.9. The source of light is located in *S*. Following the *Huygens-Fresnel* principle, S_1 and S_2 become themselves sources, but dependent sources this time.

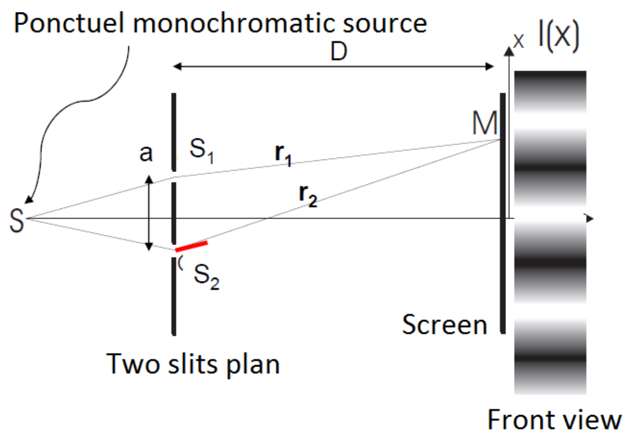


Figure 3.9: Coherence in the two slits experiment. [13]

The path difference of the sources in *M* (FIG.3.9) is

$$\delta_{21} = r_2 - r_1 = c\tau_{21} \tag{3.29}$$

The path difference can be converted in a temporal difference (τ_{21}) knowing that the waves are traveling at the speed of light, c. Using this equation and the fact that the sources are dependent, their phases can be expressed as

$$\phi_1(M, t') = \phi(M, t') \tag{3.30}$$

$$\phi_2(M, t') = k\delta_{21} + \phi(M, t' - \tau_{21}) \tag{3.31}$$

With *k*, the wave number of the emitted light from the source located in *S*.

The Eq. 3.26 is rewritten and becomes

$$I(\mathbf{r}) = I_1 + I_2 + 2\sqrt{I_1 I_2}\langle cos(k\delta_{21} + \phi(M, t' - \tau_{21}) - \phi(M, t'))\rangle \tag{3.32}$$

Again, two cases arise

- $\tau_{21} << \tau_c$

$$I(\mathbf{r}) = I_1 + I_2 + 2\sqrt{I_1 I_2}\langle cos(k\delta_{21})\rangle \tag{3.33}$$

- $\tau_{21} > \tau_c$

$$I(\mathbf{r}) = I_1 + I_2 \tag{3.34}$$

Waves coming from the same source will interfere if $\tau_{21} << \tau_c$. In other words, if the temporal difference, caused by the difference between the optical paths, is much smaller than the *coherence time* (time between two phase jumps). This condition can be expressed with respect to the path difference as

$$\delta_{21} < c\tau_c = L_c \tag{3.35}$$

With L_c the *coherence length*.

Note that those coherence principles can be extended to polychromatic light but will not be done here. However, one thing will be emphasized. For a polychromatic light of bandwidth equal to $\Delta\nu$, the *coherence time* is given by

$$\tau_c = \frac{1}{\Delta\nu} \tag{3.36}$$

The notions introduced here are those of the *temporal coherence*. The notions of the *temporal coherence* will be described in the Chapter 4, relative to the *nulling interferometry*.

3.2 Fresnel zone plate

The *Fresnel zone plate* (FZP) is a device used to focus the light. In this device, the lens uses diffraction instead of reflection or refraction like the traditional lens or mirrors. It is composed of several rings alternatively transparent and opaque as can be seen in FIG.3.10. Its name has been given in honor to Augustin-Jean Fresnel.[14]

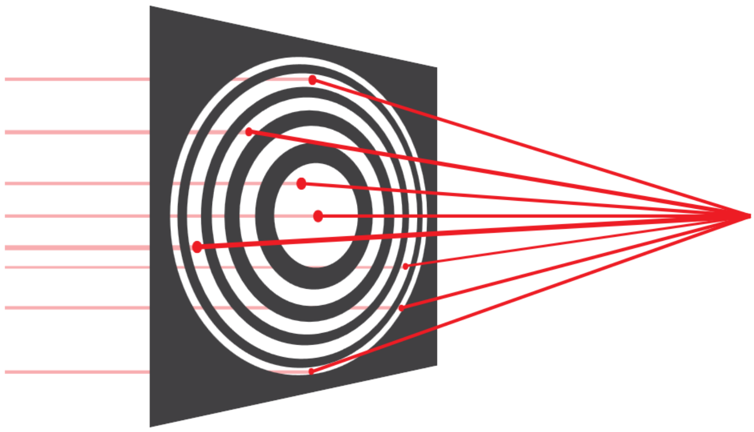


Figure 3.10: Beams crossing a *Fresnel zone plate*. [14]

The alternative rings are named "zones". This term is kept along this report. Those zones are spaced and sized so that the light crossing the transparent zones will interfere constructively at the desired focus. To compute the radius and widths of the zones, the light that arrives on the plate is supposed to come from a source located far away, and considered as a plane wave.

The optical path length (OPL), is the geometric distance between two points multiplied by the index of refraction *n*. Here, as vacuum is considered, *n*=1. According to FIG.3.11(a), the OPL, between a point of the plate and the focus is given by

$$l = \sqrt{r^2 + f^2} \tag{3.37}$$

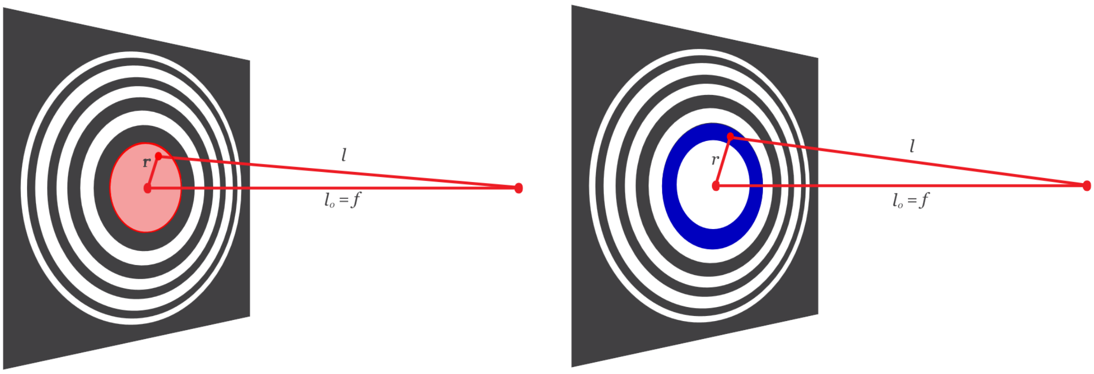
With *f*, the focal distance and *r*, the distance of a point of the plate and the optical axis.

In the first zone which is transparent (FIG.3.11(a)), the light coming from the points of the plate will interfere constructively at the focus if the OPL does not differ by more than half of the wave length, $\frac{\lambda}{2}$. So, the points of the first zone have to satisfy the inequality

$$l - f < \frac{\lambda}{2} \tag{3.38}$$

The second zone which is opaque (FIG.3.11(b)), covers a surface where the light will interfere destructively at the focus. The condition on the OPL is given by

$$\frac{\lambda}{2} < l - f < \lambda \tag{3.39}$$



(a) Constructive interference in red. (b) Destructive interference in blue.

Figure 3.11: Interference in the *Fresnel zone plate*. [14]

By following the same reasoning zones by zones, an alternation of constructive transparent zones, and destructive opaque zones is obtained. The n^{th} zone, opaque (if n is even) or transparent (if n is odd), has to satisfy the conditions

$$\frac{(n-1)\lambda}{2} < l - f < \frac{n\lambda}{2} \tag{3.40}$$

Using Eq.3.37, r_n, the radius of the boundary between the zones *n* and *(n+1)*, verifies

$$l - f = \sqrt{r_n^2 + f^2} - f = \frac{n\lambda}{2} \tag{3.41}$$

Solving for r_n, it yields to

$$r_n = \sqrt{n\lambda(f + \frac{n\lambda}{4})} \tag{3.42}$$

Concerning the width of the n^{th} zone, it is simply

$$\Delta r = r_n - r_{n-1} \tag{3.43}$$

It is important to note that the radius of the zones, given by Eq.3.42 depends on the focal length f and the wavelength λ. The pattern of the FZP is then a function of f and λ.

It is not possible to manufacture a FZP. Indeed, a support has to maintain the opaque zones. The solution is the *photon sieve* which consists of filling the constructive zones of the FZP by holes, as shown in FIG.3.12.

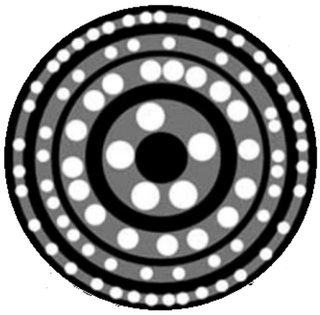


Figure 3.12: From a *Fresnel zone plate* to a *photon sieve*. [15]

3.3 Photon sieve behaviors

The FIG.3.13 shows focused spot of light obtained with a *photon sieve* when the location of the observation plane varies. The central spot is obtained when the observation plan is at the focal plane. The size of the spot increases when the observation plane is placed before or after the focal plane. It can be noted that no spherical aberrations occur.

Figure 3.13: Variation of the observation plan along the axis. The spot increases when the observation plane is placed at positions that differ from the focal plane. [16]

The *photon sieve*, as a diffractive element, has a high chromatic dispersion. Indeed, the focal length varies linearly with the wavelength. This effect is illustrated in FIG.3.14. The right part of the figure shows the variation of the spot in the observation plane with respect to the wavelength. The spot obtained at 632.8nm is the smallest one, knowing that the *photon sieve* is designed for this wavelength. The spots obtained at 400nm and 700nm are much larger. The bandwidth in which the effect of the chromatic dispersion is negligible is approximately equal to 10nm. The left part of the figure shows the focal distance obtained for different wavelengths.

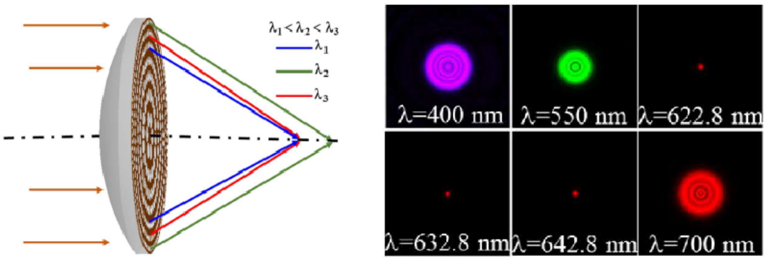
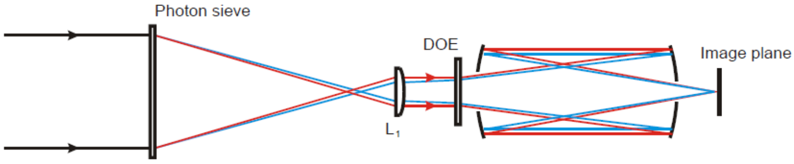
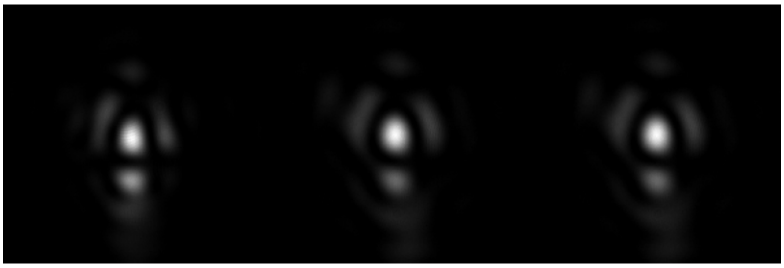


Figure 3.14: Chromatic dispersion of a *photon sieve* of 632.8nm. The spots obtained at 400nm and 700nm are expanded. [17]

However, it is possible to remove significantly the dispersion by using a secondary optical system as shown in FIG.3.15. This secondary optical system, called a dispersion corrector, is described in FIG.3.15(a). The FIG.3.15(b) shows, for a *photon sieve* of 532nm, a constant focal spot for 514nm, 532nm and 543nm. Thanks to the dispersion corrector, the bandwidth in which the chromatic dispersion is negligible is increased to 40nm.[16]



(a) Description of the optical system. It uses a Diffractial Optical Element (DOE).



(b) Focal spot at 514nm, 532nm and 543nm.

Figure 3.15: Dispersion corrector. [16]

3.4 Photon sieve manufacturability

The process used to manufacture membrane *photon sieve* is called the *photolithography*. The process is briefly described without going through the details, which are out of context.[16][33]

According to FIG.3.16, it can be seen that the membrane, is an aluminum-coated polymer film. The polymer used is the CP1 polymide, which is suitable to the space environment. Indeed, its near-zero CTE (Coefficient of Thermal Expansion) is necessary to avoid troubles of pattern changing due to the unstable thermal environment experienced in space.[19]

At first, a photomask is performed from a computerized data file and acts as a master during the *photolithography*. This one is made of glass (fused silica generally) coated by a layer of chromium. This layer is etched with the pattern of the *photon sieve*.

The mask, the membrane and a photoresist placed between them form an assembly which is exposed to UV. Then, the photoresist is etched when the master is removed. Finally, the photoresist has an etch exposed oxide on the aluminum, and is removed once the process is done. The smallest holes possible to manufacture are micron sized.

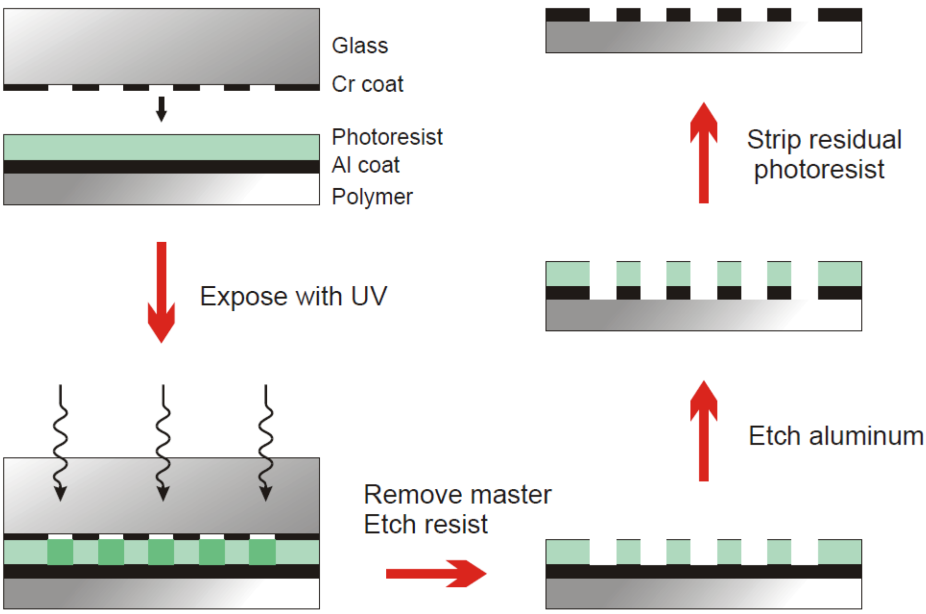


Figure 3.16: Photolithography process to produce a *photon sieve*. [16]

3.5 *FalconSAT-7*

3.5.1 Goals

FalconSAT-7 is a satellite prototype and is the first one to involve a *photon sieve*. [19] The project, studied by the USAFA (United States Air Force Academy) in Colorado Springs, began in the early 2010s but has not been launched yet. The goals are:

1. The deployment of a *photon sieve* telescope from a folded configuration,
2. To image the hydrogen alpha (H-Alpha) band of the sun in order to study the chromosphere in LEO,
3. To determine the imaging performance of a *photon sieve* in space.

The FIG.3.17 represents a CAD (Computer-aided design) of *FalconSAT-7*. It illustrates the *photon sieve* in its deployed configuration. The *photon sieve* is the circular part of the whole membrane.

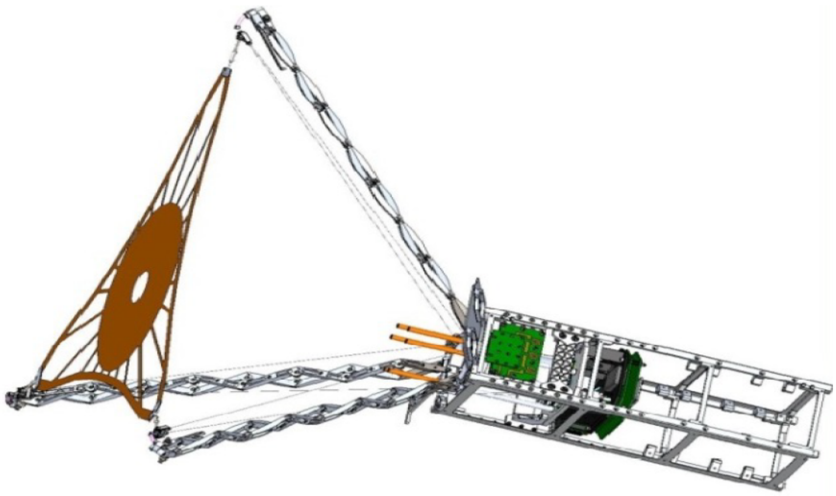


Figure 3.17: CAD of *FalconSAT-7*. The membrane *photon sieve* is in its deployed configuration. [18]

3.5.2 Spacecraft

The spacecraft is a 3U CubeSat nanosatellite which dimensions are 30x10x10cm. Its weight is approximately equal to 5kg and respects then the CubeSat standardization.

The payload, which is the world's first membrane telescope, is called *Peregrine*. It is constituted of the *photon sieve* membrane, the deployment structure, the secondary optics, two cameras and the electronics. *Peregrine* is placed in half of the 3U CubeSat.

The other half is filled by the bus avionics. The spacecraft has a 3-axis control in order to point the sun and is powered thanks to deployable solar panels.

3.5.3 Deployment of the membrane

The membrane is folded during the entire launch and is only deployed once the solar panels are operating effectively.

The deployment structure is constituted of three spring-load pantographs which are stowed under compression. Once the satellite reaches the orbit, the hinged end panels are opened by a burn wire, and the pantographs are automatically deployed. The pantographs are linked to the membrane thanks to lanyards and provide a sufficient tension in the membrane to get the required flatness once deployed. This radial tension is equal to 0.5N.

The deployment occurs in two-stage, both spring-loaded. The first one makes the assembly move linearly out of the bus. Then the pantographs are released. This sequencing is illustrated in FIG.3.18.

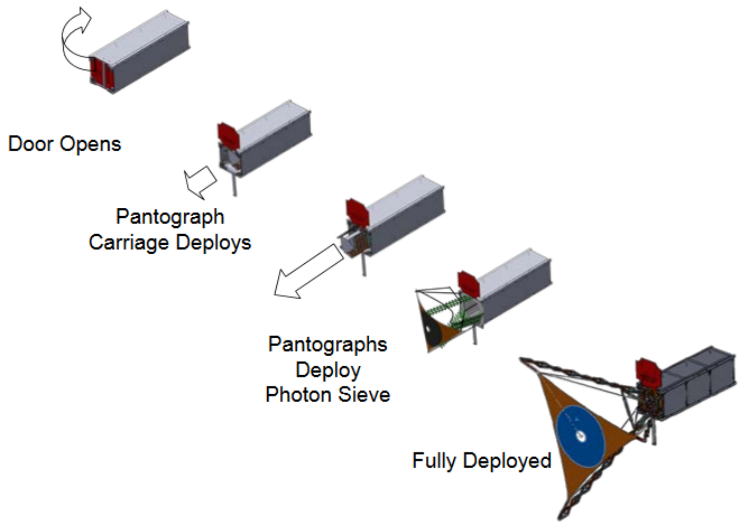


Figure 3.18: *FalconSat-7*: Deployment sequence of the membrane *photon sieve*. [16]

3.5.4 *Peregrine*

Peregrine refers to the whole optical system. It is composed of four subsystems: the primary optical element, which is the *photon sieve*; the deployment structure (already described), the secondary optics and the electronic.

The FIG.3.19 shows *Peregrine* in its stowed configuration (before the deployment). In this configuration, *Peregrine* is sized to be contained in a 1.5U.
The pantographs and the membrane are folded in the upper part. The bottom part contains the secondary optics and the electronics payload.

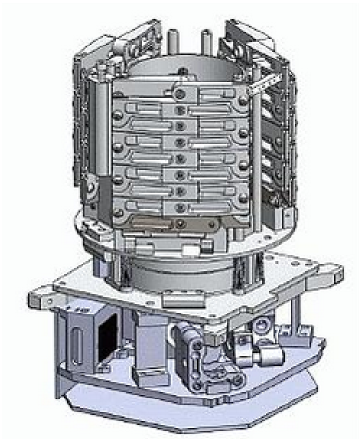


Figure 3.19: *Peregrine* in stowed configuration. [19]

Concerning the parameters of the *photon sieve*, they are the following

- Diameter $\rightarrow$ 0.2 m
- λ $\rightarrow$ 656.45nm
- f $\rightarrow$ 0.4 m

The *photon sieve* is constituted of 2.5 billion holes ranging in size from 2 to 277 microns. The performances achieved by the telescope are

- Resolution $\rightarrow$ 4 μrad
- Field of view $\rightarrow$ 6 arcmin
- Spectral bandwidth $\rightarrow$ 1 $\mathring{A}$

The secondary optics, constituted of four elements, is represented in FIG.3.21. Its exploded view (FIG.3.21) gives a clear understanding of the different elements and their utilities.

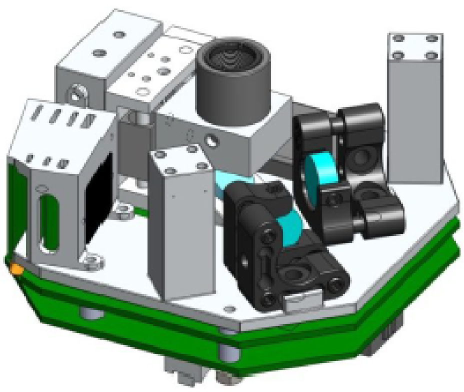


Figure 3.20: Secondary optics of *FalconSat-7*. [18]

The first element is used to collimate the beams. In order to eliminate the unwanted frequencies, the second element is a H-Alpha filter (Fabry Perot bandpass with a 0.1 nm bandwidth centered at 656.45 nm). The beams are finally focused to a CCD sensor thanks to a lens of 10mm diameter with a focal length of 100mm.

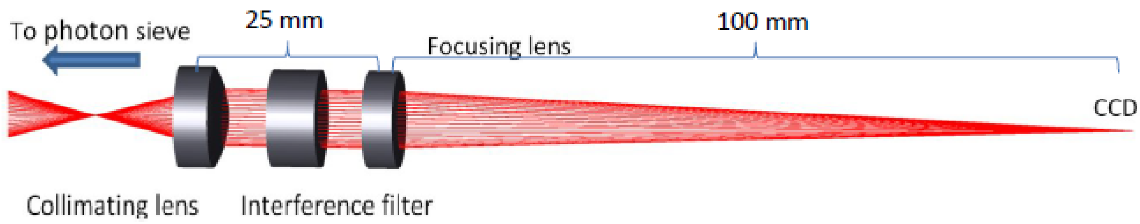


Figure 3.21: Secondary optics of *FalconSat-7* (exploded view). It is composed of 4 elements: a colimator; a filter; a focusing lens and a CCD sensor. [18]

4 Nulling interferometry and optical fiber

The finality of the CubeSat studied in this work is the research of exoplanets, which is performed thanks to the *nulling interferometry*. In this Chapter, the *stellar interferometry* and the *nulling interferometry*, are introduced.

Usually, as it will be explained, an interferometer uses mirrors to propagate the light from the telescopes to the recombiner. However, for our purposes, it has been decided to use *optical fiber* to propagate the light from the photon sieve telescopes to the recombiner. The features of the *optical fiber* are also introduced in this Chapter. [11][34]

4.1 Young experiment

To understand the *stellar interferometery* it is necessary to start from the *Young experiment* taking into account the principles of *coherence*.

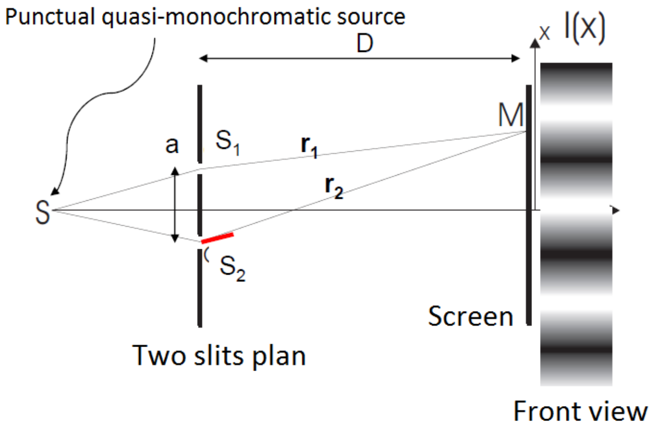


Figure 4.1: Young experiment with a quasi-monochromatic source. [13]

Assuming a punctual quasi-monochromatic source placed in *S*, as illustrated in FIG.4.1; the light passes through the slits placed in S_1 and S_2 to reach the screen (observation plane). From the screen, S_1 and S_2 can be seen as sources of light themselves.

The optical paths, $\overline{SS_1}$ and $\overline{SS_2}$, are equal. Therefore the optical path difference between $\overline{SS_1}$ and $\overline{SS_2}$ is equal to zero and is then inferior to the *coherence lenght* (L_c). That means that the sources S_1 and S_2 are *coherent* with each other. They are even phased in that case.

As the sources S_1 and S_2 are *coherent* with each other, the light that they emit makes interferences in the observation plane as explained in Chapter 3. It is important to mention that the *coherence* between S_1 and S_2 is due to the fact that they are not independent. Indeed, they both come from the source placed in *S*. The *intensity* at the position $\boldsymbol{r}$ of the observation plan is given by

$$I(\mathbf{r}) = I_1 + I_2 + 2\sqrt{I_1 I_2}\langle cos(k\delta_{21})\rangle \quad (4.1)$$

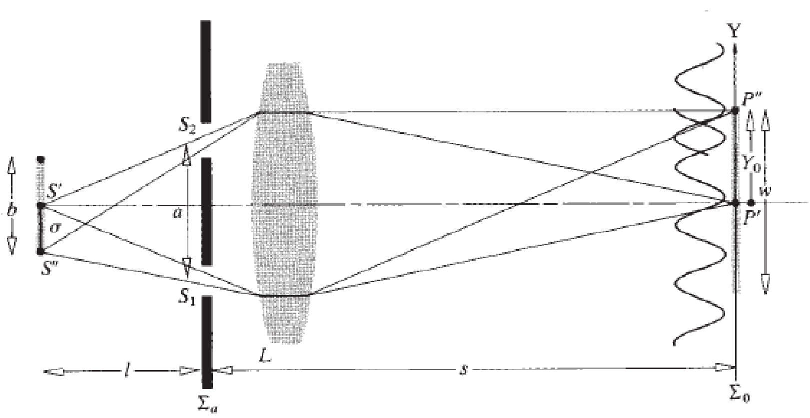
With δ_{21}=r_2-r_1. (FIG.4.1)

If instead of this experimental setup, two punctual sources are placed in S_1 and S_2; this time, the sources, S_1 and S_2, are independent and are said *incoherents*. Therefore, the light emitted by the sources does not interfere and the *intensity* in the observation plane is simply the sum of each *intensities*

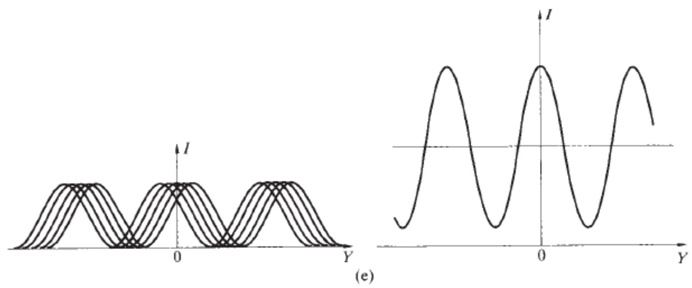
$$I(\mathbf{r}) = I_1 + I_2 \quad (4.2)$$

• Young experiment with extended source

The experiment is now explained for a case where the quasi-monochromatic source is not punctual anymore but extended. This is illustrated in *Fig.*4.2(a).



(a) Schematic view.



(b) Intensity in the observation plane.

Figure 4.2: Young experiment with an extended source. [11]

The extended source of light can be seen as a continuous set of punctual elementary sources, supposed *incoherent* to each other. The source element placed on the optical axis, *S'* (FIG.4.2(a)), produces an *intensity* pattern in the observation plane centered in *P'*, given by

$$dI = 4dI_0cos^2(\frac{Ya\pi}{s\lambda}) \tag{4.3}$$

A source element located out of the optical axis, in *S"* for instance, will produce the same *intensity* pattern, but centered in *P"* this time (FIG.4.2(a)). The FIG.4.2(b) on the left shows the different *intensity* patterns shifted coming from the different source elements. As they are supposed to be *incoherent* to each other, the total *intensity* in the observation plane is the sum of every shifted patterns. The resultant is shown in FIG.4.2(b) on the right.

The quality of the interference fringes produced by an interferometer is defined thanks to the *visibility*, $\mathcal{V}$, defined as follows

$$\mathcal{V} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{4.4}$$

Which takes a value between 0 (no interference) and 1 (perfect interference). The FIG.4.3, shows the evolution of the fringes of interference, when the size of the *incoherent* extended source increases. It can be seen that the *visibility* of the fringes is decreasing when the size of the source is increasing.

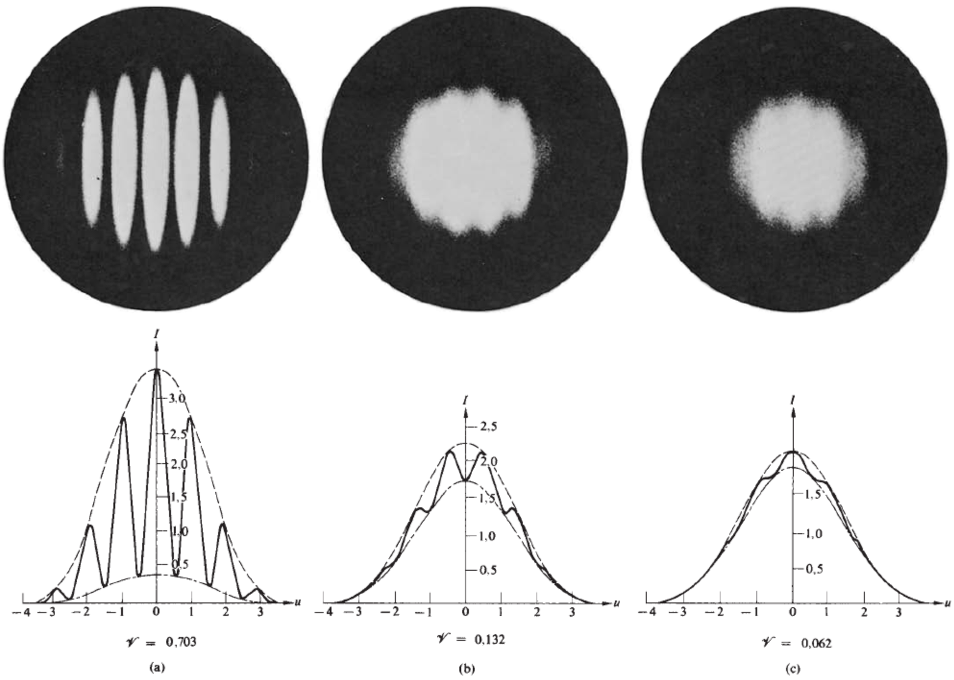


Figure 4.3: Interference figures obtained by increasing the size of the extended source.[11]

This simplified example shows how the fringes of the interference pattern can give information about the size of the source. Generally, the sources are not *incoherent* but partially *coherent*.

4.2 Michelson stellar interferometer

The assembly of the *Michelson stellar interferometer* is presented in FIG.4.4(a). Similarities with the *Young experiment* can be seen. [11]

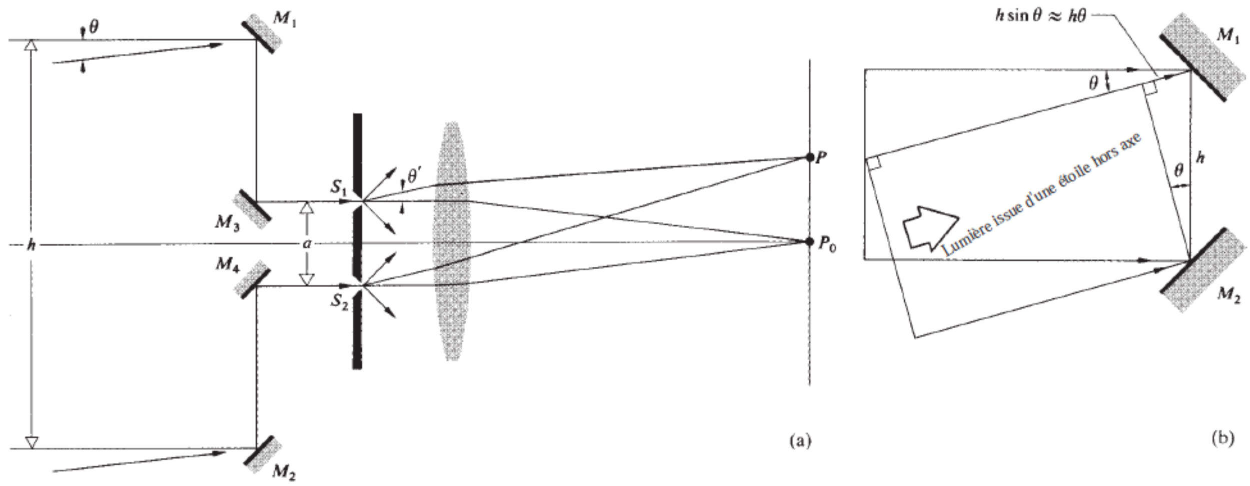


Figure 4.4: Michelson stellar interferometer.[11]

The following discussion is made with respect to the FIG.4.4, where the letter M stands for mirror.

The distance between the mirrors M_1 and M_2 is removable and is noted *h*. These two mirrors collect the light from a far star (represented by the parallel beams) and direct the light to the apertures S_1 and S_2 thanks to the mirrors M_3 and M_4. The two apertures make a system of *Young*'s fringes in the focal plane of the objective of the telescope.

The optical paths, $\overline{M_1M_3S_1}$ and $\overline{M_2M_4S_2}$, have to be equal in order to equalize the phase differences between M_1 and M_2 and the phase difference between S_1 and S_2. Usually, a delay line is added on one side to correct the optical path when it is necessary. These delay lines represent a difficulty of *interferometry*. Indeed, the order of magnitude of the required accuracy is relative to the wavelength studied. For instance, in visible light, the required accuracy is close to 550nm.

Assuming that the assembly points to a far star. The light coming from the pointed far star is represented by the beams which are parallel to the optical axis. The beams of light are phased between M_1 and M_2 and then, are phased between S_1 and S_2. As the sources S_1 and S_2 are phased, they are *coherent* and they create interference fringes centered in P_0 in the focal plane of the telescope (FIG.4.4(a)).

The beams of light coming from a secondary star, close to the one pointed, arrive with an angle θ (with respect to the optical axis) as shown in FIG.4.4(b). In that case, when the beams reach the mirrors M_1 and M_2, they are not in phase anymore. Assuming that the stars emit light in a close bandwidth centered in $\overline{\lambda_0}$; with respect to the FIG.4.4(b), the phase difference is given by

$$\Delta\phi = \overline{k_0} h sin\theta \approx \overline{k_0} h\theta \tag{4.5}$$

The light from the secondary star also produces interference fringes in the focal plane of the telescope. The interference pattern created is similar to the one of the primary star. However, because of the phase difference $\Delta\phi$ between M_1 and M_2 and then between S_1 and S_2, the interference pattern is centered in a point *P* (FIG.4.4(b)).

Since the stars are assumed *incoherent* with each other, the total *intensity* in the focal plane is the sum of the *intensities* created by each star. By adjusting the distance between the mirrors, *h*, it is possible to obtain a shift between the individual patterns such as the minimums of one are positioned on the maximums of the other, the visibility is then $\mathcal{V} = 0$. This case is obtained when the value of *h* is equal to

$$h = \frac{\overline{\lambda_0}}{2\theta} \tag{4.6}$$

And the only unknown, θ, is then found. With similar procedures, it is possible to measure the angular diameter of a star.

4.3 Nulling interferometry

Stellar nulling interferometry is used to research exoplanets. The goal is to block the light coming from the central star thanks to destructive interferences. Indeed, these destructive interferences cancel out the image of the star and the fainter light coming from the exoplanets that orbit around the star can be detected. The way to produce destructive interferences is roughly described in FIG.4.5. [20]

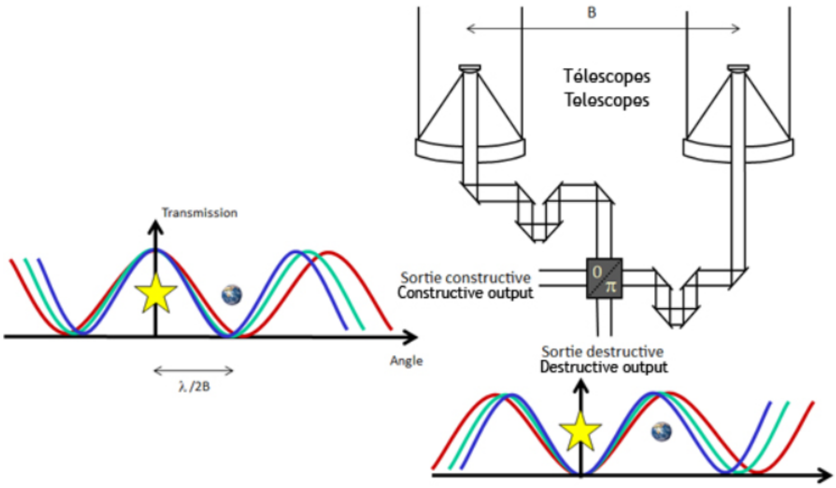


Figure 4.5: Working of a stellar nulling interferometer.[20]

The telescopes are pointed in the direction of the central star. Then, by dephasing of half a wavelength (π rad) the signal received by one of the two telescopes, destructive interferences are created for the star located in the optical axis.

For an object located slightly on the side of the star (an exoplanet for example); the interferences are not destructive. They could even increase the transmitivity of the exoplanets by adjusting the distance between the telescopes with respect to the wavelength studied. This effect is shown in FIG.4.5, in which constructive interferences are produced for the exoplanet when the distance between the telescopes, *B*, is adjusted such as its value verifies

$$\frac{\lambda}{2B} = \theta_{distance} \tag{4.7}$$

With $\theta_{distance}$, the angular separation between the star and the exoplanet.

4.4 Optical fiber

The description of the interferometers in the previous part showed that the light is directed from the telescopes to the recombiner thanks to mirrors. However, it is also possible to use *fiber optic*.[21]

Optical fiber, developed from the 1970's, is used to transmit light between its two extremities. Its working is explained by the principle of total internal reflection. It is constituted of a core made in drawing glass (silica) of tiny diameter (5-10 microns) and is surrounded by a transparent cladding material with a low index of refraction (FIG.4.6). To be correctly injected in the *optical fiber*, the beam of light has to be included in the acceptance cone as shown in FIG.4.6. The angle of the acceptance cone depends on the index of reflection of the different materials constituting the *optical fiber*.

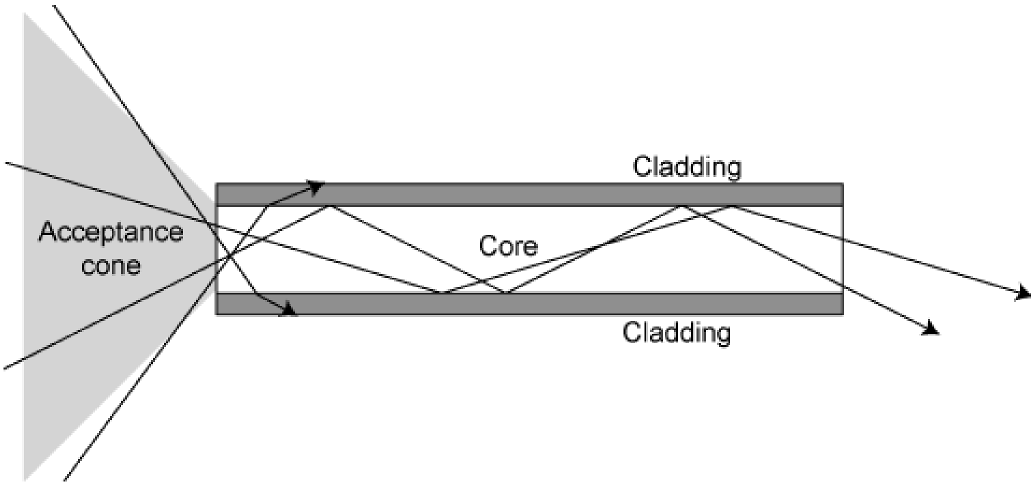


Figure 4.6: Optical fiber representation.[21]

The advantages of *optical fiber* in space environment are the following [21]

1. Compact and lightweight.
2. High accuracy and sensitivity in comparison with typical mechanical architecture.
3. Silica material properties: Usability in harsh, high temperature and rugged environments; and immunity to electromagnetic interference.

The FIG.4.7 illustrates an example of design for a stellar interferometer using *optical fiber*.

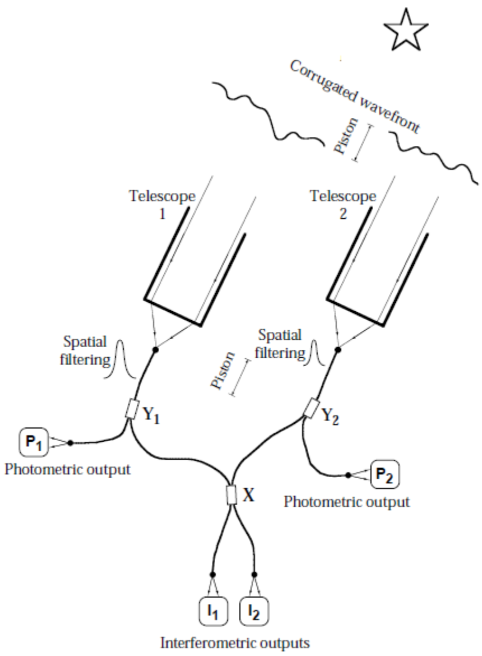


Figure 4.7: Example of design for a stellar fiber interferometer.[22]

5 Optical design

The purpose of this Chapter is to define the specifications for the sizing of the *photon sieve*. To this end, the requirements of an optical system are defined. Then, the optical tools used to analyze the performances of an optical system are presented.

5.1 Requirements

5.1.1 FOV and resolution

The main goal of an optical system is to resolve a specific sized-object over a global object called the *field of view* (FOV) which can be spatial or angular. The resolution spatial or angular, is respectively the minimum angle or distance between two objects as they can be distinguished by the optical system. The FOV and the minimum-sized object to observe depend on the mission goals. [28]

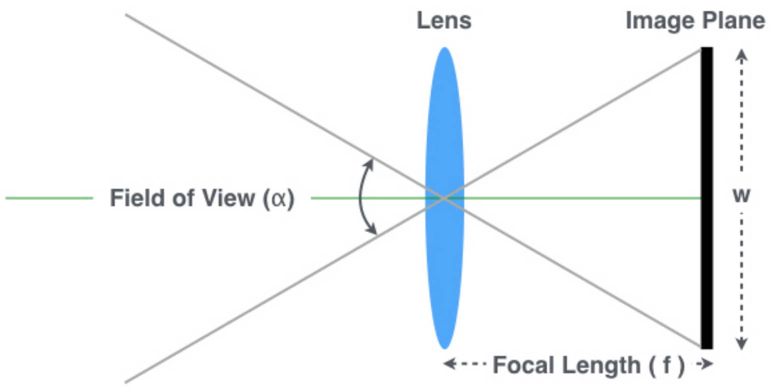


Figure 5.1: FOV for a simple lens.[23]

As shown by the exaggerated example in Fig.5.1, the FOV depends on the focal length and on the size of the image plan. Usually, a CCD or CMOS sensor is positioned at the image plan. These sensors use the photoelectric effect. They transform the incident photons into an electric signal which is amplified, digitized and then treated to obtain a numerical image.

The FOV is then related to the maximum angles of the rays which reached the sensor. The FOV should not be too large to avoid the geometrical aberrations obtained when the paraxial approximation is not respected. To respect the conditions of the paraxial approximation, the angle of the rays with respect to the optical axis have to be small enough. For example, the *Hubble Ultra Deep Field* has a *field of view* of 2.4 arcminutes on each side.[35]

Without detailing them, because out of context, some of the different geometrical aberrations are *defocus*, *spherical aberration*, *coma*, *astigmatism*, *field curvature* and *image distortion*.

5.1.2 Wavelength

The different phenomena or objects that an optical system is designed to observe, emit electromagnetic waves of specific wavelength. For instance, supernovas emit radio waves and black holes emit gamma waves. The wavelengths of the studied objects are an important feature for the optical system.

As shown in Fig.5.2, the light sources do not emit in only one frequency but in a spectrum of frequencies. The spectrum can be extended or narrow, depending on the source. An optical system is designed to take into account from 3 to 5 wavelengths of the spectrum. A weight is associated to the wavelength according to their importance in the spectrum.

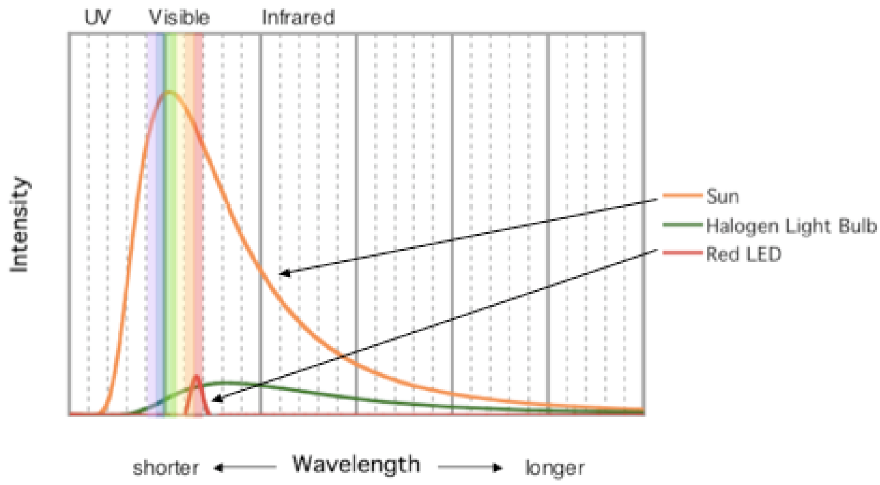


Figure 5.2: Spectrum of light sources. [24]

5.1.3 Brightness gain

An optical system has the ability to increase the brightness of an object. This effect is illustrated in Fig.5.3 for a simple system with an ocular.[25]

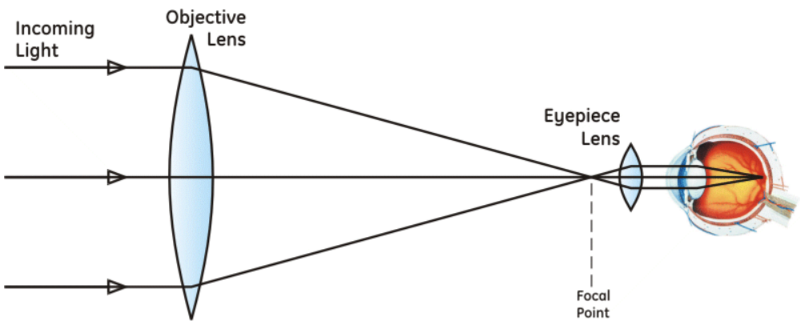


Figure 5.3: Brightness gain. [25]

The brightness gain depends on the light collected by the telescope and therefore, depends on the area by which the light enters in the system. For the simplified case in Fig.5.3, the *light grasp*, G_L is given by the ratio of the area

$$G_L = \frac{\frac{\pi}{4} D_{Opt}^2}{\frac{\pi}{4} D_{Eye}^2} = (\frac{D_{Opt}}{D_{Eye}})^2 \quad (5.1)$$

5.2 Optical tools

5.2.1 PSF

The image of a point through an optical system is not a point. Indeed, due to the aberrations of the system itself and the diffraction, the image of the point is a spot. This spot is the fundamental unit in the image formation of the system.[36]

The PSF (Point Spread Function) is the three-dimensional pattern representing the image of the light from an infinitely small source, a punctual source. The ideal *PSF* is obtained with diffracted-limited systems which are systems without any geometrical aberrations. In this case, the *PSF* is related to the *Airy disk*.

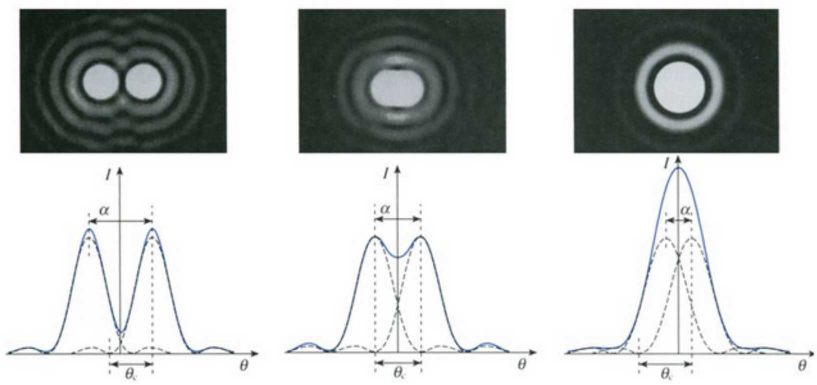


Figure 5.4: Angular resolution with Rayleigh criterion. [26]

The resolution of a system can be determined thanks to the PSF. The FIG.5.4 illustrates this statement with the *Rayleigh criterion* for a diffracted limited system. The figure shows the profile of the PSF of two points. On the left, the angular distance between the object is such that the PSF of the points do not overlap; they are resolved. On the right, the angular distance is not sufficient, the two PSF overlap, and the spot observed does not allow a clear distinction between the points; they are unresolved.

In the middle, the case illustrates the limit of the resolution with respect to the *Rayleigh criterion*. In this case, the two central peaks are separated by a distance equal to the diameter of those central peaks. It yields to an angular resolution equal to [28]

$$\Delta\theta \approx 1.22\frac{\lambda}{D} \tag{5.2}$$

With D, the aperture diameter of the optical system.

The spatial resolution in the object plane or in the sensor, can be extended from the angular resolution thanks to trigonometric formulas.

5.2.2 Encircled energy

The *encircled energy* is a diagram representing the percentage of energy contained in a circled centered in the focus with respect to the radius of this circle. It gives the distribution of energy around the focus. For instance, the encircled energy of the *Airy disk* is illustrated in FIG.5.5. The curve has an important slope at small radius, meaning that a high part of the energy is contained in the first peak of the *Airy disk*. Then, the curve tends asymptotically to 1.

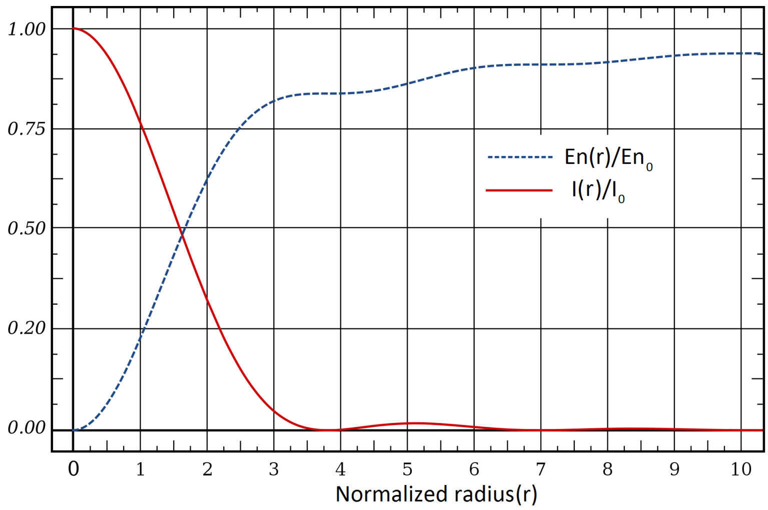


Figure 5.5: Encircled energy of the *Airy disk*. [27]

5.2.3 MTF

The MTF (Modulation Transfer Function) measures the ability of an optical system to transmit the *intensity modulation* of an object to the image created. Due to the aberrations of an optical system, the image contrast is not as pronounced as the object contrast. This is illustrated in FIG.5.6. [28]

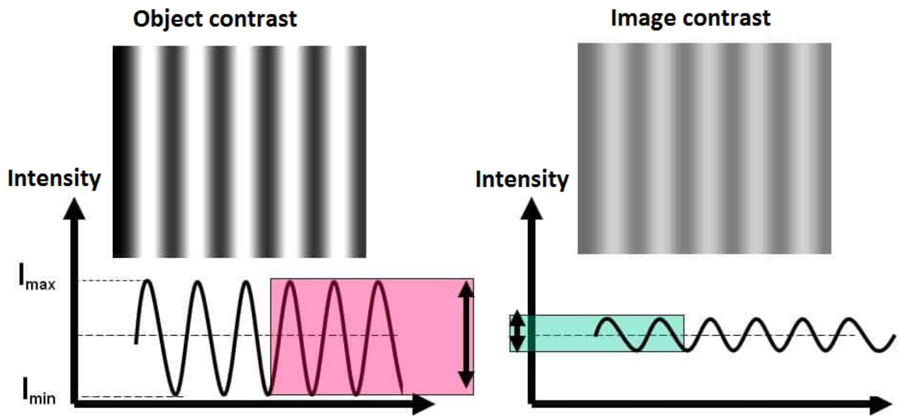


Figure 5.6: Modulation of intensity. [27]

The mathematical definition of the modulation is given by

$$Modulation = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{5.3}$$

And, the MTF is given by

$$\text{MTF} = \frac{\text{Modulation Image}}{\text{Modulation Object}} \tag{5.4}$$

The MTF depends on the spatial frequency. Typical curves are represented in FIG.5.7.

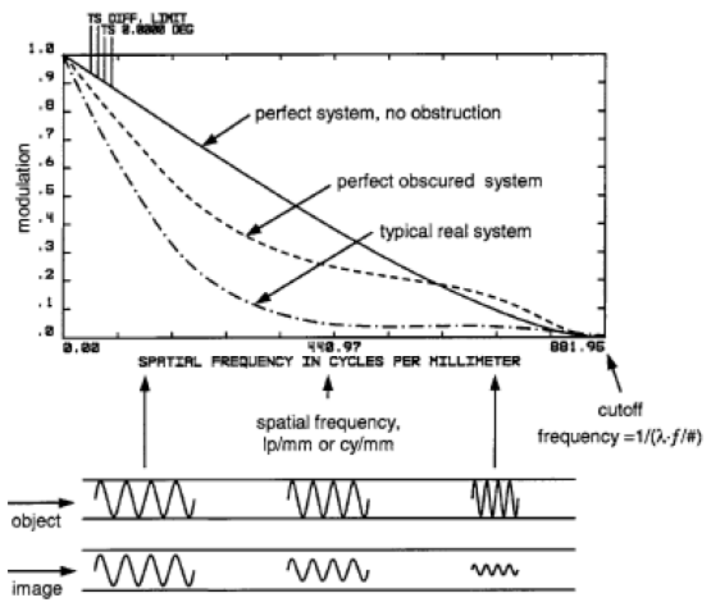


Figure 5.7: Modulation transfer function with respect to the spatial frequency. [28]

The cutoff frequency is the frequency at which the MTF becomes zero and is given by

$$\nu_{cutoff} = \frac{1}{\lambda \frac{f}{D}} \tag{5.5}$$

5.3 Specifications

In our case, the analysis is global. Indeed, no accurate requirements are stated. Given that the final goal of the interferometer is to study exoplanets, it has been decided to make the analysis in visible light, between 400 and 700nm.

The sizing of the *photon sieve* is studied in order to be injected in *optical fibers*. To this end, the PSF is adapted. The goal is to make the first peak of *intensity* suitable with the core of the *optical fiber* whose measure is between 5 and 10 microns. It will give a criteria for the diameter of the *photon sieve* with respect to the focal length and the wavelength.

Regarding the energy and its distribution, the *surface ratio*, which is the ratio between the open surface and the whole surface of the *photon sieve*, and the *encircled energy* are suitable.

Moreover, the size of the holes in the *photon sieve* has to be analyzed as well. Indeed, they could not be smaller than the manufacturing limits.

Therefore, all those parameters and tools are required to be implemented to the *Matlab* propagator.

6 Development of a propagator

It exists many software for optical design. Among them: *CodeV*, *Zemax*, *Olso* and *ASAP*. All those have differences that make their advantages and drawbacks.[5]

There are two categories in optical design software:

1) Classical lens-design software: Used to optimize the image-forming. The data of the system are entered in the software and the performances can be analyzed. The data of the system are the sizes and positions of the lenses, the wavelenght studied,... and other else. (*CodeV*, *Zemax*, *Olso*).

2) Illumination packages: Used to make more detailed analysis of the optimized system; for instance, the effect of the sunlight in a designed telescope. In this case, *ASAP* and *Zemax* are suitable.

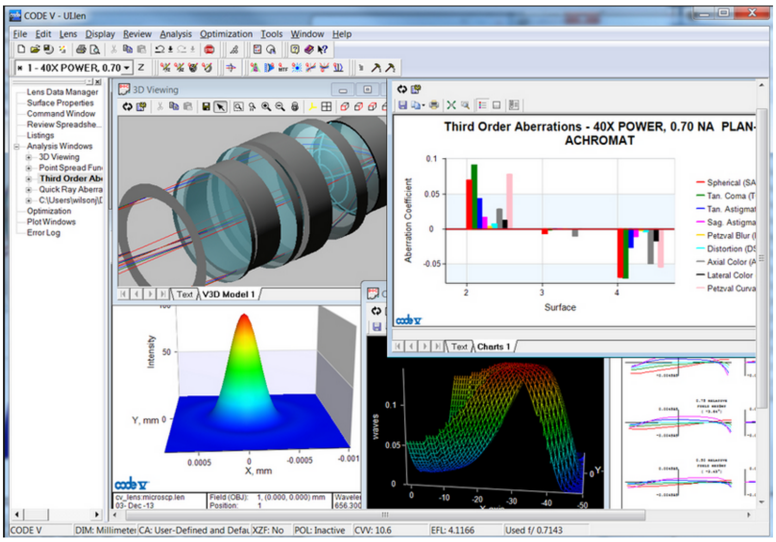


Figure 6.1: Interface of Code V. [1]

However, all of these software are ray tracing software. They simulate the light as beams of light and used geometrical optics to propagate the beams. The photon sieves are diffractive elements and their behavior can not be described by geometrical optics. Indeed, the wave nature of light is in the field of optical physics. The software presented are then not suitable to analyze the *photon sieves*.

Therefore, a light propagator using optical physics, specially dedicated to the *photon sieves*, has been implemented in the *Matlab* environment. To be more precise, the propagator is based on Fourier optics which is a particular section of the optical physics. It is designed to simulate the propagation of the light from the *photon sieve* to an observation screen. *Matlab* is well adapted to this task because it is optimized to work with vectors and matrices. Indeed, the *photon sieves* and the observation screen can be described with arrays of two dimensions corresponding to the two physical dimensions thanks to the use of scale factors. The development of the propagator has been done using the textbook "*Computational Fourier Optics*" as support. [3]

In the propagator, the light is modelized as monochromatic, which is, as said previously, an idealization of the reality. Moreover, the light is not represented as beams anymore but as planar wavefront. In ray tracing, the beams of light representing a far object, as an exoplanet, are parallel. In the wavefront description, a far object is represented by a planar wavefront. A representation of the light emited by a source as beams and wavefronts is illustrated in FIG.6.2. One can see that the shape of the wavefront is spherical close to the punctual source and becomes planar when the distance with the source increases. The planar wavefront is then well adapted to represent astronomical objects.

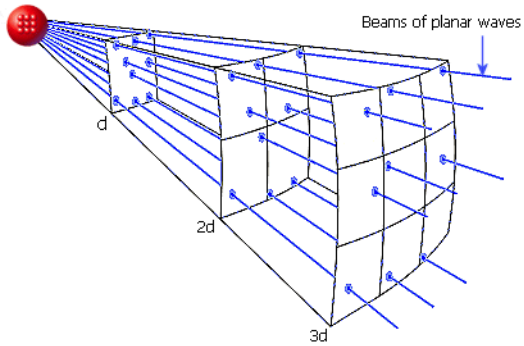


Figure 6.2: Representation of the light as beams and wavefronts. [2]

During this Chapter, the elements relative to *Matlab*, the **variables** and the **functions**, are written in bold letters to make it more readable.

6.1 From Huygens-Fresnel to Fourier optics

As indicated by its name, the Fourier optics uses the Fourier transform. To begin this section, it is necessary to remind the Fourier transform. Then, the propagation of the light is explained and mathematically described.

• Remind of the Fourier transform

The spatial *Fourier transform* (FT) in two dimensions is given by

$$F(f_x,f_y) = \iint_{-\infty}^{\infty} f(x,y) exp\left[-j2\pi(f_x x + f_y y)\right] dxdy \tag{6.1}$$

f_x and f_y are spatial frequencies associated to *x* and *y*. The *Fourier transform* can also be noted using the following short notation

$$F(f_x,f_y) = \mathfrak{F}\{f(x,y)\} \tag{6.2}$$

The inverse *Fourier transform*, is given by

$$f(x,y) = \iint_{-\infty}^{\infty} F(f_x,f_y)exp\left[+j2\pi(f_x x + f_y y)\right] df_x df_y \tag{6.3}$$

With the short notation

$$f(x,y) = \mathfrak{F}^{-1}\left\{F(f_x,f_y)\right\} \tag{6.4}$$

The relation between the inverse and direct transforms is called the identity relation and is given by

$$f(x,y) = \mathfrak{F}^{-}1\mathfrak{F}\{f(x,y)\} \tag{6.5}$$

An important theorem of the *Fourier transform*, particularly in Fourier optics, is the *convolution theorem* which gives a simplified relation for the *Fourier transform* of a convolution

$$\mathfrak{F}\left\{\iint f(\zeta,\eta)h(x-\zeta,y-\eta)d\zeta d\eta\right\} = \mathfrak{F}\{f(x,y)\}\,\mathfrak{F}\{h(x,y)\} \tag{6.6}$$

$$= F(f_x,f_y)H(f_x,f_y) \tag{6.7}$$

• Rayleigh-Sommerfeld propagation

The FIG.6.3 shows a source plane with a signal distribution $U_1(\zeta,\eta)$. The signal represents the *intensity* of the light. The objective is to compute the resulting signal, $U_2(x,y)$ in the observation plan and then the *intensity* (Reminder: the *intensity* is given by $I_1 = |U_1|^2$). [6][7]

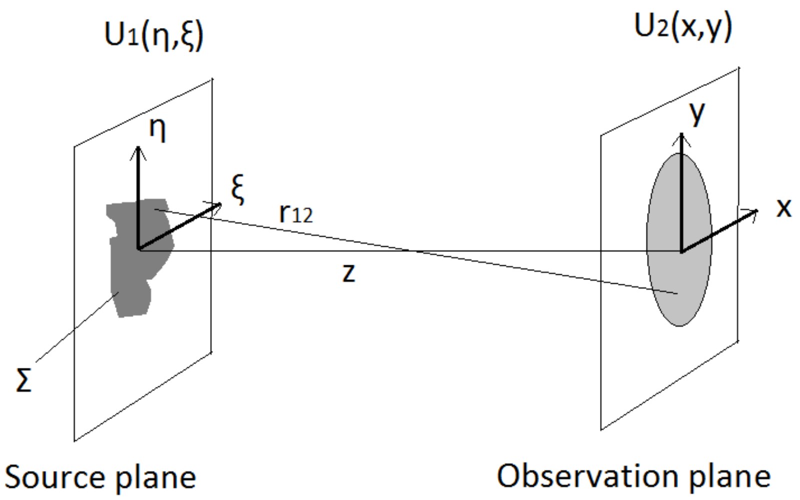


Figure 6.3: Geometrical description of the propagation of the light between two parallel planes. [3]

To compute the signal in the observation plane, the *Huygens-Fresnel principle* is applied. Therefore, every point of the source plane are seen as sources themselves. By integrating on the whole surface, *Sommerfeld* found the following formulae to compute the signal in the observation plan named the *Rayleigh-Sommerfeld solution*

$$U_2(x,y) = \frac{z}{j\lambda} \iint_{\Sigma} U_1(\zeta,\eta) \frac{exp(jkr_{12})}{r_{12}^2} d\zeta d\eta \qquad (6.8)$$

With r_{12}, the distance between a point of the source plane and the point in the observation plane where the response is computed; λ, the wavelength of the monochromatic light assumed and z the distance between the planes.

The total response in a point of the observation plane is then the superposition of the responses of every "elementary sources". The exact expression of r_{12} is

$$r_{12} = \sqrt{z^2 + (x-\zeta)^2 + (y-\eta)^2} \qquad (6.9)$$

The square root of r_{12} makes the *Rayleigh-Sommerfeld solution* numerically unstable. It also increases greatly the computation time. Some hypothesis are then made to obtain approximated solutions more numerically convenient. These approximated solutions are the *Fresnel approximation* and the *Fraunhofer approximation*.

To give a better understanding of the problematic, an illustration in the context of the *photon sieves* is given in FIG.6.4.

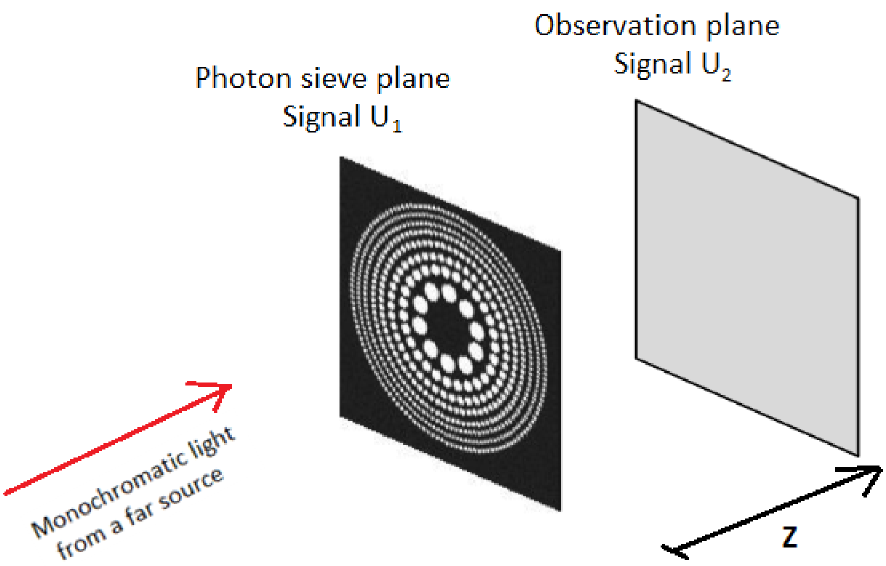


Figure 6.4: Geometrical description of the propagation in the context of the *photon sieves*.

• Fresnel approximation

The *Taylor series* of a square root is

$$\sqrt{1+x} = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + ... \quad if \quad -1 < x < 1 \tag{6.10}$$

This development is applied to the distance term, r_{12}, and only the first degree terms of *x* are kept. It becomes

$$r_{12} \approx z\left[1 + \frac{1}{2}(\frac{x-\zeta}{z})^2 + \frac{1}{2}(\frac{y-\eta}{z})^2\right] \tag{6.11}$$

This expression is used for the phase term of Eq.6.8 which is in the exponential. For the denominator term, the approximation $r_{12} = z$ is used. Finally, the *Fresnel approximation* is

$$U_2(x,y) = \frac{e^{jkz}}{j\lambda z}\iint_{\Sigma} U_1(\zeta,\eta)exp\left(j\frac{k}{2z}\left[(x-\zeta)^2 + (y-\eta)^2\right]\right)d\zeta d\eta \tag{6.12}$$

This expression can be written as the following convolution integral

$$U_2(x,y) = \iint_{\Sigma} U_1(\zeta,\eta)h(x-\zeta, y-\eta)d\zeta d\eta \tag{6.13}$$

With *h(x,y)* the impulse response

$$h(x,y) = \frac{e^{jkz}}{j\lambda z} exp\left[\frac{jk}{2z}(x^2+y^2)\right] \tag{6.14}$$

The *Fourier transform* of the impulse response is called the *transfer function* and its expression is

$$H(f_x,f_y) = e^{jkz} exp\left[-j\pi\lambda z({f^2}_x + {f^2}_y)\right] \tag{6.15}$$

Thanks to the *convolution theorem*, the signal in the observation plan is given by the useful expression

$$U_2(x,y) = \mathfrak{F}^{-1}\left\{\mathfrak{F}\left\{U_1(x,y)\right\} H(f_x,f_y)\right\} \tag{6.16}$$

The *Fresnel approximation* can be used when the approximation made on the square root development does not make a consequent impact on the phase. A criterion used to verify this hypothesis is based on the *Fresnel number*. This one, noted N_F is a dimensionless number given by

$$N_F = \frac{w^2}{\lambda z} \tag{6.17}$$

With *w*, the maximal distance of the aperture in the source plane.The *Fresnel approximation* is considered as valid for a value of N_F inferior to one, or slightly higher of one. Therefore, the observation plane has to be sufficiently far. This is expressed by a large value of *z* with respect to *w*.

• Fraunhofer approximation

The *Fraunhofer approximation*, also known as *far field approximation*, supposes a very far observation plane.

Developing the quadratic terms of Eq.6.12 yields to

$$U_2(x,y) = \frac{e^{jkz}}{j\lambda z} exp\left[j\frac{k}{2z}(x^2+y^2)\right] \iint_\Sigma U_1(\zeta,\eta) exp\left[j\frac{k}{2z}(\zeta^2+\eta^2)\right] exp\left[-j\frac{2\pi}{\lambda z}(x\zeta+y\eta)\right] d\zeta d\eta \tag{6.18}$$

The assumption of a high distance of the observation plan implies

$$z >> \left(\frac{k(\zeta^2+\eta^2)}{2}\right)_{max} \tag{6.19}$$

This assumption is used in Eq.6.18. It results to the *Fraunhofer approximation* expression

$$U_2(x,y) = \frac{e^{jkz}}{j\lambda z} exp\left[j\frac{k}{2z}(x^2+y^2)\right] \iint_\Sigma U_1(\zeta,\eta) exp\left[-j\frac{2\pi}{\lambda z}(x\zeta+y\eta)\right] d\zeta d\eta \tag{6.20}$$

The integral expression obtained is actually a *Fourier transform* with the substituted variables

$$f_\zeta \rightarrow \frac{x}{\lambda z} \tag{6.21}$$

$$f_\eta \rightarrow \frac{y}{\lambda z} \tag{6.22}$$

The condition, with respect to the *Fresnel number*, to use the *Fraunhofer approximation* is

$$N_F << 1 \tag{6.23}$$

The *Fresnel approximation* and *Fraunhofer approximation* are the formulae implemented in the propagator to compute the propagation of light. The choice between the two expressions is made with respect to the value of the *Fresnel number* of the simulation performed.

6.2 Numerical discretization

6.2.1 Discretization and Shannon-Nyquist Sampling Theorem

To perform *Fourier optics* simulations using a computer, continuous physical data are discretized. For instance, the continuous functions of two dimensions representing the signal in the plans are represented using arrays of two dimensions as well. U_1, function of the continuous variables *x* and *y*, becomes a function of the discrete variables *m* and *n*

$$U_1(x, y) \quad \rightarrow \quad U_1(m\Delta_x, n\Delta_y) \tag{6.24}$$

Where Δ_x and Δ_y are respectively the sample intervals in the *x* and *y* direction. The sampled space, finite, is composed of $M \times N$ samples. Usually, *M* and *N* are supposed even, *m* and *n* are defined in the intervals

$$m \in [-\frac{M}{2}, -\frac{M}{2} + 1, .., \frac{M}{2} - 1] \tag{6.25}$$

$$n \in [-\frac{N}{2}, -\frac{N}{2} + 1, .., \frac{N}{2} - 1] \tag{6.26}$$

The physical area, spanned by the sampled space, has the following dimensions

$$L_x = M\Delta x \tag{6.27}$$

$$L_y = N\Delta y \tag{6.28}$$

These distances are the side lengths of the planes. They have to be higher than the support of the signal; the area where the signal takes significant values. The support in the *x* direction is noted D_x and the support in the *y* direction is D_y. The conditions are written as following

$$D_x < L_x \tag{6.29}$$

$$D_y < L_y \tag{6.30}$$

Those parameters and constraints are illustrated in the context of the *photon sieves* in FIG.6.5. The support of the signal in the *photon sieve* plane is the *photon sieve* itself. Indeed, the signal used to represent the *photon sieve* is a binary signal. It takes the value of 1 in the holes of the *photon sieve* and 0 everywhere else. The support, or in other words, the area where the signal takes significant values is then the *photon sieve*.

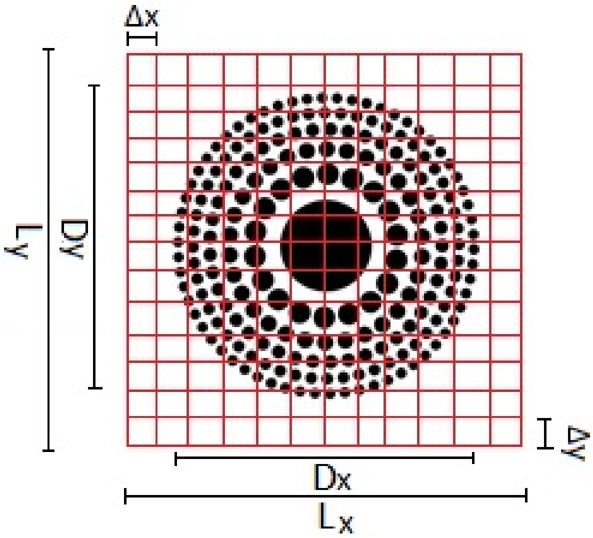


Figure 6.5: Discretization in the *photon sieve* plane.

To preserve the feature of $U_1(x, y)$, the sample intervals have to be small enough. Usually, the spectral content of a signal is not bandlimited. In this case, the *Shannon-Nyquist Sampling Theorem* gives the maximum spatial frequencies which are well represented by samples of the size Δ_x and Δ_y. The values of *Nyquist frequencies* are

$$f_{NX} = \frac{1}{2\Delta_x} \tag{6.31}$$

$$f_{NY} = \frac{1}{2\Delta_y} \tag{6.32}$$

6.2.2 DFT and FFT

DFT stands for *Discrete Fourier Transform*, which is the discretization of the *Fourier transform*. The *Fourier transform* of the signal U_1 is given by

$$U_1^{\mathfrak{F}}(f_x,f_y) = \iint_{-\infty}^{\infty} U_1(x,y)exp\left[-j2\pi(f_x x + f_y y)\right] dxdy \tag{6.33}$$

In the sampled space, the function for U_1 becomes the discrete function U_1^D

$$U_1(m\Delta_x, n\Delta_y) \quad \rightarrow \quad U_1^D(m,n) \tag{6.34}$$

The integrals of Eq.6.33, are approximated by a Riemann sum. With the sample intervals in Eq.6.26, it yields to

$$\iint_{-\infty}^{\infty} ...dxdy = \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \sum_{m=\frac{-M}{2}}^{\frac{M}{2}-1} ...\Delta_x \Delta_y \tag{6.35}$$

As for the spatial domain, the frequency domain has to be discretized. The convention to discretize the frequency domain is to keep the same number of samples than for the spatial domain. Therefore, there are *M* samples for f_x and *N* samples for f_y. The *Nyquist frequencies* are taken as the limits of the sampled frequency domain. It gives

$$f_x \rightarrow \frac{p}{M\Delta_x} \tag{6.36}$$

$$f_y \rightarrow \frac{q}{N\Delta_y} \tag{6.37}$$

With *p* and *q* included in similar intervals than *m* and *n*

$$p \in [-\frac{M}{2}, -\frac{M}{2}+1, .., \frac{M}{2}-1] \tag{6.38}$$

$$q \in [-\frac{N}{2}, -\frac{N}{2}+1, .., \frac{N}{2}-1] \tag{6.39}$$

Concerning the frequency sample intervals, they take the values

$$\Delta f_x = \frac{1}{M\Delta x} = \frac{1}{L_x} \tag{6.40}$$

$$\Delta f_y = \frac{1}{N\Delta y} = \frac{1}{L_y} \tag{6.41}$$

With the discretization notation of the spatial domain and of the frequency domain, the exponential term of the *Fourier transform* is given by

$$exp\left[-j2\pi(f_x x + f_y y)\right] \rightarrow exp\left[-j2\pi\left(\frac{pm}{M} + \frac{qn}{N}\right)\right] \tag{6.42}$$

And the DFT is given by

$$U_1^{\mathfrak{F}D} = \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \sum_{m=\frac{M}{2}}^{\frac{-M}{2}-1} U_1^D exp\left[-j2\pi\left(\frac{pm}{M}+\frac{qn}{N}\right)\right] \quad (6.43)$$

With $U_1^{\mathfrak{F}D}$ the DFT of U_1^D. The term $\Delta_x\Delta_y$ is removed in Eq.6.43. Actually, this scaling term is taken into account in the inverse DFT, which is given by the expression

$$U_1^D = \frac{1}{MN} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \sum_{m=\frac{-M}{2}}^{\frac{M}{2}-1} U_1^{\mathfrak{F}D} exp\left[+j2\pi\left(\frac{pm}{M}+\frac{qn}{N}\right)\right] \quad (6.44)$$

Indeed, the prefactor $\frac{1}{MN}$ is equal to

$$\frac{1}{MN} = \Delta x \Delta y \Delta f_x \Delta f_y \quad (6.45)$$

It exists a powerful algorithm to compute efficiently the DFT called the *Fast Fourier Transform* (FFT). This one is implemented in Matlab with the function **fft2** for the direct transform and **ifft2** for the inverse transform. To correctly use the FFT functions with *Matlab*, a shift of the arrays (representing the signals) has to be proceeded thanks to the function **fftshift**.

In the propagator, as the *photon sieve* is symmetrically circular, similar sampling are used for the two dimensions

$$M = N \quad (6.46)$$

$$\Delta x = \Delta y \quad (6.47)$$

$$L_x = L_y = L \quad (6.48)$$

6.2.3 Impact of the DFT

An aliasing phenomenon, called the periodic extension, appears when the *Discrete Fourier Transform* is performed. This effect is presented in FIG.6.7 which shows the DFT of the sampled squared signal of the FIG.6.6. The analytical *Fourier transform* (TF) of this squared signal, which is the *sinc function*, is also represented ($sinc(x) = \frac{sin(\pi x)}{(\pi x)}$).

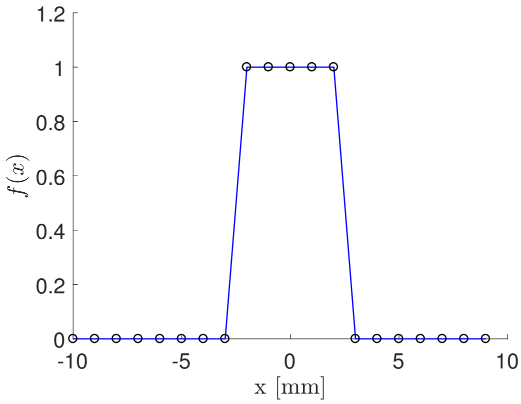


Figure 6.6: Squared signal.

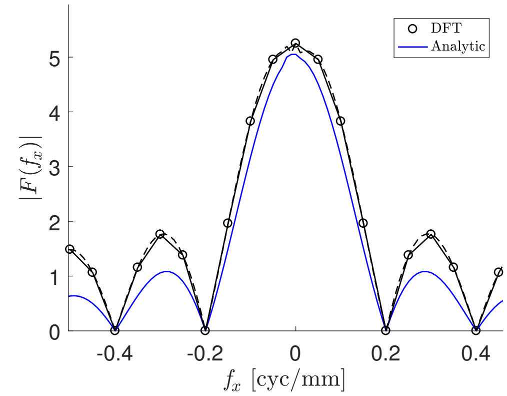


Figure 6.7: Comparison between the results of the DFT and the FT.

One can see that the curve obtained with the DFT is slightly different in comparison with the analytically one. Actually, the DFT of a given function will give the *Fourier transform* as if the signal was periodically expended.

To illustrate this statement, the periodic extension of the squared signal is represented in tilled dots in FIG.6.8.

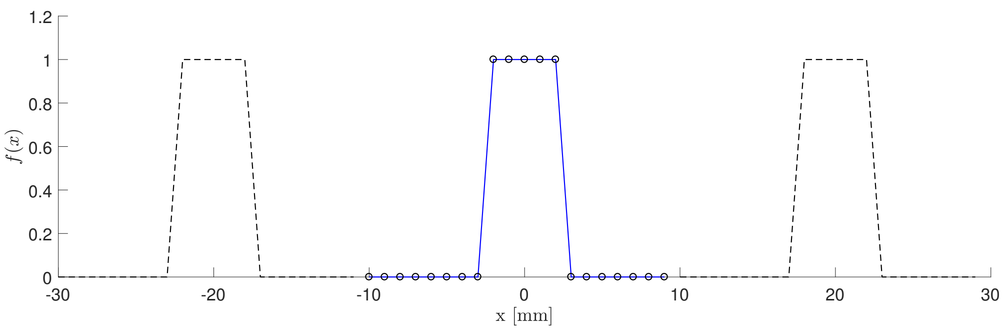


Figure 6.8: Periodic extension of the squared signal.

In FIG.6.9, the *Fourier transform* of the extended signal is also represented in tilled dots. One can see that this curve corresponds perfectly with the results obtained with the DFT.

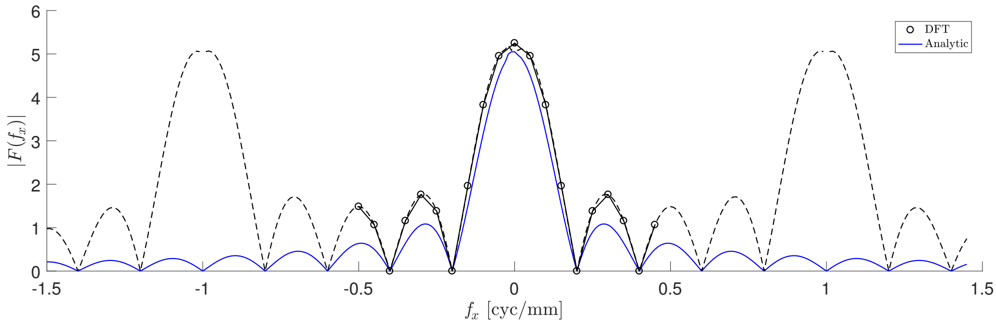


Figure 6.9: Periodic extension of the response (M=20).

The FIG.6.9 shows that this effect mainly impacts the extremities of the response signal. The *periodic extension* effect, introduced by the DFT, can be reduced by an increase of the number of samples.

Indeed, the total span of the squared signal, *L*, is equal to 20mm (FIG.6.6). In the case relative to the FIG.6.9, the number of samples, *M*, is taken equal to 20. Therefore, the sampled interval is

$$\Delta x = \frac{L}{M} = \frac{20}{20} = 1 \quad [mm] \tag{6.49}$$

As said previously, the values of the extremities in the frequency domain are equal to the *Nyquist frequency*. In that case, the *Nyquist frequency* is equal to

$$f_{Nyquist} = \frac{1}{2\Delta x} = 0.5 \quad [cyc/mm] \tag{6.50}$$

The discretized frequency domain is defined from -$f_{Nyquist}$ to $f_{Nyquist}$. The total span in the frequency domain is then 2x$f_{Nyquist}$.

The FIG.6.9 shows that the total span in the frequency domain is actually equal to the distance between the periodic repetitions. Therefore, when the number of samples is increased, keeping the same value of L, the value of Δx is decreased. The distances between the periodic repetitions are then increased, as the *Nyquist frequency* is increased, and the effect of the periodic extension is attenuated.

This analyze is represented in FIG.6.10 which shows the case where 40 samples are used instead of 20. The effect of the periodic extension is well attenuated.

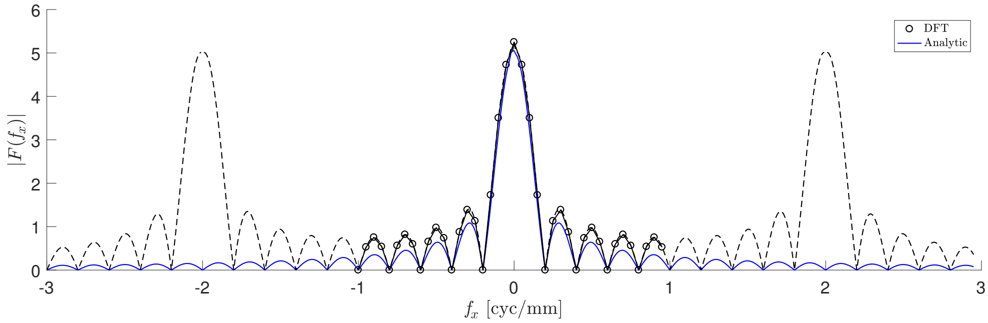


Figure 6.10: Periodic extension of the response (M=40).

The FIG.6.11 represents the periodic effect obtained during a simulation of the propagator with a badly sampled discretization.

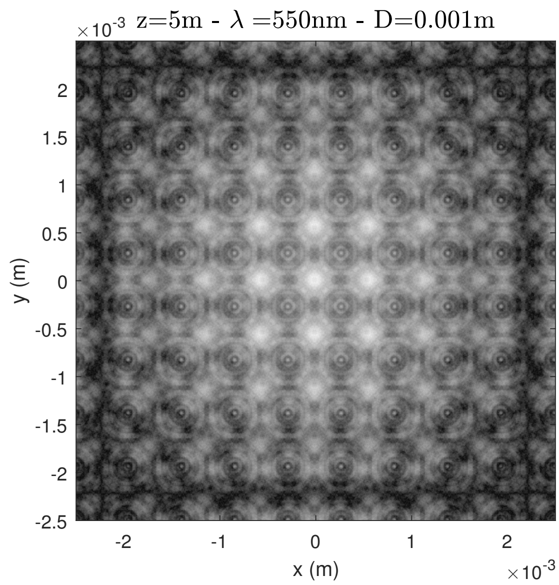


Figure 6.11: Periodic effect obtained during a simulation.

6.3 Propagator

As said in the beginning of this Chapter, the propagator uses a monochromatic representation of light. The wavefront is assumed planar and perpendicular to the z direction. The wavefront arrives then in parallel with the *photon sieve*. The amplitude of the entry signal representing this wavefront is taken unitary. Indeed, the value of the amplitude has only a scaling impact on the *intensity*. The *intensity* is also called the *irradiance*. This term is kept in the following.

The mathematical representation of such a wavefront is

$$U(\mathbf{r}, t) = 1e^{j(\mathbf{zr} - \omega t + \phi)} \tag{6.51}$$

With $\boldsymbol{r}$, the position vector and z, the unitary vector in the z-direction.

The photon sieve is represented by a binary transmittance function. The value of 1 is taken for the transparent zones and a value of 0 is taken for the opaque zones.

6.3.1 Running

The *Matlab* files implemented for the propagator are gathered in the appendix A.

It has been decided to make an interface for the propagator in order to accelerate the simulations and to keep a clear visualization of the parameters of the simulations.

To launch the propagator, the file *Photon_Sieve_Propagator.m* is run. Two types of simulations can be performed. A single simulation, or an "optimization" simulation. This last one performs many propagations, by changing only one parameter of the simulation. The goal is to analyze the impact of this parameter on the behavior of the *photon sieve*.

To recall, a *photon sieve* is based on the pattern of a Fresnel zone plate. The pattern of a Fresnel zone plate is an alternation of transparent and opaque concentric rings of precise size called zones. Therefore, the diameter of a *photon sieve* can be expressed by a number of zones and vice versa.

• Single simulation

In first, the user has to choose between the *photon sieve* and the circle. The circle shape has been implemented because it is a degenerated *photon sieve* and allowed the verification of the good behavior of the propagator.

Then, some parameters of the simulation have to be entered:

- About the *photon sieve*: the focal distance, the density and the diameter/number of zones.
- About the light: the wavelength.

The user has to enter a last parameter, relative to the accuracy of the simulation and then to its speed. The value of this parameter is kept in the variable **Speed**. This variable will be discussed lately in the report.

Finally, the results can be displayed as shown in FIG.6.12.

```
Enter:     Number of holes (1) - Number of zones (2) - Total power (3)
      Encircled energy (4) - Surface ratio (5) - Radius of the "first peak" (6)
 Irrandiance in entry plan (2D) (7) - Irrandiance in observation plan (PSF-2D) (8)
Irrandiance in entry plan (3D) (9) - Irrandiance in observation plan (PSF-3D) (10)
               Profile irrandiance in observation plan (PSF-Profile)(11)
                  Diameter of the smallest holes (12)
```

Figure 6.12: Display single results.

• **Optimization simulation**

In this case, the user has to choose the parameter he wants to analyze the impact (FIG.6.13).

Then, similarly than for the single case, the other parameters have to be entered. Regarding the analyzed parameter, a lower value and a higher value have to be chosen. After that, the number of samples between these two extreme values is chosen.

```
--------------Which parameter do you want to optimize----------------

Enter: Diameter/Zones (1) - Distance of the observation plan (2) - Density (3)

                 Wavelength (4) - Focal distance (5)
```

Figure 6.13: Choice of the parameter to analyze.

The results can be displayed as one can see in FIG.6.14.

```
Enter:     Number of holes (1) - Number of zones (2) - Total power (3)
      Percentage of the irradiance in the fist peak (4) - Surface ratio (5)
Irrandiance in entry plan (2D) (6) - Irrandiance in observation plan (PSF-2D) (7)
Irrandiance in entry plan (3D) (8) - Irrandiance in observation plan (PSF-3D) (9)
               Profile irrandiance in observation plan (PSF-Profile) (10)
            Encircled energy (11) - Radius of the first peak (12)
                  Diameter of the smallest holes (13)
```

Figure 6.14: Display optimization results.

6.3.2 Files and hierarchy

The utilities of the files are slightly described here after.

• **Photon_Sieve_Propagator.m**

This is the main file of the propagator, it has to be run to launch the simulation.

• Propagation.m, Fraunhofer.m and Fressel_TF.m

Fraunhofer.m contains a function used to make a propagation with *Fraunhofer approximation*. To recall, the formula of the *Fraunhofer approximation* is

$$U_2(x,y) = \frac{e^{jkz}}{j\lambda z} exp\left[j\frac{k}{2z}(x^2+y^2)\right] \iint_{\Sigma} U_1(\zeta,\eta) exp\left[-j\frac{2\pi}{\lambda z}(x\zeta + y\eta)\right] d\zeta d\eta \tag{6.52}$$

Which is a *Fourier transform* of the entry signal, U_1.

Fresnel_TF.m makes the propagation with the *Fresnel approximation*. The formula of the *Fresnel approximation* is

$$U_2(x,y) = \mathfrak{F}^{-1}\left\{\mathfrak{F}\left\{U_1(x,y)\right\} H(f_x,f_y)\right\} \tag{6.53}$$

With the *Transfer function*

$$H(f_x,f_y) = e^{jkz} exp\left[-j\pi\lambda z({f^2}_x + {f^2}_y)\right] \tag{6.54}$$

The *Propagation.m* is used to make the decision between the two approximations with respect to the *Fresnel number*.

• Single.m and Optimization.m

These files are used to link the functions in the single simulation case for one and in the optimization simulation case for the other.

• Display_Results_*.m

Displays the result according to the choice of the user.

• Photon_Sieve_Generator.m, Circle_Generator.m and Holes_Generator.m

Photon_Sieve_Generator.m generates the signal in the *photon sieve* plane, U_1. It also computes some features about *photon sieve* (number of zones, number of holes, size of the holes,..). The shape of the *photon sieve* is built with the holes generated by *Holes_Generator.m*.

Circle_Generator.m generates the entry signal in the circle case.

• Encircled_Energy.m

To compute the energy contained in a circle of radius *r*, the *irradiance* has to be integrated on the whole surface of the circle.

However, given the circular symmetry of the *photon sieve*, the encircled energy is computed thanks to the profile of the *irradiance*. Indeed, the increment of energy between two samples in the *irradiance* profile (hatched area in FIG.6.15) is given by

$$\int_0^{2\pi}\int_{x_1}^{x_2}\left(\frac{(I_2 - I_1)}{(x_2 - x_1)}(r - x_1) + I_1\right) r\,dr\,d\theta \tag{6.55}$$

With x_1 and x_2, the coordinate in the x-axis of the two samples.

This circular integration gives the energy contained in the ring delimited by the two samples.

Encircled_Energy.m performs this integration sample by sample on the whole profile.

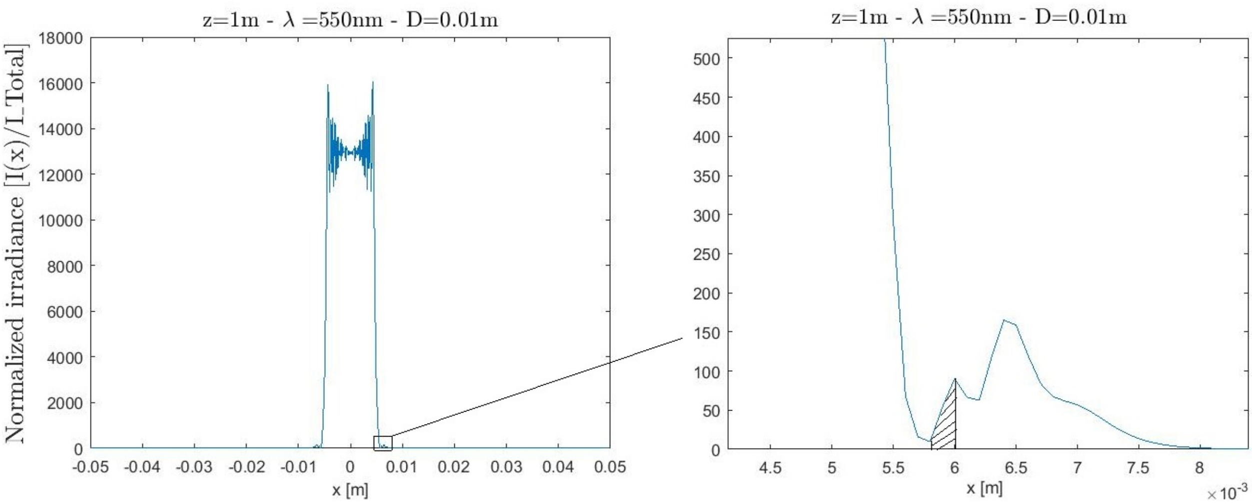


Figure 6.15: Computation of the encircled energy thanks to the *irradiance* profile.

Time_Estimation.m and Diameter_Zones_Max.m

These files are used to make an estimation of the simulation time and to display the size limits to avoid an overflow of memory.

Hierarchy of the files

The hierarchy of the files are illustrated on the organizational chart in FIG.6.16.

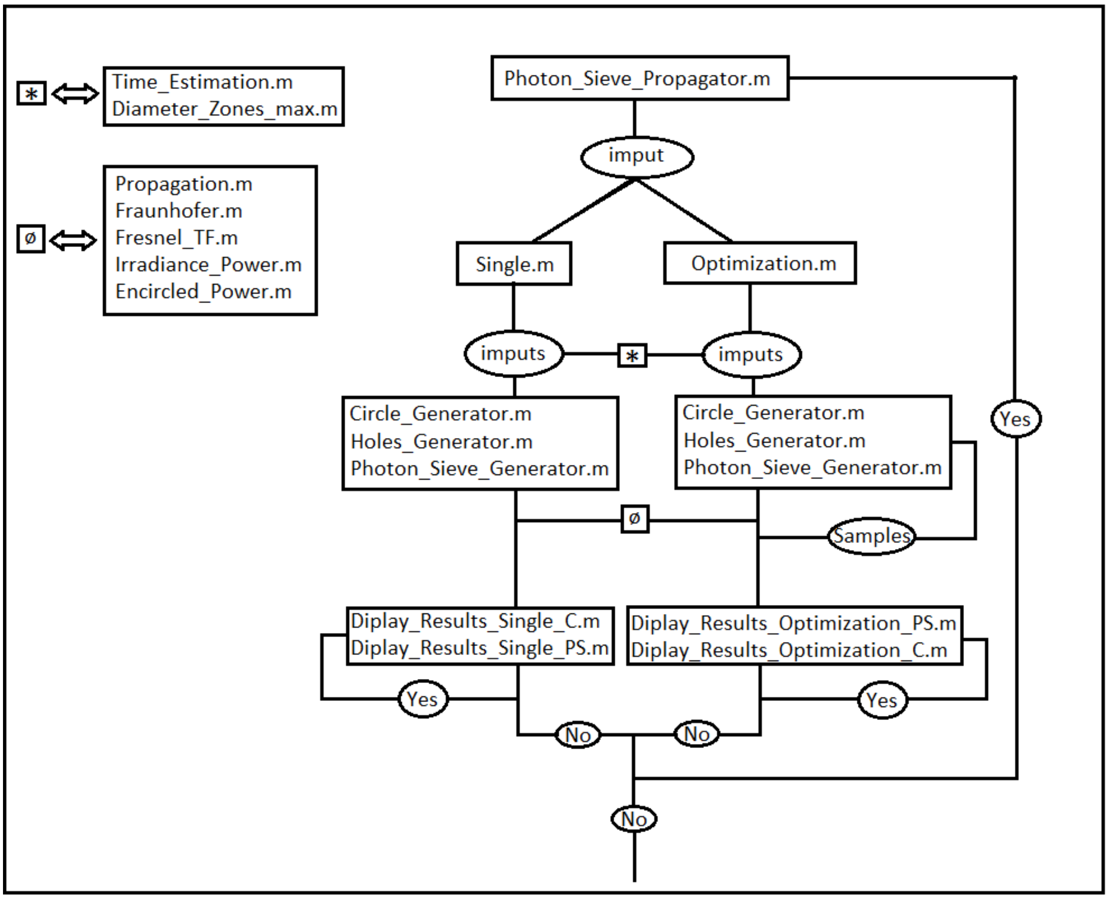


Figure 6.16: Hierarchy of the files. Photon_Sieve_Propagator.m is the file to run to launch of the simulation. At first, the user has to decide between the single simulation and the optimization simulation. Then the parameters of the simulation have to be entered. Finally, the requested results can be displayed. Once the analysis is done, the user can start a new simulation.

6.4 Limitations

6.4.1 Hypothesis

It is important to be critical when looking at the results as they are based on approximate formulas. However, as it will be seen, the propagator gives good results and the trends can be analyzed.

The monochromatic model used does not take into account the coherence effects. Indeed, remembering the formulae of the *coherence time*

$$\tau_c = \frac{1}{\Delta\nu} \tag{6.56}$$

In the case of a monochromatic wave, the bandwidth is $\Delta\nu = 0$, and the *coherence time* is then infinite.

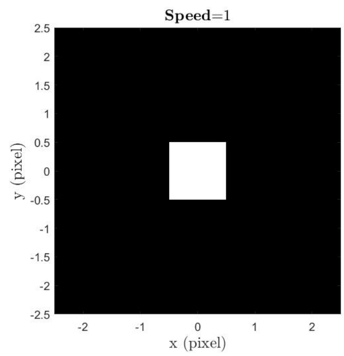
6.4.2 Computer resources

• Sampling.

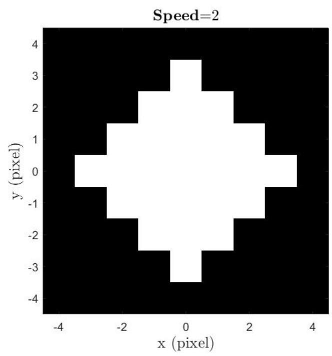
The sampling is defined with respect to the smallest holes of the *photon sieve* to simulate. As said previously, the user can choose the accuracy/speed of the propagation. Its response is recorded in the variable **Speed** which can take the values 1, 2 or 3.

In the case where **Speed** is equal to 1, the sample interval (Δx) is such that the diameters of the holes are equal to 1 sample interval. Then, when the value of **Speed** is 2, the diameters correspond to 7 sample intervals. Finally, when **Speed** is equal to 3, the diameters are represented with 11 sample intervals.

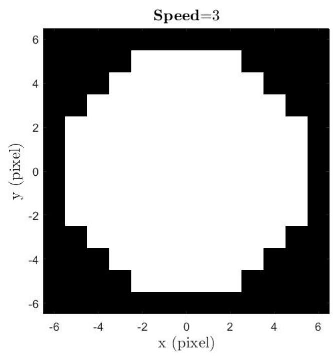
The FIG.6.17(a), 6.17(b) and 6.17(c) show respectively the pixelization of the smallest holes of the *photon sieve* for **Speed**=1, **Speed**=2 and **Speed**=3.



(a) **Speed**=1.



(b) **Speed**=2.



(c) **Speed**=3.

Figure 6.17: Pixelization of the smallest holes with respect to **Speed**.

During the analyze of the results, it has been shown that the results obtained for **Speed**=1 and **Speed**=2 are quite similar and precise in comparison with the ones of **Speed**=3.

So, the best compromise between accuracy and time is **Speed=2**. However, it can be interesting to use **Speed=3** and have a smallest size of the sample interval. For instance, when the value of the first minimum of the *irradiance* profile is sought, the closest value of the sampling is selected for that minimum. Therefore, the selected value is a multiple of the sample interval and is then more precise with a smaller sampling.

• Dimensions

Generating a *photon sieve* is very costly in memory. Indeed, it can reach billions of holes and then requires large arrays. The command **Memory** in *Matlab* displays information about the maximum size of the arrays to avoid an overflow of the memory (FIG.6.18).

```
Maximum possible array:      1154 MB (1.210e+09 bytes) *
Memory available for all arrays:      1154 MB (1.210e+09 bytes) *
Memory used by MATLAB:      1441 MB (1.511e+09 bytes)
Physical Memory (RAM):      4001 MB (4.195e+09 bytes)
```

Figure 6.18: **Memory** command.

One can see that the maximum memory available for arrays is linked to the Physical Memory (RAM) of the computer. Approximately one quarter can be allowed to an array.

The TABLE 6.1 lists the maximum number of zones that the simulated photon sieves can reach without overflowing the memory. These numbers of zones depend on the value of the RAM and of **Speed**.

	Speed=1	**Speed**=2	**Speed**=3
RAM=4Gb	2500	240	110
RAM=16Gb	5600	580	220
RAM=64Gb	11000	1150	450

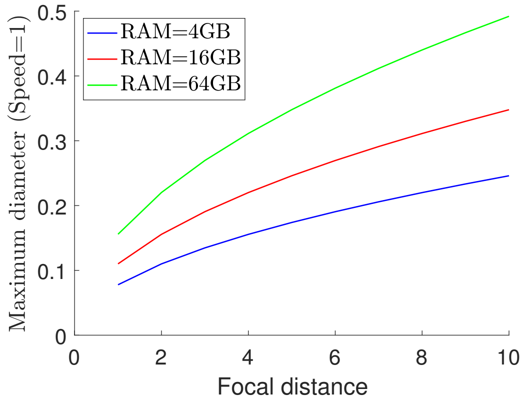
Table 6.1: Maximum numbers of zones possible for a simulated *photon sieve* with respect to the RAM and **Speed**

In FIG.6.19(a), 6.19(b) and 6.19(c) the maximum diameters of the *photons sieves* reached without overflowing the memory are illustrated.

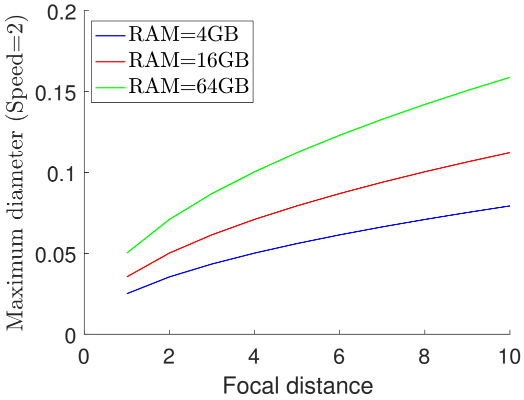
To make the conversion between the maximum numbers of zones and the maximum diameters, the following formula, presented in Chapter 3, is used

$$d_n = 2\sqrt{n\lambda(f + \frac{n\lambda}{4})} \tag{6.57}$$

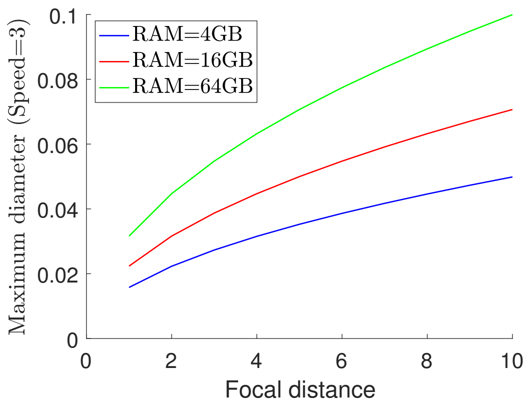
With d_n the diameter of the n^{th} zone and λ taken equal to 550.



(a) **Speed**=1.



(b) **Speed**=2.



(c) **Speed**=3.

Figure 6.19: Maximum diameter possible for the simulated *photon sieves* with respect to the focal distance. (λ=550nm)

• Time

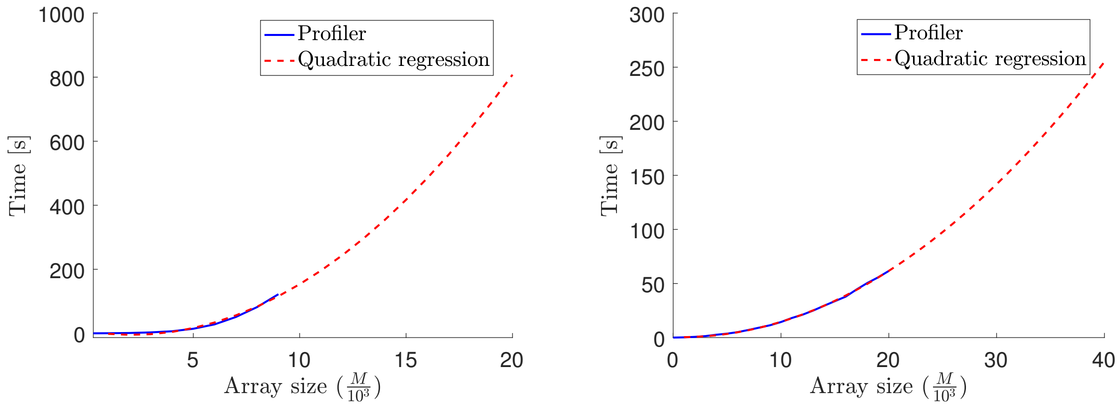
The use of high sized arrays has a significant cost of time. The command **Profiler** allows an analysis of the time consumed by the propagator. Most of the time is spent to perform the *Fast Fourier Transform*(FFT).

The FFT is an algorithm of quadratic complexity. It can be written as

$$Time_{FFT}(M) \rightarrow O(M^2) \tag{6.58}$$

With *M*, the size of the squared array.
Quadratic regressions of the simulation time are represented in FIG. 6.20(a) and 6.20(b) which are respectively the estimation for a home computer and for a professional computer as can be

found in the *CSL*.



(a) Home computer. (b) Professional computer.

Figure 6.20: Time estimation

The time difference is great. For an array of size 10^4x 10^4, the simulation done with the home computer lasts close to 3 minutes while for the professional computer, the simulation is done in 15 seconds.

6.5 Verifications

To verify that the propagator is well implemented, trials for which the results are known from the reality have been performed. They are presented in this section.

6.5.1 Circle

• Variation of the observation plan

The diffraction pattern observed with a circular aperture depends on the distance of the observation plane. When it is close to the aperture, the diffraction effects are not pronounced and the pattern in the observation plane is close to the normal shadow of the circle. On the contrary, when the observation plane is far, the effect of diffraction are present and the pattern is getting close to the *Airy pattern*.

The trials have been proceeded with a circular aperture of 1mm of diameter and a wavelength of 550nm. The FIG.6.21 shows the *irradiance* patterns in the observation plane which distance takes the values: 0.05m, 0.1m, 0.5m, 1m and 5m (the distance of the observation plane, z, is indicated above the figures). FIG.6.22 shows the *irradiance* profiles for observation planes at similar distances.

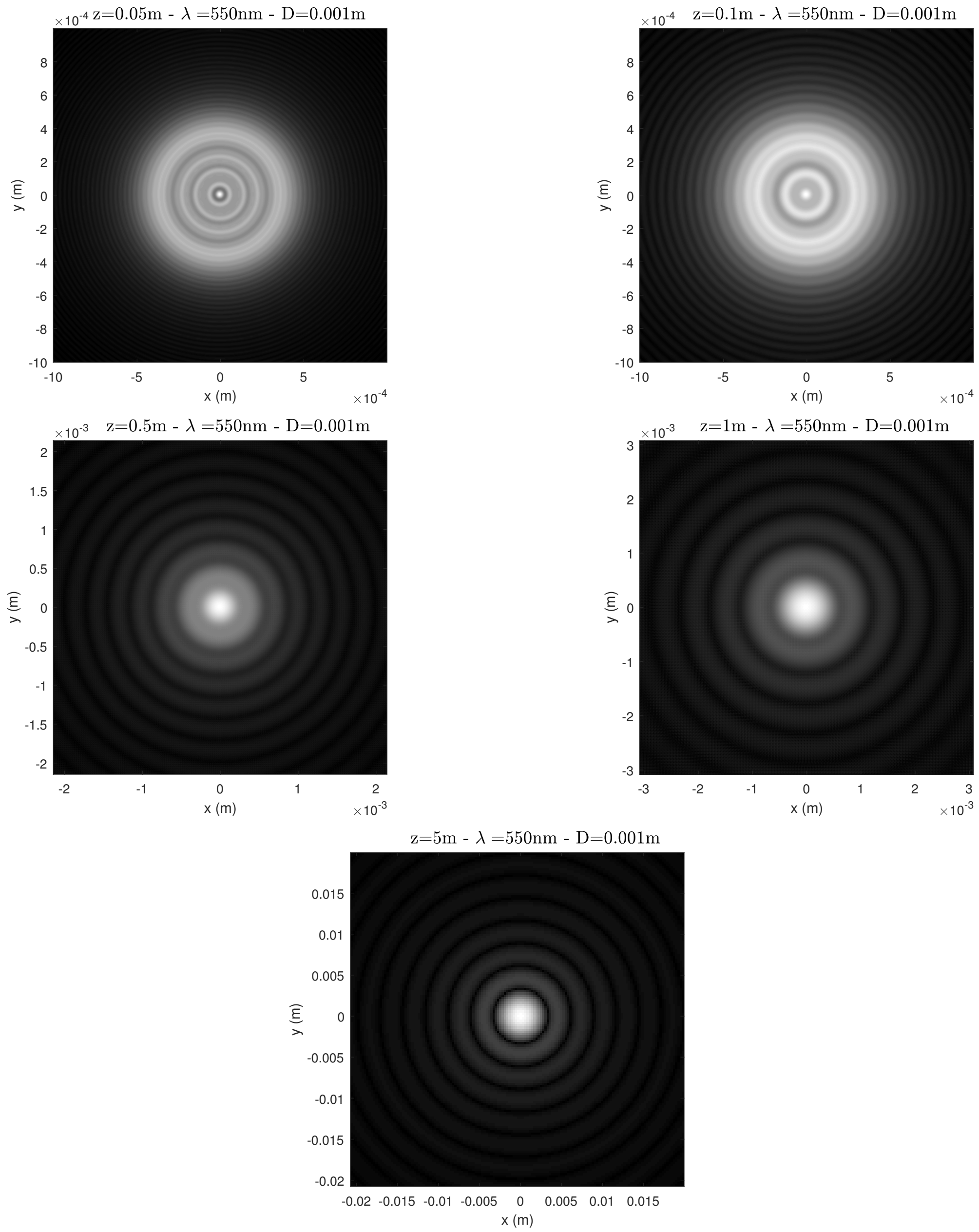


Figure 6.21: Pattern of *irradiance* obtained with a circular aperture and different positions of the observation plane.

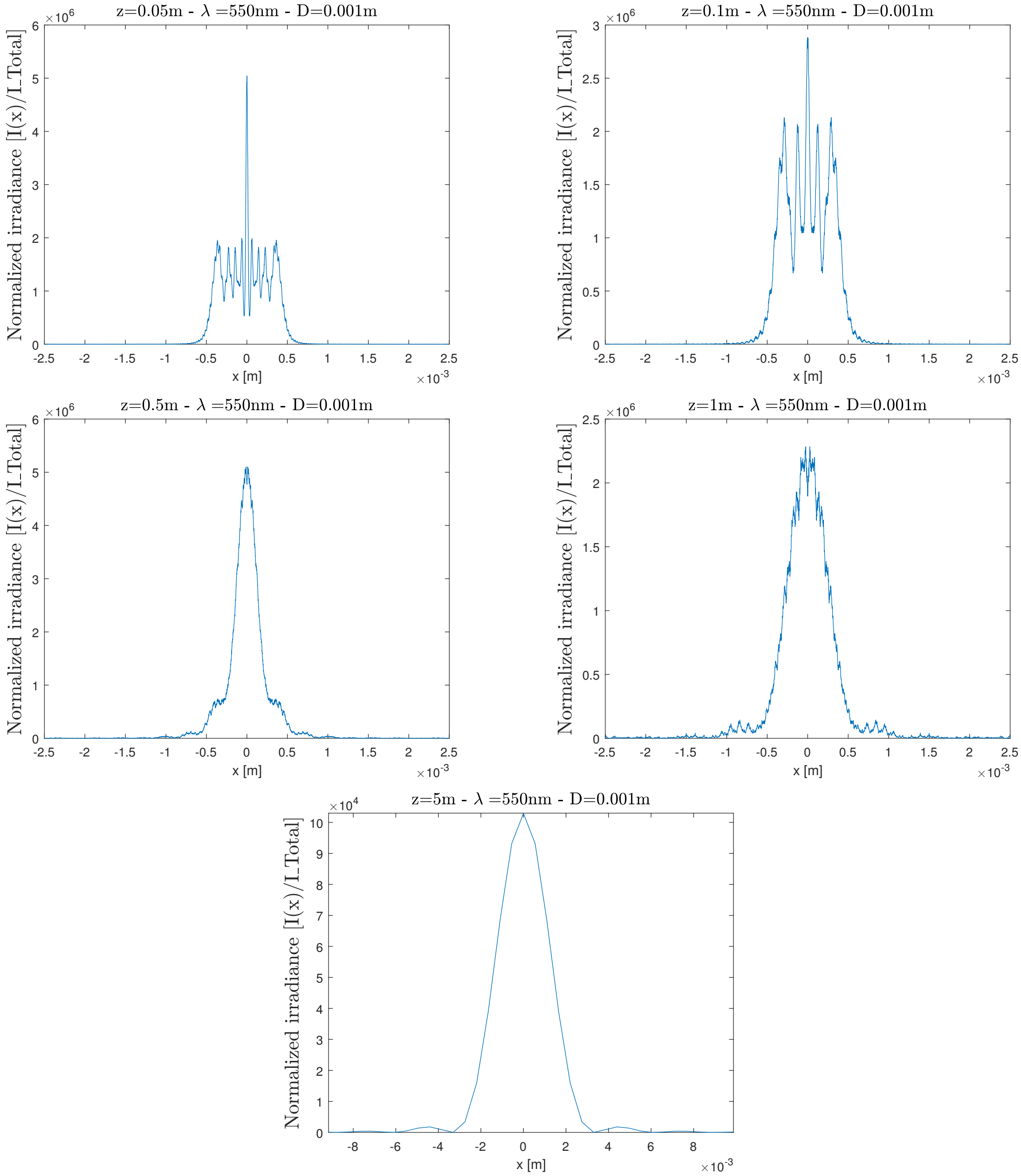


Figure 6.22: Profile of *irradiance* obtained with a circular aperture and different positions of the observation plane.

As expected, the pattern at 0.05m is close to the normal shadow of the circular aperture, and the one at 5m is close to the Airy pattern. The transition between these two states is clearly visible on the figures relative to the *irradiance* profile.

• Encircled energy

To verify the good implementation of the file *Encircled_Energy.m*, used to compute the encircled energy; a trial is performed with a circular aperture and a far observation plane.

Indeed, in the case where the observation plane is sufficiently far, the pattern of *irradiance* is close to the *Airy pattern*. Furthermore, the analytical expression of the encircled energy of the *Airy disk* is known and represented in FIG.6.23

The diameter of the simulated circle is equal to 0.001m and the distance of the observation plane is equal to 10m. The encircled energy computed is represented in FIG.6.24. One can see that the analytical curve and the one obtained with the propagator are perfectly similar.

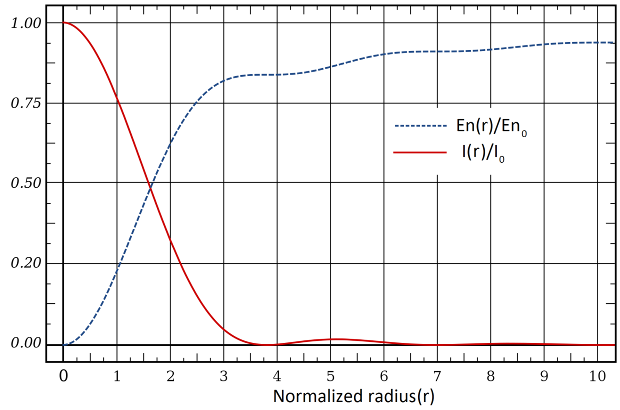


Figure 6.23: Theoretical encircled energy of the *Airy disk*. [4]

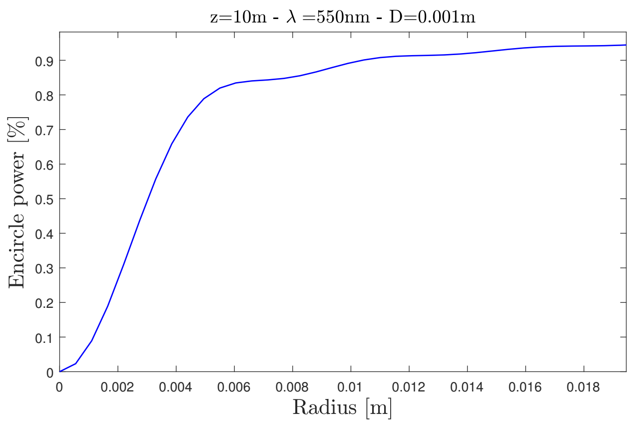


Figure 6.24: Encircled energy obtained with the propagator.

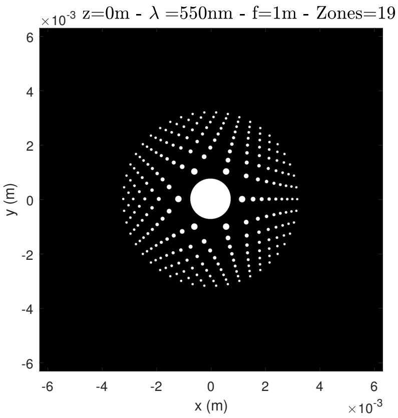
6.5.2 Photon sieve

• Variation of the density

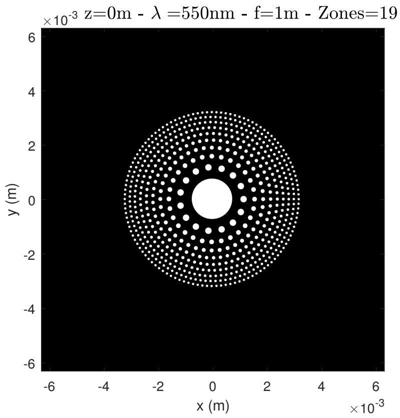
Photon sieves with different densities have been generated to verify that the shapes are visually correct. The definition of the density is the filling percentage of a zone by the holes and is given by

$$\text{Density} = \frac{\text{Diameter of the holes}}{\text{Diameter of the holes + Interstice between holes}} \tag{6.59}$$

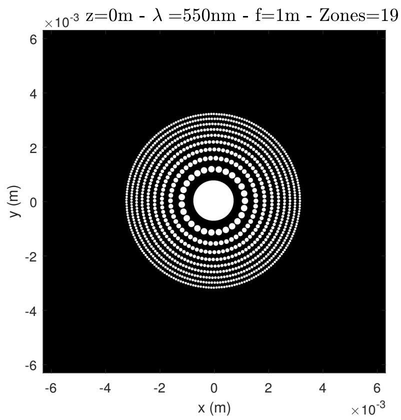
The FIG.6.25(a), 6.25(b) and 6.25(c) display respectively a *photon sieve* of 50 zones with a density equal to 20%, 50% and 90%.



(a) Photon sieve with a density of 20%.



(b) Photon sieve with a density of 50%.



(c) Photon sieve with a density of 90%.

Figure 6.25: Generated photon sieves with different densities.

The Figures show well shaped *photon sieves*.

• Variation of the observation plane

The parameter used to analyze the quality of the focus of an optical system is the radius of the first peak of *irradiance* in the observation plane. When the system is ideal and then, *diffracted-limited*, the radius of the first peak of *irradiance* is given by

$$r_{1^{st}peak} = 1.22\frac{z\lambda}{D} \tag{6.60}$$

With z, the distance of the observation plane; D, the diameter of the optical system and λ, the wavelength ot the light.

This is actually the radius of the *Airy disk* obtained with similar parameters.

The shape of a *photon sieve* depends on the focal distance. Therefore, to reach the best focus, the observation plane has to be positioned at a distance equal to the focal distance.

When the observation plane is placed slightly closer or slightly further than the focal distance the quality of the focus decreases. The FIG.6.26 illustrates well this statement. One can see that a slight change of the position of the observation plane, with respect to the focal distance, increases greatly the radius of the first peak of *irradiance*. The propagator describes correctly the behavior of focalisation of the *photon sieve*.

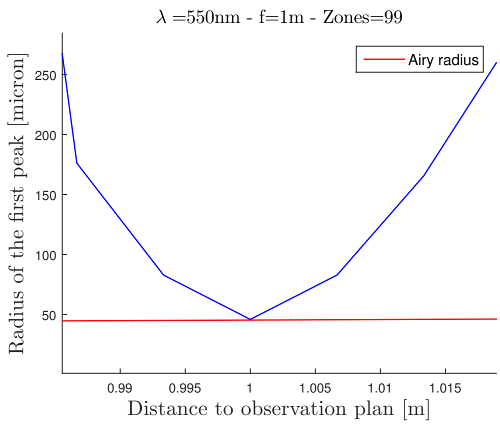


Figure 6.26: Radius of the first peak of *irradiance* when the distance of the observation plane varies.

7 Analyze and sizing of the photon sieves

7.1 Reminders of the specifications

This Chapter is dedicated to the analyze and sizing of the *photon sieve* thanks to the propagator. To recall, the target of the interferometer is to study the exoplanets. Therefore, it has been decided to make the analysis in the visible light, between 400nm and 700nm.

To begin, a global analysis of the pattern of the *photon sieve* is done. Among the parameters of the pattern, the size of the holes, which has to respect the limit of the manufacturing, is analyzed.

Then, the *photon sieve* is studied in order to be injected in an *optical fiber*. The PSF is used to this end. Indeed, the diameter of the first peak of *irradiance* has to fit with the core of the *optical fiber* whose measure is between 5 and 10 microns. This will give a criteria for the size of the *photon sieve*.

Regarding the energy and its distribution in the observation plane, the *surface ratio*, which is the ratio between the open surface and the whole surface of the *photon sieve*, and the *encircled energy* are used.

Some simulations have also been performed in order to characterize the chromatic dispersion of the *photon sieve*. Finally, some alternative patterns for the *photon sieve* are tested in order to increase its performances.

During this Chapter, the parameters of the simulated *photon sieves* and the distance with the observation plan are indicated in the titles of the figures. When it is not specified, the value of the density is fixed at 90%, which is a realistic percentage. Note that higher density could weaken the *photon sieve*.

7.2 Analysis and sizing in visible

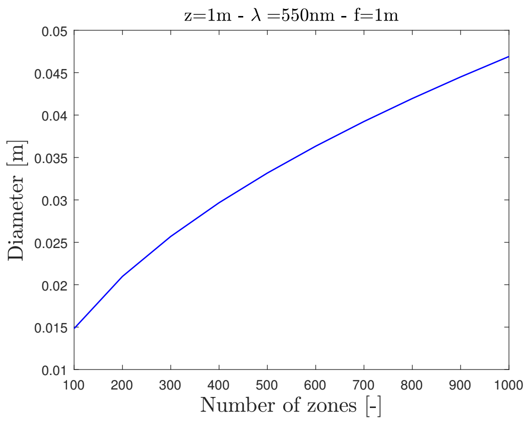
7.2.1 Pattern parameters

The FIG.7.1 shows the evolution of different parameters for a *photon sieve* with a wavelength of 550nm and a focal distance of 1 meter.

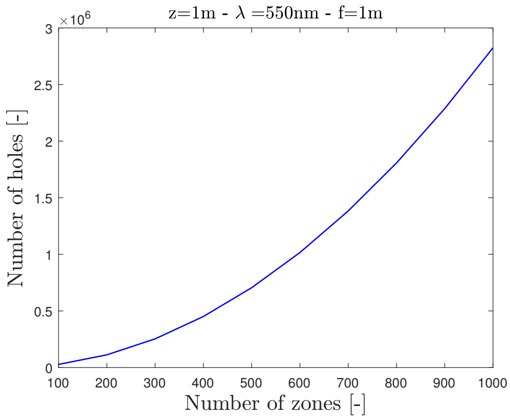
The FIG.7.1(a) illustrates the variation of the diameter with respect to the number of zones. The number of zones varies between 100 and 1000. One can see that 1000 zones are only corresponding to a diameter of 50 mm, which is quite small with respect to our goals. However, simulations can not be performed with higher number of zones without exceeding the limitations of the computers. In this Chapter, it has been decided to work with the zones instead of the diameter which is more convenient.

Even if the simulations corresponds to diameters of small sizes, the tendencies of the *photon sieve* can be extrapolated for higher diameters.

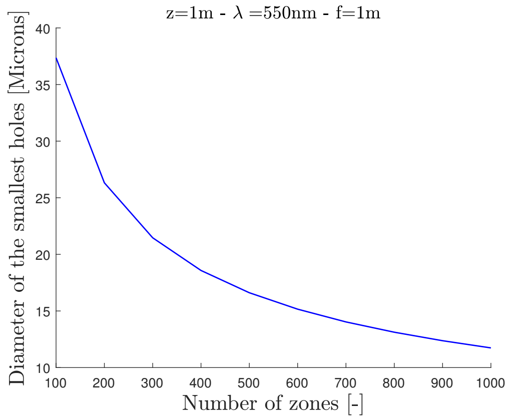
FIG.7.1(b) represents the number of holes with respect to the number of zones. The number of holes quadratically increases with the zones, the million of holes is achieved with 400 zones. It can then be easily imaginable to reach the billion of holes with a *photon sieve* with a diameter of several decimeters. Regarding their sizes, the diameter of the smallest holes are represented in FIG.7.1(c). For a *photon sieve* with 1000 zones, the smallest holes are close to 10 microns. The trends show a continuous an substantial decrease. The size of the smallest holes is an important point to consider knowing that the manufacturability limits are close to 1 micron nowadays. [8]



(a) Variation of the diameter with respect to the number of zones. 1000 zones correspond to a diameter of 50 mm.



(b) Number of holes with respect to the number of zones. It quadratically increases.



(c) Diameter of the smallest holes. With 1000 zones, the diameter is almost equal to 10 microns.

Figure 7.1: Pattern parameters for a *photon sieve* with a focal length of 1m and λ=550nm.

7.2.2 Resolution in observation plan

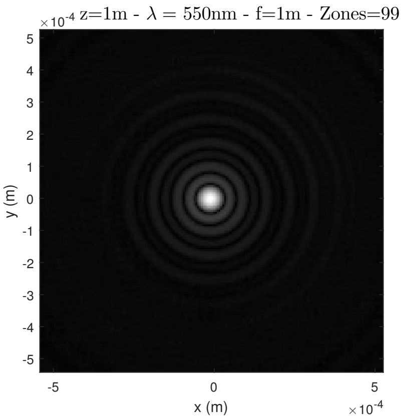
• PSF

As said previously, in order to work with *optical fiber*, it is necessary that the main peak of the PSF fits with the core of the fiber which diameter is between 5 and 10 micron.

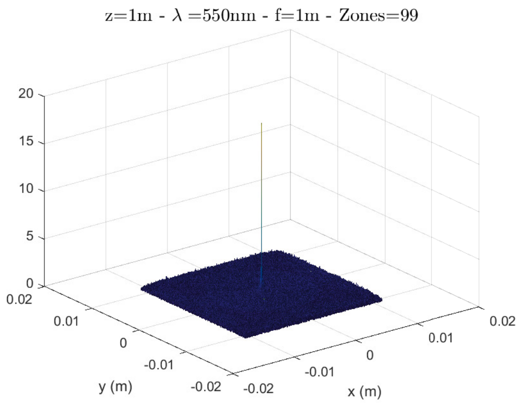
As a reminder, the PSF is the image in the observation plane of a point source. The propagator simulates light as a plane wavefront, which is related to a far point object, such as a star. The FIG.7.2 shows a typical PSF obtained with a *photon sieve*. It has been simulated with a *photon sieve* of 100 zones, a focal distance of 1m and a wavelength of 550nm.

The FIG.7.2(a) shows the 2D representation of the PSF in the observation plan; the 3D representation is standing in FIG.7.2(b) and the FIG.7.2(c) is a profile representation in the observation plane with $y = 0$.

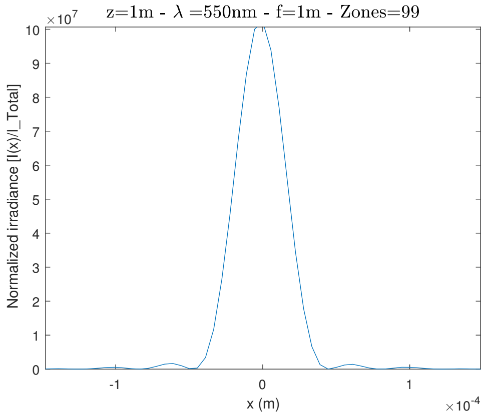
The 2D representation and the profile representation are similar to the ones of the *Airy disk* seen in Chapter 3. This is characteristic of a *diffracted-limited* system.



(a) 2D representation in the observation plane. It looks like the *Airy disk*.



(b) 3D representation in the observation plane.



(c) Profile representation. It is similar to the *Airy disk*.

Figure 7.2: PSF obtained with a *photon sieve* with a focal length of 1m, 100 zones and λ=550nm.

• Focal distance

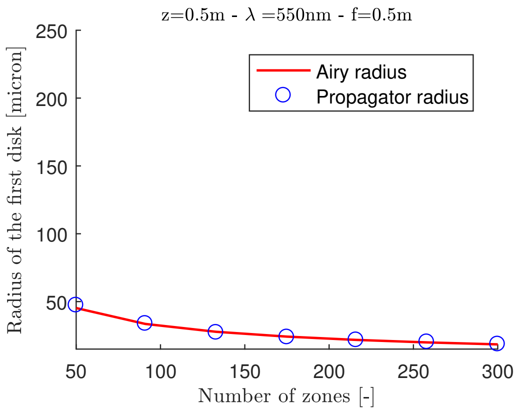
The radius of the first peak of the PSF (first value of the radius where the PSF is equal to 0) is the parameter to analyze in order to use *optical fiber*. This radius has to correspond to a diameter between 5 and 10 micron.

To study the impact of the focal distance on the radius of the first peak of the PSF, some simulations have been performed with *photon sieves* of focal length from 0.5m to 10m and a wavelength equal to 550nm. The simulations also study the impact of the increase of the number of zones on the focus. To this end, the number of zones varies between 50 to 300. The FIG.7.3(a)-(e) show the results obtained.

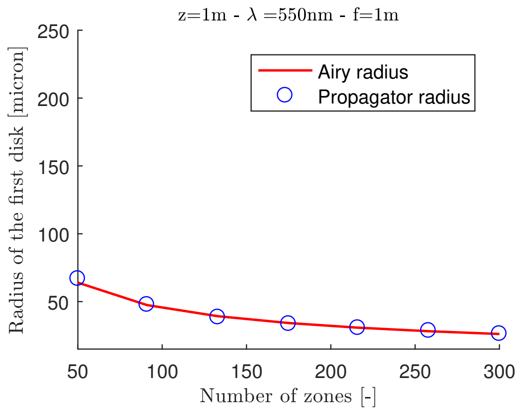
The red curve represents the radius of the *Airy disk* relative to similar parameters (diameter, wavelength and focal distance). The blue circles are the results obtained with the simulated *photon sives*. One can see that those circles, are really close to the red curve. With every focal distances, at 50 zones, there is a small difference with the *Airy radius*. However, at 300 zones, this difference is almost nonexistent and the circles are located on the red curve.

These observations mean that the *photon sieve* reaches rapidly the performances of a *diffracted-limited* system.

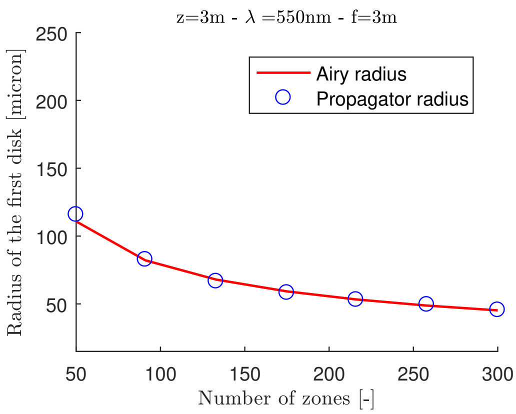
One can also note that, for a constant number of zones, a smaller radius is achieved with a smaller focal distance. The focal distance has to be taken into account for our purposes.



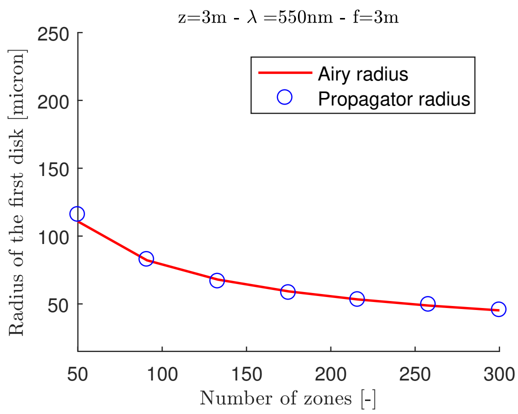
(a) Focal length of 0.5m.



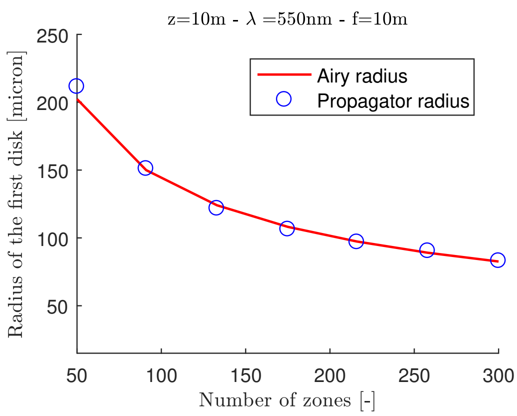
(b) Focal length of 1m.



(c) Focal length of 3m.



(d) Focal length of 5m.



(e) Focal length of 10m.

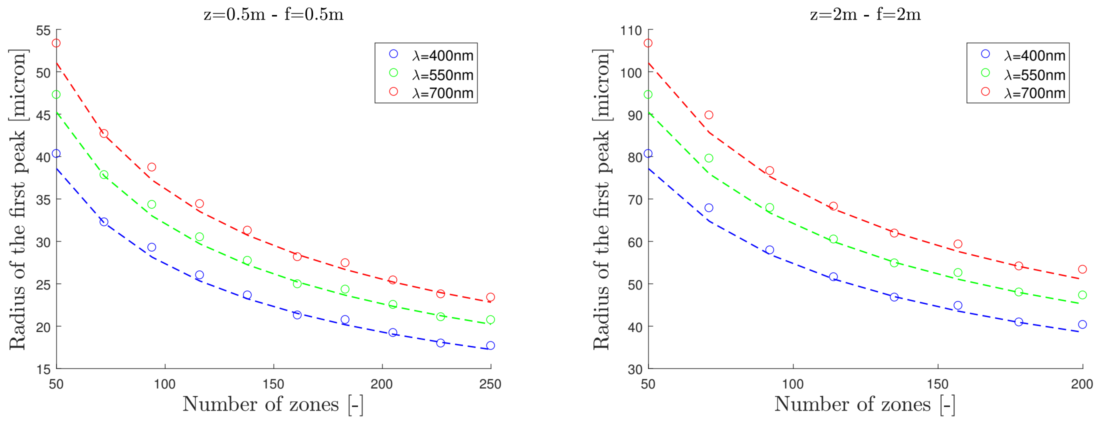
Figure 7.3: Radius of the first peak of the PSF with different focal lengths and number of zones.

- **Wavelength**

A similar analysis has been done with the wavelength. The results are shown in FIG.7.4.

To cover the whole visible spectrum, the simulations are performed with wavelengths of 400nn, 550nm and 700nm. The dotted lines represent the corresponding *Airy radius*. The results are presented in FIG.7.3(a) and (b). The same conclusions than for the focal distance analysis can be drawn. The radius obtained with the simulated *photon sieves* are close to the theoretical *Airy radius*, which is characteristic of a *diffracted-limited* optical system.

One can also note that when the wavelength decreases, while the number of zones is kept constant, the radius of the first peak decreases.



(a) Focal length of 0.5m.

(b) Focal length of 2m.

Figure 7.4: Radius of the first peak of the PSF with different wavelengths and number of zones. The dotted lines correspond to the theoretical *Airy radius*.

7.2.3 Radiometry purposes

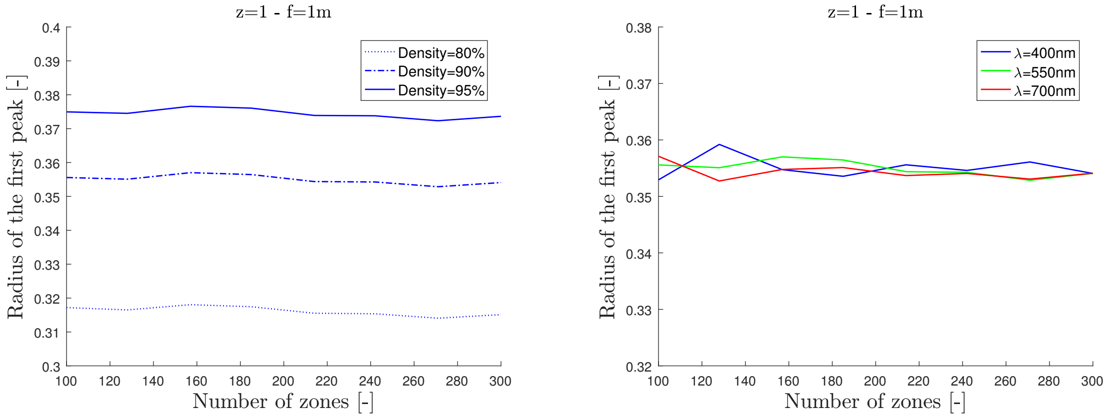
This section is concerning the analysis of the energy transmitted through the *photon sieve* and its distribution in the observation plane.

• Surface ratio

At first, the surface ratio is analyzed. The bigger it is, the higher the flux of light through the *photon sieve* will be. This ratio is supposed to be slightly inferior to 0.5 Indeed, the holes fulfill partially one zone out of two.

The FIG.7.5(a) shows the variation of the surface ratio for different values of the density while the number of zones is increased. One can see that the number of zones does not impact as much the surface ratio. However, the surface ratio is related to the filling density and is approximately equal to 0.355 for a density of 90%.

In FIG.7.5(b), the surface ratio is shown for *photon sieves* from 400nm to 700nm with a density of 90%. The surface ratios are slightly varying but the values stay close to 0.355. This observation means that it is not influenced by the wavelength.



(a) Variation of the density. (b) Variation of the wavelength.

Figure 7.5: Surface ratio of the *photon sieve*.

• Encircled energy

To recall, the encircled energy is computed thanks to a circular integration performed on the profile of *irradiance*. This integration is made by a function in *Encircled_Energy.m*.

The encircled energy of a *photon sieve* of 100 zones with a wavelength of 550nm and a focal length of 1m is illustrated in FIG.7.6.

The jump due to the focus is clearly visible. However, oddities about the curve have to be noticed. Indeed, the percentage does not tend asymptotically to the value of 100%. The curve even exceeds 100%, which is impossible knowing the principle of conservation of energy.

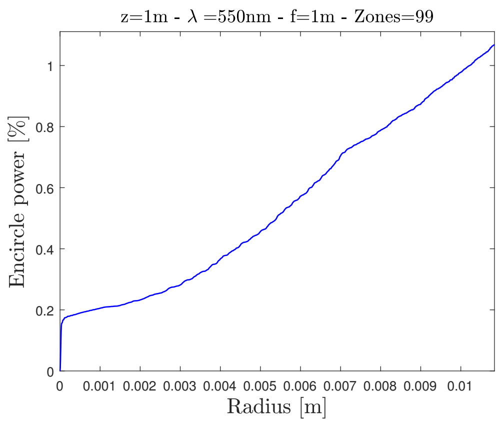


Figure 7.6: Encircled energy of a *photon sieve*.

This is actually a consequence of the periodic extension introduced by the DFT and explained in Chapter 4. To verify this statement, simulations have been made with the value of **Speed** equal to 2 and equal to 3; and therefore, with different values of the sample interval, Δx. The results are shown in textscFig.7.7 for **Speed**=2 and in FIG.7.8 for **Speed**=3.

The zooms in the upper parts (FIG.7.7(b) and FIG.7.8(b)) of the encircled energy show that the excess decreases while the value of Δx decreases. Contrariwise, the zooms in the lower parts (FIG.7.7(c) and FIG.7.8(c)) show that the values close to the center are unchanged when Δx decreases. These observations are in agreement with the periodic extension artifact which has a significant impact on the value of the extremities only. As the extremities are corresponding to the value of the biggest radius, the impact is consequent for the encircled energy.

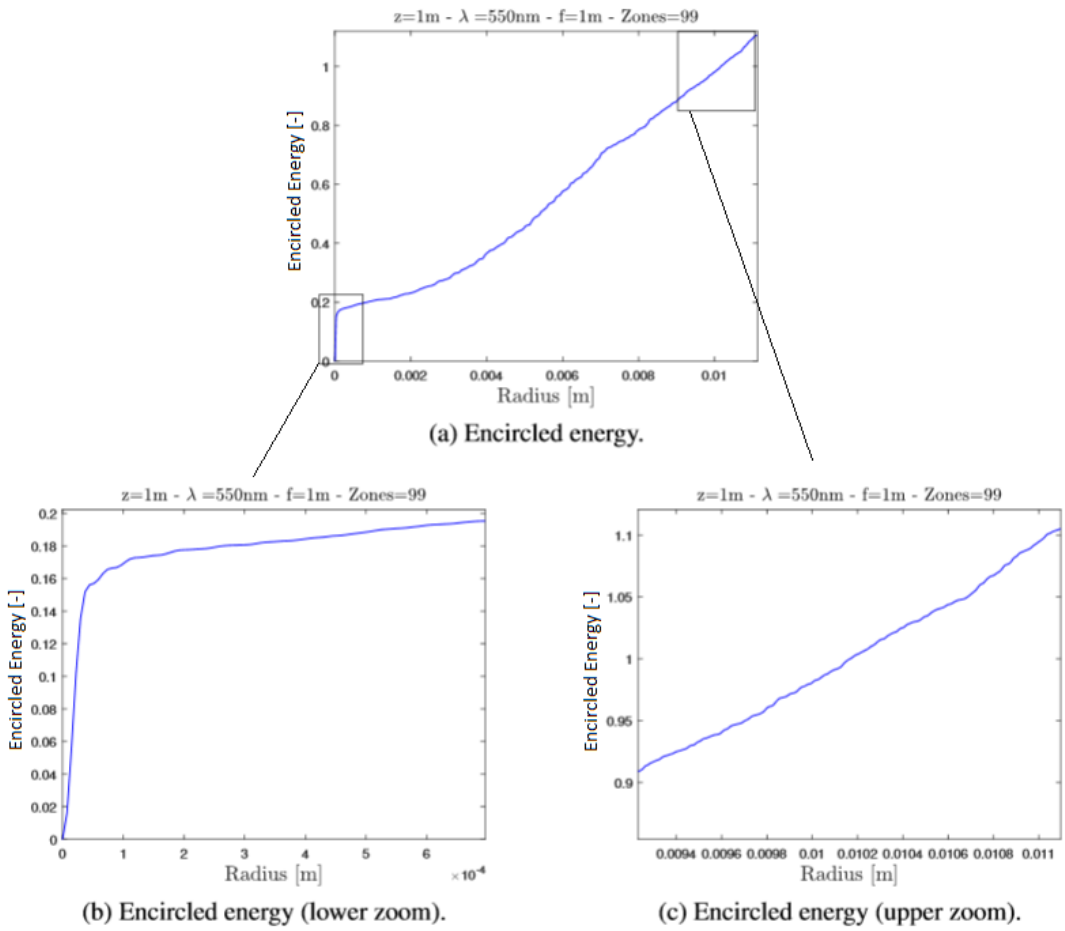


Figure 7.7: Encircled energy (**Speed**=2).

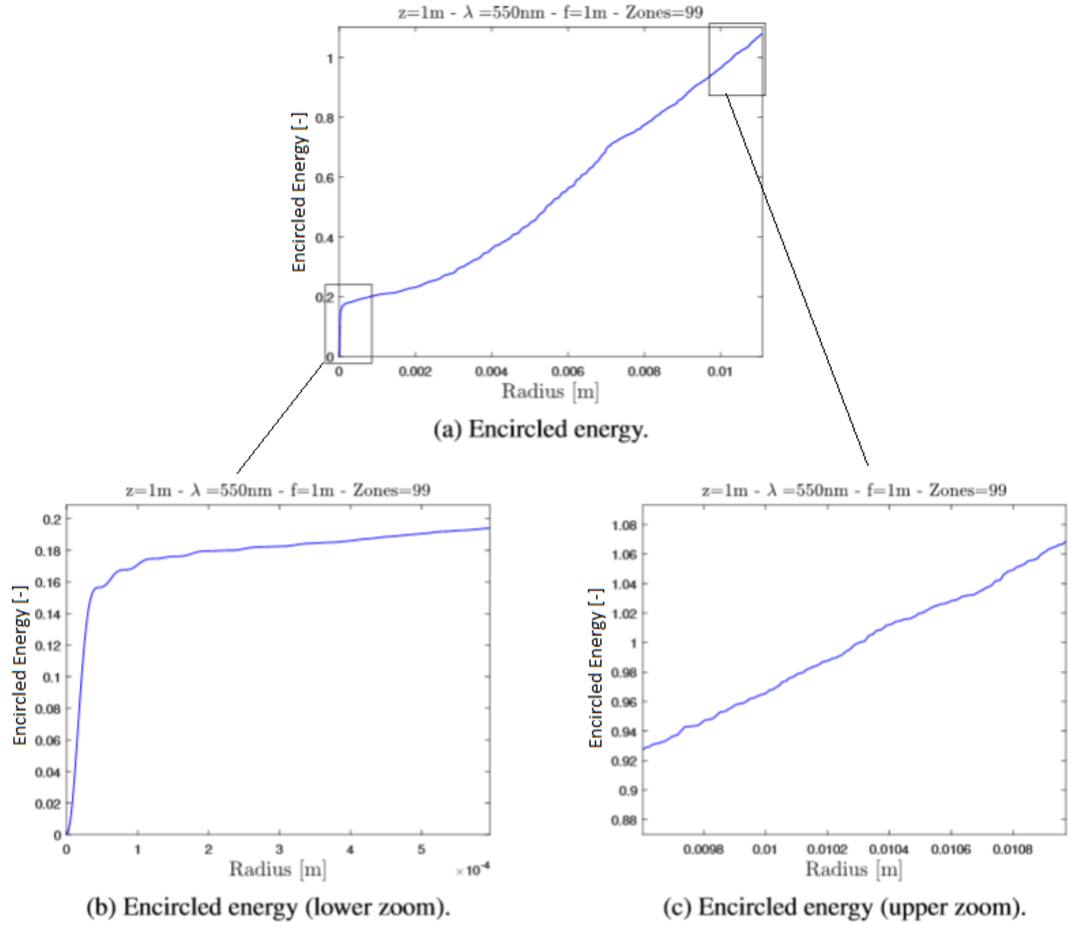


Figure 7.8: Encircled energy (**Speed=3**).

The curve of the encircled energy is then reliable for small radius and becomes unreliable for high radius with respect to the focal center.

• **Energy contained in the first peak**

The energy contained in the first peak is an important parameter; it must be as high as possible. The FIG.7.9 shows its value with different numbers of zones and different wavelengths (400nm, 550nm and 700nm).

The curves slightly oscillate but the three of them keep a value approximately equal to 14.7%. These curves are obtained with a *photon sieve* of focal distance equal to 1m. Some simulations, using different focal length, have shown similar results.

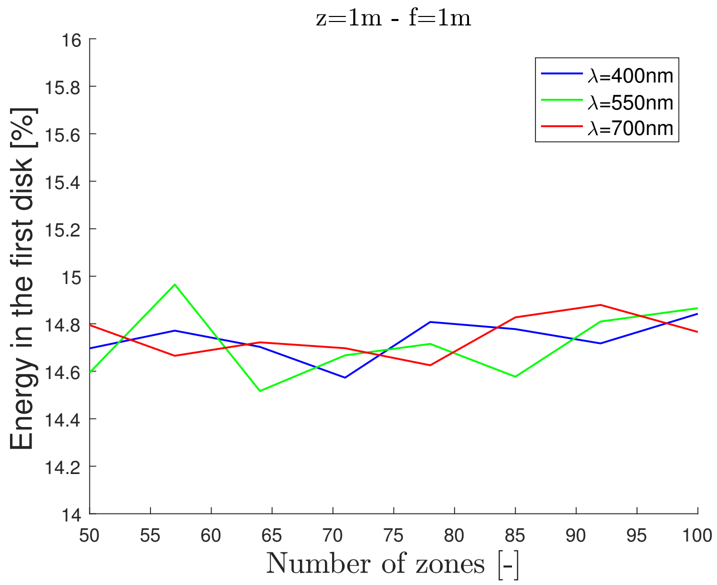


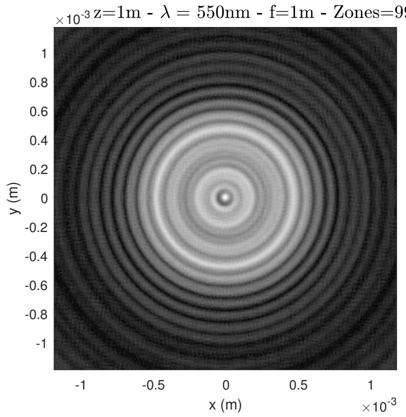
Figure 7.9: Energy contains in the first peak for a *photon sieve*.

7.2.4 Bandwidth

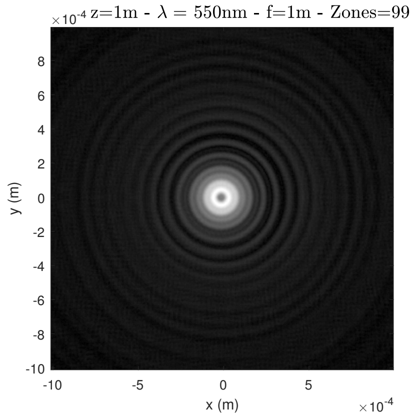
Some simulations have been performed to visualize the chromatic dispersion. The *photon sieve* studied contains 100 zones, a wavelength of 550nm and a focal distance of 1m.

The focusing is analyzed with light of 500nm, 540nm, 550nm, 560nm and 600nm. The *irradiance* in the observation plane are respectively standing in FIG.7.10(a), FIG.7.10(b), FIG.7.10(c), FIG.7.10(d) and FIG.7.10(e).

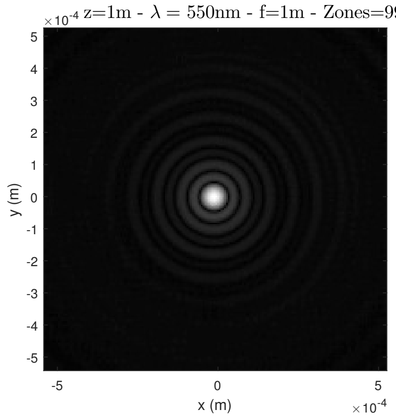
The patterns for 540nm and 560nm are similar and show a dispersed focal spot. The *irradiance* pattern obtained with the wavelength of 500nm and 600nm are also similar. In these cases, the focusing is almost nonexistent. Similar conclusions are made for their profiles, which are represented in the FIG.7.11.



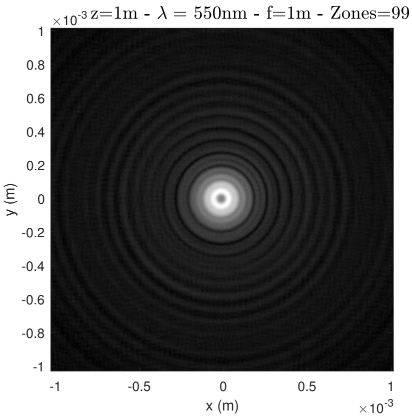
(a) $\lambda = 500nm$.



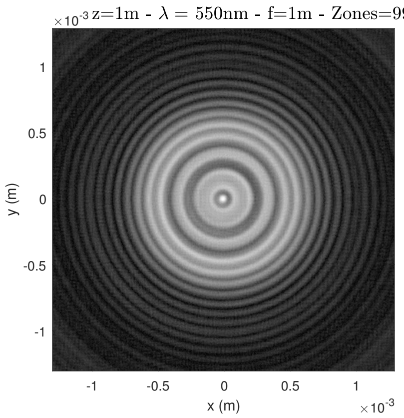
(b) $\lambda = 540nm$.



(c) $\lambda = 550nm$.

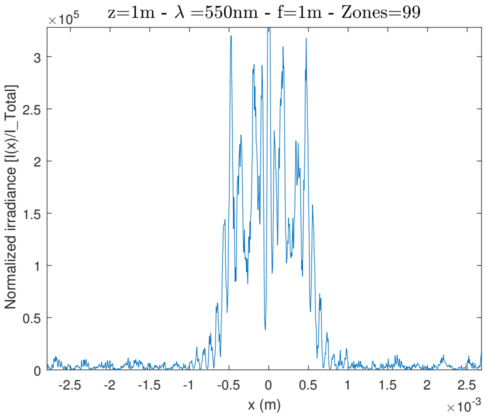


(d) $\lambda = 560nm$.

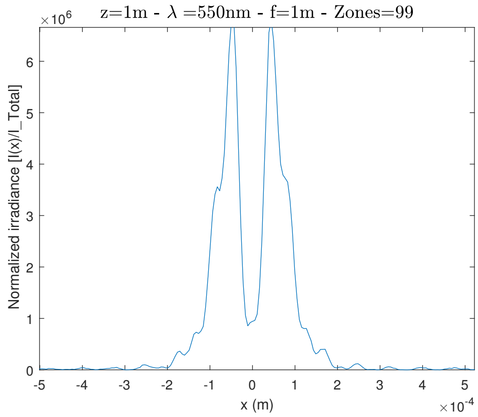


(e) $\lambda = 600nm$.

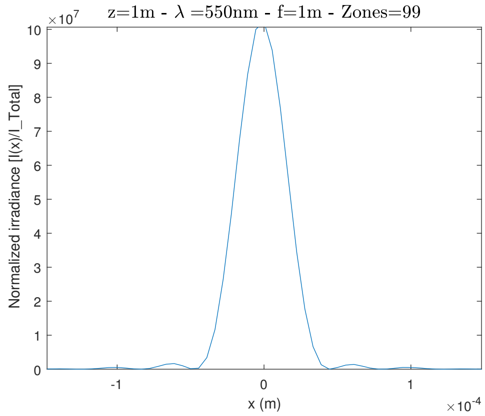
Figure 7.10: Observation of the chromatic dispersion in the observation plane for a *photon sieve* made with a wavelength of 550nm.



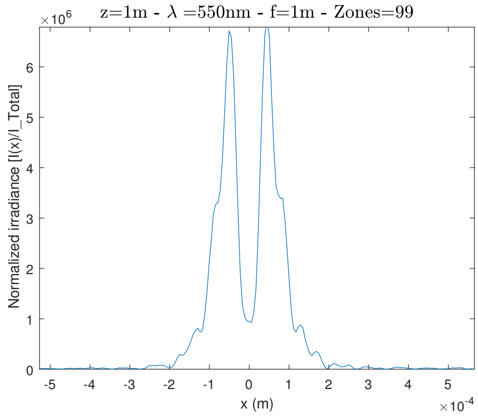
(a) $\lambda = 500nm$.



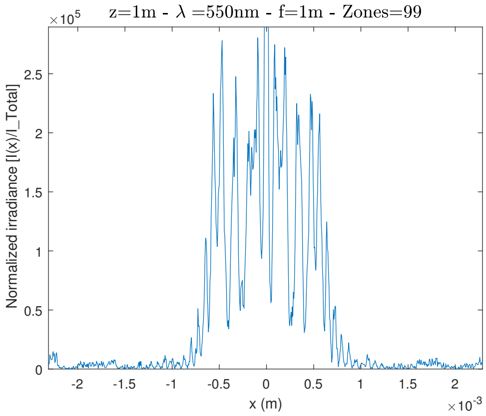
(b) $\lambda = 540nm$.



(c) $\lambda = 550nm$.



(d) $\lambda = 560nm$.



(e) $\lambda = 600nm$.

Figure 7.11: Observation of the chromatic dispersion with the profile of *intensity* for a *photon sieve* made with a wavelength of 550nm.

The FIG.7.12 shows the normalized radius of the spot obtained with respect to the wavelength. For wavelengths equal to 520nm and 580nm, the radius is already 10 times higher than for the spot at 550nm. This finding shows that the *photon sieve* is subject to a high chromatic dispersion.

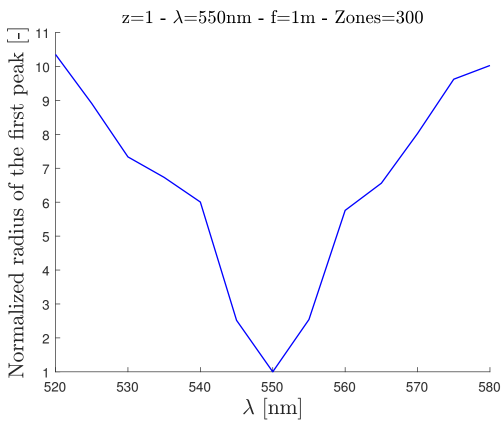


Figure 7.12: Radius of the first peak with respect to λ (*photon sieve* with a wavelength of 550nm).

7.2.5 Alternative patterns

The pattern of a *photon sieve* is made by filling the zones of a *Fresnel zone plate* with holes. Some different patterns have been tested in order to improve the performances of the *photon sieve*.

• Multilayer holes pattern

At first, the multilayer holes pattern have been tested. This pattern is shown in FIG.7.13. Instead of filling the zone with an unique layer of holes, the zones are filled with two or more layers of smaller holes.

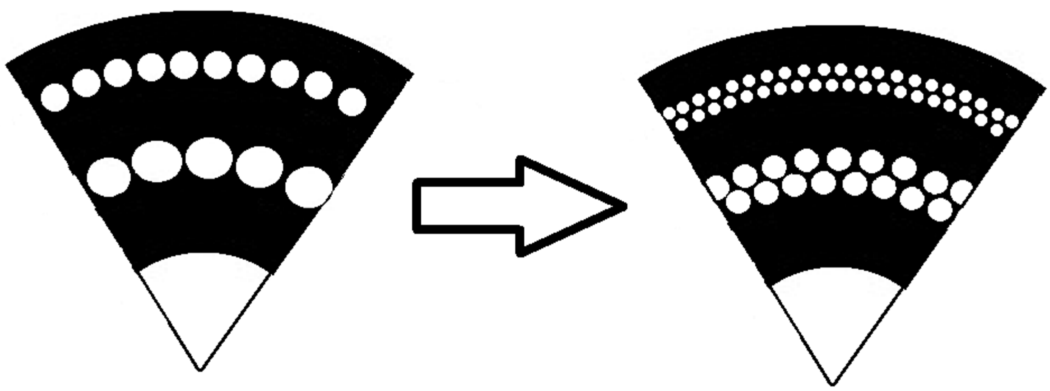
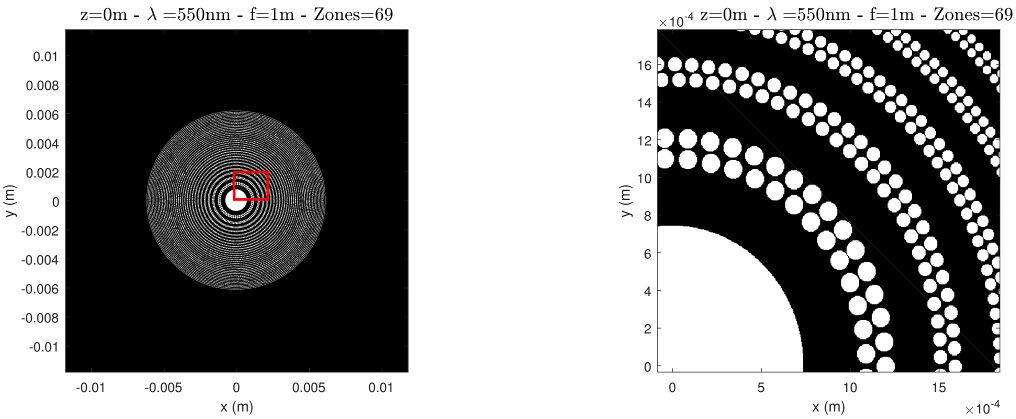


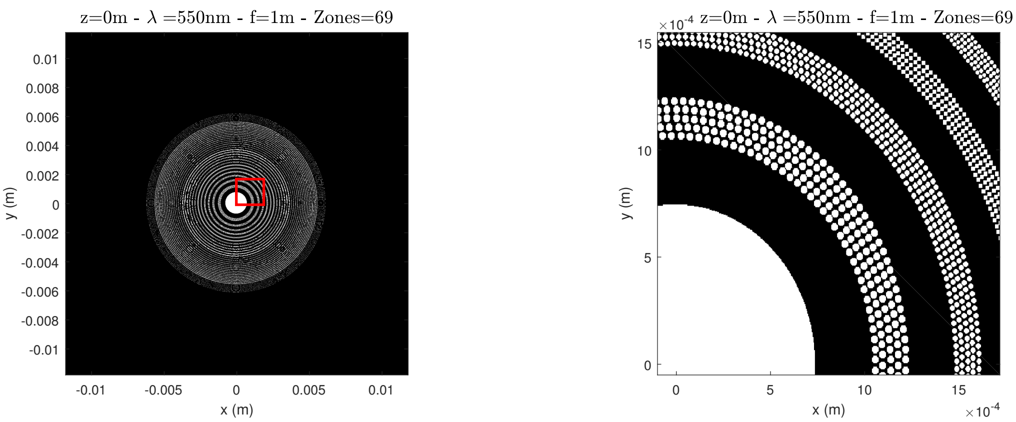
Figure 7.13: Multilayer holes pattern (Layer=2).

The *Matlab* file used to implement this pattern is *Photon_Sieve_Generator_ML.m* and can be found in Appendix B. The correct implementation is shown in FIG.7.14 and FIG.7.15 which are *photon sieves* obtained for respectively 2 and 5 layers.



(a) Photon sieve generated. (b) Zoom.

Figure 7.14: Photon sieve with 2 layers.



(a) Photon sieve generated. (b) Zoom.

Figure 7.15: Photon sieve with 5 layers.

The comparison between *photon sieves* with different numbers of layers has been performed with *photon sieves* of wavelength equal to 550nm, focal length equal to 1m and 100 zones. The density, defined similarly than for the classic *photon sieve*, takes the value of 80% in TAB.7.1, the value of 90% in TAB.7.2 and the value of 95% in TAB.7.3.

Number of Layers [-]	Radius first peak [micron]	Surface ratio [-]	Peak intensity [%]	Holes size [micron]
1	47.27	0.317	14.00	37.36
2	47.27	0.277	10.83	16.25
5	47.27	0.257	11.18	6.03

Table 7.1: Variation of the number of layers (density=80%).

Number of Layers [-]	Radius first peak [micron]	Surface ratio [-]	Peak intensity [%]	Holes size [micron]
1	47.27	0.355	14.34	37.36
2	47.27	0.346	11.49	18.15
5	47.27	0.340	10.49	7.13

Table 7.2: Variation of the number of layers (density=90%).

Number of Layers [-]	Radius first peak [micron]	Surface ratio [-]	Peak intensity [%]	Holes size [micron]
1	47.27	0.374	14.54	37.36
2	47.27	0.383	11.79	19.09
5	52.84	0.388	9.62	7.73

Table 7.3: Variation of the number of layers (density=95%).

The tables show that an increase of the number of layers is actually not interesting. The surface ratio is only slightly increased with a density of 95%. Furthermore, even if the radius of the first peak is unchanged, the percentage of energy contained in the first peak is decreased. Regarding the manufacturing aspect, the reduction of the size of the smallest holes is important enough to cause troubles.

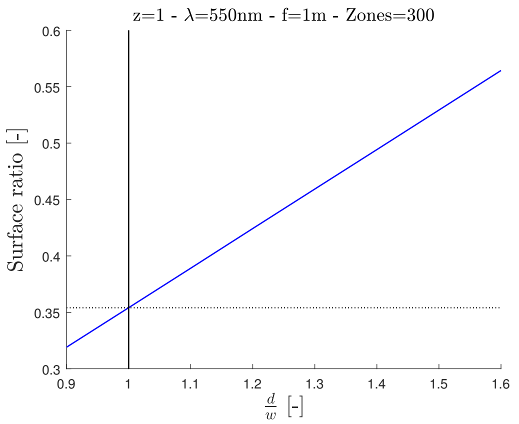
- **Variation of the holes diameter**

According to the literature (articles [8] & [9]), an increase of the diameter of the holes could increase the performances of the *photon sieves*. It could seem paradoxical because when the diameter is increased, the holes are exceeding the constructive zone. The explanation stands in higher order of diffraction than demonstrated in the presentation of the operational process of the *Fresnel zone plates* in Chapter 3.

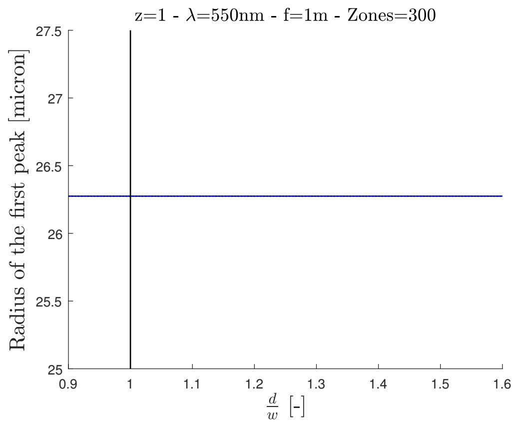
The FIG.7.16 shows the evolution of different parameters with respect to the ratio $\frac{d}{w}$ with d, the diameter of the holes and w, the width of the corresponding zone. For the classic *photon sieves*, this ratio is equal to 1 obviously. The analysis is made for values of $\frac{d}{w}$ between 0.9 and 1.6.

The FIG.7.16(b) shows that the radius of the first peak is not impacted when the value of the ratio varies. The main positive effect is the increase of the surface ratio while the ratio $\frac{d}{w}$ is increased. (FIG.7.16(a)). The negative effect is the decrease of the energy contained in the first peak which decreases when the ratio $\frac{d}{w}$ is increased (FIG.7.16(c)).

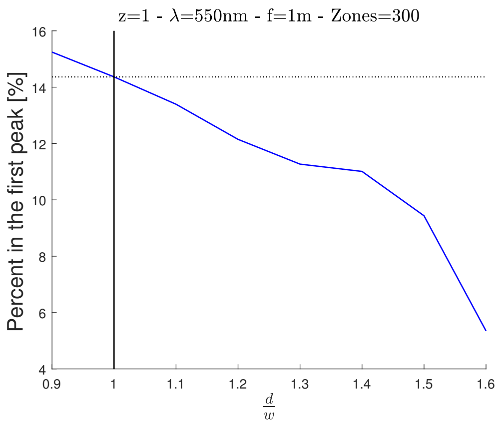
The FIG.7.16(d) shows the normalized energy contained in the first peak by taking into account the increase of the surface ratio and the decrease of the percentage in the peak. One can see that it is increased up to $\frac{d}{w}$=1.4; then, the energy undergoes a high drop.



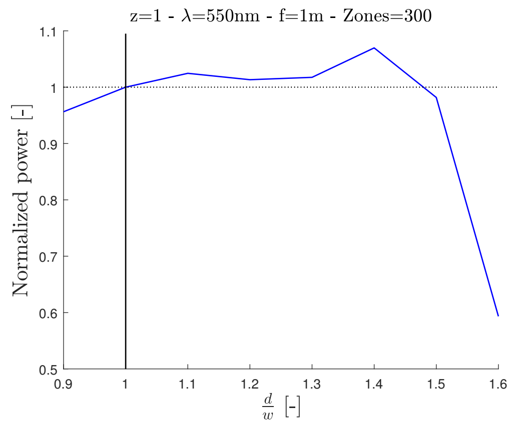
(a) Surface ratio with respect to $\frac{d}{w}$. It increases while $\frac{d}{w}$ increases. It is then a positive impact.



(b) First peak radius with respect to $\frac{d}{w}$. It is not impacted with the variation of $\frac{d}{w}$.



(c) Percentage in the first peak with respect to $\frac{d}{w}$. It decreases while $\frac{d}{w}$ increases. It is then a negative impact.



(d) Normalized power in the first peak with respect to $\frac{d}{w}$. It takes into account the increase of the surface ratio and the decrease of the energy in the first peak. The best result is obtained while $\frac{d}{w}$=1.4.

Figure 7.16: Impact introduces by the variation of the holes diameter ($\frac{d}{w}$) .

The variation of the size of the smallest holes is represented in FIG.7.17. Increasing the ratio $\frac{d}{w}$, is interesting to broaden the limits due to the manufacturing process.

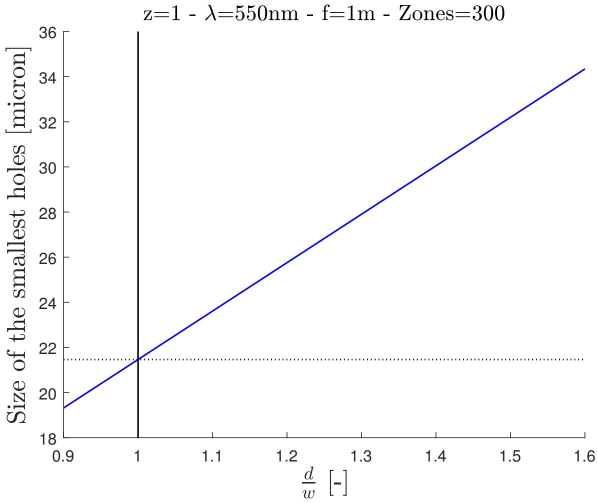


Figure 7.17: Diameter of the smallest holes with respect to $\frac{d}{w}$.

7.2.6 Discussion

Finally, the sizing and performances of the *photon sieves* in order to work with *optical fiber* have been collected in three tables related to the wavelengths: TAB.7.4 for 500nm, TAB.7.5 for 550nm and TAB.7.6 for 700nm.

The results have been computed with a ratio of the holes diameter of 1.4 as seen in the last section. The two values of the diameter correspond to the minimum sizes required in order to be correctly injected in an *optical fiber* with a core diameter from 10 micron to 5 micron. Regarding the values of the surface ratio, the peak intensity and the holes size; the values are kept pretty constant in every cases.

The minimum size of the holes, concerning the case of an *optical fiber* of 5micron, is approximately equal to 3mm, which respects the manufacturing conditions. An interesting fact to note about their constant values, is that the spatial resolution is clearly linked to the size of the smallest holes.

The angular resolution has been computed with respect to the *Rayleigh criterion*.

	λ=400nm					
f [m]	Diameter [m]	Surface ratio [-]	Peak intensity [%]	$\Delta\theta$ [arcsec]	Holes size [micron]	f/# [-]
0.5	[0.048-0.097]	≈ [0.47-0.5]	≈ [11-12]	[2.06-1.03]	[5.74-2.88]	[10.41-5.20]
1	[0.097-0.195]	≈ [0.47-0.5]	≈ [11-12]	[1.03-0.51]	[5.74-2.88]	[10.30-5.15]
2	[0.195-0.39]	≈ [0.47-0.5]	≈ [11-12]	[0.51-0.25]	[5.74-2.88]	[10.25-5.12]
5	[0.488-0.976]	≈ [0.47-0.5]	≈ [11-12]	[0.20-0.10]	[5.74-2.88]	[10.24-5.12]
10	[0.976-1.952]	≈ [0.47-0.5]	≈ [11-12]	[0.10-0.05]	[5.74-2.88]	[10.24-5.12]

Table 7.4: Results for λ=400nm

	λ=550nm					
f [m]	Diameter [m]	Surface ratio [-]	Peak intensity [%]	$\Delta\theta$ [arcsec]	Holes size [micron]	f/# [-]
0.5	[0.067-0.134]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[2.06-1.03]	[5.75-2.89]	[7.46-3.73]
1	[0.134-0.268]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[1.03-0.51]	[5.75-2.89]	[7.46-3.73]
2	[0.268-0.536]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[0.51-0.25]	[5.75-2.89]	[7.46-3.73]
5	[0.671-1.342]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[0.20-0.10]	[5.75-2.89]	[7.45-3.73]
10	[1.342-2.684]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[0.10-0.05]	[5.75-2.89]	[7.45-3.73]

Table 7.5: Results for λ=550nm

	λ=700nm					
f [m]	Diameter [m]	Surface ratio [-]	Peak intensity [%]	$\Delta\theta$ [arcsec]	Holes size [micron]	f/# [-]
0.5	[0.085-0.17]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[2.06-1.03]	[5.75-2.91]	[5.88-2.94]
1	[0.17-0.341]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[1.03-0.51]	[5.75-2.91]	[5.88-2.94]
2	[0.341-0.683]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[0.51-0.25]	[5.75-2.91]	[5.88-2.94]
5	[0.854-1.708]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[0.20-0.10]	[5.75-2.91]	[5.87-2.94]
10	[1.708-3.416]	$\approx [0.47 - 0.5]$	$\approx [11 - 12]$	[0.103-0.051]	[5.758-2.91]	[5.87-2.93]

Table 7.6: Results for λ=700nm

In Chapter 4, it has been shown that the beam of light has to be into the acceptance cone of the *optical fiber* to be injected into it. For an optical system, the criteria applied to respect that condition is related to the f-number (f/#). In the case of *optical fiber* with a core made of silica; the condition to respect is a f/# higher than 1.9 when the cladding is made of polymer and a f/# higher than 2.3 when the cladding is made of silica. The tables show that the f/# of the different *photon sieve* systems respect the two conditions.[10]

Those results are quite general. Indeed, the focal distance is taken between 0.5 to 10m. It is hard to imagine two folded *photon sieves*, with a focal distance of 10m, in a 6U CubeSat. However, a focal distance between 0.5 to 2m seems suitable for the CubeSat.

8 Conclusion and perspectives

In this work the *photon sieves* have been studied in order to be implemented in a deployable 6U CubeSat which finality is the research of exoplanets thanks to the *nulling interferometry*.

To perform the preliminary analysis; a propagator, specific to the *photon sieves* has been implemented in the *Matlab* environment. Indeed, since the *photon sieves* are diffractive lens, their behavior can not be studied thanks to the classical software of optical design (*Code V*, *ASAP*, *Zemax*,..) which use the principles of geometrical optics. Therefore, the propagator, based on the Fourier optics, has been used to analyze the *photon sieves*.

Optical fiber, suitable for the space environment, has been chosen to perform *interferometry* with the CubeSat. The most important requirement for the *photon sieves* was then to respect the conditions of a good injection in an *optical fiber*.

The analysis of the *photon sieves* has shown encouraging results. Indeed, the *photon sieves* offer a wide range of sizes suitable with the *optical fiber* and then allow to perform *inteferometry* with the 6U deployable CubeSat using *optical fiber*.

To go further in this preliminary analysis, some other aspects could be studied. To begin, it would be interesting to study the folding limits of the *photon sieves* into the CubeSat. Indeed, the analysis in this work has been made without knowing the folding limits. The analysis is therefore more general.

Afterwards, the secondary optics should be studied. This one could be designed with a filter in order to work with a smaller range of wavelengths. During this work, a device used to correct the chromatic dispersion has been presented. This device could be implemented in the secondary optics and therefore could offer a wider range of wavelengths suitable with the use of a *photon sieve*. The choice between these two kind of designs has to be made with respect to the requirements which would be stated.

Finally, the design and study of the whole interferometer, with the *photon sieve* telescopes and the deployable 6U CubeSat, could be performed in order to analyze its performances.

Appendices

A Matlab files of the propagator

1.1 Photon_Sieve_Propagator.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Main file

clc
clear all
close all


disp(['———Photon Sieve propagator based on Fraunhofer/Fresnel diffraction———']);
disp(['————————————————————[2017–2018]————————————————————']);

%% Loop in case of multi-simulations
new_simulation=1;
while new_simulation==1 %Loop to made multi-simulation if necessary
clearvars Results_PS Results_C
%% Ask the user the kind of simulation
i=1;
while i==1 %Loop in case of a bad entry

Simulation= input('\n Enter: Single simulation (1) - Optimization (2) \n \n');

if Simulation==1 %Condition to leave the loop
    disp([' ']);
    i=0;
elseif Simulation==2
    disp([' ']);
    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end
end

%% Redirection depending of the choice of simulation

if Simulation==1 %Single simulation
   Single_Simulation
elseif Simulation==2 %Optimization
   Optimization_Simulation
end

%% Dispay of the results
% Ask the user for the need of display
    i=1;
 while i==1 %"While" in case of wrong entry
disp(['———Would you like to display the results?———']);
display_results=input('\n Enter: Yes (1) - No (2) \n \n');
disp([' ']);

if display_results==1  %"If" used to leave the while
    i=0;

elseif display_results==2
    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
```

```
end %End of the "if" with display_results
 end %End of the while relative to the ask of a display

while display_results==1  %"While" used in case of multi-display
    if Simulation==1 && PS_VS_Circle==1 %"If used to know if it's the single or the
        optimization simulation and if it's the PS or the C
Display_Results_Single_PS
    elseif Simulation==1 && PS_VS_Circle==2
Display_Results_Single_C
elseif Simulation==2 && PS_VS_Circle==1
Display_Results_Optimization_PS
elseif Simulation==2 && PS_VS_Circle==2
Display_Results_Optimization_C
    end

% Ask the user for the need of an other display
    i=1;
 while i==1 %While in case of wrong entry
 disp(['------Would you like to display an other result?------']);
 new_display=input('\n Enter: Yes (1) - No (2) \n \n');
disp([' ']);

if new_display==1  %"If" used to leave the while
    i=0;
    display_results=1;

elseif new_display==2
    i=0;
    display_results=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with new_display
 end %End of the while relative to the ask of a new display
end %End of the while relative to the display of a result

%% Ask the need of an new simulation

i=1;
while i==1 %While in case of a wrong entry
disp(['------Would you like to make an other simulation?------']);
new_simulation=input('\n Enter: Yes (1) - No (2) \n \n');
disp([' ']);

if new_simulation==1  %"If" used to leave the while
    i=0;
elseif new_simulation==2
    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with new_simulation
end %End of the while relative to the ask of a new simulation

end
clearvars -except Results_PS Results_C
disp(['------End of the propagator------']);
```

1.2 Propagation.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to made the choice of the approximation
% Then the propagation

function [Signal_Observation,L2,M]=Propagation(Signal,L1,lambda,z,Fresnel_Number)

```

```
if Fresnel_Number<0.2 %Choice of the approximation with respect to FresnelNumb

    [Signal_Observation,L2,M]=Fraunhofer(Signal,L1,lambda,z);

else

    [Signal_Observation,L2,M]=Fresnel_TF(Signal,L1,lambda,z);

end


end
```

1.3 Fraunhofer.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to perform the Fraunhofer approximation

function [Signal_Observation,L2,M]=fraunhofer(Signal,L1,lambda,z)

[M,N]=size(Signal);
dx1=L1/M;
k=2*pi/lambda;
L2=lambda*z/dx1;
dx2=lambda*z/L1;
x2=-L2/2:dx2:L2/2-dx2;
[X2,Y2]=meshgrid(x2,x2);
c=1/(j*lambda*z)*exp(j*k/(2*z)*(X2.^2+Y2.^2));
Signal_Observation=c.*ifftshift(fft2(fftshift(Signal)))*dx1^2;
end
```

1.4 Fresnel_TF.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to perform the Fresnel_TF approximation
% TF=Transfer Function


function [Signal_Observation,L2,M]=Fresnel_TF(Signal,L1,lambda,z)
[M,N]=size(Signal);
dx1=L1/M;
k=2*pi/lambda;


fx=-1/(2*dx1):1/L1:1/(2*dx1)-1/L1;
[FX,FY]=meshgrid(fx,fx);
H=exp(-1i*pi*lambda*z*(FX.^2+FY.^2));
H=fftshift(H);
U1=fft2(fftshift(Signal));
U2=H.*U1;
Signal_Observation=ifftshift(ifft2(U2));
L2=L1;
end
```

1.5 Single_Simulation.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Script used for single simulation

disp(['------------------Monochromatique single simulation of the Photon Sieve------------------
    ']);
disp([' ']);
i=1;
while i==1 %While in case of a wrong entry
```

```
disp(['-----Would you like to generate Photon Sieve/Circle(Photon Sieve degenerated)?-----']);
PS_VS_Circle=input('\n Enter: Photon Sieve (1) - Circle (2) \n \n');
disp([' ']);

if PS_VS_Circle==1  %"If" used to leave the while
    i=0;
elseif PS_VS_Circle==2
    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with PS_VS_Circle
end %End of the while relative to the ask of the "kind" of Photon Sieve

%% Ask the user the accuracy of the propagation
i=1;
while i==1 %While in case of a wrong entry
disp(['--------------------------------Speed/accuracy of the simulation--------------------------------']);
Speed=input('\n Enter: Weak accuracy/Fast (1) - Normal accuracy/Normal speed (2) - Strong accuracy/Slow (3) \n \n');
disp([' ']);

if Speed==1  %"If" used to leave the while
    i=0;
elseif Speed==2
    i=0;
    elseif Speed==3
    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with PS_VS_Circle
end %End of the while relative to the ask of the "kind" of Photon Sieve

%% User choose Photon Sieve case
if PS_VS_Circle==1

% Ask user the focal distance of the Photon Sieve
Results_PS.f= input('\n Enter: Focal distance of the Photon Sieve [m] \n \n');

% Ask user the wavelenght
Lambda= input('\n Enter: Wavelength [nm] \n \n');
Results_PS.Lambda=Lambda*10^-9;


% Ask user the density of the photon Sieve
Results_PS.Density=input('\n Enter: Density of the Photon Sieve [%] \n \n');

% Ask user the distance to the observation plan
Results_PS.z=Results_PS.f;

% Ask user the diameter/nb zones of the Photon Sieve
i=1;
[Zones_Max,Diameter_Max]=Diameter_Zones_Max(Results_PS.Lambda,Results_PS.f,Speed);
while i==1 %While in case of a wrong entry
DiameterVSZones=input('\n Enter: Diameter(1) - Number of zones (2) \n');
disp([' ']);

if DiameterVSZones==1  %"If" used to leave the while
    i=0;
    Minimum=2*sqrt(3*Results_PS.Lambda*(Results_PS.f+3*Results_PS.Lambda/4));
Diameter_Display=['Enter: Diameter of the Photon Sieve [m] (at least ',num2str(Minimum),' m and at max ',num2str(Diameter_Max),' m)'];
 disp([' ']);
disp(Diameter_Display);
 disp([' ']);
Results_PS.Diameter= input('');
```

```
elseif DiameterVSZones==2
      i=0;
      Minimum=2*sqrt(3*Results_PS.Lambda*(Results_PS.f+3*Results_PS.Lambda/4));
Zones_Display=['Enter: Number of zones of the Photon Sieve [-] (at max ',num2str(Zones_Max),' zones)'];
 disp([' ']);
disp(Zones_Display);
 disp([' ']);
Zones= input('');
Results_PS.Diameter=2*sqrt((Zones)*Results_PS.Lambda*(Results_PS.f+(Zones)*Results_PS.Lambda/4));
else
      disp([' ']);
      disp(['Error in the entry']);
      disp([' ']);
end %End of the "if" with DiameterVSZones
end %End of the while relative to the ask of diameter or zones

% Display the time estimation

Time_Estimation;

% Generate the Photon Sieve
tic
[Results_PS.Smallest_Holes,Results_PS.Counter_Zones,Results_PS.Numb_Total_Holes,Results_PS.Surface_Ratio,Results_PS.Signal,Results_PS.L1,Results_PS.x,Results_PS.y]=Photon_Sieve_Generator(Results_PS.Diameter,Results_PS.f,Results_PS.Density,Results_PS.Lambda,Speed,1);

Results_PS.I1=abs(Results_PS.Signal.^2);


%Compute the Signal in the observation plan
Results_PS.Fresnel_Number=(Results_PS.Diameter/2)^2/Results_PS.Lambda/Results_PS.z;
[Results_PS.Signal_Observation,Results_PS.L2,Results_PS.M]=Propagation(Results_PS.Signal,Results_PS.L1,Results_PS.Lambda,Results_PS.z,Results_PS.Fresnel_Number);


% Compute parameters of the observation plan
dx2=Results_PS.L2/Results_PS.M;
Results_PS.x2=-Results_PS.L2/2:dx2:Results_PS.L2/2-dx2;
Results_PS.y2=Results_PS.x2;
Results_PS.I2=abs(Results_PS.Signal_Observation).^2;

%Compute totale irradiance both plans
Results_PS.Total_Power1=Irradiance_Power(Results_PS.I1,Results_PS.L1,Results_PS.M);
[Results_PS.Encircle_Power,Results_PS.First_Disk_Power,Results_PS.Total_Power2,Results_PS.Radius_First_Disk]=Encircled_Energy(Results_PS.I2,Results_PS.L2,Results_PS.M,Results_PS.x2,Results_PS.Total_Power1);
toc
%% User choose circle case
else
 % Ask user the diameter of the Circle

Results_C.Diameter= input('\n Enter: Diameter of the Circle [m] \n \n');

% Ask user the wavelenght
Lambda= input('\n Enter: Wavelength [nm] \n \n');
Results_C.Lambda=Lambda*10^-9;

% Ask user the distance with the observation plan
Results_C.z= input('\n Enter: Distance with the observation plan [m] \n \n');

% Generate the Circle

[Results_C.Signal,Results_C.L1,Results_C.x,Results_C.y]=Circle_Generator(Results_C.Diameter,Speed);
Results_C.I1=abs(Results_C.Signal).^2;
% Theoretical rayon of Airy

```

```
Results_C.Radius_Theoretical_Airy=1.22*Results_C.Lambda*Results_C.z/Results_C.Diameter;

% Generate the signal in the observation plan

Results_C.Fresnel_Number=(Results_C.Diameter/2)^2/Results_C.Lambda/Results_C.z;
[Results_C.Signal_Observation,Results_C.L2,Results_C.M]=Propagation(Results_C.Signal,Results_C
    .L1,Results_C.Lambda,Results_C.z,Results_C.Fresnel_Number);

%Values of the observation plan
Results_C.dx2=Results_C.L2/Results_C.M;
Results_C.x2=-Results_C.L2/2:Results_C.dx2:Results_C.L2/2-Results_C.dx2;
Results_C.y2=Results_C.x2;
Results_C.I2=abs(Results_C.Signal_Observation).^2;

%Irradiances and information about Airy circle

Results_C.Total_Power1=Irradiance_Power(Results_C.I1,Results_C.L1,Results_C.M);
[Results_C.Encircle_Power,Results_C.First_Disk_Power,Results_C.Total_Power2,Results_C.
    Radius_Airy]=Encircled_Energy(Results_C.I2,Results_C.L2,Results_C.M,Results_C.x2,Results_C
    .Total_Power1);
Results_C.Error_Airy=abs(Results_C.Radius_Theoretical_Airy-Results_C.Radius_Airy)/Results_C.
    Radius_Theoretical_Airy*100;

end
```

1.6 Optimization_Simulation.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Script used for optimization simulation


disp(['------------------Monochromatique optimization of the Photon Sieve------------------']);
disp([' ']);
i=1;
while i==1 %While in case of a wrong entry
disp(['----Would you like to generate Photon Sieve/Circle(Photon Sieve degenerated)?----']);
PS_VS_Circle=input('\n Enter: Photon Sieve (1) - Circle (2) \n \n');
disp([' ']);

if PS_VS_Circle==1  %"If" used to leave the while
    i=0;
elseif PS_VS_Circle==2
    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with PS_VS_Circle
end %End of the while relative to the ask of the "kind" of Photon Sieve

%% Ask the user the accuracy of the propagation
i=1;
while i==1 %While in case of a wrong entry
disp(['--------------------------------Speed/accuracy of the simulation
    --------------------------------']);
Speed=input('\n Enter: Weak accuracy/Fast (1) - Normal accuracy/Normal speed (2) - Strong
    accuracy/Slow (3) \n \n');
disp([' ']);

if Speed==1  %"If" used to leave the while
    i=0;
elseif Speed==2
    i=0;
    elseif Speed==3
    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
```

```
    disp([' ']);
end %End of the "if" with PS_VS_Circle
end %End of the while relative to the ask of the "kind" of Photon Sieve

%% User choose Photon Sieve case
if PS_VS_Circle==1
    i=1;
while i==1 %While in case of a wrong entry
disp(['-------------------------Which parameter do you want to optimize-------------------------']);disp([' ']);
disp(['Enter: Diameter/Zones (1) - Distance of the observation plan (2) - Density (3)']);
Optimization_Parameter=input('\n                         Wavelength (4) - Focal distance (5) \n \n');
disp([' ']);

if Optimization_Parameter==1  %"If" used to leave the while
    i=0;
elseif Optimization_Parameter==2
    i=0;
    elseif Optimization_Parameter==3
    i=0;
        elseif Optimization_Parameter==4
    i=0;
            elseif Optimization_Parameter==5
    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with Optimization_Parameter
end %End of the while relative to the ask of the parameter to optimize

   if  Optimization_Parameter==1 %Diameter case
f= input('\n Enter: Focal distance of the Photon Sieve [m] \n \n');
Lambda= input('\n Enter: Wavelength [nm] \n \n');
Lambda=Lambda*10^-9;
Density=input('\n Enter: Density of the Photon Sieve [%] \n \n');
% Ask user the diameter/nb zones of the Photon Sieve
i=1;
[Zones_Max,Diameter_Max]=Diameter_Zones_Max(Lambda,f,Speed);
Minimum=2*sqrt(3*Lambda*(f+3*Lambda/4));
while i==1 %While in case of a wrong entry
DiameterVSZones=input('\n Enter: Diameter(1) - Number of zones (2) \n');
disp([' ']);

if DiameterVSZones==1  %"If" used to leave the while
    i=0;
    X=[' Enter: Lower diameter [m] (at least ',num2str(Minimum),' m and at max ',num2str(Diameter_Max),' m)'];
disp([' ']);
disp(X);
disp([' ']);
Diameter_Lower= input('');
Diameter_Higher=input('\n Enter: Higher diameter [m]\n \n');
elseif DiameterVSZones==2
    i=0;
 X=[' Enter: Lower number of zones [-](at max ',num2str(Zones_Max),' zones)'];
disp([' ']);
disp(X);
disp([' ']);
Zones_Lower= input('');
Diameter_Lower=2*sqrt((Zones_Lower)*Lambda*(f+(Zones_Lower)*Lambda/4));
Zones_Higher=input('\n Enter: Higher number of zones [-] \n\n');
Diameter_Higher=2*sqrt((Zones_Higher)*Lambda*(f+(Zones_Higher)*Lambda/4));
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with DiameterVSZones
end %End of the while relative to the ask of diameter or zones

z=f;
```

```
Number_Echantillons=input('\n Enter: Number of samples \n \n');
Time_Estimation;
        for i=1:1:Number_Echantillons
            Results_PS(i).f=f;
            Results_PS(i).Lambda=Lambda;
            Results_PS(i).z=z;
            Results_PS(i).Density=Density;
            if DiameterVSZones==2
                Results_PS(i).Zones=floor(Zones_Lower+(Zones_Higher-Zones_Lower)*(i-1)/(Number_Echantillons-1));
                Results_PS(i).Diameter=2*sqrt((Results_PS(i).Zones)*Lambda*(f+(Results_PS(i).Zones)*Lambda/4));
            else
            Results_PS(i).Diameter=Diameter_Lower+(Diameter_Higher-Diameter_Lower)*(i-1)/(Number_Echantillons-1);
            end
            [Results_PS(i).Smallest_Holes,Results_PS(i).Counter_Zones,Results_PS(i).Numb_Total_Holes,Results_PS(i).Surface_Ratio,Results_PS(i).Signal,Results_PS(i).L1,Results_PS(i).x,Results_PS(i).y]=Photon_Sieve_Generator(Results_PS(i).Diameter,Results_PS(i).f,Results_PS(i).Density,Results_PS(i).Lambda,Speed,0);
Results_PS(i).I1=abs(Results_PS(i).Signal.^2);
Results_PS(i).Fresnel_Number=(Results_PS(i).Diameter/2)^2/Results_PS(i).Lambda/Results_PS(i).z;
[Results_PS(i).Signal_Observation,Results_PS(i).L2,Results_PS(i).M]=Propagation(Results_PS(i).Signal,Results_PS(i).L1,Results_PS(i).Lambda,Results_PS(i).z,Results_PS(i).Fresnel_Number);
dx2=Results_PS(i).L2/Results_PS(i).M;
Results_PS(i).x2=-Results_PS(i).L2/2:dx2:Results_PS(i).L2/2-dx2;
Results_PS(i).y2=Results_PS(i).x2;
Results_PS(i).I2=abs(Results_PS(i).Signal_Observation).^2;
Results_PS(i).Total_Power1=Irradiance_Power(Results_PS(i).I1,Results_PS(i).L1,Results_PS(i).M);
[Results_PS(i).Encircle_Power,Results_PS(i).First_Disk_Power,Results_PS(i).Total_Power2,Results_PS(i).Radius_First_Disk]=Encircled_Energy(Results_PS(i).I2,Results_PS(i).L2,Results_PS(i).M,Results_PS(i).x2,Results_PS(i).Total_Power1);
        end


    elseif Optimization_Parameter==2 %Distance to the observation plan case
f= input('\n Enter: Focal distance of the Photon Sieve [m] \n \n');
Lambda= input('\n Enter: Wavelength [nm] \n \n');
Lambda=Lambda*10^-9;

Density=input('\n Enter: Density of the Photon Sieve [%] \n \n');
% Ask user the diameter/nb zones of the Photon Sieve
i=1;
Minimum=2*sqrt(3*Lambda*(f+3*Lambda/4));
[Zones_Max,Diameter_Max]=Diameter_Zones_Max(Lambda,f,Speed);
while i==1 %While in case of a wrong entry
DiameterVSZones=input('\n Enter: Diameter(1) - Number of zones (2) \n');
disp([' ']);

if DiameterVSZones==1 %"If" used to leave the while
    i=0;
Diameter_Display=['Enter: Diameter of the Photon Sieve [m] (at least ',num2str(Minimum),' m and at max ',num2str(Diameter_Max),' m)'];
 disp([' ']);
disp(Diameter_Display);
 disp([' ']);
Diameter= input('');
elseif DiameterVSZones==2
    i=0;
Zones_Display=['Enter: Number of zones of the Photon Sieve [-](at max ',num2str(Zones_Max),' zones)'];
 disp([' ']);
disp(Zones_Display);
 disp([' ']);
Zones= input('');
Diameter=2*sqrt((Zones)*Lambda*(f+(Zones)*Lambda/4));
else
```

```
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with DiameterVSZones
end %End of the while relative to the ask of diameter or zones

Lower_z=input('\n Enter: Lower distance to the observation plan [m] \n \n');
Higher_z=input('\n Enter: Higher distance to the observation plan [m] \n \n');
Number_Echantillons=input('\n Enter: Number of echantillons \n \n');
Time_Estimation;
        for i=1:1:Number_Echantillons
            Results_PS(i).f=f;
            Results_PS(i).Lambda=Lambda;
            Results_PS(i).z=Lower_z+(Higher_z-Lower_z)*(i-1)/(Number_Echantillons-1);
            Results_PS(i).Density=Density;
            Results_PS(i).Diameter=Diameter;
            [Results_PS(i).Smallest_Holes,Results_PS(i).Counter_Zones,Results_PS(i).
                Numb_Total_Holes,Results_PS(i).Surface_Ratio,Results_PS(i).Signal,Results_PS(i)
                .L1,Results_PS(i).x,Results_PS(i).y]=Photon_Sieve_Generator(Results_PS(i).
                Diameter,Results_PS(i).f,Results_PS(i).Density,Results_PS(i).Lambda,Speed,0);
Results_PS(i).I1=abs(Results_PS(i).Signal.^2);
Results_PS(i).Fresnel_Number=(Results_PS(i).Diameter/2)^2/Results_PS(i).Lambda/Results_PS(i).z
    ;
[Results_PS(i).Signal_Observation,Results_PS(i).L2,Results_PS(i).M]=Propagation(Results_PS(i).
    Signal,Results_PS(i).L1,Results_PS(i).Lambda,Results_PS(i).z,Results_PS(i).Fresnel_Number)
    ;
dx2=Results_PS(i).L2/Results_PS(i).M;
Results_PS(i).x2=-Results_PS(i).L2/2:dx2:Results_PS(i).L2/2-dx2;
Results_PS(i).y2=Results_PS(i).x2;
Results_PS(i).I2=abs(Results_PS(i).Signal_Observation).^2;
Results_PS(i).Total_Power1=Irradiance_Power(Results_PS(i).I1,Results_PS(i).L1,Results_PS(i).M)
    ;
[Results_PS(i).Encircle_Power,Results_PS(i).First_Disk_Power,Results_PS(i).Total_Power2,
    Results_PS(i).Radius_First_Disk]=Encircled_Energy(Results_PS(i).I2,Results_PS(i).L2,
    Results_PS(i).M,Results_PS(i).x2,Results_PS(i).Total_Power1);
        end

    elseif Optimization_Parameter==3 %Density case
        f= input('\n Enter: Focal distance of the Photon Sieve [m] \n \n');
Lambda= input('\n Enter: Wavelength [nm] \n \n');
Lambda=Lambda*10^-9;
% Ask user the diameter/nb zones of the Photon Sieve
i=1;
Minimum=2*sqrt(3*Lambda*(f+3*Lambda/4));
[Zones_Max,Diameter_Max]=Diameter_Zones_Max(Lambda,f,Speed);
while i==1 %While in case of a wrong entry
DiameterVSZones=input('\n Enter: Diameter(1) - Number of zones (2) \n');
disp([' ']);

if DiameterVSZones==1  %"If" used to leave the while
    i=0;
Diameter_Display=['Enter: Diameter of the Photon Sieve [m] (at least ',num2str(Minimum),' m
    and at max ',num2str(Diameter_Max),' m)'];
 disp([' ']);
disp(Diameter_Display);
 disp([' ']);
Diameter= input('');
elseif DiameterVSZones==2
    i=0;
Zones_Display=['Enter: Number of zones of the Photon Sieve [-](at max ',num2str(Zones_Max),'
    zones)'];
 disp([' ']);
disp(Zones_Display);
 disp([' ']);
Zones= input('');
Diameter=2*sqrt((Zones)*Lambda*(f+(Zones)*Lambda/4));
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
```

```
end %End of the "if" with DiameterVSZones
end %End of the while relative to the ask of diameter or zones
Lower_Density=input('\n Enter: Lower density of the Photon Sieve [%] \n \n');
Higher_Density=input('\n Enter: Higher density of the Photon Sieve [%] \n \n');
Number_Echantillons=input('\n Enter: Number of samples \n \n');
Time_Estimation;
        for i=1:1:Number_Echantillons
            Results_PS(i).f=f;
            Results_PS(i).Lambda=Lambda;
            Results_PS(i).z=f;
            Results_PS(i).Density=Lower_Density+(Higher_Density-Lower_Density)*(i-1)/(Number_Echantillons-1);
            Results_PS(i).Diameter=Diameter;
            [Results_PS(i).Smallest_Holes,Results_PS(i).Counter_Zones,Results_PS(i).Numb_Total_Holes,Results_PS(i).Surface_Ratio,Results_PS(i).Signal,Results_PS(i).L1,Results_PS(i).x,Results_PS(i).y]=Photon_Sieve_Generator(Results_PS(i).Diameter,Results_PS(i).f,Results_PS(i).Density,Results_PS(i).Lambda,Speed,0);
Results_PS(i).I1=abs(Results_PS(i).Signal.^2);
Results_PS(i).Fresnel_Number=(Results_PS(i).Diameter/2)^2/Results_PS(i).Lambda/Results_PS(i).z;
[Results_PS(i).Signal_Observation,Results_PS(i).L2,Results_PS(i).M]=Propagation(Results_PS(i).Signal,Results_PS(i).L1,Results_PS(i).Lambda,Results_PS(i).z,Results_PS(i).Fresnel_Number);
dx2=Results_PS(i).L2/Results_PS(i).M;
Results_PS(i).x2=-Results_PS(i).L2/2:dx2:Results_PS(i).L2/2-dx2;
Results_PS(i).y2=Results_PS(i).x2;
Results_PS(i).I2=abs(Results_PS(i).Signal_Observation).^2;
Results_PS(i).Total_Power1=Irradiance_Power(Results_PS(i).I1,Results_PS(i).L1,Results_PS(i).M);
[Results_PS(i).Encircle_Power,Results_PS(i).First_Disk_Power,Results_PS(i).Total_Power2,Results_PS(i).Radius_First_Disk]=Encircled_Energy(Results_PS(i).I2,Results_PS(i).L2,Results_PS(i).M,Results_PS(i).x2,Results_PS(i).Total_Power1);
        end

    elseif Optimization_Parameter==4 %Wavelength case
            f= input('\n Enter: Focal distance of the Photon Sieve [m] \n \n');
Lower_Lambda= input('\n Enter: Lower wavelength [nm] \n \n');
Lower_Lambda=Lower_Lambda*10^-9;
Higher_Lambda= input('\n Enter: Higher wavelength [nm] \n \n');
Higher_Lambda=Higher_Lambda*10^-9;

% Ask user the diameter/nb zones of the Photon Sieve
i=1;
Minimum=2*sqrt(3*Higher_Lambda*(f+3*Higher_Lambda/4));
[Zones_Max,Diameter_Max]=Diameter_Zones_Max(Lower_Lambda,f,Speed);
while i==1 %While in case of a wrong entry
DiameterVSZones=input('\n Enter: Diameter(1) - Number of zones (2) \n');
disp([' ']);

if DiameterVSZones==1  %"If" used to leave the while
    i=0;
Diameter_Display=['Enter: Diameter of the Photon Sieve [m] (at least ',num2str(Minimum),' m and at max ',num2str(Diameter_Max),' m)'];
 disp([' ']);
disp(Diameter_Display);
 disp([' ']);
Diameter= input('');
elseif DiameterVSZones==2
    i=0;
Zones_Display=['Enter: Number of zones of the Photon Sieve [-](at max ',num2str(Zones_Max),' zones)'];
 disp([' ']);
disp(Zones_Display);
 disp([' ']);
Zones= input('');
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with DiameterVSZones
```

```
end

Density=input('\n Enter: Density of the Photon Sieve [%] \n \n');
Number_Echantillons=input('\n Enter: Number of samples \n \n');
Time_Estimation;
        for i=1:1:Number_Echantillons
            Results_PS(i).f=f;
            Results_PS(i).Lambda=Lower_Lambda+(Higher_Lambda-Lower_Lambda)*(i-1)/(Number_Echantillons-1);
            Results_PS(i).z=f;
            Results_PS(i).Density=Density;
            if DiameterVSZones==1
            Results_PS(i).Diameter=Diameter;
            else
                Results_PS(i).Diameter=2*sqrt((Zones)*Results_PS(i).Lambda*(f+(Zones)*Results_PS(i).Lambda/4));
            end

            [Results_PS(i).Smallest_Holes,Results_PS(i).Counter_Zones,Results_PS(i).Numb_Total_Holes,Results_PS(i).Surface_Ratio,Results_PS(i).Signal,Results_PS(i).L1,Results_PS(i).x,Results_PS(i).y]=Photon_Sieve_Generator(Results_PS(i).Diameter,Results_PS(i).f,Results_PS(i).Density,Results_PS(i).Lambda,Speed,0);
Results_PS(i).I1=abs(Results_PS(i).Signal.^2);
Results_PS(i).Fresnel_Number=(Results_PS(i).Diameter/2)^2/Results_PS(i).Lambda/Results_PS(i).z;
[Results_PS(i).Signal_Observation,Results_PS(i).L2,Results_PS(i).M]=Propagation(Results_PS(i).Signal,Results_PS(i).L1,Results_PS(i).Lambda,Results_PS(i).z,Results_PS(i).Fresnel_Number);
dx2=Results_PS(i).L2/Results_PS(i).M;
Results_PS(i).x2=-Results_PS(i).L2/2:dx2:Results_PS(i).L2/2-dx2;
Results_PS(i).y2=Results_PS(i).x2;
Results_PS(i).I2=abs(Results_PS(i).Signal_Observation).^2;
Results_PS(i).Total_Power1=Irradiance_Power(Results_PS(i).I1,Results_PS(i).L1,Results_PS(i).M);
[Results_PS(i).Encircle_Power,Results_PS(i).First_Disk_Power,Results_PS(i).Total_Power2,Results_PS(i).Radius_First_Disk]=Encircled_Energy(Results_PS(i).I2,Results_PS(i).L2,Results_PS(i).M,Results_PS(i).x2,Results_PS(i).Total_Power1);
        end


    elseif Optimization_Parameter==5 %Focal distance case
                Lower_f= input('\n Enter: Lower focal distance of the Photon Sieve [m] \n \n');
          Higher_f= input('\n Enter: Higher focal distance of the Photon Sieve [m] \n \n');
Lambda= input('\n Enter: Wavelength [nm] \n \n');
Lambda=Lambda*10^-9;
% Ask user the diameter/nb zones of the Photon Sieve
i=1;
Minimum=2*sqrt(3*Lambda*(Higher_f+3*Lambda/4));
[Zones_Max,Diameter_Max]=Diameter_Zones_Max(Lambda,Lower_f,Speed);
while i==1 %While in case of a wrong entry
DiameterVSZones=input('\n Enter: Diameter(1) - Number of zones (2) \n');
disp([' ']);

if DiameterVSZones==1  %"If" used to leave the while
    i=0;
Diameter_Display=['Enter: Diameter of the Photon Sieve [m] (at least ',num2str(Minimum),' m and at max ',num2str(Diameter_Max),' m)'];
 disp([' ']);
disp(Diameter_Display);
 disp([' ']);
Diameter= input('');
elseif DiameterVSZones==2
    i=0;
Zones_Display=['Enter: Number of zones of the Photon Sieve [-](at max ',num2str(Zones_Max),' zones)'];
 disp([' ']);
disp(Zones_Display);
 disp([' ']);
Zones= input('');
```

```
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with DiameterVSZones
end

Density=input('\n Enter: Density of the Photon Sieve [%] \n \n');
Number_Echantillons=input('\n Enter: Number of samples \n \n');
Time_Estimation;
        for i=1:1:Number_Echantillons
            Results_PS(i).f=Lower_f+(Higher_f-Lower_f)*(i-1)/(Number_Echantillons-1);
            Results_PS(i).Lambda=Lambda;
            Results_PS(i).z=Results_PS(i).f;
            Results_PS(i).Density=Density;
                        if DiameterVSZones==1
            Results_PS(i).Diameter=Diameter;
            else
                Results_PS(i).Diameter=2*sqrt((Zones)*Results_PS(i).Lambda*(Results_PS(i).f+(
                    Zones)*Results_PS(i).Lambda/4));
            end
            [Results_PS(i).Smallest_Holes,Results_PS(i).Counter_Zones,Results_PS(i).
                Numb_Total_Holes,Results_PS(i).Surface_Ratio,Results_PS(i).Signal,Results_PS(i)
                .L1,Results_PS(i).x,Results_PS(i).y]=Photon_Sieve_Generator(Results_PS(i).
                Diameter,Results_PS(i).f,Results_PS(i).Density,Results_PS(i).Lambda,Speed,0);
Results_PS(i).I1=abs(Results_PS(i).Signal.^2);
Results_PS(i).Fresnel_Number=(Results_PS(i).Diameter/2)^2/Results_PS(i).Lambda/Results_PS(i).z
    ;
[Results_PS(i).Signal_Observation,Results_PS(i).L2,Results_PS(i).M]=Propagation(Results_PS(i).
    Signal,Results_PS(i).L1,Results_PS(i).Lambda,Results_PS(i).z,Results_PS(i).Fresnel_Number)
    ;
dx2=Results_PS(i).L2/Results_PS(i).M;
Results_PS(i).x2=-Results_PS(i).L2/2:dx2:Results_PS(i).L2/2-dx2;
Results_PS(i).y2=Results_PS(i).x2;
Results_PS(i).I2=abs(Results_PS(i).Signal_Observation).^2;
Results_PS(i).Total_Power1=Irradiance_Power(Results_PS(i).I1,Results_PS(i).L1,Results_PS(i).M)
    ;
[Results_PS(i).Encircle_Power,Results_PS(i).First_Disk_Power,Results_PS(i).Total_Power2,
    Results_PS(i).Radius_First_Disk]=Encircled_Energy(Results_PS(i).I2,Results_PS(i).L2,
    Results_PS(i).M,Results_PS(i).x2,Results_PS(i).Total_Power1);
        end

   end



 %% User choose circle case
else
        i=1;
while i==1 %While in case of a wrong entry
disp(['----Which parameter do you want to optimize----']);
Optimization_Parameter=input('\n Enter: Diameter (1) - Distance of the observation plan (2) -
     Wavelength (3) \n \n');
disp([' ']);

if Optimization_Parameter==1  %"If" used to leave the while
    i=0;
elseif Optimization_Parameter==2
    i=0;
    elseif Optimization_Parameter==3

    i=0;
else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
end %End of the "if" with Optimization_Parameter
end %End of the while relative to the ask of the parameter to optimize

   if  Optimization_Parameter==1 %Diameter case
```

```
Diameter_Lower= input('\n Enter: Lower diameter of the Circle [m] \n \n');
Diameter_Higher= input('\n Enter: Higher diameter of the Circle [m] \n \n');
Lambda= input('\n Enter: Wavelength [nm] \n \n');
Lambda=Lambda*10^-9;
z= input('\n Enter: Distance with the observation plan [m] \n \n');
Number_Echantillons=input('\n Enter: Number of samples \n \n');
for i=1:1:Number_Echantillons
    Results_C(i).z=z;
    Results_C(i).Lambda=Lambda;
    Results_C(i).Diameter=Diameter_Lower+(Diameter_Higher-Diameter_Lower)*(i-1)/(
        Number_Echantillons-1);
    [Results_C(i).Signal,Results_C(i).L1,Results_C(i).x,Results_C(i).y]=Circle_Generator(
        Results_C(i).Diameter,Speed);
Results_C(i).I1=abs(Results_C(i).Signal).^2;
Results_C(i).Radius_Theoretical_Airy=1.22*Results_C(i).Lambda*Results_C(i).z/Results_C(i).
    Diameter;
Results_C(i).Fresnel_Number=(Results_C(i).Diameter/2)^2/Results_C(i).Lambda/Results_C(i).z;
[Results_C(i).Signal_Observation,Results_C(i).L2,Results_C(i).M]=Propagation(Results_C(i).
    Signal,Results_C(i).L1,Results_C(i).Lambda,Results_C(i).z,Results_C(i).Fresnel_Number);
Results_C(i).dx2=Results_C(i).L2/Results_C(i).M;
Results_C(i).x2=-Results_C(i).L2/2:Results_C(i).dx2:Results_C(i).L2/2-Results_C(i).dx2;
Results_C(i).y2=Results_C(i).x2;
Results_C(i).I2=abs(Results_C(i).Signal_Observation).^2;
Results_C(i).Total_Power1=Irradiance_Power(Results_C(i).I1,Results_C(i).L1,Results_C(i).M);
[Results_C(i).Encircle_Power,Results_C(i).First_Disk_Power,Results_C(i).Total_Power2,Results_C
    (i).Radius_Airy]=Encircled_Energy(Results_C(i).I2,Results_C(i).L2,Results_C(i).M,Results_C
    (i).x2,Results_C(i).Total_Power1);
Results_C(i).Error_Airy=abs(Results_C(i).Radius_Theoretical_Airy-Results_C(i).Radius_Airy)/
    Results_C(i).Radius_Theoretical_Airy*100;
end

   elseif Optimization_Parameter==2 %Distance to the observation plan case

     Diameter= input('\n Enter: Diameter of the Circle [m] \n \n');
Lambda= input('\n Enter: Wavelength [nm] \n \n');
Lambda=Lambda*10^-9;
z_Lower= input('\n Enter: Lower distance with the observation plan [m] \n \n');
z_Higher= input('\n Enter: Higher distance with the observation plan [m] \n \n');
Number_Echantillons=input('\n Enter: Number of samples \n \n');
for i=1:1:Number_Echantillons
    Results_C(i).z=z_Lower+(z_Higher-z_Lower)*(i-1)/(Number_Echantillons-1);
    Results_C(i).Lambda=Lambda;
    Results_C(i).Diameter=Diameter;
    [Results_C(i).Signal,Results_C(i).L1,Results_C(i).x,Results_C(i).y]=Circle_Generator(
        Results_C(i).Diameter,Speed);
Results_C(i).I1=abs(Results_C(i).Signal).^2;
Results_C(i).Radius_Theoretical_Airy=1.22*Results_C(i).Lambda*Results_C(i).z/Results_C(i).
    Diameter;
Results_C(i).Fresnel_Number=(Results_C(i).Diameter/2)^2/Results_C(i).Lambda/Results_C(i).z;
[Results_C(i).Signal_Observation,Results_C(i).L2,Results_C(i).M]=Propagation(Results_C(i).
    Signal,Results_C(i).L1,Results_C(i).Lambda,Results_C(i).z,Results_C(i).Fresnel_Number);
Results_C(i).dx2=Results_C(i).L2/Results_C(i).M;
Results_C(i).x2=-Results_C(i).L2/2:Results_C(i).dx2:Results_C(i).L2/2-Results_C(i).dx2;
Results_C(i).y2=Results_C(i).x2;
Results_C(i).I2=abs(Results_C(i).Signal_Observation).^2;
Results_C(i).Total_Power1=Irradiance_Power(Results_C(i).I1,Results_C(i).L1,Results_C(i).M);
[Results_C(i).Encircle_Power,Results_C(i).First_Disk_Power,Results_C(i).Total_Power2,Results_C
    (i).Radius_Airy]=Encircled_Energy(Results_C(i).I2,Results_C(i).L2,Results_C(i).M,Results_C
    (i).x2,Results_C(i).Total_Power1);
Results_C(i).Error_Airy=abs(Results_C(i).Radius_Theoretical_Airy-Results_C(i).Radius_Airy)/
    Results_C(i).Radius_Theoretical_Airy*100;
end

    elseif Optimization_Parameter==3 %Wavelength case
       Diameter= input('\n Enter: Diameter of the Circle [m] \n \n');
Lambda_Lower= input('\n Enter: Lower wavelength [nm] \n \n');
Lambda_Lower=Lambda_Lower*10^-9;
Lambda_Higher= input('\n Enter: Higher wavelength [nm] \n \n');
Lambda_Higher=Lambda_Higher*10^-9;
z= input('\n Enter: Distance with the observation plan [m] \n \n');
```

```
Number_Echantillons=input('\n Enter: Number of samples \n \n');
for i=1:1:Number_Echantillons
    Results_C(i).z=z;
    Results_C(i).Lambda=Lambda_Lower+(Lambda_Higher-Lambda_Lower)*(i-1)/(Number_Echantillons
        -1);
    Results_C(i).Diameter=Diameter;
    [Results_C(i).Signal,Results_C(i).L1,Results_C(i).x,Results_C(i).y]=Circle_Generator(
        Results_C(i).Diameter,Speed);
Results_C(i).I1=abs(Results_C(i).Signal).^2;
Results_C(i).Radius_Theoretical_Airy=1.22*Results_C(i).Lambda*Results_C(i).z/Results_C(i).
    Diameter;
Results_C(i).Fresnel_Number=(Results_C(i).Diameter/2)^2/Results_C(i).Lambda/Results_C(i).z;
[Results_C(i).Signal_Observation,Results_C(i).L2,Results_C(i).M]=Propagation(Results_C(i).
    Signal,Results_C(i).L1,Results_C(i).Lambda,Results_C(i).z,Results_C(i).Fresnel_Number);
Results_C(i).dx2=Results_C(i).L2/Results_C(i).M;
Results_C(i).x2=-Results_C(i).L2/2:Results_C(i).dx2:Results_C(i).L2/2-Results_C(i).dx2;
Results_C(i).y2=Results_C(i).x2;
Results_C(i).I2=abs(Results_C(i).Signal_Observation).^2;
Results_C(i).Total_Power1=Irradiance_Power(Results_C(i).I1,Results_C(i).L1,Results_C(i).M);
[Results_C(i).Encircle_Power,Results_C(i).First_Disk_Power,Results_C(i).Total_Power2,Results_C
    (i).Radius_Airy]=Encircled_Energy(Results_C(i).I2,Results_C(i).L2,Results_C(i).M,Results_C
    (i).x2,Results_C(i).Total_Power1);
Results_C(i).Error_Airy=abs(Results_C(i).Radius_Theoretical_Airy-Results_C(i).Radius_Airy)/
    Results_C(i).Radius_Theoretical_Airy*100;
end


   end


end
```

1.7 Display_Results_Optimization_C.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Script used for display the results of the Circle (Optimization case)

i=1;
while i==1
    disp([' ']);
        disp([' ']);
    disp(['Enter: Error with the array radius (1) - Total power (2) - Encircled power (3)']);
    disp([' Irrandiance in entry plan (2D) (4) - Irrandiance in observation plan (2D) (5)']);
    disp([' Irrandiance in entry plan (3D) (6) - Irrandiance in observation plan (3D) (7)']);
    disp(['                    Profile irrandiance in observation plan (8)']);
    disp([' ']);
    Display=input('\n');

    if Optimization_Parameter==1

            if Display==1
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_C(i).Diameter;
            y(i)=Results_C(i).Error_Airy;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Error in Airy radius [%]','interpreter','latex','fontsize',16);
xlabel('Diameter [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_C(1).z),'m - $\lambda=$',num2str(Results_C(1).Lambda*10^9),'nm'],'
    Interpreter','latex', 'FontSize', 14);
    elseif Display==2
        i=0;
                    for i=1:Number_Echantillons
            x(i)=Results_C(i).Diameter;
            y(i)=Results_C(i).Total_Power1;
                    end
```

```
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Total Power [Watts]','interpreter','latex','fontsize',16);
xlabel('Diameter [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_C(1).z),'m - $\lambda=$',num2str(Results_C(1).Lambda*10^9),'nm'],'
    Interpreter','latex', 'FontSize', 14);
    elseif Display==3
   i=0;
          disp([' ']);
    disp(['Which diameter? [',num2str(Results_C(1).Diameter),'[m]-',num2str(Results_C(
        Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]']);
    disp([' ']);
        disp([' ']);
disp(['Purcentage of the total power in the "first disk": ',num2str(Results_C(j).
    First_Disk_Power*100),' [%]']);
 disp([' ']);
    figure
 plot(Results_C(j).Encircle_Power(1,:),Results_C(j).Encircle_Power(2,:), 'LineWidth',1,'Color'
    ,'b');
ylabel('Encircle power [%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D
    =',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);

    elseif Display==4
      i=0;
          disp([' ']);
    disp(['Which diameter? [',num2str(Results_C(1).Diameter),'[m]-',num2str(Results_C(
        Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]']);
    disp([' ']);
      imagesc(Results_C(j).x,Results_C(j).y,nthroot(Results_C(j).I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).
    Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==5
        i=0;
                disp([' ']);
    disp(['Which diameter? [',num2str(Results_C(1).Diameter),'[m]-',num2str(Results_C(
        Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
```

```
    disp(['Closer diameter from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]']);
    disp([' ']);
      imagesc(Results_C(j).x2,Results_C(j).y2,nthroot(Results_C(j).I2,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==6
        i=0;
                disp([' ']);
    disp(['Which diameter? [',num2str(Results_C(1).Diameter),'[m]-',num2str(Results_C(Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]']);
    disp([' ']);
        surf(Results_C(j).x,Results_C(j).y,nthroot(Results_C(j).I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==7
        i=0;
                disp([' ']);
    disp(['Which diameter? [',num2str(Results_C(1).Diameter),'[m]-',num2str(Results_C(Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]']);
    disp([' ']);
        surf(Results_C(j).x2,Results_C(j).y2,nthroot(Results_C(j).I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==8
        i=0;
                disp([' ']);
    disp(['Which diameter? [',num2str(Results_C(1).Diameter),'[m]-',num2str(Results_C(Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
```

```
    disp(['Closer diameter from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]']);
    disp([' ']);
    figure
        plot(Results_C(j).x2,Results_C(j).I2(Results_C(j).M/2+1,:));
xlabel('x (m)');ylabel('Irradiance');
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
            end

    elseif  Optimization_Parameter==2


            if Display==1
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_C(i).z;
            y(i)=Results_C(i).Error_Airy;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Error in Airy radius [\%]','interpreter','latex','fontsize',16);
xlabel('Distance to the observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_C(1).Lambda*10^9),'nm - D=',num2str(Results_C(1).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==2
        i=0;
                        for i=1:Number_Echantillons
            x(i)=Results_C(i).z;
            y(i)=Results_C(i).Total_Power1;
                        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Total Power [Watts]','interpreter','latex','fontsize',16);
xlabel('Distance to the observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_C(1).Lambda*10^9),'nm - D=',num2str(Results_C(1).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);

    elseif Display==3
   i=0;
         disp([' ']);
    disp(['Which distance to the observation plan ? [',num2str(Results_C(1).z),'[m]-',num2str(Results_C(Number_Echantillons).z),'[m]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons-1)/(Diameter_Higher-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance to the observation plan from the echantillons: [',num2str(Results_C(j).z),'[m]]']);
    disp([' ']);
        disp([' ']);
disp(['Percentage of the total power in the "first disk": ',num2str(Results_C(j).First_Disk_Power*100),' [%]']);
 disp([' ']);
     figure
 plot(Results_C(j).Encircle_Power(1,:),Results_C(j).Encircle_Power(2,:), 'LineWidth',1,'Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);

```

```

    elseif Display==4
      i=0;
        disp([' ']);
    disp(['Which distance to the observation plan ? [',num2str(Results_C(1).z),'[m]-',num2str(Results_C(Number_Echantillons).z),'[m]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons-1)/(Diameter_Higher-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance to the observation plan from the echantillons: [',num2str(Results_C(j).z),'[m]]']);
    disp([' ']);
      imagesc(Results_C(j).x,Results_C(j).y,nthroot(Results_C(j).I1,3));
      title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m']);
    elseif Display==5
        i=0;
          disp([' ']);
    disp(['Which distance to the observation plan ? [',num2str(Results_C(1).z),'[m]-',num2str(Results_C(Number_Echantillons).z),'[m]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons-1)/(Diameter_Higher-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance to the observation plan from the echantillons: [',num2str(Results_C(j).z),'[m]]']);
    disp([' ']);
      imagesc(Results_C(j).x2,Results_C(j).y2,nthroot(Results_C(j).I2,3));
      title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_C(j).z),'m [Observation plan]'], 'FontSize', 14);
    elseif Display==6
        i=0;
          disp([' ']);
    disp(['Which distance to the observation plan ? [',num2str(Results_C(1).z),'[m]-',num2str(Results_C(Number_Echantillons).z),'[m]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons-1)/(Diameter_Higher-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance to the observation plan from the echantillons: [',num2str(Results_C(j).z),'[m]]']);
    disp([' ']);
        surf(Results_C(j).x,Results_C(j).y,nthroot(Results_C(j).I1,3));
camlight left;
lighting phong
shading interp
```

```
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).
    Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==7
        i=0;
          disp([' ']);
    disp(['Which distance to the observation plan ? [',num2str(Results_C(1).z),'[m]-',num2str(
        Results_C(Number_Echantillons).z),'[m]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons -1)/(Diameter_Higher-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance to the observation plan from the echantillons: [',num2str(Results_C
        (j).z),'[m]]']);
    disp([' ']);
        surf(Results_C(j).x2,Results_C(j).y2,nthroot(Results_C(j).I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D
    =',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==8
        i=0;
          disp([' ']);
    disp(['Which distance to the observation plan ? [',num2str(Results_C(1).z),'[m]-',num2str(
        Results_C(Number_Echantillons).z),'[m]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons -1)/(Diameter_Higher-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance to the observation plan from the echantillons: [',num2str(Results_C
        (j).z),'[m]]']);
    disp([' ']);
    figure
        plot(Results_C(j).x2,Results_C(j).I2(Results_C(j).M/2+1,:));
xlabel('x (m)');ylabel('Irradiance');
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D
    =',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
            end


    elseif Optimization_Parameter==3

            if Display==1
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_C(i).Lambda;
            y(i)=Results_C(i).Error_Airy;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Error in Airy radius [\%]','interpreter','latex','fontsize',16);
xlabel('Lambda [m]','interpreter','latex','fontsize',16);
```

```
title(['z=',num2str(Results_C(1).z),'m -  D=',num2str(Results_C(1).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==2
        i=0;
                            for i=1:Number_Echantillons
            x(i)=Results_C(i).Lambda;
            y(i)=Results_C(i).Total_Power1;
                            end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Total Power [Watts]','interpreter','latex','fontsize',16);
xlabel('Lambda [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_C(1).z),'m - D=',num2str(Results_C(1).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);

    elseif Display==3
   i=0;
          disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_C(1).Lambda),'[nm]-',num2str(10^9*Results_C(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons-1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]'])
        ;
    disp([' ']);
        disp([' ']);
disp(['Purcentage of the total power in the "first disk": ',num2str(Results_C(j).First_Disk_Power*100),' [%]']);
 disp([' ']);
 figure
 plot(Results_C(j).Encircle_Power(1,:),Results_C(j).Encircle_Power(2,:), 'LineWidth',1,'Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);

    elseif Display==4
      i=0;
          disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_C(1).Lambda),'[nm]-',num2str(10^9*Results_C(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons-1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]'])
        ;
    disp([' ']);
      imagesc(Results_C(j).x,Results_C(j).y,nthroot(Results_C(j).I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==5
```

```
            i=0;
              disp([' ']);
        disp(['Which wavelength? [',num2str(10^9*Results_C(1).Lambda),'[nm]-',num2str(10^9*
            Results_C(Number_Echantillons).Lambda),'[nm]]']);
        disp([' ']);
        Lambda=input('\n');
        Lambda=Lambda/10^9;
        j=(Lambda-Lower_Lambda)*(Number_Echantillons-1)/(Higher_Lambda-Lower_Lambda)+1;
        if abs(ceil(j)-j)>abs(floor(j)-j)
            j=floor(j);
        else
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer wavelength from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]'])
            ;
        disp([' ']);
          imagesc(Results_C(j).x2,Results_C(j).y2,nthroot(Results_C(j).I2,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D
        =',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
        elseif Display==6
            i=0;
              disp([' ']);
        disp(['Which wavelength? [',num2str(10^9*Results_C(1).Lambda),'[nm]-',num2str(Results_C(
            Number_Echantillons).Lambda),'[nm]]']);
        disp([' ']);
        Lambda=input('\n');
        Lambda=Lambda/10^9;
        j=(Lambda-Lower_Lambda)*(Number_Echantillons-1)/(Higher_Lambda-Lower_Lambda)+1;
        if abs(ceil(j)-j)>abs(floor(j)-j)
            j=floor(j);
        else
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer wavelength from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]'])
            ;
        disp([' ']);
            surf(Results_C(j).x,Results_C(j).y,nthroot(Results_C(j).I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D=',num2str(Results_C(j).
        Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
        elseif Display==7
            i=0;
              disp([' ']);
        disp(['Which wavelength? [',num2str(10^9*Results_C(1).Lambda),'[nm]-',num2str(10^9*
            Results_C(Number_Echantillons).Lambda),'[nm]]']);
        disp([' ']);
        Lambda=input('\n');
        Lambda=Lambda/10^9;
        j=(Lambda-Lower_Lambda)*(Number_Echantillons-1)/(Higher_Lambda-Lower_Lambda)+1;
        if abs(ceil(j)-j)>abs(floor(j)-j)
            j=floor(j);
        else
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer wavelength from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]'])
            ;
        disp([' ']);
            surf(Results_C(j).x2,Results_C(j).y2,nthroot(Results_C(j).I2,3));
camlight left;
lighting phong
```

```
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D =',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==8
        i=0;
            disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_C(1).Lambda),'[nm]-',num2str(10^9*Results_C(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons -1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(Results_C(j).Diameter),'[m]]'])
        ;
    disp([' ']);
    figure
        plot(Results_C(j).x2,Results_C(j).I2(Results_C(j).M/2+1,:));
xlabel('x (m)');ylabel('Irradiance');
title(['z=',num2str(Results_C(j).z),'m - $\lambda=$',num2str(Results_C(j).Lambda*10^9),'nm - D =',num2str(Results_C(j).Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);

    end

    end

end
```

1.8 Display_Results_Optimization_PS.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Script used for display the results of the Photon Sieve(Optimization case)
    i=1;
    while i==1
    disp([' ']);
    disp(['Enter:     Number of holes (1) - Number of zones (2) - Total power (3) ']);
    disp(['      Percentage of the irradiance in the fist peak (4) - Surface ratio (5)']);
    disp(['Irrandiance in entry plan (2D) (6) - Irrandiance in observation plan (PSF-2D) (7)'
        ]);
    disp(['Irrandiance in entry plan (3D) (8) - Irrandiance in observation plan (PSF-3D) (9)'
        ]);
    disp(['            Profile irrandiance in observation plan (PSF-Profile) (10)']);
    disp(['          Encircled energy (11) - Radius of the first peak (12)']);
    disp(['              Diameter of the smallest holes (13)']);
    disp([' ']);
    Display=input('\n');

if  Optimization_Parameter==1 && DiameterVSZones==2

    if Display==1
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Zones;
            y(i)=Results_PS(i).Numb_Total_Holes;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
```

```
ylabel('Number of holes [-]','interpreter','latex','fontsize',16);
xlabel('Number of zones [-]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
    f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==2
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).Zones;
            y(i)=Results_PS(i).Diameter;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Diameter [m]','interpreter','latex','fontsize',16);
xlabel('Number of zones [-]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
    f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==3
   i=0;
    for i=1:Number_Echantillons
            x(i)=Results_PS(i).Zones;
            y(i)=Results_PS(i).Total_Power1;
    end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power [Watts]','interpreter','latex','fontsize',16);
xlabel('Number of zones [-]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
    f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);
     elseif Display==4
   i=0;
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Zones;
            y(i)=Results_PS(i).First_Disk_Power*100;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power in the first disk [\%]','interpreter','latex','fontsize',16);
xlabel('Number of zones [-]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
    f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);
     elseif Display==5
   i=0;
         for i=1:Number_Echantillons
            x(i)=Results_PS(i).Zones;
            y(i)=Results_PS(i).Surface_Ratio;
         end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Surface ratio [-]','interpreter','latex','fontsize',16);
xlabel('Number of zones [-]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
    f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);

    elseif Display==6
      i=0;
    disp([' ']);
    disp(['Which number of zones? [',num2str(Results_PS(1).Zones),'[-]-',num2str(Results_PS(
        Number_Echantillons).Zones),'[-]]']);
    disp([' ']);
    Zones=input('\n');
    j=(Zones-Zones_Lower)*(Number_Echantillons-1)/(Zones_Higher-Zones_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer number of zones from the echantillons: [',num2str(Results_PS(j).Zones),'[m]]
        ']);
```

```
    disp([' ']);

      imagesc(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==7
      i=0;
    disp([' ']);
    disp(['Which number of zones? [',num2str(Results_PS(1).Zones),'[-]-',num2str(Results_PS(Number_Echantillons).Zones),'[-]]']);
    disp([' ']);
    Zones=input('\n');
    j=(Zones-Zones_Lower)*(Number_Echantillons-1)/(Zones_Higher-Zones_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer number of zones from the echantillons: [',num2str(Results_PS(j).Zones),'[m]']);
    disp([' ']);
       imagesc(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==8
      i=0;
    disp([' ']);
    disp(['Which number of zones? [',num2str(Results_PS(1).Zones),'[-]-',num2str(Results_PS(Number_Echantillons).Zones),'[-]]']);
    disp([' ']);
    Zones=input('\n');
    j=(Zones-Zones_Lower)*(Number_Echantillons-1)/(Zones_Higher-Zones_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer number of zones from the echantillons: [',num2str(Results_PS(j).Zones),'[m]']);
    disp([' ']);
        surf(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==9
      i=0;
    disp([' ']);
    disp(['Which number of zones? [',num2str(Results_PS(1).Zones),'[-]-',num2str(Results_PS(Number_Echantillons).Zones),'[-]]']);
    disp([' ']);
    Zones=input('\n');
    j=(Zones-Zones_Lower)*(Number_Echantillons-1)/(Zones_Higher-Zones_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
```

```
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer number of zones from the echantillons: [',num2str(Results_PS(j).Zones),'[m]
            ']);
        disp([' ']);
            surf(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
     f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);
        elseif Display==10
          i=0;
        disp([' ']);
        disp(['Which number of zones? [',num2str(Results_PS(1).Zones),'[-]-',num2str(Results_PS(
            Number_Echantillons).Zones),'[-]]']);
        disp([' ']);
        Zones=input('\n');
        j=(Zones-Zones_Lower)*(Number_Echantillons-1)/(Zones_Higher-Zones_Lower)+1;
        if abs(ceil(j)-j)>abs(floor(j)-j)
            j=floor(j);
        else
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer number of zones from the echantillons: [',num2str(Results_PS(j).Zones),'[m]
            ']);
        disp([' ']);
        figure
            plot(Results_PS(j).x2(2:end),Results_PS(j).I2(Results_PS(j).M/2,1:(end-1))/Results_PS(
                j).Total_Power1);
xlabel('x (m)','latex');ylabel('Irradiance','latex');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
     f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);
        elseif Display==11
          i=0;
        disp([' ']);
        disp(['Which number of zones? [',num2str(Results_PS(1).Zones),'[-]-',num2str(Results_PS(
            Number_Echantillons).Zones),'[-]]']);
        disp([' ']);
        Zones=input('\n');
        j=(Zones-Zones_Lower)*(Number_Echantillons-1)/(Zones_Higher-Zones_Lower)+1;
        if abs(ceil(j)-j)>abs(floor(j)-j)
            j=floor(j);
        else
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer number of zones from the echantillons: [',num2str(Results_PS(j).Zones),'[m]
            ']);
        disp([' ']);
        figure
 plot(Results_PS(j).Encircle_Power(1,:),Results_PS(j).Encircle_Power(2,:), 'LineWidth',1,'
     Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
     f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);
        elseif Display==12
            i=0;
            for i=1:Number_Echantillons
                x(i)=Results_PS(i).Zones;
                y(i)=Results_PS(i).Radius_First_Disk*10^6;
                Airy_Radius(i)=1.22*Results_PS(i).Lambda*Results_PS(i).z/Results_PS(i).Diameter
```

```
                *10^6;
        end
        figure
        hold on
plot(x,Airy_Radius, 'LineWidth',1,'Color','r');
legend({'Airy radius'},'fontsize',11);
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Radius of the first disk [micron]','interpreter','latex','fontsize',16);
xlabel('Number of zones [-]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
     f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);
hold off
    elseif Display==13
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).Zones;
            y(i)=Results_PS(i).Smallest_Holes*10^6;
                end

                figure
        hold on
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Diameter of the smallest holes [Microns]','interpreter','latex','fontsize',16);
xlabel('Number of zones [-]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
     f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);
hold off
    else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);

    end

elseif  Optimization_Parameter==1 && DiameterVSZones==1

    if Display==1
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Diameter;
            y(i)=Results_PS(i).Numb_Total_Holes;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of holes [-]','interpreter','latex','fontsize',16);
xlabel('Diameter [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
     f=',num2str(Results_PS(1j).f),'m'],'Interpreter','latex', 'FontSize', 14);

    elseif Display==2
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).Diameter;
            y(i)=Results_PS(i).Counter_Zones;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of zones [-]','interpreter','latex','fontsize',16);
xlabel('Diameter [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm -
     f=',num2str(Results_PS(1j).f),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==3
   i=0;
    for i=1:Number_Echantillons
            x(i)=Results_PS(i).Diameter;
            y(i)=Results_PS(i).Total_Power1;
    end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power [Watts]','interpreter','latex','fontsize',16);
```

```
xlabel('Diameter [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1j).f),'m'],'Interpreter','latex', 'FontSize', 14);
     elseif Display==4
   i=0;
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Diameter;
            y(i)=Results_PS(i).First_Disk_Power*100;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power in the first disk [\%]','interpreter','latex','fontsize',16);
xlabel('Diameter of the Photon Sieve [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1j).f),'m'],'Interpreter','latex', 'FontSize', 14);
     elseif Display==5
   i=0;
         for i=1:Number_Echantillons
            x(i)=Results_PS(i).Diameter;
            y(i)=Results_PS(i).Surface_Ratio;
         end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Surface ratio [-]','interpreter','latex','fontsize',16);
xlabel('Diameter [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1j).f),'m'],'Interpreter','latex', 'FontSize', 14);

    elseif Display==6
      i=0;
    disp([' ']);
    disp(['Which diameter? [',num2str(Results_PS(1).Diameter),'[m]-',num2str(Results_PS(Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_PS(j).Diameter),'[m]']);
    disp([' ']);

      imagesc(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==7
        i=0;
            disp([' ']);
    disp(['Which diameter? [',num2str(Results_PS(1).Diameter),'[m]-',num2str(Results_PS(Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_PS(j).Diameter),'[m]']);
    disp([' ']);
       imagesc(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
```

```
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
    f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);

    elseif Display==8
        i=0;
            disp([' ']);
    disp(['Which diameter? [',num2str(Results_PS(1).Diameter),'[m]-',num2str(Results_PS(
        Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons -1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_PS(j).Diameter),'[m]']);
    disp([' ']);
        surf(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).
    f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize'
    , 14);

    elseif Display==9
        i=0;
            disp([' ']);
    disp(['Which diameter? [',num2str(Results_PS(1).Diameter),'[m]-',num2str(Results_PS(
        Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons -1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_PS(j).Diameter),'[m]']);
    disp([' ']);
        surf(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_PS(j).z),'m [Observation plan]'], 'FontSize', 14);
    elseif Display==10
        i=0;
          disp([' ']);
    disp(['Which diameter? [',num2str(Results_PS(1).Diameter),'[m]-',num2str(Results_PS(
        Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons -1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
```

```
    disp(['Closer diameter from the echantillons: [',num2str(Results_PS(j).Diameter),'[m]']);
    disp([' ']);
    figure
        plot(Results_PS(j).x2(2:end),Results_PS(j).I2(Results_PS(j).M/2,1:(end-1))/Results_PS(j).Total_Power1);
xlabel('x (m)');ylabel('Irradiance');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);


    elseif Display==11
        i=0;
                disp([' ']);
    disp(['Which diameter? [',num2str(Results_PS(1).Diameter),'[m]-',num2str(Results_PS(Number_Echantillons).Diameter),'[m]]']);
    disp([' ']);
    Diameter=input('\n');
    j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer diameter from the echantillons: [',num2str(Results_PS(j).Diameter),'[m]']);
    disp([' ']);
    figure
 plot(Results_PS(j).Encircle_Power(1,:),Results_PS(j).Encircle_Power(2,:), 'LineWidth',1,'Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

        elseif Display==12
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Diameter;
            y(i)=Results_PS(i).Radius_First_Disk*10^6;
            Airy_Radius(i)=1.22*Results_PS(i).Lambda*Results_PS(i).z/Results_PS(i).Diameter*10^6;
        end
        figure
        hold on
plot(x,Airy_Radius, 'LineWidth',1,'Color','r');
legend({'Airy radius'},'fontsize',11);
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Radius of the first peak [micron]','interpreter','latex','fontsize',16);
xlabel('Diameter of the Photon Sieve [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);

hold off
    elseif Display==13
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).Diameter;
            y(i)=Results_PS(i).Smallest_Holes*10^6;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Diameter of the smallest holes [Microns]','interpreter','latex','fontsize',16);
xlabel('Diameter [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m'],'Interpreter','latex', 'FontSize', 14);


    else
```

```
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);

    end

    elseif  Optimization_Parameter==2

    if Display==1
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).z;
            y(i)=Results_PS(i).Numb_Total_Holes;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of holes [-]','interpreter','latex','fontsize',16);
xlabel('Distance to observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==2
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).z;
            y(i)=Results_PS(i).Counter_Zones;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of zones [-]','interpreter','latex','fontsize',16);
xlabel('Distance to observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==3
   i=0;
    for i=1:Number_Echantillons
            x(i)=Results_PS(i).z;
            y(i)=Results_PS(i).Total_Power1;
    end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power [Watts]','interpreter','latex','fontsize',16);
xlabel('Distance to observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
     elseif Display==4
   i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).z;
            y(i)=Results_PS(i).First_Disk_Power*100;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power in the first disk [\%]','interpreter','latex','fontsize',16);
xlabel('Distance to observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
     elseif Display==5
   i=0;
         for i=1:Number_Echantillons
            x(i)=Results_PS(i).z;
            y(i)=Results_PS(i).Surface_Ratio;
         end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Surface ratio [-]','interpreter','latex','fontsize',16);
xlabel('Distance to observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==6
```

```
          i=0;
      disp([' ']);
      disp(['Which distance to the observation plan? [',num2str(Results_PS(1).z),'[m]-',num2str(Results_PS(Number_Echantillons).z),'[m]]']);
      disp([' ']);
      z=input('\n');
      j=(z-Lower_z)*(Number_Echantillons-1)/(Higher_z-Lower_z)+1;
      if abs(ceil(j)-j)>abs(floor(j)-j)
          j=floor(j);
      else
          j=ceil(j);
      end
          disp([' ']);
      disp(['Closer distance from the echantillons: [',num2str(Results_PS(j).z),'[m]]']);
      disp([' ']);

         imagesc(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

      elseif Display==7
          i=0;
      disp([' ']);
      disp(['Which distance to the observation plan? [',num2str(Results_PS(1).z),'[m]-',num2str(Results_PS(Number_Echantillons).z),'[m]]']);
      disp([' ']);
      z=input('\n');
      j=(z-Lower_z)*(Number_Echantillons-1)/(Higher_z-Lower_z)+1;
      if abs(ceil(j)-j)>abs(floor(j)-j)
          j=floor(j);
      else
          j=ceil(j);
      end
          disp([' ']);
      disp(['Closer distance from the echantillons: [',num2str(Results_PS(j).z),'[m]]']);
      disp([' ']);
         imagesc(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
      elseif Display==8
          i=0;
              disp([' ']);
      disp(['Which diameter? [',num2str(Results_PS(1).Diameter),'[m]-',num2str(Results_PS(Number_Echantillons).Diameter),'[m]]']);
      disp([' ']);
      Diameter=input('\n');
      j=(Diameter-Diameter_Lower)*(Number_Echantillons-1)/(Diameter_Higher-Diameter_Lower)+1;
      if abs(ceil(j)-j)>abs(floor(j)-j)
          j=floor(j);
      else
          j=ceil(j);
      end
          disp([' ']);
      disp(['Closer diameter from the echantillons: [',num2str(Results_PS(j).Diameter),'[m]]']);
      disp([' ']);
          surf(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).
```

```
    f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize'
    , 14);

    elseif Display==9
      i=0;
    disp([' ']);
    disp(['Which distance to the observation plan? [',num2str(Results_PS(1).z),'[m]-',num2str(
        Results_PS(Number_Echantillons).z),'[m]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons -1)/(Higher_z-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance from the echantillons: [',num2str(Results_PS(j).z),'[m]]']);
    disp([' ']);
        surf(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
    f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);

    elseif Display==10
      i=0;
    disp([' ']);
    disp(['Which distance to the observation plan? [',num2str(Results_PS(1).z),'[%]-',num2str(
        Results_PS(Number_Echantillons).z),'[%]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons -1)/(Higher_z-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance from the echantillons: [',num2str(Results_PS(j).z),'[%]]']);
    disp([' ']);
    figure
    plot(Results_PS(j).x2(2:end),Results_PS(j).I2(Results_PS(j).M/2,1:(end-1))/Results_PS(j).
        Total_Power1);
xlabel('x (m)');ylabel('Irradiance');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
    f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);

    elseif Display==11
        i=0;
                disp([' ']);
    disp(['Which distance to the observation plan? [',num2str(Results_PS(1).z),'[%]-',num2str(
        Results_PS(Number_Echantillons).z),'[%]]']);
    disp([' ']);
    z=input('\n');
    j=(z-Lower_z)*(Number_Echantillons -1)/(Higher_z-Lower_z)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer distance from the echantillons: [',num2str(Results_PS(j).z),'[%]]']);
    disp([' ']);
    figure
```

```
 plot(Results_PS(j).Encircle_Power(1,:),Results_PS(j).Encircle_Power(2,:), 'LineWidth',1,'Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

        elseif Display==12
         i=0;
         for i=1:Number_Echantillons
             x(i)=Results_PS(i).z;
             y(i)=Results_PS(i).Radius_First_Disk*10^6;
             Airy_Radius(i)=1.22*Results_PS(i).Lambda*Results_PS(i).z/Results_PS(i).Diameter*10^6;
         end
         figure
         hold on

plot(x,Airy_Radius, 'LineWidth',1,'Color','r');
legend({'Airy radius'},'fontsize',11);
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Radius of the first peak [micron]','interpreter','latex','fontsize',16);
xlabel('Distance to observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

 hold off
    elseif Display==13
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).z;
            y(i)=Results_PS(i).Smallest_Holes*10^6;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Diameter of the smallest holes [Microns]','interpreter','latex','fontsize',16);
xlabel('Distance to the observation plan [m]','interpreter','latex','fontsize',16);
title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);

    end

        elseif  Optimization_Parameter==3
    if Display==1
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Density;
            y(i)=Results_PS(i).Numb_Total_Holes;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of holes [-]','interpreter','latex','fontsize',16);
xlabel('Density [\%]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==2
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).Density;
            y(i)=Results_PS(i).Counter_Zones;
                end
        figure
```

```
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of zones [-]','interpreter','latex','fontsize',16);
xlabel('Density [\%]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==3
   i=0;
    for i=1:Number_Echantillons
            x(i)=Results_PS(i).Density;
            y(i)=Results_PS(i).Total_Power1;
    end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power [Watts]','interpreter','latex','fontsize',16);
xlabel('Density [\%]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

     elseif Display==4
   i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Density;
            y(i)=Results_PS(i).First_Disk_Power*100;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power in the first disk [\%]','interpreter','latex','fontsize',16);
xlabel('Density [\%]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

     elseif Display==5
   i=0;
         for i=1:Number_Echantillons
            x(i)=Results_PS(i).Density;
            y(i)=Results_PS(i).Surface_Ratio;
         end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Surface ratio [-]','interpreter','latex','fontsize',16);
xlabel('Density [\%]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);


    elseif Display==6
      i=0;
    disp([' ']);
    disp(['Which density? [',num2str(Results_PS(1).Density),'[%]-',num2str(Results_PS(Number_Echantillons).Density),'[%]]']);
    disp([' ']);
    Density=input('\n');
    j=(Density-Lower_Density)*(Number_Echantillons-1)/(Higher_Density-Lower_Density)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer density from the echantillons: [',num2str(Results_PS(j).Density),'[%]]']);
    disp([' ']);

      imagesc(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
axis square; axis xy;
colormap('gray');
```

```
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).
    f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize'
    , 14);

    elseif Display==7
      i=0;
    disp([' ']);
    disp(['Which density? [',num2str(Results_PS(1).Density),'[%]-',num2str(Results_PS(
        Number_Echantillons).Density),'[%]]']);
    disp([' ']);
    Density=input('\n');
    j=(Density-Lower_Density)*(Number_Echantillons -1)/(Higher_Density-Lower_Density)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer density from the echantillons: [',num2str(Results_PS(j).Density),'[%]]']);
    disp([' ']);
       imagesc(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
     f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);

    elseif Display==8
        i=0;
      i=0;
    disp([' ']);
    disp(['Which density? [',num2str(Results_PS(1).Density),'[%]-',num2str(Results_PS(
        Number_Echantillons).Density),'[%]]']);
    disp([' ']);
    Density=input('\n');
    j=(Density-Lower_Density)*(Number_Echantillons -1)/(Higher_Density-Lower_Density)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer density from the echantillons: [',num2str(Results_PS(j).Density),'[%]]']);
    disp([' ']);
        surf(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).
    f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize'
    , 14);

    elseif Display==9
      i=0;
    disp([' ']);
    disp(['Which density? [',num2str(Results_PS(1).Density),'[%]-',num2str(Results_PS(
        Number_Echantillons).Density),'[%]]']);
    disp([' ']);
    Density=input('\n');
    j=(Density-Lower_Density)*(Number_Echantillons -1)/(Higher_Density-Lower_Density)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
```

```
    disp(['Closer density from the echantillons: [',num2str(Results_PS(j).Density),'[%]]']);
    disp([' ']);
        surf(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==10
      i=0;
    disp([' ']);
    disp(['Which density? [',num2str(Results_PS(1).Density),'[%]-',num2str(Results_PS(Number_Echantillons).Density),'[%]]']);
    disp([' ']);
    Density=input('\n');
    j=(Density-Lower_Density)*(Number_Echantillons-1)/(Higher_Density-Lower_Density)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer density from the echantillons: [',num2str(Results_PS(j).Density),'[%]]']);
    disp([' ']);
    figure
       plot(Results_PS(j).x2(2:end),Results_PS(j).I2(Results_PS(j).M/2,1:(end-1))/Results_PS(j).Total_Power1);
xlabel('x (m)');ylabel('Irradiance');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==11
        i=0;
    disp([' ']);
    disp(['Which density? [',num2str(Results_PS(1).Density),'[%]-',num2str(Results_PS(Number_Echantillons).Density),'[%]]']);
    disp([' ']);
    Density=input('\n');
    j=(Density-Lower_Density)*(Number_Echantillons-1)/(Higher_Density-Lower_Density)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer density from the echantillons: [',num2str(Results_PS(j).Density),'[%]]']);
    disp([' ']);
    figure
  plot(Results_PS(j).Encircle_Power(1,:),Results_PS(j).Encircle_Power(2,:), 'LineWidth',1,'Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

       elseif Display==12
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Density;
            y(i)=Results_PS(i).Radius_First_Disk*10^6;
            Airy_Radius(i)=1.22*Results_PS(i).Lambda*Results_PS(i).z/Results_PS(i).Diameter*10^6;
        end
 figure
```

```
 hold on
plot(x,Airy_Radius, 'LineWidth',1,'Color','r');
legend({'Airy radius'},'fontsize',11);
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Radius of the first disk [micron]','interpreter','latex','fontsize',16);
xlabel('Density [\%]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
hold off

    elseif Display==13
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).Density;
            y(i)=Results_PS(i).Smallest_Holes*10^6;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Diameter of the smallest holes [Microns]','interpreter','latex','fontsize',16);
xlabel('Density [\%]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - $\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);

    end


            elseif  Optimization_Parameter==4
                    if Display==1
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Lambda*10^9;
            y(i)=Results_PS(i).Numb_Total_Holes;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of holes [-]','interpreter','latex','fontsize',16);
xlabel('Lambda [nm]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==2
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).Lambda*10^9;
            y(i)=Results_PS(i).Counter_Zones;
                end
                figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of zones [-]','interpreter','latex','fontsize',16);
xlabel('Lambda [nm]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==3
   i=0;
    for i=1:Number_Echantillons
            x(i)=Results_PS(i).Lambda*10^9;
            y(i)=Results_PS(i).Total_Power1;
    end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power [Watts]','interpreter','latex','fontsize',16);
xlabel('Lambda [nm]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
```

```
     elseif Display==4
   i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Lambda*10^9;
            y(i)=Results_PS(i).First_Disk_Power*100;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power in the first disk [\%]','interpreter','latex','fontsize',16);
xlabel('Lambda [nm]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str
    (Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
     elseif Display==5
   i=0;
         for i=1:Number_Echantillons
            x(i)=Results_PS(i).Lambda*10^9;
            y(i)=Results_PS(i).Surface_Ratio;
         end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Surface ratio [-]','interpreter','latex','fontsize',16);
xlabel('Lambda [nm]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str
    (Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);


    elseif Display==6
      i=0;
    disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_PS(1).Lambda),'[nm]-',num2str(10^9*
        Results_PS(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons -1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(10^9*Results_PS(j).Lambda),'[nm
        ]]']);
    disp([' ']);

      imagesc(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).
    f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize'
    , 14);

    elseif Display==7
      i=0;
    disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_PS(1).Lambda),'[nm]-',num2str(10^9*
        Results_PS(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons -1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(10^9*Results_PS(j).Lambda),'[nm
```

```
        ]]']);
    disp([' ']);
        imagesc(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==8
        i=0;
    disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_PS(1).Lambda),'[nm]-',num2str(10^9*Results_PS(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons-1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(10^9*Results_PS(j).Lambda),'[nm]]']);
    disp([' ']);
        surf(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==9
      i=0;
    disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_PS(1).Lambda),'[nm]-',num2str(10^9*Results_PS(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons-1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(10^9*Results_PS(j).Lambda),'[nm]]']);
    disp([' ']);
        surf(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==10
      i=0;
    disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_PS(1).Lambda),'[nm]-',num2str(10^9*
```

```
        Results_PS(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons -1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(10^9*Results_PS(j).Lambda),'[nm
        ]]']);
    disp([' ']);
    figure
      plot(Results_PS(j).x2(2:end),Results_PS(j).I2(Results_PS(j).M/2,1:(end-1))/Results_PS(j)
          .Total_Power1);
xlabel('x (m)');ylabel('Irradiance');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
     f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);

    elseif Display==11
      i=0;
    disp([' ']);
    disp(['Which wavelength? [',num2str(10^9*Results_PS(1).Lambda),'[nm]-',num2str(10^9*
        Results_PS(Number_Echantillons).Lambda),'[nm]]']);
    disp([' ']);
    Lambda=input('\n');
    Lambda=Lambda/10^9;
    j=(Lambda-Lower_Lambda)*(Number_Echantillons -1)/(Higher_Lambda-Lower_Lambda)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer wavelength from the echantillons: [',num2str(10^9*Results_PS(j).Lambda),'[nm
        ]]']);
    disp([' ']);
    figure
  plot(Results_PS(j).Encircle_Power(1,:),Results_PS(j).Encircle_Power(2,:), 'LineWidth',1,'
      Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
     f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
    Interpreter','latex', 'FontSize', 14);

        elseif Display==12
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).Lambda*10^9;
            y(i)=Results_PS(i).Radius_First_Disk*10^6;
            Airy_Radius(i)=1.22*Results_PS(i).Lambda*Results_PS(i).z/Results_PS(i).Diameter
                *10^6;
        end

        figure
        hold on
plot(x,Airy_Radius, 'LineWidth',1,'Color','r');
legend({'Airy radius'},'fontsize',11);
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Radius of the first disk [micron]','interpreter','latex','fontsize',16);
xlabel('Lambda [nm]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str
    (Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

hold off
    elseif Display==13
```

```
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).Lambda*10^9;
            y(i)=Results_PS(i).Smallest_Holes*10^6;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Diameter of the smallest holes [Microns]','interpreter','latex','fontsize',16);
xlabel('Lambda[nm]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS(1).z),'m - f=',num2str(Results_PS(1).f),'m - ','Zones=',num2str
    (Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);


    else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);
      end
                elseif Optimization_Parameter==5
                        if Display==1
        i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).f;
            y(i)=Results_PS(i).Numb_Total_Holes;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of holes [-]','interpreter','latex','fontsize',16);
xlabel('Focal distance [m]',interpreter','latex','fontsize',16);
title([ '$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - ','Zones=',num2str(Results_PS(1)
    .Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==2
        i=0;
                for i=1:Number_Echantillons
            x(i)=Results_PS(i).f;
            y(i)=Results_PS(i).Counter_Zones;
                end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Number of zones [-]','interpreter','latex','fontsize',16);
xlabel('Focal distance [m]','interpreter','latex','fontsize',16);
title([ '$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - ','Zones=',num2str(Results_PS(1)
    .Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==3
    i=0;
    for i=1:Number_Echantillons
            x(i)=Results_PS(i).f;
            y(i)=Results_PS(i).Total_Power1;
    end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power [Watts]','interpreter','latex','fontsize',16);
xlabel('Focal distance [m]','interpreter','latex','fontsize',16);
title([ '$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - ','Zones=',num2str(Results_PS(1)
    .Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
     elseif Display==4
    i=0;
        for i=1:Number_Echantillons
            x(i)=Results_PS(i).f;
            y(i)=Results_PS(i).First_Disk_Power*100;
        end
        figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Power in the first disk [\%]','interpreter','latex','fontsize',16);
xlabel('Focal distance [m]','interpreter','latex','fontsize',16);
title([ '$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - ','Zones=',num2str(Results_PS(1)
    .Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
     elseif Display==5
    i=0;
```

```
				for i=1:Number_Echantillons
				    x(i)=Results_PS(i).f;
				    y(i)=Results_PS(i).Surface_Ratio;
				end
			   figure
plot(x,y, 'LineWidth',1,'Color','b');
ylabel('Surface ratio [-]','interpreter','latex','fontsize',16);
xlabel('Focal distance [m]','interpreter','latex','fontsize',16);
title([ '$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - ','Zones=',num2str(Results_PS(1).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==6
      i=0;
    disp([' ']);
    disp(['Which focal distance? [',num2str(Results_PS(1).f),'[m]-',num2str(Results_PS(Number_Echantillons).f),'[m]]']);
    disp([' ']);
    f=input('\n');
    j=(f-Lower_f)*(Number_Echantillons-1)/(Higher_f-Lower_f)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer focal distance from the echantillons: [',num2str(Results_PS(j).f),'[m]]']);
    disp([' ']);

      imagesc(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==7
      i=0;
    disp([' ']);
    disp(['Which focal distance? [',num2str(Results_PS(1).f),'[m]-',num2str(Results_PS(Number_Echantillons).f),'[m]]']);
    disp([' ']);
    f=input('\n');
    j=(f-Lower_f)*(Number_Echantillons-1)/(Higher_f-Lower_f)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
        j=ceil(j);
    end
        disp([' ']);
    disp(['Closer focal distance from the echantillons: [',num2str(Results_PS(j).f),'[m]]']);
    disp([' ']);
       imagesc(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

    elseif Display==8
        i=0;
    disp([' ']);
    disp(['Which focal distance? [',num2str(Results_PS(1).f),'[m]-',num2str(Results_PS(Number_Echantillons).f),'[m]]']);
    disp([' ']);
    f=input('\n');
    j=(f-Lower_f)*(Number_Echantillons-1)/(Higher_f-Lower_f)+1;
    if abs(ceil(j)-j)>abs(floor(j)-j)
        j=floor(j);
    else
```

```
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer focal distance from the echantillons: [',num2str(Results_PS(j).f),'[m]]']);
        disp([' ']);
            surf(Results_PS(j).x,Results_PS(j).y,nthroot(Results_PS(j).I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

        elseif Display==9
          i=0;
        disp([' ']);
        disp(['Which focal distance? [',num2str(Results_PS(1).f),'[m]-',num2str(Results_PS(Number_Echantillons).f),'[m]]']);
        disp([' ']);
        f=input('\n');
        j=(f-Lower_f)*(Number_Echantillons-1)/(Higher_f-Lower_f)+1;
        if abs(ceil(j)-j)>abs(floor(j)-j)
            j=floor(j);
        else
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer focal distance from the echantillons: [',num2str(Results_PS(j).f),'[m]]']);
        disp([' ']);
            surf(Results_PS(j).x2,Results_PS(j).y2,nthroot(Results_PS(j).I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

        elseif Display==10
          i=0;
        disp([' ']);
        disp(['Which focal distance? [',num2str(Results_PS(1).f),'[m]-',num2str(Results_PS(Number_Echantillons).f),'[m]]']);
        disp([' ']);
        f=input('\n');
        j=(f-Lower_f)*(Number_Echantillons-1)/(Higher_f-Lower_f)+1;
        if abs(ceil(j)-j)>abs(floor(j)-j)
            j=floor(j);
        else
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer focal distance from the echantillons: [',num2str(Results_PS(j).f),'[m]]']);
        disp([' ']);
        figure
        plot(Results_PS(j).x2(2:end),Results_PS(j).I2(Results_PS(j).M/2,1:(end-1))/Results_PS(j).Total_Power1);
xlabel('x (m)');ylabel('Irradiance');
title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm - f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'Interpreter','latex', 'FontSize', 14);

        elseif Display==11
          i=0;
        disp([' ']);
        disp(['Which focal distance? [',num2str(Results_PS(1).f),'[m]-',num2str(Results_PS(Number_Echantillons).f),'[m]]']);
```

```
        disp([' ']);
        f=input('\n');
        j=(f-Lower_f)*(Number_Echantillons -1)/(Higher_f-Lower_f)+1;
        if abs(ceil(j)-j)>abs(floor(j)-j)
            j=floor(j);
        else
            j=ceil(j);
        end
            disp([' ']);
        disp(['Closer focal distance from the echantillons: [',num2str(Results_PS(j).f),'[m]]']);
        disp([' ']);
        figure
      plot(Results_PS(j).Encircle_Power(1,:),Results_PS(j).Encircle_Power(2,:), 'LineWidth',1,'
          Color','b');
    ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
    xlabel('Radius [m]','interpreter','latex','fontsize',16);
    title(['z=',num2str(Results_PS(j).z),'m - $\lambda=$',num2str(Results_PS(j).Lambda*10^9),'nm -
         f=',num2str(Results_PS(j).f),'m - ','Zones=',num2str(Results_PS(j).Counter_Zones)],'
         Interpreter','latex', 'FontSize', 14);
            elseif Display==12
            i=0;
            for i=1:Number_Echantillons
                x(i)=Results_PS(i).f;
                y(i)=Results_PS(i).Radius_First_Disk*10^6;
                Airy_Radius(i)=1.22*Results_PS(i).Lambda*Results_PS(i).z/Results_PS(i).Diameter
                    *10^6;
            end
      figure
      hold on
    plot(x,Airy_Radius, 'LineWidth',1,'Color','r');
    legend({'Airy radius'},'fontsize',11);
    plot(x,y, 'LineWidth',1,'Color','b');
    ylabel('Radius of the first disk [micron]','interpreter','latex','fontsize',16);
    xlabel('Focal distance [m]','interpreter','latex','fontsize',16);
    title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - Zones=',num2str(Results_PS(1)).
        Counter_Zones],'Interpreter','latex', 'FontSize', 14);

    hold off
        elseif Display==13
            i=0;
                    for i=1:Number_Echantillons
                x(i)=Results_PS(i).f;
                y(i)=Results_PS(i).Smallest_Holes*10^6;
            end
    plot(x,y, 'LineWidth',1,'Color','b');
    ylabel('Diameter of the smallest holes [Microns]','interpreter','latex','fontsize',16);
    xlabel('Focal distance [m]','interpreter','latex','fontsize',16);
    title(['$\lambda=$',num2str(Results_PS(1).Lambda*10^9),'nm - Zones=',num2str(Results_PS(1)).
        Counter_Zones],'Interpreter','latex', 'FontSize', 14);
        else
        disp([' ']);
        disp(['Error in the entry']);
        disp([' ']);

        end

    end
        end
```

1.9 Display_Results_Single_C.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Script used for display the results of the Circle (Single case)
i=1;
while i==1
    disp([' ']);
    disp(['Enter: Error with the array radius (1) - Total power (2) - Encircled power (3)']);
    disp([' Irrandiance in entry plan (2D) (4) - Irrandiance in observation plan (2D) (5)']);
```

```
    disp(['  Irrandiance in entry plan (3D) (6) - Irrandiance in observation plan (3D) (7)']);
    disp(['                    Profile irrandiance in observation plan (8)']);
    disp([' ']);
    Display=input('\n');

    if Display==1
        i=0;
 disp([' ']);
disp([num2str(Results_C.Error_Airy),' [%]']);
 disp([' ']);
    elseif Display==2
        i=0;
         disp([' ']);
disp([num2str(Results_C.Total_Power1),' [Watts]']);
 disp([' ']);
    elseif Display==3
   i=0;
    disp([' ']);
disp(['Purcentage of the total power in the "first disk": ',num2str(Results_C.First_Disk_Power*100),' [%]']);
 disp([' ']);
figure
 plot(Results_C.Encircle_Power(1,:),Results_C.Encircle_Power(2,:), 'LineWidth',1,'Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_C.z),'m - $\lambda=$',num2str(Results_C.Lambda*10^9),'nm - D=',num2str(Results_C.Diameter),'m'],'Interpreter','latex', 'FontSize', 14);

    elseif Display==4
      i=0;
      figure
      imagesc(Results_C.x,Results_C.y,nthroot(Results_C.I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_C.Lambda*10^9),'nm - D=',num2str(Results_C.Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==5
        i=0;
        figure
      imagesc(Results_C.x2,Results_C.y2,nthroot(Results_C.I2,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['\quad z=',num2str(Results_C.z),'m - $\lambda=$',num2str(Results_C.Lambda*10^9),'nm - D=',num2str(Results_C.Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==6
        i=0;
        figure
        surf(Results_C.x,Results_C.y,nthroot(Results_C.I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_C.Lambda*10^9),'nm - D=',num2str(Results_C.Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==7
        i=0;
        figure
        surf(Results_C.x2,Results_C.y2,nthroot(Results_C.I2,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_C.z),'m - $\lambda=$',num2str(Results_C.Lambda*10^9),'nm - D=',num2str(Results_C.Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    elseif Display==8
        i=0;
```

```
        figure
        plot(Results_C.x2,Results_C.I2(Results_C.M/2+1,:)/Results_C.Total_Power2);
xlabel('x [m]');ylabel('Normalized irradiance [I(x)/I\_Total]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_C.z),'m - $\lambda=$',num2str(Results_C.Lambda*10^9),'nm - D=',num2str(Results_C.Diameter),'m'],'Interpreter','latex', 'FontSize', 14);
    else
    disp([' ']);
    disp(['Error in the entry']);
    disp([' ']);

    end


end
```

1.10 Display_Results_Single_PS.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Script used for display the results of the Photon Sieve(Single case)
i=1;
while i==1
    disp([' ']);
    disp(['Enter:     Number of holes (1) - Number of zones (2) - Total power (3) ']);
    disp(['     Encircled energy (4) - Surface ratio (5) - Radius of the "first peak" (6)']);
    disp([' Irrandiance in entry plan (2D) (7) - Irrandiance in observation plan (PSF-2D) (8)']);
    disp([ 'Irrandiance in entry plan (3D) (9) - Irrandiance in observation plan (PSF-3D) (10)']);
    disp(['                 Profile irrandiance in observation plan (PSF-Profile)(11) ']);
    disp(['                   Diameter of the smallest holes (12)']);
    disp([' ']);
    Display=input('\n');

    if Display==1
        i=0;
 disp([' ']);
disp([num2str(Results_PS.Numb_Total_Holes),' [Holes]']);
 disp([' ']);
    elseif Display==2
        i=0;
         disp([' ']);
disp([num2str(Results_PS.Counter_Zones),' [Zones]']);
 disp([' ']);
    elseif Display==3
   i=0;
    disp([' ']);
disp([num2str(Results_PS.Total_Power1),' [Watts]']);
 disp([' ']);
      elseif Display==4
   i=0;
   i=0;
    disp([' ']);
disp(['Percentage of the total power in the "first disk": ',num2str(Results_PS.First_Disk_Power*100),' [%]']);
 disp([' ']);
figure
 plot(Results_PS.Encircle_Power(1,:),Results_PS.Encircle_Power(2,:)*100, 'LineWidth',1,'Color','b');
ylabel('Encircle power [\%]','interpreter','latex','fontsize',16);
xlabel('Radius [m]','interpreter','latex','fontsize',16);
title(['z=',num2str(Results_PS.z),'m - $\lambda=$',num2str(Results_PS.Lambda*10^9),'nm - f=',num2str(Results_PS.f),'m - ','Zones=',num2str(Results_PS.Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
axis([0 0.011 0 120])
      elseif Display==5
   i=0;
    disp([' ']);
```

```
disp([num2str(Results_PS.Surface_Ratio),' [-]']);
 disp([' ']);

     elseif Display==6
   i=0;
    disp([' ']);
disp([num2str(Results_PS.Radius_First_Disk*10^6),' [micron]']);
 disp([' ']);
    elseif Display==7
      i=0;
      figure
      imagesc(Results_PS.x,Results_PS.y,nthroot(Results_PS.I1,3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS.Lambda*10^9),'nm - f=',num2str(Results_PS.f),'m - ','Zones=',num2str(Results_PS.Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==8
        i=0;
        figure
       imagesc(Results_PS.x2(2:end),Results_PS.y2(2:end),nthroot(Results_PS.I2(1:(end-1),1:(end-1)),3));
axis square; axis xy;
colormap('gray');
xlabel('x (m)');ylabel('y (m)');
title(['z=',num2str(Results_PS.z),'m - $\lambda=$',num2str(Results_PS.Lambda*10^9),'nm - f=',num2str(Results_PS.f),'m - ','Zones=',num2str(Results_PS.Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==9
        i=0;
        figure
        surf(Results_PS.x,Results_PS.y,nthroot(Results_PS.I1,3));
camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=0m - $\lambda=$',num2str(Results_PS.Lambda*10^9),'nm - f=',num2str(Results_PS.f),'m - ','Zones=',num2str(Results_PS.Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==10
        i=0;
        figure
        surf(Results_PS.x2,Results_PS.y2,nthroot(Results_PS.I2,3));

camlight left;
lighting phong
shading interp
ylabel('y (m)');
xlabel('x (m)');
title(['z=',num2str(Results_PS.z),'m - $\lambda=$',num2str(Results_PS.Lambda*10^9),'nm - f=',num2str(Results_PS.f),'m - ','Zones=',num2str(Results_PS.Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==11
        i=0;
        figure
        plot(Results_PS.x2(2:end),Results_PS.I2(Results_PS.M/2,1:(end-1))/Results_PS.Total_Power1);
xlabel('x (m)');ylabel('Normalized irradiance [I(x)/I_Total]','Interpreter','latex');
title(['z=',num2str(Results_PS.z),'m - $\lambda=$',num2str(Results_PS.Lambda*10^9),'nm - f=',num2str(Results_PS.f),'m - ','Zones=',num2str(Results_PS.Counter_Zones)],'Interpreter','latex', 'FontSize', 14);
    elseif Display==12
        i=0;
 disp([' ']);
disp([num2str(Results_PS.Smallest_Holes*10^6),' [Microns]']);
 disp([' ']);

    else
    disp([' ']);
    disp(['Error in the entry']);
```

```
        disp([' ']);

        end


end
```

1.11 Photon_Sieve_Generator.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to generate the Photon Sieve and the parameters associated

function[Smallest_Holes,Counter_Zones,Numb_Total_Holes,Surface_Ratio,Signal,L,x,y]=Photon_Sieve_Generator(Diameter,f,Density,Lambda,Speed,Single)

%% Compute the number of zones comprises in that diameter
i=1;
Counter_Zones=-1;
while i==1 %"While" to know the the number of zones comprise in that diameter
        Counter_Zones=Counter_Zones+2; %Solely the extern borders of the passing zones
      r=sqrt(Counter_Zones*Lambda*(f+Counter_Zones*Lambda/4)); %Width of the nth zone
      r_before=sqrt((Counter_Zones-1)*Lambda*(f+(Counter_Zones-1)*Lambda/4));
      w=r-r_before; %Size of the nth zone
      test=Diameter/2-(r-w/2+w/2);%The radius of the hole is taken into account

      if test<0 %"If" used to verify if the radius overide the diameter of the Photon Sieve
          i=0;
          Counter_Zones=Counter_Zones-2;
      end
end
if Counter_Zones<1 %In case of too small diameter
        disp([' ']);
        disp(['!!!!!!! Error in the entry, diameter too small for the Photon Sieve !!!!!!!']);
        disp([' ']);
end
%% Compute the total number of holes and the surface ratio
Numb_Total_Holes=0;
open_surface=0;
for i=1:2:Counter_Zones
        if i==1 %Special case for the central hole
        r=sqrt(i*Lambda*(f+i*Lambda/4));
        r_before=sqrt((i-1)*Lambda*(f+(i-1)*Lambda/4));
        w=r-r_before;
        Numb_Total_Holes=Numb_Total_Holes+1;
        open_surface=open_surface+pi*(w)^2;
        else
        r=sqrt(i*Lambda*(f+i*Lambda/4));
        r_before=sqrt((i-1)*Lambda*(f+(i-1)*Lambda/4));
        w=(r-r_before);

        add=floor((2*pi*(r-w/2))/(1*w)*Density/100); %times 1.53?
        Numb_Total_Holes=Numb_Total_Holes+add;
        open_surface=open_surface+add*pi*((w/2))^2;  %Verify the optimization of the size of the hole
        end
end
Surface_Ratio=open_surface/(pi*(Diameter/2)^2);

%% Generation of the entry signal (The Photon Sieve)
%Size of the array
L=Diameter*(1.1+0.2*2*(Speed-1));

%Diameter of the smallest holes
      r=sqrt(Counter_Zones*Lambda*(f+Counter_Zones*Lambda/4));
      r_before=sqrt((Counter_Zones-1)*Lambda*(f+(Counter_Zones-1)*Lambda/4));
      w=r-r_before;
      Smallest_Holes=w;
% Sizing the minimum pixel with respect to the smallest hole
dx_max=w/(1+5*(Speed-1));
```

```
M_min=L/dx_max;
M=ceil(M_min/1000)*1000;


dx=L/M;
x=-L/2:dx:L/2-dx;
y=x;
[X,Y]=meshgrid(x,y);
Signal=zeros(max(size(x)),max(size(x)));

%Display the number of pixels and its size!
if Single==1
disp([' ']);
disp(['Size of a pixel: ',num2str(10^6*dx),'x',num2str(10^6*dx),' [micron x micron]']);
disp([' ']);
disp(['Number of pixels: ',num2str(M),'x',num2str(M)]);
disp([' ']);
end

%Computation of the signal
counter_holes=0;
for i=1:2:(Counter_Zones)
     n=Counter_Zones-(i-1);
    if n==1 %Special case for the center
        r=sqrt(Lambda*(f+Lambda/4));
     Signal_Add=abs(sqrt((X).^2+(Y).^2)<=(1*r);
     Signal=Signal+Signal_Add;

    else %External cases
 r=sqrt(n*Lambda*(f+n*Lambda/4));
 r_before=sqrt((n-1)*Lambda*(f+(n-1)*Lambda/4));
 w=(r-r_before);
 numb_holes=floor((2*pi*(r-w/2))/w*Density/100);

 %Coordinate angles of the nodes
 angles=1:1:numb_holes;
 angles=angles/numb_holes*2*pi;
 [Holes,Size]=Holes_Generator(dx,w/2);

 for i=1:1:numb_holes %Add a hole at every point centered by the angle
     x_center=cos(angles(i))*(r-w/2);
     y_center=sin(angles(i))*(r-w/2);
     i_center=floor(cos(angles(i))*(r-w/2)/dx);
     j_center=floor(sin(angles(i))*(r-w/2)/dx);
     i_begin=ceil(M/2)+i_center-floor(Size/2);
     i_end=ceil(M/2)+i_center+ceil(Size/2)-1;
     j_begin=ceil(M/2)+j_center-floor(Size/2);
     j_end=ceil(M/2)+j_center+ceil(Size/2)-1;
     Signal(j_begin:j_end,i_begin:i_end) =Signal(j_begin:j_end,i_begin:i_end)+Holes(:,:);

 end

    end %End of the "if" condition
end %End of the generation loop

for i=1:1:M %In case of superposition of the hole, put the value upper than 2 to 1
    for j=1:1:M
        if Signal(j,i)>1
            Signal(j,i)=1;
        end
    end
end %End of the correction

end %End of the function
```

1.12 Circle_Generator.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to generate the Circle
```

```
function [Signal ,L,x,y]=Circle_Generator(Diameter ,Speed)

M=3000*Speed;
L=Diameter*10;
dx=L/M;
x=-L/2:dx:L/2-dx;
y=x;
[X,Y]=meshgrid(x,y);
Signal=abs(sqrt((X).^2+Y.^2))<=(Diameter/2);
end
```

1.13 Holes_Generator.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to generate the holes necessary at the Photon Sieve
% construction

function [Holes, Size]=Holes_Generator(dx,r)

L=dx*(ceil(2*r/dx));

x=-L/2:dx:L/2-dx;
y=x;
[X,Y]=meshgrid(x,y);
Holes=abs(sqrt((X).^2+Y.^2))<=(r);
Size=max(size(x));

end
```

1.14 Irradiance_Power.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to compute the power of the irradiance

function [Total_Power]=Irradiance_Power(I,L,M)

dx=L/M;
Total_Power=0;
for i=1:1:M
    for j=1:1:M
        Total_Power=Total_Power+I(j,i)*dx^2;
    end
end

end
```

1.15 Encircled_Energy.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to compute the encircle energy and the energy contain in

function [Encircle_Power, First_Disk_Power, Total_Power, Radius_First_Disk]=Encircled_Energy(I,L,M
    ,x,Power)

Radius_min=0;
Radius_inter=L/4;
Radius_max=L/2;

[X,Y]=meshgrid(x,x);
Encircle_Power=zeros(2,M/2);
power=0;
First_Circle_Right=1;
First_Circle_Left=1;
Radius_First_Disk=0;
```

```
I_inter=zeros(1,M);
I_inter(1)=I(M/2,end);
I_inter(2:end)=I(M/2,1:(end-1));
I(M/2,:)=I_inter;
for i=0:1:(M/2-2)
        f_2=(I(M/2,M/2+1+i+1)+I(M/2,M/2+1-i-1))/2;
        f_1=(I(M/2,M/2+1+i)+I(M/2,M/2+1-i))/2;
        r_2=X(M/2+1,M/2+1+i+1);
         r_1=X(M/2+1,M/2+1+i);
         S=(f_2-f_1)/(r_2-r_1);
         power=power+2*pi*((f_1-r_1*S)*r_2^2/2-(f_1-r_1*S)*r_1^2/2+S*r_2^3/3-S*r_1^3/3);
         Encircle_Power(1,i+2)=r_2;
         Encircle_Power(2,i+2)=power;

         if First_Circle_Right==1 && I(M/2,M/2+1+i+1)>I(M/2,M/2+1+i) && I(M/2,M/2+1+i)<(I(M/2,M
             /2+1+1)/100)%Compute the energy of the "First disk"
             First_Circle_Right=0;
             First_Disk_Power_Right=power;
             Radius_First_Disk_Right=X(M/2+1,M/2+1+i);
         end
         if First_Circle_Left==1 && I(M/2,M/2+1-i-1)>I(M/2,M/2+1-i) && I(M/2,M/2+1-i)<(I(M/2,M
             /2+1+1)/100)%Compute the energy of the "First disk"
             First_Circle_Left=0;
             First_Disk_Power_Left=power;
             Radius_First_Disk_Left=X(M/2+1,M/2+1-i);
         end

end
Radius_First_Disk=0;
if First_Circle_Left==0 && First_Circle_Right==0
Radius_First_Disk=(-Radius_First_Disk_Left+Radius_First_Disk_Right)/2;
end
Total_Power=Encircle_Power(2,end);
First_Disk_Power=0;
if First_Circle_Left==0 && First_Circle_Right==0
First_Disk_Power=(First_Disk_Power_Right+First_Disk_Power_Left)/2/Encircle_Power(2,end);
end
Encircle_Power(2,:)=Encircle_Power(2,:)/Power;



end
```

1.16 Time_Estimation.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Script used to estimate the simulation time

%Capacity of the computer: DELL-Home (4GB RAM)
% 8-bytes by case of a array
a=2.5364; %Coefficients of the regression
b=-10.6836;
c=6.6553;

%Capacity of the computer: CSL (64GB RAM)
% 8-bytes by case of a array
%a=0.165; %Coefficients of the regression
%b=-0.2374;
%c=0.4494;


Time=0;
if Simulation==1
 %Computation of M_Thousands
Counter_Zones=-1;
test=0;
while test>=0 %"While" to know the the number of zones comprise in that diameter
    Counter_Zones=Counter_Zones+2; %Solely the extern borders of the passing zones
```

```
    r=sqrt(Counter_Zones*Results_PS.Lambda*(Results_PS.f+Counter_Zones*Results_PS.Lambda/4)); % Width of the nth zone
    r_before=sqrt((Counter_Zones-1)*Results_PS.Lambda*(Results_PS.f+(Counter_Zones-1)*Results_PS.Lambda/4));
    w=r-r_before; %Size of the nth zone
    test=Results_PS.Diameter/2-(r-w/2+w/2);%The radius of the hole is taken into account (-(r-w/2+1.531*w/2 in case of higher radius holes)
end
    r=sqrt(Counter_Zones*Results_PS.Lambda*(Results_PS.f+Counter_Zones*Results_PS.Lambda/4));
    r_before=sqrt((Counter_Zones-1)*Results_PS.Lambda*(Results_PS.f+(Counter_Zones-1)*Results_PS.Lambda/4));
    w=r-r_before;
    dx_max=w/(1+5*(Speed-1));
    L=Results_PS.Diameter*(1.1+0.2*2*(Speed-1));
M_min=floor(L/dx_max);
M_min=L/dx_max;
M=ceil(M_min/1000)*1000;


      Time=a*(M/1000)^2+b*(M/1000)+c;
      Time=ceil(Time);

else
      for i=1:1:Number_Echantillons
          %Computation of M_Thousands
Counter_Zones=-1;
test=0;
if Optimization_Parameter==1  %If depending of the optimization
            Results_PS.f=f;
            Results_PS.Lambda=Lambda;
            if DiameterVSZones==2
                Results_PS.Zones=floor(Zones_Lower+(Zones_Higher-Zones_Lower)*(i-1)/(Number_Echantillons-1));
                Results_PS.Diameter=2*sqrt((Results_PS.Zones)*Lambda*(f+(Results_PS.Zones)*Lambda/4));
            else
            Results_PS.Diameter=Diameter_Lower+(Diameter_Higher-Diameter_Lower)*(i-1)/(Number_Echantillons-1);
            end

elseif Optimization_Parameter==2
            Results_PS.f=f;
            Results_PS.Lambda=Lambda;
            Results_PS.Diameter=Diameter;

    elseif Optimization_Parameter==3
            Results_PS.f=f;
            Results_PS.Lambda=Lambda;
            Results_PS.Diameter=Diameter;
        elseif Optimization_Parameter==4
            Results_PS.f=f;
            Results_PS.Lambda=Lower_Lambda+(Higher_Lambda-Lower_Lambda)*(i-1)/(Number_Echantillons-1);
                       if DiameterVSZones==1
            Results_PS.Diameter=Diameter;
            else
                Results_PS.Diameter=2*sqrt((Zones)*Results_PS.Lambda*(f+(Zones)*Results_PS.Lambda/4));
                       end
          elseif Optimization_Parameter==5
            Results_PS.f=Lower_f+(Higher_f-Lower_f)*(i-1)/(Number_Echantillons-1);
            Results_PS.Lambda=Lambda;
                                   if DiameterVSZones==1
            Results_PS.Diameter=Diameter;
            else
                Results_PS.Diameter=2*sqrt((Zones)*Results_PS.Lambda*(Results_PS.f+(Zones)*Results_PS.Lambda/4));
                                   end
end

```

```
while test>=0 %"While" to know the the number of zones comprise in that diameter
     Counter_Zones=Counter_Zones+2; %Solely the extern borders of the passing zones
    r=sqrt(Counter_Zones*Results_PS.Lambda*(Results_PS.f+Counter_Zones*Results_PS.Lambda/4)); %
        Width of the nth zone
    r_before=sqrt((Counter_Zones-1)*Results_PS.Lambda*(Results_PS.f+(Counter_Zones-1)*
        Results_PS.Lambda/4));
    w=r-r_before; %Size of the nth zone
    test=Results_PS.Diameter/2-(r-w/2+w/2);%The radius of the hole is taken into account (-(r-w
        /2+1.531*w/2 in case of higher radius holes)
end
    r=sqrt(Counter_Zones*Results_PS.Lambda*(Results_PS.f+Counter_Zones*Results_PS.Lambda/4));
    r_before=sqrt((Counter_Zones-1)*Results_PS.Lambda*(Results_PS.f+(Counter_Zones-1)*
        Results_PS.Lambda/4));
    w=r-r_before;
    dx_max=w/(1+5*(Speed-1));
    L=Results_PS.Diameter*(1.1+0.2*2*(Speed-1));
M_min=floor(L/dx_max);
M_min=L/dx_max;
M=ceil(M_min/1000)*1000;
M_Thousands=M/1000;

            Time=Time+a*(M/1000)^2+b*(M/1000)+c;

     end


end

Time=ceil(Time);
if Time<3600 && Time>60
     Time=ceil(Time/60);
      X=[' Time estimation of the simulation: ',num2str(Time),' minutes '];
elseif Time>3600
     Time=Time/3600;
     X=[' Time estimation of the simulation: ',num2str(Time),' hours '];
else
X=[' Time estimation of the simulation: ',num2str(Time),' seconds '];
end
waitbar(0.2,X)
```

1.17 Diameter_Zones_Max.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to compute the diameter maximum with respect to the RAM

function[Zones_Max,Diameter_Max]=Diameter_Zones_Max(Lambda,f,Speed)
%Capacity of the computer: DELL-Home (4GB RAM)
% 8-bytes by case of a array
RAM_Disponible=0.8*10^9; %[Bytes]
M_Max=floor(sqrt(RAM_Disponible/8));

%Capacity of the computer: CSL(64GB RAM)
% 8-bytes by case of a array
%RAM_Disponible=15*10^9; %[Bytes]
%M_Max=floor(sqrt(RAM_Disponible/8));

Counter_Zones=0;
M=0;
while M_Max>M %Loop until the number of zones generate a M higher than M_Max
    Counter_Zones=Counter_Zones+1;
    r=sqrt(Counter_Zones*Lambda*(f+Counter_Zones*Lambda/4));
    r_before=sqrt((Counter_Zones-1)*Lambda*(f+(Counter_Zones-1)*Lambda/4));
    w=r-r_before;
    dx_max=w/(1+5*(Speed-1));
    L=(1.1+0.2*2*(Speed-1))*2*sqrt(Counter_Zones*Lambda*(f+Counter_Zones*Lambda/4));
M_min=floor(L/dx_max);
M=ceil(M_min/1000)*1000;

end
```

```
Zones_Max=Counter_Zones -1;
Diameter_Max=2*sqrt(Zones_Max*Lambda*(f+Zones_Max*Lambda/4));

end
```

B Multilayer holes

2.1 Photon_Sieve_Generator_ML.m

```
%% Photon Sieve based telescope for interferometer CubeSat
% Adrien Mercenier 2017-2018
% Function used to generate the Photon Sieve with multilayer and the parameters associated

function[Smallest_Holes,Counter_Zones,Numb_Total_Holes,Surface_Ratio,Signal,L,x,y]=
    Photon_Sieve_Generator_ML(Diameter,f,Density,Lambda,Speed,Single)

Number_Layer=5;
%% Compute the number of zones comprises in that diameter
i=1;
Counter_Zones=-1;
while i==1 %"While" to know the the number of zones comprise in that diameter
    Counter_Zones=Counter_Zones+2; %Solely the extern borders of the passing zones
   r=sqrt(Counter_Zones*Lambda*(f+Counter_Zones*Lambda/4)); %Width of the nth zone
   r_before=sqrt((Counter_Zones-1)*Lambda*(f+(Counter_Zones-1)*Lambda/4));
   w=r-r_before; %Size of the nth zone
   test=Diameter/2-(r-w/2+w/2);%The radius of the hole is taken into account

   if test<0 %"If" used to verify if the radius overide the diameter of the Photon Sieve
       i=0;
       Counter_Zones=Counter_Zones-2;
   end
end
if Counter_Zones<1 %In case of too small diameter
    disp([' ']);
    disp(['!!!!!!! Error in the entry, diameter too small for the Photon Sieve !!!!!!!']);
    disp([' ']);
end
%% Compute the total number of holes and the surface ratio
Numb_Total_Holes=0;
open_surface=0;
for i=1:2:Counter_Zones
    if i==1 %Special case for the central hole
           r=sqrt(i*Lambda*(f+i*Lambda/4));
    r_before=sqrt((i-1)*Lambda*(f+(i-1)*Lambda/4));
    w=r-r_before;
    Numb_Total_Holes=Numb_Total_Holes+1;
    open_surface=open_surface+pi*(w)^2;
    else
    r=sqrt(i*Lambda*(f+i*Lambda/4));
    r_before=sqrt((i-1)*Lambda*(f+(i-1)*Lambda/4));
    w=(r-r_before);

    Holes_Diameter=w/(1+(Number_Layer-1)/(Density/100)*(2-Density/100)*sqrt(3)/2);
    add=0;
    for j=1:1:Number_Layer
    add=add+floor((2*pi*(r_before+Holes_Diameter/2+(j-1)/0.9*1.1*sqrt(3)/2*Holes_Diameter))/(
        Holes_Diameter/(Density/100)));
    end
    Numb_Total_Holes=Numb_Total_Holes+add;
    open_surface=open_surface+add*pi*((Holes_Diameter/2))^2; %Verify the optimization of the
        size of the hole
    end
end
Surface_Ratio=open_surface/(pi*(Diameter/2)^2);

%% Generation of the entry signal (The Photon Sieve)
%Size of the array
L=Diameter*(1.1+0.2*2*(Speed-1));
```

```
%Diameter of the smallest holes
    r=sqrt(Counter_Zones*Lambda*(f+Counter_Zones*Lambda/4));
    r_before=sqrt((Counter_Zones-1)*Lambda*(f+(Counter_Zones-1)*Lambda/4));
    w=r-r_before;
    Smallest_Holes=w/(1+(Number_Layer-1)/(Density/100)*(2-Density/100)*sqrt(3)/2);
% Sizing the minimum pixel with respect to the smallest hole
dx_max=w/(1+5*(Speed-1));
M_min=L/dx_max;
M=ceil(M_min/1000)*1000;

dx=L/M;
x=-L/2:dx:L/2-dx;
y=x;
[X,Y]=meshgrid(x,y);
Signal=zeros(max(size(x)),max(size(x)));

%Display the number of pixels and its size!
if Single==1
disp([' ']);
disp(['Size of a pixel: ',num2str(10^6*dx),'x',num2str(10^6*dx),' [micron x micron]']);
disp([' ']);
disp(['Number of pixels: ',num2str(M),'x',num2str(M)]);
disp([' ']);
end

%Computation of the signal
counter_holes=0;
for i=1:2:(Counter_Zones)
      n=Counter_Zones-(i-1);
     if n==1 %Special case for the center
          r=sqrt(Lambda*(f+Lambda/4));
      Signal_Add=abs(sqrt((X).^2+(Y).^2))<=(1*r);
      Signal=Signal+Signal_Add;

     else %External cases
 r=sqrt(n*Lambda*(f+n*Lambda/4));
 r_before=sqrt((n-1)*Lambda*(f+(n-1)*Lambda/4));
 w=(r-r_before);
 Holes_Diameter=w/(1+(Number_Layer-1)/(Density/100)*(2-Density/100)*sqrt(3)/2);

          for Layer=1:1:Number_Layer
               numb_holes=floor((2*pi*(r_before+Holes_Diameter/2+(Layer-1)/0.9*1.1*sqrt(3)/2*
                    Holes_Diameter))/(Holes_Diameter/(Density/100)));
               if (Layer/2-(floor(Layer/2)))==0 %Angles depend of the layer (different for odd or
                    even layer)
 %Coordinate angles of the nodes
 angles=1:1:numb_holes;
 angles=angles/numb_holes*2*pi;
               else
                    angles=1.5:1:(numb_holes+0.5);
                    angles=angles/numb_holes*2*pi;

               end

 [Holes,Size]=Holes_Generator(dx,Holes_Diameter/2);
 Radius=(r_before+Holes_Diameter/2+(Layer-1)/0.9*1.1*sqrt(3)/2*Holes_Diameter);

 for i=1:1:numb_holes %Add a hole at every point centered by the angle
      x_center=cos(angles(i))*(Radius);
      y_center=sin(angles(i))*(Radius);
      i_center=floor(cos(angles(i))*(Radius)/dx);
      j_center=floor(sin(angles(i))*(Radius)/dx);
      i_begin=ceil(M/2)+i_center-floor(Size/2);
      i_end=ceil(M/2)+i_center+ceil(Size/2)-1;
      j_begin=ceil(M/2)+j_center-floor(Size/2);
      j_end=ceil(M/2)+j_center+ceil(Size/2)-1;
      Signal(j_begin:j_end,i_begin:i_end) =Signal(j_begin:j_end,i_begin:i_end)+Holes(:,:);
 end

```

```
        end
    end %End of the "if" condition
end %End of the generation loop

for i=1:1:M %In case of superposition of the hole, put the value upper than 2 to 1
    for j=1:1:M
        if Signal(j,i)>1
            Signal(j,i)=1;
        end
    end
end %End of the correction

end %End of the function
```

References

[1] Guillermo Peñate. How to make a zone plate, photon sieve or pinhole sieve
URL:http://pinholeday.org/docs/zone_plate_and_sieves/
Date:04-10-2018.

[2] G. McHarg T. Quiller C. Maldonado G. Andersen, O. Asmolova. Optical testing of a membrane diffractive optic for space-based solar imaging. SPIE, 9906:1–7, 2014.

[3] ESO. Elt instruments
URL:http://www.eso.org/public/images/elt-instruments-rectangle/
Date:05-03-2018.

[4] ESO. Interferometry
URL:http://www.eso.org/public/teles-instr/technology/interferometry/
Date:05-03-2018.

[5] ESO. Transparency of the atmosphere
URL:https://www.eso.org/public/images/atm_opacity//
Date:05-03-2018.

[6] D. Leisawitz. The space infrared interferomet ric telescope (spirit)
URL:https://arxiv.org/ftp/arxiv/papers/0707/0707.0883.pdf
Date:03-03-2018.

[7] ESA. Darwin overview
URL:https://www.esa.int/our_activities/space_science/darwin_overview
Date:03-03-2018.

[8] ULiege. Oufti-1, premier nano-satellite belge, prêt pour l'espace
URL:https://www.uliege.be/cms/c_3606000/fr/oufti-1-premier-nano-satellite-belge-pret-pour-l-espace
Date:05-03-2018.

[9] Duijsens Sébastien. Preliminary study of a deployable CubeSat. Liège, Université, 2016-2017.

[10] Mahesh Shenoy. What would the earth or the universe look like
URL:https://www.quora.com/what-would-the-earth-or-the-universe-look-liken/
Date:03-24-2018.

[11] Eugene Hecht. Optique 4e édition (Traduction française). Pearson Education France, 47 Rue des vinaigriers, 75010 Paris, 2005.

[12] Young's double slits interference
URL:https://www.quizover.com/physics3/course/3-1-young-s-double-slit-interference-by-openstax?page=1
Date:04-05-2018.

[13] Cours2: Notion de coherence
URL:http://hebergement.u-psud.fr/l3papp/wp-content/uploads/2016/01/cours-2-coh%c3%a9rence.pdf
Date:04-01-2018.

[14] Lawrence Berkeley National Laboratory. Fresnel zone plate theory
URL:http://zoneplate.lbl.gov/
Date: 04-07-2018.

[15] Adam Jeff. Zone plate for gas-jet focusing
URL:http://slideplayer.com/slide/9395513/28
Date:03-10-2018.

[16] G. McHarg G. Andersen, M. Deadborn. Photon sieve space telescope. pages 1–7.

[17] Y. Lin F. Xu X. Zhu D. Pu L. Chen C. Whang X. Zaho, J. Hu. Ultra-broadband achromatic imaging with diffractive photon sieve. Scientific Reports, pages 1–9, 2016.

[18] G. McHarg T. Quiller C. Maldonado G. Andersen, O. Asmolova. Falconsat-7: a membrane solar space telescope. SPIE, 9904:1–9, 2006.

[19] EoPortal. Falconsat-7
URL:https://eoportal.org/web/eoportal/satellite-missions/pag-filter/-/asset_publisher/8jbnpfmcmhvk/content/falconsat-7
Date:04-10-2018.

[20] ULiege. Interférométrie destructive
URL:http://www.reflexions.uliege.be/cms/c_23277/interferometrie-destructive
Date: 05-10-2018.

[21] Utkarsh Sharma and Xing Wei. Fiber optic interferometric devices. Springer Science+Business Media New York, pages 1–26, 2013.

[22] S. Ridgway V. Coude du Foresto and J.-M. Mariotti. Deriving object visibilities from interferograms obtained with a fiber stellar interferometer. ASTRONOMY ASTROPHYSICS FEBRUARY 1997, PAGE 379 SUPPLEMENT SERIES, pages 379–392, 1997.

[23] Satya Mallick. Approximate focal length
URL:https://www.learnopencv.com/approximate-focal-length-for-webcams-and-cell-phone-cameras/
Date: 05-01-2018.

[24] DEW. Hand held light sensors: Leds
URL:http://dew.globalsystemsscience.org/activities/investigations/exploring-measuring-light/hand-held-light-sensors-leds/
Date: 05-01-2018.

[25] Telescope equations
URL:http://www.rocketmime.com/astronomy/telescope/magnitudegain.html
Date: 05-02-2018.

[26] Critère de rayleigh
URL:https://www.ilephysique.net/sujet-critere-de-rayleigh-272633.html
Date: 05-02-2018.

[27] Damien Gatinel. Mtf – modulation transfer function
URL:https://www.gatinel.com/recherche-formation/acuite-visuelle-definition/qualite-de-vision/
Date: 03-03-2018.

[28] Kintziger Christian. Feasibility of a UV imager onboard a Cubesat platform. Liège, Université, 2012-2013.

[29] Synopsys. Optical design
URL:https://www.synopsys.com/optical-solutions/codev/optical-design.html
Date:04-17-2018.

[30] What is a spherical wave front?
URL:https://www.quora.com/what-is-a-spherical-wave-front
Date: 03-10-2018.

[31] David Voelz. Computational Fourier Optics. SPIE, Bellingham, Washington USA, 2011.

[32] Airy pattern intensity and encircled power
URL:https://commons.wikimedia.org/wiki/file:airy_pattern_intensity_and_encircled_power.svg
Date:05-05-2018.

[33] Nola Taylor Redd. Hubble space telescope: Pictures, facts history
URL:https://www.space.com/15892-hubble-space-telescope.html
Date:05-15-2018.

[34] Rémy Decourt. Télescope spatial james-webb : le lancement repoussé à mai 2020
URL:https://www.futura-sciences.com/sciences/actualites/astronautique-telescope-spatial-james-webb-lancement-repousse-mai-2020-52187/
Date:05-15-2018.

[35] Gary Dagorn. Comment spacex s'est fait une place dans l'aérospatiale
URL:http://www.lemonde.fr/les-decodeurs/article/2018/02/07/comment-spacex-s-est-fait-une-place-dans-l-aerospatial_5253404_4355770.html
Date:04-02-2018.

[36] ULiege. Oufti-1, lancement réussi !
URL:http://events.ulg.ac.be/oufti-1/oufti-1/
Date:05-03-2018.

[37] Hsiu-Hsin Chung. Fabrication and testing of dual-wavelength photon sieves. University of Florida, 2006.

[38] Defrère D. Space interferometry. Liège, Université, 2017-2018.

[39] What is hubble's field of view?
URL:https://www.physicsforums.com/threads/what-is-hubbles-field-of-view.883507/
Date: 05-01-2018.

[40] Rudi Rottenfusser. The point spread function
URL:https://www.zeiss.com/microscopy/int/solutions/reference/basic-microscopy/the-point-spread-function.html
Date: 03-03-2018.

[41] Devdatta Biswas. Design software: which package do you need?
URL:http://optics.org/article/17912
Date: 04-20-2018.

[42] J. W. Goodman. Introduction to Fourier Optics. Toberts Company, Greenwood Village, CO, 2005.

[43] Introduction à l'optique de fourier
URL:https://lob.epfl.ch/webdav/site/lob/users/113390/public/p1%20fourier%20optics.pdf
Date:02-15-2018.

[44] Lutz Kipp. Photon sieves - new way of focusing fel radiation
URL:http://photon-science.desy.de/research/research_highlights
/archive/photon_sieves/index_eng.html
Date:04-05-2018.

[45] Gabor Patonay. Advances in near-infrared measurments
URL:https://books.google.be/books?id=zcoxbqaaqbajpg=pa5lpg=pa5dq=cone+d
Date: 06-05-2018.