

Analytic Modeling of the Segmented Telescope Performance with Adaptive Optics Correction of Static and Dynamic Aberrations, followed by OPTICA user's manual

Laurent Jolissaint,
Herzberg Institute of Astrophysics, National Research Council Canada,
Victoria, BC, Canada

February 19, 2007

1 Introduction

Primary mirrors of the next generation of extremely large telescopes will be highly segmented. Since these telescopes will be equipped with adaptive optics (AO), telescope aberrations will be partly corrected, therefore, when designing both the telescope and the client AO system(s), this telescope/AO interaction needs to be taken into account. For example, availability of AO correction might help in relaxing somewhat the telescope aberration specifications, but reversely, unavoidable residual aberrations, after active optics (aO) correction, will be corrected by the client AO system, capturing some deformable mirror (DM) actuators stroke resources and wavefront sensor (WFS) dynamical range.

To study this interaction, two kind of models are needed: (1) models that allow analysis of the interaction between the main parameters of the system, by a quick exploration of the system's parameter space; (2) detailed physical models of the telescope optics and structure, linked with the AO system model, for detailed analysis. Generally, models of the first type are analytic, in a sense that they can give an estimate of a given metric as a function of a given parameter by using a single formula, hence the possibility to quickly explore the parameter space and get trends in parameters relationship. Models of the second type are generally integrated models, where every single component of the system is simulated as an input/output independent module, connected with the other sub-systems modules. The accuracy of the modeling is virtually not limited with this approach, and relies on the accuracy of each sub-system modeling, and availability of computing time. These type of models are useful to concentrate on highly non-linear behavior that are difficult or impossible to model analytically (for instance pupil and WFS misregistration).

To summarize, it can be said that analytical models are useful to have a general view and understanding of the behavior and limits of the system, while integrated models are useful to study very specific, non-linear or complex interactions between the system's components. We assert here that both approaches are required to have a good understanding of the system and its very detailed behavior.

In this document, we report on our development of a first type, analytic model of the optical performance of the system telescope plus AO. It has been developed as a support to the determination of the Thirty Meter Telescope project budget error. The model is build around three main components: (1) the static point spread function (PSF) modeling, (2) the dynamic aberrations optical transfer function (OTF) modeling - and here we must give credit to Brent Ellerbroek for initiating the approach

that we have developed further here, and (3) the model for AO correction of both static and dynamic aberrations.

Fourier optics, as described by Goodman [1], is the central tool on which the construction of the model relies: indeed, in general, optical quantities do have an analytical representation in the spatial frequency space, and operations like AO filtering are described as simple products. Where needed, Fast Fourier Transform algorithms (FFT) are applied to travel between the spatial coordinates and the spatial frequencies domains.

The model has been coded into a MATLAB-based tool, OPTICA¹, that takes as inputs the static aberrations and the dynamic aberrations temporal covariance of each optical surfaces, the telescope optics architecture, the basic parameters of the AO system, and gives at the output the static amplitude PSF, the long exposure intensity PSF and OTF, and other quantities of interest. It is worth noting that AO is an option in OPTICA, and the code can also be used to study telescope performance without AO correction.

This document is organized as follows: in section 2, we show how the telescope OTF can be split as a product of static and dynamic OTFs, using the stationary phase approximation; in section 3, we give the expressions for the calculation of the telescope static amplitude PSF, from which the static OTF is easily obtained; in section 4, the dynamic OTF formulas are given; and in section 5, we develop the AO filter, and show how it can be applied to the modeling of the static and dynamic aberrations correction; finally, section 6 is the user manual of the code OPTICA. Long and complicated analytical developments are to be found in the appendices.

Some introductory material

Before starting, let us recall some of the basics of the relationship between the pupil plane and the focal plane, in the framework of Fourier optics.

- **the pupil plane phasor**, noted U_p in this report (we are following Goodman's notations) is an other term to describe the complex wave amplitude in the pupil plane (i.e. proportional to the electrical field of the wave), where the temporal oscillation $\exp(i\omega t)$ has been removed, as it does not affect the image structure; the argument of the pupil plane phasor is the position vector \mathbf{u} in the pupil plane.
- **the focal plane phasor**, noted U_f , is the quantity one would expect, in the focal plane, and apart from a multiplicative constant that again does not affect the image structure, is given by the Fourier transform of the pupil plane phasor. Its argument is a angular position vector projected in the sky plane at infinity, noted $\boldsymbol{\alpha}$, and because of the Fourier transform relationship, $\boldsymbol{\alpha}$ is related to the pupil plane spatial frequency \mathbf{f} , with $\boldsymbol{\alpha} = \lambda \mathbf{f}$, where λ is the optical wavelength.
- **the optical transfer function** is the Fourier transform of the point spread function, and its argument is the angular frequency in the focal plane, noted $\boldsymbol{\nu}$ in this report. The OTF is normalized at the origin, i.e. $\text{OTF}(0)=1$. It can be shown that the OTF can be calculated from the autocorrelation of the pupil plane phasor U_p - Goodman [1].

¹Optical Performance of a Telescope Including Correction with Adaptive optics

To summarize, we have the following relationship between the phasors, the PSF and the OTF - \mathcal{F} is the Fourier transform operator

$$\begin{array}{ccc}
 U_p(\mathbf{u}) & \xrightarrow{\mathcal{F}} & U_f(\boldsymbol{\alpha} = \lambda \mathbf{f}) \\
 \text{autocorrelation} \downarrow & & \downarrow |\cdot|^2 \\
 \text{OTF}(\boldsymbol{\nu}) & \xleftarrow{\mathcal{F}} & \text{PSF}(\boldsymbol{\alpha})
 \end{array}$$

Also, we have the following relationship between the pupil plane and focal plane coordinate systems:

- a feature at a spatial frequency \mathbf{f} in the pupil plane will affect the structure of the focal plane PSF at the angular positions $\boldsymbol{\alpha} = \pm \lambda \mathbf{f}$;
- a feature at a scale \mathbf{u} in the pupil plane will affect the structure of the focal plane OTF at the angular frequency $\boldsymbol{\nu} = \lambda \mathbf{u}$

Finally, and as the Fourier transform is used intensively throughout this report, it is useful to recall its definition: if g is any complex-valued function of two independent variables x and y , and if $\iint |g(x, y)|^2 dx dy < \infty$, then the Fourier transform of g is the two dimensional complex-valued quantity G defined by

$$G(f_x, f_y) = \iint_{\mathbb{R}^2} g(x, y) \exp[-i2\pi(f_x x + f_y y)] dx dy \quad (1)$$

and the reciprocal relation defines the inverse Fourier transform,

$$g(x, y) = \iint_{\mathbb{R}^2} G(f_x, f_y) \exp[i2\pi(f_x x + f_y y)] df_x df_y \quad (2)$$

Details on the properties of the two dimensional Fourier transform can be found in Goodman [1].

2 The telescope optical transfer function as a product of static and dynamic optical transfer functions

Telescope aberrations can be separated into static - φ_{sta} - and dynamic ones - $\delta\varphi$:

$$\varphi(\mathbf{u}, t) = \varphi_{\text{sta}}(\mathbf{u}) + \delta\varphi(\mathbf{u}, t) \quad (3)$$

where \mathbf{u} is the coordinate in the pupil plane, and by definition, $\langle \delta\varphi(\mathbf{u}, t) \rangle_t = 0$. We give in table 1 a list of potential sources of static and dynamic aberrations. The most convenient way of characterizing the effect of these aberrations is via the system's OTF, and as we are dealing here with dynamical

Static	Dynamic
polishing errors	wind pressure
positioning errors	telescope vibrations
mirrors print-through	thermal effects

Table 1: **Potential sources of static and dynamic telescope aberrations - non exhaustive.**

aberrations, the time average, or long exposure OTF is of particular interest. For an incoherent optical system, the long exposure OTF can be calculated from the pupil plane phasor auto-correlation,

$$\begin{aligned} \langle \text{OTF} \rangle(\boldsymbol{\nu}) = & \\ \frac{1}{S_P} \iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda \boldsymbol{\nu}) \exp \{ -i[\varphi_{\text{sta}}(\mathbf{u}) - \varphi_{\text{sta}}(\mathbf{u} + \lambda \boldsymbol{\nu})] \} & \langle \exp \{ -i[\delta\varphi(\mathbf{u}, t) - \delta\varphi(\mathbf{u} + \lambda \boldsymbol{\nu}, t)] \} \rangle_t d^2u \end{aligned} \quad (4)$$

where S_P is the pupil surface, P the pupil transmission (1 inside, 0 outside), λ the optical wavelength, and $\boldsymbol{\nu}$ the focal plane angular frequency. The time average of the dynamic phasor in Eq. (4) can be developed further, considering the following: let ξ be a random variable of probability distribution p_ξ ; $\exp(-i\xi)$ is also a random variable, and its expectation value is given by

$$\langle \exp(-i\xi) \rangle = \int p_\xi(\xi) \exp(-i\xi) d\xi = \mathcal{F}\{p_\xi(\xi)\}(\nu = 1/2\pi) \quad (5)$$

which is simply the Fourier transform of p_ξ taken at the frequency $\nu = 1/2\pi$. In our case, ξ represents the phase difference between two points in the pupil plane ($\lambda \boldsymbol{\nu} \rightarrow \xi$) and with the assumption that the perturbed phase has a centered Gaussian statistics, i.e.

$$p_\xi(\xi) = \frac{\exp[-\xi^2/(2\sigma_\xi^2)]}{\sigma_\xi \sqrt{2\pi}} \quad (6)$$

it comes

$$\tilde{p}_\xi(\nu) = \exp(-2\pi^2\sigma_\xi^2\nu^2) \quad (7)$$

therefore

$$\langle \exp(-i\xi) \rangle = \exp(-\sigma_\xi^2/2) \quad (8)$$

Applying this result to Eq. (4), we get

$$\langle \text{OTF} \rangle(\boldsymbol{\nu}) = \frac{1}{S_P} \iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda \boldsymbol{\nu}) \exp \{ -i[\varphi_{\text{sta}}(\mathbf{u}) - \varphi_{\text{sta}}(\mathbf{u} + \lambda \boldsymbol{\nu})] \} \exp \left[-\frac{1}{2}D_{\delta\varphi}(\mathbf{u}, \lambda \boldsymbol{\nu}) \right] d^2u \quad (9)$$

where $D_{\delta\varphi}$ is the phase structure function, defined by the average quadratic phase difference as a function of the separation distance within the pupil,

$$D_{\delta\varphi}(\mathbf{u}, \lambda \boldsymbol{\nu}) = \langle [\delta\varphi(\mathbf{u}, t) - \delta\varphi(\mathbf{u} + \lambda \boldsymbol{\nu}, t)]^2 \rangle_t \quad (10)$$

Assuming that the telescope primary and secondary mirrors (in this document, we call secondary mirror any mirror that comes after the primary) are independently affected by dynamical mechanical perturbations, independent structure functions can be defined for each mirror, and their sum will define the telescope structure function,

$$D_{\delta\varphi}^{\text{tsc}}(\mathbf{u}, \lambda \boldsymbol{\nu}) = \sum_i D_{\delta\varphi}^{\text{M}_i}(\mathbf{u}, \lambda \boldsymbol{\nu}) \quad (11)$$

Stationarity approximation

The exact calculation of the long exposure OTF requires, for each focal plane angular frequency $\boldsymbol{\nu}$, a numerical integration over the pupil position \mathbf{u} - see Eq. (9). This takes a lot of time, and does not match with the requirement of a fast OTF modeling tool. Fortunately, we know from previous developments in astronomical adaptive optics (AO) analytical modeling [2] that the complexity of the calculation can be greatly reduced by the use of the so-called *stationary approximation*. The idea is that if the dynamical phase statistics would not be a function of the position \mathbf{u} in the telescope pupil, i.e. if the phase properties were the same everywhere in the pupil independently of the position \mathbf{u} , or in other words if the phase was stationary in the pupil, then it would be possible to write its structure function only as a function of the separation distance $\boldsymbol{\rho} = \lambda\boldsymbol{\nu}$ in the pupil plane, and consequently extract the structure function exponential $\exp(-D_{\delta\varphi}/2)$ from the OTF integral - Eq. (9). This would reduce the OTF calculation to a simple evaluation of an expression for each focal plane angular frequency, i.e. reduce the dimension of the calculation from an order $\mathcal{O}(N^4)$ to $\mathcal{O}(N^2)$, where N is the OTF matrix size.

Stationarity is realized for instance for the non-corrected atmospheric turbulent phase [3], but not in the case of telescope aberrations, as sources of optical perturbation can be expected to be highly asymmetric and anisotropic (telescope vibrations for instance, or wind load on the secondary). For segmented pupils, though, we will see that the stationarity assumption does not have to be done on the full segmented pupil, but only at the level of the segments: i.e. we need only to assume stationarity within the segments themselves. Therefore, inhomogeneous perturbations can still be specified for the segmented primary mirror.

In order to simplify the calculation complexity, then, we will assume that the phase *is* stationary, and this stationarity can be forced into our model by replacing the phase structure function by its spatial average in the pupil, over \mathbf{u} . Let us examine how this average must be defined, and what is the consequence of this stationarity approximation. We start by defining the average of $\exp[-D_{\delta\varphi}(\mathbf{u}, \lambda\boldsymbol{\nu})/2]$ in the pupil as

$$\begin{aligned} \langle \exp[-\frac{1}{2}D_{\delta\varphi}(\mathbf{u}, \lambda\boldsymbol{\nu})] \rangle_{\mathbf{u}} &\equiv \frac{\langle \text{OTF} \rangle(\boldsymbol{\nu})}{\text{OTF}_{\text{sta}}(\boldsymbol{\nu})} \\ &= \frac{\iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\boldsymbol{\nu}) \exp\{-i[\varphi_{\text{sta}}(\mathbf{u}) - \varphi_{\text{sta}}(\mathbf{u} + \lambda\boldsymbol{\nu})]\} \exp[-\frac{1}{2}D_{\delta\varphi}(\mathbf{u}, \lambda\boldsymbol{\nu})] d^2u}{\iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\boldsymbol{\nu}) \exp\{-i[\varphi_{\text{sta}}(\mathbf{u}) - \varphi_{\text{sta}}(\mathbf{u} + \lambda\boldsymbol{\nu})]\} d^2u} \end{aligned} \quad (12)$$

where OTF_{sta} indicates the telescope static OTF. Now, from Jensen's inequality [4], we know that as the function $\exp(x)$ is a concave down function, the exponential of the average over a range $x \in [a, b]$ is smaller than the average of the exponential over $[\exp(a), \exp(b)]$, i.e. $\exp\langle x \rangle \leq \langle \exp x \rangle$, so we must have

$$\exp[-\frac{1}{2}\langle D_{\delta\varphi}(\mathbf{u}, \lambda\boldsymbol{\nu}) \rangle_{\mathbf{u}}] \leq \langle \exp[-\frac{1}{2}D_{\delta\varphi}(\mathbf{u}, \lambda\boldsymbol{\nu})] \rangle_{\mathbf{u}} \quad (13)$$

and the average structure function would have to be computed using

$$\langle D_{\delta\varphi}(\mathbf{u}, \lambda\boldsymbol{\nu}) \rangle_{\mathbf{u}} = \frac{\iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\boldsymbol{\nu}) \exp\{-i[\varphi_{\text{sta}}(\mathbf{u}) - \varphi_{\text{sta}}(\mathbf{u} + \lambda\boldsymbol{\nu})]\} D_{\delta\varphi}(\mathbf{u}, \lambda\boldsymbol{\nu}) d^2u}{\iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\boldsymbol{\nu}) \exp\{-i[\varphi_{\text{sta}}(\mathbf{u}) - \varphi_{\text{sta}}(\mathbf{u} + \lambda\boldsymbol{\nu})]\} d^2u} \quad (14)$$

but this is a complex quantity, yet by definition a structure function is positive and real. Now, as the telescope static phase amplitude is in principle relatively small, $\exp(i\varphi_{\text{sta}}) \lesssim 1$, so we will neglect it here and define the stationary structure function as

$$\mathbb{R}\overline{D}_{\delta\varphi}(\lambda\boldsymbol{\nu}) \equiv \frac{\iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\boldsymbol{\nu}) D_{\delta\varphi}(\mathbf{u}, \lambda\boldsymbol{\nu}) d^2u}{\iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\boldsymbol{\nu}) d^2u} \quad (15)$$

where the symbol \mathbb{R} is here to recall that the static aberration is neglected in this definition. Now, the structure function exponential can be extracted from the integral Eq. (9), and using the notation $\langle \text{OTF} \rangle_{\{\approx\}}$ to recall the stationary approximation, we get

$$\begin{aligned} \langle \text{OTF} \rangle_{\{\approx\}}(\boldsymbol{\nu}) &= \exp \left[-\frac{1}{2\mathbb{R}} \overline{D}_{\delta\varphi}(\lambda\boldsymbol{\nu}) \right] \frac{1}{S_P} \iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\boldsymbol{\nu}) \exp \left\{ -i[\varphi_{\text{sta}}(\mathbf{u}) - \varphi_{\text{sta}}(\mathbf{u} + \lambda\boldsymbol{\nu})] \right\} d^2u \\ &= \langle \text{OTF} \rangle_{\text{dyn}}(\boldsymbol{\nu}) \text{OTF}_{\text{sta}}(\boldsymbol{\nu}) \leq \langle \text{OTF} \rangle(\boldsymbol{\nu}) \end{aligned} \quad (16)$$

where OTF_{sta} is the telescope static OTF, and $\langle \text{OTF} \rangle_{\text{dyn}}$ is what we call here the dynamic OTF. It is worth noting that this stationary telescope OTF is, according to Eq. (13), always smaller or equal to the real telescope OTF, therefore the assessment of the effect of aberrations on the telescope performances will always be pessimistic, which is safe, from the point of view of error budgeting. Also, we have been using this stationarity approximation in others contexts (modeling of AO correction of atmospheric turbulence aberrations, and AO PSF reconstruction) and it has always been seen that it has very little effect on the models' accuracy, except in the strong aberration regime.

Note on the calculation of the static and dynamic OTFs In the spirit of our first order approach, amplitude effects due to Fresnel propagation between each optical surfaces are neglected. Therefore, we are in the geometric aberrations regime, and the total static phase aberration is given by the sum of each mirror static phase aberration, projected onto the segmented pupil. It is worth noting that any type of static aberration can be taken into account with this approach, and in particular, (1) off-axis aberrations can be seen as a n-ary mirror with surface errors, (2) aberrations inside the footprint of the beam on a given secondary mirror can be projected back onto the pupil. No aberration model is constructed in our analytical approach: we are just assuming a set of aberrations inputs, and compute the associated OTF/PSF using Fourier optics theory.

There is, of course, only one static OTF for the whole telescope, so a method is required to project the secondary mirrors static aberrations onto the segments static aberrations basis, and this method is provided in page 12. Moreover, spider and secondary mirrors support structures imprint shading on the segmented pupil, and a method is also provided to take this into account - see page 14. This shaded segmented pupil, with its projected total static aberration, is used to define the static pupil plane phasor, from which the amplitude PSF is calculated, using Fourier optics theory.

The case of the dynamic OTF is quite different. Indeed, we mentioned above that there is one structure function per mirror, therefore, there is also one dynamic OTF per mirror, and their product defines the global dynamic OTF,

$$\langle \text{OTF} \rangle_{\text{dyn}}^{\text{tsc}} = \prod_i \langle \text{OTF} \rangle_{\text{dyn}}^{M_i} \quad (17)$$

These dynamic OTFs are not OTFs in the usual sense (PSF Fourier transform), and must rather be interpreted as low pass filters applied on the telescope static OTF. Finally, as we will see, the calculation of the dynamic structure function also relies on the basis of the Fourier optics theory. These important precisions being made, we can now move on to the calculation of the static and dynamic OTFs.

3 The static amplitude point spread function and the static optical transfer function

This section start with the description of the method to compute the static amplitude PSF, for a segmented pupil made of circular or hexagonal segments. It continues by giving the formulas for the projection of the secondary mirrors static aberrations onto the segments basis, and ends with the demonstration of how the spider and secondary mirrors support shading can be taken into account.

3.1 Segmented telescope amplitude point spread function

In the pupil plane of a N_S segments telescope, the complex amplitude, or focal plane phasor, with aberrations, is given by

$$U_p(\mathbf{u}) = \sum_{s=1}^{N_S} \{ \mathcal{S}_s(\mathbf{u}) \exp[-i\varphi_s(\mathbf{u})] \} \star \delta(\mathbf{u} - \mathbf{u}_s) \quad (18)$$

where \star is the convolution product, \mathcal{S}_s is the transmission of segment s , φ_s the aberration within the segment, \mathbf{u}_s the segment center coordinate, and δ the Dirac- δ distribution. The focal plane phasor - or amplitude PSF - is proportional to the Fourier transform of the pupil plane phasor (Fraunhofer diffraction equation). Here we assume that the segment transmission, as seen from the sky, is identical for every segments. Before applying the Fourier transform operator, we can rewrite Eq. (18) as the sum of a perfect phasor and an aberrated phasor component. Indeed, as $\mathcal{S}^n(\mathbf{u}) = 1$ or 0 for any n , we can write

$$\begin{aligned} \mathcal{S} \exp(-i\varphi_s) &= \mathcal{S} \sum_{n=0}^{\infty} \frac{(-i)^n}{n!} \varphi_k^n = \mathcal{S} - i \mathcal{S} \varphi_k - \frac{1}{2} (\mathcal{S} \varphi_k)^2 - \dots \\ &= \mathcal{S} - 1 + 1 - i \mathcal{S} \varphi_k - \frac{1}{2} (\mathcal{S} \varphi_k)^2 - \dots \\ &= \mathcal{S} - 1 + \sum_{n=0}^{\infty} \frac{(-i)^n}{n!} (\mathcal{S} \varphi_k)^n \\ &= \mathcal{S} - 1 + \exp(-i \mathcal{S} \varphi_k) \end{aligned} \quad (19)$$

therefore,

$$U_p(\mathbf{u}) = \mathcal{S}(\mathbf{u}) \star \sum_{s=1}^{N_S} \delta(\mathbf{u} - \mathbf{u}_s) + \sum_{s=1}^{N_S} \{ \exp[-i \mathcal{S}(\mathbf{u}) \varphi_s(\mathbf{u})] - 1 \} \star \delta(\mathbf{u} - \mathbf{u}_s) \quad (20)$$

where the first term is the segmented pupil transmission, and the second term is the aberrated phasor. We apply now the Fourier transform operator on Eq. (20), and we get²

$$U_f(\lambda \mathbf{f}) = \underbrace{\tilde{\mathcal{S}}(\mathbf{f}) \sum_{s=1}^{N_S} \exp(-i2\pi \mathbf{f} \cdot \mathbf{u}_s)}_{\text{pupil Fourier transform } \tilde{P}(\mathbf{f})} + \sum_{s=1}^{N_S} \mathcal{F}\{ \exp[-i \mathcal{S}(\mathbf{u}) \varphi_s(\mathbf{u})] - 1 \} \exp(-i2\pi \mathbf{f} \cdot \mathbf{u}_s) \quad (21)$$

²in the Fraunhofer equation, there is a constant term in front of the Fourier transform, but as this term does not affect the structure of the amplitude PSF, we will neglect it.

where \mathbf{f} is the pupil plane spatial frequency, associated to the focal plane angular coordinate $\alpha = \lambda \mathbf{f}$ (here in radian on the sky). The first term is the perfect telescope amplitude PSF, and the second term is the aberrated component of the amplitude PSF.

This representation offers several advantages. First, it clearly separates both the perfect and the aberrated amplitude PSF. Second, as the Fourier transform of the segment transmission $\tilde{\mathcal{S}}$ can be given in a closed analytic form, the perfect amplitude PSF can be calculated analytically without any need to numerically built a pupil transmission matrix. Third, the Fourier transform of the product $\mathcal{S} \varphi_s$ can be calculated analytically as well, as we will show below, therefore the aberrated amplitude PSF can also be calculated without the need to build a pupil plane phasor. Finally - and this is the main interest of this Fourier transform approach - this representation allows an easy modeling of AO correction of the static aberrations: indeed, AO correction can be modeled, in a first approximation, as a spatial filter, γ_{AO} , applied on the segmented phase (see section 5) and AO correction is modeled by replacing the Fourier transform $\mathcal{F}\{\mathcal{S} \varphi_s\}$ by its filtered version, $\gamma_{\text{AO}}\{\mathcal{F}\{\mathcal{S} \varphi_s\}\}$. As we can see, the Fourier transform of the product $\mathcal{S} \varphi_s$ is central in our modeling, and we give their expressions below for both circular and hexagonal segments, as these type of segment shape are the most commonly used in practice.

We will assume that phase aberration is given as a set of Zernike aberration coefficients, for the segments as well as for the secondary mirrors, so we have

$$\mathcal{S}(\mathbf{u}) \varphi_s(\mathbf{u}) = \sum_{j=1}^{\infty} z_{s,j} \tilde{\mathcal{S}}(\mathbf{u}) Z_j(\mathbf{u}/a) \quad (22)$$

i.e.

$$\mathcal{F}\{\mathcal{S}(\mathbf{u}) \varphi_s(\mathbf{u})\} = \sum_{j=1}^{\infty} z_{s,j} \mathcal{F}\{\tilde{\mathcal{S}}(\mathbf{u}) Z_j(\mathbf{u}/a)\} \quad (23)$$

where a is the radius of the segment (see figure 1), and $z_{s,j}$ are the segments aberration Zernike coefficients. What we need then is the Fourier transform of the products $\mathcal{S} Z_j$.

3.1.1 Circular segments

For circular segments, it comes that $\mathcal{S} Z_j = Z_j$, so we simply need, to compute the Fourier transform of the Zernike polynomials, with a proper scaling relative to the segment radius a . Noll [5] has calculated the Fourier transform of the Zernike polynomials (written Q_j), but was using a sign convention such that

$$Z_j(\mathbf{u}') = \pi \iint Q_j(\mathbf{f}'_{\text{Noll}}) \exp(-i2\pi \mathbf{f}'_{\text{Noll}} \cdot \mathbf{u}') d^2 f'_{\text{Noll}} \quad (24)$$

where $\mathbf{u}' = \mathbf{u}/a$, i.e. a sign convention opposite to Goodman's [1]), that we have adopted in our work, so we have to invert the sign of the spatial frequency vector \mathbf{f} in Noll's equations, so, with $\mathbf{f} = -\mathbf{f}'_{\text{Noll}}$,

$$Z_j(\mathbf{u}') = \pi \iint Q_j(-\mathbf{f}') \exp(i2\pi \mathbf{f}' \cdot \mathbf{u}') d^2 f' = \pi \iint Q_j^*(\mathbf{f}') \exp(i2\pi \mathbf{f}' \cdot \mathbf{u}') d^2 f' \quad (25)$$

where Q_j^* indicates the complex conjugate of Q_j . The FT of Z_j can be calculated now, and we get

$$\begin{aligned} \mathcal{F}\{Z_j(\mathbf{u}/a)\} &= \pi \mathcal{F}\left\{ \iint Q_j^*(\mathbf{f}') \exp(i2\pi \mathbf{f}' \cdot \mathbf{u}/a) d^2 f' \right\} \\ &\stackrel{f=f'/a}{=} \pi a^2 \mathcal{F}\left\{ \iint Q_j^*(a\mathbf{f}) \exp(i2\pi \mathbf{f} \cdot \mathbf{u}) d^2 f \right\} \\ &= \pi a^2 \mathcal{F}\{\mathcal{F}^{-1}\{Q_j^*(a\mathbf{f})\}\} = \pi a^2 Q_j^*(a\mathbf{f}) \end{aligned} \quad (26)$$

therefore, for circular segments,

$$\mathcal{F}\{\mathcal{S}(\mathbf{u}) \varphi_s(\mathbf{u})\} = \pi a^2 \sum_{j=1}^{\infty} z_{s,j} Q_j^*(af) \quad (27)$$

where (we take the complex conjugate of Noll's expressions)

$$Q_j^*(af) = \sqrt{(n+1)} (-1)^{(n-m)/2} (-i)^m \frac{2 J_{n+1}(2\pi af)}{2\pi af} \begin{cases} 1 & m = 0 \\ \sqrt{2} \cos(m\phi) & m \neq 0 \text{ and } j \text{ even} \\ \sqrt{2} \sin(m\phi) & m \neq 0 \text{ and } j \text{ odd} \end{cases} \quad (28)$$

with f and ϕ the polar coordinates associated to the vector \mathbf{f} .

3.1.2 Hexagonal segments

We will assume here that the hexagonal transmission ($\mathcal{S} \rightarrow \mathcal{H}$) is inscribed within the disc domain of the Zernike polynomials - see figure 1. It has to be noted that these truncated Zernike polynomials do not define an orthonormal basis anymore, but this is of little concern here as we do not use the orthogonality properties of the Zernike basis. It is not a difficulty, though, to define a new set of polynomials which defines an orthonormal basis on the hexagonal transmission, as shown for instance by Upton and Ellerbroek [6].

Unfortunately, nowhere in the literature can we find the Fourier transform of the Zernike polynomials truncated with hexagonal transmission, therefore we have to develop this ourselves (see appendix B). As it is not possible to keep the polar coordinates as Noll was able to, because the hexagonal transmission is not easily described with polar coordinates, we have to use the Cartesian representation of the Zernike polynomials (see appendix A),

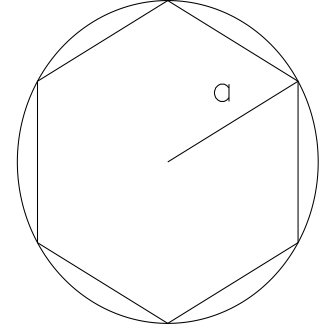


Figure 1: **Definition of hexagonal segment radius a.**

$$Z_j(u, v) = \sqrt{[(m \neq 0) + 1](n+1)} \sum_{s=0}^{\frac{n-m}{2}} \sum_{l=0}^{\frac{n-m}{2}-s} \sum_{k=0}^{q_{j,m}} (-1)^{(s+k)} \binom{m}{2k+p_{j,m}} \binom{\frac{n-m}{2}-s}{l} \times \frac{(n-s)!}{s! (\frac{n+m}{2}-s)! (\frac{n-m}{2}-s)!} u^{[n-2(s+l+k)-p_{j,m}]} v^{[2(l+k)+p_{j,m}]} \quad (29)$$

where

$q_{j,m} = \dots$	$m \text{ odd}$	$m \text{ even}$	$m = 0$	and	$p_{j,m} = \dots$	$m \neq 0$	$m = 0$
$j \text{ odd}$	$\frac{m-1}{2}$	$\frac{m-2}{2}$	0		$j \text{ odd}$	1	0
$j \text{ even}$	$\frac{m-1}{2}$	$\frac{m}{2}$			$j \text{ even}$	0	0

With this Cartesian representation, each polynomial is written as a sum of monomials in $u^\mu v^\nu$, therefore, the Fourier transform of the segmented phase can be written as a sum of monomials Fourier transforms,

$$\mathcal{F}\{\mathcal{H}(\mathbf{u}) \varphi_s(\mathbf{u})\} = \sum_j z_{s,j} \sum_{\mu,\nu} c_{j,\mu,\nu} \mathcal{F}\{\mathcal{H}(\mathbf{u}) u^\mu v^\nu\} \quad (30)$$

The calculation of the terms $\mathcal{F}\{\mathcal{H}(\mathbf{u})u^\mu v^\nu\}$ is given in the appendix B, and we find

$$\mathcal{F}\{\mathcal{H}(\mathbf{u})u^\mu v^\nu\} = \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{3^{-\frac{\kappa+\lambda+\zeta}{2}}}{(-1)^{\kappa+\lambda} \kappa! \lambda!} \binom{\mu+\nu-\kappa-\lambda+1}{\zeta} \times \frac{f_u^\nu}{f_v^{\nu+1}} \left(\frac{f_v}{f_u}\right)^{\lambda+\zeta} \left[\frac{(-1)^{(\mu+\nu-\kappa-\lambda-\zeta)} \Theta_{\kappa,\lambda}^+(\mathbf{f})}{(f_v - \sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} + \frac{\Theta_{\kappa,\lambda}^-(\mathbf{f})}{(f_v + \sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} \right] \quad (31)$$

where the so-called angular functions $\Theta_{\kappa,\lambda}^\pm$ are defined by

$$\Theta_{\kappa,\lambda}^\pm(\mathbf{f}) = (\pm\sqrt{3})^\kappa (\pi a)^{\kappa+\lambda} \begin{cases} (-1)^{\frac{\kappa+\lambda}{2}} \cos[\pi a(f_v \pm \sqrt{3}f_u)] - \delta_{\kappa,0} 2^\lambda (-1)^{\frac{\lambda}{2}} \cos(2\pi a f_v) & \kappa + \lambda \text{ even} \\ (-1)^{\frac{\kappa+\lambda+1}{2}} \sin[\pi a(f_v \pm \sqrt{3}f_u)] - \delta_{\kappa,0} 2^\lambda (-1)^{\frac{\lambda+1}{2}} \sin(2\pi a f_v) & \kappa + \lambda \text{ odd} \end{cases} \quad (32)$$

Now, Eq. (31) is undetermined at the origin $(f_u, f_v) = (0, 0)$, along the horizontal axis f_u ($f_v \rightarrow 0$), and along the two lines at $\pm 60^\circ$ relative to the horizontal axis ($f_v \rightarrow \pm\sqrt{3}f_u$), but these limits can be calculated, and we find:

At the origin

$$\mathcal{F}\{\mathcal{H}(\mathbf{u})u^\mu v^\nu\}(0, 0) = \begin{cases} \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+1}}{\pi^2} (\pi a)^{\mu+\nu+2} (-1)^{\frac{\mu+\nu+2}{2}} \times \\ \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \frac{1 - \delta_{\kappa,0} 2^{\mu+\nu+2}}{(-1)^\lambda \kappa! \lambda! (\nu-\lambda)! (\mu-\kappa+1)! (\mu+\nu-\kappa-\lambda+2)} & \mu \text{ and } \nu \text{ even} \\ 0 & \mu \text{ or } \nu \text{ odd} \end{cases} \quad (33)$$

Along the f_u axis

$$\mathcal{F}\{\mathcal{H}(\mathbf{u})u^\mu v^\nu\}(f_u, 0) = \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(-\sqrt{3})^{\mu+\nu+1} (\mu+\nu+1)!}{2\pi^2 (\sqrt{3}f_u)^{\mu+\nu+2}} \times \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \binom{\mu+\nu-\kappa-\lambda+1}{\zeta} \frac{(\kappa+\lambda)! (-f_u^{\kappa+\lambda})}{\kappa! \lambda! (\nu+1-\lambda-\zeta)! (\sqrt{3})^{\nu-\lambda}} \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} (\sqrt{3}f_u)^\alpha \sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \binom{\nu+1-\lambda-\zeta-\alpha}{\beta} \frac{[(-1)^{\mu+\nu-\zeta-\beta} \partial_\alpha \Theta_{\kappa,\lambda}^+(f_u, 0) + (-1)^{\mu+\lambda+\zeta+\alpha+\beta} \partial_\alpha \Theta_{\kappa,\lambda}^-(f_u, 0)]}{(\kappa+\lambda-\beta)! (\mu+\lambda+\zeta+\alpha+\beta)!} \quad (34)$$

where

$$\partial_\alpha \Theta_{\kappa,\lambda}^\pm(f_u, 0) = (\pm\sqrt{3})^\kappa (\pi a)^{\kappa+\lambda+\alpha} \begin{cases} (-1)^{\frac{\kappa+\lambda+\alpha}{2}} \cos[\pi a(\pm\sqrt{3}f_u)] - \delta_{\kappa,0} 2^{\lambda+\alpha} (-1)^{\frac{\lambda+\alpha}{2}} & \kappa + \lambda + \alpha \text{ even} \\ (-1)^{\frac{\kappa+\lambda+\alpha+1}{2}} \sin[\pi a(\pm\sqrt{3}f_u)] & \kappa + \lambda + \alpha \text{ odd} \end{cases} \quad (35)$$

Along the $f_v \rightarrow \pm\sqrt{3}f_u$ asymptotes

$$\begin{aligned} \mathcal{F}\{\mathcal{H}(\mathbf{u})u^\mu v^\nu\}(f_u, -\sqrt{3}f_u) &= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{(-1)^\mu \mu! \nu! (\sqrt{3})^{\mu+1}}{\pi^2 (2\sqrt{3}f_u)^{\mu+\nu+2}} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta} (2\sqrt{3}f_u)^{\kappa+\lambda}}{(-1)^{\lambda+\zeta} \kappa! \lambda! (\sqrt{3})^\kappa} \times \\ &\quad \left[(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(f_u, -\sqrt{3}f_u) + (\lambda+\zeta)! (\kappa+\lambda)! \times \right. \\ &\quad \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{2^\alpha}{(\lambda+\zeta-\alpha)!} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \times \\ &\quad \left. \frac{\partial_\beta \Theta_{\kappa,\lambda}^-(f_u, -\sqrt{3}f_u) (-2\sqrt{3}f_u)^\beta}{(\alpha+\beta+\kappa+\lambda)!} \times \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases} \right] \quad (36) \end{aligned}$$

and

$$\begin{aligned} \mathcal{F}\{\mathcal{H}(\mathbf{u})u^\mu v^\nu\}(f_u, +\sqrt{3}f_u) &= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{\mu! \nu! (\sqrt{3})^{\mu+1}}{\pi^2 (2\sqrt{3}f_u)^{\mu+\nu+2}} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta} (2\sqrt{3}f_u)^{\kappa+\lambda}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^\kappa} \times \\ &\quad \left[\Theta_{\kappa,\lambda}^-(f_u, \sqrt{3}f_u) + (-1)^{\mu+\nu-\kappa-\lambda-\zeta} (\lambda+\zeta)! (\kappa+\lambda)! \times \right. \\ &\quad \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{2^\alpha}{(\lambda+\zeta-\alpha)!} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \times \\ &\quad \left. \frac{\partial_\beta \Theta_{\kappa,\lambda}^+(f_u, \sqrt{3}f_u) (2\sqrt{3}f_u)^\beta}{(\alpha+\beta+\kappa+\lambda)!} \times \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases} \right] \quad (37) \end{aligned}$$

with the β -derivative of the angular factor given by

$$\begin{aligned} \partial_\beta \Theta_{\kappa,\lambda}^\pm(f_u, \pm\sqrt{3}f_u) &= (\pm\sqrt{3})^\kappa (\pi a)^{\kappa+\lambda+\beta} \\ &\quad \times \begin{cases} \cos(\pm 2\pi a \sqrt{3} f_u) [(-1)^{\frac{\kappa+\lambda+\beta}{2}} - \delta_{\kappa,0} 2^{\lambda+\beta} (-1)^{\frac{\lambda+\beta}{2}}] & \kappa+\lambda+\beta \text{ even} \\ \sin(\pm 2\pi a \sqrt{3} f_u) [(-1)^{\frac{\kappa+\lambda+\beta+1}{2}} - \delta_{\kappa,0} 2^{\lambda+\beta+1} (-1)^{\frac{\lambda+\beta+1}{2}}] & \kappa+\lambda+\beta \text{ odd} \end{cases} \quad (38) \end{aligned}$$

We show in figure 2 a representation of the products $\mathcal{H}(\mathbf{u})u^\mu v^\nu$ and their associated Fourier transforms, calculated from Eqs. (31) to (38), and in figure 3 an example of calculation of a static PSF for a hexagonal telescope with 37-segments with piston and tip-tilt positioning errors.

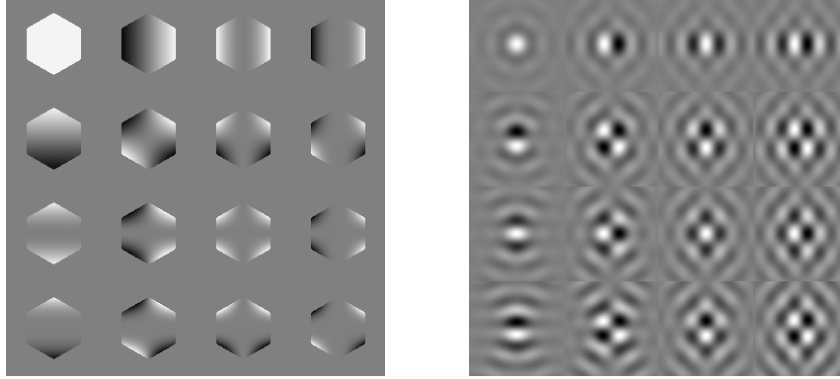


Figure 2: **Left:** products $\mathcal{H}(\mathbf{u}) u^\mu v^\nu$, with $\mu = 0\dots3$ from left to right, and $\nu = 0\dots3$ from top to bottom. **Right:** associated Fourier transform, calculated with the expressions given above.

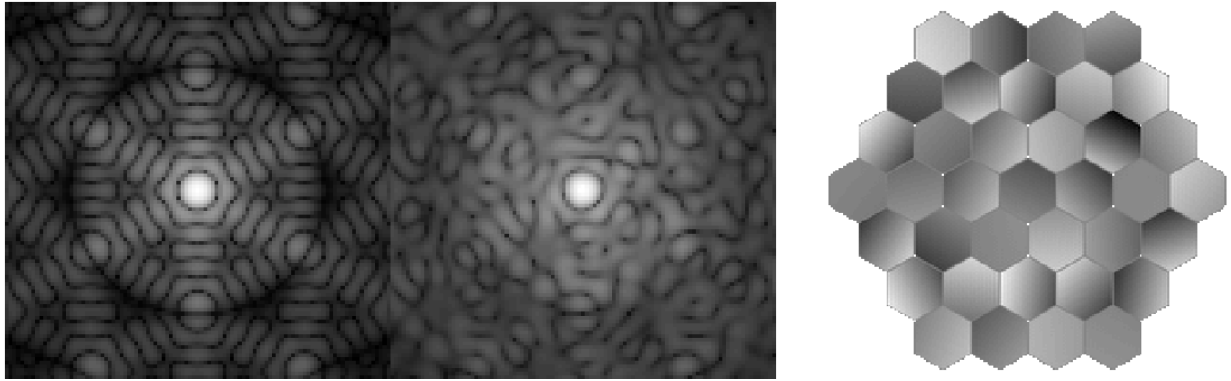


Figure 3: **PSF for a 37-segments hexagonal telescope.** **Left,** no aberrations; **Center,** with 60 nm rms of piston error and 0.04'' rms of tilt error, $\lambda = 1 \mu m$, Strehl=0.81. **Right,** associated segmented phase aberration - calculated from an inverse FFT applied on the amplitude PSF.

3.2 Monolithic secondary mirrors aberrations projected onto segmented primary mirror

In the case of a segmented primary with monolithic secondary mirrors, the total phase aberration is given in our geometrical approach by the sum of each mirror aberration projected onto the segmented pupil. Note that the footprint of the optical beam on secondary mirrors changes with the position of the observed light source, therefore secondary mirror diameters are always larger, when projected back onto the pupil plane, than the pupil diameter - see figure 4.

In this document, when we are referring to the secondary mirror diameter, we are actually referring to the diameter of the back projected beam footprint into the primary mirror plane, where we will assume the entrance pupil is located. In this section, we give the formula for the projection of a secondary mirror phase aberration expressed in Zernike polynomials onto the Zernike basis defined on the segments.

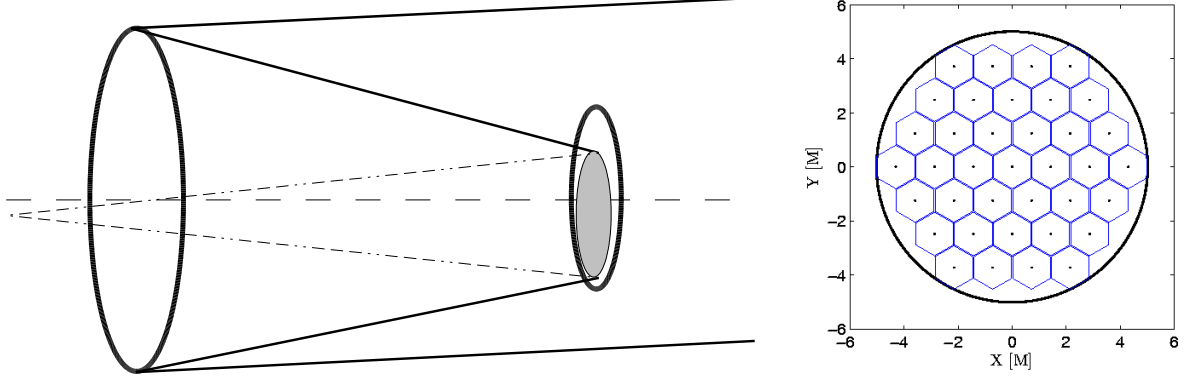


Figure 4: **Left: Footprint of an off-axis beam onto the secondary mirror. Right: secondary mirror footprint projected back on a segmented pupil.**

Let us define $\Pi_{s,\mu,\nu}$ as the coefficient of the projection of the secondary mirrors ν -th Zernike aberration onto the μ -th Zernike polynomial associated to the s -th segment (assumed circular for now),

$$\Pi_{s,\mu,\nu} = \left\langle z_\nu Z_\nu \left(\frac{\mathbf{u}}{R} \right) \middle| Z_\mu \left(\frac{\mathbf{u} - \mathbf{u}_s}{a} \right) \right\rangle \quad (39)$$

where R is the radius of the secondary mirrors, as seen in the pupil plane, a is the segment radius, and $\mathbf{u}_s = (u_s, v_s)$ is the s -th segment center position. Note that the projection of a given mirror mode with j -index ν onto segments modes of higher j -indexes $\mu > \nu$ is necessarily null, therefore $\Pi_{s,\mu,\nu} = 0$ (indeed, one can for example decompose a tip-tilt of the secondary mirrors, at the level of a single segment, as a piston + a tip-tilt, but there cannot be higher order aberrations in this decomposition).

For a set of secondary mirror Zernike aberrations (indexes $\nu_1, \nu_2, \nu_3, \dots$) projected on the segment s , the segment-based μ -th Zernike coefficient is given by the sum over each ν_i index of the projection coefficient,

$$a_{s,\mu} = \sum_{\text{all } \nu_i \geq \mu} \Pi_{s,\mu,\nu_i} \quad (40)$$

and the total phase aberration, for the **circular segment** number s , is

$${}^c\varphi_s(\mathbf{u}) = \sum_{j=1}^{\max \mu} a_{s,\mu} Z_j(\mathbf{u}/a) \quad (41)$$

For **hexagonal segments**, we use exactly the same technique: the projection coefficient is calculated for virtual circular segments circumscribing the hexagons (therefore these virtual circular segments are overlapping), and the total phase aberration into each hexagonal segment is obtained by truncating the total phase aberration on the circular segments with the transmission of the hexagonal segment,

$${}^h\varphi_s(\mathbf{u}) = \sum_{j=1}^{\max \mu} a_{s,\mu} \mathcal{H}(\mathbf{u}) Z_j(\mathbf{u}/a) \quad (42)$$

The calculation of the projection coefficient $\Pi_{s,\mu,\nu \geq \mu}$ is given in the appendix C. We find

$$\begin{aligned} \Pi_{s,\mu,\nu \geq \mu} &= \frac{z_\nu \sqrt{[(m_\mu \neq 0) + 1][(m_\nu \neq 0) + 1](n_\mu + 1)(n_\nu + 1)}}{2\pi R^{n_\nu}} \times \\ &\quad \sum_{s_\mu=0}^{\frac{n_\mu - m_\mu}{2}} \sum_{s_\nu=0}^{\frac{n_\nu - m_\nu}{2}} \sum_{l_\mu=0}^{\frac{n_\mu - m_\mu}{2} - s_\mu} \sum_{l_\nu=0}^{\frac{n_\nu - m_\nu}{2} - s_\nu} \sum_{k_\mu=0}^{q_{j_\mu, m_\mu}} \sum_{k_\nu=0}^{q_{j_\nu, m_\nu}} \binom{m_\mu}{2k_\mu + p_{j_\mu, m_\mu}} \binom{m_\nu}{2k_\nu + p_{j_\nu, m_\nu}} \times \\ &\quad \left(\binom{\frac{n_\mu - m_\mu}{2} - s_\mu}{l_\mu} \right) \left(\binom{\frac{n_\nu - m_\nu}{2} - s_\nu}{l_\nu} \right) \frac{(-1)^{s_\mu + s_\nu + k_\mu + k_\nu} (n_\mu - s_\mu)! (n_\nu - s_\nu)! R^{2s_\nu}}{s_\mu! s_\nu! \left(\frac{n_\mu + m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu + m_\nu}{2} - s_\nu\right)! \left(\frac{n_\mu - m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu - m_\nu}{2} - s_\nu\right)!} \times \\ &\quad \sum_{\alpha=0}^{\omega_\nu} \sum_{\beta=0}^{\varphi_\nu} \binom{\omega_\nu}{\alpha} \binom{\varphi_\nu}{\beta} \frac{a^{\alpha+\beta} [1 + (-1)^{\omega_\mu + \alpha}] [1 + (-1)^{\varphi_\mu + \beta}] \Gamma\left(\frac{\omega_\mu + \alpha + 1}{2}\right) \Gamma\left(\frac{\varphi_\mu + \beta + 3}{2}\right) u_s^{\omega_\nu - \alpha} v_s^{\varphi_\nu - \beta}}{(\varphi_\mu + \beta + 1) \Gamma\left(\frac{\omega_\mu + \varphi_\mu + \alpha + \beta + 4}{2}\right)} \quad (43) \end{aligned}$$

with

$q_{j,m} = \dots$	$m \text{ odd}$	$m \text{ even}$	$m = 0$	and	$p_{j,m} = \dots$	$m \neq 0$	$m = 0$
$j \text{ odd}$	$\frac{m-1}{2}$	$\frac{m-2}{2}$	0		$j \text{ odd}$	1	0
$j \text{ even}$	$\frac{m-1}{2}$	$\frac{m}{2}$			$j \text{ even}$	0	0

and where we have defined

$$\begin{cases} \omega_\mu \equiv n_\mu - 2(s_\mu + l_\mu + k_\mu) - p_{j_\mu, m_\mu} \\ \omega_\nu \equiv n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j_\nu, m_\nu} \\ \varphi_\mu \equiv 2(l_\mu + k_\mu) + p_{j_\mu, m_\mu} \\ \varphi_\nu \equiv 2(l_\nu + k_\nu) + p_{j_\nu, m_\nu} \end{cases} \quad (44)$$

to simplify the notation. Figure 5 shows an example of a continuous secondary mirror phase coma Z_8 projected on the 37-segments pupil, where the segments aberrations have been calculated using Eq. (43).

3.3 Central obscuration and spider shading

Central obscuration and spider shading diffract light and add structure to an already complicated PSF. Here, we show how to include this shading into the calculation of the static PSF, using the Fourier transform approach. The focal plane phasor, as we know, is given by the Fourier transform of the pupil plane phasor, and dropping as usual the constant factor,

$$U_f(\lambda \mathbf{f}) = \mathcal{F}\{U_p(\mathbf{u})\}(\mathbf{f}) \quad (45)$$

Multiplying the pupil plane phasor with the obscuration Ω due to the spider and secondary structures (0 inside the shadow, 1 outside) the previous equation becomes:

$$U_f(\lambda \mathbf{f}) = \mathcal{F}\{\Omega(\mathbf{u}) U_p(\mathbf{u})\} = \tilde{\Omega}(\mathbf{f}) \star \tilde{U}_p(\mathbf{f}) \quad (46)$$

$\Omega(\mathbf{u})$ can be written as the product of the shadowing mask associated to each component of the spider+secondary structure, and each of these components can be decomposed again in a product of

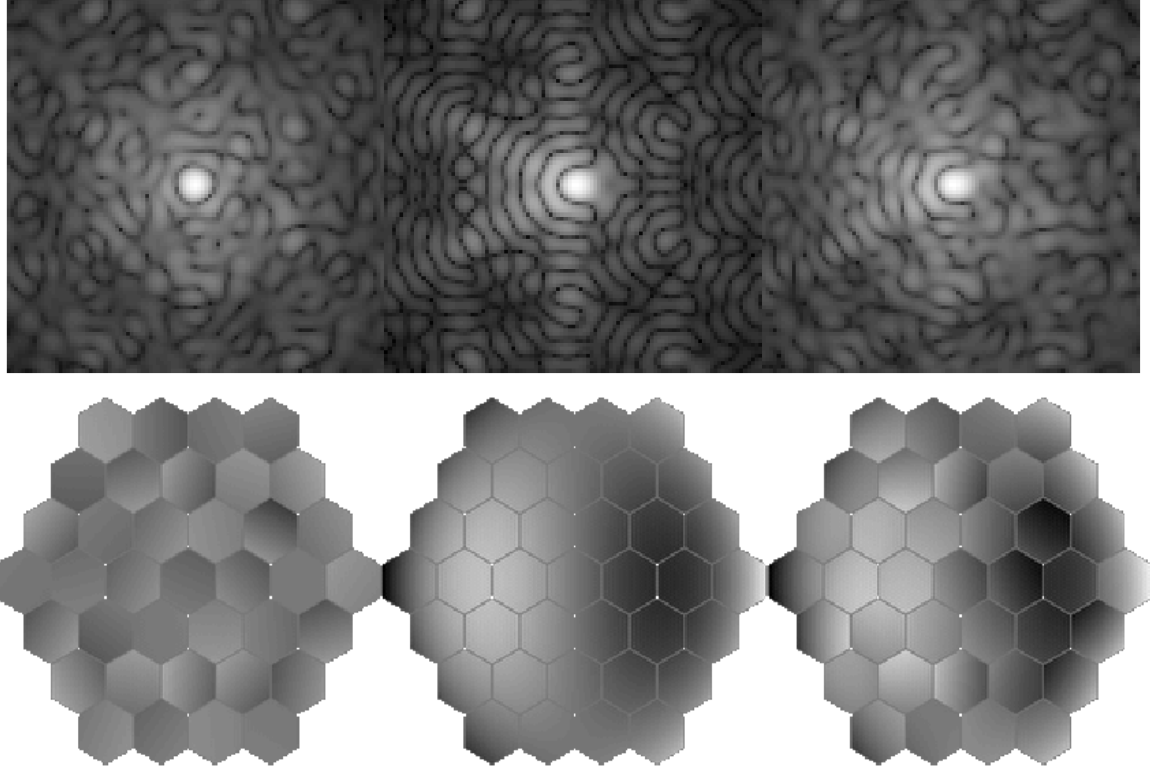


Figure 5: **Top, from left to right: PSF with segments piston and tilt errors (same conditions than figure 3, $\lambda = 1 \mu m$, Strehl=0.81), PSF with a perfect primary and 104 nm of Z_8 coma on the secondary (Strehl=0.79, that would be 0.65 if the projected coma was not truncated by the segmented pupil), PSF when summing the primary and secondary aberration (Strehl=0.64). Bottom: associated phase maps - from IFFT applied on amplitude PSFs.**

simple geometrical shapes - rectangles, triangles, hexagons, and discs. So, $\Omega(\mathbf{u})$ is basically a product of simple geometrical masks, $\Omega(\mathbf{u}) = \prod_i M_i(\mathbf{u})$, and the focal plane phasor becomes a multiple convolution product

$$U_f(\lambda \mathbf{f}) = \tilde{U}_p(\mathbf{f}) \star \prod_i^* \tilde{M}_i(\mathbf{f}) \quad (47)$$

Analytical expressions for simple shapes Fourier transforms are easy to obtain, and the focal plane phasor without obscuration - \tilde{U}_p - is calculated from Eq. (21). Taking the IFFT of the masks FT and focal plane phasor, we go back into the pupil plane where the product $\Omega(\mathbf{u}) = \prod_i M_i(\mathbf{u})$ can be built, and multiplied with U_p . We come back then into the focal plane by applying a direct FFT on the final product ΩU_p . Spider and secondary obscuration are now included in the telescope PSF. An example is shown in figure 6.

Simple shapes Fourier transforms

We collect here the Fourier transforms of simple shapes that might be useful for creating most types of spider shadowing, including support cables (equivalent to narrow rectangles).

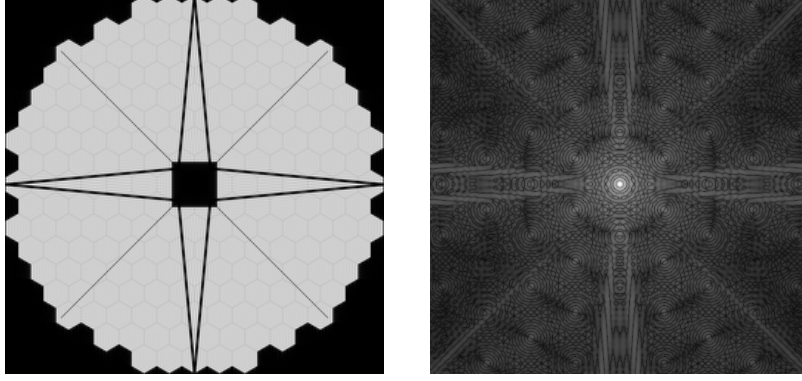


Figure 6: **Left:** A complicated spider with cables and square M2 obscuration, computed using the inverse Fourier transform method described here. M1 is a model of TMT (August 2004 design), the spider is an arbitrary example. Segments gaps (4 mm) appear in grey. **Right:** Logarithmic stretch of the PSF associated to the pupil transmission on the left. Pixel size 1.7 masec, FoV 1.4”.

Centered equilateral triangle, side $2a$

$$\tilde{T}(\mathbf{f}) = \frac{\sqrt{3}f_u [2 \cos^2(\pi a \sqrt{3} f_v) - i \sin(2\pi a \sqrt{3} f_v) - 2 \cos^2(\pi a f_u)] + i 3 f_v \sin(2\pi a f_u)}{\exp(-i 2\pi a / \sqrt{3} f_v) 2 \pi^2 f_u (f_u^2 - 3 f_v^2)} \quad (48)$$

with the limits

$$\tilde{T}(0, 0) = \sqrt{3} a^2 \quad (49)$$

$$\tilde{T}(0, f_v \neq 0) = \frac{2 + i \sin(2\pi a \sqrt{3} f_v) - 2 \cos^2(\pi a \sqrt{3} f_v) - 2 i \pi a \sqrt{3} f_v}{\exp(-i 2\pi a / \sqrt{3} f_v) 2 \pi^2 \sqrt{3} f_v^2} \quad (50)$$

$$\tilde{T}(f_u \neq 0, 0) = \frac{\sqrt{3} [1 - \cos^2(\pi a f_u)]}{\pi^2 f_u^2} \quad (51)$$

$$\tilde{T}(-\sqrt{3} f_v, f_v) = \tilde{T}(+\sqrt{3} f_v, f_v) = \frac{6 i \pi a f_v \exp(-i 2\pi a \sqrt{3} f_v) - i \sqrt{3} \sin(2\pi a \sqrt{3} f_v)}{\exp(-i 2\pi a / \sqrt{3} f_v) 12 \pi^2 f_v^2} \quad (52)$$

Centered rectangle, length a , width b

$$\tilde{R}(\mathbf{f}) = a b \operatorname{sinc}(a f_u) \operatorname{sinc}(b f_v) \quad \text{and} \quad \tilde{R}(0) = a b \quad (53)$$

Centered hexagon, side a See Eq. (96).

Centered disc, radius R

$$\tilde{D}(\mathbf{f}) = R \frac{J_1(2\pi R f)}{f} \quad \text{where} \quad f = \|\mathbf{f}\| \quad \text{and} \quad \tilde{D}(0) = \pi R^2 \quad (54)$$

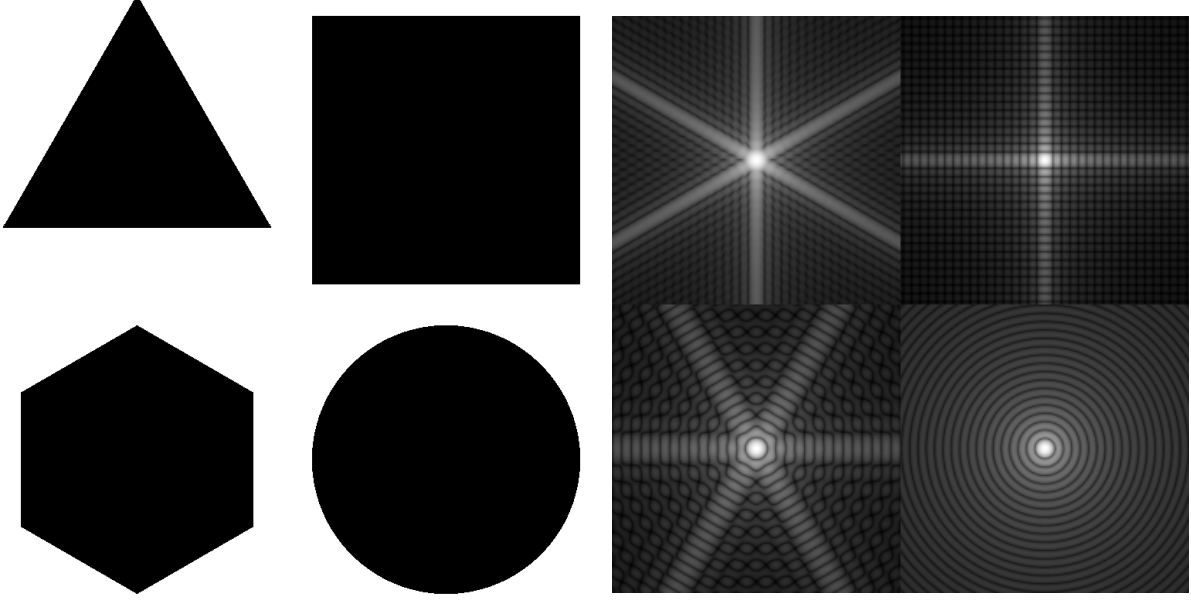


Figure 7: A few simple geometrical shapes, and their Fourier transforms.

4 The dynamic OTF

4.1 Segmented primary mirror structure function

The primary mirror segmented phase is given by

$$\delta\varphi_{\text{M1}}(\mathbf{u}, t) = \sum_{i=1}^{N_S} \delta\varphi_i(\mathbf{u}, t) \star \delta(\mathbf{u} - \mathbf{u}_i) = \sum_{i=1}^{N_S} \delta\varphi_i(\mathbf{u} - \mathbf{u}_i, t) \quad (55)$$

and the phase on a given segment is written as the sum over N_H hexagonal modes $\mathcal{H}_k \equiv \mathcal{H} Z_k$,

$$\delta\varphi_i(\mathbf{u}, t) = \sum_{k=1}^{N_H} a_{i,k}(t) \mathcal{H}_k(\mathbf{r}) \quad (56)$$

Inserting Eqs. (55) and (56) in the definition of the structure function - Eq. (10), we find

$$\begin{aligned} D_{\delta\varphi}^{\text{M1}}(\mathbf{u}, \lambda\nu) &= \sum_{i=1}^{N_S} \sum_{k,l=1}^{N_H} \gamma_{i,i;k,l} [\mathcal{H}_k(\mathbf{u} - \mathbf{u}_i) \mathcal{H}_l(\mathbf{u} - \mathbf{u}_i) + \mathcal{H}_k(\mathbf{u} - \mathbf{u}_i + \lambda\nu) \mathcal{H}_l(\mathbf{u} - \mathbf{u}_i + \lambda\nu)] \\ &\quad - 2 \sum_{i,j=1}^{N_S} \sum_{k,l=1}^{N_H} \gamma_{i,j;k,l} \mathcal{H}_k(\mathbf{u} - \mathbf{u}_i) \mathcal{H}_l(\mathbf{u} - \mathbf{u}_j + \lambda\nu) \end{aligned} \quad (57)$$

where

$$\gamma_{i,j;k,l} \equiv \langle a_{i,k}(t) a_{j,l}(t) \rangle_t \quad (58)$$

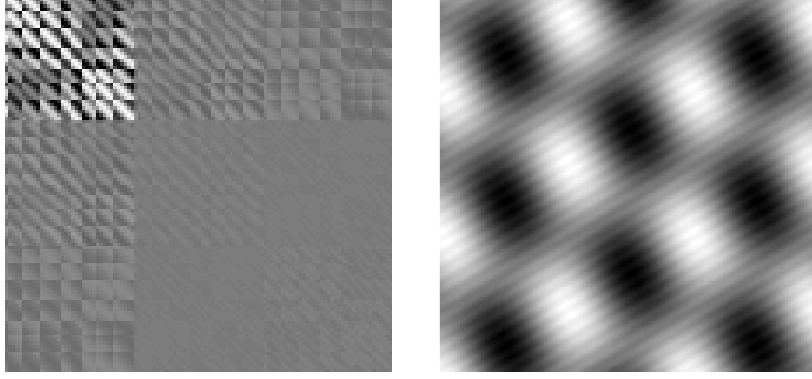


Figure 8: **Left: Segment-to-segment aberration covariance "matrix" $\gamma_{i,j;k,l}$ associated to a set of cosine wave (right figure) traveling across the segmented primary and generating segments piston and tip-tilt. Top blocks: covariance \mathcal{H}_1 with $\mathcal{H}_{1,2,3}$, second line \mathcal{H}_2 with $\mathcal{H}_{1,2,3}$ etc.**

defines the aberrations temporal covariance between segments i & j and modes \mathcal{H}_k & \mathcal{H}_l . The following symmetries are worth noting

$$\begin{cases} \gamma_{i,i;l,k} = \gamma_{i,i;k,l} & \text{for a given segment } i \\ \gamma_{j,i;k,k} = \gamma_{i,j;k,k} & \text{for a given aberration } k \\ \gamma_{j,i;l,k} = \gamma_{i,j;k,l} & \text{switch segments and aberrations} \end{cases} \quad (59)$$

but

$$\begin{cases} \gamma_{i,j;l,k} \neq \gamma_{i,j;k,l} & \text{if } i \neq j \\ \gamma_{j,i;k,l} \neq \gamma_{i,j;k,l} & \text{if } k \neq l \end{cases} \quad (60)$$

For visualization purpose, these covariances can be arranged into a square matrix of $N_H \times N_H$ square blocks of dimension $N_S \times N_S$. An example of such a matrix is shown in figure 8. It corresponds to the piston & tip-tilt aberrations generated by a set of cosine waves, $\sum_i A_i \cos(\mathbf{k}_i \cdot \mathbf{u} + \omega_i t)$, traveling across a 37 segments hexagonal mirror at constant velocities and directions. The covariance indexes symmetry appears clearly with this representation.

We give now the expression for the primary mirror stationary structure function $\mathbb{R} \overline{D}_{\delta\varphi}^{M_1}$. Applying Eq. (15) to Eq. (57), we can see that the stationary structure function is basically a sum of correlation products. In the Fourier space, a correlation product becomes a simple product, and we show in the appendix D that the Fourier transform of the numerator of $\mathbb{R} \overline{D}_{\delta\varphi}^{M_1}$ can be decomposed into a sum of segment modes Fourier transforms (previously given in page 9),

$$\begin{aligned} \mathcal{F}\left\{\text{NUM}\left\{\mathbb{R} \overline{D}_{\delta\varphi}^{M_1}(\lambda\nu)\right\}\right\} &= 2 \sum_{i=1}^{N_S} \sum_{k=1}^{N_H} \mathcal{U}_{i;k}(\mathbf{f}) + 4 \sum_{i=1}^{N_S} \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \mathcal{V}_{i;k,l}(\mathbf{f}) \\ &- 4 \sum_{i=1}^{N_S-1} \sum_{j=i+1}^{N_S} \sum_{k=1}^{N_H} \mathcal{X}_{i,j;k}(\mathbf{f}) - 4 \sum_{i=1}^{N_S-1} \sum_{j=i+1}^{N_S} \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \mathcal{Y}_{i,j;k,l}(\mathbf{f}) \end{aligned} \quad (61)$$

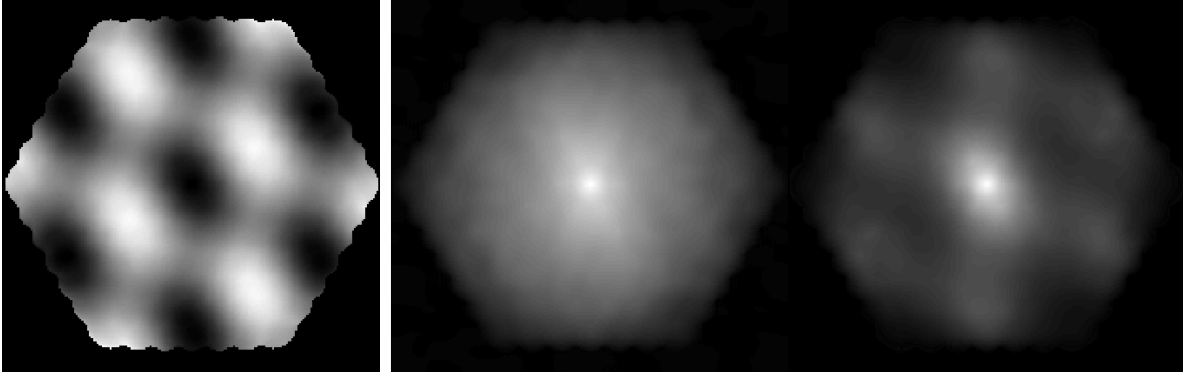


Figure 9: **Left:** Structure function associated to the dynamic cosine wave phase aberration shown in figure 8. **Center:** static aberration OTF associated to the static PSF shown in figure 3. **Right:** same OTF, after multiplication with the cosine wave OTF.

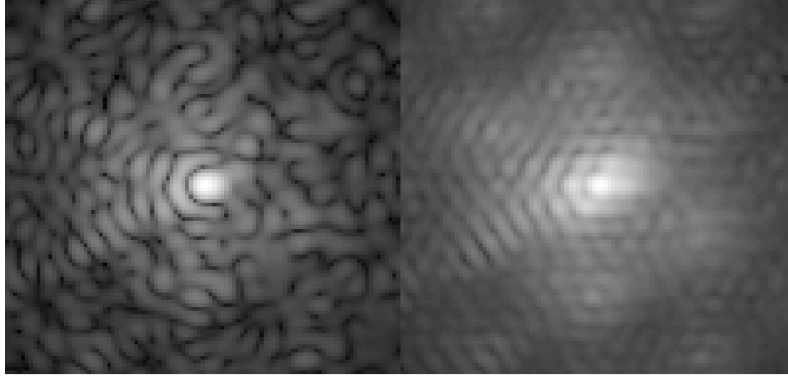


Figure 10: **Effect of the cosine wave aberration on the static PSF. Left:** static PSF (from figure 3, Strehl=0.64); **Right:** long exposure PSF, Strehl=0.27.

where

$$\mathcal{U}_{i;k}(\mathbf{f}) = \gamma_{i,i;k,k} \left[\tilde{P}(\mathbf{f}) \cos(2\pi\mathbf{f}\cdot\mathbf{u}_i) \mathcal{F}\{\mathcal{H}_k^2(\mathbf{u})\} - |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 \right] \quad (62)$$

$$\mathcal{V}_{i;k,l}(\mathbf{f}) = \gamma_{i,i;k,l} \begin{cases} \tilde{P}(\mathbf{f}) \cos(2\pi\mathbf{f}\cdot\mathbf{u}_i) \text{val}\{\mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\} - & m_k + m_l \text{ even} \\ \text{val}\{\tilde{\mathcal{H}}_k(\mathbf{f})\}\text{val}\{\tilde{\mathcal{H}}_l(\mathbf{f})\} & \\ \tilde{P}(\mathbf{f}) \sin(2\pi\mathbf{f}\cdot\mathbf{u}_i) \text{val}\{\mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\} & m_k + m_l \text{ odd} \end{cases} \quad (63)$$

$$\mathcal{X}_{i,j;k}(\mathbf{f}) = \gamma_{i,j;k,k} \cos[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 \quad (64)$$

$$\mathcal{Y}_{i,j;k,l}(\mathbf{f}) = \text{val}\{\tilde{\mathcal{H}}_k(\mathbf{f})\} \text{val}\{\tilde{\mathcal{H}}_l(\mathbf{f})\} \begin{cases} (\gamma_{i,j;k,l} + \gamma_{i,j;l,k}) \cos[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k + m_l \text{ even} \\ (\gamma_{i,j;k,l} - \gamma_{i,j;l,k}) \sin[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ odd}, m_l \text{ even} \\ (-\gamma_{i,j;k,l} + \gamma_{i,j;l,k}) \sin[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ even}, m_l \text{ odd} \end{cases} \quad (65)$$

and where $\text{val}\{*\}$ is an operator defined, for a complex function G , by

$$\text{val}\{G(\mathbf{f})\} \equiv \Re\{G(\mathbf{f})\} + \Im\{G(\mathbf{f})\} = \begin{cases} \Re\{G(\mathbf{f})\} & \text{if } G \text{ is purely real} \\ \Im\{G(\mathbf{f})\} & \text{if } G \text{ is purely imaginary} \end{cases} \quad (66)$$

with \Re and \Im the real and imaginary part operators. As we can see, forcing stationarity across the segmented primary did not remove the segment numbering: stationarity is assumed on the segments only, but not for the correlations between segments, and there can still be inhomogeneous dynamical perturbations across the primary mirror.

What we need now is an expression for the denominator of ${}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_1}$, given by the autocorrelation of the telescope pupil - see Eq. (15). Again, the pupil autocorrelation can be easily calculated analytically in the Fourier space, and we find

$$\mathcal{F}\left\{\text{DEN}\left\{{}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\boldsymbol{\nu})\right\}\right\} = |\widetilde{\mathcal{H}}_1(\boldsymbol{\nu})|^2 \left\{ N_S + 2 \sum_{i=1}^{N_S-1} \sum_{j=i+1}^{N_S} \cos[2\pi\boldsymbol{\nu}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \right\} \quad (67)$$

Once the Fourier transform of the numerator and denominator of the structure function have been calculated using the previous expressions, we get ${}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_1}$ by application of the inverse FFT algorithm

$${}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\boldsymbol{\nu}) = \frac{\text{IFFT}\left\{\mathcal{F}\left\{\text{NUM}\left\{{}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\boldsymbol{\nu})\right\}\right\}\right\}}{\text{IFFT}\left\{\mathcal{F}\left\{\text{DEN}\left\{{}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\boldsymbol{\nu})\right\}\right\}\right\}} \quad (68)$$

In the figure 9, we show an example of the effect of a dynamic aberration on the long exposure PSF. The dynamic aberration was modeled as a set of cosine waves, the same as shown in figure 8. The spatial scaling is the same between the cosine wave image and the structure function image, and as we can expect, the structure function has its maximal value for a separation distance in the pupil equal to half of the main perturbation wavelength, and in the same direction. The structure function is bounded into a hexagonal domain because it cannot be defined for pupil plane separations larger than the telescope pupil, in any direction. The effect on the static PSF is shown in figure 10: the blurring of the PSF at high frequency appears clearly.

4.2 Secondary mirrors stationary structure function

To compute the secondary mirrors structure functions, we develop the phase aberration in terms of Zernike polynomials,

$$\delta\varphi_{\text{M}_2}(\mathbf{u}, t) = \sum_{k=1}^{N_Z} a_k(t) Z_k(\mathbf{u}) \quad (69)$$

and we find,

$$D_{\varphi}^{\text{M}_2}(\mathbf{u}, \lambda\boldsymbol{\nu}) = \sum_{k,l=1}^{N_Z} \zeta_{k,l} [Z_k(\mathbf{u})Z_l(\mathbf{u}) - 2Z_k(\mathbf{u})Z_l(\mathbf{u} + \lambda\boldsymbol{\nu}) + Z_k(\mathbf{u} + \lambda\boldsymbol{\nu})Z_l(\mathbf{u} + \lambda\boldsymbol{\nu})] \quad (70)$$

where

$$\zeta_{k,l} \equiv \langle a_k(t) a_l(t) \rangle_t \quad (71)$$

defines the covariance of the secondary mirrors aberrations, a symmetric matrix, $\zeta_{k,l} = \zeta_{l,k}$. Applying Eq. (15) to Eq. (70), we get, - see appendix E,

$$\begin{aligned} \mathcal{F}\left\{\text{NUM}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_2}(\lambda\nu)\right\}\right\} &= 2\sum_{k=1}^{N_Z}\zeta_{k,k}\left[\mathcal{F}\{Z_k^2(\mathbf{u})\}\tilde{P}(\mathbf{f}) - |\tilde{Z}_k(\mathbf{f})|^2\right] \\ &+ 4\sum_{k=1}^{N_Z-1}\sum_{l=k+1}^{N_Z}\zeta_{k,l}\begin{cases} \tilde{P}(\mathbf{f})\text{val}\{\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\}\} - \text{val}\{\tilde{Z}_k(\mathbf{f})\}\text{val}\{\tilde{Z}_l(\mathbf{f})\} & m_k+m_l \text{ is even} \\ 0 & m_k+m_l \text{ is odd} \end{cases} \end{aligned} \quad (72)$$

The denominator of the secondary stationary structure function, as defined in Eq. (15), is the circular pupil auto-correlation, and can be calculated directly, using - see for instance Schroeder [7]

$$\text{DEN}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_2}(\lambda\nu)\right\} = \frac{D_p^2}{2}\left[\arccos\left(\frac{\lambda\nu}{D_p}\right) - \left(\frac{\lambda\nu}{D_p}\right)\sqrt{1 - \left(\frac{\lambda\nu}{D_p}\right)^2}\right] \quad (73)$$

4.3 Geometrical matrices

At this point, we can see that the calculation of the static and long exposure OTF involves in one hand the telescope static aberrations and covariances $\gamma_{i,j;k,l}$ and $\zeta_{i,j}$, and on the other hand some geometrical objects - pupil Fourier transform, modes products Fourier transforms, etc - that do not depend on the aberration amplitudes, but only on the segment positions and modes indexes. These will be called the **geometrical matrices**, and if we examine carefully the structure functions expressions, we can see that these geometrical matrices can be restricted to three kind:

1. the pupil Fourier transform \tilde{P} , for segmented and circular pupil transmissions;
2. the segments modes product Fourier transform $\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}$;
3. the secondary mirrors modes product Fourier transform $\mathcal{F}\{Z_kZ_l\}$.

and as any product of modes can be projected onto the modes themselves, modes product Fourier transforms can be computed from the single modes Fourier transforms - $\mathcal{F}\{\mathcal{H}_k\}$, $\mathcal{F}\{Z_k\}$ - given page 8,

$$\mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\} = \sum_{j=1}^{f(k,l)} p_{k,l,j}\tilde{\mathcal{H}}_j(\mathbf{f}) \quad (74)$$

$$\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\} = \sum_{j=1}^{f(k,l)} p_{k,l,j}\tilde{Z}_j(\mathbf{f}) \quad (75)$$

and the formula to compute the projection coefficients $p_{k,l,j}$ is given in appendix F. Finally, the segmented pupil transmission Fourier transform is easily calculated from

$$\tilde{P}(\nu) = \tilde{\mathcal{H}}_1(\nu)\sum_j \exp(i2\pi\nu\cdot\mathbf{u}_j) \quad (76)$$

and for the secondary mirrors footprint, from Eq. (54)

$$\tilde{P}(\nu) = R_p\frac{J_1(2\pi R_p\nu)}{\nu} \quad \text{where } \nu = \|\nu\| \quad \text{and } \tilde{P}(0) = \pi R_p^2 \quad (77)$$

5 The adaptive optics spatial frequency filter model

The approach we are using here has been initially developed for modeling AO correction of the atmospheric turbulence phase (Jolissaint et al. [2]), but can be adapted to any type of phase aberration, in particular telescope aberrations. It relies on an analytical representation of the AO correction as a spatial frequency filter in the telescope pupil plane: AO correction is modeled by applying this linear filter to the phase Fourier transform. Let us describe in details the principle behind the construction of the AO filter.

The AO corrected phase is given by the difference between the aberrated phase φ and its best estimate $\hat{\varphi}$,

$$\varphi_{\text{ao}}(\mathbf{u}, t) = \varphi(\mathbf{u}, t) - \hat{\varphi}(\mathbf{u}, t) \quad (78)$$

The phase estimate is never perfect, though, and is corrupted by five main error terms (common to any AO system):

1. **the fitting error** - the AO system cannot correct phase aberrations at spatial frequencies above the AO system half spatial sampling frequency (or WFS cut-off frequency) $f_{\text{WFS}} = 1/(2\Lambda_{\text{WFS}})$ where Λ_{WFS} is the WFS sampling interval (e.g. the lenslets pitch for a SH-WFS) so high order modes aberrations are not corrected. In practice, this term is by far the most important one of the residual phase; of course, the DM also has a cut-off frequency, defined by $f_{\text{DM}} = 1/(2\Lambda_{\text{DM}})$ where Λ_{DM} is the actuator pitch: in general, the DM pitch matches the WFS lenslets pitch, so $f_{\text{WFS}} = f_{\text{DM}} \equiv f_{\text{AO}}$. When it is not the case, either the DM or the WFS sets the AO system's cut-off frequency $f_{\text{AO}} = \min\{f_{\text{WFS}}, f_{\text{DM}}\}$;
2. **the WFS spatial aliasing**: due to its finite sampling frequency, the high frequency phase is seen by the WFS as an aliased low frequency phase, and these corrupts the measurement of the actual low frequency phase component; this component is in general the second most important source of error;
3. **the servo-lag error**, due to the AO system time-lag and limited WFS temporal sampling frequency;
4. **the WFS noise term**, due to the photon noise and WFS detector noise, which is essentially a low spatial frequency term;
5. **the anisoplanatism error**, but for telescope aberrations, there is no such an error, because the pupil plane errors are by nature isoplanatic as seen from AO FoV (for large FoV, this might not be the case).

In the case of AO correction of telescope aberration, it is reasonable to neglect the servo-lag error, as the AO system is optimized for the correction of atmospheric turbulence phase aberrations, which are generally much faster than the rather slow varying telescope optics errors. For that same reason, WFS noise can be neglected, as we can assume that the system has more time to compensate for telescope errors (than for turbulence-based ones), and can use several loops even at relatively low SNR to converge.

Assuming stationarity of the DM correction, and splitting the telescope phase aberration into its low and high spatial frequency components φ_l and φ_h (below and above f_{AO}), we find that the residual phase Fourier transform is given by

$$\tilde{\varphi}_{\text{ao}}(\mathbf{f}, t) = \tilde{\varphi}_h(\mathbf{f}, t) + [1 - \gamma_{\text{DM}}(\mathbf{f})]\tilde{\varphi}_l(\mathbf{f}, t) - \gamma_{\text{DM}}(\mathbf{f})\tilde{\mathcal{A}}\{\tilde{\varphi}_h(\mathbf{f}, t)\} \quad (79)$$

where γ_{DM} is the DM spatial transfer function, \mathcal{A} the aliasing filter operator, and $\tilde{\mathcal{A}}$ its Fourier transform.

Let us discuss the DM transfer function first. An ideal DM would be able to reproduce perfectly the phase aberration below the DM cut-off frequency, and transmits nothing above it

$$\gamma_{\text{DM}}^{\text{ideal}} = \begin{cases} 1 & |f_u| < f_{\text{DM}} \text{ and } |f_v| < f_{\text{DM}} \\ 0 & |f_u| \geq f_{\text{DM}} \text{ or } |f_v| \geq f_{\text{DM}} \end{cases} \quad (80)$$

This perfect transfer function is represented by a square of width $2f_{\text{DM}}$ in the spatial frequency domain. Its corresponding ideal response (response to a Dirac- δ type phase aberration) is therefore a sinc function in both directions, see Eq. (53),

$$\mathcal{F}^{-1}\{\gamma_{\text{DM}}^{\text{ideal}}\} = \frac{\text{sinc}(u/\Lambda_{\text{DM}})\text{sinc}(v/\Lambda_{\text{DM}})}{\Lambda_{\text{DM}}^2} \quad (81)$$

What would be now the transfer function of a real, non ideal DM ? If the set of the DM influence functions is known (they can be measured) we can compute the response of the DM by projecting the δ phase aberration onto the DM influence function basis, and take the Fourier transform of that to get the DM spatial transfer function. Figure 11, left, shows such a response function, for a Xinetics Inc. DM (the influence function was taken from high resolution measurements on Altairs's DM - Gemini North AO system). The DM response function closely resemble a sinc function, which is not surprising, and is significantly narrower than the IF itself (figure 11, right).

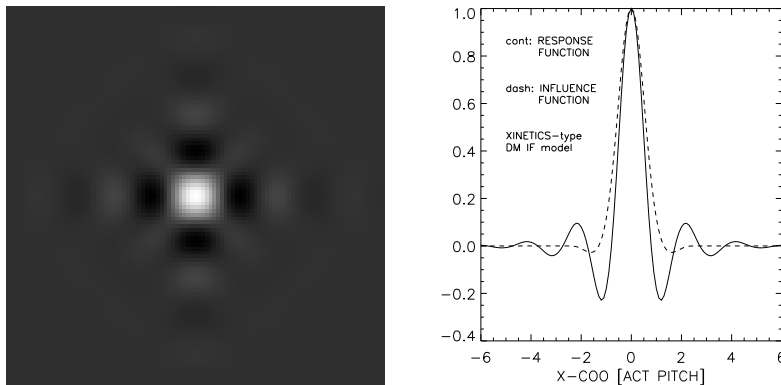


Figure 11: **Left: Xinetics Inc. deformable mirror response function, computed from laboratory measurement of the DM influence functions; Right: DM response and influence functions compared, same Xinetics DM.**

Figure 12, left, shows the Fourier transform of the previous DM response. The graph on the right compares two DM models transfer functions (Xinetics and Boston MicroMachines MEMS DMs) with the perfect transfer function. One can see that taking the perfect square filter as an approximation of the real DM transfer function is roughly acceptable - at first order - as long as the phase aberration is mainly concentrated below or above the DM cutoff frequency. Now, let us introduce the aliasing filter \mathcal{A} . Obviously, it is not a linear filter: it is not possible to write the aliased phase as a convolution product, and from [2], we find that the aliased phase Fourier transform, when servo-lag error can be

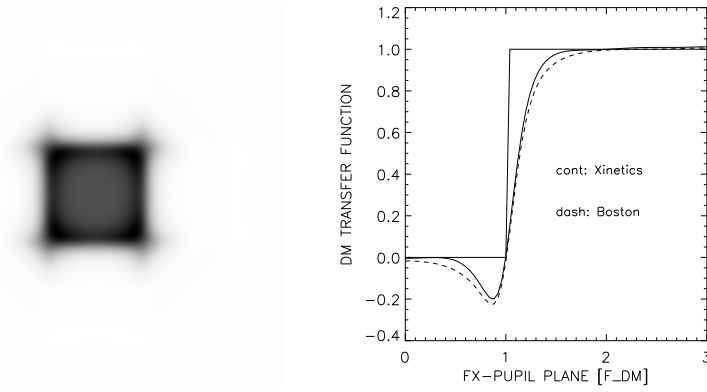


Figure 12: **Left: Xinetics Inc DM transfer function; Right: Xinetics and Boston MM DM transfer functions fx-profiles (actually, $1 - \gamma_{DM}^{ideal}$) compared to the ideal step transfer function .**

neglected, is given by

$$\tilde{\mathcal{A}}\{\tilde{\varphi}_h\} = \frac{f_u f_v}{f_u^2 + f_v^2} \sum_{\substack{k,l=-\infty, \\ \neq 0}}^{\infty} (-1)^{k+l} \left(\frac{f_u}{f_v - \frac{l}{\Lambda_{WFS}}} + \frac{f_v}{f_u - \frac{k}{\Lambda_{WFS}}} \right) \tilde{\varphi}_h \left(f_u - \frac{k}{\Lambda_{WFS}}, f_v - \frac{l}{\Lambda_{WFS}} \right) \quad (82)$$

Practically, the sum can be limited over a few indexes only, up to a spatial frequency that includes most of the phase spectrum. For telescope aberrations, we practically find that limiting the sum over -/+ three indexes is generally sufficient.

AO filtering of the static modes

The procedure to numerically compute the AO filtered amplitude PSF is the following (step-by-step),

1. compute the segment modes Fourier transforms using the analytical expressions in Eq. (28) or Eq. (31), depending on the type of segment;
2. from the modes Fourier transforms, compute the segmented phase Fourier transform, with (here for hexagonal segments)

$$\tilde{\delta\varphi}(\mathbf{f}) = \sum_{s=1}^{N_S} \sum_{j=1}^{N_H} z_{s,j} \exp(-i 2\pi \mathbf{f} \cdot \mathbf{u}_s) \tilde{\mathcal{H}}_j(\mathbf{f}) \quad (83)$$

3. compute the AO filtered phase Fourier transform $\tilde{\delta\varphi}_{ao}$ by applying the AO filter on the segmented phase Fourier transform, using an ideal or realistic DM transfer function (which has to be provided by other means, in the later case), plus WFS aliasing;
4. compute analytically the segmented pupil Fourier transform, using first part of Eq. (21), and the segment transmission Fourier transform given by Q_1 or $\tilde{\mathcal{H}}_1$;
5. compute numerically the segmented pupil transmission by applying the IFFT algorithm on $\tilde{P}(\mathbf{f})$;

6. compute numerically the AO corrected segmented phase by applying the IFFT algorithm on $\widetilde{\delta\varphi_{ao}}(\mathbf{f})$;
7. compute numerically the focal plane amplitude PSF by applying a direct FFT on the product $P(\mathbf{u}) \exp[-i\delta\varphi_{ao}(\mathbf{u})]$.

In figure 13, we show the effect of the AO filter on the static PSF and phase that were shown in figure 3. The actuator pitch was set to the segment size (perfect DM, no aliasing), and we can see on the phase images and x-profile that the low spatial frequencies aberration (initially set on the secondary mirror) have been well corrected (the coma aberration does not appear anymore) but not so much the segment aberrations, which are at higher spatial frequencies. The correction of the coma term is also well apparent in the PSF (bottom figure).

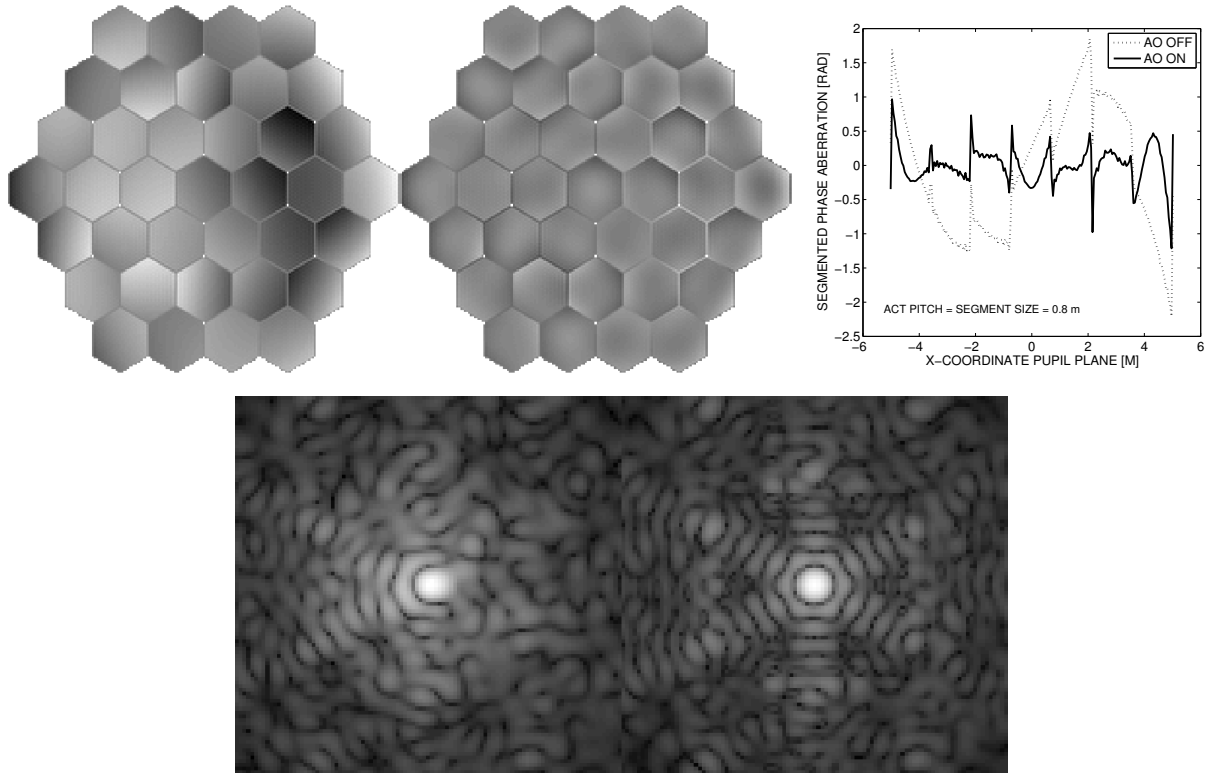


Figure 13: **Top:** segmented phase aberration before and after AO correction; **top right:** x-cut across the phase; **bottom:** static PSF before (Strehl=0.64) and after AO correction (Strehl=0.87).

AO filtering of the dynamic modes

AO correction of the dynamic aberrations is modeled by replacing the segments modes Fourier transforms $\tilde{\mathcal{H}}_i$ by their AO-filtered version, (here for hexagonal segments)

$$\tilde{\mathcal{H}}_{j,\text{ao}}(\mathbf{f}) = \tilde{\mathcal{H}}_{j,h}(\mathbf{f}) + [1 - \gamma_{\text{DM}}(\mathbf{f})]\tilde{\mathcal{H}}_{j,l}(\mathbf{f}) - \gamma_{\text{DM}}(\mathbf{f})\tilde{\mathcal{A}}\{\tilde{\mathcal{H}}_{j,h}(\mathbf{f})\} \quad (84)$$

and the geometrical matrices have to be replaced with

$$\mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\} \longrightarrow \mathcal{F}\{\mathcal{H}_{k,\text{ao}}(\mathbf{u})\mathcal{H}_{l,\text{ao}}(\mathbf{u})\} = \text{FFT}\left\{\text{IFFT}\{\tilde{\mathcal{H}}_{k,\text{ao}}(\boldsymbol{\nu})\}\text{IFFT}\{\tilde{\mathcal{H}}_{l,\text{ao}}(\boldsymbol{\nu})\}\right\} \quad (85)$$

$$\tilde{\mathcal{H}}_k^*(\boldsymbol{\nu})\tilde{\mathcal{H}}_l(\boldsymbol{\nu}) \longrightarrow \tilde{\mathcal{H}}_{k,\text{ao}}^*(\boldsymbol{\nu})\tilde{\mathcal{H}}_{l,\text{ao}}(\boldsymbol{\nu}) \quad (86)$$

For the secondary mirrors structure functions, the procedure is exactly the same, replacing the hexagonal segments modes by the Zernike modes Z_j .

In the figures 14-16 we show the effect of AO correction on the dynamic case example shown in figure 9. In the structure function, the low spatial frequency modulation has disappeared, and what is left is the high frequency structure due to segments aberrations - the little hexagonal shadow insert shows the segment transmission, at the same scale than the structure function. It can be shown that the structure function saturates at twice the value of the phase variance for large pupil plane vector separation. This is well apparent here in the x-cut of the structure functions before and after AO correction (horizontal dashed lines). It is worth noting too, on the AO corrected PSF, the square domain centered on the PSF core: if one recalls the relationship between the pupil plane spatial frequency and the focal plane angular position, one can understand that this square domains corresponds to the spatial frequencies domain that have been filtered out by the DM transfer function.

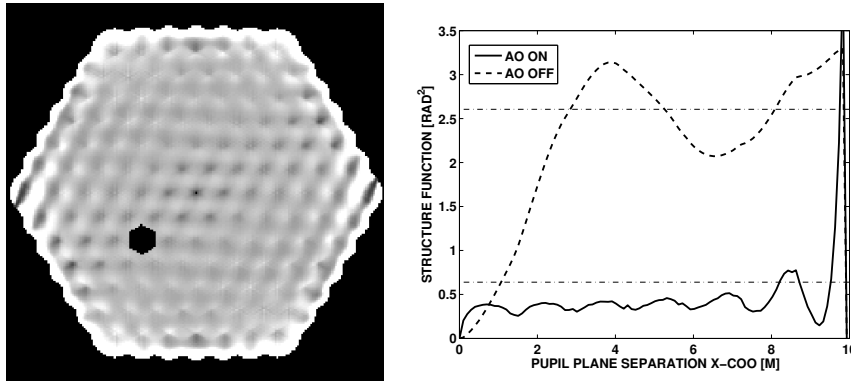


Figure 14: **Left:** structure function after AO correction. The hexagonal insert shows the segment transmission at the same scale than the structure function. **Right:** cut across the x-coordinates of the structure function before and after AO correction. The horizontal dashed lines show the theoretical saturation value of the structure function at twice the phase variance. The divergence of the AO corrected structure function at the edge (close to 10 m) is a numerical effect due to IFFT algorithm around sharp boundaries.

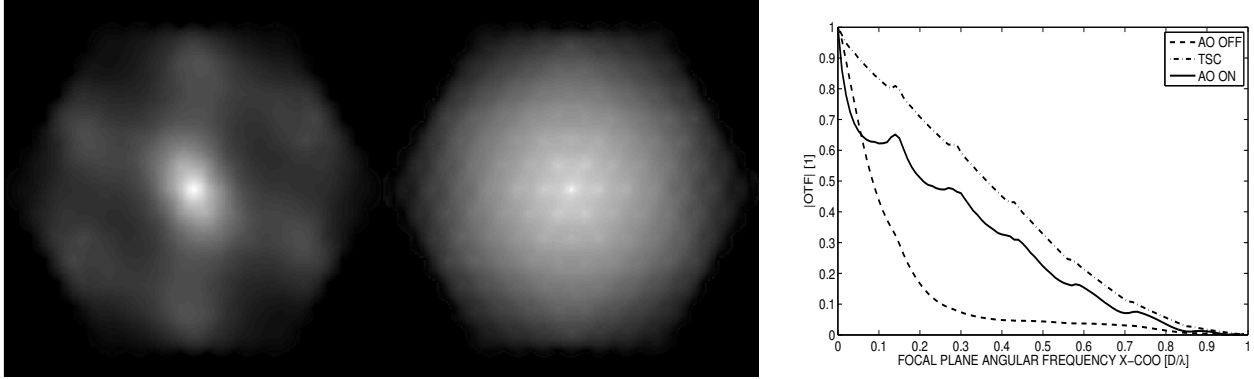


Figure 15: Left and center: OTF before and after AO correction. Right: OTF profiles across ν_x .

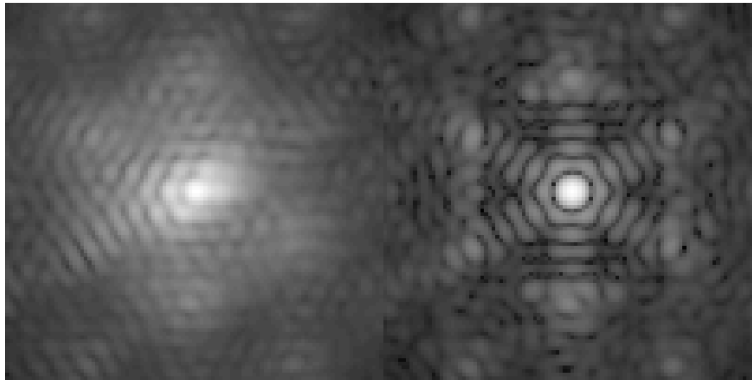


Figure 16: PSF before (Strehl=0.27) and after (Strehl=0.73) AO correction.

6 The code OPTICA, user manual

OPTICA is a MATLAB-based code implementing the method presented in the previous sections. The first version of the toolbox has been released in February 2007. Here, we describe the main functions and their usage, with examples. A user manual for every function of the toolbox (whether a main or secondary function) is provided in the header of the functions, that can be accessed from MATLAB with the command

```
>> doc <function name>.m
```

6.1 Algorithm, flow chart

The algorithm of OPTICA is shown in figure 17, and the functions associated to each task are listed in each block. OPTICA is not a single function, but a set of tools, and is organized around five main blocks: (1) definition of the telescope optics architecture and loading of calculation parameters - first "line" of the flow chart, (2) calculation of the geometrical matrices - 2nd line, (3) calculation of the static PSF/OTF - 3rd line, (4) calculation of the dynamic OTF - 3rd line, (5) calculation of the final PSF/OTF - last line. Operations (2) to (5) are embedded in a single function - `optica.m`.

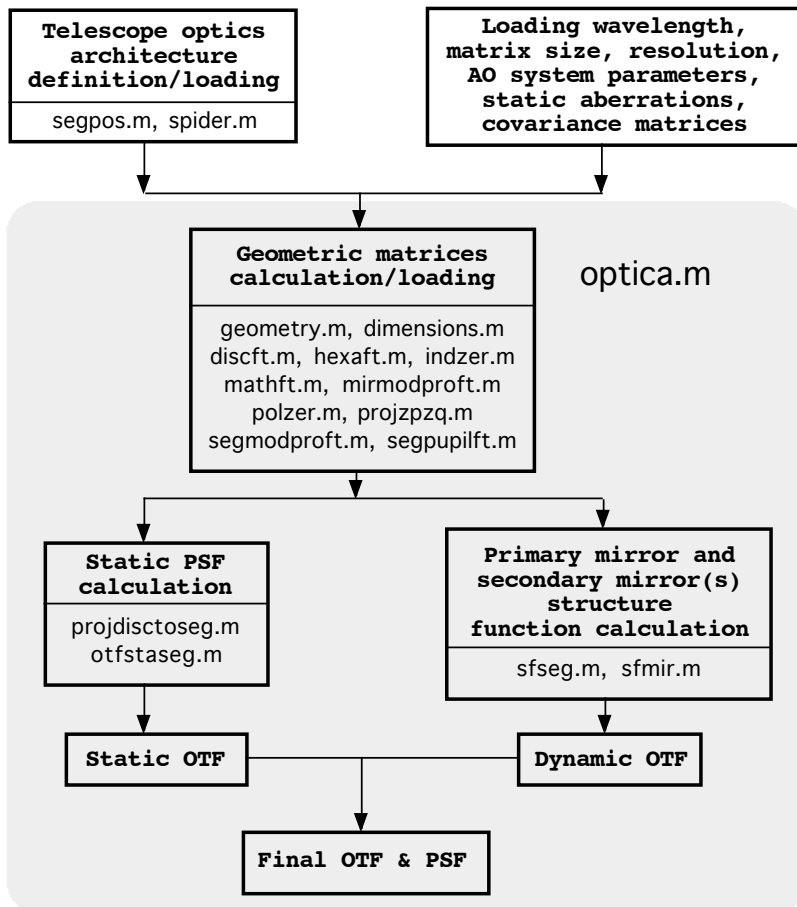


Figure 17: OPTICA toolbox algorithm.

6.2 The functions of the toolbox, and their purpose

We give here a brief explanation of what are each function used for. User's manuals for each function are provided starting page 50.

<code>dimensions.m</code>	built pupil and focal plane position ($\boldsymbol{\alpha}$, \mathbf{u}) and frequency (\mathbf{f} , $\boldsymbol{\nu}$) arrays
<code>discft.m</code>	Fourier transform of a disc, from analytical expression (no FFT) - Eq. (54)
<code>geometry.m</code>	built geometrical matrices - see section 4.3
<code>hexaft.m</code>	Fourier transform of $\mathcal{H}(\mathbf{u}) u^\mu v^\nu$ - Eqs. (31)-(38)
<code>indzer.m</code>	Zernike polynomial indexes n, m from index j
<code>mathft.m</code>	FFT with amplitude scaling and origin positioning to match Eqs. (1)-(2)
<code>mirmodproft.m</code>	secondary mirror(s) modes products Fourier transforms - $\mathcal{F}\{Z_k Z_l\}$
<code>optica.m</code>	the main function , calculation of static and dynamic OTFs
<code>otfstaseg.m</code>	computes the static amplitude PSF, static phase map, static OTF
<code>polherft.m</code>	Fourier transform of the hexagonal modes, $\mathcal{F}\{\mathcal{H} Z_k\} = \mathcal{F}\{\mathcal{H}_k\} = \tilde{\mathcal{H}}_k$
<code>polzerft.m</code>	Fourier transform of the Zernike polynomials, Eq. (28)
<code>projdisctoseg.m</code>	projection of a secondary Zernike aberration on the segments modes - Eq. (43)
<code>rectft.m</code>	Fourier transform of the rectangle. No FFT, analytic.
<code>segmodproft.m</code>	segments modes products Fourier transforms - $\mathcal{F}\{\mathcal{H}_k \mathcal{H}_l\}$
<code>segpos.m</code>	telescope optics architecture definition
<code>segpupilft.m</code>	segmented pupil transmission Fourier transform - Eq. (21)
<code>sfmir.m</code>	secondary mirror stationary structure function - Eq. (72)
<code>sfseg.m</code>	segmented pupil stationary structure function - Eqs. (61) to (67)
<code>spider.m</code>	secondary mirror support structure definition
<code>triaft.m</code>	Fourier transform of the equilateral triangle. No FFT, analytic.
<code>zpzqtozj.m</code>	Zernike polynomials product projected into the Zernike basis, Eqs. (190)-(191)

and here are two other functions, not used in the algorithm, but that the user may find useful:

<code>covwave.m</code>	covariances $\gamma_{i,j;k,l}$ associated to a set of cosine waves across the segmented primary
<code>encene.m</code>	OTF-based encircled energy calculation

6.3 FIRST STEP : defining the telescope optics architecture - functions `segpos.m` and `spider.m`

The first step in any OPTICA run is to define the telescope optics architecture - mirrors diameters, segment positions etc... using the function `segpos.m`. Optionally, a spider shadowing structure can be added to the telescope optics architecture, using the function `spider.m`. Here is a copy of the header (user manual) of the function `segpos.m`:

```
FUNCTION mir=segpos(DEXTFLAT,DEXTCTR,DINT,NSD,GAP,VIEW,FNAME)
```

```
    segpos computes the coordinates (centre) of each segment of
    a hexagonal segmented mirror with an outer diameter DEXTFLAT and an
    inner diameter DINT, with precisely NSD segments/diameter,
    and segments gaps=GAP. These numbers are used to define the
    structure of the pupil transmission, and are used as an input of
    the function geometry.m.
```

```
    segpos.m is part of the toolbox OPTICA.
```

EXAMPLE

```
dext=10;
dctr=10;
dint=0;
nsd=7;
gap=0.04;
mir=segpos(dext,dctr,dint,nsd,gap,'yes','test/mir_test.mat');
```

```
mir content:
```

```
mir =
    nseg: 37
      swd: 1.39428571428571
      ssz: 0.80499123247011
      dbs: 1.43428571428571
    segpos: [2x37 double]
dextflat: 10
dextctr: 10
dextmax: 10.03234822383841
  dint: 0
   gap: 0.0400000000000000
   nsd: 7
  surf: 62.29252154348677
```

INPUTS name | type | unit

DEXTFLAT | double scalar | METERS

Primary mirror diameter, flat to flat. This is actually not the maximum distance between most opposed segments. Such maximal diameter is calculated within the code - see OUTPUT mir.dextmax.

DEXTCTR | double scalar | METERS

Primary mirror diameter taken as measured from segment center rather than segment flats. Segments whose centers are outside of this diameter are excluded from the mirror.

This input is used to define the diameter of a disc whom each segment whose center lies outside of it are excluded. Playing with that parameter, one can remove external segments to generate a more "smooth" or circular pupil shape. Setting DEXTCTR=DEXTFLAT is a good default value.

DINT | double scalar | METERS

Central obscuration diameter.

NSD | long scalar | -

segments across primary mirror diameter, flat-to-flat. Must be an ODD number.

GAP | double scalar | METERS

Gap between the segments.

OPTIONAL INPUTS name | type | unit

VIEW | character string | -

enter 'yes' if you want to see a drawing of the mirror, and 'no' if you don't.

FNAME | character string | -

The mirror structure variable will be saved in the file FNAME. If not set, the mirror structure variable is saved in the file './mir.mat', by default.

OUTPUTS name | type | unit

mir is a structure variable, with components:

.dextflat | scalar | METERS

copy of DEXTFLAT input.

.dextctr | scalar | METERS

copy of dextctr input.

.dextmax | scalar | METERS

maximal diameter, between most opposed segment corners.

.dint | scalar | METERS

central obscuration diameter.

.nsd | scalar | 1

of segments along the mirror diameter.

.gap | scalar | METERS

gap between segments (smaller distance if disc segments).

.nseg | scalar | 1

total number of segments

.ssz | scalar | METERS

segment radius, i.e. half of point-to-point diameter, also equal to the side length of the segment (1/6 of perimeter).

.swd | scalar | METERS

segment width (hexagon inner diameter, flat-to-flat).

.dbs | scalar | METERS

distance between segments centre to centre.

.segpos | array(2,nb segments) | METERS

x and y segment center coordinates.

.surf | scalar | METERS²

total surface of the segmented mirror.

ASSOCIATED FUNCTIONS/PROCEDURES

None.

An example

In the figure 18 we show the example of the definition of a segmented mirror with 19 segments across the mirror, for a total of 306 segments. The function call, and the content of the resultant mirror structure variable, are

```
>> mir=segpos(10,10,1.5,19,0.04,'yes','test/mir_test.mat');
>> mir

mir =

      nseg: 306                number of segments
      swd: 0.48842105263158    segment width flat-to-flat
      ssz: 0.28199002621472    segment radius
      dbs: 0.52842105263158    distance between segments
      segpos: [2x306 double]    segments center position
      dextflat: 10              diameter flat-to-flat
      dextctr: 10               "exclusion" diameter
      dextmax: 10.2206962555381 diameter point-to-point
      dint: 1.50000000000000    central obscuration diameter
      gap: 0.040000000000000    segments gaps
      nsd: 19                   number of segment along dextflat
      surf: 63.21800823484919   total surface of mirror
```

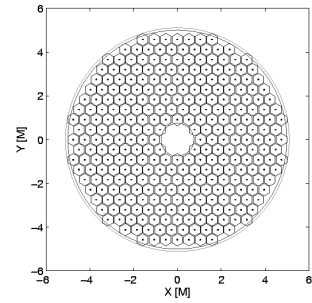


Figure 18: A 306 segments primary mirror.

The mirror structure variable was saved in the file `test/mir_test.mat`. When re-running an OTF calculation on the same mirror architecture, there is no need to recompute the mirror structure variable, and it can be loaded using

```
>> load test/mir_test.mat
```

Adding a spider to the mirror structure variable - function `spider.m`

Once the segmented pupil architecture has been defined with `segpos.m`, spider structure parameters can be added to the mirror structure variable, with `spider.m`. This function simply reads the spider structure parameters in input, process them, and add them to the pre-existing mirror structure variable. These spider parameters will then be used in the function `optica.m` to add the effect of spider shading on the static PSF, following the procedure described in section 3.3.

It is assumed here that the shadow cast by the spider mechanical structure is made of two components: the secondary mirrors support, and the spider's arms and tension cables. Therefore, two structure variables inputs are needed to specify the spider shadow: the secondary mirror one, and the spider's arms one. The function call syntax is

```
>> mir=spider(mir,m2shadow,arms,view,fname);
```

`mir` is the segmented pupil architecture structure variable, build with `segpos.m`, `m2shadow` is the secondary mirror input, `arms` the spider's arms parameters input, `view` a tag to be set to 'yes' if the user wants to see a drawing of the new pupil with spider superimposed on it, and `fname` is there to specify a file name to save the new mirror structure variable. If no file name is specified, the new

mirror structure variable is saved, by default, in the file `mirwithspider.mat`. The components of the secondary mirror shadow structure variable are:

```
>> <name>.shape : shape of M2 support shadow, options are:
    'tria' : an equilateral triangle, barycentre on the telescope optical axis
    'squa' : a square
    'hexa' : a hexagon
    'disc' : a disc
>> <name>.size : size of the M2 support shadow cast on the primary mirror
    if 'tria' : triangle basis
    if 'squa' : square side
    if 'hexa' : hexagon diameter, point-to-point
    if 'disc' : disc diameter
>> <name>.angle : orientation angle of the shadow relative to the x-axis
```

where `name` can be any character string. The components of the arms structure variable are:

```
>> <name>.width : arms or cables widths
>> <name>.cooint : x/y-coordinate of the inner attachment point,
    a (2-by-number of arms/cables) array
>> <name>.cooext : x/y-coordinate of the outer attachment point,
    a (2-by-number of arms/cables) array
```

where again `name` can be any character string.

From these parameters, `spider.m` computes the arms/cables lengths, the coordinate of the arms/cables centers, and the arms/cables orientation relative to the horizontal axis. Here is a complete example, with a intentionally *ridiculous* spider structure, in order to highlight the versatility of the method:

```
>> dext=10;
>> dctr=10;
>> dint=1;
>> nsd=7;
>> gap=0.04;
>> mir=segpos(dext,dctr,dint,nsd,gap,'yes');
>> shadow.shape='tria';
>> shadow.size=3;
>> shadow.angle=20;
>> arm.width=[0.1 0.2 0.3 0.4];
>> arm.cooint=zeros(2,4);
>> arm.cooext=zeros(2,4);
>> arm.cooint(1,:)= [0.5 0 -0.5 0];
>> arm.cooint(2,:)= [1 0.5 0 -0.5];
>> arm.cooext=arm.cooint*10;
>> arm.cooint(1,4)=-1;
>> newmir=spider(mir,shadow,arm,'yes')
```

```

newmir =

    nseg: 36
    swd: 1.39428571428571
    ssz: 0.80499123247011
    dbs: 1.43428571428571
    segpos: [2x36 double]
    dextflat: 10
    dextctr: 10
    dextmax: 10.03234822383841
    dint: 1
    gap: 0.040000000000000
    nsd: 7
    surf: 60.60893988014930
    m2shape: 'tria'
    m2size: 3
    m2angle: 20
    armwidth: [1x4 double]
    armlength: [1x4 double]
    armangle: [63.43494882292201 90 0 -77.47119229084849]
    armcenter: [2x4 double]

```

In figure 19, we show from left to right (1) a drawing of the mirror with spider structure, (2) the associated pupil image and (3) the associated PSF. Diffraction of the light perpendicularly to the spider's arms is well apparent in the later image.

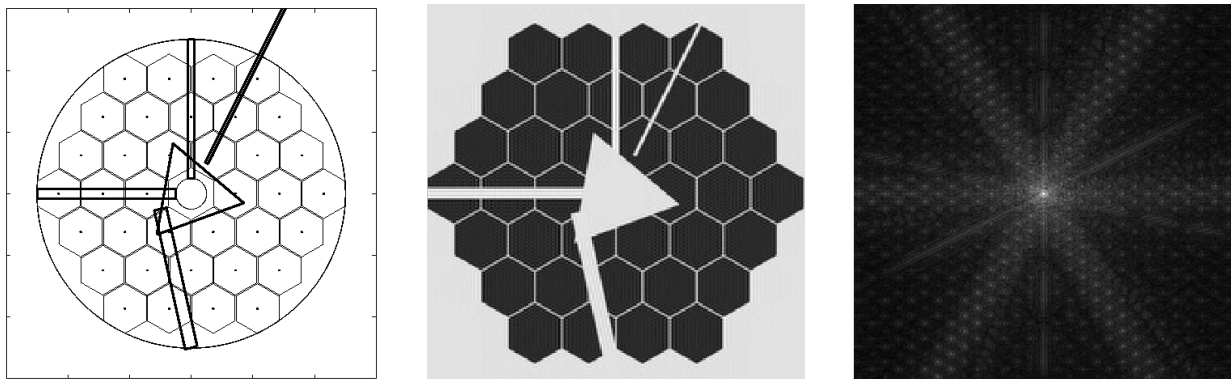


Figure 19: A rather improbable spider structure, and its effect on the telescope PSF. From left to right: mirror drawing with spider, pupil generated with `segpupilft.m`, associated PSF.

6.4 SECOND STEP : loading and formatting the static aberrations and the dynamic covariances input variables

We assume now that the mirror structure variable has been constructed and is available at the main level of the MATLAB workspace. Next step is to load the aberrations and dynamic covariances data (output of telescope mechanical structure modeling, or/and air flow analysis models) and format them as structure variables to be readable by the main code, `optica.m`. The structure of all aberration/covariance variables must be the following:

```
<name>.ind : Zernike polynomials j-indexes - can be given in any order,  
<name>.mat : static aberrations coefficients, or covariance matrix,  
<name>.type : type of data in <name>.mat.
```

<name> can be any string. There can be four types of aberration data:

6.4.1 Static aberration on the primary

<name>.type must be set to 'staM1', and <name>.mat must be given as a 2D array of dimension

```
size(<name>.mat) = (number of segments, number of modes)
```

The syntax must be as given in the following example

```
>> aM1.ind=[2,3,7,8,11];  
>> load('a1abb.mat') % we assume that we load a Nseg-by-5 matrix -> tmp  
>> aM1.mat=tmp; % UNIT MUST BE RADIAN OF PHASE (NOT SURFACE)  
>> aM1.type='staM1';
```

When the variable `aM1` will be read by `optica.m`, the code will look at the data type, and here it will take this input as a set of aberrations coefficients on the segments. The compatibility with the number of segments in the mirror structure variable will be checked, as well as the compatibility between the length of <name>.ind and the 2nd dimension of <name>.mat. And as there is - of course - only one segmented primary,

There can be only one input of type 'staM1'

6.4.2 Static aberration on any of the secondary surfaces

<name>.type must be set to 'staM2', and <name>.mat must be given as a vector of dimension

```
size(<name>.mat) = (1, number of modes)
```

and the syntax must be

```
>> aM2.ind=[5,6,7,8,11];  
>> load('a2abb.mat') % we assume that we load a 5-elements row vector -> tmp  
>> aM2.mat=tmp; % UNIT MUST BE RADIAN OF PHASE (NOT SURFACE)  
>> aM2.type='staM2';
```

again, internal consistency between <name>.ind and <name>.mat sizes is checked within `optica.m`. Now, as the number of secondary, continuous optical elements (surfaces or lenses) can be arbitrary, there can be also an arbitrary number of type 'staM2' inputs. This allows the user to specify aberrations of other type than surface errors, like for instance misalignment coma, or off-axis aberrations. For instance, assume that there is a 20 nm off-axis astigmatism in the x-direction, at 1 μm ,

```

>> astOffAxis.ind=[5];
>> a5=20;
>> lambda=1;
>> astOffAxis.mat=a5*1e-9*2*pi/(lambda*1e-6);
>> astOffAxis.type='staM2';

```

and so on, if there are other sources of static aberrations. In the function `optica.m`, all type 'staM2' inputs will be merged, and projected onto the primary segments hexagonal basis, using the method described in section 3.2, Eq. (43).

6.4.3 Dynamic aberration on the primary, covariance array $\gamma_{i,j;k,l}$

<name>.type must be set to 'dynM1', and <name>.mat must be given as a 4-D array, of size

```
size(<name>.mat) = (N seg, N seg, N M1 dyn modes, N M1 dyn modes)
```

with the syntax

```

>> cM1.ind=[1,2,3];
>> load('covM1.mat') % we assume that we load a 3 x 3 x Nseg x Nseg array
>> cM1.mat=tmp;
>> cM1.type='dynM1';

```

size compatibility is checked, and again as there is only one segmented primary mirror,

There can be only one input of type 'dynM1'

6.4.4 Dynamic aberration on the secondary mirrors, covariance matrices $\zeta_{k,l}$

<name>.type must be set to 'dynM2', and <name>.mat must be given as a 2-D array, of size

```
size(<name>.mat) = (N M2 dyn modes, N M2 dyn modes)
```

and the syntax is the following

```

>> cM2.ind=[6,5,8,7,11,10,9,12,13,16,17,22,14,15,18,19,24,23,30,29,37];
>> load('covM2.mat') % we load a 21-by-21 covariance matrix -> tmp
>> cM2.mat=tmp;
>> cM2.type='dynM2';

```

size compatibility is checked as usual. As for secondary mirrors static aberrations, there can be an arbitrary number of type 'dynM2' inputs. Also, the covariance matrices are merged into one, defining a unique so-called M2 covariance matrix, structure function, and OTF.

To summarize, after this process of aberrations/covariance array loading, our worksheet example (batch file content) would look like that:

```

% tsc optics definition
mir=segpos(10,10,1.5,19,0.04,'test/mir_test.mat','yes');
% static on M1
aM1.ind=[2,3,7,8,11];
load('a1abb.mat')

```

```

    aM1.mat=tmp;
    aM1.type='staM1';
% static on M2
    aM2.ind=[5,6,7,8,11];
    load('a2abb.mat')
    aM2.mat=tmp;
    aM2.type='staM2';
% off-axis astigmatism
    astOffAxis.ind=[5];
    a5=20;
    lambda=1;
    astOffAxis.mat=a5*1e-9*2*pi/(lambda*1e-6);
    astOffAxis.type='staM2';
% dynamic on M1
    cM1.ind=[1,2,3];
    load('covM1.mat')
    cM1.mat=tmp;
    cM1.type='dynM1';
% dynamic on M2
    cM2.ind=[6,5,8,7,11,10,9,12,13,16,17,22,14,15,18,19,24,23,30,29,37];
    load('covM2.mat')
    cM2.mat=tmp;
    cM2.type='dynM2';

```

6.5 THIRD STEP : calculation of the static and dynamic OTFs - function `optica.m`

So now we have the telescope optics defined, and lots of different type of aberrations. Let us compute the OTF/PSF, with `optica.m`, the main code of the toolbox. The calling syntax is

```
>> res=optica(geofname,res,[mir,npix,lambda,actpitch],{aberrations data});
```

and the inputs definitions are:

`geofname` : file name to store the geometrical matrices, for future usage, of type 'mat', like 'geo.mat';

`res` : to get a pixel size in the focal plane smaller than the usual Nyquist limit $\lambda/2D$ by a factor `res`, i.e. $dx = \lambda/2D/res$;

`mir` : telescope optics architecture structure variable;

`npix` : general matrix size - all 2D pupil/focal plane quantities are defined on `npix-by-mpix` matrices, except for the PSFs that are defined on `res*npix-by-res*npix` matrices;

`lambda` : optical wavelength, unit is micrometers;

`actpitch`: AO DM actuator pitch, unit is meters. **In the present version of the toolbox (February 2007), AO correction is simply modeled as an ideal DM transfer function (square filtering, without WFS aliasing.** If AO correction modeling is not required, this input must be set to "-1".

The inputs between square brackets [*mir,npix,lambda,actpitch*] are optional, and the inputs between curved brackets {*aberrations data*} are the static aberrations and dynamic covariance.

As we already know, two types of objects are needed for the calculation of the static/dynamic PSF/OTF: the static aberrations coefficients and dynamic covariances in one side, and the geometrical matrices on the other side - see page 4.3. These matrices depend on the four optional inputs [*mir,npix,lambda,actpitch*] plus the aberration/covariance indexes <name>.ind. Therefore, if one only wants to change the aberrations amplitude, there is no need to recompute the geometrical matrices all over again, and these can be stored on memory for subsequent use. The calculation of the geometrical matrices is done within the main code *optica.m*, by the function *geometry.m*. In the example below, the geometrical matrices do not exist and have to be generated, so the calling syntax to use is

```
>> res=optica('geo.mat',res,mir,npix,lambda,actpitch,{aberrations data});
```

the geometrical matrices are written in a structure variable named *geo*, saved in the file '*geo.mat*'. In the next example, the geometrical matrices do exist, and the inputs *mir,npix,lambda,actpitch* have been removed, prompting the code to look for the file *geo.mat* and load its content

```
>> res=optica('geo.mat',res,{aberrations data});
```

Looking at the content of the structure variable *geo* is interesting. It contains valuable informations on the optical system - like the pupil mask - that can be useful for debugging, or post-processing analysis. Here is a copy of the content of *geo* for an OPTICA run on a 37-segments mirror, with AO correction of static aberrations on M1 and M2 and dynamic aberrations on M1 (this is the example that was shown in page 26, figures 14-16),

```
>> load mat/geo_test_200_aoon_am1_am2_cm1.mat
>> geo
```

```
geo =
```

lambda: 1	wavelength in microns
npix: 200	general matrix size
dxfov: 0.01027998638230	focal plane pixel scale [asec]
segpos: [2x37 double]	segments position
ssz: 0.80499123247011	segments radius
dextmax: 10.03234822383841	max mirror diameter
r_focus: [1x1 struct]	focal plane coordinates
f_focus: [1x1 struct]	focal plane angular frequency
r_pupil: [1x1 struct]	pupil plane coordinates
f_pupil: [1x1 struct]	pupil plane spatial frequency
M1pupil: [200x200 double]	segmented pupil transmission
M1pupilft: [200x200 double]	segmented pupil Fourier transform
M1pupcor: [200x200 double]	segmented pupil autocorrelation
dmpitch: 0.80499123247011	AO system's DM actuator pitch [m]
dynM1modft: [3x200x200 double]	dynamic M1 modes Fourier transforms
dynM1modproft: [6x200x200 double]	dynamic M1 modes products FT
dynM1ind: [1 2 3]	dynamic modes j-indexes

```

staM1modft: [8x200x200 double] static M1 modes Fourier transforms
staM1ind: [1 2 3 4 5 6 7 8] static modes j-indexes

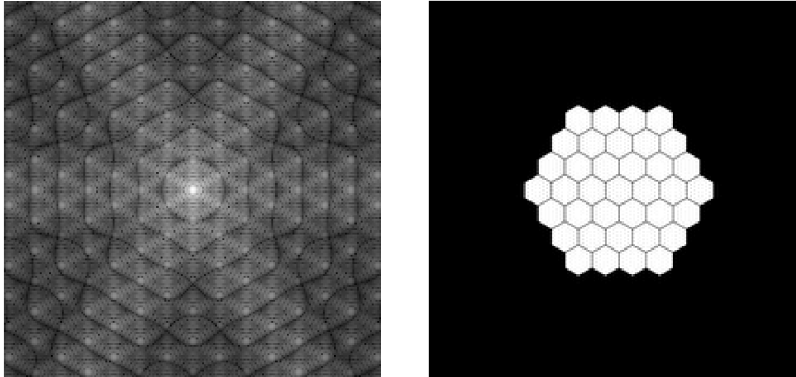
```

and here is for instance how the pupil Fourier transform `geo.M1pupilft` and the pupil transmission `geo.M1pupil` look like - remember that we got the transmission from applying an IFFT on the analytically calculated pupil Fourier transform:

```

>> imagesc(abs(geo.M1pupilft).^0.125)
>> imagesc(geo.M1pupil)

```

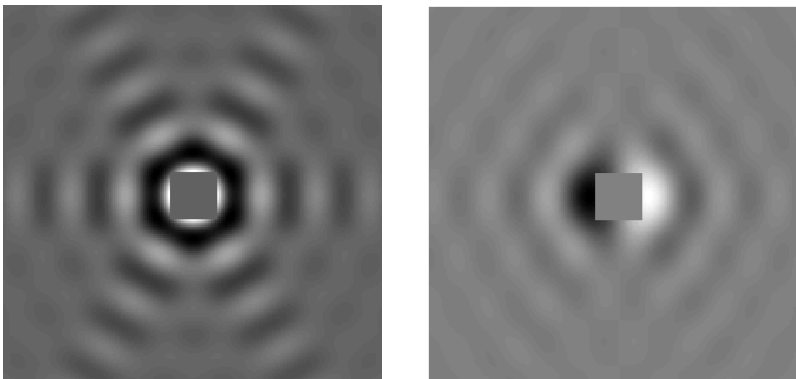


and here is how the Fourier transform of the static hexagonal modes $\tilde{\mathcal{H}}_1(\mathbf{f})$ and $\tilde{\mathcal{H}}_8(\mathbf{f})$ look like with AO filtering of the low spatial frequencies (square in the middle):

```

>> imagesc(squeeze(real(geo.staM1modft(1, :, :))))
>> imagesc(squeeze(imag(geo.staM1modft(8, :, :))))

```



Let us come back now on the usage of the main function `optica.m`. At this point, our worksheet would look like that (if geometrical matrices have to be generated)

```

% tsc optics definition
mir=segpos(10,10,1.5,19,0.04,'test/mir_test.mat','yes');
% static on M1
aM1.ind=[2,3,7,8,11];

```

```

load('a1abb.mat')
aM1.mat=tmp;
aM1.type='staM1';
% static on M2
aM2.ind=[5,6,7,8,11];
load('a2abb.mat')
aM2.mat=tmp;
aM2.type='staM2';
% off-axis astigmatism
astOffAxis.ind=[5];
a5=20;
lambda=1;
astOffAxis.mat=a5*1e-9*2*pi/(lambda*1e-6);
astOffAxis.type='staM2';
% dynamic on M1
cM1.ind=[1,2,3];
load('covM1.mat')
cM1.mat=tmp;
cM1.type='dynM1';
% dynamic on M2
cM2.ind=[6,5,8,7,11,10,9,12,13,16,17,22,14,15,18,19,24,23,30,29,37];
load('covM2.mat')
cM2.mat=tmp;
cM2.type='dynM2';
% optica run
dxres=2;
npix=200;
lambda=1;
actpitch=mir.ssz; %actuator pitch set to segment size
results=optica('geo.mat',dxres,mir,lambda,actpitch,aM1,aM2,astOffAxis,cM1,cM2);

```

So, what did optica.m compute for us ? Plenty of useful things:

```
>> results
```

```
results =
```

sfM1: [200x200 double]	M1 structure function
sfM2: [200x200 double]	M2 structure function
otftsc: [200x200 double]	perfect telescope OTF
psftsc: [400x400 double]	perfect telescope PSF
otfstata: [200x200 double]	static OTF
apsfstata: [200x200 double]	static amplitude PSF
stapha: [200x200 double]	total segmented static phase
psfstata: [400x400 double]	static intensity PSF
otfM1: [200x200 double]	M1 dynamic OTF= $\exp(-\text{sfM1}/2)$
otfM2: [200x200 double]	M2 dynamic OTF= $\exp(-\text{sfM2}/2)$

```
    otf: [200x200 double]    total OTF=OTF(sta)*OTF(M1)*OTF(M2)
    psf: [400x400 double]    final PSF = IFFT{OTF}
    strehl: 0.72638600889398  final Strehl ratio
```

from there, any kind of post-processing/diagnostic tool can be applied to get an assessment of the telescope optical performance: for instance we can use the function `encene.m` on the final OTF to get an encircled energy curve, or any PSF full-width at half maximum (FWHM) code, or anything else. Post processing of the PSF/OTF is not included in the toolbox. To finish, here is a copy of the header of the function `optica.m`

```
FUNCTION results=optica(variable number of arguments)
```

```
optica.m is the main function of the software package OPTICA, and is
used to compute, amongst other things, the long exposure PSF/OTF of
a hexagonal segmented telescope with static and dynamic aberrations,
on both the segmented primary and secondary mirror(s) - M2,M3 etc -
including, as an option, adaptive optics (AO) correction of these
static and dynamic errors (deformable mirror fitting error and
wavefront sensor spatial aliasing).
```

```
The number of inputs is not fixed, and depends on the calculation
option, and on the number of optical surfaces for which static
or dynamic aberrations are specified.
```

USAGES

```
You have two type of inputs to optica:
```

- (1) geometric parameters and matrices (mirrors, AO system ...),
- (2) static aberrations and dynamic aberrations covariances, for the segmented primary and secondary mirror(s),

```
and two different ways to use the code:
```

FIRST USAGE

```
The geometric matrices have already been calculated,
and are simply loaded from a file. If the geometric parameters of
the studied case are constant (same mirrors, same aberration indexes,
same wavelength, same AO correction), and only the amplitude of the
aberrations is changing, then this option is the recommended way to
use the code:
```

```
results=optica('geo.mat',2,A_M1,A_M2,A_M3,...,C_M1,C_M2,C_M3,...);
    where, in this example,
    > 'geo.mat' is the file name where the geometric parameters and
        matrices are stored;
```

- > 2 = the focal plane pixel size resolution - actually 2 means that you want the pixels = (Nyquist limit)/2;
- > A_Mn are the static aberrations Zernike coefficients of mirror n, and there can be as many different mirrors as you want; BUT ONLY ONE FOR THE SEGMENTED PRIMARY
- > C_Mm are the dynamic aberrations covariance for mirror n, and there can be again as many different mirrors with dynamic aberrations as you want, independently of the number of mirrors with static aberrations, BUT ONLY ONE FOR THE SEGMENTED PRIMARY. See INPUTS below for details on how to enter the aberrations data.

SECOND USAGE

is for when the geometric matrices need to be calculated according to new geometric parameters (suppose for instance that you want to see what is the effect of segment size for a particular set of aberrations), then stored for future usage:

```
results=optica('geo.mat',2,MIR,NPIX,LAMBDA,ACTPITCH,
              A_M1,A_M2,A_M3,...,C_M1,C_M2,C_M3,...);
```

- where the additional inputs - relative to the FIRST USAGE, are
- > MIR is a structure variable where the parameters of the primary segmented mirror are stored (segments position, size etc...) and is the output of the function segpos.m;
 - > NPIX is the general size of the square matrices used along the calculation - with the exception of the PSF matrices, whose size depends also on the resolution factor (e.g. if, as above, res=2, then npix(PSF)=2*given npix;
 - > LAMBDA is the optical wavelength;
 - > ACTPITCH is the AO system deformable mirror actuator pitch.

INPUTS FIRST USAGE name | type | units

THE FOLLOWING ORDER OF INPUT APPEARANCE MUST BE IMPERATIVELY RESPECTED:

```
optica(GEOfILENAME,RES,aberrations or covariance, in any order)
```

GEOfILENAME | string | -

Name of the file where the geometric matrices and geometric parameters are stored or should be stored after being built.

RES | real scalar > or = 1 | 1

focal plane pixel size / maximum Nyquist focal plane pixel size. The higher RES, the smaller the pixel size: $DX=DX(Nyquist)/RES$.

MUST BE > or = 1. When choosing RES > 1, the matrix size NPIX is

increased accordingly, (NPIX->newNPIX) and the Field-of-View, defined when the geometrical matrices were calculated, stays the same.

There can be an arbitrary number of static aberrations specified at the input of optica - but only one of type 'segmented primary'. The syntax to enter a static aberration is the following:

Assume that the root name of the static aberration is A. The following components must appears in AB:

AB.type | string | -

Type of static aberration.

'staM1' for the primary, THERE CAN BE ONLY ONE INPUT OF THAT TYPE.

'staM2' for secondary mirrors.

AB.ind | real array (nb modes) | rad

Segments static phase aberration Zernike indexes, following

Noll's numbering convention.

AB.mat | real array (nb segments,nb M1 modes) or (nb M2 modes) | rad

Segments static phase aberration Zernike coefficients, following

Noll's numbering & normalization convention.

Example:

```
>> aM1.ind=[2,3,7,8,11];
```

```
>> load('a1abb.mat') loading a (nseg,5) array -> tmp
```

```
>> aM1.mat=tmp;
```

```
>> aM1.type='staM1';
```

There can be an arbitrary number of dynamic aberrations covariance matrices specified at the input of optica - but again only one for the segmented primary. The syntax to enter a dynamic covariance matrix is the following:

Assume that the root name of the covariance matrix is COV. The following components must appears in COV:

COV.type | string | -

Type of dynamic aberration.

'dynM1' for the primary, THERE CAN BE ONLY ONE INPUT OF THAT TYPE.

'dynM2' for secondary mirrors.

COV.ind | real array (nb modes) | rad

Dynamic phase aberration Zernike indexes, following Noll's numbering convention.

COV.mat | real array (nseg,nseg,nbM1modes,nbM1modes) or
or (nbM2modes,nbM2modes) | rad²
Dynamical aberrations covariance matrix. For the segmented primary, a
4-D array: first & second indexes are for the segments numbering, and
third and fourth indexes are for aberrations numbering. For instance,
C_5,6,7,5 is the covariance between segments 5 and 6 and
respectively 7th and 5th aberrations. For secondary mirrors, a 2-D
array.

Example:

```
>> cM1.ind=[1,2,3];
>> load('covM1.mat') -> tmp
>> cM1.mat=tmp;
>> cM1.type='dynM1';

>> cM2.ind=[6,5,8,7,11,10,9,12,13,16,17,22,14,15,18,19,24,23,30,29];
>> load('covM2.mat') -> tmp
>> cM2.mat=tmp;
>> cM2.type='dynM2';
```

INPUTS SECOND USAGE name | type | units

THE FOLLOWING ORDER OF INPUT APPEARANCE MUST BE IMPERATIVELY
RESPECTED:

optica(GEOfILENAME,RES,MIR,NPIX,LAMBDA,AOPITCH,
aberrations or covariance, in any order)

MIR | structure variable | -
NEEDED ONLY FOR USAGE OF SECOND TYPE
Parameters of the telescope pupil. Output of function "segpos.m".

NPIX | integer scalar | 1
general matrix size for all 2D quantities defined in either the
pupil or focal plane - except for PSFs which are defined on
(RES*NPIX)² matrices.

LAMBDA | real scalar | MICRONS
optical wavelength in microns.

AOPITCH | real scalar | METERS
AO system actuator pitch. IF AO CORRECTION MODELING IS NOT DESIRED,
PUT THIS PARAMETER TO -1.

OUTPUTS name | type | units

results is a structure variable, with the following components:

psf | real array(NPIX*RES,NPIX*RES) | strehl
Telescope PSF with static and dynamic aberrations.
Note that the matrix size is adapted to the RES factor.

otf | complex array(NPIX,NPIX) | max(otf)=1
Telescope OTF with static and dynamic aberrations.

psftsc | real array(NPIX*RES,NPIX*RES) | strehl
Telescope PSF without aberrations.
Note that the matrix size is adapted to the RES factor.

otftsc | real array(NPIX,NPIX) | max(otftsc)=1
Telescope OTF without aberrations.

ONLY IF STATIC ABERRATIONS ARE SPECIFIED:

psfsta | real array(NPIX*RES,NPIX*RES) | strehl
Telescope PSF with static aberrations.
Note that the matrix size is adapted to the RES factor.

otfsta | complex array(NPIX,NPIX) | max(otftsc)=1
Telescope OTF with static aberrations.

ONLY IF DYNAMIC ABERRATIONS ARE SPECIFIED:

sfM1 | real array(NPIX,NPIX) | rad²
segmented pupil stationary phase structure function.

sfM2 | real array(NPIX,NPIX) | rad²
sum of the secondary mirrors stationary phase structure functions.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

sfseg.m
sfmir.m
otfstaseg.m
otfstamir.m
projdisctoseg.m
mathft.m
geometry.m

6.6 An other calculation sheet example

Practically, the most convenient way of using OPTICA is through the execution of a batch file where the actions are executed in a sequential order, starting with the loading of the calculation parameters and/or the definition of the telescope optics architecture, and followed by the calculation of the PSF/OTFs. Here is a copy of a typical calculation sheet (batch.m) that was used during the test of the toolbox:

```
%%%%%%%% PRIMARY MIRROR ARCHITECTURE

dext=10; % segment primary diameter flat-to-flat
dctr=10; % diameter of a disk to exclude the segment whose centers are outside
dint=10/7; % central obscuration dia - segments that are inside are excluded
nsd=7; % number of segments across dext
gap=0.04; % segments gaps (flat-to-flat)
mir=segpos(dext,dctr,dint,nsd,gap,'yes');
m2shadow.shape='tria';
m2shadow.size=3*mir.ssz;
m2shadow.angle=30;
arms.width=[0.1 0.1 0.1];
arms.cooint=zeros(2,3);
arms.cooint(1,:)=[0 0 0];
arms.cooint(2,:)=[0 0 0];
arms.cooext=zeros(2,3);
arms.cooext(1,:)=[mir.dextmax/2 -mir.dextmax/4 -mir.dextmax/4];
arms.cooext(2,:)=[0 mir.dextmax/2*sqrt(3)/2 -mir.dextmax/2*sqrt(3)/2];
mir=spider(mir,m2shadow,arms,'yes');

%%%%%%%% LOADING STATIC ABERRATIONS

% optical wavelength in microns
lambda=1;

% here we are generating a random set of segment aberrations,
% following a Gaussian distribution
z1=30; % piston rms in nm
z2=0.02; % tt rms in asec
a_m1.ind=[1,2,3]; % aberration j-indexes (Zernike, Noll numbering)
a_m1.mat=randn(mir.nseg,length(a_m1.ind));
a_m1.mat(:,1)=a_m1.mat(:,1)*z1*1e-9*(2*pi/(1e-6*lambda)); % piston
a_m1.mat(:,2:3)=a_m1.mat(:,2:3)*z2*4.848e-6*mir.ssz/4*(2*pi/(1e-6*lambda)); % tt
a_m1.type='staM1'; % type of aberration - static on primary

% now we are generating a set of secondary mirror(s) random aberrations
a_m2.ind=[4,5,6,7,8]; % defocus to coma
a_m2.mat=randn(1,5)*30e-9*(2*pi/(1e-6*lambda)); % random coeffs - 30nm rms
a_m2.type='staM2'; % type of aberration - static on secondary
```

```

% an other set of secondary mirror(s) random aberrations - for instance M3
% there can be an arbitrary number of type 'staM2' aberrations, but only one
% of type 'staM1'
a_m3.ind=[9,10,11,12,13,14,15,16];
a_m3.mat=randn(1,8)*40e-9*(2*pi/(1e-6*lambda));
a_m3.type='staM2';

%%%%%% LOADING COVARIANCE MATRICES

% here I am building a covariance for the segmented primary using a
% model of a set of cosine wave going across the primary.
cov_m1.ind=[1,2,3]; % j-indexes of the dynamic aberration
% here piston and tip-tilt, because we assume rigid segments.
lam=[10,10,1]; % wavelength of the cosine waves [m]
dir=[0,120,240]; % propagation direction of the waves [deg]
vel=[1,2,3]; % waves velocity [arbitrary unit]
amp=[0.1,0.1,0.01]*lambda*(2*pi/lambda); % wave amplitude in rad of phase
wave_npix=200; % matrix size for the visualization of a wave snapshot
[cov_m1.mat,mat,wave,wavefull]=...
covwave(mir,lam,dir,vel,amp,wave_npix,2*mir.dextmax/wave_npix);
cov_m1.type='dynM1'; % type of aberration and surface - dynamic, M1

% if a M1 covariance is available:
%load test/cov_mir05.mat

% here I am loading an existing covariance for the secondary mirror
cov_m2.ind=[2,3,4,6,5]; % setting up the j-indexes
load ~/tmt/OpticalModelling/mirrors_studies/dynamic/mat/SzzNoll.mat
cov_m2.mat=SzzNoll(1:5,1:5)*(2*pi/(lambda*1e-6))^2; % scaling to rad^2 unit
cov_m2.type='dynM2'; % type of aberration and surface - dynamic, M2
% there can be an arbitrary number of type 'dynM2' covariance matrices,
% but only one of type 'dynM1'

%%%%%% OTF, PSF

actpitch=mir.ssz; % AO system actuator pitch (here set to the segment size)
npix=200; % general matrix size for the calculation of OTFs, PSFs etc

% example 1: static aberrations on M1 only, no AO (actpitch=-1),
% PSF pixel size = Nyquist/2
res1=optica('geo1.mat',2,mir,npix,lambda,-1,a_m1);

% example 2: static aberrations on M1 only, AO on,
% PSF pixel size = Nyquist/2
res2=optica('geo2.mat',2,mir,npix,lambda,actpitch,a_m1);

```

```

% example 3: static aberrations on M1 & M2(s), no AO (actpitch=-1),
% PSF pixel size = Nyquist/2
res3=optica('geo3.mat',2,mir,npix,lambda,-1,a_m1,a_m2,a_m3);

% example 4: static aberrations on M1 & M2(s), AO on,
% PSF pixel size = Nyquist/2
res4=optica('geo4.mat',2,mir,npix,lambda,actpitch,a_m1,a_m2,a_m3);

% example 5: static+dynamic on M1 & M2(s), no AO (actpitch=-1),
% PSF pixel size = Nyquist/2
res5=optica('geo5.mat',2,mir,npix,lambda,-1,a_m1,a_m2,a_m3,cov_m1,cov_m2);

% example 6: static+dynamic on M1 & M2(s), AO on,
% PSF pixel size = Nyquist/2
res6=optica('geo6.mat',2,mir,npix,lambda,actpitch,a_m1,a_m2,a_m3,cov_m1,cov_m2);

```

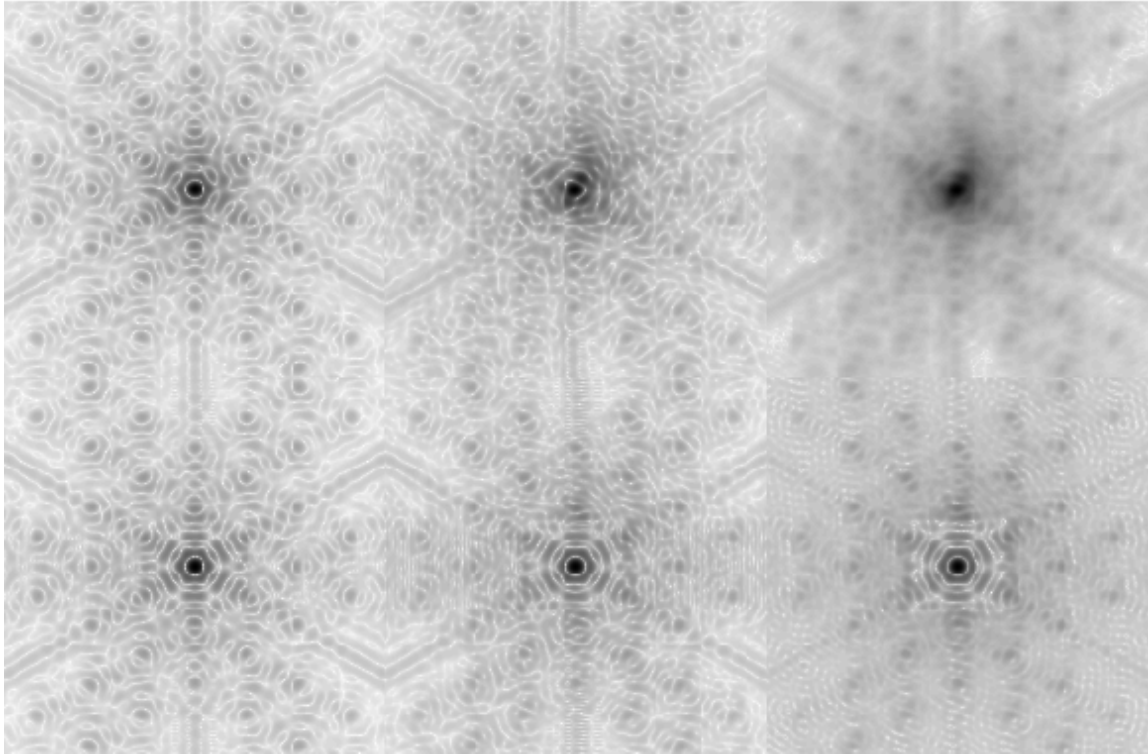


Figure 20: **Top: PSF without AO correction; bottom: PSF with AO correction. From left to right, cases 1 & 2, 3 & 4, 5 & 6. Power 1/8th intensity scaling.**

6.7 Secondary functions user manuals

6.7.1 covwave.m

```
FUNCTION [cov,mat,wave,wavefull]=  
    covwave(MIRROR,WAVELENGTH,THETA,VELOCITY,AMPLITUDE,NPIX,DX,FNAME)
```

Calculation of seg-2-seg & modes-2-modes covariance of Zernike coefficients for a wave made of NW independent cosine wavelets, across the primary segmented mirror. Segments are assumed rigid, so only the covariance of piston and tip-tilt is taken into account. Unit of covariance is square of unit of wave amplitude.

A 2D representation of the covariance is given in the output "mat".

With the inputs NPIX and DX, a random selection of a realization of the total wave on the mirror, in a matrix NPIX-by-NPIX, with a pixel size of DX [m] is given in the outputs "wave" and "wavefull".

INPUTS name | type | units

MIRROR | structure variable | -
Parameters of the segmented mirror. See output of function segpos.m for details.

WAVELENGTH | real array(NW) | m
Wavelength of each wavelets.

THETA | real array(NW) | deg
Direction of propagation of each wavelets making the full wave.
Wave numbers of the cosine wavelets making the full wave.

AMPLITUDE | real array(NW) | arbitrary
Wavelets amplitudes.

VELOCITY | real array(NW) | m/s
Wavelets traveling velocity module.

NPIX | scalar EVEN integer | 1
Matrix size for optional snap-shot of the wave.

DX | scalar integer | m
Metric pixel size for optional snap-shot of the wave.

OPTIONAL INPUT

FNAME | char string | -

File name to save the covariance matrix.

OUTPUT name | type | units

cov | array(MIRROR.nseg,MIRROR.nseg,3,3) | unit(AMPLITUDE)²
covariances seg-2-seg & modes-2-modes. covmat(1,2,i,j) is the
covariance between aberration i & j of segments 1 & 2.

mat | array(3*MIRROR.nseg,3*MIRROR.nseg) | unit(AMPLITUDE)²
covariances seg-2-seg & modes-2-modes, arranged in a square matrix
for visualization purpose only, in 3x3 blocks of nseg-by-nseg.

wave | array(128,128) | unit(AMPLITUDE)
a realization of the wave at t=0 for visualization purpose.

wavefull | array(128,128) | unit(AMPLITUDE)
a realization of the wave at t=0 for visualization purpose, but not
bounded by the circular pupil (full square matrix).

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

none.

6.7.2 dimensions.m

```
FUNCTION [r_focus,f_focus,r_pupil,f_pupil]=...
    dimensions(NPIX,RES,LAMBDA,TSCDIA,['INFO'])
```

This function computes the focal and pupil planes Cartesian coordinates and spatial/angular frequency arrays, in both X & Y. These arrays are needed when calculating focal and pupil plane related quantities, like, for instance, the optical transfer function, or the segment transmission Fourier transform, from analytical expressions.

INPUTS name | type | unit

NPIX | scalar even > 0, or -1 | pixels
Coordinate arrays matrix size. Must be even. There is a default value option, set by the matrix size required to have a good representation of the seeing limited PSF halo wings. If you want this default value, set NPIX to -1, and it will be replaced by the closest even integer > 16*(TSCDIA/r0) where r0 is set to a typical r0 value at the wavelength LAMBDA (using a r0=0.15 m at 500 nm).

RES | real scalar > or = 1 | 1
focal plane pixel size / maximum Nyquist focal plane pixel size. The higher RES, the smaller the pixel size: $DX=DX(\text{Nyquist})/\text{RES}$.
MUST BE > or = 1. When choosing RES > 1, the matrix size for the focal plane coordinate array and the PSF is increased accordingly, (NPIX->newNPIX) but the Field-of-View stays the same.

LAMBDA | scalar, real, >0 | microns
Working wavelength in MICRONS.

TSCDIA | scalar, real, >0 | meters
Telescope diameter.

KEYWORDS

'INFO' : if set, the matrix and pixel size are displayed in the screen at the end of the calculation.

OUTPUTS name | type | units

r_focus is a structure variable:
r_focus.x/y | array RES*NPIX-by-RES*NPIX, real | arcseconds
x/y-coordinate in the focal plane.

f_focus is a structure variable:
f_focus.x/y | array NPIX-by-NPIX, real | 1/rad
angular frequency x/y-coordinate in the focal plane.

r_pupil is a structure variable:
r_pupil.x/y | array NPIX-by-NPIX, real | meters
x/y-coordinate in the pupil plane.

f_pupil is a structure variable:
f_pupil.x/y | array NPIX-by-NPIX, real | 1/meters
spatial frequency x/y-coordinate in the pupil plane.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

none.

6.7.3 discft.m

```
FUNCTION FT=discft(F_PUPIL,REXT,RINT);
```

discft returns the 2D Fourier transform of a annulus with diameter $2*REXT$ and obscuration $2*RINT$. There is no FFT here, the FT is analytic, and defined with the following convention
 $FT(A)(fx,fy)=\text{double integral of } A(x,y)*\exp(-i*2*\pi*(fx*x+fy*y))$
where $A(x,y)=1$ inside the annulus and 0 outside, and fx,fy are the spatial frequencies.

INPUTS name | type | unit | default if any

F_PUPIL | 2D ARRAYS | 1/M

A structure variable, with two components, F_PUPIL.x & F_PUPIL.y, the pupil plane spatial frequencies.

REXT | SCALAR | ANY LENGTH | -
annulus external radius

RINT | SCALAR | SAME AS REXT | -
annulus internal radius

OUTPUTS name | type | unit

FT | DOUBLE ARRAY(size(F_PUPIL.x)) | unit(REXT) | -
Fourier transform of the annulus.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

none.

6.7.4 encene.m

FUNCTION [ee,d]=encene(otf,dia,lambda,eta,maxdia)

Computes EE curve as a function of integration aperture diameter d, using the OTF method.

INPUTS name | type | unit

otf | array(N,N) | 1
Optical transfer function. MAX = 1 !!

dia | scalar real | METERS
Telescope pupil diameter.

lambda | scalar real | MICRONS
optical wavelength associated to OTF.

eta | array real | 1
Nyquist focal plane pixel size / actual pixel size.
Must be >=1.

maxdia | scalar real | ASEC
diameter of the larger integration aperture to consider.
If set to -1, the largest aperture is the largest compatible with the PSF field size.

OUTPUT name | type | unit

ee | array real | 1
Encircled energy in an integration aperture of diameter d.

d | array real | asec
Integration aperture diameter.

ASSOCIATED FUNCTIONS

discft.m : analytical value of FT(disc with radius=d/2)

6.7.5 geometry.m

```
FUNCTION geo=  
    geometry(MIR,LAMBDA,NPIX,RES,DYNM1IND,DYNM2IND,STAM1IND,DMPITCH)
```

This function computes, saves & returns all the geometrical quantities that are needed in the calculation of the segmented telescope static and long exposure Optical Transfer Functions using the toolbox OPTICA.

Optionally, by giving an AO system actuator pitch, the AO correction of the segmented pupil and secondary mirror(s) modes can be taken into account, implementing DM filtering of the wavefront, plus WFS aliasing.

This function belongs to the OPTICA toolbox. The generated geometrical matrices are stored in an output structure variable, whose components are:

```
geo.*  
.lambda : working wavelength [microns]  
.npix   : matrix size  
.dxfoc  : pixel size focal plane  
.segpos : segments' positions  
.ssz    : segment radius  
.dextmax : max mirror diameter  
.r_focus : focal plane x,y coordinate arrays [asec]  
.f_focus : focal plane angular frequencies fx,fy arrays [1/rad]  
.r_pupil : pupil plane x,y coordinates arrays [m]  
.f_pupil : pupil plane spatial frequencies fx,fy arrays [1/m]  
.M1pupil : segmented pupil  
.M1pupilft : segmented pupil Fourier transform  
.M1pupcor : segmented pupil auto-correlation  
.M1shadow : spider+M2 shadow mask  
.M2pupil : circular pupil  
.M2pupilft : circular pupil Fourier transform  
.M2pupcor : circular pupil auto-correlation  
.staM1modft : segments static modes Fourier transforms  
.dynM1modft : segments dynamic modes Fourier transforms  
.dynM2modft : M2 dynamic modes Fourier transforms  
.dynM1modproft : segments dynamic modes products Fourier transforms  
.dynM2modproft : M2 dynamic modes products Fourier transforms  
.dynM1ind : j-indexes of dynamic segment modes (Zernike-Noll)  
.dynM2ind : j-indexes of dynamic mirror modes (Zernike-Noll)  
.staM1ind : j-indexes of static segment modes (Zernike-Noll)  
.dmpitch : AO deformable mirror actuator pitch
```

Basically, when using OPTICA, the user can either load these pre-computed matrices, or recompute them, when other parameters values are needed. Please see OPTICA user manual for details.

INPUTS name | type | unit

MIR | structure variable | -
parameters of the telescope pupil.

LAMBDA | scalar | micrometer
Working wavelength in MICRONS.

NPIX | scalar, even | pixels
OTF arrays matrix size. Must be even.

RES | real scalar > or = 1 | 1
focal plane pixel size / maximum Nyquist focal plane pixel size. The higher RES, the smaller the pixel size: $DX=DX(Nyquist)/RES$.
MUST BE > or = 1. When choosing RES > 1, the matrix size for the focal plane coordinate array and the PSF is increased accordingly, (NPIX->newNPIX) but the Field-of-View stays the same.

OPTIONAL INPUTS name | type | unit

dynM1ind | scalar or array | 1
j-indexes of the Zernike modes to consider in the description of the segments dynamic aberrations. Set it to 0 to ignore.

dynM2ind | scalar or array | 1
j-indexes of the Zernike modes to consider in the description of the secondary mirror(s) dynamic aberrations. Set to 0 to ignore.

staM1ind | scalar or array | 1
j-indexes of the Zernike modes to consider in the description of the segments static aberrations. Set to 0 to ignore.

dmpitch | scalar | m
Client AO system actuator pitch as seen from the telescope primary mirror. Set to 0 to ignore.

OUTPUTS name | type | unit

geo is a structure variable, whose components are:

.lambda | scalar | micrometer
Working wavelength in MICRONS (copy of input).

.npix | scalar, even, or -1 | pixels
Coordinate arrays matrix size (copy of input).

.dx foc | real scalar | asec/pixel
pixel size in the focal plane.

.segpos | array(2,nb segments) | METERS
x and y segment center coordinates.

.ssz | scalar | METERS
segment radius, i.e. half of point-to-point diameter, also equal to the side length of the segment (1/6 of perimeter).

.dextmax | scalar | METERS
maximal diameter, between most opposed segment corners.

.r_focus, a structure variable:
r_focus.x,y | real array(NPIX,NPIX) | arcseconds
x,y-coordinate in the focal plane.

.f_focus, a structure variable:
f_focus.x,y | real array(NPIX,NPIX) | 1/rad
angular frequency x,y-coordinate in the focal plane.

.r_pupil, a structure variable:
r_pupil.x,y | real array(NPIX,NPIX) | meters
x,y-coordinate in the pupil plane.

.f_pupil, a structure variable:
f_pupil.x,y | real array(NPIX,NPIX) | 1/meters
spatial frequency x,y-coordinate in the pupil plane.

.M1pupilft | complex array(NPIX,NPIX) | metre²
Fourier transform of the segmented pupil transmission. From analytical expressions (no FFT).

.M1pupil | complex array(NPIX,NPIX) | 1
inverse FFT of M1pupilft.

.M1shadow | complex array(NPIX,NPIX) | 1
Spider and M2 projected shadow mask onto M1. 0 inside, 1 outside.
If there is no spider, this component is set to -1.

.M1pupcor | real array(NPIX,NPIX) | metre²
Segmented pupil transmission auto-correlation. When normalized to 1

at the origin, defines the telescope Optical Transfer Function.

OPTIONAL OUTPUTS name | type | unit

Optional outputs depend on the setting of the inputs dynM1ind, dynM2ind, and staM1ind.

.dynM1ind | scalar | 1

If dynM1ind is set.

j-indexes of the Zernike modes to consider in the description of the segments dynamic aberrations.

.dynM1modft | real array(length(dynM1ind),NPIX,NPIX) | metre²

If dynM1ind is set.

Dynamic segments modes Fourier transforms.

.dynM1modproft | real array(X,NPIX,NPIX) | metre²

If dynM1ind is set.

Dynamic segments modes products Fourier transforms. As these products are commutative relative to the modes indexes, only one of the couples FTMk*Ml & FTML*Mk are kept, so the total number of such couples is $X=(\text{dynM1ind}^2+\text{dynM1ind})/2$.

.staM1ind | scalar | 1

If staM1ind is set.

j-indexes of the Zernike modes to consider in the description of the segments static aberrations.

.staM1modft | real array(staM1ind,NPIX,NPIX) | metre²

If staM1ind is set.

Static segments modes Fourier transform.

.M2pupilft | complex array(NPIX,NPIX) | metre²

If dynM2ind is set.

Fourier transform of the circular pupil transmission, i.e. the circle that circumscribe the segmented pupil. From analytical expressions (no FFT).

.M2pupcor | real array(NPIX,NPIX) | metre²

If dynM2ind is set.

Circular pupil transmission auto-correlation. From analytical expression (no FFT).

.dynM2ind | scalar | 1

If dynM2ind is set.

j-indexes of the Zernike modes to consider in the description of the

secondary mirrors dynamic aberrations.

.dynM2modft | real array(length(dynM2ind),NPIX,NPIX) | metre²

If dynM2ind is set.

Dynamic M2 modes product Fourier transforms.

.dynM2modproft | real array(X,NPIX,NPIX) | metre²

If dynM2ind is set.

Dynamic M2 modes product Fourier transforms. Here also,

$X=(\text{dynM2ind}^2+\text{dynM2ind})/2$.

.dmpitch | scalar | m

A0 system actuator pitch as seen from the telescope primary mirror.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

dimensions.m

discft.m

hexaft.m

indzer.m

mathft.m

mirmodproft.m

polzer.m

segpupilft.m

segmodproft.m

zpzqtozj.m

6.7.6 hexaft.m

FUNCTION mat=hexaft(F_PUPIL,SSZ,MU,NU);

hexaft returns the 2D Fourier transform of a hexagon with diameter point-to-point $2*SSZ$.
Optionally, it gives the Fourier transform of the same hexagon multiplied by a monomial x^n*y^m , a useful option for calculating FT of phases defined on hexagonal apertures. There is no FFT here, the FT is analytic, && defined with the following convention
 $FT(H)(fx,fy)=\text{double integral of } H(x,y)*\exp(-i*2*\pi*(fx*x+fy*y))$
where $H(x,y)=1$ inside the hexagon && 0 outside, and fx,fy are the spatial frequencies.

INPUTS name | type | unit

F_PUPIL | 2D ARRAYS | 1/M

A structure variable, with two components, F_PUPIL.x & F_PUPIL.y, the pupil plane spatial frequencies.

SSZ | SCALAR | M

hexagon radius, i.e. half distance point-to-point.

MU,NU | SCALAR | 1

Power of the x && y pupil space coordinates multiplying the hexagon transmission $[x^MU*y^NU*H(x,y)]$. FT of $[x^MU*y^NU*H(x,y)]$ is given in the output structure variable.

OUTPUTS name | type | unit | default

mat | COMPLEX ARRAY(size(F_PUPIL.x)) | $M^{(2+MU+NU)}$ | -
Fourier transform of $[x^MU*y^NU*H(x,y)]$.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

none.

6.7.7 indzer.m

FUNCTION ind=indzer(JMIN,JMAX)

Computes Zernike polynomials radial and azimuthal indexes associated to j-indexes JMIN to JMAX.

Usage option 1: ind=indzer(a scalar, a scalar)

Usage option 2: ind=indzer(a vector)

INPUTS name | type | unit

JMIN | scalar or vector | 1

Zernike j-index.

OPTIONAL INPUT name | type | unit

JMAX | scalar ONLY | 1

2nd Zernike j-index.

OUTPUT name | type | unit

ind | array(n lines, 3 columns) | 1

Indexes j,n,m

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

none.

6.7.8 mathft.m

FUNCTION `mat=mathft(OBJECT,DX,IC,JC,'INVERSE')`

MATHFT computes the discrete Fourier transform (DFT) (or inverse DFT) of a 1D or 2D object. The difference with the matlab FFT function is that the amplitude of the result is such that the DFT gives exactly the theoretical value of the continuous Fourier transform at each frequency discrete value. In other words, the normalization of the FT is respected, & the Parseval theorem is directly applicable to the result, following the convention:
$$\text{total}(\text{abs}(\text{object})^2) * \text{DX}^2 = \text{total}(\text{abs}(\text{result})^2) * \text{df}^2.$$

MATHFT works also on matrix sizes not a power of 2. But at least, matrix size must be even. If the input object matrix as an odd size, the top line & the right column will be truncated.

INPUTS name | type | unit

OBJECT | array 1D | 2D | -
array from which we compute the Fourier or inverse Fourier transform. Size of the array must be even (not necessarily a power of 2).

OPTIONAL INPUTS name | type | unit

DX | scalar | length
spatial sampling in the object space. Default is DX=1.

IC,JC | scalar | px
pixel position associated with the (0,0) coordinate in the linear space (the conjugate of the frequency space). Default is top right of the four central pixels, or N/2+1 for a 1D input array.

ORIENTATION | string | -
Set it to 'INVERSE' for computing the inverse FT. Default value is 'FORWARD'.

NOTE

I have checked if the amplitude of the result is correct following the mathematical definition of the Fourier transform, & this is OK. Beside, I have checked if the sign of the result is OK too. No bug here. Concerning the energy conservation, I have checked that
$$\text{sum}(\text{abs}(\text{object})^2) * \text{DX}^2 = \text{sum}(\text{abs}(\text{object FT})^2) * \text{df}^2.$$

All the check have been done in 1D & 2D. Last point : the aliasing. This is usual concern with Discrete Fourier Transform. The user has to be aware of that problem, and should use MATHFT knowing that the high frequencies are going to be folded in the $(-F_n, F_n)$ domain. Nothing can be done against this, except choosing a sufficiently small sampling of the object.

OUTPUTS name | type | unit

mat | double complex array | -
Fourier transform of the input object.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

none.

6.7.9 mirmodproft.m

FUNCTION mat=mirmodproft(J1,J2,DIA,F_PUPIL,AOFILTER)

Fourier transform of the product of Zernike polynomials Z_{J1} & Z_{J2} .
No FFT are used here, the calculation is analytic only:
1) the product is projected onto the Zernike basis;
2) the sum(a_i*Q_i) is computed, from analytic expression of Zernike polynomials.

This function is part of the toolbox OPTICA.

INPUTS name | type | unit

J1 & J2 | scalar ≥ 1 | 1
Zernike indexes of the two polynomials. Order can be arbitrary, and J1 can be equal to J2 as well. Must be > 0 .

DIA | scalar real | meter
telescope diameter.

F_PUPIL | real array, square | 1/metre
A STRUCTURE VARIABLE:
F_PUPIL.x : pupil plane spatial frequency x-coordinate.
F_PUPIL.y : pupil plane spatial frequency y-coordinate.

OPTIONAL INPUT name | type | unit

AOFILTER | real array | 1
A0 spatial transfer function.

OUTPUT name | type | unit

mat | complex square array, same dim as F_PUPIL.x,y | 1/metre²
Fourier transform of $Z_{J1}*Z_{J2}$.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

zpzqtozj.m
polzerft.m
mathft.m

6.7.10 otfstaseg.m

FUNCTION [otf,apsf,phase]=otfstaseg(SEGABB,SEGMODFT,SEGPOS,SPUPILFT,F_PUPIL,SHADOW)

otfstaseg is a function for computing the amplitude OTF and PSF associated to the static aberrations of a hexagonal segmented telescope. This function is part of the MATLAB toolbox OPTICA.

INPUTS name | type | units

SEGABB | real array (nb segments,nb modes) | radians
Segments static phase aberration Zernike coefficients, following Noll's numbering & normalization convention.

SEGMODFT | real array(nb modes,NPIX,NPIX) | 1
Segments Hermite modes Fourier transforms.

SEGPOS | real array (2, nb segments) or (nb segments, 2) | m
Segments centers.

SPUPILFT | real array(NPIX,NPIX) | 1/m²
Segmented pupil transmission Fourier transform.

F_PUPIL | 2D ARRAYS | 1/M
A structure variable, with two components, F_PUPIL.x & F_PUPIL.y, the pupil plane spatial frequencies.

SHADOW | complex array(NPIX,NPIX) | 1
Pupil spider and M2 shadow mask, 1 outside, 0 inside. If there is no spider shading, SHADOW must be set to -1.

OUTPUTS name | type | units

otf | complex array(NPIX,NPIX) | max(otf)=1
Optical transfer function of the aberrated segmented telescope.

OPTIONAL OUTPUTS name | type | units

apsf | complex array(NPIX,NPIX) | $\text{strehl}^{(1/2)}$
Amplitude point spread function of the aberrated segmented telescope.

phase | real array(NPIX,NPIX) | radian
optical phase.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

mathft.m

6.7.11 polherft.m

FUNCTION hft=polherft(J,SSZ,F_PUPIL)

Computes Hernike polynomial = (Zernike * hexagon) Fourier transform
for index J.

INPUTS name | type | unit

J | integer | 1
Zernike polynomial j-index.

SSZ | scalar | metre
Hexagonal segment radius, i.e. half of distance point-to-point.

F_PUPIL | real array, square | 1/m
A STRUCTURE VARIABLE:
F_PUPIL.x : pupil plane spatial frequency x-coordinate.
F_PUPIL.y : pupil plane spatial frequency y-coordinate.

OUTPUT name | type | unit

hft | array real | 1
Fourier transform of Hernike polynomial, index J.

ASSOCIATED FUNCTIONS

hexaft.m
indzer.m

6.7.12 polzerft.m

FUNCTION zft=polzerft(J,DIA,F_PUPIL)

Computes Zernike polynomial Fourier transform for index J, following (Noll, 1976) but with an inverse definition of the FT, according to J. Goodman (Introduction to Fourier Optics).

In this function, the Zernike is defined on a pupil of radius DIA/2, therefore the FT is defined on this space, which is not normalized, and we have $FTZ_j = \pi \cdot R^2 \cdot Q_j$, where Q_j is given in Noll 1976 (except with a sign change $(i)^m \rightarrow (-i)^m$).

INPUTS name | type | unit

J | integer | 1
Zernike polynomial j-index.

F_PUPIL | real array, square | 1/m
A STRUCTURE VARIABLE:
F_PUPIL.x : pupil plane spatial frequency x-coordinate.
F_PUPIL.y : pupil plane spatial frequency y-coordinate.

DIA | scalar integer | meters
Telescope diameter.

OUTPUT name | type | unit

zft | array real | 1
Fourier transform of Zernike polynomial, index J.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

indzer.m

6.7.13 projdisctoseg.m

FUNCTION asj=projdisctoseg(POS,SSZ,RAD,ZC,ZJ)

Computes the Zernike coefficients asj of the projection of the phase aberration defined on a continuous circular optical surface of radius RAD (e.g. the secondary mirror), onto the segments at position POS. The method is analytic, it is not a projection of modes constructed numerically. See OPTICA user manual.

INPUTS name | type | unit

POS | real array (2,NSEG) | M
x,y-coordinate of the center of the NSEG segments.
POS(1,:) = x-coo, POS(2,:) = y-coo.

SSZ | real scalar | M
Segment radius. For hexagonal segments, SSZ is the radius of the disc that circumscribe the hexagon.

RAD | real scalar | M
Radius of the continuous surface disc. Cannot be < than SSZ.

ZC | real array | any unit
Coefficients of the continuous disc Zernike aberrations. Can be given in any order, as the ZJ input indicates the j-indexes associated to ZC, i.e. there is no need to follow the usual order piston, tip-tilt, defocus etc... and any combination of the Zernike aberrations in any order can be given.

ZJ | integer array | 1
j-indexes of the Zernike coefficients given in ZC.

OUTPUT name | type | unit

asj | real array (NSEG,Nasj) | 1
Zernike coefficient of the projection of the continuous phase aberration onto the segment Zernike basis. Nasj, the number of segments' Zernike aberration is = max(ZJ), i.e. there cannot be segment aberrations at a higher j-index than there are in the initial continuous phase.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

indzer.m

6.7.14 rectft.m

FUNCTION ft=rectft(F_PUPIL,A,B)

rectft returns the 2D Fourier transform of a rectangle with side A (length) & B (width). There is no FFT here, the FT is analytic, & defined with the following convention
 $FT(R)(fx,fy)=\text{double integral of } R(x,y)*\exp(-i*\pi*(fx*x+fy*y))$
where $R(x,y)=1$ inside the rectangle & 0 outside, & fx,fy are the spatial frequencies.

INPUTS name | type | unit

F_PUPIL | 2D ARRAYS | 1/M

A structure variable, with two components, F_PUPIL.x & F_PUPIL.y, the pupil plane spatial frequencies.

A | scalar | any length | any
rectangle length, along horizontal axis.

B | scalar | any length | any, but same as A
rectangle width, along vertical axis.

OUTPUT name | type | unit | default

ft | DOUBLE ARRAY(size(F_PUPIL.x)) | unit(A,B) | -
Fourier transform of the rectangle.

ASSOCIATED FUNCTIONS/PROCEDURES

none.

6.7.15 segmodproft.m

FUNCTION mat=segmodproft(J1,J2,SSZ,F_PUPIL,AOFILTER)

Compute the Fourier transform of the product of Zernike polynomials Z_J1 & Z_J2 inscribed into a hexagonal segment of diameter SSZ (point-to-point). The Fourier transform is calculated from an analytical model, and not from the usage of a FFT algorithm. The result is therefore exact.

This function is part of the toolbox OPTICA.

INPUTS name | type | unit

J1 & J2 | scalar ≥ 1 | 1

Zernike indexes of the two polynomials. Order can be arbitrary, and J1 can be equal to J2 as well. Must be > 0 .

SSZ | scalar | metre

Hexagonal segment radius, i.e. $1/2$ distance point-to-point.

F_PUPIL | real array, square | 1/metre

A STRUCTURE VARIABLE:

F_PUPIL.x : pupil plane spatial frequency x-coordinate.

F_PUPIL.y : pupil plane spatial frequency y-coordinate.

OPTIONAL INPUT name | type | unit

AOFILTER | real array | 1

A0 spatial transfer function.

OUTPUT name | type | unit

mat | complex square array, same dim as F_PUPIL.x,y | $1/\text{metre}^2$

Fourier transform of $Z_{J1} * Z_{J2} * \text{hexagon}$.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

polherft.m

hexaft.m

mathft.m

zpzqtozj.m

6.7.16 segpupilft.m

FUNCTION [pupilft,pupil,shadow]=segpupilft(MIRROR,F_PUPIL,R_PUPIL)

This function computes the segmented pupil transmission Fourier transform, from analytical expressions, and the pupil transmission from an inverse numerical FFT of this analytical pupil FT.

INPUTS name | type | unit

MIRROR is a structure variable:
parameters of the telescope pupil. See output of segpos.m and spider.m

F_PUPIL is a structure variable:
F_PUPIL.x,y | real array(NPIX,NPIX) | 1/meters
spatial frequency x,y-coordinate in the pupil plane. Output of function dimensions.m

R_PUPIL is a structure variable:
R_PUPIL.x,y | real array(NPIX,NPIX) | meters
x,y-coordinate in the pupil plane. Output of function dimensions.m

OUTPUTS name | type | unit

pupilft | real array(NPIX,NPIX) | metre²
Pupil transmission Fourier transform from analytical expression.

pupil | real array(NPIX,NPIX) | 1
Pupil transmission from inverse FFT of pupilft (above).

shadow | complex array(NPIX,NPIX) | 1
spider shadow, 0 inside, 1 outside. If there is no spider, set to -1.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

mathft.m
triaft.m
rectft.m
hexaft.m
discft.m

6.7.17 sfmir.m

FUNCTION sfmir

```
sf=sfmir(COV,J_INDEXES,F_PUPIL,MIRMODFT,MIRMODPROFT,PUPILFT,PUPCORR)
```

Computation of the secondary/tertiary mirrors wavefront structure function, averaged over the mirror positions. The average SF calculated here is a function of the separation and orientation between 2 points into the mirror, but does not depend on the absolute location of these points, as it would be the case in the non averaged SF. See user manual for details.

INPUTS name | type | unit

COV | real array (nmod,nmod) | can be any unit
Zernike aberrations coefficients temporal covariance matrix.
Indexes start from (piston, the tip-tilt etc) following Noll's convention (See Noll, R. JOSA 1976).

J_INDEXES | integer vector(nmod) | 1
j-indexes of the mirror modes, following Noll's definition.

F_PUPIL.x/y | real array, square | 1/metre
pupil plane spatial frequency x/y-coordinate.

MIRMODFT | complex array (number of modes,npix,npix) | metre²
Fourier transform of mirror modes. See mirmodproft.m.

MIRMODPROFT | complex array (X,npix,npix) | metre²
Fourier transform of each couple of mirror modes. See function mirmodproft.m. $X=(\text{number of modes}^2+\text{number of modes})/2$.

PUPILFT | complex array(npix,npix) | metre²
Pupil transmission Fourier transform. See function pupilft.m

PUPCORR | real array(npix,npix) | metre²
Pupil transmission auto-correlation.

OUTPUT | name | type | unit

sf | array(npix,npix) | same unit than COV
Circular mirror averaged structure function.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

mathft.m

6.7.18 sfseg.m

```
FUNCTION sf=sfseg(COV,J_INDEXES,SEGPOS,F_PUPIL,SEGMODFT,...  
                SEGMODPROFT,PUPILFT,PUPCORR)
```

Segmented mirror stationary structure function.

INPUTS name | type | unit

COV | real array, 4-D (nseg,nseg,nmod,nmod) | can be any unit
Zernike aberrations coefficients temporal covariance between each segment
for each aberrations. This is a 4-D array: first & second indexes are for
the segments numbering, and the third and fourth indexes for aberrations
numbering. For instance, C_5,6,7,5 is the covariance between segments
5 and 6 and respectively 7th and 5th aberrations.

J_INDEXES | integer vector(nmod) | 1
j-indexes of the segment modes, following Noll's definition.

SEGPOS | real array (2,#seg) or (#seg,2) | metre
Segments center position.

F_PUPIL.x/y | real array, square | 1/metre
pupil plane spatial frequency y-coordinate.

SEGMODFT | complex array (number of modes,npix,npix) | metre²
Fourier transform of segment modes. Output of "segmodproft.m".

SEGMODPROFT | complex array (X,npix,npix) | metre²
Fourier transform of segment modes products. Output of function
"segmodproft.m". X=(number of modes²+number of modes)/2.

PUPILFT | complex array(npix,npix) | metre²
Pupil transmission Fourier transform.

PUPCORR | real array(npix,npix) | metre²
Pupil transmission auto-correlation.

OUTPUT | name | type | unit

sf | array(npix,npix) | same unit as COV input
Segmented primary mirror wavefront structure function.

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

mathft.m

6.7.19 spider.m

FUNCTION mir=spider(MIRROR,M2SHADOW,ARMS,VIEW,FNAME)

spider is an optional function of the toolbox OPTICA to specify the secondary mirror support structure shadow (the ''spider''), and add the spider parameters to the segmented mirror structure variable.

The secondary mirror shadow shape can be specified either as a disc, a hexagon, a square, or an equilateral triangle. Supporting arms and cables are specified as long, narrow rectangles.

When executed, spider.m reads the spider parameters, computes the length, orientation and position of the center of the arms/cables, and add these values to the mirror structure variable (input MIRROR) created by the function segpos.m. These parameters are then used within function optica.m to add the effect of the spider obscuration to the PSF, using the Fourier transform approach described in the OPTICA report and user manual.

INPUTS name | type | unit

MIRROR is a structure variable. See output of function segpos.m for detailed description of components.

M2SHADOW is a structure variable, with components:

.shape | STRING | -

Type of M2 support structure shadow:

'tria' for an equilateral triangle, centered on its barycentre,

'squa' for a square,

'hexa' for a hexagonal,

'disc' for a disc.

.size | SCALAR | METERS

Size of M2 support structure. Depends on the type:

if 'tria': triangle basis,

if 'squa': square side,

if 'hexa': hexagon diameter, point-to-point,

if 'disc': disc diameter.

.angle | SCALAR | DEG

Orientation of M2 support structure relative to x-axis,

counter-clockwise. An angle of 0 deg corresponds to a

triangle/square/hexagon seated on its basis. Not operational for

disc type shadow (input ignored).

ARMS is a structure variable, with components:

.width | array(nb of arms and cables) double | METERS
Width of spider's arms and cables.

.cooint | array(2,nb arms) double | METERS
Coordinates of the inner attachment point of the arm or cable. Does not have to be exactly on the M2 shadow edge. Can be anywhere.
Syntax: <name>.cooint(1,:) = x coordinate,
<name>.cooint(2,:) = y coordinate.

.cooext | array(2,nb arms) double | METERS
Coordinates of the outer attachment point of the arm or cable.
Syntax: <name>.cooext(1,:) = x coordinate,
<name>.cooext(2,:) = y coordinate.

OPTIONAL INPUTS

VIEW: is set to 'yes', then a drawing of the segmented mirror with the spider is displayed on the screen, otherwise set it to 'no'.

FNAME | character string | -
The mirror structure variable will be saved in the file FNAME. If not set, the mirror structure variable is saved in the file './mirwithspider.mat', by default.

OUTPUTS name | type | unit

The input MIRROR structure variable, on which the M2 shadow and arms/cables parameters have been added, i.e.

.m2shape | STRING | -
Type of M2 support structure shadow. Copied from input.

.m2size | SCALAR | METERS
Size of M2 support structure. Copied from input.

.m2angle | SCALAR | DEG
Orientation of M2 support structure. Copied from input.

.armwidth | array(nb of arms/cables) double | METERS
Width of spider's arms and cables. Copied from input.

.armlength | array(nb of arms/cables) double | METERS
Length of spider's arms and cables.

.armangle | array(nb of arms/cables) double | DEG
Orientation of arms/cable, counterclockwise relative to x-axis.

.armcenter | array(2,nb of arms/cables) double | METERS
Coordinates of the arms/cables center.
.armcenter(1,:) = x coordinate, .armcenter(2,:) = y coordinate.

EXAMPLE

```
>> dext=10;
>> dctr=10;
>> dint=1.5;
>> nsd=7;
>> gap=0.04;
>> mir=segpos(dext,dctr,dint,nsd,gap,'yes','test/mir_test.mat')

mir =
    nseg: 37
     swd: 1.39428571428571
     ssz: 0.80499123247011
     dbs: 1.43428571428571
  segpos: [2x36 double]
dextflat: 10
dextctr: 10
dextmax: 10.03234822383841
   dint: 1.50000000000000
   gap: 0.04000000000000
   nsd: 7
   surf: 60.60893988014930

>> M2SHADOW.shape='tria'; % triangle M2 support structure
>> M2SHADOW.size=3; % basis of triangle is 3 meters
>> M2SHADOW.angle=90; % the triangle points to the right, as '>>>'.
>> ARMS.width=[0.1 0.1 0.1 0.1]; % four 10 cm wide arms spider
>> ARMS.cooint=zeros(2,4);
>> ARMS.cooext=zeros(2,4);
>> ARMS.cooint(1,1:4)=[1 0 -1 0];
>> ARMS.cooint(2,1:4)=[0 1 0 -1];
>> ARMS.cooext(1,1:4)=[5 0 -5 0];
>> ARMS.cooext(2,1:4)=[0 5 0 -5];
>> newmir=spider(mir,M2SHADOW,ARMS,'yes','test/mir_test.mat')

newmir =

    nseg: 36
     swd: 1.39428571428571
```

```
      ssz: 0.80499123247011
      dbs: 1.43428571428571
      segpos: [2x36 double]
dextflat: 10
dextctr: 10
dextmax: 10.03234822383841
      dint: 1.50000000000000
      gap: 0.04000000000000
      nsd: 7
      surf: 60.60893988014930
m2shape: 'tria'
m2size: 3
m2angle: 90
armwidth: [0.1 0.1 0.1 0.1]
armlength: [4 4 4 4]
armangle: [180 -90 0 90]
armcenter: [2x4 double]
```

ASSOCIATED FUNCTIONS/PROCEDURES

none.

6.7.20 triaft.m

FUNCTION ft=triaft(F_PUPIL,BASE)

rectft returns the 2D Fourier transform of an equilateral triangle with basis BASE. There is no FFT here, the FT is analytic, & defined with the following convention
 $FT(T)(fx,fy)=\text{double integral of } T(x,y)*\exp(-i*\pi*(fx*x+fy*y))$
where $T(x,y)=1$ inside the triangle & 0 outside, & fx,fy are the spatial frequencies.

INPUTS name | type | unit

F_PUPIL | 2D ARRAYS | 1/M
A structure variable, with two components, F_PUPIL.x & F_PUPIL.y, the pupil plane spatial frequencies.

BASE | scalar | any length | any
Equilateral triangle base length.

OUTPUT name | type | unit | default

ft | DOUBLE ARRAY(size(F_PUPIL.x)) | unit(BASE) | -
Fourier transform of the triangle.

ASSOCIATED FUNCTIONS/PROCEDURES

none.

6.7.21 zpzqtozj.m

FUNCTION [coeff,existind]=zpzqtozj(J1,J2)

Computes the coefficients a_i of the projection of the polynomials product $Z_{J1} * Z_{J2}$ in the Zernike basis, i.e. $Z_{J1} * Z_{J2} = \sum_i (a_i * Z_i)$. The method is analytic, it is not a projection of the product on a set of modes constructed numerically.

The array given at the input only includes the coefficients that are not null, and "existind" gives the j-number of these non-zero coefficients.

For instance, we find $Z_2 * Z_5 = 0.8165 * Z_3 + 0.2887 * Z_7 + 0.8660 * Z_9$

INPUTS name | type | unit

J1,J2 | integer scalar > 0 | 1
j-indexes of the two Zernike polynomials

OUTPUTS name | type | unit

coeff | array real | 1
Zernike coefficient of the projection of the product $Z_{J1} * Z_{J2}$ in the Zernike space.

existind | array integers | 1
j-indexes associated to the coefficients "coeff".

EXTERNAL CUSTOM FUNCTIONS/PROCEDURES

indzer.m

A Cartesian representation of the Zernike polynomials

The segment transmission is defined in Cartesian coordinates, so must be the Zernike polynomials. Let us therefore give the Zernike polynomial in a Cartesian representation. Zernike polynomials are given by Noll (1976) in polar coordinates:

$$Z_j(r, \theta) = \sqrt{[(m \neq 0) + 1](n + 1)} \begin{cases} R_n^m(r) & m = 0 \\ R_n^m(r) \cos(m\theta) & m \neq 0 \text{ and } j \text{ even} \\ R_n^m(r) \sin(m\theta) & m \neq 0 \text{ and } j \text{ odd} \end{cases} \quad (87)$$

where

$$R_n^m(r) = \sum_{s=0}^{\frac{n-m}{2}} \frac{(-1)^s (n-s)! r^{n-2s}}{s! \left(\frac{n+m}{2} - s\right)! \left(\frac{n-m}{2} - s\right)!} \quad (88)$$

By expanding $\cos(m\theta)$ and $\sin(m\theta)$ in terms of products of $\cos^k\theta \cos^l\theta$, $\sin^k\theta \cos^l\theta$ and $\sin^k\theta \sin^l\theta$, we find,

$$\cos(m\theta) = \frac{1}{r^m} \sum_{k=0}^q (-1)^k \binom{m}{2k} x^{(m-2k)} y^{2k} \quad \text{where } q = \begin{cases} \frac{m-1}{2} & m \text{ odd} \\ \frac{m}{2} & m \text{ even} \end{cases} \quad (89)$$

$$\sin(m\theta) = \frac{y}{x r^m} \sum_{k=0}^q (-1)^k \binom{m}{2k+1} x^{(m-2k)} y^{2k} \quad \text{where } q = \begin{cases} \frac{m-1}{2} & m \text{ odd} \\ \frac{m}{2} - 1 & m \text{ even} \end{cases} \quad (90)$$

which can be squeezed into a single expression, including the case $m = 0$

$$\Xi_j^m(x, y) = \frac{1}{r^m} \left(\frac{y}{x}\right)^{p(j)} \sum_{k=0}^{q_{j,m}} (-1)^k \binom{m}{2k + p_{j,m}} x^{(m-2k)} y^{2k} \quad (91)$$

with

$q_{j,m} = \dots$	$m \text{ odd}$	$m \text{ even}$	$m = 0$	and	$p_{j,m} = \dots$	$m \neq 0$	$m = 0$
$j \text{ odd}$	$\frac{m-1}{2}$	$\frac{m-2}{2}$	0		$j \text{ odd}$	1	0
$j \text{ even}$	$\frac{m-1}{2}$	$\frac{m}{2}$			$j \text{ even}$	0	0

With the later equality, Eq. (87) becomes

$$Z_j(x, y) = \sqrt{[(m \neq 0) + 1](n + 1)} \sum_{s=0}^{\frac{n-m}{2}} \frac{(-1)^s (n-s)! r^{(n-m-2s)}}{s! \left(\frac{n+m}{2} - s\right)! \left(\frac{n-m}{2} - s\right)!} \times \left(\frac{y}{x}\right)^{p_{j,m}} \sum_{k=0}^{q_{j,m}} (-1)^k \binom{m}{2k + p_{j,m}} x^{(m-2k)} y^{2k} \quad (92)$$

and it must be noted that as $n - m$ is always even, the exponent of r in Eq. (92), $n - m - 2s$, is even, therefore we can write

$$r^{2\left(\frac{n-m}{2} - s\right)} = (x^2 + y^2)^{\left(\frac{n-m}{2} - s\right)} = \sum_{l=0}^{\frac{n-m}{2} - s} \binom{\frac{n-m}{2} - s}{l} x^{2\left(\frac{n-m}{2} - s - l\right)} y^{2l} \quad (93)$$

so,

$$Z_j(x, y) = \sqrt{[(m \neq 0) + 1](n + 1)} \sum_{s=0}^{\frac{n-m}{2}} \sum_{l=0}^{\frac{n-m}{2}-s} \sum_{k=0}^{q_{j,m}} (-1)^{(s+k)} \binom{m}{2k + p_{j,m}} \binom{\frac{n-m}{2} - s}{l} \times \frac{(n-s)!}{s! \left(\frac{n+m}{2} - s\right)! \left(\frac{n-m}{2} - s\right)!} x^{[n-2(s+l+k)-p_{j,m}]} y^{[2(l+k)+p_{j,m}]} \quad (94)$$

B Fourier transform of the hexagonal modes $\mathcal{H}(\mathbf{u})\mathbf{u}^\mu\mathbf{v}^\nu$

We develop in this section the calculation of the Fourier transform of the products $\mathcal{H}(\mathbf{u})\mathbf{u}^\mu\mathbf{v}^\nu$. According to the theorem of the derivative of the Fourier transform, we have

$$\mathcal{F} \{ \mathcal{H}(\mathbf{u})\mathbf{u}^\mu\mathbf{v}^\nu \} \equiv \tilde{\mathcal{H}}_{\mu,\nu}(\mathbf{f}) = \left(\frac{i}{2\pi} \right)^{\mu+\nu} \frac{\partial^{\mu+\nu}}{\partial f_u^\mu \partial f_v^\nu} \tilde{\mathcal{H}}(\mathbf{f}) \quad (95)$$

The Fourier transform of the hexagon is calculated without difficulties (a is the segment radius),

$$\begin{aligned} \tilde{\mathcal{H}}(\mathbf{f}) &= \frac{2\sqrt{3}}{(2\pi)^2} \left\{ \frac{\cos[\pi a(f_v + \sqrt{3}f_u)] - \cos(2\pi a f_v)}{f_v(f_v - \sqrt{3}f_u)} + \frac{\cos[\pi a(f_v - \sqrt{3}f_u)] - \cos(2\pi a f_v)}{f_v(f_v + \sqrt{3}f_u)} \right\} \\ &\equiv \frac{2\sqrt{3}}{(2\pi)^2} \left\{ \frac{N_+(\mathbf{f})}{D_-(\mathbf{f})} + \frac{N_-(\mathbf{f})}{D_+(\mathbf{f})} \right\} \end{aligned} \quad (96)$$

and its order μ, ν derivative is calculated with the Leibniz rule (derivative of a product)

$$\frac{\partial^{\mu+\nu}}{\partial f_u^\mu \partial f_v^\nu} \left[\frac{N_\pm(\mathbf{f})}{D_\mp(\mathbf{f})} \right] = \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \binom{\mu}{\kappa} \binom{\nu}{\lambda} \partial_{\kappa,\lambda} N_\pm(\mathbf{f}) \partial_{\mu-\kappa, \nu-\lambda} [1/D_\mp(\mathbf{f})] \quad (97)$$

With the formula

$$\partial_{\alpha,\beta} \left(\frac{1}{y(y \mp ax)} \right) = \frac{(\mp a)^\alpha (-1)^{\alpha+\beta} \alpha! \beta!}{y^{\beta+1} (y \mp ax)^{\alpha+\beta+1}} \sum_{\gamma=0}^{\beta} \binom{\alpha + \beta + 1}{\gamma} (\mp ax)^{\beta-\gamma} y^\gamma \quad (98)$$

we find, for the derivative of the denominator term D_\mp ,

$$\begin{aligned} \partial_{\mu-\kappa, \nu-\lambda} [1/D_\mp(\mathbf{f})] &= \frac{(\mp \sqrt{3})^{\mu-\kappa} (-1)^{\mu+\nu-\kappa-\lambda} (\mu-\kappa)! (\nu-\lambda)!}{f_v^{\nu-\lambda+1} (f_v \mp \sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} \times \\ &\quad \sum_{\zeta=0}^{\nu-\lambda} \binom{\mu + \nu - \kappa - \lambda + 1}{\zeta} (\mp \sqrt{3}f_u)^{\nu-\lambda-\zeta} f_v^\zeta \end{aligned} \quad (99)$$

and for the derivative of the cosine terms, in the numerator $N_\pm(\mathbf{f})$, we find respectively

$$\partial_{\kappa,\lambda} \cos[\pi a(f_v \pm \sqrt{3}f_u)] = (\pm \sqrt{3})^\kappa (\pi a)^{\kappa+\lambda} \begin{cases} (-1)^{\frac{\kappa+\lambda}{2}} \cos[\pi a(f_v \pm \sqrt{3}f_u)] & \kappa + \lambda \text{ even} \\ (-1)^{\frac{\kappa+\lambda+1}{2}} \sin[\pi a(f_v \pm \sqrt{3}f_u)] & \kappa + \lambda \text{ odd} \end{cases} \quad (100)$$

and

$$\partial_{\kappa,\lambda} \cos(2\pi a f_v) = \delta_{\kappa,0} (2\pi a)^\lambda \begin{cases} (-1)^{\frac{\lambda}{2}} \cos(2\pi a f_v) & \lambda \text{ even} \\ (-1)^{\frac{\lambda+1}{2}} \sin(2\pi a f_v) & \lambda \text{ odd} \end{cases} \quad (101)$$

where $\delta_{\kappa,0} = 1$ if $\kappa = 0$, and 0 otherwise. It is practical to regroup these trigonometric factors in one term, so we define

$$\begin{aligned} \Theta_{\kappa,\lambda}^\pm(\mathbf{f}) &\equiv \partial_{\kappa,\lambda} N_\pm(\mathbf{f}) = \partial_{\kappa,\lambda} \cos[\pi a(f_v \pm \sqrt{3}f_u)] - \partial_{\kappa,\lambda} \cos(2\pi a f_v) = (\pm\sqrt{3})^\kappa (\pi a)^{\kappa+\lambda} \\ &\times \begin{cases} (-1)^{\frac{\kappa+\lambda}{2}} \cos[\pi a(f_v \pm \sqrt{3}f_u)] - \delta_{\kappa,0} 2^\lambda (-1)^{\frac{\lambda}{2}} \cos(2\pi a f_v) & \kappa \text{ even}, \lambda \text{ even} \\ (-1)^{\frac{\kappa+\lambda+1}{2}} \sin[\pi a(f_v \pm \sqrt{3}f_u)] - \delta_{\kappa,0} 2^\lambda (-1)^{\frac{\lambda+1}{2}} \sin(2\pi a f_v) & \kappa \text{ even}, \lambda \text{ odd} \\ (-1)^{\frac{\kappa+\lambda+1}{2}} \sin[\pi a(f_v \pm \sqrt{3}f_u)] & \kappa \text{ odd}, \lambda \text{ even} \\ (-1)^{\frac{\kappa+\lambda}{2}} \cos[\pi a(f_v \pm \sqrt{3}f_u)] & \kappa \text{ odd}, \lambda \text{ odd} \end{cases} \\ &= (\pm\sqrt{3})^\kappa (\pi a)^{\kappa+\lambda} \begin{cases} (-1)^{\frac{\kappa+\lambda}{2}} \cos[\pi a(f_v \pm \sqrt{3}f_u)] - \delta_{\kappa,0} 2^\lambda (-1)^{\frac{\lambda}{2}} \cos(2\pi a f_v) & \kappa + \lambda \text{ even} \\ (-1)^{\frac{\kappa+\lambda+1}{2}} \sin[\pi a(f_v \pm \sqrt{3}f_u)] - \delta_{\kappa,0} 2^\lambda (-1)^{\frac{\lambda+1}{2}} \sin(2\pi a f_v) & \kappa + \lambda \text{ odd} \end{cases} \end{aligned} \quad (102)$$

Regrouping all the terms into Eq. (95), we get

$$\begin{aligned} \tilde{\mathcal{H}}_{\mu,\nu}(\mathbf{f}) &= \left(\frac{i}{2\pi}\right)^{\mu+\nu} \frac{2\sqrt{3}}{(2\pi)^2} \times \\ &\left\{ \left[\sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \binom{\mu}{\kappa} \binom{\nu}{\lambda} \Theta_{\kappa,\lambda}^+(\mathbf{f}) \frac{(-\sqrt{3})^{\mu-\kappa} (-1)^{\mu+\nu-\kappa-\lambda} (\mu-\kappa)! (\nu-\lambda)!}{f_v^{\nu-\lambda+1} (f_v - \sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} \times \right. \right. \\ &\quad \left. \sum_{\zeta=0}^{\nu-\lambda} \binom{\mu+\nu-\kappa-\lambda+1}{\zeta} (-\sqrt{3}f_u)^{\nu-\lambda-\zeta} f_v^\zeta \right] \\ &+ \left[\sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \binom{\mu}{\kappa} \binom{\nu}{\lambda} \Theta_{\kappa,\lambda}^-(\mathbf{f}) \frac{(\sqrt{3})^{\mu-\kappa} (-1)^{(\mu+\nu-\kappa-\lambda)} (\mu-\kappa)! (\nu-\lambda)!}{f_v^{\nu-\lambda+1} (f_v + \sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} \times \right. \\ &\quad \left. \sum_{\zeta=0}^{\nu-\lambda} \binom{\mu+\nu-\kappa-\lambda+1}{\zeta} (\sqrt{3}f_u)^{\nu-\lambda-\zeta} f_v^\zeta \right] \Big\} \end{aligned}$$

and regrouping the sums,

$$\begin{aligned}
\tilde{\mathcal{H}}_{\mu,\nu}(\mathbf{f}) &= \left(\frac{i}{2\pi}\right)^{\mu+\nu} \frac{2\sqrt{3}}{(2\pi)^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} (-1)^{(\mu+\nu-\kappa-\lambda)} (\mu-\kappa)! (\nu-\lambda)! \binom{\mu}{\kappa} \binom{\nu}{\lambda} \binom{\mu+\nu-\kappa-\lambda+1}{\zeta} \times \\
&\quad \left[\frac{\Theta_{\kappa,\lambda}^+(\mathbf{f})(-\sqrt{3})^{\mu-\kappa}(-\sqrt{3}f_u)^{\nu-\lambda-\zeta} f_v^\zeta}{f_v^{\nu-\lambda+1}(f_v-\sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} + \frac{\Theta_{\kappa,\lambda}^-(\mathbf{f})(\sqrt{3})^{\mu-\kappa}(\sqrt{3}f_u)^{\nu-\lambda-\zeta} f_v^\zeta}{f_v^{\nu-\lambda+1}(f_v+\sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} \right] \\
&= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{3^{-\frac{\kappa+\lambda+\zeta}{2}}}{(-1)^{\kappa+\lambda} \kappa! \lambda!} \binom{\mu+\nu-\kappa-\lambda+1}{\zeta} \times \\
&\quad \left[\frac{(-1)^{(\mu+\nu-\kappa-\lambda-\zeta)} \Theta_{\kappa,\lambda}^+(\mathbf{f}) f_u^{\nu-\lambda-\zeta} f_v^\zeta}{f_v^{\nu-\lambda+1}(f_v-\sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} + \frac{\Theta_{\kappa,\lambda}^-(\mathbf{f}) f_u^{\nu-\lambda-\zeta} f_v^\zeta}{f_v^{\nu-\lambda+1}(f_v+\sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} \right] \\
&= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{3^{-\frac{\kappa+\lambda+\zeta}{2}}}{(-1)^{\kappa+\lambda} \kappa! \lambda!} \binom{\mu+\nu-\kappa-\lambda+1}{\zeta} \times \\
&\quad f_u^{\nu-\lambda-\zeta} \frac{f_v^{\lambda+\zeta}}{f_v^{\nu+1}} \left[\frac{(-1)^{(\mu+\nu-\kappa-\lambda-\zeta)} \Theta_{\kappa,\lambda}^+(\mathbf{f})}{(f_v-\sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} + \frac{\Theta_{\kappa,\lambda}^-(\mathbf{f})}{(f_v+\sqrt{3}f_u)^{\mu+\nu-\kappa-\lambda+1}} \right] \quad (103)
\end{aligned}$$

writing the term in brackets with a common denominator (note that we need to keep the exponents of the different terms positive by distributing them in the numerator or the denominator, in order for l'Hopital's rule to be usable when computing limits at asymptotes - see below)

$$\begin{aligned}
\tilde{\mathcal{H}}_{\mu,\nu}(\mathbf{f}) &= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\lambda+\zeta}} f_u^{\nu-\lambda-\zeta} f_v^{\lambda+\zeta} \times \\
&\quad \frac{(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(\mathbf{f})(f_v-\sqrt{3}f_u)^{\kappa+\lambda}(f_v+\sqrt{3}f_u)^{\mu+\nu+1} + \Theta_{\kappa,\lambda}^-(\mathbf{f})(f_v+\sqrt{3}f_u)^{\kappa+\lambda}(f_v-\sqrt{3}f_u)^{\mu+\nu+1}}{f_v^{\nu+1}(f_v^2-3f_u^2)^{\mu+\nu+1}} \\
&= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta} f_u^{\nu-\lambda-\zeta}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\lambda+\zeta}} \frac{N_{\kappa,\lambda,\zeta}(\mathbf{f})}{D_{\kappa,\lambda,\zeta}(\mathbf{f})} \quad (104)
\end{aligned}$$

where

$$\begin{aligned}
N_{\kappa,\lambda,\zeta}(\mathbf{f}) &\equiv f_v^{\lambda+\zeta} \left[(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(\mathbf{f})(f_v-\sqrt{3}f_u)^{\kappa+\lambda}(f_v+\sqrt{3}f_u)^{\mu+\nu+1} \right. \\
&\quad \left. + \Theta_{\kappa,\lambda}^-(\mathbf{f})(f_v+\sqrt{3}f_u)^{\kappa+\lambda}(f_v-\sqrt{3}f_u)^{\mu+\nu+1} \right] \quad (105)
\end{aligned}$$

and

$$D_{\kappa,\lambda,\zeta}(\mathbf{f}) \equiv f_v^{\nu+1}(f_v^2-3f_u^2)^{\mu+\nu+1} \quad (106)$$

Now, $\tilde{\mathcal{H}}_{\mu,\nu}$ is undetermined at the origin $(f_u, f_v) = (0, 0)$, for the horizontal axis f_u ($f_v \rightarrow 0$), and along the two lines at $\pm 60^\circ$ relative to the horizontal axis ($f_v \rightarrow \pm\sqrt{3}f_u$), so we need to compute

$$\tilde{\mathcal{H}}_{\mu,\nu}(0, 0) = \lim_{f_v \rightarrow 0} \tilde{\mathcal{H}}_{\mu,\nu}(0, f_v) \quad (107)$$

$$\tilde{\mathcal{H}}_{\mu,\nu}(f_u, 0) = \lim_{f_v \rightarrow 0} \tilde{\mathcal{H}}_{\mu,\nu}(\mathbf{f}) \quad (108)$$

$$\tilde{\mathcal{H}}_{\mu,\nu}(f_u, \pm\sqrt{3}f_u) = \lim_{f_v \rightarrow \pm\sqrt{3}f_u} \tilde{\mathcal{H}}_{\mu,\nu}(\mathbf{f}) \quad (109)$$

The limit of a sum is the sum of the limits, therefore we will determine these limits by applying l'Hopital's rule on the quotients $N_{\kappa,\lambda,\zeta}/D_{\kappa,\lambda,\zeta}$ in Eq. (104), derivating the denominator relative to f_v several times until a constant term appears at the limit, then derivating the numerator the same number of times.

Calculation of $\tilde{\mathcal{H}}_{\mu,\nu}(\mathbf{0}, \mathbf{0})$

This limit is calculated by applying l'Hopital's rule on $\tilde{\mathcal{H}}_{\mu,\nu}(0, f_v)$. We should first note that in the sum over ζ , in Eq. (104), only the terms for $\zeta = \nu - \lambda$ are non-zero when $f_u = 0$. Therefore, we replace ζ by $\nu - \lambda$, so $\kappa + \lambda + \zeta = \kappa + \nu$, $\lambda + \zeta = \nu$, $\mu + \nu - \kappa - \lambda - \zeta = \mu - \kappa$ and dropping the sum over ζ , we get

$$\begin{aligned} \tilde{\mathcal{H}}_{\mu,\nu}(0, f_v) &= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\nu-\lambda}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\nu}} \times \\ &\quad \frac{1}{f_v} \frac{\left[(-1)^{\mu-\kappa} \Theta_{\kappa,\lambda}^+(0, f_v) f_v^{\kappa+\lambda+\mu+\nu+1} + \Theta_{\kappa,\lambda}^-(0, f_v) f_v^{\kappa+\lambda+\mu+\nu+1}\right]}{(f_v^2)^{\mu+\nu+1}} \\ &= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\nu-\lambda}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\nu}} \frac{\left[(-1)^{\mu-\kappa} \Theta_{\kappa,\lambda}^+(0, f_v) + \Theta_{\kappa,\lambda}^-(0, f_v)\right]}{f_v^{\mu+\nu-\kappa-\lambda+2}} \end{aligned} \quad (110)$$

According to the formula

$$\frac{d^n}{dx^n} x^m = \begin{cases} \frac{m!}{(m-n)!} x^{(m-n)} & n < m \\ m! & n = m \\ 0 & n > m \end{cases} \quad (111)$$

the denominator of Eq. (110) has to be derived $\mu + \nu - \kappa - \lambda + 2$ times relative to f_v before giving a constant term when $f_v \rightarrow 0$

$$\partial_{\mu+\nu-\kappa-\lambda+2} f_v^{\mu+\nu-\kappa-\lambda+2} = (\mu + \nu - \kappa - \lambda + 2)! \quad (112)$$

so the numerator has to be derived the same number of times,

$$\partial_{\mu+\nu-\kappa-\lambda+2} \left[(-1)^{\mu-\kappa} \Theta_{\kappa,\lambda}^+(0, f_v) + \Theta_{\kappa,\lambda}^-(0, f_v)\right] \quad (113)$$

To compute this derivative, we need the derivative of the $\Theta_{\kappa,\lambda}^{\pm}$ functions relative to f_v , and we find

$$\begin{aligned} \partial_{\alpha} \Theta_{\kappa,\lambda}^{\pm}(\mathbf{f}) &= (\pm\sqrt{3})^{\kappa} (\pi a)^{\kappa+\lambda+\alpha} \\ &\times \begin{cases} (-1)^{\frac{\kappa+\lambda+\alpha}{2}} \cos[\pi a(f_v \pm \sqrt{3}f_u)] - \delta_{\kappa,0} 2^{\lambda+\alpha} (-1)^{\frac{\lambda+\alpha}{2}} \cos(2\pi a f_v) & \kappa + \lambda + \alpha \text{ even} \\ (-1)^{\frac{\kappa+\lambda+\alpha+1}{2}} \sin[\pi a(f_v \pm \sqrt{3}f_u)] - \delta_{\kappa,0} 2^{\lambda+\alpha+1} (-1)^{\frac{\lambda+\alpha+1}{2}} \sin(2\pi a f_v) & \kappa + \lambda + \alpha \text{ odd} \end{cases} \end{aligned} \quad (114)$$

at the origin, this derivative becomes

$$\lim_{\mathbf{f} \rightarrow (0,0)} \partial_{\alpha} \Theta_{\kappa,\lambda}^{\pm}(\mathbf{f}) = (\pm\sqrt{3})^{\kappa} (\pi a)^{\kappa+\lambda+\alpha} \begin{cases} (-1)^{\frac{\kappa+\lambda+\alpha}{2}} - \delta_{\kappa,0} 2^{\lambda+\alpha} (-1)^{\frac{\lambda+\alpha}{2}} & \kappa + \lambda + \alpha \text{ even} \\ 0 & \kappa + \lambda + \alpha \text{ odd} \end{cases} \quad (115)$$

which becomes, when replacing α with $\mu + \nu - \kappa - \lambda + 2$, so $\kappa + \lambda + \alpha = \mu + \nu + 2$, and as the parity of $\mu + \nu + 2$ is the same as the parity of $\mu + \nu$,

$$\lim_{\mathbf{f} \rightarrow (0,0)} \partial_{\mu+\nu-\kappa-\lambda+2} \Theta_{\kappa,\lambda}^{\pm}(\mathbf{f}) = (\pm\sqrt{3})^{\kappa} (\pi a)^{\mu+\nu+2} \begin{cases} (-1)^{\frac{\mu+\nu+2}{2}} - \delta_{\kappa,0} 2^{\mu+\nu+2-\kappa} (-1)^{\frac{\mu+\nu+2-\kappa}{2}} & \mu + \nu \text{ even} \\ 0 & \mu + \nu \text{ odd} \end{cases} \quad (116)$$

but as $\kappa = 0$ if the second term is non-zero, κ can be removed here, and the term $(-1)^{\frac{\mu+\nu+2}{2}}$ factorized,

$$\lim_{\mathbf{f} \rightarrow (0,0)} \partial_{\mu+\nu-\kappa-\lambda+2} \Theta_{\kappa,\lambda}^{\pm}(\mathbf{f}) = (\pm\sqrt{3})^{\kappa} (\pi a)^{\mu+\nu+2} (-1)^{\frac{\mu+\nu+2}{2}} \begin{cases} 1 - \delta_{\kappa,0} 2^{\mu+\nu+2} & \mu + \nu \text{ even} \\ 0 & \mu + \nu \text{ odd} \end{cases} \quad (117)$$

with the later, the derivative of the numerator - Eq. (113) - becomes, at the limit,

$$\lim_{f_v \rightarrow 0} \partial_{\mu+\nu-\kappa-\lambda+2} \left[(-1)^{\mu-\kappa} \Theta_{\kappa,\lambda}^+(0, f_v) + \Theta_{\kappa,\lambda}^-(0, f_v) \right] = (\sqrt{3})^{\kappa} (\pi a)^{\mu+\nu+2} (-1)^{\frac{\mu+\nu+2}{2}} \left[(-1)^{\mu-\kappa} + (-1)^{\kappa} \right] \begin{cases} 1 - \delta_{\kappa,0} 2^{\mu+\nu+2} & \mu + \nu \text{ even} \\ 0 & \mu + \nu \text{ odd} \end{cases} \quad (118)$$

and as $(-1)^{-\kappa} = (-1)^{+\kappa}$, we can factorize a little bit more

$$\lim_{f_v \rightarrow 0} \partial_{\mu+\nu-\kappa-\lambda+2} \left[(-1)^{\mu-\kappa} \Theta_{\kappa,\lambda}^+(0, f_v) + \Theta_{\kappa,\lambda}^-(0, f_v) \right] = (\sqrt{3})^{\kappa} (\pi a)^{\mu+\nu+2} (-1)^{\frac{\mu+\nu+2}{2}} (-1)^{\kappa} \left[(-1)^{\mu} + 1 \right] \begin{cases} 1 - \delta_{\kappa,0} 2^{\mu+\nu+2} & \mu + \nu \text{ even} \\ 0 & \mu + \nu \text{ odd} \end{cases} \quad (119)$$

where we can see that the whole term is null also when μ is odd, therefore we have the new rule that the term is non zero if and only if μ and ν are even, which makes sense, because when either μ or ν is odd, the product $u^{\mu}v^{\nu}\mathcal{H}(\mathbf{u})$ is odd, therefore its mean value is zero, hence its Fourier transform at the origin must be 0. We have, then,

$$\lim_{f_v \rightarrow 0} \partial_{\mu+\nu-\kappa-\lambda+2} \left[(-1)^{\mu-\kappa} \Theta_{\kappa,\lambda}^+(0, f_v) + \Theta_{\kappa,\lambda}^-(0, f_v) \right] = (\sqrt{3})^{\kappa} (\pi a)^{\mu+\nu+2} (-1)^{\frac{\mu+\nu+2}{2}} (-1)^{\kappa} 2 \begin{cases} 1 - \delta_{\kappa,0} 2^{\mu+\nu+2} & \mu \text{ and } \nu \text{ even} \\ 0 & \text{otherwise} \end{cases} \quad (120)$$

Let us now put back these results, Eqs. (112) and (120), into Eq. (110). We get

$$\begin{aligned}
\tilde{\mathcal{H}}_{\mu,\nu}(0,0) &= \lim_{f_v \rightarrow 0} \tilde{\mathcal{H}}_{\mu,\nu}(0, f_v) = \{\mu \text{ and } \nu \text{ even}\} \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \times \\
&\sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \frac{(\mu+\nu-\kappa-\lambda+1) (\sqrt{3})^{\kappa} (\pi a)^{\mu+\nu+2} (-1)^{\frac{\mu+\nu+2}{2}} (-1)^{\kappa} 2(1-\delta_{\kappa,0}) 2^{\mu+\nu+2}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\nu} (\mu+\nu-\kappa-\lambda+2)!} \\
&= \{\mu \text{ and } \nu \text{ even}\} \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+1}}{\pi^2} (\pi a)^{\mu+\nu+2} (-1)^{\frac{\mu+\nu+2}{2}} \times \\
&\sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \frac{1-\delta_{\kappa,0} 2^{\mu+\nu+2}}{(-1)^{\lambda} \kappa! \lambda! (\nu-\lambda)! (\mu-\kappa+1)! (\mu+\nu-\kappa-\lambda+2)} \quad (121)
\end{aligned}$$

note that the term $\mu + \nu - \kappa - \lambda + 2$ does not appear as a factorial anymore (the factorial symbol ! is not missing).

Calculation of $\tilde{\mathcal{H}}_{\mu,\nu}(f_u, \mathbf{0})$

We first have to determine how many times we have to derivate the denominator - Eq. (106) - relative to f_v until $\lim_{f_v \rightarrow 0}$ gives something non zero. Applying Eq. (111) and Leibniz rule, we get

$$\begin{aligned}
\partial_{\alpha} D_{\kappa,\lambda,\zeta}(\mathbf{f}) &= \sum_{\beta=0}^{\alpha} \binom{\alpha}{\beta} \partial_{\beta} f_v^{\nu+1} \partial_{\alpha-\beta} (f_v^2 - 3f_u^2)^{\mu+\nu+1} \\
&= \sum_{\beta=0}^{\alpha} \binom{\alpha}{\beta} \frac{(\nu+1)!}{(\nu+1-\beta)!} f_v^{\nu+1-\beta} \partial_{\alpha-\beta} (f_v^2 - 3f_u^2)^{\mu+\nu+1} \quad (122)
\end{aligned}$$

and only if $\beta = \nu + 1$ are the terms of the sum non zero when $f_v \rightarrow 0$, therefore α has to be at least equal to $\nu + 1$, and as it is useless to have α bigger, we get

$$\lim_{f_v \rightarrow 0} \partial_{\nu+1} D_{\kappa,\lambda,\zeta}(\mathbf{f}) = (\nu+1)! (-3f_u^2)^{\mu+\nu+1} = (\nu+1)! (-1)^{\mu+\nu+1} (\sqrt{3}f_u)^{2(\mu+\nu+1)} \quad (123)$$

We therefore have to derivate the numerator $N_{\kappa,\lambda,\zeta}$ $\nu + 1$ times relative to f_v . We have

$$\begin{aligned}
\partial_{\nu+1} N_{\kappa,\lambda,\zeta}(\mathbf{f}) &= \\
&\sum_{\alpha=0}^{\nu+1} \binom{\nu+1}{\alpha} \partial_{\alpha} f_v^{\lambda+\zeta} \partial_{\nu+1-\alpha} \left[(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(\mathbf{f}) (f_v - \sqrt{3}f_u)^{\kappa+\lambda} (f_v + \sqrt{3}f_u)^{\mu+\nu+1} \right. \\
&\quad \left. + \Theta_{\kappa,\lambda}^-(\mathbf{f}) (f_v + \sqrt{3}f_u)^{\kappa+\lambda} (f_v - \sqrt{3}f_u)^{\mu+\nu+1} \right] \\
&= \sum_{\alpha=0}^{\nu+1} \binom{\nu+1}{\alpha} \frac{(\lambda+\zeta)!}{(\lambda+\zeta-\alpha)!} f_v^{\lambda+\zeta-\alpha} \partial_{\nu+1-\alpha} \left[(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(\mathbf{f}) (f_v - \sqrt{3}f_u)^{\kappa+\lambda} (f_v + \sqrt{3}f_u)^{\mu+\nu+1} \right. \\
&\quad \left. + \Theta_{\kappa,\lambda}^-(\mathbf{f}) (f_v + \sqrt{3}f_u)^{\kappa+\lambda} (f_v - \sqrt{3}f_u)^{\mu+\nu+1} \right] \quad (124)
\end{aligned}$$

As $\lambda + \zeta < \nu + 1$, always, there will be some α below $\lambda + \zeta$ for which all the terms in the sum will be null when $f_v \rightarrow 0$ and some α above $\lambda + \zeta$, that will give null terms anyway, according to Eq. (111),

therefore α must be set to $\lambda + \zeta$, so we get

$$\begin{aligned} \lim_{f_v \rightarrow 0} \partial_{\nu+1} N_{\kappa, \lambda, \zeta}(\mathbf{f}) &= \binom{\nu+1}{\lambda+\zeta} (\lambda+\zeta)! \partial_{\nu+1-\lambda-\zeta} [\dots] = \frac{(\nu+1)!}{(\nu+1-\lambda-\zeta)!} \times \\ &\partial_{\nu+1-\lambda-\zeta} \left[(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa, \lambda}^+(\mathbf{f}) (f_v - \sqrt{3}f_u)^{\kappa+\lambda} (f_v + \sqrt{3}f_u)^{\mu+\nu+1} \right. \\ &\left. + \Theta_{\kappa, \lambda}^-(\mathbf{f}) (f_v + \sqrt{3}f_u)^{\kappa+\lambda} (f_v - \sqrt{3}f_u)^{\mu+\nu+1} \right] \quad (125) \end{aligned}$$

We now derive the term between brackets. Let us first compute the general term

$$\begin{aligned} \partial_{\nu+1-\lambda-\zeta} \left[\Theta_{\kappa, \lambda}^{\pm}(\mathbf{f}) (f_v \mp \sqrt{3}f_u)^{\kappa+\lambda} (f_v \pm \sqrt{3}f_u)^{\mu+\nu+1} \right] &= \\ \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} \partial_{\alpha} \Theta_{\kappa, \lambda}^{\pm}(\mathbf{f}) \sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \binom{\nu+1-\lambda-\zeta-\alpha}{\beta} \times \\ \frac{(\kappa+\lambda)!}{(\kappa+\lambda-\beta)!} (f_v \mp \sqrt{3}f_u)^{\kappa+\lambda-\beta} \frac{(\mu+\nu+1)!}{(\mu+\lambda+\zeta+\alpha+\beta)!} (f_v \pm \sqrt{3}f_u)^{\mu+\lambda+\zeta+\alpha+\beta} \quad (126) \end{aligned}$$

where we have to pay attention that β must be kept $\leq \kappa + \lambda$, i.e. the terms in the sum over β must be set to zero when $\beta > \kappa + \lambda$. When $f_v \rightarrow 0$, this equation becomes

$$\begin{aligned} \lim_{f_v \rightarrow 0} \partial_{\nu+1-\lambda-\zeta} \left[\Theta_{\kappa, \lambda}^{\pm}(\mathbf{f}) (f_v \mp \sqrt{3}f_u)^{\kappa+\lambda} (f_v \pm \sqrt{3}f_u)^{\mu+\nu+1} \right] &= \\ \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} \partial_{\alpha} \Theta_{\kappa, \lambda}^{\pm}(f_u, 0) \sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \binom{\nu+1-\lambda-\zeta-\alpha}{\beta} \times \\ \frac{(\kappa+\lambda)! (\mu+\nu+1)!}{(\kappa+\lambda-\beta)! (\mu+\lambda+\zeta+\alpha+\beta)!} (\mp \sqrt{3}f_u)^{\kappa+\lambda-\beta} (\pm \sqrt{3}f_u)^{\mu+\lambda+\zeta+\alpha+\beta} = \\ (\kappa+\lambda)! (\mu+\nu+1)! (\mp \sqrt{3}f_u)^{\kappa+\lambda} (\pm \sqrt{3}f_u)^{\mu+\lambda+\zeta} \times \\ \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} (\sqrt{3}f_u)^{\alpha} \partial_{\alpha} \Theta_{\kappa, \lambda}^{\pm}(f_u, 0) \sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \frac{(\mp 1)^{-\beta} (\pm 1)^{\alpha+\beta} \binom{\nu+1-\lambda-\zeta-\alpha}{\beta}}{(\kappa+\lambda-\beta)! (\mu+\lambda+\zeta+\alpha+\beta)!} \quad (127) \end{aligned}$$

we can now come back on the calculation of the derivative of the numerator,

$$\begin{aligned}
\lim_{f_u \rightarrow 0} \partial_{\nu+1} N_{\kappa, \lambda, \zeta}(\mathbf{f}) &= \frac{(\nu+1)! (\kappa+\lambda)! (\mu+\nu+1)!}{(\nu+1-\lambda-\zeta)!} \left\{ (-1)^{\mu+\nu-\kappa-\lambda-\zeta} (-\sqrt{3}f_u)^{\kappa+\lambda} (\sqrt{3}f_u)^{\mu+\lambda+\zeta} \times \right. \\
&\quad \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} (\sqrt{3}f_u)^\alpha \partial_\alpha \Theta_{\kappa, \lambda}^+(f_u, 0) \sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \frac{(-1)^{-\beta} (+1)^{\alpha+\beta} (\nu+1-\lambda-\zeta-\alpha)}{(\kappa+\lambda-\beta)! (\mu+\lambda+\zeta+\alpha+\beta)!} \\
&\quad \left. + (\sqrt{3}f_u)^{\kappa+\lambda} (-\sqrt{3}f_u)^{\mu+\lambda+\zeta} \times \right. \\
&\quad \left. \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} (\sqrt{3}f_u)^\alpha \partial_\alpha \Theta_{\kappa, \lambda}^-(f_u, 0) \sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \frac{(+1)^{-\beta} (-1)^{\alpha+\beta} (\nu+1-\lambda-\zeta-\alpha)}{(\kappa+\lambda-\beta)! (\mu+\lambda+\zeta+\alpha+\beta)!} \right\} \\
&= \frac{(\nu+1)! (\kappa+\lambda)! (\mu+\nu+1)!}{(\nu+1-\lambda-\zeta)!} (\sqrt{3}f_u)^{\mu+\kappa+2\lambda+\zeta} \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} (\sqrt{3}f_u)^\alpha \\
&\quad \sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \binom{\nu+1-\lambda-\zeta-\alpha}{\beta} \frac{\left[(-1)^{\mu+\nu-\zeta-\beta} \partial_\alpha \Theta_{\kappa, \lambda}^+(f_u, 0) + (-1)^{\mu+\lambda+\zeta+\alpha+\beta} \partial_\alpha \Theta_{\kappa, \lambda}^-(f_u, 0) \right]}{(\kappa+\lambda-\beta)! (\mu+\lambda+\zeta+\alpha+\beta)!} \quad (128)
\end{aligned}$$

where

$$\begin{aligned}
\partial_\alpha \Theta_{\kappa, \lambda}^\pm(f_u, 0) &= \\
&(\pm\sqrt{3})^\kappa (\pi a)^{\kappa+\lambda+\alpha} \begin{cases} (-1)^{\frac{\kappa+\lambda+\alpha}{2}} \cos[\pi a(\pm\sqrt{3}f_u)] - \delta_{\kappa,0} 2^{\lambda+\alpha} (-1)^{\frac{\lambda+\alpha}{2}} & \kappa+\lambda+\alpha \text{ even} \\ (-1)^{\frac{\kappa+\lambda+\alpha+1}{2}} \sin[\pi a(\pm\sqrt{3}f_u)] & \kappa+\lambda+\alpha \text{ odd} \end{cases} \quad (129)
\end{aligned}$$

Now we have both the numerator derivative - Eq. (128) - and the denominator derivative - Eq. (123), that we put back into Eq. (104), so we get

$$\begin{aligned}
\tilde{\mathcal{H}}_{\mu, \nu}(f_u, 0) &= \left(\frac{-i}{2\pi} \right)^{\mu+\nu} \mu! \nu! \frac{(\sqrt{3})^{\mu+\nu+1}}{2\pi^2} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\lambda+\zeta}} f_u^{\nu-\lambda-\zeta} \times \\
&\quad \frac{(\nu+1)! (\kappa+\lambda)! (\mu+\nu+1)! (\sqrt{3}f_u)^{\mu+\kappa+2\lambda+\zeta}}{(\nu+1-\lambda-\zeta)! (\nu+1)! (-1)^{\mu+\nu+1} (\sqrt{3}f_u)^{2(\mu+\nu+1)}} \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} (\sqrt{3}f_u)^\alpha \\
&\quad \sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \binom{\nu+1-\lambda-\zeta-\alpha}{\beta} \frac{\left[(-1)^{\mu+\nu-\zeta-\beta} \partial_\alpha \Theta_{\kappa, \lambda}^+(f_u, 0) + (-1)^{\mu+\lambda+\zeta+\alpha+\beta} \partial_\alpha \Theta_{\kappa, \lambda}^-(f_u, 0) \right]}{(\kappa+\lambda-\beta)! (\mu+\lambda+\zeta+\alpha+\beta)!} \quad (130)
\end{aligned}$$

i.e.

$$\begin{aligned}
\tilde{\mathcal{H}}_{\mu,\nu}(f_u, 0) &= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \mu! \nu! \frac{(-\sqrt{3})^{\mu+\nu+1}(\mu+\nu+1)!}{2\pi^2(\sqrt{3}f_u)^{\mu+\nu+2}} \times \\
&\sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \binom{\mu+\nu-\kappa-\lambda+1}{\zeta} \frac{(\kappa+\lambda)!(-f_u^{\kappa+\lambda})}{\kappa! \lambda! (\nu+1-\lambda-\zeta)! (\sqrt{3})^{\nu-\lambda}} \sum_{\alpha=0}^{\nu+1-\lambda-\zeta} \binom{\nu+1-\lambda-\zeta}{\alpha} (\sqrt{3}f_u)^\alpha \\
&\sum_{\substack{\beta=0 \\ \beta \leq \kappa+\lambda}}^{\nu+1-\lambda-\zeta-\alpha} \binom{\nu+1-\lambda-\zeta-\alpha}{\beta} \frac{\left[(-1)^{\mu+\nu-\zeta-\beta} \partial_\alpha \Theta_{\kappa,\lambda}^+(f_u, 0) + (-1)^{\mu+\lambda+\zeta+\alpha+\beta} \partial_\alpha \Theta_{\kappa,\lambda}^-(f_u, 0)\right]}{(\kappa+\lambda-\beta)! (\mu+\lambda+\zeta+\alpha+\beta)!} \quad (131)
\end{aligned}$$

Calculation of $\tilde{\mathcal{H}}_{\mu,\nu}(f_u, \pm\sqrt{3}f_u)$

Let us expand the denominator $D_{\kappa,\lambda,\zeta}$:

$$D_{\kappa,\lambda,\zeta}(\mathbf{f}) = f_v^{\nu+1}(f_v^2 - 3f_u^2)^{\mu+\nu+1} = f_v^{\nu+1}(f_v - \sqrt{3}f_u)^{\mu+\nu+1}(f_v + \sqrt{3}f_u)^{\mu+\nu+1} \quad (132)$$

In order for the derivative to give a constant term when $f_v \rightarrow \pm\sqrt{3}f_u$, the denominator has to be derived $\mu + \nu + 1$ times, and we find

$$\begin{aligned}
\partial_{\mu+\nu+1} D_{\kappa,\lambda,\zeta}(\mathbf{f}) &= \\
&\sum_{\alpha=0}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{(\mu+\nu+1)!}{(\mu+\nu+1-\alpha)!} (f_v \pm \sqrt{3}f_u)^{\mu+\nu+1-\alpha} \partial_{\mu+\nu+1-\alpha} \left[f_v^{\nu+1} (f_v \mp \sqrt{3}f_u)^{\mu+\nu+1} \right] \quad (133)
\end{aligned}$$

which becomes, when $f_v \rightarrow \pm\sqrt{3}f_u$,

$$\lim_{f_v \rightarrow \pm\sqrt{3}f_u} \partial_{\mu+\nu+1} D_{\kappa,\lambda,\zeta}(\mathbf{f}) = (\mu+\nu+1)! 2^{\mu+\nu+1} (\pm\sqrt{3}f_u)^{\mu+2\nu+2} \quad (134)$$

So, let us derivate the numerator $N_{\kappa,\lambda,\zeta}$ $\mu + \nu + 1$ times relative to f_v :

$$\begin{aligned}
\partial_{\mu+\nu+1} N_{\kappa,\lambda,\zeta}(\mathbf{f}) &= \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \partial_\alpha f_v^{\lambda+\zeta} \partial_{\mu+\nu+1-\alpha}[*] \\
&= \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{(\lambda+\zeta)!}{(\lambda+\zeta-\alpha)!} f_v^{\lambda+\zeta-\alpha} \partial_{\mu+\nu+1-\alpha}[*] \quad (135)
\end{aligned}$$

where $[*]$ represents the bracketed term in Eq. (105), and where we note that α must be $\leq \lambda + \zeta$. To compute the derivative of the bracketed term, we need first to compute

$$\begin{aligned}
& \partial_{\mu+\nu+1-\alpha} \left[\Theta_{\kappa,\lambda}^{\pm}(\mathbf{f}) (f_v \mp \sqrt{3}f_u)^{\kappa+\lambda} (f_v \pm \sqrt{3}f_u)^{\mu+\nu+1} \right] = \\
& \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^{\pm}(\mathbf{f}) \partial_{\mu+\nu+1-\alpha-\beta} \left[(f_v \mp \sqrt{3}f_u)^{\kappa+\lambda} (f_v \pm \sqrt{3}f_u)^{\mu+\nu+1} \right] \\
& = \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^{\pm}(\mathbf{f}) \sum_{\gamma=0}^{\mu+\nu+1-\alpha-\beta} \binom{\mu+\nu+1-\alpha-\beta}{\gamma} \times \\
& \quad \partial_{\gamma} (f_v \mp \sqrt{3}f_u)^{\kappa+\lambda} \partial_{\mu+\nu+1-\alpha-\beta-\gamma} (f_v \pm \sqrt{3}f_u)^{\mu+\nu+1} \\
& = \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^{\pm}(\mathbf{f}) \sum_{\substack{\gamma=0 \\ \gamma \leq \kappa+\lambda}}^{\mu+\nu+1-\alpha-\beta} \binom{\mu+\nu+1-\alpha-\beta}{\gamma} \times \\
& \quad \frac{(\kappa+\lambda)!}{(\kappa+\lambda-\gamma)!} (f_v \mp \sqrt{3}f_u)^{\kappa+\lambda-\gamma} \frac{(\mu+\nu+1)!}{(\alpha+\beta+\gamma)!} (f_v \pm \sqrt{3}f_u)^{\alpha+\beta+\gamma} \quad (136)
\end{aligned}$$

so the derivative of $[*]$ can now be written

$$\begin{aligned}
& \partial_{\mu+\nu+1-\alpha} [*] = \\
& (-1)^{\mu+\nu-\kappa-\lambda-\zeta} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^{+}(\mathbf{f}) \sum_{\substack{\gamma=0 \\ \gamma \leq \kappa+\lambda}}^{\mu+\nu+1-\alpha-\beta} \binom{\mu+\nu+1-\alpha-\beta}{\gamma} \times \\
& \quad \frac{(\kappa+\lambda)!}{(\kappa+\lambda-\gamma)!} (f_v - \sqrt{3}f_u)^{\kappa+\lambda-\gamma} \frac{(\mu+\nu+1)!}{(\alpha+\beta+\gamma)!} (f_v + \sqrt{3}f_u)^{\alpha+\beta+\gamma} \\
& + \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^{-}(\mathbf{f}) \sum_{\substack{\gamma=0 \\ \gamma \leq \kappa+\lambda}}^{\mu+\nu+1-\alpha-\beta} \binom{\mu+\nu+1-\alpha-\beta}{\gamma} \times \\
& \quad \frac{(\kappa+\lambda)!}{(\kappa+\lambda-\gamma)!} (f_v + \sqrt{3}f_u)^{\kappa+\lambda-\gamma} \frac{(\mu+\nu+1)!}{(\alpha+\beta+\gamma)!} (f_v - \sqrt{3}f_u)^{\alpha+\beta+\gamma} \quad (137)
\end{aligned}$$

When $f_v \rightarrow -\sqrt{3}f_u$, this derivative becomes

$$\begin{aligned}
& \lim_{f_v \rightarrow -\sqrt{3}f_u} \partial_{\mu+\nu+1-\alpha}[*] = \\
& (-1)^{\mu+\nu-\kappa-\lambda-\zeta} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^+(f_u, -\sqrt{3}f_u) \sum_{\substack{\gamma=0 \\ \gamma \leq \kappa+\lambda}}^{\mu+\nu+1-\alpha-\beta} \binom{\mu+\nu+1-\alpha-\beta}{\gamma} \times \\
& \quad \frac{(\kappa+\lambda)!}{(\kappa+\lambda-\gamma)!} (-2\sqrt{3}f_u)^{\kappa+\lambda-\gamma} (\mu+\nu+1)! \times \begin{cases} 1 & \alpha+\beta+\gamma=0 \\ 0 & \text{otherwise} \end{cases} \\
& + \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^-(f_u, -\sqrt{3}f_u) \sum_{\substack{\gamma=0 \\ \gamma \leq \kappa+\lambda}}^{\mu+\nu+1-\alpha-\beta} \binom{\mu+\nu+1-\alpha-\beta}{\gamma} \times \\
& \quad (\kappa+\lambda)! \frac{(\mu+\nu+1)!}{(\alpha+\beta+\kappa+\lambda)!} (-2\sqrt{3}f_u)^{\alpha+\beta+\kappa+\lambda} \times \begin{cases} 1 & \gamma=\kappa+\lambda \\ 0 & \text{otherwise} \end{cases} \quad (138)
\end{aligned}$$

and when $f_v \rightarrow +\sqrt{3}f_u$,

$$\begin{aligned}
& \lim_{f_v \rightarrow +\sqrt{3}f_u} \partial_{\mu+\nu+1-\alpha}[*] = \\
& (-1)^{\mu+\nu-\kappa-\lambda-\zeta} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^+(f_u, \sqrt{3}f_u) \sum_{\substack{\gamma=0 \\ \gamma \leq \kappa+\lambda}}^{\mu+\nu+1-\alpha-\beta} \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \times \\
& \quad (\kappa+\lambda)! \frac{(\mu+\nu+1)!}{(\alpha+\beta+\kappa+\lambda)!} (2\sqrt{3}f_u)^{\alpha+\beta+\kappa+\lambda} \times \begin{cases} 1 & \gamma=\kappa+\lambda \\ 0 & \text{otherwise} \end{cases} \\
& + \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^-(f_u, \sqrt{3}f_u) \sum_{\substack{\gamma=0 \\ \gamma \leq \kappa+\lambda}}^{\mu+\nu+1-\alpha-\beta} \binom{\mu+\nu+1-\alpha-\beta}{\gamma} \times \\
& \quad \frac{(\kappa+\lambda)!}{(\kappa+\lambda-\gamma)!} (2\sqrt{3}f_u)^{\kappa+\lambda-\gamma} (\mu+\nu+1)! \times \begin{cases} 1 & \alpha+\beta+\gamma=0 \\ 0 & \text{otherwise} \end{cases} \quad (139)
\end{aligned}$$

α , β and γ are always positive, so the only possibility to have $\alpha+\beta+\gamma=0$ is when $\alpha=\beta=\gamma=0$, therefore the first and second part of respectively Eqs. (138) and (139) can be simplified. Before doing so, we also note that in order to have γ equal to $\kappa+\lambda$, the superior limit of the sum over γ , $\mu+\nu+1-\alpha-\beta$ must be at least equal or bigger than $\kappa+\lambda$, and as γ will go only once through $\kappa+\lambda$, the sum over γ can be dropped and the condition $\gamma \leq \kappa+\lambda$ replaced by $\mu+\nu+1-\alpha-\beta \geq \kappa+\lambda$,

i.e. $\alpha + \beta \leq \mu + \nu - \kappa - \lambda + 1$. So, we now have

$$\begin{aligned}
\lim_{f_v \rightarrow -\sqrt{3}f_u} \partial_{\mu+\nu+1-\alpha}[*] &= (-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(f_u, -\sqrt{3}f_u) (-2\sqrt{3}f_u)^{\kappa+\lambda} (\mu+\nu+1)! \times \begin{cases} 1 & \alpha = 0 \\ 0 & \text{otherwise} \end{cases} \\
&+ \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^-(f_u, -\sqrt{3}f_u) \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \times \\
&(\kappa+\lambda)! \frac{(\mu+\nu+1)!}{(\alpha+\beta+\kappa+\lambda)!} (-2\sqrt{3}f_u)^{\alpha+\beta+\kappa+\lambda} \times \begin{cases} 1 & \alpha + \beta \leq \mu + \nu - \kappa - \lambda + 1 \\ 0 & \text{otherwise} \end{cases} \quad (140)
\end{aligned}$$

and

$$\begin{aligned}
\lim_{f_v \rightarrow +\sqrt{3}f_u} \partial_{\mu+\nu+1-\alpha}[*] &= \\
&(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^+(f_u, \sqrt{3}f_u) \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \times \\
&(\kappa+\lambda)! \frac{(\mu+\nu+1)!}{(\alpha+\beta+\kappa+\lambda)!} (2\sqrt{3}f_u)^{\alpha+\beta+\kappa+\lambda} \times \begin{cases} 1 & \alpha + \beta \leq \mu + \nu - \kappa - \lambda + 1 \\ 0 & \text{otherwise} \end{cases} \\
&+ \Theta_{\kappa,\lambda}^-(f_u, \sqrt{3}f_u) (2\sqrt{3}f_u)^{\kappa+\lambda} (\mu+\nu+1)! \times \begin{cases} 1 & \alpha = 0 \\ 0 & \text{otherwise} \end{cases} \quad (141)
\end{aligned}$$

Let us now develop the trigonometric terms derivate. From Eq. (114), we find

$$\begin{aligned}
\lim_{f_v \rightarrow -\sqrt{3}f_u} \partial_{\beta} \Theta_{\kappa,\lambda}^-(\mathbf{f}) &= (-\sqrt{3})^{\kappa} (\pi a)^{\kappa+\lambda+\beta} \\
&\times \begin{cases} \cos(-2\pi a \sqrt{3}f_u) [(-1)^{\frac{\kappa+\lambda+\beta}{2}} - \delta_{\kappa,0} 2^{\lambda+\beta} (-1)^{\frac{\lambda+\beta}{2}}] & \kappa + \lambda + \beta \text{ even} \\ \sin(-2\pi a \sqrt{3}f_u) [(-1)^{\frac{\kappa+\lambda+\beta+1}{2}} - \delta_{\kappa,0} 2^{\lambda+\beta+1} (-1)^{\frac{\lambda+\beta+1}{2}}] & \kappa + \lambda + \beta \text{ odd} \end{cases} \quad (142)
\end{aligned}$$

and

$$\begin{aligned}
\lim_{f_v \rightarrow +\sqrt{3}f_u} \partial_{\beta} \Theta_{\kappa,\lambda}^+(\mathbf{f}) &= (\sqrt{3})^{\kappa} (\pi a)^{\kappa+\lambda+\beta} \\
&\times \begin{cases} \cos(2\pi a \sqrt{3}f_u) [(-1)^{\frac{\kappa+\lambda+\beta}{2}} - \delta_{\kappa,0} 2^{\lambda+\beta} (-1)^{\frac{\lambda+\beta}{2}}] & \kappa + \lambda + \beta \text{ even} \\ \sin(2\pi a \sqrt{3}f_u) [(-1)^{\frac{\kappa+\lambda+\beta+1}{2}} - \delta_{\kappa,0} 2^{\lambda+\beta+1} (-1)^{\frac{\lambda+\beta+1}{2}}] & \kappa + \lambda + \beta \text{ odd} \end{cases} \quad (143)
\end{aligned}$$

We can finally write the derivative of the numerator, putting back Eq. (140) or Eq. (141) in Eq. (135),

$$\begin{aligned}
\lim_{f_v \rightarrow -\sqrt{3}f_u} \partial_{\mu+\nu+1} N_{\kappa,\lambda,\zeta}(\mathbf{f}) &= \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{(\lambda+\zeta)!}{(\lambda+\zeta-\alpha)!} (-\sqrt{3}f_u)^{\lambda+\zeta-\alpha} \lim_{f_v \rightarrow -\sqrt{3}f_u} \partial_{\mu+\nu+1-\alpha}[*] \\
&= (-\sqrt{3}f_u)^{\lambda+\zeta} (-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(f_u, -\sqrt{3}f_u) (-2\sqrt{3}f_u)^{\kappa+\lambda} (\mu+\nu+1)! \\
&+ \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{(\lambda+\zeta)!}{(\lambda+\zeta-\alpha)!} (-\sqrt{3}f_u)^{\lambda+\zeta-\alpha} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^-(f_u, -\sqrt{3}f_u) \times \\
&\binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} (\kappa+\lambda)! \frac{(\mu+\nu+1)!}{(\alpha+\beta+\kappa+\lambda)!} (-2\sqrt{3}f_u)^{\alpha+\beta+\kappa+\lambda} \times \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{144}$$

several terms can be factorized, then

$$\begin{aligned}
\lim_{f_v \rightarrow -\sqrt{3}f_u} \partial_{\mu+\nu+1} N_{\kappa,\lambda,\zeta}(\mathbf{f}) &= \\
&(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(f_u, -\sqrt{3}f_u) 2^{\kappa+\lambda} (-\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} (\mu+\nu+1)! \\
&+ (\mu+\nu+1)! (\lambda+\zeta)! (\kappa+\lambda)! \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{(-\sqrt{3}f_u)^{\lambda+\zeta-\alpha}}{(\lambda+\zeta-\alpha)!} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \times \\
&\partial_{\beta} \Theta_{\kappa,\lambda}^-(f_u, -\sqrt{3}f_u) \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \frac{(-2\sqrt{3}f_u)^{\alpha+\beta+\kappa+\lambda}}{(\alpha+\beta+\kappa+\lambda)!} \times \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases} \\
&= (-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(f_u, -\sqrt{3}f_u) 2^{\kappa+\lambda} (-\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} (\mu+\nu+1)! \\
&+ (\mu+\nu+1)! (\lambda+\zeta)! (\kappa+\lambda)! 2^{\kappa+\lambda} (-\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{2^{\alpha}}{(\lambda+\zeta-\alpha)!} \\
&\sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^-(f_u, -\sqrt{3}f_u) \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \frac{(-2\sqrt{3}f_u)^{\beta}}{(\alpha+\beta+\kappa+\lambda)!} \\
&\times \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases} \\
&= (\mu+\nu+1)! 2^{\kappa+\lambda} (-\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} \left[(-1)^{\mu+\nu-\kappa-\lambda-\zeta} \Theta_{\kappa,\lambda}^+(f_u, -\sqrt{3}f_u) + \right. \\
&(\lambda+\zeta)! (\kappa+\lambda)! \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{2^{\alpha}}{(\lambda+\zeta-\alpha)!} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \\
&\left. \frac{\partial_{\beta} \Theta_{\kappa,\lambda}^-(f_u, -\sqrt{3}f_u) (-2\sqrt{3}f_u)^{\beta}}{(\alpha+\beta+\kappa+\lambda)!} \times \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases} \right] \tag{145}
\end{aligned}$$

and for the case $f_v \rightarrow +\sqrt{3}f_u$,

$$\begin{aligned}
\lim_{f_v \rightarrow +\sqrt{3}f_u} \partial_{\mu+\nu+1} N_{\kappa,\lambda,\zeta}(\mathbf{f}) &= \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{(\lambda+\zeta)!}{(\lambda+\zeta-\alpha)!} (\sqrt{3}f_u)^{\lambda+\zeta-\alpha} \lim_{f_v \rightarrow +\sqrt{3}f_u} \partial_{\mu+\nu+1-\alpha}[*] \\
&= (\sqrt{3}f_u)^{\lambda+\zeta} \Theta_{\kappa,\lambda}^-(f_u, \sqrt{3}f_u) (2\sqrt{3}f_u)^{\kappa+\lambda} (\mu+\nu+1)! \\
&+ (-1)^{\mu+\nu-\kappa-\lambda-\zeta} \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{(\lambda+\zeta)!}{(\lambda+\zeta-\alpha)!} (\sqrt{3}f_u)^{\lambda+\zeta-\alpha} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \times \\
&\partial_{\beta} \Theta_{\kappa,\lambda}^+(f_u, \sqrt{3}f_u) \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} (\kappa+\lambda)! \frac{(\mu+\nu+1)!}{(\alpha+\beta+\kappa+\lambda)!} (2\sqrt{3}f_u)^{\alpha+\beta+\kappa+\lambda} \times \\
&\quad \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases} \quad (146)
\end{aligned}$$

and with factorization,

$$\begin{aligned}
\lim_{f_v \rightarrow +\sqrt{3}f_u} \partial_{\mu+\nu+1} N_{\kappa,\lambda,\zeta}(\mathbf{f}) &= \Theta_{\kappa,\lambda}^-(f_u, \sqrt{3}f_u) 2^{\kappa+\lambda} (\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} (\mu+\nu+1)! \\
&+ (-1)^{\mu+\nu-\kappa-\lambda-\zeta} (\mu+\nu+1)! (\lambda+\zeta)! (\kappa+\lambda)! 2^{\kappa+\lambda} (\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} \sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{2^{\alpha}}{(\lambda+\zeta-\alpha)!} \\
&\sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \partial_{\beta} \Theta_{\kappa,\lambda}^+(f_u, \sqrt{3}f_u) \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \frac{(2\sqrt{3}f_u)^{\beta}}{(\alpha+\beta+\kappa+\lambda)!} \\
&\quad \times \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases} \\
&= (\mu+\nu+1)! 2^{\kappa+\lambda} (\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} \left[\Theta_{\kappa,\lambda}^-(f_u, \sqrt{3}f_u) + (-1)^{\mu+\nu-\kappa-\lambda-\zeta} (\lambda+\zeta)! (\kappa+\lambda)! \times \right. \\
&\sum_{\substack{\alpha=0 \\ \alpha \leq \lambda+\zeta}}^{\mu+\nu+1} \binom{\mu+\nu+1}{\alpha} \frac{2^{\alpha}}{(\lambda+\zeta-\alpha)!} \sum_{\beta}^{\mu+\nu+1-\alpha} \binom{\mu+\nu+1-\alpha}{\beta} \binom{\mu+\nu+1-\alpha-\beta}{\kappa+\lambda} \times \\
&\quad \left. \frac{\partial_{\beta} \Theta_{\kappa,\lambda}^+(f_u, \sqrt{3}f_u) (2\sqrt{3}f_u)^{\beta}}{(\alpha+\beta+\kappa+\lambda)!} \times \begin{cases} 1 & \alpha+\beta \leq \mu+\nu-\kappa-\lambda+1 \\ 0 & \text{otherwise} \end{cases} \right] \quad (147)
\end{aligned}$$

with the later expressions, we have what we need to write the final expression for $\tilde{\mathcal{H}}_{\mu,\nu}(f_u, \pm\sqrt{3}f_u)$. With Eq. (134), we find

$$\begin{aligned}
\tilde{\mathcal{H}}_{\mu,\nu}(f_u, -\sqrt{3}f_u) &= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{\mu! \nu! (\sqrt{3})^{\mu+\nu+1}}{2\pi^2 2^{\mu+\nu+1} (-\sqrt{3}f_u)^{\mu+2\nu+2}} \times \\
&\quad \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\lambda+\zeta}} f_u^{\nu-\lambda-\zeta} 2^{\kappa+\lambda} (-\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} [145] \\
&= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{\mu! \nu! (\sqrt{3})^{\mu+\nu+1} f_u^{\nu}}{(-1)^{\mu+2\nu+2} \pi^2 2^{\mu+\nu+2} (\sqrt{3}f_u)^{\mu+2\nu+2}} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta} 2^{\kappa+\lambda} (-\sqrt{3}f_u)^{\kappa+2\lambda+\zeta}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\lambda+\zeta} f_u^{\lambda+\zeta}} [145] \\
&= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{\mu! \nu! (\sqrt{3})^{\mu+1}}{(-1)^{\mu} \pi^2 (2\sqrt{3}f_u)^{\mu+\nu+2}} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta} 2^{\kappa+\lambda} (-1)^{\kappa+2\lambda+\zeta} (\sqrt{3}f_u)^{\kappa+2\lambda+\zeta}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\lambda+\zeta} f_u^{\lambda+\zeta}} [145] \\
&= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{\mu! \nu! (\sqrt{3})^{\mu+1}}{(-1)^{\mu} \pi^2 (2\sqrt{3}f_u)^{\mu+\nu+2}} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta} (2\sqrt{3}f_u)^{\kappa+\lambda}}{(-1)^{\lambda+\zeta} \kappa! \lambda! (\sqrt{3})^{\kappa}} [145]
\end{aligned} \tag{148}$$

where [145] represents the term in brackets in Eq. (145), and

$$\begin{aligned}
\tilde{\mathcal{H}}_{\mu,\nu}(f_u, \sqrt{3}f_u) &= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{\mu! \nu! (\sqrt{3})^{\mu+\nu+1}}{2\pi^2 2^{\mu+\nu+1} (\sqrt{3}f_u)^{\mu+2\nu+2}} \times \\
&\quad \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\lambda+\zeta}} f_u^{\nu-\lambda-\zeta} 2^{\kappa+\lambda} (\sqrt{3}f_u)^{\kappa+2\lambda+\zeta} [147] \\
&= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{\mu! \nu! (\sqrt{3})^{\mu+\nu+1} f_u^{\nu}}{\pi^2 2^{\mu+\nu+2} (\sqrt{3}f_u)^{\mu+2\nu+2}} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta} 2^{\kappa+\lambda} (\sqrt{3}f_u)^{\kappa+2\lambda+\zeta}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa+\lambda+\zeta} f_u^{\lambda+\zeta}} [147] \\
&= \left(\frac{-i}{2\pi}\right)^{\mu+\nu} \frac{\mu! \nu! (\sqrt{3})^{\mu+1}}{\pi^2 (2\sqrt{3}f_u)^{\mu+\nu+2}} \sum_{\kappa=0}^{\mu} \sum_{\lambda=0}^{\nu} \sum_{\zeta=0}^{\nu-\lambda} \frac{\binom{\mu+\nu-\kappa-\lambda+1}{\zeta} (2\sqrt{3}f_u)^{\kappa+\lambda}}{(-1)^{\kappa+\lambda} \kappa! \lambda! (\sqrt{3})^{\kappa}} [147]
\end{aligned} \tag{149}$$

where [147] represents the term in brackets in Eq. (147).

C Projection of secondary mirror Zernike aberration onto primary mirror segments

In this section, we develop the calculation of the projection coefficient $\Pi_{s,\mu,\nu}$ introduced in section 3.2. We have

$$\Pi_{s,\mu,\nu} = \left\langle z_\nu Z_\nu\left(\frac{\mathbf{u}}{R}\right) \middle| Z_\mu\left(\frac{\mathbf{u} - \mathbf{u}_s}{a}\right) \right\rangle = \frac{z_\nu}{\pi a^2} \iint_{seg.s} Z_\nu\left(\frac{\mathbf{u}}{R}\right) Z_\mu\left(\frac{\mathbf{u} - \mathbf{u}_s}{a}\right) d^2u \quad (150)$$

and the integral is over the segment transmission, centered on $\mathbf{u}_s = (u_s, v_s)$. Note that the projection of a given mirror mode with j-index ν onto segments modes of higher j-indexes $\mu > \nu$ is necessarily null, therefore $\Pi_{s,\mu,\nu < \mu} = 0$ (indeed, one can for example decompose a tip-tilt of the secondary mirror, at the level of a single segment, as a piston + a tip-tilt, but there cannot be higher order aberrations in this decomposition).

We are now resolving this integral, using the Cartesian representation of the Zernike polynomials, as given in Eq. (94). First, we displace the coordinate system to the center of the segment, to simplify the limits of the integrals, with the change of variables $\mathbf{r} = (x, y) = \mathbf{u} - \mathbf{u}_s$, so $d^2r = d^2u$, $\mathbf{u} = \mathbf{r} + \mathbf{u}_s$, and we get

$$\Pi_{s,\mu,\nu \geq \mu} = \frac{z_\nu}{\pi a^2} \int_{-a}^{+a} dx \int_{-\sqrt{a^2-x^2}}^{+\sqrt{a^2-x^2}} dy Z_\nu\left(\frac{\mathbf{u} + \mathbf{r}_s}{R}\right) Z_\mu\left(\frac{\mathbf{u}}{a}\right) \quad (151)$$

Replacing the Zernike polynomials with their expression in Cartesian coordinates, we find

$$\begin{aligned} \Pi_{s,\mu,\nu \geq \mu} &= \frac{z_\nu}{\pi a^2} \frac{\sqrt{[(m_\mu \neq 0) + 1][(m_\nu \neq 0) + 1](n_\mu + 1)(n_\nu + 1)}}{R^{n_\nu} a^{n_\mu}} \times \\ &\sum_{s_\mu=0}^{\frac{n_\mu - m_\mu}{2}} \sum_{s_\nu=0}^{\frac{n_\nu - m_\nu}{2}} \sum_{l_\mu=0}^{\frac{n_\mu - m_\mu}{2} - s_\mu} \sum_{l_\nu=0}^{\frac{n_\nu - m_\nu}{2} - s_\nu} \sum_{k_\mu=0}^{q_{j\mu,m_\mu}} \sum_{k_\nu=0}^{q_{j\nu,m_\nu}} (-1)^{s_\mu + s_\nu + k_\mu + k_\nu} \binom{m_\mu}{2k_\mu + p_{j\mu,m_\mu}} \binom{m_\nu}{2k_\nu + p_{j\nu,m_\nu}} \times \\ &\binom{\frac{n_\mu - m_\mu}{2} - s_\mu}{l_\mu} \binom{\frac{n_\nu - m_\nu}{2} - s_\nu}{l_\nu} \frac{(n_\mu - s_\mu)! (n_\nu - s_\nu)! R^{2s_\nu} a^{2s_\mu}}{s_\mu! s_\nu! \left(\frac{n_\mu + m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu + m_\nu}{2} - s_\nu\right)! \left(\frac{n_\mu - m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu - m_\nu}{2} - s_\nu\right)!} \times \\ &\int_{-a}^{+a} dx \int_{-\sqrt{a^2-x^2}}^{+\sqrt{a^2-x^2}} dy (x + u_s)^{n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j\nu,m_\nu}} (y + v_s)^{2(l_\nu + k_\nu) + p_{j\nu,m_\nu}} \times \\ &x^{n_\mu - 2(s_\mu + l_\mu + k_\mu) - p_{j\mu,m_\mu}} y^{2(l_\mu + k_\mu) + p_{j\mu,m_\mu}} \quad (152) \end{aligned}$$

using the binomial formula, we can separate the x and y from the segment center coordinates,

$$\begin{aligned} (x + u_s)^{n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j\nu,m_\nu}} (y + v_s)^{2(l_\nu + k_\nu) + p_{j\nu,m_\nu}} &= \\ \sum_{\alpha=0}^{n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j\nu,m_\nu}} \binom{n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j\nu,m_\nu}}{\alpha} x^\alpha u_s^{n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j\nu,m_\nu} - \alpha} \times \\ \sum_{\beta=0}^{2(l_\nu + k_\nu) + p_{j\nu,m_\nu}} \binom{2(l_\nu + k_\nu) + p_{j\nu,m_\nu}}{\beta} y^\beta v_s^{2(l_\nu + k_\nu) + p_{j\nu,m_\nu} - \beta} \quad (153) \end{aligned}$$

so

$$\begin{aligned}
\Pi_{s,\mu,\nu \geq \mu} &= \frac{z_\nu}{\pi a^2} \frac{\sqrt{[(m_\mu \neq 0) + 1][(m_\nu \neq 0) + 1](n_\mu + 1)(n_\nu + 1)}}{R^{n_\nu} a^{n_\mu}} \times \\
&\sum_{s_\mu=0}^{\frac{n_\mu - m_\mu}{2}} \sum_{s_\nu=0}^{\frac{n_\nu - m_\nu}{2}} \sum_{l_\mu=0}^{\frac{n_\mu - m_\mu}{2} - s_\mu} \sum_{l_\nu=0}^{\frac{n_\nu - m_\nu}{2} - s_\nu} \sum_{k_\mu=0}^{q_{j_\mu, m_\mu}} \sum_{k_\nu=0}^{q_{j_\nu, m_\nu}} (-1)^{s_\mu + s_\nu + k_\mu + k_\nu} \binom{m_\mu}{2k_\mu + p_{j_\mu, m_\mu}} \binom{m_\nu}{2k_\nu + p_{j_\nu, m_\nu}} \times \\
&\left(\binom{\frac{n_\mu - m_\mu}{2} - s_\mu}{l_\mu} \right) \left(\binom{\frac{n_\nu - m_\nu}{2} - s_\nu}{l_\nu} \right) \frac{(n_\mu - s_\mu)! (n_\nu - s_\nu)! R^{2s_\nu} a^{2s_\mu}}{s_\mu! s_\nu! \left(\frac{n_\mu + m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu + m_\nu}{2} - s_\nu\right)! \left(\frac{n_\mu - m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu - m_\nu}{2} - s_\nu\right)!} \times \\
&\sum_{\alpha=0}^{n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j_\nu, m_\nu}} \sum_{\beta=0}^{2(l_\nu + k_\nu) + p_{j_\nu, m_\nu}} \binom{n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j_\nu, m_\nu}}{\alpha} \binom{2(l_\nu + k_\nu) + p_{j_\nu, m_\nu}}{\beta} \times \\
u_s & \int_{-a}^{+a} dx \int_{-\sqrt{a^2 - x^2}}^{+\sqrt{a^2 - x^2}} dy x^{n_\mu - 2(s_\mu + l_\mu + k_\mu) - p_{j_\mu, m_\mu} + \alpha} y^{2(l_\mu + k_\mu) + p_{j_\mu, m_\mu} + \beta}
\end{aligned} \tag{154}$$

and we now solve the integral over x and y . To keep the notation simple, let us define the indexes

$$\omega_\mu \equiv n_\mu - 2(s_\mu + l_\mu + k_\mu) - p_{j_\mu, m_\mu} \tag{155}$$

$$\omega_\nu \equiv n_\nu - 2(s_\nu + l_\nu + k_\nu) - p_{j_\nu, m_\nu} \tag{156}$$

$$\varphi_\mu \equiv 2(l_\mu + k_\mu) + p_{j_\mu, m_\mu} \tag{157}$$

$$\varphi_\nu \equiv 2(l_\nu + k_\nu) + p_{j_\nu, m_\nu} \tag{158}$$

so the integral becomes

$$\begin{aligned}
\int_{-a}^{+a} dx \int_{-\sqrt{a^2 - x^2}}^{+\sqrt{a^2 - x^2}} dy x^{\omega_\mu + \alpha} y^{\varphi_\mu + \beta} &= \frac{1}{\varphi_\mu + \beta + 1} \int_{-a}^{+a} dx x^{\omega_\mu + \alpha} \left[\left(\sqrt{a^2 - x^2} \right)^{\varphi_\mu + \beta + 1} - \left(-\sqrt{a^2 - x^2} \right)^{\varphi_\mu + \beta + 1} \right] \\
&= \frac{1 - (-1)^{\varphi_\mu + \beta + 1}}{\varphi_\mu + \beta + 1} \int_{-a}^{+a} dx x^{\omega_\mu + \alpha} (a^2 - x^2)^{\frac{\varphi_\mu + \beta + 1}{2}} = \dots \\
x = at, dx = a dt, (a^2 - x^2)^{\frac{\varphi_\mu + \beta + 1}{2}} &= a^{\varphi_\mu + \beta + 1} (1 - t^2)^{\frac{\varphi_\mu + \beta + 1}{2}}, x^{\omega_\mu + \alpha} = a^{\omega_\mu + \alpha} t^{\omega_\mu + \alpha} \\
\dots &= \frac{[1 - (-1)^{\varphi_\mu + \beta + 1}] a^{\omega_\mu + \alpha + \varphi_\mu + \beta + 2}}{\varphi_\mu + \beta + 1} \int_{-1}^{+1} dt t^{\omega_\mu + \alpha} (1 + t^2)^{\frac{\varphi_\mu + \beta + 1}{2}} \\
&= \frac{[1 + (-1)^{\varphi_\mu + \beta}] a^{\omega_\mu + \alpha + \varphi_\mu + \beta + 2} [1 + (-1)^{\omega_\mu + \alpha}] \Gamma\left(\frac{\omega_\mu + \alpha + 1}{2}\right) \Gamma\left(\frac{\varphi_\mu + \beta + 3}{2}\right)}{\varphi_\mu + \beta + 1 \cdot 2 \Gamma\left(\frac{\omega_\mu + \alpha + \varphi_\mu + \beta + 4}{2}\right)} \\
&= \frac{a^{\omega_\mu + \alpha + \varphi_\mu + \beta + 2} [1 + (-1)^{\omega_\mu + \alpha}] [1 + (-1)^{\varphi_\mu + \beta}] \Gamma\left(\frac{\omega_\mu + \alpha + 1}{2}\right) \Gamma\left(\frac{\varphi_\mu + \beta + 3}{2}\right)}{2(\varphi_\mu + \beta + 1) \Gamma\left(\frac{\omega_\mu + \varphi_\mu + \alpha + \beta + 4}{2}\right)} \tag{159}
\end{aligned}$$

where the integral over t has been solved with the help of the mathematical software MAPLE. With the later result, the projection coefficient becomes

$$\begin{aligned}
\Pi_{s,\mu,\nu \geq \mu} &= \frac{z_\nu \sqrt{[(m_\mu \neq 0) + 1][(m_\nu \neq 0) + 1](n_\mu + 1)(n_\nu + 1)}}{\pi a^2 R^{n_\nu} a^{n_\mu}} \times \\
&\sum_{s_\mu=0}^{\frac{n_\mu - m_\mu}{2}} \sum_{s_\nu=0}^{\frac{n_\nu - m_\nu}{2}} \sum_{l_\mu=0}^{\frac{n_\mu - m_\mu}{2} - s_\mu} \sum_{l_\nu=0}^{\frac{n_\nu - m_\nu}{2} - s_\nu} \sum_{k_\mu=0}^{q_{j_\mu, m_\mu}} \sum_{k_\nu=0}^{q_{j_\nu, m_\nu}} (-1)^{s_\mu + s_\nu + k_\mu + k_\nu} \binom{m_\mu}{2k_\mu + p_{j_\mu, m_\mu}} \binom{m_\nu}{2k_\nu + p_{j_\nu, m_\nu}} \times \\
&\binom{\frac{n_\mu - m_\mu}{2} - s_\mu}{l_\mu} \binom{\frac{n_\nu - m_\nu}{2} - s_\nu}{l_\nu} \frac{(n_\mu - s_\mu)! (n_\nu - s_\nu)! R^{2s_\nu} a^{2s_\mu}}{s_\mu! s_\nu! \left(\frac{n_\mu + m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu + m_\nu}{2} - s_\nu\right)! \left(\frac{n_\mu - m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu - m_\nu}{2} - s_\nu\right)!} \times \\
&\sum_{\alpha=0}^{\omega_\nu} \sum_{\beta=0}^{\varphi_\nu} \binom{\omega_\nu}{\alpha} \binom{\varphi_\nu}{\beta} u_s^{\omega_\nu - \alpha} v_s^{\varphi_\nu - \beta} \times \\
&\frac{a^{\omega_\mu + \varphi_\mu + \alpha + \beta + 2} [1 + (-1)^{\omega_\mu + \alpha}] [1 + (-1)^{\varphi_\mu + \beta}] \Gamma\left(\frac{\omega_\mu + \alpha + 1}{2}\right) \Gamma\left(\frac{\varphi_\mu + \beta + 3}{2}\right)}{2(\varphi_\mu + \beta + 1) \Gamma\left(\frac{\omega_\mu + \varphi_\mu + \alpha + \beta + 4}{2}\right)} \quad (160)
\end{aligned}$$

and after factorizing some terms, we get

$$\begin{aligned}
\Pi_{s,\mu,\nu \geq \mu} &= \frac{z_\nu \sqrt{[(m_\mu \neq 0) + 1][(m_\nu \neq 0) + 1](n_\mu + 1)(n_\nu + 1)}}{2\pi R^{n_\nu}} \times \\
&\sum_{s_\mu=0}^{\frac{n_\mu - m_\mu}{2}} \sum_{s_\nu=0}^{\frac{n_\nu - m_\nu}{2}} \sum_{l_\mu=0}^{\frac{n_\mu - m_\mu}{2} - s_\mu} \sum_{l_\nu=0}^{\frac{n_\nu - m_\nu}{2} - s_\nu} \sum_{k_\mu=0}^{q_{j_\mu, m_\mu}} \sum_{k_\nu=0}^{q_{j_\nu, m_\nu}} \binom{m_\mu}{2k_\mu + p_{j_\mu, m_\mu}} \binom{m_\nu}{2k_\nu + p_{j_\nu, m_\nu}} \times \\
&\binom{\frac{n_\mu - m_\mu}{2} - s_\mu}{l_\mu} \binom{\frac{n_\nu - m_\nu}{2} - s_\nu}{l_\nu} \frac{(-1)^{s_\mu + s_\nu + k_\mu + k_\nu} (n_\mu - s_\mu)! (n_\nu - s_\nu)! R^{2s_\nu}}{s_\mu! s_\nu! \left(\frac{n_\mu + m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu + m_\nu}{2} - s_\nu\right)! \left(\frac{n_\mu - m_\mu}{2} - s_\mu\right)! \left(\frac{n_\nu - m_\nu}{2} - s_\nu\right)!} \times \\
&\sum_{\alpha=0}^{\omega_\nu} \sum_{\beta=0}^{\varphi_\nu} \binom{\omega_\nu}{\alpha} \binom{\varphi_\nu}{\beta} \frac{a^{\alpha + \beta} [1 + (-1)^{\omega_\mu + \alpha}] [1 + (-1)^{\varphi_\mu + \beta}] \Gamma\left(\frac{\omega_\mu + \alpha + 1}{2}\right) \Gamma\left(\frac{\varphi_\mu + \beta + 3}{2}\right) u_s^{\omega_\nu - \alpha} v_s^{\varphi_\nu - \beta}}{(\varphi_\mu + \beta + 1) \Gamma\left(\frac{\omega_\mu + \varphi_\mu + \alpha + \beta + 4}{2}\right)} \quad (161)
\end{aligned}$$

This expression is rather cumbersome, but we must keep in mind that it does not involve any sum over any array, so its evaluation should not require any significant computation time for a desktop computer.

D Calculation of the segmented primary mirror stationary structure function

We compute the segmented primary stationary structure function by applying Eq. (15) to Eq. (57). It comes

$$\begin{aligned} \mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu) &= \left[\sum_{i=1}^{N_S} \sum_{k,l=1}^{N_H} \gamma_{i,i;k,l} \iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\nu) \mathcal{H}_k(\mathbf{u} - \mathbf{u}_i) \mathcal{H}_l(\mathbf{u} - \mathbf{u}_i) d^2u \right. \\ &\quad \left. + \iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\nu) \mathcal{H}_k(\mathbf{u} - \mathbf{u}_i + \lambda\nu) \mathcal{H}_l(\mathbf{u} - \mathbf{u}_i + \lambda\nu) d^2u \right. \\ &\quad \left. - 2 \sum_{i,j=1}^{N_S} \sum_{k,l=1}^{N_H} \gamma_{i,j;k,l} \iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\nu) \mathcal{H}_k(\mathbf{u} - \mathbf{u}_i) \mathcal{H}_l(\mathbf{u} - \mathbf{u}_j + \lambda\nu) d^2u \right] / \iint_{\mathbb{R}^2} P(\mathbf{u}) P(\mathbf{u} + \lambda\nu) d^2u \end{aligned} \quad (162)$$

where we can see that the stationary structure function (its numerator) is basically a sum of correlation products, and using the notation $\mathcal{C}\{f(\mathbf{u}); g(\mathbf{u})\} \equiv \iint f(\mathbf{u})g(\mathbf{u} + \mathbf{v})d^2u$, we get,

$$\begin{aligned} \text{NUM}\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu)\} &= \\ &\sum_{i=1}^{N_S} \sum_{k,l=1}^{N_H} \gamma_{i,i;k,l} \left[\mathcal{C}\{\mathcal{H}_k(\mathbf{u} - \mathbf{u}_i) \mathcal{H}_l(\mathbf{u} - \mathbf{u}_i); P(\mathbf{u})\} + \mathcal{C}\{P(\mathbf{u}); \mathcal{H}_k(\mathbf{u} - \mathbf{u}_i) \mathcal{H}_l(\mathbf{u} - \mathbf{u}_i)\} \right] \\ &\quad - 2 \sum_{i,j=1}^{N_S} \sum_{k,l=1}^{N_H} \gamma_{i,j;k,l} \mathcal{C}\{\mathcal{H}_k(\mathbf{u} - \mathbf{u}_i); \mathcal{H}_l(\mathbf{u} - \mathbf{u}_j)\} \end{aligned} \quad (163)$$

and

$$\text{DEN}\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu)\} = \mathcal{C}\{P(\mathbf{u}); P(\mathbf{u})\} \quad (164)$$

In the Fourier space, these correlation products become simple products, i.e.

$$\mathcal{F}\{\mathcal{C}\{f(\mathbf{u} - \mathbf{u}_i); g(\mathbf{u} - \mathbf{u}_j)\}\} = \exp[i2\pi\mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] \tilde{f}^*(\mathbf{f}) \tilde{g}(\mathbf{f}) \quad (165)$$

so we get,

$$\begin{aligned} \mathcal{F}\{\text{NUM}\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu)\}\} &= \\ &\sum_{i=1}^{N_S} \sum_{k,l=1}^{N_H} \gamma_{i,i;k,l} \left[\exp(i2\pi\mathbf{f} \cdot \mathbf{u}_i) \mathcal{F}^*\{\mathcal{H}_k(\mathbf{u}) \mathcal{H}_l(\mathbf{u})\} \tilde{P}(\mathbf{f}) + \exp(-i2\pi\mathbf{f} \cdot \mathbf{u}_i) \mathcal{F}\{\mathcal{H}_k(\mathbf{u}) \mathcal{H}_l(\mathbf{u})\} \tilde{P}^*(\mathbf{f}) \right] \\ &\quad - 2 \sum_{i,j=1}^{N_S} \sum_{k,l=1}^{N_H} \gamma_{i,j;k,l} \exp[i2\pi\mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) \end{aligned} \quad (166)$$

and,

$$\mathcal{F}\{\text{DEN}\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu)\}\} = \tilde{P}^*(\mathbf{f}) \tilde{P}(\mathbf{f}) = |\tilde{P}(\mathbf{f})|^2 \stackrel{\text{Eq. (21)}}{=} |\tilde{\mathcal{H}}(\mathbf{f})|^2 \left| \sum_{s=1}^{N_S} \exp(-i2\pi\mathbf{f} \cdot \mathbf{u}_s) \right|^2 \quad (167)$$

In what follows, the sums are systematically reduced to the minimum number possible of sums, by extensive use of the index symmetries of $\gamma_{i,j;k,l}$. First, with $\gamma_{i,i;l,k} = \gamma_{i,i;k,l}$, it comes

$$\begin{aligned}
& \mathcal{F}\left\{\text{NUM}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu)\right\}\right\} = \\
& \sum_{i=1}^{N_S} \sum_{k=1}^{N_H} \gamma_{i,i;k,k} \left[\exp(i2\pi\mathbf{f}\cdot\mathbf{u}_i) \mathcal{F}^*\{\mathcal{H}_k^2(\mathbf{u})\} \tilde{P}(\mathbf{f}) + \exp(-i2\pi\mathbf{f}\cdot\mathbf{u}_i) \mathcal{F}\{\mathcal{H}_k^2(\mathbf{u})\} \tilde{P}^*(\mathbf{f}) \right] \\
& + 2 \sum_{i=1}^{N_S} \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \gamma_{i,i;k,l} \left[\exp(i2\pi\mathbf{f}\cdot\mathbf{u}_i) \mathcal{F}^*\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\} \tilde{P}(\mathbf{f}) + \exp(-i2\pi\mathbf{f}\cdot\mathbf{u}_i) \mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\} \tilde{P}^*(\mathbf{f}) \right] \\
& \quad - 2 \sum_{i,j=1}^{N_S} \left\{ \sum_{k=1}^{N_H} \gamma_{i,j;k,k} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 + \right. \\
& \quad \left. \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \gamma_{i,j;k,l} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) + \sum_{l=1}^{N_H-1} \sum_{k=l+1}^{N_H} \gamma_{i,j;k,l} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) \right\}
\end{aligned} \tag{168}$$

splitting the sums over segments indexes i and j ,

$$\begin{aligned}
& \mathcal{F}\left\{\text{NUM}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu)\right\}\right\} = \\
& \sum_{i=1}^{N_S} [\dots + 2\dots] - 2 \sum_{i=1}^{N_S} \left\{ \sum_{k=1}^{N_H} \gamma_{i,i;k,k} |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 + \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} [\gamma_{i,i;k,l} \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) + \gamma_{i,i;l,k} \tilde{\mathcal{H}}_l^*(\mathbf{f}) \tilde{\mathcal{H}}_k(\mathbf{f})] \right\} \\
& \quad - 2 \sum_{i=1}^{N_S-1} \sum_{j=i+1}^{N_S} \left(\sum_{k=1}^{N_H} \gamma_{i,j;k,k} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 + \right. \\
& \quad \left. \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \left\{ \gamma_{i,j;k,l} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) + \gamma_{i,j;l,k} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_l^*(\mathbf{f}) \tilde{\mathcal{H}}_k(\mathbf{f}) \right\} \right) \\
& \quad - 2 \sum_{j=1}^{N_S-1} \sum_{i=j+1}^{N_S} \left(\sum_{k=1}^{N_H} \gamma_{i,j;k,k} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 + \right. \\
& \quad \left. \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \left\{ \gamma_{i,j;k,l} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) + \gamma_{i,j;l,k} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_l^*(\mathbf{f}) \tilde{\mathcal{H}}_k(\mathbf{f}) \right\} \right)
\end{aligned} \tag{169}$$

which becomes, using again the symmetry $\gamma_{i,i;l,k} = \gamma_{i,i;k,l}$,

$$\begin{aligned}
\mathcal{F}\left\{\text{NUM}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu)\right\}\right\} = & \\
& \sum_{i=1}^{N_S} [\dots + 2\dots] - 2 \sum_{i=1}^{N_S} \left\{ \sum_{k=1}^{N_H} \gamma_{i,i;k,k} |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 + \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \gamma_{i,i;k,l} [\tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) + \tilde{\mathcal{H}}_l^*(\mathbf{f}) \tilde{\mathcal{H}}_k(\mathbf{f})] \right\} \\
& - 2 \sum_{i=1}^{N_S-1} \sum_{j=i+1}^{N_S} \left(\sum_{k=1}^{N_H} \left\{ \gamma_{i,j;k,k} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 + \gamma_{i,j;k,k} \exp[-i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 \right\} \right. \\
& + \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \left\{ \gamma_{i,j;k,l} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) + \gamma_{i,j;l,k} \exp[-i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) \right. \\
& \left. \left. + \gamma_{i,j;l,k} \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_l^*(\mathbf{f}) \tilde{\mathcal{H}}_k(\mathbf{f}) + \gamma_{i,j;k,l} \exp[-i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_l^*(\mathbf{f}) \tilde{\mathcal{H}}_k(\mathbf{f}) \right\} \right) \quad (170)
\end{aligned}$$

and now using $\gamma_{j,i;k,k} = \gamma_{i,j;k,k}$ and $\gamma_{j,i;l,k} = \gamma_{i,j;k,l}$,

$$\begin{aligned}
\mathcal{F}\left\{\text{NUM}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_1}(\lambda\nu)\right\}\right\} = & \\
& \sum_{i=1}^{N_S} [\dots + 2\dots] - 2 \sum_{i=1}^{N_S} \left\{ \sum_{k=1}^{N_H} \gamma_{i,i;k,k} |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 + 2 \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \gamma_{i,i;k,l} \Re\left\{ \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) \right\} \right\} \\
& - 2 \sum_{i=1}^{N_S-1} \sum_{j=i+1}^{N_S} \left[2 \sum_{k=1}^{N_H} \gamma_{i,j;k,k} \cos[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 \right. \\
& + \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \left(\gamma_{i,j;k,l} \left\{ \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) + \exp[-i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_l^*(\mathbf{f}) \tilde{\mathcal{H}}_k(\mathbf{f}) \right\} \right. \\
& \left. \left. + \gamma_{i,j;l,k} \left\{ \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_l^*(\mathbf{f}) \tilde{\mathcal{H}}_k(\mathbf{f}) + \exp[-i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) \right\} \right) \right] = \dots \quad (171)
\end{aligned}$$

where \Re is the real part operator. Regrouping terms with common indexes i, j, k, l , and **assuming that the pupil transmission is symmetric**, so $P(-\mathbf{u}) = P(\mathbf{u}) \Rightarrow \tilde{P}^* = \tilde{P}$, we get

$$\begin{aligned}
\dots = & 2 \sum_{i=1}^{N_S} \sum_{k=1}^{N_H} \gamma_{i,i;k,k} \left[\mathcal{F}\left\{\mathcal{H}_k^2(\mathbf{u})\right\} \tilde{P}(\mathbf{f}) \cos(2\pi\mathbf{f}\cdot\mathbf{u}_i) - |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 \right] \\
& + 4 \sum_{i=1}^{N_S} \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \gamma_{i,i;k,l} \left[\tilde{P}(\mathbf{f}) \Re\left\{ \exp(i2\pi\mathbf{f}\cdot\mathbf{u}_i) \mathcal{F}^*\left\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\right\} \right\} - \Re\left\{ \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) \right\} \right] \\
& - 4 \sum_{i=1}^{N_S-1} \sum_{j=i+1}^{N_S} \sum_{k=1}^{N_H} \gamma_{i,j;k,k} \cos[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] |\tilde{\mathcal{H}}_k(\mathbf{f})|^2 \quad (172) \\
& - 4 \sum_{i=1}^{N_S-1} \sum_{j=i+1}^{N_S} \sum_{k=1}^{N_H-1} \sum_{l=k+1}^{N_H} \left\{ \gamma_{i,j;k,l} \Re\left\{ \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) \right\} \right. \\
& \left. + \gamma_{i,j;l,k} \Re\left\{ \exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k(\mathbf{f}) \tilde{\mathcal{H}}_l^*(\mathbf{f}) \right\} \right\}
\end{aligned}$$

A structure function is a real, positive and even quantity, therefore its Fourier transform must be real and even as well (but not necessarily positive). As a consequence, it should be possible to break down the complex exponentials and real part operators into real quantities only. Noting that \mathcal{H} modes are either even or odd, we know that their Fourier transform will be either exclusively real or imaginary. We define here the **value** of a complex function G as

$$\text{val}\{G(\mathbf{f})\} \equiv \Re\{G(\mathbf{f})\} + \Im\{G(\mathbf{f})\} = \begin{cases} \Re\{G(\mathbf{f})\} & G \text{ is purely real} \\ \Im\{G(\mathbf{f})\} & G \text{ is purely imaginary} \end{cases} \quad (173)$$

where \Re and \Im are the usual real part and imaginary part operators. Let us now explore the parity of the diverse complex quantities found in Eq. (172). In the second line of this equation, we have $\Re\{\exp(i2\pi\mathbf{f}\cdot\mathbf{u}_i)\mathcal{F}^*\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\}$, and depending on the parity of the hexagon modes (E is for even, O is for odd),

m_k	m_l	$m_k + m_l$	\mathcal{H}_k	\mathcal{H}_l	$\mathcal{H}_k\mathcal{H}_l$	$\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}$	$\mathcal{F}^*\{\mathcal{H}_k\mathcal{H}_l\}$	\dots
O	O	E	O	O	E	$\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$	$\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$	
O	E	O	O	E	O	$i \text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$	$-i \text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$	
E	O	O	E	O	O	$i \text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$	$-i \text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$	
E	E	E	E	E	E	$\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$	$\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$	
\dots	$\cos(2\pi\mathbf{f}\cdot\mathbf{u}_i)\mathcal{F}^*\{\mathcal{H}_k\mathcal{H}_l\}$				$i \sin(2\pi\mathbf{f}\cdot\mathbf{u}_i)\mathcal{F}^*\{\mathcal{H}_k\mathcal{H}_l\}$			$\Re\{\exp(\dots)\mathcal{F}^*\{\dots\}\}$
	$\cos(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$				$i \sin(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$			$\cos(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$
	$-i \cos(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$				$\sin(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$			$\sin(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$
	$-i \cos(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$				$\sin(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$			$\sin(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$
	$\cos(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$				$i \sin(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$			$\cos(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k\mathcal{H}_l\}\}$

i.e.

$$\Re\{\exp(i2\pi\mathbf{f}\cdot\mathbf{u}_i)\mathcal{F}^*\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\} = \begin{cases} \cos(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\} & m_k + m_l \text{ even} \\ \sin(2\pi\mathbf{f}\cdot\mathbf{u}_i)\text{val}\{\mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\} & m_k + m_l \text{ odd} \end{cases} \quad (174)$$

in the second line of Eq. (172), there is also the term $\Re\{\tilde{\mathcal{H}}_k^*(\mathbf{f})\tilde{\mathcal{H}}_l(\mathbf{f})\}$, and this one becomes

m_k	m_l	$m_k + m_l$	\mathcal{H}_k	\mathcal{H}_l	$\tilde{\mathcal{H}}_k$	$\tilde{\mathcal{H}}_l$	$\tilde{\mathcal{H}}_k^*\tilde{\mathcal{H}}_l$	$\Re\{\tilde{\mathcal{H}}_k^*\tilde{\mathcal{H}}_l\}$
O	O	E	O	O	$i \text{val}\{\tilde{\mathcal{H}}_k\}$	$i \text{val}\{\tilde{\mathcal{H}}_l\}$	$\text{val}\{\tilde{\mathcal{H}}_k\}\text{val}\{\tilde{\mathcal{H}}_l\}$	$\text{val}\{\tilde{\mathcal{H}}_k\}\text{val}\{\tilde{\mathcal{H}}_l\}$
O	E	O	O	E	$i \text{val}\{\tilde{\mathcal{H}}_k\}$	$\text{val}\{\tilde{\mathcal{H}}_l\}$	$-i \text{val}\{\tilde{\mathcal{H}}_k\}\text{val}\{\tilde{\mathcal{H}}_l\}$	0
E	O	O	E	O	$\text{val}\{\tilde{\mathcal{H}}_k\}$	$i \text{val}\{\tilde{\mathcal{H}}_l\}$	$i \text{val}\{\tilde{\mathcal{H}}_k\}\text{val}\{\tilde{\mathcal{H}}_l\}$	0
E	E	E	E	E	$\text{val}\{\tilde{\mathcal{H}}_k\}$	$\text{val}\{\tilde{\mathcal{H}}_l\}$	$\text{val}\{\tilde{\mathcal{H}}_k\}\text{val}\{\tilde{\mathcal{H}}_l\}$	$\text{val}\{\tilde{\mathcal{H}}_k\}\text{val}\{\tilde{\mathcal{H}}_l\}$

i.e.

$$\Re\{\tilde{\mathcal{H}}_k^*(\mathbf{f})\tilde{\mathcal{H}}_l(\mathbf{f})\} = \begin{cases} \text{val}\{\tilde{\mathcal{H}}_k(\mathbf{f})\}\text{val}\{\tilde{\mathcal{H}}_l(\mathbf{f})\} & m_k + m_l \text{ even} \\ 0 & m_k + m_l \text{ odd} \end{cases} \quad (175)$$

This result can be merged with the result in Eq. (174), so the second term of Eq. (172) becomes

$$\begin{aligned} \tilde{P}(\mathbf{f}) \Re\{\exp(i2\pi\mathbf{f}\cdot\mathbf{u}_i) \mathcal{F}^*\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\} + \Re\{\tilde{\mathcal{H}}_k^*(\mathbf{f})\tilde{\mathcal{H}}_l(\mathbf{f})\} = \\ \begin{cases} \tilde{P}(\mathbf{f}) \cos(2\pi\mathbf{f}\cdot\mathbf{u}_i) \text{val}\{\mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\} - \text{val}\{\tilde{\mathcal{H}}_k(\mathbf{f})\}\text{val}\{\tilde{\mathcal{H}}_l(\mathbf{f})\} & m_k + m_l \text{ even} \\ \tilde{P}(\mathbf{f}) \sin(2\pi\mathbf{f}\cdot\mathbf{u}_i) \text{val}\{\mathcal{F}\{\mathcal{H}_k(\mathbf{u})\mathcal{H}_l(\mathbf{u})\}\} & m_k + m_l \text{ odd} \end{cases} \end{aligned} \quad (176)$$

we continue with the complex terms in the last line of Eq. (172), and let us write $v_k = \text{val}\{\tilde{\mathcal{H}}_k\}$, $v_l = \text{val}\{\tilde{\mathcal{H}}_l\}$, and c and s for cos and sin,

m_k	m_l	$m_k + m_l$	$\tilde{\mathcal{H}}_k$	$\tilde{\mathcal{H}}_k^*$	$\tilde{\mathcal{H}}_l$	$\tilde{\mathcal{H}}_l^*$	$\tilde{\mathcal{H}}_k^*\tilde{\mathcal{H}}_l$	$\tilde{\mathcal{H}}_k\tilde{\mathcal{H}}_l^*$	\dots
O	O	E	iv_k	$-iv_k$	iv_l	$-iv_l$	$v_k v_l$	$v_k v_l$	
O	E	O	iv_k	$-iv_k$	v_l	v_l	$-iv_k v_l$	$iv_k v_l$	
E	O	O	v_k	v_k	iv_l	$-iv_l$	$iv_k v_l$	$-iv_k v_l$	
E	E	E	v_k	v_k	v_l	v_l	$v_k v_l$	$v_k v_l$	
$\exp(\dots)\tilde{\mathcal{H}}_k^*\tilde{\mathcal{H}}_l$			$\exp(\dots)\tilde{\mathcal{H}}_k\tilde{\mathcal{H}}_l^*$			$\Re\{\exp(\dots)\tilde{\mathcal{H}}_k^*\tilde{\mathcal{H}}_l\}$		$\Re\{\exp(\dots)\tilde{\mathcal{H}}_k\tilde{\mathcal{H}}_l^*\}$	
$cv_k v_l + isv_k v_l$			$cv_k v_l + isv_k v_l$			$cv_k v_l$		$cv_k v_l$	
$-icv_k v_l + sv_k v_l$			$icv_k v_l - sv_k v_l$			$sv_k v_l$		$-sv_k v_l$	
$icv_k v_l - sv_k v_l$			$-icv_k v_l + sv_k v_l$			$-sv_k v_l$		$sv_k v_l$	
$cv_k v_l + isv_k v_l$			$cv_k v_l + isv_k v_l$			$cv_k v_l$		$cv_k v_l$	

so

$$\begin{aligned} \Re\{\exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f})\tilde{\mathcal{H}}_l(\mathbf{f})\} = \\ \text{val}\{\tilde{\mathcal{H}}_k(\mathbf{f})\}\text{val}\{\tilde{\mathcal{H}}_l(\mathbf{f})\} \begin{cases} \cos[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k + m_l \text{ even} \\ \sin[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ odd}, m_l \text{ even} \\ -\sin[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ even}, m_l \text{ odd} \end{cases} \end{aligned} \quad (177)$$

and

$$\begin{aligned} \Re\{\exp[i2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k(\mathbf{f})\tilde{\mathcal{H}}_l^*(\mathbf{f})\} = \\ \text{val}\{\tilde{\mathcal{H}}_k(\mathbf{f})\}\text{val}\{\tilde{\mathcal{H}}_l(\mathbf{f})\} \begin{cases} \cos[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k + m_l \text{ even} \\ -\sin[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ odd}, m_l \text{ even} \\ \sin[2\pi\mathbf{f}\cdot(\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ even}, m_l \text{ odd} \end{cases} \end{aligned} \quad (178)$$

again, merging this two results, last term of Eq. (172) becomes,

$$\begin{aligned}
& \gamma_{i,j;k,l} \Re \left\{ \exp [i2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k^*(\mathbf{f}) \tilde{\mathcal{H}}_l(\mathbf{f}) \right\} + \gamma_{i,j;l,k} \Re \left\{ \exp [i2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] \tilde{\mathcal{H}}_k(\mathbf{f}) \tilde{\mathcal{H}}_l^*(\mathbf{f}) \right\} = \\
& \text{val} \left\{ \tilde{\mathcal{H}}_k(\mathbf{f}) \right\} \text{val} \left\{ \tilde{\mathcal{H}}_l(\mathbf{f}) \right\} \begin{cases} \gamma_{i,j;k,l} \cos [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] + \gamma_{i,j;l,k} \cos [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] & m_k + m_l \text{ even} \\ \gamma_{i,j;k,l} \sin [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] - \gamma_{i,j;l,k} \sin [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ odd, } m_l \text{ even} \\ -\gamma_{i,j;k,l} \sin [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] + \gamma_{i,j;l,k} \sin [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ even, } m_l \text{ odd} \end{cases} \\
& = \text{val} \left\{ \tilde{\mathcal{H}}_k(\mathbf{f}) \right\} \text{val} \left\{ \tilde{\mathcal{H}}_l(\mathbf{f}) \right\} \begin{cases} (\gamma_{i,j;k,l} + \gamma_{i,j;l,k}) \cos [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] & m_k + m_l \text{ even} \\ (\gamma_{i,j;k,l} - \gamma_{i,j;l,k}) \sin [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ odd, } m_l \text{ even} \\ (-\gamma_{i,j;k,l} + \gamma_{i,j;l,k}) \sin [2\pi \mathbf{f} \cdot (\mathbf{u}_i - \mathbf{u}_j)] & m_k \text{ even, } m_l \text{ odd} \end{cases} \quad (179)
\end{aligned}$$

E Calculation of the secondary mirrors stationary structure function

The case of the secondary mirrors stationary structure function is much easier to handle than the segmented case. We start by writing the structure function - Eq. (70),

$$D_\varphi^{\text{M}_2}(\mathbf{u}, \lambda \boldsymbol{\nu}) = \sum_{k,l=1}^{N_Z} \zeta_{k,l} [Z_k(\mathbf{u})Z_l(\mathbf{u}) - 2Z_k(\mathbf{u})Z_l(\mathbf{u} + \lambda \boldsymbol{\nu}) + Z_k(\mathbf{u} + \lambda \boldsymbol{\nu})Z_l(\mathbf{u} + \lambda \boldsymbol{\nu})] \quad (180)$$

The numerator of the stationary structure function ${}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_2}$, using the covariance notation introduced in Eq. (165), can be written

$$\text{NUM} \left\{ {}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_2}(\lambda \boldsymbol{\nu}) \right\} = \sum_{k,l=1}^{N_Z} \zeta_{k,l} \left[\mathcal{C} \{ Z_k(\mathbf{u})Z_l(\mathbf{u}); P(\mathbf{u}) \} - 2\mathcal{C} \{ Z_k(\mathbf{u}); Z_l(\mathbf{u}) \} + \mathcal{C} \{ P(\mathbf{u}); Z_k(\mathbf{u})Z_l(\mathbf{u}) \} \right] \quad (181)$$

It is important to note that in the secondary mirrors case, the quantity P is taken as the circular area that circumscribe the segmented pupil, so P is a disk whose radius is the radius of the secondary mirrors projected in the segmented pupil plane.

Applying Eq. (165), we get, for the Fourier transform of the numerator,

$$\mathcal{F} \left\{ \text{NUM} \left\{ {}_{\mathbb{R}}\overline{D}_{\delta\varphi}^{\text{M}_2}(\lambda \boldsymbol{\nu}) \right\} \right\} = \sum_{k,l=1}^{N_Z} \zeta_{k,l} \left[\mathcal{F} \{ Z_k(\mathbf{u})Z_l(\mathbf{u}) \}^* \tilde{P}(\mathbf{f}) - 2\tilde{Z}_k(\mathbf{f})\tilde{Z}_l(\mathbf{f}) + \tilde{P}^*(\mathbf{f})\mathcal{F} \{ Z_k(\mathbf{u})Z_l(\mathbf{u}) \} \right] \quad (182)$$

As we did for the segmented case, let us minimize the number of sums, using $\zeta_{k,l}$ symmetry, and

assuming that the pupil transmission is symmetric,

$$\begin{aligned}
\mathcal{F}\left\{\text{NUM}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{\text{M}_2}(\lambda\nu)\right\}\right\} &= \sum_{k=1}^{N_Z} \zeta_{k,k} \left[\mathcal{F}\{Z_k^2(\mathbf{u})\}^* \tilde{P}(\mathbf{f}) - 2|\tilde{Z}_k(\mathbf{f})|^2 + \tilde{P}^*(\mathbf{f})\mathcal{F}\{Z_k^2(\mathbf{u})\} \right] \\
&+ \sum_{k=1}^{N_Z-1} \sum_{l=k+1}^{N_Z} \zeta_{k,l} \left[\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\}^* \tilde{P}(\mathbf{f}) + \mathcal{F}\{Z_l(\mathbf{u})Z_k(\mathbf{u})\}^* \tilde{P}(\mathbf{f}) - 2\tilde{Z}_k^*(\mathbf{f})\tilde{Z}_l(\mathbf{f}) - 2\tilde{Z}_l^*(\mathbf{f})\tilde{Z}_k(\mathbf{f}) \right. \\
&\left. + \tilde{P}^*(\mathbf{f})\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\} + \tilde{P}^*(\mathbf{f})\mathcal{F}\{Z_l(\mathbf{u})Z_k(\mathbf{u})\} \right] \\
&= 2 \sum_{k=1}^{N_Z} \zeta_{k,k} \left[\Re\{\mathcal{F}\{Z_k^2(\mathbf{u})\}\} \tilde{P}(\mathbf{f}) - |\tilde{Z}_k(\mathbf{f})|^2 \right] \\
&+ 2 \sum_{k=1}^{N_Z-1} \sum_{l=k+1}^{N_Z} \zeta_{k,l} \left\{ \tilde{P}(\mathbf{f}) \left[\Re\{\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\}\} + \Re\{\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\}\} \right] - 2 \Re\{\tilde{Z}_k^*(\mathbf{f})\tilde{Z}_l(\mathbf{f})\} \right\} \\
&= 2 \sum_{k=1}^{N_Z} \zeta_{k,k} \left[\Re\{\mathcal{F}\{Z_k^2(\mathbf{u})\}\} \tilde{P}(\mathbf{f}) - |\tilde{Z}_k(\mathbf{f})|^2 \right] \\
&+ 4 \sum_{k=1}^{N_Z-1} \sum_{l=k+1}^{N_Z} \zeta_{k,l} \left[\tilde{P}(\mathbf{f}) \Re\{\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\}\} - \Re\{\tilde{Z}_k^*(\mathbf{f})\tilde{Z}_l(\mathbf{f})\} \right]
\end{aligned} \tag{183}$$

the complex terms can be broken down,

m_k	m_l	m_k+m_l	Z_k	Z_l	$Z_k Z_l$	$\mathcal{F}\{Z_k Z_l\}$	$\Re\{\mathcal{F}\{Z_k Z_l\}\}$
O	O	E	O	O	E	$\text{val}\{\mathcal{F}\{Z_k Z_l\}\}$	$\text{val}\{\mathcal{F}\{Z_k Z_l\}\}$
O	E	O	O	E	O	$i \text{val}\{\mathcal{F}\{Z_k Z_l\}\}$	0
E	O	O	E	O	O	$i \text{val}\{\mathcal{F}\{Z_k Z_l\}\}$	0
E	E	E	E	E	E	$\text{val}\{\mathcal{F}\{Z_k Z_l\}\}$	$\text{val}\{\mathcal{F}\{Z_k Z_l\}\}$

and

m_k	m_l	m_k+m_l	Z_k	Z_l	\tilde{Z}_k^*	\tilde{Z}_l	$\tilde{Z}_k^* \tilde{Z}_l$	$\Re\{\tilde{Z}_k^* \tilde{Z}_l\}$
O	O	E	O	O	$-i \text{val}\{\tilde{Z}_k\}$	$i \text{val}\{\tilde{Z}_l\}$	$\text{val}\{\tilde{Z}_k\} \text{val}\{\tilde{Z}_l\}$	$\text{val}\{\tilde{Z}_k\} \text{val}\{\tilde{Z}_l\}$
O	E	O	O	E	$-i \text{val}\{\tilde{Z}_k\}$	$\text{val}\{\tilde{Z}_l\}$	$-i \text{val}\{\tilde{Z}_k\} \text{val}\{\tilde{Z}_l\}$	0
E	O	O	E	O	$\text{val}\{\tilde{Z}_k\}$	$i \text{val}\{\tilde{Z}_l\}$	$i \text{val}\{\tilde{Z}_k\} \text{val}\{\tilde{Z}_l\}$	0
E	E	E	E	E	$\text{val}\{\tilde{Z}_k\}$	$\text{val}\{\tilde{Z}_l\}$	$\text{val}\{\tilde{Z}_k\} \text{val}\{\tilde{Z}_l\}$	$\text{val}\{\tilde{Z}_k\} \text{val}\{\tilde{Z}_l\}$

i.e.

$$\begin{aligned}
\tilde{P}(\mathbf{f}) \Re\{\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\}\} - \Re\{\tilde{Z}_k^*(\mathbf{f})\tilde{Z}_l(\mathbf{f})\} &= \\
&\begin{cases} \tilde{P}(\mathbf{f}) \text{val}\{\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\}\} - \text{val}\{\tilde{Z}_k(\mathbf{f})\} \text{val}\{\tilde{Z}_l(\mathbf{f})\} & m_k+m_l \text{ is even} \\ 0 & m_k+m_l \text{ is odd} \end{cases} \tag{184}
\end{aligned}$$

then

$$\begin{aligned} \mathcal{F}\left\{\text{NUM}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{M_2}(\lambda\nu)\right\}\right\} &= 2\sum_{k=1}^{N_Z}\zeta_{k,k}\left[\mathcal{F}\{Z_k^2(\mathbf{u})\}\tilde{P}(\mathbf{f})-|\tilde{Z}_k(\mathbf{f})|^2\right] \\ &+ 4\sum_{k=1}^{N_Z-1}\sum_{l=k+1}^{N_Z}\zeta_{k,l}\begin{cases} \tilde{P}(\mathbf{f})\text{val}\{\mathcal{F}\{Z_k(\mathbf{u})Z_l(\mathbf{u})\}\}-\text{val}\{\tilde{Z}_k(\mathbf{f})\}\text{val}\{\tilde{Z}_l(\mathbf{f})\} & m_k+m_l \text{ is even} \\ 0 & m_k+m_l \text{ is odd} \end{cases} \end{aligned} \quad (185)$$

Finally, the denominator of the secondary stationary structure function, as defined in Eq. (15), is the pupil auto-correlation,

$$\text{DEN}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{M_2}(\lambda\nu)\right\} = \iint_{\mathbb{R}^2} P(\mathbf{u})P(\mathbf{u}+\lambda\nu)d^2u \quad (186)$$

which is easy to compute for a circular pupil of diameter D_p , and we get - see for instance Goodman (1996),

$$\text{DEN}\left\{\mathbb{R}\overline{D}_{\delta\varphi}^{M_2}(\lambda\nu)\right\} = \frac{D_p^2}{2}\left[\arccos\left(\frac{\lambda\nu}{D_p}\right)-\left(\frac{\lambda\nu}{D_p}\right)\sqrt{1-\left(\frac{\lambda\nu}{D_p}\right)^2}\right] \quad (187)$$

F Projection of a product of Zernike polynomials onto the Zernike basis

We compute here the coefficients of the projection of the product of Zernike polynomials $Z_k Z_l$ onto the Zernike basis itself. The maximum number of Zernike polynomials required to represent the product $Z_k Z_l$ is set by the last j-index of the set of j-indexes associated to the radial order $n_j = n_k + n_l$ (for instance, if $k = 2$ and $l = 5$, $n_k = 1$, $n_l = 2$, so $n_j = 3$, and there are 4 polynomials with this radial order, with j-indexes [7,8,9,10], therefore $\max(j)=10$). We have,

$$\begin{aligned} p_{k,l,j} &\equiv \langle Z_k Z_l | Z_j \rangle = \frac{1}{\pi} \int_0^{2\pi} \int_0^1 Z_k(r,\theta) Z_l(r,\theta) Z_j(r,\theta) r dr d\theta \\ &= \sqrt{[1+(m_k \neq 0)][1+(m_l \neq 0)][1+(m_j \neq 0)]} \sqrt{(n_k+1)(n_l+1)(n_j+1)} \times \\ &\quad \int_0^1 R_{n_k}^{m_k}(r) R_{n_l}^{m_l}(r) R_{n_j}^{m_j}(r) r dr \times \\ &\quad \frac{1}{\pi} \int_0^{2\pi} \begin{cases} 1 & m_k = 0 \\ \cos(m_k\theta) & k \text{ even} \\ \sin(m_k\theta) & k \text{ odd} \end{cases} \begin{cases} 1 & m_l = 0 \\ \cos(m_l\theta) & l \text{ even} \\ \sin(m_l\theta) & l \text{ odd} \end{cases} \begin{cases} 1 & m_j = 0 \\ \cos(m_j\theta) & j \text{ even} \\ \sin(m_j\theta) & j \text{ odd} \end{cases} d\theta \end{aligned} \quad (188)$$

let us define

$$\sigma_{k,l,j} \equiv \sqrt{[1+(m_k \neq 0)][1+(m_l \neq 0)][1+(m_j \neq 0)](n_k+1)(n_l+1)(n_j+1)} \int_0^1 R_{n_k}^{m_k}(r) R_{n_l}^{m_l}(r) R_{n_j}^{m_j}(r) r dr \quad (189)$$

replacing the radial function by its definition - Eq. (88) - and integrating over r , we find

$$\sigma_{k,l,j} = \sqrt{[1+(m_k \neq 0)][1+(m_l \neq 0)][1+(m_j \neq 0)](n_k+1)(n_l+1)(n_j+1)} \times$$

$$\sum_{s_k=0}^{\frac{n_k-m_k}{2}} \sum_{s_l=0}^{\frac{n_l-m_l}{2}} \sum_{s_j=0}^{\frac{n_j-m_j}{2}} \frac{(-1)^{s_k+s_l+s_j} (n_k-s_k)! (n_l-s_l)! (n_j-s_j)! [n_k+n_l+n_j-2(s_k+s_l+s_j)+2]^{-1}}{s_k! s_l! s_j! \left(\frac{n_k-m_k}{2}-s_k\right)! \left(\frac{n_l-m_l}{2}-s_l\right)! \left(\frac{n_j-m_j}{2}-s_j\right)!}$$
(190)

We give now the integral over the angular functions,

$$\alpha_{k,l,j} \equiv \frac{1}{\pi} \int_0^{2\pi} \left\{ \begin{array}{cc} 1 & m_k = 0 \\ \cos(m_k \theta) & k \text{ even} \\ \sin(m_k \theta) & k \text{ odd} \end{array} \right\} \left\{ \begin{array}{cc} 1 & m_l = 0 \\ \cos(m_l \theta) & l \text{ even} \\ \sin(m_l \theta) & l \text{ odd} \end{array} \right\} \left\{ \begin{array}{cc} 1 & m_j = 0 \\ \cos(m_j \theta) & j \text{ even} \\ \sin(m_j \theta) & j \text{ odd} \end{array} \right\} d\theta \quad (191)$$

and we find,

$$\begin{aligned} \text{if } \underline{(m_k=0)+(m_l=0)+(m_j=0)} = 3, & \quad \alpha_{k,l,j} = 2 \\ \text{if } \underline{(m_k=0)+(m_l=0)+(m_j=0)} = 2, & \quad \alpha_{k,l,j} = 0 \\ \text{if } \underline{(m_k=0)+(m_l=0)+(m_j=0)} = 1, & \quad \left\{ \begin{array}{l} \text{if } (m_k \neq 0)j_k + (m_l \neq 0)j_l + (m_j \neq 0)j_j \text{ is even,} \\ \alpha_{k,l,j} = \delta_{m_k, m_l} + \delta_{m_k, m_j} + \delta_{m_l, m_j} \\ \text{otherwise } \alpha_{k,l,j} = 0 \end{array} \right. \\ \text{if } \underline{(m_k=0)+(m_l=0)+(m_j=0)} = 0, & \quad \dots \\ \text{if } (j_k, \text{even}) + (j_l, \text{even}) + (j_j, \text{even}) = 3, & \quad \left\{ \begin{array}{l} \text{if } (m_k = |m_l - m_j|) + (m_l = |m_k - m_j|) + \\ (m_j = |m_k - m_l|) = 1, \quad \alpha_{k,l,j} = 1/2 \\ \text{otherwise } \alpha_{k,l,j} = 0 \end{array} \right. \\ \text{if } (j_k, \text{even}) + (j_l, \text{even}) + (j_j, \text{even}) = 2, & \quad \alpha_{k,l,j} = 0 \\ \text{if } (j_k, \text{even}) + (j_l, \text{even}) + (j_j, \text{even}) = 1, & \quad \left\{ \begin{array}{l} \text{if } (m_k = m_l + m_j) + (m_l = m_k + m_j) + (m_j = m_k + m_l) \neq 1, \\ \alpha_{k,l,j} = 0 \\ \text{if } (m_k = m_l + m_j) + \left\{ \begin{array}{l} \text{if } m_k + m_l + m_j = 2m_i |j_i \text{ even,} \\ \alpha_{k,l,j} = -1/2 \\ \text{if } m_k + m_l + m_j \neq 2m_i |j_i \text{ even,} \\ \alpha_{k,l,j} = +1/2 \end{array} \right. \\ (m_l = m_k + m_j) + \\ (m_j = m_k + m_l) = 1, \end{array} \right. \\ \text{if } (j_k, \text{even}) + (j_l, \text{even}) + (j_j, \text{even}) = 0, & \quad \alpha_{k,l,j} = 0 \end{aligned}$$

References

- [1] J. W. Goodman, *Introduction to Fourier Optics, 2nd Ed.*, McGraw-Hill, 1996.
- [2] L. Jolissaint, J.-P. Veran, and R. Conan, “Analytical modeling of adaptive optics: foundations of the phase spatial power spectrum approach,” *Journal of the Optical Society of America A* **23**(2), pp. 382–394, 2006.
- [3] F. Roddier, *The effects of atmospheric turbulence in optical astronomy*, E. Wolf, ed., Progress in optics 19, 1981.
- [4] E. W. Weisstein, “Jensen’s Inequality,” tech. rep., From MathWorld – A Wolfram Web Resource, <http://mathworld.wolfram.com/JensensInequality.html>, 2006.
- [5] R. Noll, “Zernike polynomials and atmospheric turbulence,” *Journal of the Optical Society of America* **66**(3), 1976.
- [6] R. Upton and B. Ellerbroek, “Gram-Schmidt Orthogonalization of the Zernike Polynomials on a Hexagonal Aperture,” *Optics Letters* **29**, p. 2840, 2004.
- [7] D. J. Schroeder, *Astronomical Optics, 2nd Ed.*, Academic Press, 2000.