Master Thesis

Optimum design of freeform-enabled space optical instruments

Alvaro Menduina

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Master Thesis

Optimum design of freeform-enabled space optical instruments

by

Alvaro Menduina

Master of Science
in Aerospace Engineering

at the Delft University of Technology,

Supervisor: Dr. ir. J. M. Hans Kuiper

Abstract

Nowadays space missions rely heavily on optical payloads to carry out a wide range of tasks including earth observation, weather monitoring, astrophysics research and communications. Until recently, the design of such systems was done according to design principles such as rotational symmetry as it simplifies the theory, reduces technological risks, manufacturing costs and cuts down assembly, integration and testing efforts.

Nevertheless, the ever-growing need to develop more compact and lightweight payloads with enhanced performance often forces designers to adopt tilted off-axis configurations which break the rotational symmetry of the system. In such circumstances, rotationally symmetric optical surfaces cannot compensate the tilt-induced optical aberrations and thus no longer provide optimum performance over the complete field of view. The solution to this problem is to abandon the conventional approach and adopt a new design paradigm based on the use of *non-rotationally symmetric* surfaces, also known as *freeform optics*.

Most of the state-of-the-art methods used for the design of conventional optical payloads are not entirely suitable for freeform optics because: first of all, they were developed on the basis of rotational symmetry (which is no longer applicable) and secondly, because they do not cope well with the large amount of additional degrees of freedom that freeform optics usually entails. Therefore, this Master Thesis was devoted to the development of a novel methodology for the design and optimization of payloads based on freeform optics.

In contrast with most approaches to freeform design, we adopted a *high-dimensional* approach based on surface modelling with a substantial amount of degrees of freedom (in the order of 100 to 1000). In addition, this methodology is built upon the framework of *differential ray tracing*, a technique which provides valuable information on how small changes on the parameters of the system or in the ray data alter its optical performance. Although this technique is well known in optical design, it has hardly ever been used in realistic applications partly because of the difficulty of generalizing it to complex systems, such as those based on freeform optics. During the course of this research, we have demonstrated that differential ray tracing can indeed be generalized to freeform systems by means of *automatic differentiation* tools which can compute derivatives of computer programs up to machine precision.

Based on those findings, a toolbox for the design and optimization of freeform systems called **GDRT** was developed. **GDRT** has its own ray tracing capabilities; it can model different types of surfaces which includes *reflective* (mirrors), *refractive* (lenses), *diffractive* (gratings) in both conventional and freeform approaches; it has built-in optimization techniques which allow for optimization of many metrics of optical performance (distortion, spot size, wavefront error, keystone, magnification, spectral dispersion, etc) and it is easily customizable to cope with new demands and tasks. Consequently, it can effectively operate autonomously, with minimal communication with commercial software packages for optical design like Zemax Optics Studio.

Our investigations have revealed that it is possible to take a design based on conventional optics which has stagnated (its optical performance can no longer be improved with Zemax), construct a freeform version and optimize it with **GDRT** reaching substantial performance enhancements. This way, **GDRT** has been tested for a variety of optical systems in all regimes (reflective, refractive, diffractive) and configurations (on-axis and off-axis), for wavelengths in the visible as well as thermal infra-red, with varying degrees of complexity; always with satisfactory results.

Contents

List of Figures

LIST OF TABLES

1 INTRODUCTION

1.1. FREEFORM OPTICS

Freeform optics is undoubtedly the cornerstone of this Master Thesis. The concept itself is simple and it boils down to *designing optical systems with surfaces which are **not** rotationally symmetric*. At first glance, this may not sound like a very innovative idea; but abandoning the assumption of rotational symmetry constitutes a radical change with respect to what has been the design paradigm until quite recently. In fact, the implications of freeform optics can be rather dramatic and are hindering their application in industries such as the space industry where technological risks and disruptive innovation are treated with extreme care. In this section, we will address the potential benefits of freeform optics, give examples of current applications (both inside and outside the space industry) and analyse the research efforts that have made this technology a reality.

1.1.1. INTRODUCTION TO FREEFORM OPTICS

Historically, the design of optical systems for imaging applications has followed design principles based on rotational symmetry for several reasons. First of all, the industry capabilities for manufacturing complex surfaces used to be quite limited. Secondly, in those cases were manufacturing was actually feasible, it usually entailed a large increase in cost per surface which was hard to justify in terms of performance gains. Moreover, surfaces without rotational symmetry impose additional challenges to the assembly, integration and alignment stages of the instrument, further increasing the project costs and risks. Last but not least, the achievable optical performance of instruments developed under this *simple design* philosophy used to be more than sufficient for most applications, which meant little to no demand existed for complicated designs.

Nevertheless, the constant eagerness to extend the frontiers of scientific knowledge in fields such as astronomy, earth observation, planetary exploration and all space-related disciplines in general, generates an ever-growing need to develop lightweight solutions for optical instruments with higher overall optical performance and enhanced compactness. In fact, new generation payloads are almost always expected to substantially outperform their parents which means having to push the technological boundaries further at each iteration. As a result of this, nowadays it is often inevitable for optical systems to operate in highly folded, off-axis configurations (see Figure 1.1), breaking the symmetry of the system.

In such circumstances, where components are highly tilted and placed outside the optical axis, it is reasonable to conclude that the assumption of rotational symmetry can no longer provide optical results in terms of optical performance. The main reason for that is that off-axis systems suffer from tilt-induced optical aberrations which exhibit complicated field dependencies. Rotationally symmetric surfaces cannot easily compensate those aberrations and thus, the only solution is to completely abandon the idea of symmetry and employ *freeform* optical surfaces.

This is the essence behind the relatively novel design philosophy of freeform optics, a field which has witnessed a fast development in recent years. The key to the success of freeform optics as the new paradigm for

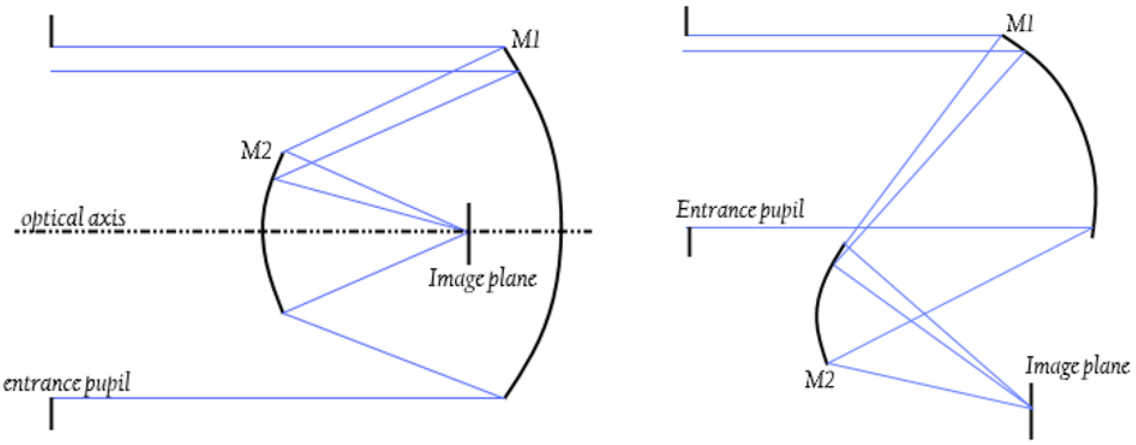


Figure 1.1: Purely *on-axis* (left) two-mirror system vs its folded *off-axis* version

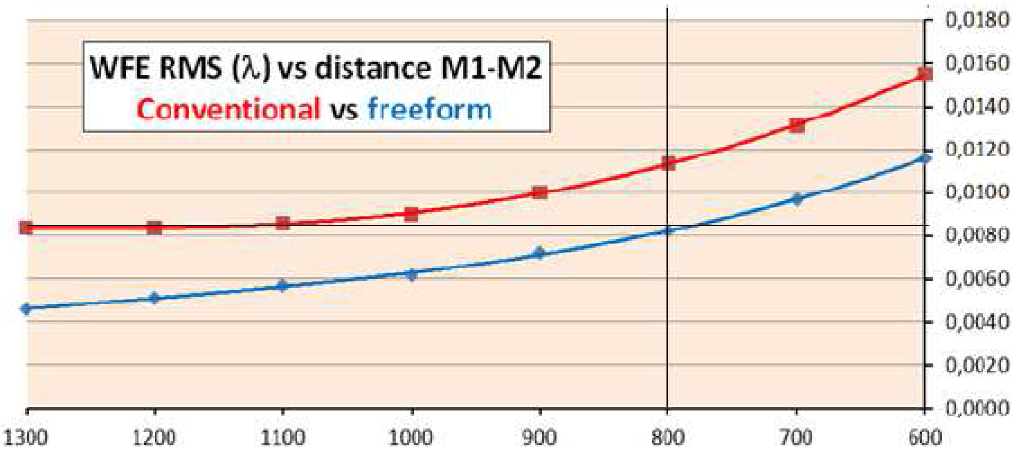


Figure 1.2: Performance vs volume for the Pleiades design, both in conventional and freeform configuration. Source: [1]

optical design is the inherent increase in available degrees of freedom which comes when rotational symmetry is discarded. With more degrees of freedom to work with, the possibilities for optimizing complex off-axis systems beyond their current capabilities increases dramatically.

But the potential of freeform optics is not limited to off-axis systems. In fact, even when the system configuration is co-axial and symmetric, the use of rotationally symmetric surfaces usually results in a *field mismatch* because the majority of image sensors have rectangular dimensions with a certain aspect ratio [8]. This is because for those systems the optical performance tends to be rotationally symmetric, i.e. the performance contour lines on the image plane are circles, and thus do not match the rectangular shape of the detector especially if the aspect ratio is high. In such systems, freeform optics offer the possibility of properly adapting the performance contour lines so that they resemble the shape of the detector.

One of the major strengths of freeform optics is that it can provide higher performance for essentially the same volume and mass, or the same performance but with a more compact an lightweight design (see Figure 1.2). In some cases, the use of freeform optics has led to designs which are up 5 times more compact than those based on traditional surfaces [9]. Other studies have also addressed the possibility of *miniaturization* of optical devices by using freeform optics, leading to smaller, thinner and lighter designs [10].

Freeform optics is quickly gaining popularity; a literature survey of the topic shows that research into freeform optics is dramatically increasing with most effort related to optical design [11]. In fact, freeform optics has been used in ground-based astronomy applications for at least a decade. Examples include NASA

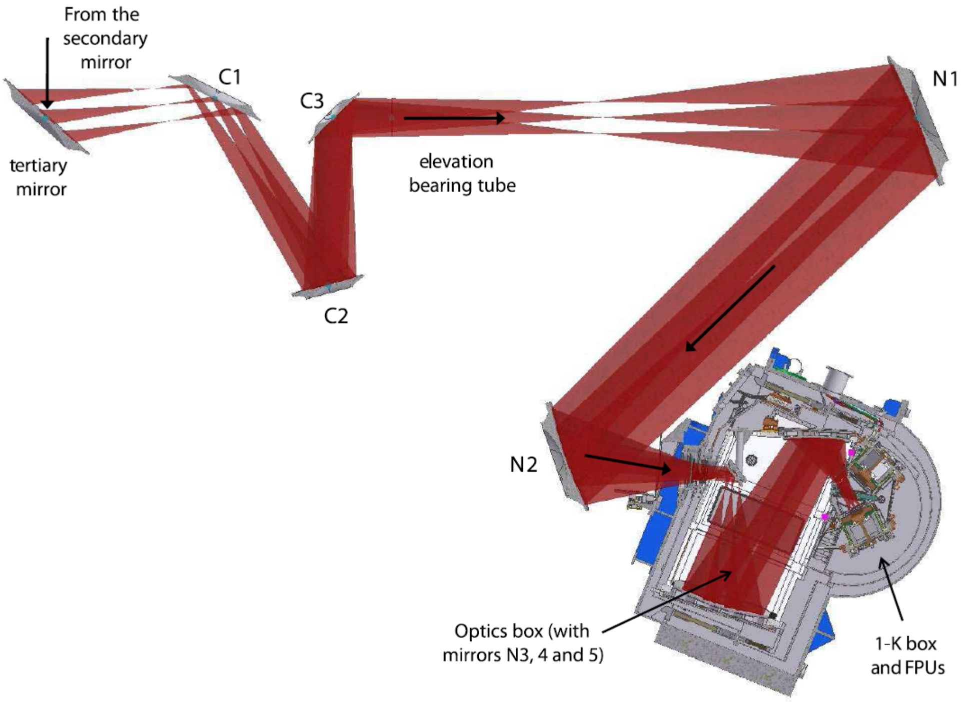


Figure 1.3: Optical layout of SCUBA-2. Source: [2]

Infrared Multi-Object Spectrometer (IRMOS) at the Kitt Peak National Observatory [12] for which the use of freeform optics helped reduce the size by an order of magnitude [13]; as well as the *Submillimetre Common-User Bolometer Array 2* (SCUBA-2) on the James Clerk Maxwell telescope whose scientific requirements and constraints on the telescope mechanical structure led to a complex optical design using freeform mirrors [14].

Due to the inherent high-risk nature of space endeavours and to the scarcity of opportunities for error correction, the space industry usually adopts a risk-adverse culture which leaves little freedom for innovation not strictly needed for mission success [15]. Nevertheless, the extension of freeform optics to space-borne systems is slowly gaining speed. Most of the relevant players of the space sector have shown their interest for freeform optics as the key technology for future generations of space instruments.

On the one hand, NASA has already acknowledged the "enormous potential of freeform surfaces for improving optical systems" [16] and is actively involved in gaining expertise in this emerging field, enabling instruments for CubeSat platforms [17]. In 2016, NASA funding was allocated under the Instrument Incubator Program for the MiniSpec proposal: *Miniaturized Imaging Spectrometer to measure vegetation structure and function* which will utilize freeform optics to enable high spectral and spatial resolution on a very small bus [18].

On the European side, ESA and the Netherlands Space Office (NSO) are cooperating to deploy a precursor of the Sentinel-5 mission: the Sentinel-5P satellite whose launch is planned for September 2017 will carry the TROPOMI payload containing a freeform primary mirror for its telescope. TROPOMI is a major re-design of OMI by TNO, but with more demanding performance requirements such as a factor of six smaller ground pixel and signal-to-noise ratios to fit an order of magnitude lower signals [19]. In addition, the German Aerospace Centre (DLR) has recently developed a new hyperspectral Earth Sensing Imaging Spectrometer (DESIS) which will be integrated into Multi-User-System for Earth Sensing (MUSES) a platform on-board the International Space Station (ISS) which can host up to 4 Earth observation instruments. DESIS, which is scheduled for launch towards the ISS this year, employs freeform surfaces to improve the performance of its Offner-type spectrometer [4]. These initiatives combined with other recent industry-university partnerships (like the creation of the Center for Freeform Optics CeFo at the University of Rochester) suggest that freeform optics will become a mature and widespread technology in future space missions.

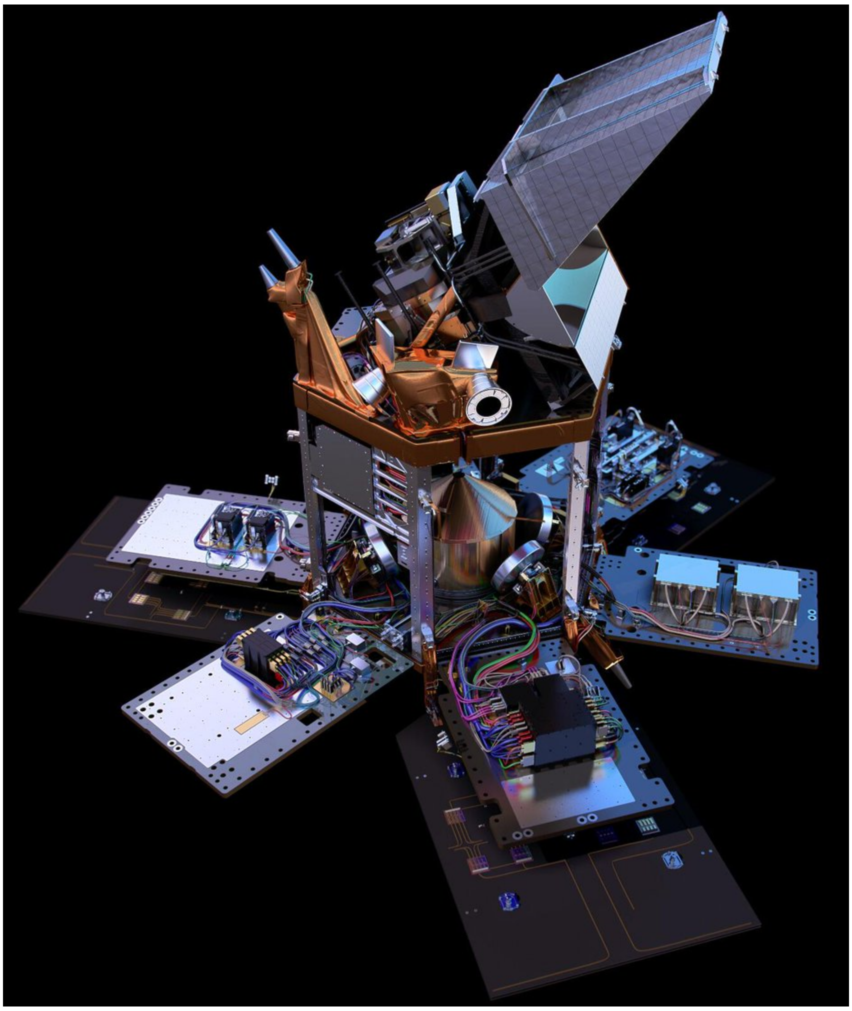


Figure 1.4: CAD render of Sentinel-5p including the TROPOMI payload. Source: [3]

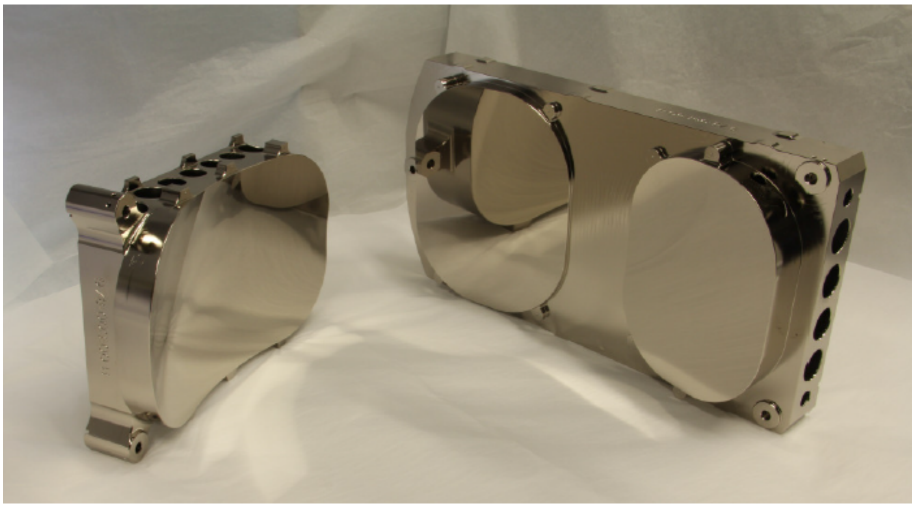


Figure 1.5: Offner-type freeform surface (*left*) and combined M1-M3 mirrors of the TMA (*right*) of the DESIS instrument. Source: [4]

1.1.2. DEVELOPMENT OF FREEFORM OPTICS

The change in design paradigm from conventional optics to the novel freeform optics approach required a series of developments in all the stages of the design process, ranging from the theoretical framework to the manufacturing of components and the integration of complete optical payloads. The first and most fundamental step was the extension of optical aberration theories (which were built upon the assumptions of rotational symmetry) to general cases of off-axis systems with non-rotationally symmetric surfaces. Extending the theoretical framework is essential as it allows designers to understand the origin of the aberrations degrading the optical performance of these systems and to develop suitable correction strategies. These recent developments started a decade ago with the work by Thompson when he extended the wave aberration theory to systems which are not rotationally symmetric but employ rotationally symmetric surfaces [20]. Later on, Fuerschbach built upon those findings and derived the formalism for the aberration theory of systems based on freeform surfaces, with emphasis on Zernike polynomials [21].

The next step required the development of tools which implement that theoretical knowledge and exploit it to design freeform systems. This includes constructing a proper framework for modelling the system and its optical surfaces and to optimize it according to a particular choice of degrees of freedom, metrics of performance and optimization algorithms. It is in this particular field where the major research effort is currently being spent; including this particular Master Thesis. The common approach in recent years has been to re-use the same techniques, although slightly adapted, already present in state-of-the-art software packages for optical design. But this may not be the optimum procedure as conventional optics and freeform optics design are substantially different in nature. Novel tools and techniques tailored to the particular needs, strengths and weaknesses of freeform optics might in the end be needed to exploit the full potential of this technology.

The final step involved innovations in the stages of the design process where actual freeform components are involved. Unfortunately, due to their inherent complexity, the beneficial effects of freeform optics usually come at the cost of introducing new challenges in the manufacturing, metrology, assembly, integration and alignment. Extensive research has been devoted to develop novels techniques for the manufacturing of freeform surfaces including diamond turning [22], high-speed micromilling [23] and ultra-precision grinding [24]; and compensation strategies to ensure proper form accuracy [25]. A thorough review of the most relevant research into manufacturing and measurement of freeform optics can be found in [11].

1.2. PROBLEM DEFINITION

These days most players in the design chain (system engineers, optical designers, AIT engineers, project managers, etc) acknowledge that freeform systems can be superior to conventional designs in terms of performance and compactness. But it is no less true that a reluctance to adopt freeform optics exists, when conventional designs can meet the requirements. This reluctance is usually fuelled by widespread beliefs such as: *freeform components are inherently more sensitive to mechanical tolerances, freeform systems are very complicated to integrate and align* and *the freeform surface departures are so small that they cannot be resolved by manufacturers* amongst others. These assumptions although true up to a certain extent, represent a short-sighted view of the whole field of freeform optics; and some are refuted in the literature [7] and in this very report.

It is true that in the past, optical designers were constrained by limitations on the manufacturing capabilities and sometimes requested components beyond what was achievable at that time. But nowadays, with the aforementioned advances in manufacturing, the tables have turned and those capabilities exceed the demands imposed by designers; especially when freeform surfaces are involved. Companies with extensive experience in optical surface manufacturing often argue that their freeform manufacturing capabilities are widely underused [1], and have to encourage designers to take advantage of them and incorporate freeform surfaces into their designs.

So the manufacturing capabilities are there; the theoretical framework to deal with freeform systems is there; but what is currently lacking is specific tools for the design and optimization of those systems. As already mentioned, current efforts to design freeform optics, what one would call the state-of-the-art approach,

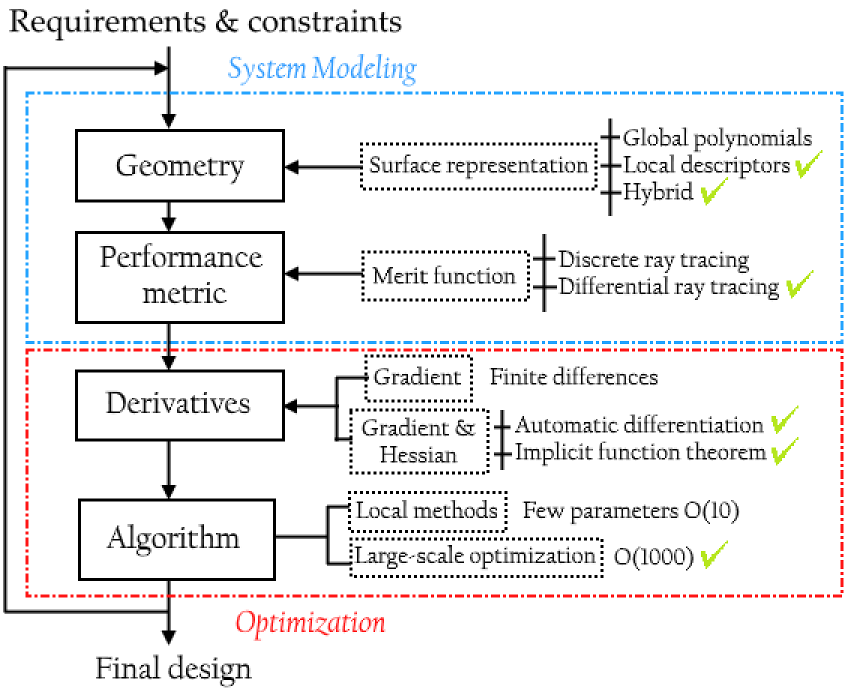


Figure 1.6: Typical workflow in optical design and optimization. Conventional tools and techniques are compared with those used in this project (ticked in green)

are based on the use of the same tools (Zemax and CODE V), techniques and design philosophies that were historically used for conventional systems. Perfectly reasonable, well-performing designs can be achieved that way and the literature is full of examples ([26], [27], [28], [29]), but in this report we argue that a specific approach tailored to the particular traits of freeform optics could be used instead, with equally satisfactory results but without the limitations imposed by the conventional tools.

To properly understand what the essence of this statement is, let us first compare the state-of-the-art freeform design methodology with the novel approach adopted for this Master Thesis.

1.2.1. STATE-OF-THE-ART VS. NOVEL APPROACH

The natural workflow of the optical design process with conventional tools like Zemax (widely used to design systems based on rotational symmetric surfaces) can be easily visualized by looking at Figure 1.6. The designer usually starts with a set of **requirements** on optical performance (target wavefront error, distortion, ground resolution...) and some **constraints** on the physical envelope of the system (maximum volume, type of surfaces, target focal length...). Based on that information, the designer sets up a concept for the physical architecture of the optical system and generates a baseline design by defining the **geometry** of the different surfaces in the software tool. Then, a merit function is constructed which encodes the **performance** of the system as a function of the parameters (degrees of freedom) of the defined geometry. The final step is an **optimization** stage which alters the system parameters until a design which complies with the requirements and constraints is reached. This is in itself an iterative process, and several runs are usually needed before a proper solution is found.

In what one might call the *state-of-the-art* in freeform optics design, the stages described above are tackled from a perspective which can be summarised into the following key points:

1. **Surface representation** models usually employ some form of series expansion of *global* polynomials such as Zernike, Chebyshev, XY polynomials, etc. This means that a change in one coefficient of the series expansion affects the complete surface. Therefore, only a small number of coefficients is needed

to represent the complete surface, keeping the amount of **degrees of freedom** quite low. Nevertheless, **local** and **hybrid** methods are starting to gain popularity.

2. **Conventional tools** for standard optical design and their associated techniques. This includes software tools like Zemax and CODE V, with **discrete ray tracing** and **finite-difference** schemes for the computation of derivatives of merit functions.

3. **Gradient-based optimization algorithms** as implemented in the aforementioned tools are commonly used in freeform optimization. Nevertheless, conventional gradient-based algorithms which are successful for rotationally symmetric systems with around a dozen degrees of freedom might not perform well when applied to large-scale optimization of freeform systems. This partially motivates the choice of **global** polynomials with few degrees of freedom, as these algorithms tend to suffer from convergence and speed issues.

In contrast, the novel approach presented in this report introduces some changes into the state-of-the-art philosophy for freeform optics design which we will justify here.

1. As far as **surface representation** is concerned, the use of **global** polynomials can sometimes lead to several issues. The particular details about this topic are given in subsection 2.7.2 but, in essence, what we propose is a *hybrid* model which combines **global** and **local** surface representation.

2. With respect to **ray tracing** and the computation of **derivatives** we argue that the technique called **differential ray tracing** combined with the use of **automatic differentiation** tools allows the computation of derivatives of interest in the optical system.

3. Those **derivatives** are computed for arbitrary order up to machine precision, without any assumption about the geometry of the system and without a severe penalty on the amount of degrees of freedom used.

4. This is made possible by a mathematical framework which takes advantage of **Fermat's path principle** and the **Implicit Function Theorem**

5. As a result of this, we are able to optimize freeform systems with a large number of degrees of freedom (100 - 1000) using second order algorithms which exploit Hessian information, leading to improvements in convergence speed.

1.3. RESEARCH QUESTION

Now that the necessary background has been presented, we can announce the research question at the core of this Master Thesis. The research objective for this thesis project can be summarized as follows:

The research objective is to develop an optimization toolbox based on the concept of differential ray tracing and automatic differentiation tools for the design and optimization of space instruments based on freeform optics

Which can be rephrased into the following research question:

Is it possible to improve the optical performance and compactness of space instruments based on freeform optical surfaces?

In order to properly address the research question, started by tackling a series of relevant subquestions which provide insight into the complete research problem. Answering these subquestions has led to a satisfactory answer of the main research question.

1. *Is it possible to apply differential ray tracing to general freeform optics systems using automatic differentiation tools?*. The tedious and error-prone derivations of differential ray tracing in the general case for complex systems constitute the main drawback which currently hinders the application of this technique to freeform optics. Combining a formulation based on Fermat's path principle and the Implicit Function Theorem with the use of automatic differentiation tools we have circumvented this drawback.

2. *Can differential ray tracing data be used for the optimization of freeform systems?*. Differential ray tracing provides information about the optical system which is of great interest for optimization: how slight changes in the system parameters or in the ray data affect the propagation of rays through the system. This allowed us to compute derivative information in the form of Jacobian and Hessians of optical merit functions and to successfully use them for numerical optimization.

3. *What is the best optimization technique to design and optimize freeform systems?*. Differential ray tracing gives us access to a gradients, Hessian-vector products and Hessian matrices. Not all optimization algorithms exploit the same kind of information and thus do not lead to the same results and performance. We have shown that second-order optimization algorithms (trust region) provide the best balance between convergence speed and final results.

4. *Can differential ray tracing be used for other applications of interest in optical design?*. The information from differential ray tracing is not limited to optimization purposes. Tolerancing, which studies how optical performance degrades when mechanical tolerances are considered can benefit from the information from differential ray tracing.

Due to the way they are set up, these questions can be addressed in a consistent and systematic manner. Throughout this project we have tackled each question with satisfactory results. In a way, each question relates to a specific milestone in the development of **GDRT**, the toolbox for design and optimization of freeform optics which constitutes the main outcome of this research project.

1.4. REPORT STRUCTURE

This report has been structured as follows. First of all, in chapter 2 the complete mathematical framework of this project is thoroughly explained. This includes the technique of *differential ray tracing* (of paramount importance for this research), *Fermat's path principle* and how the *implicit function theorem* is used to extract valuable information from that principle. In addition, *automatic differentiation* a key tool for this project is presented; the main outcome of this project **Generalized Differential Ray Tracing** (GDRT) toolbox for optimization of freeform optics is thoroughly described. Topics such as the mathematical modelling of surfaces and the definition of merit functions are also covered.

Secondly, in chapter 3 the optimization case studies used to benchmark GDRT and their most relevant results are presented. We have investigated four optical systems which include: the **Cooke triplet** a simple and well-known system based on the use of lenses, an **spectrometer** which includes a diffraction grating, a compact **telescope** based on a monolithic block of infra-red material and a reflective **telescope** based on mirrors.

Then, in chapter 4 we present a technique to characterize the merit function landscape and look for additional local minima in freeform systems with a large amount of degrees of freedom. This technique takes advantage of the curvature information encoded in the Hessian matrix (available via GDRT) to generate a *smart* set of starting points around a known minimum for subsequent local optimization.

A proof of concept of parallel optimization based on the Alternating Direction Method of Multipliers (ADMM) is presented in chapter 5. We have shown that the technique of *consensus optimization* can be used to exploit some inherent parallelism in the optimization of optical systems.

In chapter 6, a complete tolerance analysis of the freeform spectrometer is provided, showing that it is possible to construct the optical system and still maintain sufficient levels of performance. Moreover, we investigated the *sensitivity* of freeform systems to mechanical tolerances and compared it to the sensitivity of a conventional system.

Possible extensions and improvements of GDRT for the future have been identified and analysis in chapter 7, the most important being the application of differential ray tracing to *tolerancing*.

Finally, chapter 8 summarizes the most relevant findings of this research and draws some conclusions.

2

Mathematical framework

In this chapter we analyse the key mathematical techniques that make GDRT, the toolbox for the optimization of freeform systems, possible. The name of GDRT comes from *Generalized Differential Ray Tracing*, therefore we will dedicate the first section of this chapter to introduce the notion of *differential ray tracing*. Soon, it will become clear that the main goal of this MSc Thesis, the extension of differential ray tracing to a general freeform optical system for optimization purposes, can be a challenging task without the use of certain methods and techniques.

These methods mainly include *Fermat's path principle*, *Implicit Differentiation* and *Automatic Differentiation*. We will thoroughly describe them in the remaining sections of this chapter. We will conclude by explaining the part of GDRT which takes care of how freeform surfaces are modeled mathematically and the *merit functions* which are differentiated via differential ray tracing to make optimization possible.

2.1. Differential ray tracing

Differential ray tracing is one of the cornerstones of this research project. In order to properly understand this concept, let us first introduce the idea of standard ray tracing.

2.1.1. Introduction

Ray tracing, i.e., computing the propagation of a discrete set of rays through a system, is the most common operation in optical design [30] and it is used extensively in commercial software tools to evaluate optical performance. The idea behind ray tracing is quite simple: given an input ray (defined by its position in the pupil plane and its direction as a field vector), one can trace the ray through the system, computing its intersections with the different optical surfaces, in order to determine its output position and direction in the image plane. Repeat this process for a sufficiently large number of rays and properly aggregate the results and you will have a realistic evaluation of the performance of your system. Whether you get a measure in terms of spot size, wavefront error, distortion, Modulation Transfer Function, or other exotic metrics only depends on the way you post-process your ray tracing results.

Nevertheless, in many situations it is convenient not only to have discrete ray data, but also to have access to derivatives of traced rays with respect to the construction parameters of the optical system [31]. This is the main idea behind ***differential ray tracing***, a collection of methods for computing the changes in optical performance metrics as ray data or design parameters are varied continuously [32]; this is of great importance for optical design, especially for the tolerancing and optimization of optical systems. The essence of differential ray tracing can be easily understood by looking at Figure 2.1. There we show a ray propagating from A to B through an optical system consisting of two mirrors (M1 and M2). We define x, y, z as the *global coordinates* of the reference system, while u_i, v_i, w_i correspond to *generalized local coordinates*.

The local coordinates u_i, v_i define the local reference system centred at surface i and, in the simplest case, they correspond explicitly to the *cartesian local coordinates* x_l, y_l but, in principle, they can adopt

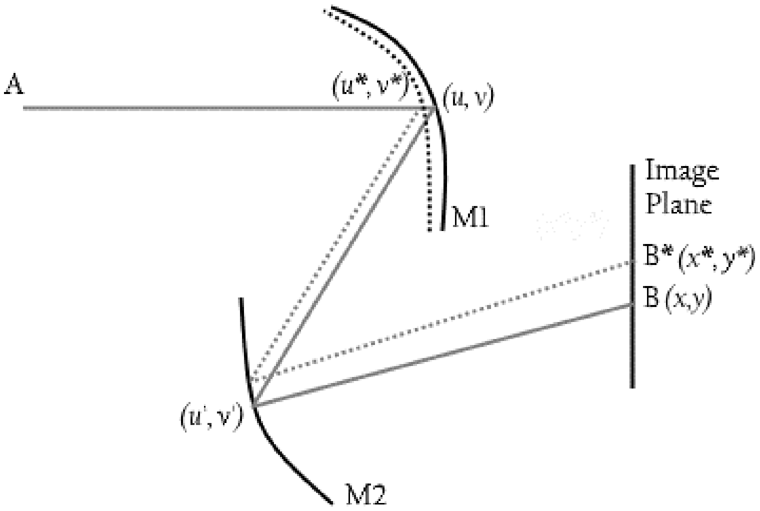


Figure 2.1: Schematic representation of *differential ray tracing*

any form. The sag equation w_i is just a mathematical way of representing the shape of an optical surface (sphere, paraboloid, plane...) and it is normally parametrized as a function of the generalized local coordinates and some defining parameters $w_i \equiv w_i(u_i, v_i, \Psi_i)$. Changes in the geometry of the surface are encoded via changes in Ψ_i. An example of Ψ_i is given below containing a matrix Θ which represents the rotation matrix needed to transform between the surface local reference frame and the global reference frame, the curvature of the surface C (needed for the sag equation) and a set of coefficients α_k of a polynomial series expansion which could define a freeform sag equation.

$$\Psi = \{\Theta, C, \alpha_k\} \tag{2.1}$$

In its nominal path, the ray intersects the mirrors at the points (u, v) and (u', v') and arrives at the image plane at point $B(x, y)$ (we directly use x, y coordinates for the image because it is a simple plane so there is no need for exotic parametrizations). If a certain change is induced in the parameters Ψ_1 of the first mirror M1, this would result in changes in the ray propagation through the system so that this time the ray reaches the image plane at a slightly different point $B^*(x^*, y^*)$. Differential ray tracing is the technique which studies *how infinitesimal changes in the parameters of the system* (a change in the definition of M1, in our case) alter the *propagation of rays through the optical system.*

It is easy to see how important this is for optimization purposes. We have already mentioned that the fundamental building blocks of optical performance evaluation are the *ray tracing* results (the position of rays on the image plane). And the main goal of optimization is to find a proper way to modify the *parameters* of the system so that the performance is enhanced. Differential ray tracing provides a direct link between those two things and thus is an extremely valuable technique.

There are basically two approaches to differential ray tracing: finite difference approximation or analytic methods. The first method is based on introducing small variations in the design parameters and performing discrete ray tracing. Then, the resulting changes in a performance metric are related to those variations via a finite difference approximation formula to obtain a numerical approximation of the derivatives. Although this method is used in many commercial software packages, its precision has been shown to be inadequate in some cases [33].

Finite difference approximations also require at least two ray tracing operations per design parameter and can therefore be prohibitively expensive for freeform optics systems when the number of degrees of freedom becomes large. The second analytic approach is to directly differentiate the algebraic ray tracing equations

[31], leading to purely analytic expressions that link the changes in system parameters to the behavior of rays. This method produces exact results, which do not suffer from the inherent inaccuracies of finite differences and it is the one we have adopted for this research project.

2.1.2. HISTORICAL DEVELOPMENT

Analytic differential ray tracing is not a modern discovery. Early studies in the field were limited to simple symmetric surfaces and a few design parameters [31]. Later developments extended the formulation to conic sections and aspheric surfaces ([30], [32], [34]).

In recent years, elaborate methods for complex and more general cases have been presented ([35], [36], [37],[38]). However, all these studies share a common characteristic: the mathematical framework is derived by hand and the necessity of keeping it fully analytic gives rise to long and tedious derivations of the differential ray tracing expressions for complex optical systems. The formulas usually contain a myriad of terms and the functional relationships between the different variables are in most cases highly non-linear; thus deriving a closed-form expression for the Jacobian matrix of aberrations is not easily achieved [33]. The computation of the Hessian matrix, of great importance for the convergence of optimization algorithms, is even more challenging [35].

Consequently, researchers are usually forced to adopt simplifying assumptions regarding the nature of the system (rotational symmetry, on-axis geometries, or simple surface representation), which hinder the applicability of differential ray tracing in the most general case. Even the most advanced commercial software packages for optical design which have built-in tools for differential ray tracing, are limited to simple examples [39]. This severely limits the direct applicability of this method to realistic scenarios in modern optical design where the ever-increasing performance requirements and constraints on optical systems force designers to come up with solutions of relatively high complexity.

Nowadays, with the onset of freeform optics as the new design paradigm, extending the formalism of differential ray tracing to these systems with a manual approach is close to being an unfeasible task. This fundamental drawback is circumvented in GDRT by the use of modern *automatic differentiation* algorithms applied to the expressions of differential ray tracing. This approach completely eliminates the necessity of hand coding the expressions and can handle any type of differentiable surface geometry in an efficient and accurate way. As with current state-of-the-art analytical differential ray tracing approaches, the resulting algorithm, which is the baseline of GDRT, only requires a unique ray tracing operation, independent of the number of design parameters under consideration.

In order to understand how GDRT operates, let us take a closer look at its fundamental building blocks.

2.2. LIGHT PROPAGATION

We begin by presenting the general approach to modeling the propagation of rays through an optical system. Let a ray represent the propagation of light between two points of the optical system $\mathbf{r}_0$ (a point in the object plane) and $\mathbf{k}_0$ (a point on the image plane). Let Ψ be a vector which contains the necessary parameters which define the optical system (radii of curvature, refractive index, freeform modeling parameters, etc).

The propagating ray will intersect each optical surface i at the points $\mathbf{P}_i$ which can be defined as follows:

$$\mathbf{P}_i = \mathbf{R}_i(\Psi) \cdot \{u_i, v_i, w_i\} + \mathbf{O}_i(\Psi) \tag{2.2}$$

$$w_i \equiv w_i(u_i, v_i, \Psi) \tag{2.3}$$

where $\mathbf{R}_i(\Psi)$ and $\mathbf{O}_i(\Psi)$ are the rotation matrix and offset vector representing the coordinate changes from the local reference system of surface i to the global reference system. The variables u_i, v_i represent the generalized local coordinates of surface i, which would correspond to local x, y in a commercial optical design software. They are referred to as *local* because they are defined on a reference frames which are centred on each of the surfaces i. The surface itself is given by the sag equation $w_i(u_i, v_i, \Psi)$ evaluated at the generalized coordinates and which depends on the particular choice of system parameters Ψ. This surface description is as general as it can be, and allows for both standard and freeform surfaces.

As a result of this, the propagation of a ray through the optical system is given by the different intersection with the surfaces, and can be represented by the following function:

$$f : \mathbf{r}_0, \mathbf{k}_0, \Psi \rightarrow \{\{u_1, v_1\}, \cdots, \{u_N, v_N\}\} \tag{2.4}$$

Which means that for a given ray $\{\mathbf{r}_0, \mathbf{k}_0\}$ and a set of system parameters Ψ we associate a set of *dependent* variables, the generalized local coordinates $\{\{u_1, v_1\}, \cdots, \{u_N, v_N\}\}$, which represent the propagation of that ray through the system. Once the values of the local coordinates are known, it is possible to transform them into global coordinates by using (2.2).

In light of this, we can conclude that the main objective of differential ray tracing is to obtain the following information about the derivatives of those quantities:

$$\Upsilon^k = \frac{d^k\{u_i, v_i\}}{d\{\mathbf{r}_0, \mathbf{k}_0, \Psi\}^k} \tag{2.5}$$

Where Υ^k constitutes a tensor containing the $k-th$ order derivatives of the ray tracing results with respect to: the ray parameters (like pupil position, field, etc) and the system parameters. The reason why these derivatives are important is because most metrics of optical performance are defined on the basis of ray tracing results. Therefore, knowing how the ray tracing results change as a function of changes on the system parameters and ray data is extremely valuable for optical optimization purposes.

The question of how GDRT gains access to these derivatives is not clear at this moment as u_i, v_i constitute ray tracing results which can only be solved via numerical techniques for the vast majority of cases. In otder to understand how this question can be answered, we need to take a look into *Fermat's path principle* and the *Implicit Function Theorem*.

2.3. FERMAT'S PATH PRINCIPLE

Discovered by Fermat in 1650, the principle of least time simply states that out of all possible paths that it might take to get from one point to another, light takes the path which requires the shortest time [40]. A modern generalization of the principle applicable to quantum mechanics states that the path must be *extremum* (not necessarily the *shortest*). Expressed in a mathematical way, the optical path length L travelled by a ray of light between two points A and B must be an extremum, in the sense of variational calculus:

$$\delta L = \delta \int_A^B n ds = 0 \tag{2.6}$$

In this case, n represents the index of refraction and ds the path length differential. When travelling through several optical surfaces, with different refractive index, the integral can be broken down into piecewise intervals. Assuming perfectly homogeneous media, the rays travel in straight lines and the integral can be transformed into the standard Euclidean norm between two points. Thus, for a system containing $N+2$ optical surfaces, with index of refraction $n_{i,i+1}$ between surfaces i and $i+1$, the optical path length can be written as:

$$L = \sum_{i=0}^{N+1} n_{i,i+1} \|\mathbf{P}_i \mathbf{P}_{i+1}\| \tag{2.7}$$

The application of Fermat's path principle to Equation (2.7) directly yields that all partial derivatives of L with respect to the generalized local variables must be equal to zero. Otherwise, the optical path would not be extremum:

$$\frac{\partial L}{\partial u_i} = \frac{\partial L}{\partial v_i} = 0 \qquad \forall i \in [1, \cdots, N] \tag{2.8}$$

which can be written as

$$f_{u,i} = n_{i-1,i} \frac{\partial \|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial u_i} + n_{i,i+1} \frac{\partial \|\mathbf{P}_i\mathbf{P}_{i+1}\|}{\partial u_i} = 0 \tag{2.9}$$

$$f_{v,i} = n_{i-1,i} \frac{\partial \|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial v_i} + n_{i,i+1} \frac{\partial \|\mathbf{P}_i\mathbf{P}_{i+1}\|}{\partial v_i} = 0 \tag{2.10}$$

We will refer to the different terms $f_{u,i}, f_{v,i}$ as the *Fermat error terms* throughout the remaining sections. As we are considering $N+2$ surfaces, the total amount of u_i, v_i variables is $2(N+2)$. The system of N equations defined in Equation (2.8) allows us to directly determine $2N$ variables, corresponding to the generalized coordinates of surfaces in between the first and the last. The remaining 4 variables (the pair of coordinates for the object plane and the image plane, for instance) are fully determined by the parameters of the ray being traced $\{\mathbf{r}_0, \mathbf{k}_0\}$.

2.4. IMPLICIT DIFFERENTIATION

We have seen that Fermat's path principle links the derivatives of the optical length $\partial L/\partial u_i, \partial L/\partial v_i$ so that the light propagation represents a physical system. But this does not provide (at least directly) the relevant quantities of interest for differential ray tracing $\partial\{u_i, v_i\}/\partial\Psi$.

Thankfully, the Implicit Function Theorem [41] allows us to unlock that information in an elegant way. Consider a generic function f such that $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$

$$f(x, y) = 0 \tag{2.11}$$

With $x \in \mathbb{R}^n$ the *dependent* variables and $y \in \mathbb{R}^m$ the *independent* variables. Under mild assumptions such as that the function is continuous and differentiable around the neighbourhood of a solution (x_*, y_*), and that the partial Jacobian $\nabla_x f(x_*, y_*)$ is non-singular, the Implicit Function Theorem states that:

"An implicit function $x = g(y)$ exists around y_* such that is continuous and differentiable with respect to u according to":

$$\nabla_y g(y) = -\nabla_x f(g(y), y)^{-1} \cdot \nabla_y f(g(y), y) \tag{2.12}$$

The immediate consequence of the theorem is that, provided that one has access to the system f, one can directly compute the derivatives of the *dependent* variables x with respect to the *independent* variables y without having to solve the system and without knowing the actual implicit dependency $x = g(y)$. Moreover, the theorem can be extended to compute derivatives of arbitrary order by applying the theorem to the previous result $\nabla_y g(y)$. How this theorem can be applied to Fermat's path principle will be clear in a moment.

Let us recall the set of $2N$ *Fermat error terms* $f_{u,i}, f_{v,i}$. These terms, constitute a system equivalent to that of the generic function f which depends both on the generalized local coordinates $\{u_i, v_i\}$ (consider them as the *dependent* variables) and the system parameters Ψ which act as *independent* variables. Consequently, the Implicit Function Theorem can be applied to compute $\partial\{u_i, v_i\}/\partial\Psi$. We will illustrate the whole procedure below, to show that it is in accordance with the general formulation of the theorem.

For any system parameter Ψ_j each *Fermat error term* must fulfill:

$$\frac{df_{u,i}}{d\Psi_j} = \frac{df_{v,i}}{d\Psi_j} = 0 \qquad \forall i \in [1, \cdots, N] \tag{2.13}$$

Which can be expanded according to the chain rule, for the $f_{u,i}$ terms

$$\frac{df_{u,i}}{d\Psi_j} = \frac{\partial f_{u,i}}{\partial \Psi_j} + \sum_{k=0}^{N} \frac{\partial f_{u,i}}{\partial u_k}\frac{\partial u_k}{\partial \Psi_j} + \sum_{k=0}^{N} \frac{\partial f_{u,i}}{\partial v_k}\frac{\partial v_k}{\partial \Psi_j} = 0 \tag{2.14}$$

With an analogous expression for the $f_{v,i}$ terms. The terms $\partial f_{u,i}/\partial u_k$ in Equation (2.14) represent derivatives of the *Fermat error terms* with respect to generalized coordinates of the surfaces, which mainly contain derivatives of the surface sag $w_k(u_k, v_k, \Psi)$ and of the coordinate transformations $\mathbf{R}_k(\Psi), \mathbf{O}_k(\Psi)$. In contrast, the terms $\partial f_{u,i}/\partial \Psi_j$ are derivatives of the *Fermat error terms* with respect to system parameters such as the radius of curvature or the freeform model parameters.

The equations from (2.14) can be arranged into a linear system of equations of the form $Ax = b$

$$\begin{bmatrix} \frac{\partial f_{u,1}}{\partial u_1} & \cdots & \frac{\partial f_{u,1}}{\partial u_k} & \cdots & \frac{\partial f_{u,1}}{\partial u_N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial f_{u,i}}{\partial u_1} & \cdots & \frac{\partial f_{u,i}}{\partial u_k} & \cdots & \frac{\partial f_{u,i}}{\partial u_N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial f_{u,N}}{\partial u_1} & \cdots & \frac{\partial f_{u,N}}{\partial u_k} & \cdots & \frac{\partial f_{u,N}}{\partial u_N} \end{bmatrix} \begin{pmatrix} \frac{\partial u_1}{\partial \Psi_j} \\ \vdots \\ \frac{\partial u_k}{\partial \Psi_j} \\ \vdots \\ \frac{\partial u_N}{\partial \Psi_j} \end{pmatrix} = - \begin{pmatrix} \frac{\partial f_{u,i}}{\partial \Psi_j} \\ \vdots \\ \frac{\partial f_{u,i}}{\partial \Psi_j} \\ \vdots \\ \frac{\partial f_{u,i}}{\partial \Psi_j} \end{pmatrix} \tag{2.15}$$

For the sake of simplicity, we only show the derivation for the $f_{u,i}$, but the extended form would include the $f_{v,i}$ terms as well. The system of equations f resulting from the application of Fermat's path principle is something which is readily available for any system once it has been defined. Therefore, GDRT has full access to it and could compute both Jacobians with respect to the generalized coordinates and the system parameters.

Consequently, by direct application of the Implicit Function Theorem, GDRT can unlock the derivatives of the ray tracing results with respect to the parameters of the optical system $\partial\{u_i, v_i\}/\partial\Psi_j$ for any surface of interest i and any chosen parameter j. This is by definition, the outcome of a differential ray tracing calculation.

To summarize, the application of Fermat's path principle to a generic optical system gives rise to a system of equations which links the different generalized local coordinates so that the ray propagation remains physically meaningful. In addition, the Implicit Function Theorem can be applied to that system of equations to automatically compute differential ray tracing information, in the form of derivatives of the ray tracing results with respect to system parameters. The formulation presented in this chapter, has been derived without making any assumptions regarding the number and type of optical surfaces or the parameters used to define the system; and thus is completely general and applicable to freeform systems of arbitrary complexity.

The only issue here is that the aforementioned Jacobians contain extremely complicated dependencies of the surface sags, coordinate transformation, ray trace results and systems parameters; and therefore cannot be derived by hand (this will become clear in the following section). It is at this point, that the use of *Automatic Differentiation* tools becomes a necessity. But before analysing the importance of Automatic Differentiation, let us first introduce the concept of *bilevel optimization* and how it relates to the *Implicit Function Theorem* and the problem of *optical design*.

2.4.1. BILEVEL OPTIMIZATION AND THE OPTICAL DESIGN PROBLEM

The way we exploit Fermat's system of equations and use the Implicit Function Theorem to unlock derivatives of optical merit functions is basically equivalent to a *bilevel optimization* problem [42]: a special kind of hierarchical problem where an inner optimization task (lower level) is embedded within an outer optimization (upper level). Its mathematical formulation adopts the following form:

$$\Phi(x) = \underset{y}{\operatorname{argmin}} \big(f(x,y): \qquad g(x,y) \leq 0 \big) \tag{2.16}$$

$$x = \underset{x}{\operatorname{argmin}} \big(F(x,y): \qquad y \in \Phi(x) \big) \tag{2.17}$$

The lower level problem shown in (2.16) represents a parametric optimization which determines the constraints $y = \Phi(x)$ which apply to the upper level problem shown in (2.17). The goal is to find the value x which minimizes the merit function $F(x,y)$ knowing that y will depend on the specific choice x, a requirement which makes bilevel optimization problems more difficult to solve.

Another way to understand the essence of bilevel optimization is to look at it from the perspective of *Stackelberg games*: a type of strategic game in economics in which a *leader* makes the first move and the *follower* decides its move based upon the result of the leader's action [43]. In this *optical game*, the leader is in charge of deciding the state vector x representing the parameters of the optical system, while the follower can be regarded as the one in charge of the ray tracing through the optical system, in this case y. The leader starts by making a move, i.e. by selecting a possible state x_0 for optimization. Based on that decision the follower computes the solution of the ray tracing problem $y_0 = \Phi(x_0)$ in such a way that the Fermat error terms $f(x,y)$ are minimized.

At this point, with the information about the follower's move, the leader can ponder over his next move, in other words, he can evaluate the merit function $F(x_0, y_0)$ which defines his strategy and decide which action x_1 to take next. The goal of the leader is to carefully decide which actions to take (how to adapt the optical system) taking into account the fact that the response of the follower will depend on his own choices (the ray tracing results which define performance directly depend on the leader's choice of optical system).

The application of the Implicit Function Theorem to obtain information regarding how the ray tracing results change as a function of the state vector (the $\partial y(x)/\partial x$ derivatives), is just a way for the leader to gain the knowledge of how his actions will affect the decision of his follower and exploit that knowledge to devise the best possible strategy. Here, the term *implicit* precisely represents the situation we are dealing with because the follower's response (which is ultimately affecting the success or failure of the leader) implicitly depends on the actions taken by the leader. The elegance and power of the Implicit Function Theorem is that it allows the leader to know how his actions would impact the outcome even before deciding which action to take. In other words, without implicit differentiation, the leader has no way of inferring the follower's response before actually making his decision and registering the response, and therefore the optimization problem would become extremely challenging.

2.5. AUTOMATIC DIFFERENTIATION

The use of *automatic differentiation* techniques to compute the derivatives which appear in differential ray tracing calculations in a precise and efficient way is one of the key features of GDRT. But before analysing in detail how we apply such techniques, we will motivate with a simple example why *automatic differentiation* is so important for this project.

2.5.1. MOTIVATION

Once all the necessary mathematical framework for generalized differential ray tracing from the previous sections has been set up, one could argue that a direct approach of hand-coding all the derivatives and functional relationships could be used to construct GDRT. In principle, such an approach would render the use of *automatic differentiation* unnecessary as all the differentiation would be taken care by the user.

However, it is easy to show that for the most general case of an optical system of moderate complexity, this can quickly become an overwhelming task. The amount of dependencies between the different components of the mathematical framework leads to chain rule-based derivations which are extremely tedious to construct and more importantly, extraordinarily difficult to debug.

To illustrate this, let us consider a single *Fermat error term* $f_{u,i}$ for a generic surface i. We will begin by analysing in detail what is needed to construct $f_{u,i}$. Later on, we will apply implicit differentiation to see what additional dependencies are required for the complete differential ray tracing. If we recall Equation (2.9), the Fermat term has the following structure:

$$f_{u,i} = n_{i-1,i}\frac{\partial\|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial u_i} + n_{i,i+1}\frac{\partial\|\mathbf{P}_i\mathbf{P}_{i+1}\|}{\partial u_i} = 0 \tag{2.18}$$

$f_{u,i}$ mainly contains derivatives of an Euclidean norm between pairs of points corresponding to the intersection with the previous surface $\mathbf{P}_{i-1}$, the surface of interest $\mathbf{P}_i$ and the following surface $\mathbf{P}_{i+1}$. One should note that in order to compute the norm properly, all three points must be expressed in the global coordinate system. The first term in the right-hand side can be expanded as:

$$\frac{\partial\|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial u_i} = \frac{\partial\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2+(z_i-z_{i-1})^2}}{\partial u_i} \tag{2.19}$$

Recalling Equation (2.2), each point in global coordinates depends on the generalized local coordinates and the reference transformation such that:

$$(x_i, y_i, z_i) = \mathbf{R}_i(\Psi)\cdot\{u_i, v_i, w_i\} + \mathbf{O}_i(\Psi) \tag{2.20}$$

$$w_i \equiv w_i(u_i, v_i, \Psi) \tag{2.21}$$

After expanding Equation (2.19), we reach

$$\frac{\partial\|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial u_i} = \frac{1}{\|\mathbf{P}_{i-1}\mathbf{P}_i\|}\left[2(x_i-x_{i-1})\frac{\partial x_i}{\partial u_i} + 2(y_i-y_{i-1})\frac{\partial y_i}{\partial u_i} + 2(z_i-z_{i-1})\frac{\partial z_i}{\partial u_i}\right] \tag{2.22}$$

Due to the dependency of $w_i(u_i, v_i, \Psi)$, the term $\frac{\partial z_i}{\partial u_i}$ must be further expanded to:

$$\frac{\partial z_i}{\partial u_i} = \frac{\partial z_i}{\partial w_i}\frac{\partial w_i}{\partial u_i} \tag{2.23}$$

Consequently, just to define half of a single Fermat term $\frac{\partial\|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial u_i}$ term, one would need to provide:

1. **Derivatives of the global coordinates with respect to local generalized coordinates** $\frac{\partial x_i}{\partial u_i}, \frac{\partial x_i}{\partial u_i}, \frac{\partial z_i}{\partial w_i}$ which will depend on the reference transformations for each surface $\mathbf{R}_i$ and on the system parameters Ψ.

2. **Derivatives of the sag equation with respect to local generalized coordinates** $\frac{\partial w_i}{\partial u_i}$. When using a combination of standard surfaces and freeform departures, these derivatives will expand into additional terms. These derivatives will carry dependencies on the local coordinates and the system parameters.

Up to here the derivations are tedious but doable. For a system with $N+2$ surfaces, one would have to set up $2N$ Fermat terms in the form of pairs $f_{u,i}, f_{v,i}$. Each will contain two terms like $\partial\|\mathbf{P}_{i-1}\mathbf{P}_i\|/\partial u_i$, which in turn contain 4 distinct derivatives.

The real issue comes when *implicit differentiation* is applied. Let us begin with the Jacobian $\nabla_u f$ from Equation (2.15). This requires us to differentiate each Fermat error term $f_{u,i}, f_{v,i}$ once again with respect to the generalized coordinates. A quick look at Equation (2.22) tells us that this will give rise to several situations:

When differentiating first with respect to u_i and then with respect to $u_{j\neq i}$, cross terms like the following will appear:

$$\frac{\partial\left(2(x_i - x_{i-1})\dfrac{\partial x_i}{\partial u_i}\right)}{\partial u_{i-1}} \tag{2.24}$$

$$\frac{\partial\left(2(z_i - z_{i-1})\dfrac{\partial z_i}{\partial w_i}\dfrac{\partial w_i}{\partial u_i}\right)}{\partial u_{i-1}} \tag{2.25}$$

When differentiating first with respect to u_i and then with respect to v_i, cross terms like the following will appear:

$$\frac{\partial\left(2(x_i - x_{i-1})\dfrac{\partial x_i}{\partial u_i}\right)}{\partial v_i} \tag{2.26}$$

$$\frac{\partial\left(2(z_i - z_{i-1})\dfrac{\partial z_i}{\partial w_i}\dfrac{\partial w_i}{\partial u_i}\right)}{\partial v_i} \tag{2.27}$$

Which will include, among other terms, cross derivatives of the sag equations $\dfrac{\partial^2 w_i}{\partial u_i \partial v_i}$.

When differentiating twice with respect to u_i or v_i, a similar situation will be observed, this time with second derivatives $\dfrac{\partial^2 w_i}{\partial u_i^2}$

And we have not even mentioned the derivatives of the $1/\|\mathbf{P}_{i-1}\mathbf{P}_i\|$ which will need to be added when expanding the terms $\dfrac{\partial^2\|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial u_j,\partial u_k}$. One can easily understand that the expressions will grow to enormous magnitude after all the chain rule applications. The possibility of successfully hand coding these relationships is close to negligible.

What's more, so far we have only analysed the derivatives with respect to **local generalized coordinates** u_i or v_i; but there is whole family of derivatives with respect to the system parameters Ψ. If we consider now the Jacobian $\nabla_\Psi f$, we would need to compute the derivatives of the whole set of $2N$ Fermat terms with respect to each parameter Ψ_j in our system definition Ψ. This set can contain as many types of variables as: transformation related (distances, tilts, decenters...), standard surface information (curvature, conic constant, groove frequency of diffraction gratings), freeform surface parameters, etc. Therefore, a wide variety of derivative terms will appear.

The derivative of the Fermat term with respect to Ψ_j takes the following form:

$$\frac{\partial^2\|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial u_i \partial \Psi_j} = \frac{\partial\left(\dfrac{1}{\|\mathbf{P}_{i-1}\mathbf{P}_i\|}[2(x_i - x_{i-1})\dfrac{\partial x_i}{\partial u_i} + 2(y_i - y_{i-1})\dfrac{\partial y_i}{\partial u_i} + 2(z_i - z_{i-1})\dfrac{\partial z_i}{\partial w_i}\dfrac{\partial w_i}{\partial u_i}]\right)}{\partial \Psi_j} \tag{2.28}$$

Imagine that Ψ_j represents a **transformation-related** parameter (the tilt X of surface i for instance). This will be included in the transformation matrix $\mathbf{R}_i(\Psi)$, which appears in derivatives of the type $\dfrac{\partial x_i}{\partial u_i}$. Therefore, Equation (2.28) will contain terms such as:

1. $\dfrac{\partial \|\mathbf{P}_{i-1}\mathbf{P}_i\|}{\partial \Psi_j}$ which will consist of terms of the form $\dfrac{\partial x_i}{\partial \Psi_j}$ which depend on the transformation matrix $\mathbf{R}_i$

2. $\dfrac{\partial^2 x_i}{\partial u_i \partial \Psi_j}$ which contain the derivatives of the transformation matrix with respect to Ψ_j.

Now let's assume $\Psi_{k\neq j}$ corresponds to **surface-related** parameters, like the curvature of a conic surface. Then Equation (2.28) will give rise to terms like:

$$\frac{\partial \left(2(z_i - z_{i-1}) \dfrac{\partial z_i}{\partial w_i} \dfrac{\partial w_i}{\partial u_i} \right)}{\partial \Psi_k} \tag{2.29}$$

After applying the chain rule, we will find terms of the form:

1. $\dfrac{\partial w_i}{\partial \Psi_k}$ first derivatives of the sag equation

2. $\dfrac{\partial^2 w_i}{\partial u_i \partial \Psi_k}$ cross-derivatives of the sag equation with respect to the local coordinates and the surface parameter

3. $\dfrac{\partial z_i}{\partial \Psi_k} \dfrac{\partial z_i}{\partial w_i} \dfrac{\partial w_i}{\partial u_i}$ combinations of derivatives of the global coordinates with respect to system parameters, derivatives of global coordinates with respect to local coordinates (related to coordinate transformations) and derivatives of the sag equation

Let us recall that this differential ray tracing calculation has been done for first derivatives, i.e. it only provides useful information for gradient-based optimization. This already required computing an extensive list of first and second order derivatives of very different nature and use the chain rule to combine them into immense expressions. If one wants to exploit Hessian information, the complete derivation presented in this section needs to be taken one step further, which means additional complexity in the form of third order derivatives of sag equations and the like.

We can thus conclude that although the general framework of differential ray tracing adopts a fairly simple form, the actual implementation of all the necessary expressions for a realistic optical system is a gargantuan task in itself. Hand-crafting the propagation of all the dependencies using the chain rule without making a mistake is almost impossible, and debugging a code of such scale and complexity would take an unreasonable amount of it.

There is actually no real gain on constructing these expressions by hand, except that of mathematical elegance. All these derivations are simply means to a more important end: the results of differential ray tracing to be used during optimization. Therefore, how we reach that end is of no concern to us so long as we obtain the correct result in the least amount of time. That is the main reason why *automatic differentiation* algorithms are so important to GDRT, as they automatically take care of propagating the derivatives according to the chain rule.

2.5.2. THE CONCEPT OF AUTOMATIC DIFFERENTIATION

Automatic differentiation (AD), also known as *algorithmic* differentiation, is a set of computational tools which are primarily based on the mechanical application of the chain rule to calculate the derivatives of functions given in the form of computer programs. AD tools take advantage of the fact that all computer programs, no matter their complexity, boil down to a sequence of basic arithmetic operations (addition, product), evaluations of elementary functions whose derivatives are known (such as $exp(\cdot)$) and some flow control operations (*if*, *else*, etc). By means of repeatedly applying the chain rule of differential calculus to all operations and functions inside the computer program, AD tools can compute any derivative of arbitrary order; and more importantly, the result is accurate up to machine precision. This task, which can be extremely time-consuming and error-prone if done by human means, instantly becomes trivial when AD tools are employed.

2.5.3. DIFFERENCES WITH *symbolic* AND *numerical* DIFFERENTIATION

Automatic differentiation is often mistaken with other techniques for the evaluation of derivatives. It is important to understand that there are fundamental differences between these methods and automatic differentiation

Symbolic differentiation.

This technique is probably the one closest to automatic differentiation and it is implemented by well-known computational tools such as Maple and Mathematica. The main difference between the two techniques is that *symbolic differentiation* treats the algebraic expression to be differentiated as a *tree* (a graph in which two nodes are connected only by *one* path), whereas *automatic differentiation* is based on *acyclic graphs* (which allow for two nodes to be connected by several paths. In addition, *symbolic differentiation* constructs an extra tree which represents the computation of the derivative, while *automatic differentiation* extends the function graph so that it can compute the derivative as well.

Consequently, the symbolic approach can cause the same sub-expression to appear in several places of the constructed derivative. Moreover, the length of the derivative expression tends to increase rapidly with the number of independent variables of the function, potentially leading to exponentially large expression which take unreasonable time to evaluate. In contrast, due to the aforementioned characteristics, automatic differentiation retains and shares intermediate results between the function and its derivative, leading to substantial simplification and a more efficient use of resources.

Another important point is that symbolic differentiation requires an *algebraic* expression of the function whereas automatic differentiation can easily cope with functions which are defined in the form of computational algorithms containing exotic features like loops and branches. Let us recall that the ray tracing results usually come in the form of numerical solutions of non-linear equations and thus do not have an algebraic form.

Numerical differentiation.

It is easy to see that *numerical differentiation* (probably the most popular technique for evaluating derivatives) and *automatic differentiation* are completely different things. First of all, numerical differentiation only provides *numerical approximations* of the derivatives, computed via a truncated series expansion, a *finite difference scheme* (such as the one shown below), while the method of automatic differentiation is entirely based on analytic expressions and chain rule operations.

$$f(x+h) = f(x) + \frac{\partial f}{\partial x} h + \mathcal{O}(h^2) \tag{2.30}$$

$$\frac{\partial f}{\partial x} \simeq \frac{f(x+h) - f(x)}{h} \tag{2.31}$$

Numerical differentiation is actually quite powerful and widely used by the scientific community, but it suffers from some inherent disadvantages; the most important being the fact that accuracy of the result depends on a proper choice of the step size h. In fact, the belief that an arbitrary reduction of h will enhance

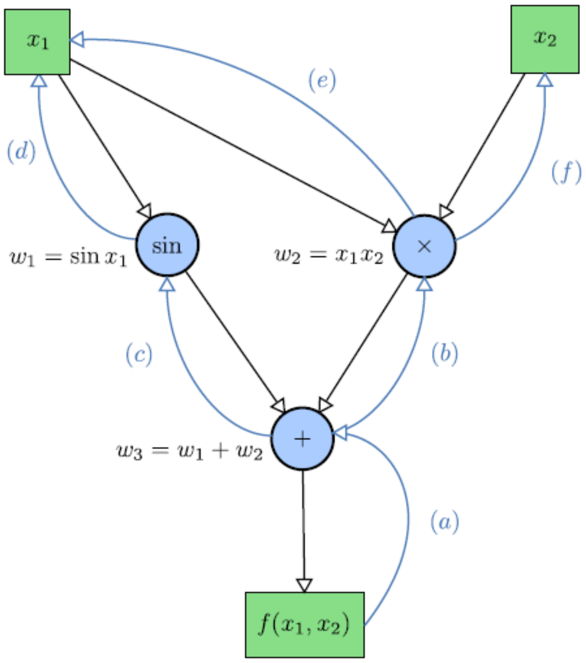


Figure 2.2: Computational graph of the function $f(x_1, x_2) = \sin(x_1) + x_1 x_2$. The flowchart for *reverse mode* automatic differentiation is shown in blue

the accuracy of the finite difference scheme indefinitely is a common misconception which goes against the basics of floating-point arithmetic.

It is important to note that the use of automatic differentiation for calculating derivatives is not unknown of in optical design. Some years ago a study showed that automatic differentiation can be faster and more accurate than numerical differentiation, and particularly well-suited for freeform optics [44]. Besides, the study stressed that despite of its superiority, automatic differentiation is yet disregarded for this applications.

2.5.4. COMPUTATIONAL GRAPH AND AD MODES

It was already mentioned that automatic differentiation treats the function as a computational graph. Now we will thoroughly explain what this means and how it relates to the two main modes of automatic differentiation: *forward* and *reverse* mode. Not to be mistaken with forward and backward differences of finite difference schemes.

Let us consider the function $f(x_1, x_2) = \sin(x_1) + x_1 x_2$. As shown in Figure 2.2 we can represent the function as an acyclic graph in which each operation is a node, which can have multiple inputs, and outputs can be used more than once. This is a great of visualizing how the function is mainly based on simple operations (the nodes) and how the computational flow goes. But how is the derivative constructed with automatic differentiation?

There are two major approaches in automatic differentiation. *Forward mode* starts at the fundamental part of the computational graph (the independent variables x_1, x_2 and works its way *forward* through the graph by constructing derivatives of intermediate variables in terms of their parents. This is the most intuitive way of thinking about the derivative, and it is how most of the people would apply the chain rule.

In contrast, *reverse mode* starts at the end of the graph and propagates the derivatives of the final result with respect to intermediate quantities, working its way back to the inputs.

The main difference between the two methods lies on the computational cost. Forward mode run time complexity scales linearly with the number of independent variables. Reverse mode scales linearly with the number of operations in the function. Consequently, when dealing with scalar functions with many inputs, the computation of gradients is more efficient via reverse mode differentiation.

To properly understand how the flow of reverse mode differentiation works, let us examine the example function in detail, using Figure 2.2 as a reference. We will refer to the quantities $\frac{\partial f}{\partial w_i}$ as gradients and to $\frac{\partial w_i}{\partial w_j}$ as adjoints. Starting from the result, we follow the flow backwards (from **(a)** to **(f)**) constructing the gradients of the function with the help of the adjoints:

$$(a) \quad \frac{\partial f}{\partial w_3} = 1 \tag{2.32}$$

$$(b) \quad \frac{\partial w_3}{\partial w_2} = 1 \quad \frac{\partial f}{\partial w_2} = \frac{\partial f}{\partial w_3}\frac{\partial w_3}{\partial w_2} = 1 \cdot 1 \tag{2.33}$$

$$(c) \quad \frac{\partial w_3}{\partial w_1} = 1 \quad \frac{\partial f}{\partial w_1} = \frac{\partial f}{\partial w_3}\frac{\partial w_3}{\partial w_1} = 1 \cdot 1 \tag{2.34}$$

$$(d) \quad \frac{\partial w_1}{\partial x_1} = \cos(x_1) \tag{2.35}$$

$$(e) \quad \frac{\partial w_2}{\partial x_1} = x_2 \quad \frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial w_1}\frac{\partial w_1}{\partial x_1} + \frac{\partial f}{\partial w_2}\frac{\partial w_2}{\partial x_1} = 1 \cdot \cos(x_1) + 1 \cdot x_2 \tag{2.36}$$

$$(f) \quad \frac{\partial w_2}{\partial x_2} = x_1 \quad \frac{\partial f}{\partial x_2} = \frac{\partial f}{\partial w_1}\frac{\partial w_1}{\partial x_2} + \frac{\partial f}{\partial w_2}\frac{\partial w_2}{\partial x_2} = 1 \cdot 0 + 1 \cdot x_1 \tag{2.37}$$

Judging from the expressions above it is clear why this method is called *reverse mode*. The natural thing for a person to do would have been to take the function $f = w_1 + w_2$, directly write the chain rule $\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial w_1}\frac{\partial w_1}{\partial x_1} + \frac{\partial f}{\partial w_2}\frac{\partial w_2}{\partial x_1}$ and then find the particular subexpressions. Instead reverse mode starts from the small and simple subexpressions and goes back until it reaches the complete gradient.

2.5.5. AUTOMATIC DIFFERENTIATION IN GDRT - THEANO

The usual case for optical design is to have a single scalar merit function f which contains all the performance metrics and constraints for the optical system; which depends on a set of degrees of freedom. For the particular case of GDRT, this set of degrees of freedom can be quite large, meaning that we have to deal with a scalar function with many inputs. Consequently, reverse mode (which we have already said it works well with many inputs and few outputs) is the best choice of automatic differentiation techniques.

Nowadays, there are many libraries available for automatic differentiation purposes, some specifically written in Python and others written in languages like C and C++ but well-prepared for Python use. To cite some examples: CppAD [45], Tensorflow [46] and Theano ([47]). Broadly speaking, the two main differences between purely Python-based tools and C or C++ tools for AD are *speed* and *ease-of-use*. Low-level language based tools like CppAD obviously outperform Python tools in terms of computational speed, but tend to lack the user-friendliness and familiarity that makes Python so suitable for fast prototyping and experimentation. Therefore, for GDRT which is already conceived as a Python toolbox, we decided to stick to AD tools with strong Python features, in this case Theano. This allowed for fast development and facilitated the integration of the AD functionalities with the rest of the GDRT modules.

Here, we present an example of derivative computation in Theano for a simple function, to quickly visualize the fundamental ideas behind this tool. The idea is to first define a Theano variable x and the expression for the function we want to derive y.

$$y = \sqrt{x^2 + 1} \tag{2.38}$$

$$\frac{dy}{dx} = \frac{x}{\sqrt{x^2+1}} \tag{2.39}$$

The call to **Theano.tensor.grad** takes care of computing the gradient and the **theano.function** call constructs a function which receives the value of the input x and returns the value of the gradient. For this simple example the derivative is straightforward and we can check the Theano result against its true value

```
import numpy as np
import theano
import theano.tensor as T

x = T.dscalar('x')
y = T.sqrt(x**2 + 1)
grad = T.grad(y, x)
f_grad = theano.function([x], grad)

f_grad(1)
>>> 0.70710678118654746

1./np.sqrt(2.)
>>> 0.70710678118654746
```

2.6. GDRT TOOLBOX

To further illustrate how GDRT is constructed and how this approach to freeform optics design differs from the state-of-the-art methods, we will present a detailed description of the structure and components of GDRT. The toolbox has been developed and expanded almost from scratch for this project, including new functionalities as the project progressed. The main GDRT module follows a modular structure composed of different Python Objects, each carrying out a specific set of tasks, which can be roughly classified into the following categories:

1. **Surface**: these are Python Objects in charge of the surface description. They include generic surface models like standard conic and biconic surfaces, planes and different kinds of freeform surfaces, as well as models for special surfaces like the object, entrance pupil and image plane.

2. **Field**: in charge of the reference system transformations and the normal vector of the Surface objects.

3. **Fermat Solver**: its main function is to set up the Fermat system of equations containing all the Fermat error terms and solving it.

4. **Ray Tracer**: takes care of all standard ray tracing operations between the different surfaces. It takes advantage of the already defined Fermat error terms to construct a 'Fermat-based Ray Tracer' (this idea is explained in the following section).

5. **Implicit**: takes care of the direct application of the Implicit Function Theorem to unlock the results of differential ray tracing.

6. **Optimizer**: in charge of all optimization related tasks such as merit function definition, handling of degrees of freedom and the computation of derivatives (gradient, Hessian, Hessian-vector product)

The GDRT toolbox also contains a set of auxiliary scripts and functions which take care of tasks like: solving large batches of linear systems efficiently, vectorizing important operations, definition of physical parameters (refractive index of materials as a function of wavelength), generating pupil-field-wavelength ray sampling schemes, etc.

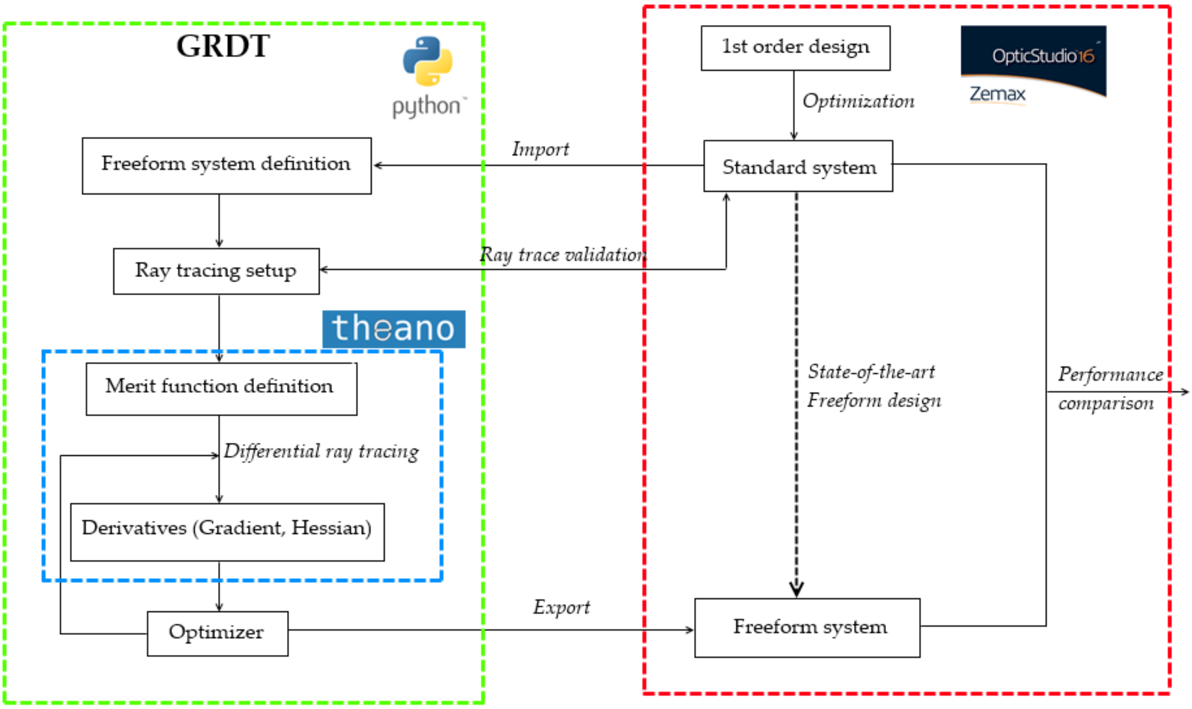


Figure 2.3: Work-flow diagram of GDRT

The general work-flow for defining and optimizing a freeform optical system with GDRT, and the interfaces with Zemax are shown in Figure 2.3. The different stages of the process can be summarized as follows:

1. The process starts with an optimization based on standard rotationally symmetric surfaces done in Zemax. Once the optimizer stagnates and can no longer improve the performance, the system is translated into GDRT.
2. A GDRT ray tracing is done to validate the Python-based system definition. The results of the ray trace are compared to those given by the Zemax ray tracing.
3. Once the system definition is validated, the merit function is defined in Theano as a function of the ray tracing variables and the system parameters.
4. The framework of differential ray tracing is applied to obtain the derivatives of the merit function with respect to system parameters. Theano automatically takes care of handling all chain-rule propagations.
5. Numerical optimization is done based on the merit function and derivative information to obtain a freeform system with enhanced performance.
6. Finally, GDRT uses the final system parameters to generate surface files compatible with Zemax. These files are loaded into Zemax and the optical performance is compared to that of the initial non-freeform system.

The validation of the imported initial system based on ray tracing cross-check ensures that the GDRT system and the Fermat ray tracer are properly set up and mimic the operation of Zemax. After that, GDRT is capable of operating autonomously without depending on Zemax. This is a very important feature as the import-export operations with Zemax are remarkably slow and should be kept to a minimum.

The fact that the final freeform system is exported to Zemax and that all performance evaluations are done using that state-of-the-art package ensures the validity and reproducibility of the toolbox results. GDRT is essentially in charge of *optimization* tasks. Once the optimization has finished, the final results of that optimization are judged from the perspective of an independently validated software tool like Zemax.

2.6.1. GDRT RAY TRACING

It is important to note that the ray tracing performed by **Ray Tracer** inside GDRT is substantially different to the conventional ray tracing which Zemax uses. The usual approach to ray tracing is to solve the equation for the intersection between a propagating ray (defined by a point P and a vector $\widehat{w}$) and a surface of interest (defined by its equation $z \equiv z(u, v, T)$, which depends on the local coordinates (u, v) and the local-global transformation T). In other words, the intersection $I(u, v)$ must be aligned with P along the ray direction and fulfil the surface equation:

$$P + \mu\widehat{w} = z(I, T) \tag{2.40}$$

Once the intersection I has been computed for one surface, Snell's law can be applied to calculate the outgoing direction, which involves the index of refraction n_{in}, n_{out} as well as the surface normal $\widehat{N}$. Then, the previous operation is repeated but this time I becomes the new P and the computed direction becomes $\widehat{w}$. By sequentially performing these steps one can calculate the complete set of intersections, i.e. the ray tracing results.

But GDRT uses a slightly unconventional form of ray tracing in order to exploit some of its key features. The underlying physical principle behind computing the intersections and applying Snell's law is that the ray tracing must always fulfil Fermat's path principle; i.e. the resulting paths computed with the conventional ray tracing are extremal paths. Therefore, the *Fermat error terms* representing the partial derivatives of the optical path length as defined earlier in (2.9) must vanish at the physical solution of a ray tracing operation. Any deviation from that path results in non-zero values of the terms.

As GDRT already employs the *Fermat error terms* to construct the differential ray tracing framework, one can easily recycle those expressions to construct a "Fermat ray tracing" based on the minimization of vectorial function $f \equiv (f_{u,i}, f_{v,i})$. As the expressions $f_{u,i}$ and $f_{v,i}$ constitute non-linear functions of the local coordinates (u, v), one must start by providing a guess for the intersection $I(u^0, v^0)$ and apply an iterative method, such as a non-linear least squares to find the value which minimizes the *Fermat error terms*. To further illustrate the differences between the two approach, Figure 2.4 is shown.

The main advantage of this type of ray tracing is that it efficiently uses information which was already needed in GDRT and avoids the development of a completely new ray tracing module. Nevertheless, it has several drawbacks; the most important being that it is not as numerically robust as a conventional ray tracing (an issue which probably could be improved by refining the choice of the iterative method used to minimize the terms).

2.7. SURFACE MODELLING

In this section we present the different types of surfaces implemented in GDRT. It is important to note that the kind of surface used determines what type of optical system we are dealing with, and thus influences how Fermat's path principle is applied. Below, we analyse how the principle applies to each particular type of system.

2.7.1. FERMAT'S PATH PRINCIPLE BY TYPE OF SURFACE

Reflective systems. For purely *reflective* systems which only consider mirrors, like a TMA telescope, the formulation of Fermat's path principle adopts the simplest form. The optical path length OP between two points A and B going through an reflective optical element can be written as:

$$OP = n_0|AP| + n_1|PB| \tag{2.41}$$

For the point P to be an extremum, the following conditions on the derivatives of the optical path length must be fulfilled

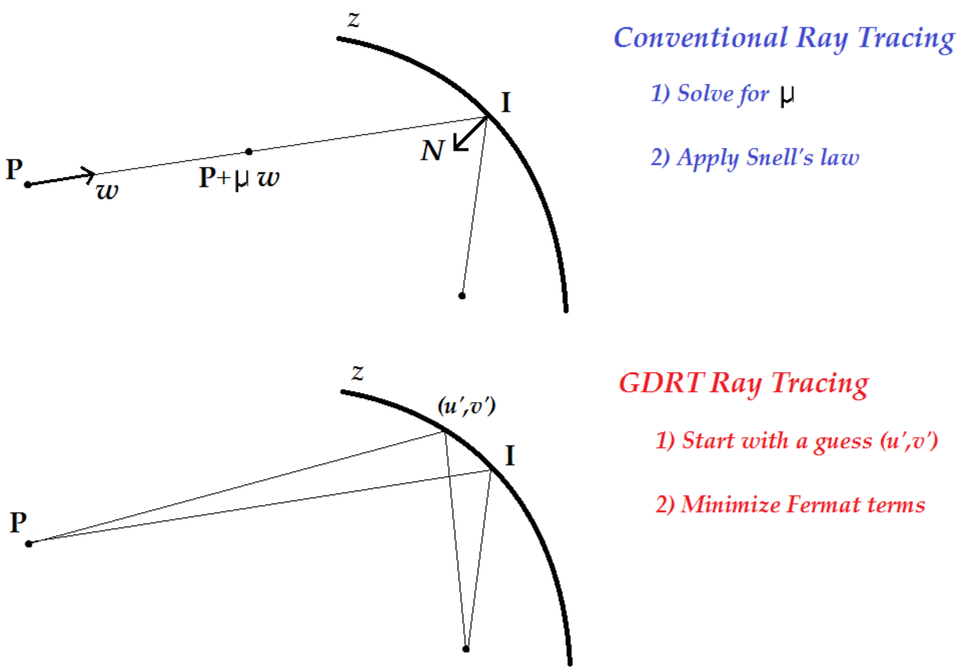


Figure 2.4: A comparison between conventional ray tracing and GDRT ray tracing

$$\frac{\partial OP}{\partial x} = \frac{\partial OP}{\partial y} = 0 \tag{2.42}$$

Refractive systems. The situation is slightly different when *refractive* elements are involved. This time, the dependency of the index of refraction n with the wavelength λ must be taken into account; leading to:

$$OP = n_0(\lambda)|AP| + n_1(\lambda)|PB| \tag{2.43}$$

Consequently, when tracing rays of different wavelengths, Fermat's path principle will give different results.

Diffractive systems. If the system of interest contains a *diffractive* element such as a diffraction grating, Fermat's path principle requires an additional modification to account for the grating effect. Diffraction gratings can be considered as optical elements which introduce an increment in the phase of the propagating ray depending on its wavelength, as given by the grating phase function Φ. That increment in phase can be translated into optical path and added to Fermat's path principle, given rise to:

$$OP = n_0|AP| + n_1|PB| + m\lambda\Phi \tag{2.44}$$

Where m is the diffraction order being considered and $\Phi(x, y)$ is the phase function of the grating. In conventional diffraction gratings containing straight lines constantly spaced along the x-axis, the phase function has the following form:

$$\Phi(x, y) = a_0 x \tag{2.45}$$

Where a_0 is the so-called line density, measured in lines per mm.

General systems. Consider a system which contains optical elements of the three types mentioned above: refractive, reflective and diffractive. Such a system could be an spectrometer containing several mirrors, a prism and a diffraction grating. The formulation of Fermat's path principle in this case, adopts the most general form possible:

$$OP = n_0(\lambda)|AP| + n_1(\lambda)|PB| + m\lambda\Phi \tag{2.46}$$

After applying the necessary conditions, we reach:

$$\frac{\partial OP}{\partial x} = n_0(\lambda)\frac{\partial|AP|}{\partial x} + n_1(\lambda)\frac{\partial|PB|}{\partial x} + m\lambda\frac{\partial\Phi}{\partial x} = 0 \tag{2.47}$$

$$\frac{\partial OP}{\partial y} = n_0(\lambda)\frac{\partial|AP|}{\partial y} + n_1(\lambda)\frac{\partial|PB|}{\partial y} + m\lambda\frac{\partial\Phi}{\partial y} = 0 \tag{2.48}$$

2.7.2. FREEFORM SURFACE MODELLING

The way we operate in this research project is by adding some sort of freeform model on top of standard surfaces, just like Zemax does. The modelling of surfaces for freeform optics is a complicated matter in itself. There is a wide variety of mathematical models to choose from, each with its own strengths and weaknesses. All freeform models can be classified according to two categories:

1. **Global**: these are models whose influence is global, meaning that a change in one parameter of the model will affect the complete domain of the surface. Examples from this category are: Zernike polynomials [48], XY polynomials [49], Q polynomials [50] or Chebyshev polynomials [51].

2. **Local**: these models only influence the surface locally, so that a change in one parameter affects a specific region of the domain. Examples of this type of model are: Splines [52], Radial Basis Functions [53], etc.

The particular choice of which models to use depends on a range of factors:

1. **The strengths and weaknesses of the model**. Global and local surface representations are completely different, and each has its own advantages. The designer must take that into consideration and choose the model which is more suitable for the application of interest.

2. **General characteristics of the optical system**. For instance: the type of instrument, on-axis or off-axis, the existence of symmetry...

3. **The type of optical elements it contains**. Is it a monolithic block of infrared material? Does it use mirrors, lenses or both? Does it contain a diffraction grating? Depending on the answer to these questions certain freeform models may be more suitable.

4. **The demands on performance**. This is certainly a key factor. In some cases, the system just needs a slight improvement and a very simple *soft-freeform* model with just a few terms (e.g. some astigmatism, coma and spherical aberration) and very mild departures are sufficient. In other situations, the demands on performance are so strict that only a *hard-freeform* freeform approach which changes the system radically can lead to compliance with the specifications.

5. **The amount of *degrees of freedom* your optimizer can handle**. More often than not this becomes the major limitation, usually because conventional optimization algorithms tend to scale poorly with dimensionality.

6. **The availability of the models**. This may seem trivial but not all freeform models are implemented in state-of-the-art optical design packages. Therefore, sometimes designers are forced to "work with what they have" and do have little or no choice as far as the model is concerned. The obvious alternative is to implement your own models but that takes significant time and effort.

7. **The designer's personal preferences** may also play a role. There is no definite answer to which freeform model is best so sometimes designers just go with what they think is better, or what they are more comfortable with.

Global models have the main advantage of only requiring a small number of parameters to completely reproduce a complex surface, thanks to their global behaviour; whereas *local* models usually require a higher amount of parameters to retrieve the same surface. Nevertheless, *local* models have the advantage of allowing the user to model localized effects which *global* models can simply not reproduce.

Some feasibility studies done at OHB before the start of this thesis revealed that a *hybrid* approach, which combines both *global* and *local* surface representation yields the best balance and helps reach competitive levels of performance while keeping the number of degrees of freedom at a reasonable level. The main idea is that an underlying *global* model with just a couple of parameters can reproduce most of the bulk surface effects needed to improve the performance, while a *local* model takes care of the remaining local effects that a *global* model cannot capture.

Below we thoroughly describe the implementation of this *hybrid* model for all the types of optical surfaces GDRT can handle.

A) Freeform mirrors and lenses.

This type of freeform surface representation is mainly used for the **spectrometer** system, the **Cooke** triplet and the **telescopes**, where reflective and refractive surfaces are involved. It contains a combination of *global* and *local* terms to model the additional departure with respect to a standard surface. Thus we can write the freeform contribution as:

$$z(x, y) = z_{\text{global}}(x, y) + z_{RBF}(x, y) \tag{2.49}$$

$$z_{\text{global}}(x, y) = ay^2 + by + cx^2 + d \tag{2.50}$$

$$z_{RBF}(x, y) = \sum_{i}^{N} w_i \Psi(\|\mathbf{x} - \mathbf{x}_i\|) \tag{2.51}$$

In this case the *global* terms correspond to a form of XY-polynomials. The *local* terms represent a Radial Basis Functions (RBF) definition. This type of surface modelling is very simple. It works by setting up a grid of nodes $\mathbf{x}_i$ in a certain domain. At each node, a basis function Ψ is placed with an associated weight w_i. Then, the value of the surface at a certain point $\mathbf{x}$ is computed as a linear combination of the basis functions evaluated at that point. The degrees of freedom of this model are the weights w_i. By changing their value, any type of surface can be reproduced as RBF constitute a complete base, which might not necessarily be the case for other surface representation.

The basis function $\Psi(\|\mathbf{x} - \mathbf{x}_i\|)$ is chosen by the designer and it takes the form of any function which fulfils the following conditions:

1. It is **local**, meaning that it has influence in a limited area of its domain
2. It depends on some measure of **distance** between the evaluation point (x, y) and each of the nodes of the RBF grid (x_i, y_i)

This means that functions such a Gaussian or an inverse quadratic can act as basis for an RBF surface model. For this research project we have decided to use Gaussians as our basis functions; as previous studies have shown that they are useful and simple to implement.

The fact that the basis functions depend on the **distance** between points and nodes, opens up the possibility of using two different kinds of RBF models: *Cartesian* and *polar*.

Cartesian freeform. The simplest approach is to define the RBF terms in such a way that they depend on the standard Cartesian distance as shown below for a Gaussian RBF. This formulation is the most common one, and it is frequently used throughout this project

$$\Psi(\|\mathbf{x} - \mathbf{x}_i\|) = e^{-((x-x_i)^2+(y-y_i)^2)/\varepsilon^2} \tag{2.52}$$

Polar freeform. However, in some situations the designer might need the optical elements to exhibit some kind of *polar* behaviour. For instance, one might be interested in having a freeform character only along the radial coordinate. In that sense, although the surface does not strictly follow a standard conic definition, it has azimuthal symmetry. We will refer to this type of surface as **polar freeforms**.

The only difference with respect to a Cartesian freeform is that this time the RBF depend on the radial coordinate ρ and not on θ. Both types of surfaces have been investigated during this research project, with promising results.

$$\Psi(\|\mathbf{x} - \mathbf{x}_i\|) = e^{-((\rho-\rho_i)^2)/\varepsilon^2} \tag{2.53}$$

B) Freeform diffraction gratings. Most lithographic gratings are usually manufactured in the form of planar gratings with perfectly straight grooves separated at a constant frequency a_0 [54]. In this situation, and assuming that the light is perfectly collimated at the diffraction grating, all rays experience the same diffraction effect no matter the field and pupil position, as given by the grating equation:

$$m\lambda a_0 = sin\theta_i + sin\theta_r \tag{2.54}$$

Where m is the diffraction order ($m = \cdots, -1, 0, 1, \cdots$), λ is the wavelength, θ_i is the incidence angle and θ_r is the reflected angle (both measured with respect to the surface normal). Only having the single degree of freedom of the groove frequency a_0, forces the designer (or more correctly, Zemax) to choose a value which *on average* works best for all field points. One can easily understand that although this can simplify the manufacturing process of a diffraction grating, from the point of view of optical performance, it is far from being the optimum choice. Ideally one would like to adapt the shape of the grating to the particular needs of each ray. This type of technology is usually referred to as Variable Line Spacing diffraction gratings ([55], [56]).

By allowing the diffraction grating to become a freeform surface, meaning that the groove frequency is now not constant but a function of the grating coordinates, one can locally change the amount of diffraction effect imparted by the grating to each ray. As a result of this, the diffraction grating not only does its job of dispersing the light but is starts playing an active part in the correction of optical aberrations. This is a similar idea to that of *hybrid lenses*, a type of lenses which have diffractive structures directly machined on top to combine their refractive effect with diffractive capabilities for infra-red applications [57].

One might argue that allowing this variation in line spacing will lead to non-physical diffraction grating designs impossible to manufacture. But the empirical results presented in this report suggest that the freeform departures on optical surfaces tend to be very small compared to the original shape and that the freeform shape effects in diffraction gratings are so small that the constant groove frequency term remains dominant. Consequently, the final design exhibits a slight "bending" of the grooves due to the spatial variation of the groove density; but although noticeable the overall effect is mild.

There is no doubt that the manufacturing of these freeform gratings will be more challenging than that of standard design. Nevertheless, a quantitative analysis of these issues, at this point of the research work (where no data on the manufacturing capabilities is available), is impossible and certainly beyond the scope of this project. In any case, some successful examples of manufacturing of freeform gratings can be found in the literature [58], showing that ESA considers this a very promising solution to achieve compact and cost effective instruments for small satellites.

Coming back to the modelling of these gratings, we have mentioned that the intensity of the diffraction effect needs to be a function of the grating coordinates. This is the same as saying that the phase function of the grating needs additional spatial dependencies apart from the standard linear term. The phase function Φ is needed for the application of Fermat's path principle because it represents the additional optical path induced by the diffraction grating on the propagating rays. By adopting a similar approach to the one used for the mirrors, we can write the phase function Φ as:

$$\Phi(x, y) = \Phi_{global}(x, y) + \Phi_{RBF}(x, y) \tag{2.55}$$

Φ_{global} contains both the standard linear term $a_0 x$ plus some *global* terms in the form of polynomials which respect the required x-symmetry.

$$\Phi_{global}(x, y) = a_0 x + bx^2 + cy^2 \tag{2.56}$$

Φ_{RBF} follows the same type of RBF approach that we used for the Cartesian freeform mirrors

$$\Phi_{RBF}(x, y) = \sum_{i}^{N} w_i \Psi(\|\mathbf{x} - \mathbf{x}_i\|) \tag{2.57}$$

In the end, no matter what type of surface you are modelling (lens, mirror, grating...), GDRT will work with a set of **parameters** of a *global* model and a set of **weights** of an *RBF* model. Those are the degrees of freedom of the freeform surface, and will be used as variables for the optimization process.

2.8. Merit functions

Merit functions are a key feature of GDRT as they represent the basis for any optimization procedure. The main purpose of a merit function is to encode into a scalar mathematical expression the optical performance upon which the system will be optimized. In complex systems such as the freeform spectrometer, one wants to optimize for several quantities at once (in this case RMS spot radius, dispersion, keystone and magnification), but not all of them are of the same importance to the designer. Therefore, the particular preferences and needs of each system must be properly accounted for in the merit function itself.

In other to do so, some "importance weights" w_i can be introduced so that the crucial components have a greater impact on the value of the merit function; thus the optimizer will make a greater effort to reduce those terms. As a result of this, a generic merit function in GDRT will look like this:

$$f(x) = w_1 \frac{f_1(x)}{f_1(x_0)} + \dots + + w_n \frac{f_n(x)}{f_n(x_0)} \tag{2.58}$$

Where $f_i(x)$ is a metric of optical performance such as RMS spot radius evaluated at the variable state x. It is important to note that all metrics are scaled by its initial value $f_n(x_0)$ before optimization so that all quantities are comparable. This helps avoid situations in which several metrics have very different orders of magnitude. In that case, the optimizer could reduce the overall merit function by greatly increasing the smaller metrics and only slightly reducing the bigger ones. For instance, if $f = A + B$, where $A_0 = 1000$ and $B_0 = 1$, a suitable new state would be $A = 995$, $B = 4$; but it is obvious that from a design point of view this is a terrible update as A is only a 0.5% better, while B is 4 times worse.

Finally, it may be necessary to incorporate some constraints into the optimization to ensure the system does not deviate too much from a certain limit. This can easily be done by defining penalty functions p_x into the merit function itself.

$$p_x = g(x - x_0)^2 \tag{2.59}$$

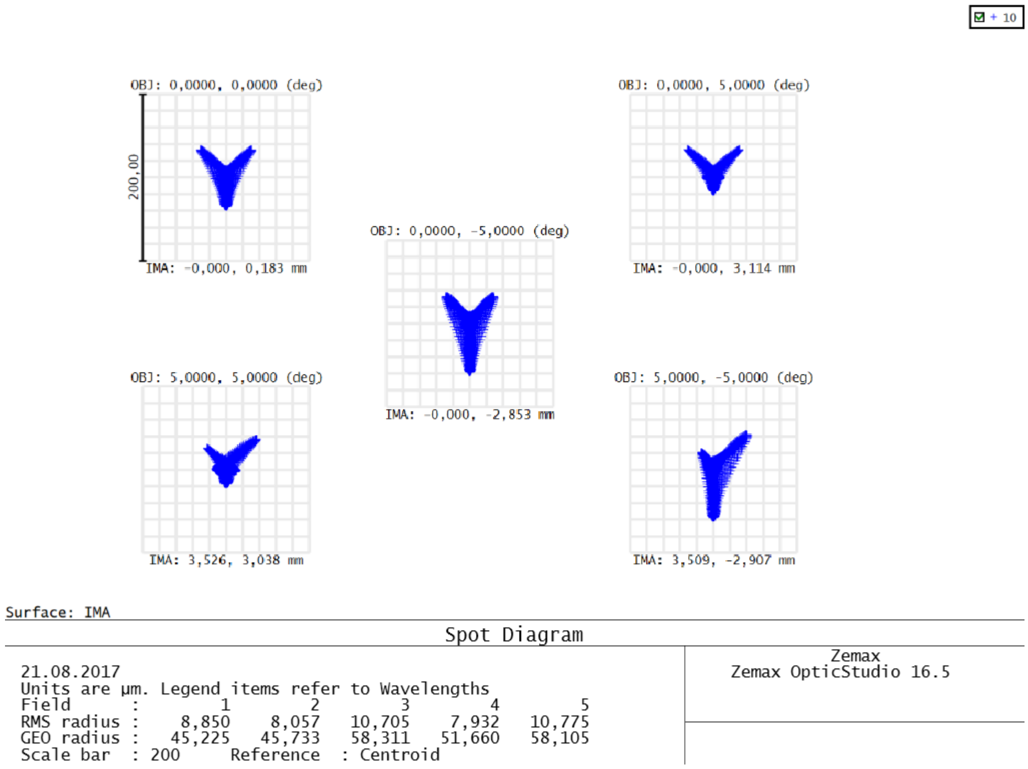
$$f(x) = w_1 \frac{f_1(x)}{f_1(x_0)} + \dots + + w_n \frac{f_n(x)}{f_n(x_0)} + \dots + p_x \tag{2.60}$$

Provided that the final expression of the merit function depends on information about the ray tracing operations, GDRT will be perfectly capable of computing its **gradient**, **Hessian-vector** product and the **Hessian** up to machine precision.

Here, we provide a short summary of common metrics of performance which we have implemented in GDRT. With rudimentary knowledge of Python, a user can combine these metrics or quickly define new ones to create a merit function for a particular optical system of interest.

2.8.1. RMS spot radius

An ideal optical system, without accounting for diffraction, would focus all rays coming from the pupil with the same field angle onto a perfect point on the image plane. However, in a realistic system this will never be the case and, in the presence of optical aberrations, the rays will intersect the image plane forming a cloud of points of finite size which is usually referred to as the *spot*, see Figure 2.5. Therefore, the size of


☑ + 10
OBJ: 0,0000, 0,0000 (deg)
200,00
IMA: -0,000, 0,183 mm
OBJ: 0,0000, 5,0000 (deg)
IMA: -0,000, 3,114 mm
OBJ: 0,0000, -5,0000 (deg)
IMA: -0,000, -2,853 mm
OBJ: 5,0000, 5,0000 (deg)
IMA: 3,526, 3,038 mm
OBJ: 5,0000, -5,0000 (deg)
IMA: 3,509, -2,907 mm

```
Surface: IMA
                                 Spot Diagram
21.08.2017                                                  Zemax
Units are µm. Legend items refer to Wavelengths      Zemax OpticStudio 16.5
Field       :       1        2        3        4        5
RMS radius  :   8,850    8,057   10,705    7,932   10,775
GEO radius  :  45,225   45,733   58,311   51,660   58,105
Scale bar   : 200      Reference  : Centroid
```

Figure 2.5: Example of a Spot Diagram from Zemax. Each plot corresponds to a field point, the size of the spots is given at the bottom

the spots represents a good metric of performance for systems which exhibit a substantial amount of optical aberrations and are far from the diffraction limit.

$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{2.61}$$

$$RMS_{spot} = \sqrt{\frac{1}{N}\sum_{i}^{N} r_i^2} \tag{2.62}$$

The usual way to define the metric is by means of a Root Mean Square method applied to the different intersections of the rays being traced with the image plane, leading to the notion of **RMS spot radius**. The RMS is computed with respect to a reference of the spot such as its geometrical centroid (x_c, y_c), but sometimes the chief ray (the one passing through the center of the pupil) is used. This metric which is widely used in optical software packages like Zemax, is one of the basic merit functions defined in GDRT.

2.8.2. WAVEFRONT ERROR

The notion of wavefront refers to the surface of equal phase perpendicular to the light path propagation. Thus, for a perfect collimated beam the wavefront is a plane; while for a perfect convergent or divergent beam the wavefront is a sphere. Inevitably, the propagation of light through a real optical system induces optical path differences which translate into aberrations of the wavefront (see Figure 2.6). The outgoing light which arrives at the image plane will no longer have a spherical wavefront; the shape of the actual wavefront tells a lot about the dominant aberrations of the optical system, such as coma, astigmatism, spherical aberration, etc.

In light of this, one can see that wavefront error defined as the departure from a reference sphere serves as a valid metric of performance for an optical system. In contrast with RMS spot radius, which only required the ray tracing results (x_i, y_i) at the image plane, the calculation of wavefront error is more complicated. It requires computing the optical path for each ray from the pupil up to the image plane; a computation which

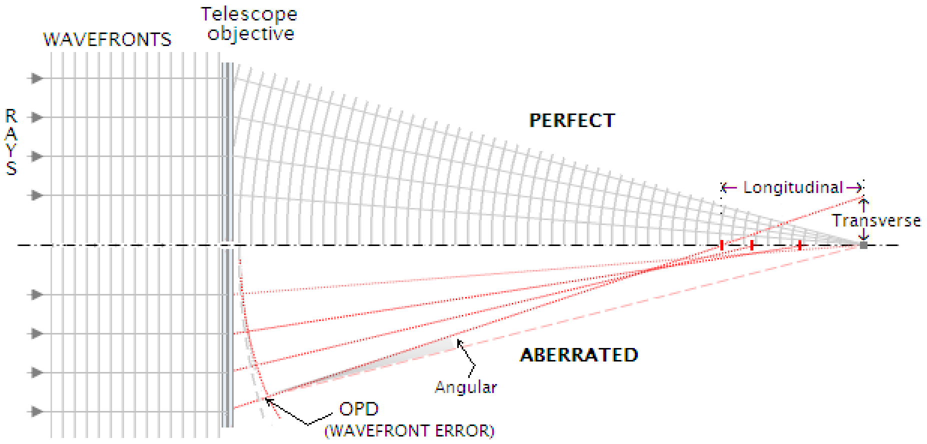


Figure 2.6: The origin of wavefront error as an optical path difference (OPD). Source: Telescope-Optics

additionally requires compensation of the inherent tilt of the incoming wavefront for all field points outside the axis. And then an extra ray tracing operation up to a reference sphere is needed to calculate the complete wavefront error. As the wavefront error represents a complete 2D map over the pupil, it is not directly suitable as a merit function and it is common to compute the RMS wavefront error to translate it into a scalar value:

$$w_i = OP_i - OP_{ref} \tag{2.63}$$

$$RMS_{wave} = \sqrt{\frac{1}{N}\sum_{i}^{N} w_i^2} \tag{2.64}$$

Where OP_i represents the total optical path for a point i in the pupil, OP_{ref} is the optical path of the reference ray (usually the chief ray).

Both spot radius and wavefront error are consequences of the same phenomenon, the existence of optical aberrations, but they have different realms of applicability as merit functions. For systems with low impact of optical aberrations, optimizing for RMS spot radius can eventually lead to a situation in which the size of the spot is close to the radius of the Airy disk. In that case, as the system is close to being diffraction-limited the notion of spot radius looses significance and it is advisable to switch to an optimization based on wavefront error, which is still valid for this situation.

2.8.3. DISTORTION

The essence of distortion is that a real optical system will not image a square grid perfectly, but rather distort its shape. This is caused by the actual wavefront arriving shifted with respect to a reference sphere, which causes varying point height magnification in the image (see Figure 2.7). Despite the shift, as the wavefront remains spherical distortion does not lead to a loss of image quality, but it is an undesirable effect to control and minimize as much as possible. If the image is distorted, parts of it could lie outside the detector array losing information; or the detector would have to be oversized and thus some pixels will be left unused.

Defining distortion as a merit function is quite simple because knowing the ideal focal length of the system and the field of view, one can infer how a square grid should look like on the image plane, and use it as a reference to compute deviations from the ray tracing.

$$d_{i,j} = \sqrt{(x_i - x_i^{ref})^2 + (y_j - y_j^{ref})^2} \tag{2.65}$$

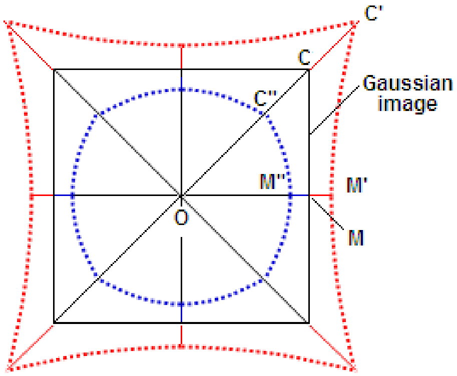


Figure 2.7: Distortion in an optical system. Source: Telescope-Optics

$$RMS_{dist} = \sqrt{\frac{1}{N}\frac{1}{M}\sum_{i}^{N}\sum_{j}^{M} d_{i,j}^{2}} \tag{2.66}$$

2.8.4. FOCAL LENGTH, F-NUMBER, IMAGE SIZE

Controlling the size of the image, for a given entrance pupil diameter and field of view, implies controlling the focal length and F-number of the optical system. The size of the image is actually just a particular case of distortion control in which we are only interested in controlling the boundaries of the grid and do not care about the distortion within the grid. Therefore, when inner distortion is not an issue but the control of either focal length or F-number is necessary, image size can be used as a merit function.

2.8.5. MAGNIFICATION

In the most general case, magnification is defined as the ratio between the size of the image and the size of the object. If a particular requirement on magnification applies, one can define the magnification error as the deviation between the true image size and the nominal size. Based on that a merit function can be constructed.

But for GDRT, magnification was used in the context of *slit spectrometers*. In such a system, the entrance slit defining the field of view of the spectrometer is imaged along the x dimension of the detector for the different wavelengths λ_i of the spectral range. This allows the study of the spectral properties of the sources being imaged, such as the identification of absorption lines on the atmosphere or the characterization of the spectrum of distant stars.

In a similar way to distortion and image size, one would like the image of the slit to have the same size for all wavelengths on the focal plane so that it fills exactly the pixel array. Thus we can define magnification as a merit function in the following way:

$$m_i = x_{max}(\lambda_i) - x_{ref} \tag{2.67}$$

$$M = \sqrt{\frac{1}{N}\sum_{i}^{N} m_i^2} \tag{2.68}$$

It is important to note that this metric only accounts for what happens at the edge of the array. The image of the slit could have varying distortion along the x dimension but, on average, lead to the same magnification.

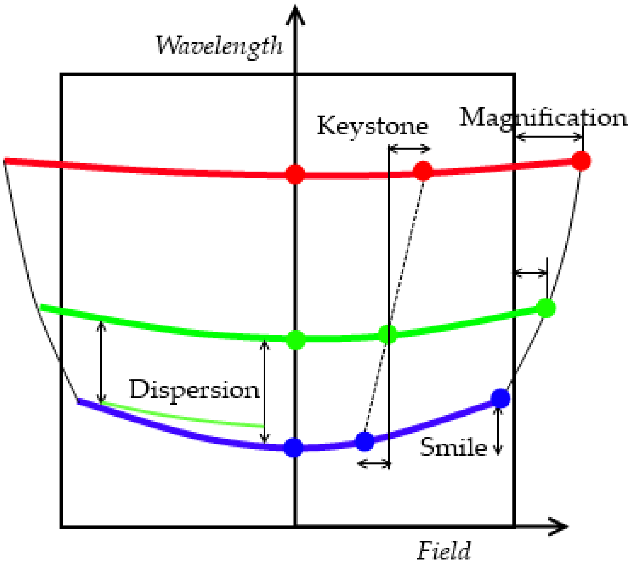


Figure 2.8: Spectral features to be optimized or at least characterized in a conventional spectrometer design

In the particular case of spectrometers, light is separated by wavelength and imaged onto the detector plane, which means that one spatial dimension is lost at the cost of introducing a spectral dimension. This particular feature brings up a new set of performance metrics specific to this type of system, see Figure 2.8, which include **keystone** and **dispersion**.

2.8.6. KEYSTONE

Keystone is defined as a change in field-magnification with wavelength. As mentioned above, even for perfect magnification, between the inner and outer field the imaging of the slit could have distortion. Keystone accounts for the distortion along the x dimension which depends on the wavelength, meaning that points with the same nominal field f_x but different nominal wavelength λ, have a different x position on the detector plane. Thus keystone can be translated into a merit function as follows:

$$k_i(x_i) = \frac{1}{M}\sum_j^M (x_i(\lambda_j) - x_i^{ref})^2 \tag{2.69}$$

$$\kappa = \frac{1}{N}\sum_i^N k_i \tag{2.70}$$

2.8.7. DISPERSION

If keystone was the change of field-magnification with wavelength, dispersion is just its spectral equivalent; defined as the change in spectral-magnification with field. Ideally, the sampling distance between wavelengths along the spectral axis should not depend on the field point and be equal to the intended spectral sampling. In reality, this distance between wavelengths will change slightly along the x-axis, resembling a change in spectral magnification. Therefore, we can define dispersion as:

$$\delta_j(\lambda_j) = \frac{1}{N}\sum_i^N (y_j(x_i) - y_j^{ref})^2 \tag{2.71}$$

$$\Delta = \frac{1}{M}\sum_j^M \delta_j \tag{2.72}$$

2.8.8. SMILE

One should note that in order to compute dispersion, one wavelength (usually the primary wavelength) is used as a reference which means that the metric is relative instead of absolute. Consequently, even in the absence of dispersion the shape of the different slit images on the detector can end up being curved. This phenomenon is known as the *smile* of the spectrometer. Although being an undesired effect, smile is usually not taking into account during optimization and it can be compensated via post-processing of the data cubes, provided that proper calibration and characterization has been done.

3
Results

In this chapter, four case studies of optical systems which have been optimized with GDRT are presented to illustrate the capabilities of this tool. The first one corresponds to the famous **Cooke triplet** which has served as a sort of benchmark for optical research in the past and has a more *academic* feel to it. The second one corresponds to a realistic space optical instrument: a **spectrometer** based on a double TMA and a diffraction grating. The third one is a concept design of freeform **telescope** based on a monolithic block of infra-red material, aimed at fitting within the volume limitations of a CubeSat platform. The third system, is a reflective **telescope** consisting of three freeform mirrors.

One of the key assets of these examples is that they display all the relevant features of GDRT in its complete range of optical systems: purely *refractive* (Cooke triplet), *diffractive* + *reflective* (freeform spectrometer), *refractive-reflective* (compact monolithic telescope) and purely *reflective* (telescope).

3.1. Cooke triplet

The Cooke triplet, invented in 1893 by H. Dennis Taylor for the optical company T. Cooke & Sons, constitutes the design with the smallest number of optical elements capable of correcting all 7 Seidel aberrations [59]:

1. *Achromatic*: spherical aberration, coma, astigmatism, field curvature and distortion.
2. *Chromatic*: axial color and lateral color.

It consists of a set o 3 lenses, one negative *flint* glass lens in the middle surrounded by two positive *crown* glass lenses (see Figure 3.1). This leaves the system with a total of 14 degrees of freedom which correspond to the 6 curvatures, 3 lens thicknesses, 2 air spaces and 3 glass choices.

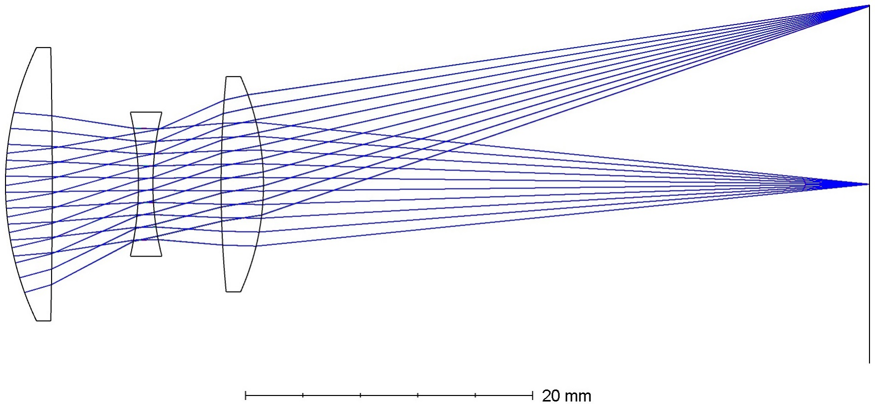


Figure 3.1: Optical layout of the Cooke triplet

Despite its apparent simplicity, the Cooke triplet was a great technological advancement for its time, and since then it has served as an example in many academic papers on optical design (e.g. [60]). For this reason, we decided to use the Cooke triplet as a benchmark for GDRT.

3.1.1. Differential ray tracing

One of the major strengths of differential ray tracing is that it can provide accurate derivatives of quantities of interest of optical systems without having to rely on finite-difference approximations [30]. In this example, we show how this technique can be used to compute derivatives of the position of rays on the image plane, with respect to variables of the Cooke triplet.

In order to make this possible, first of all GDRT had to be extended so that it could handle the fact that the Cooke triplet is a refractive system. Refractive elements such as glass lenses suffer from chromatism, meaning that their index of refraction varies with wavelength, giving rise to chromatic aberrations such as axial and lateral color. Consequently, the ray tracing operations embedded in GDRT as well as all differential ray tracing calculations need to incorporate this wavelength dependency.

This requires the implementation of the so-called **Sellmeier** equation [61], which is an empirical relationship between n the index of refraction of a medium and λ the incident wavelength:

$$n^2(\lambda) = 1 + \frac{B_1\lambda^2}{\lambda^2 - C_1} + \frac{B_2\lambda^2}{\lambda^2 - C_2} + \frac{B_3\lambda^2}{\lambda^2 - C_3} \tag{3.1}$$

The Sellmeier coefficients B_i, C_i can easily be found in the databases of optical glasses. For this example we decided to use the Cooke triplet file available in Zemax Optics Studio. We select a single ray to be traced through the system up to the image plane; corresponding to pupil coordinates $P_x = 0, P_y = 1$ and field coordinates $H_x = 0, H_y = 0$. Our quantity of interest as far as differential ray tracing is concerned will be the first derivatives of the y-coordinate on the image plane with respect to each of the design curvatures of our system:

$$\frac{dy}{dc_i} \tag{3.2}$$

In spite of its simple look, that derivative is anything but straightforward. It contains information on how the outcome of a ray tracing operation varies with respect to a fundamental quantity of the optical system. In order to obtain that derivative by hand, one would need to compute the derivatives of the ray tracing equations which contain information on curvatures, positions (distances and tilts), index of refraction, surface derivatives, angles of incidence for each surface; all of that entangled in the form of highly non-linear relationships. It is tedious and error-prone (but doable) for single ray in simple on-axis systems with few spherical surfaces like the Cooke triplet, but it is extremely cumbersome for a complex optical system with many degrees of freedom and many rays being traced.

That is the main reason why most of the time, differential ray tracing results are approximated by finite-difference techniques. Nevertheless, thanks to the use of *automatic differentiation* tools in GDRT, that propagation is done automatically under the surface, reducing the effort to a simple call to Theano.

The results as given by GDRT for these derivatives are presented below in Table 3.1. In order to validate the results, finite-difference approximations were computed using the ray tracing information from Zemax. To further strengthen the claim that differential ray tracing computed via automatic differentiation is superior to the finite-difference approach, we decided to compare the results of one derivative dy/dc_i with those from Zemax, for varying values of the step size.

In other words, we compared the accuracy in derivative computation of GDRT with that of Zemax as a function of the finite-difference step size h. The results as shown in Figure 3.2 reveal a well-known issue of finite-differences: the accuracy of the approximation improves (i.e. the difference with respect to GDRT

Table 3.1: Derivatives of the Cooke triplet, computed with differential ray tracing

Curvature	**Value** $[mm^{-1}]$	**Derivative** dy/dc_i $[mm^2]$
c_1	0.045426	-166.025083
c_2	-0.002295	147.432003
c_3	-0.045018	-107.559584
c_4	0.049281	103.207876
c_5	0.012550	-118.378438
c_6	-0.054362	132.589605

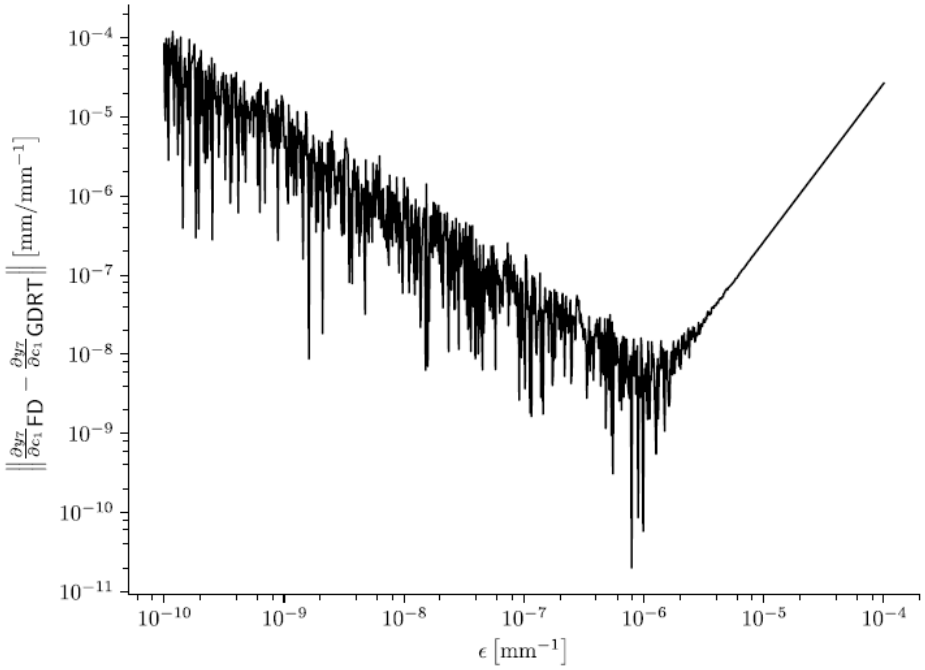


Figure 3.2: Comparison of first derivative accuracy bewteen GDRT-based automatic differentiation and Zemax-based finite differences

goes down) as the step size h decreases with a slope proportional to the order of the scheme being used. Nevertheless, the improvement is not eternal and at some point floating-point arithmetic errors of stochastic origin start to severely degrade the accuracy of the approximation. Although the deviations are usually small, they certainly do not help when precise evaluations of gradients are required to ensure decent convergence of optimization algorithms; in which case automatic differential becomes a must.

An equivalent procedure of automatic differentiation can be set up to compute the derivatives with respect to other variables of interest such as the thickness of the lenses, air spaces, conic constants, tilts (in case we want to create an off-axis system) or any parameter of a freeform surface representation, as we will show in the following section.

Moreover, the choice of the function to differentiate is unrestricted. Any scalar quantity which depends on the behaviour of rays can be defined as a quantity of interest for GDRT; this includes: distortion, wavefront error, spot size, MTF and spectral metrics such as keystone and dispersion.

Differential ray tracing is not limited to the calculation of first order derivatives [62]. In fact, derivatives of any order are easily accessible no matter their exotic nature. This way, GDRT can gain access to the Hessian matrix of the system and use it either for optimization purposes, or sensitivity analysis.

In Figure 3.3, the Hessian matrix $H(c_i, c_j)$ as computed by GDRT is provided. This matrix encodes very valuable information regarding the Cooke triplet. The diagonal of H reveals that the last curvature c_6 has the highest impact on the position of y for the ray being traced. From the rest of the matrix we can extract

$$\begin{bmatrix}
\mathbf{-203.391} & -122.903 & 667.954 & -1069.534 & 1560.172 & -1834.037 \\
-122.903 & \mathbf{-362.302} & -502.203 & 865.443 & -1295.523 & 1515.861 \\
667.954 & -502.203 & \mathbf{294.147} & -207.153 & 492.681 & -538.422 \\
-1069.534 & 865.443 & -207.153 & \mathbf{340.875} & -311.585 & 314.512 \\
1560.172 & -1295.523 & 492.681 & -311.585 & \mathbf{-276.584} & 177.157 \\
-1834.037 & 1515.861 & -538.422 & 314.512 & 177.157 & \mathbf{-807.840}
\end{bmatrix}$$

Figure 3.3: Hessian matrix of the Cooke triplet

that changing the two curvatures of a lens at the same time, i.e. $\frac{d^2y}{dc_i dc_{i+1}}$ for $i = 1,3,5$ has little impact on y (something to be expected as each lens has curvatures of similar value but opposite sign). Moreover, changing the first and last curvatures at the same time has the biggest impact on the result.

All this information might seem intuitive once the results are available, and the general trends can be guessed from the beginning. But having access to the actual values in the form of a Hessian matrix for optimization purposes is extremely useful. Another advantage of GDRT (in addition to the accuracy we mentioned ealier) is that this result via numerical differentiation requires a substantial amount of ray tracing operations. For n curvatures, the Hessian matrix contains $n(n+1)/2$ independent derivatives, in this case 21. For each derivative (assuming we use a central difference scheme for the second derivative) one needs to do 3 ray tracing operations $f(x), f(x-h), f(x+h)$, which leads to a total number of 63.

63 consecutive ray tracing operations for a Cooke triplet might not take a lot of time, however, for a complex freeform system with a large number of degrees of freedom N, not only does the amount of ray tracings scale with N^2, but also each each operation becomes more time consuming. In contrast, differential ray tracing on GDRT has the advantage of only requiring a single ray tracing operation for the computation of all the derivatives.

To summarise, due to its analytic nature, the results of differential ray tracing are not subject to accuracy losses and numerical instabilities like finite-difference approximations; and require a smaller amount of ray tracing operations leading to significant savings in computational time.

3.1.2. OPTIMIZATION

The previous section showed how GDRT can provide derivatives of any quantity of the optical system for a single ray trace. That is a somewhat academic exercise of limited application which demonstrates the fundamental workings of the GDRT. Therefore, in this section we present a full scale example of how freeform optics optimization can be used in the Cooke triplet.

The underlying idea is very simple. With the 14 degrees of freedom inherent to the Cooke triplet there is a limit to what can be achieved in terms of optical performance. But the standard Cooke triplet can be taken as a starting point for a design which relies on freeform surfaces to bring the performance to unexplored levels.

GDRT has been developed in such a way that it can generate a starting design based on the parameters of any optical system created with Zemax Optics Studio or a similar package. Then GDRT extends the system definition so that it can handle freeform surfaces, mainly in the form of additional surface departures modeled with Radial Basis Functions (RBF).

Based on its differential ray tracing capabilities, GDRT can compute (up to machine precision) the gradient and Hessian matrix of any merit function, no matter the complexity of the optical system or the number of degrees of freedom. With that information available, we can perform numerical optimization of the system.

In order to make the results as realistic and verifiable as possible, we copied the conditions for the Cooke triplet example from Zemax, in terms of pupil, field, wavelength sampling. This particular Cooke triplet has a circular field of view of 20° in radius. Due to the rotational symmetry of the system, a small number of field points defined in the first octant can be used to control the complete field of view (see Figure 3.4). Based on

Table 3.2: Physical constraints of Cooke triplet

Parameter	**Value**
Pupil radius	5 mm
Wavelength range	480 - 550 - 650 nm
Field radius	20 deg

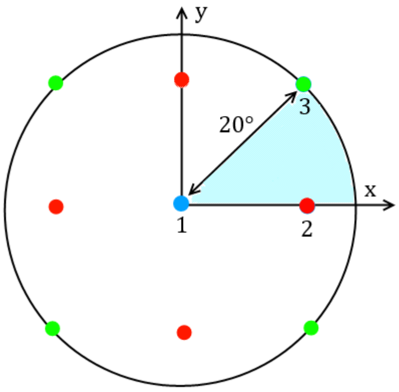


Figure 3.4: Sampling configuration for the Field of View of the Cooke triplet

these three field points (blue, green and red) the rest is simply symmetrised to account for the complete field of view.

This symmetry property is also exploited by GDRT in its freeform optics definition. Only RBF weights influencing the first octant are set as variables. The rest of the surface area is symmetrized according to the values for the control octant. This significantly reduces the amount of variables needed for optimization, but also ensures that the final performance is symmetric. The merit function was set as a simple linear combination of RMS spot radius and distortion. The starting values of RMS spot radius for the Zemax Cooke triplet are shown in Figure 3.5. Then, the system was optimized using GDRT until convergence.

In order to ensure the GDRT are completely valid, the final performance of the system was entirely verified in Zemax Optics Studio, just like for the initial system. The freeform departures generated during the optimization process can easily be uploaded into Zemax in the form of .dat files as Grid Sag surfaces.

The outcome of the optimization in terms of RMS spot radius can be seen in Figure 3.6. For all wavelengths, the performance across the field of view varies between 2 and 6 μm; a substantial improvement with respect to the initial design, specially around the edge of the field of view.

More details on the actual performance of the *freeform* Cooke triplet, as well as a comparison with the initial design, can be found in Tables 3.3 and 3.4. One can observe that the average improvement in distortion is approximately 95% and around 45% for RMS spot radius.

Judging by the results, one might tend to think that the optical system must have changed radically during optimization. The truth is that the departures introduced by the freeform terms are usually very small compared to the original surfaces. In Figure 3.7 the surface sag of the front glass of the Cooke triplet is shown at different stages. In Figure 3.7a the complete sag which comprises both the *standard* spherical sag and the *freeform* sag is presented, while Figure 3.7b only shows the *freeform* contribution as computed by GDRT. It can be observed that the magnitude of the freeform sag is substantially smaller than that of the spherical term.

Nevertheless, we have seen that despite the small magnitude of the freeform terms, the impact on the overall optical performance of the system is quite extraordinary, effectively reducing the distortion by more

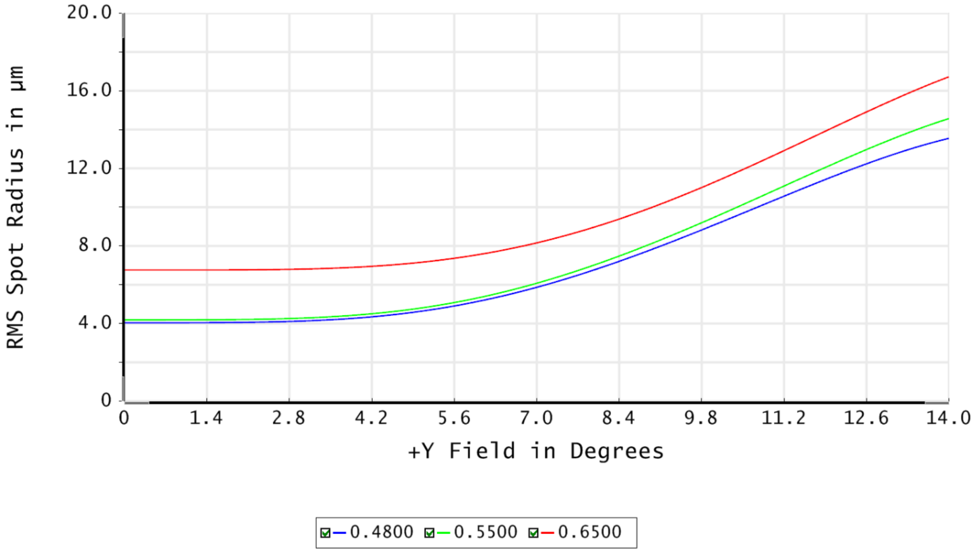


Figure 3.5: Initial RMS spot radius vs field of the Cooke triplet

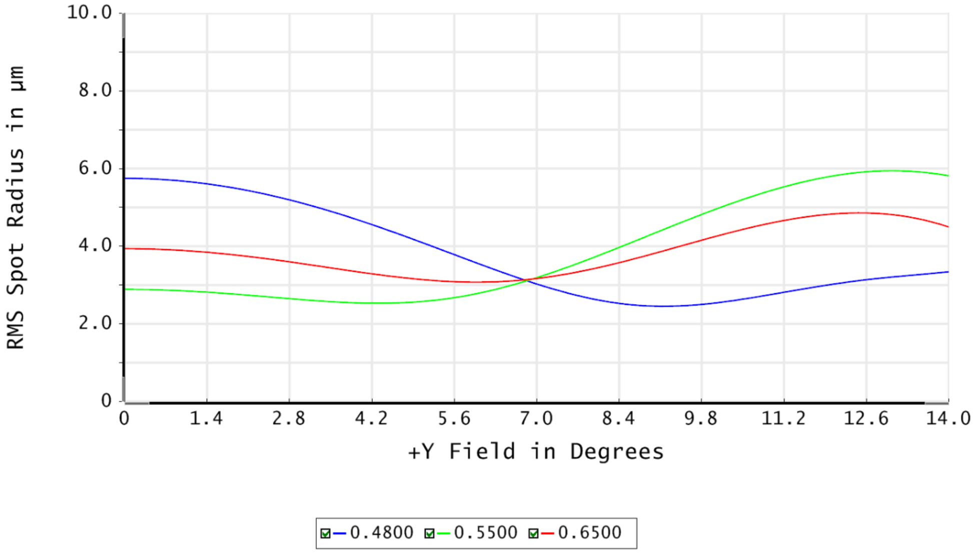


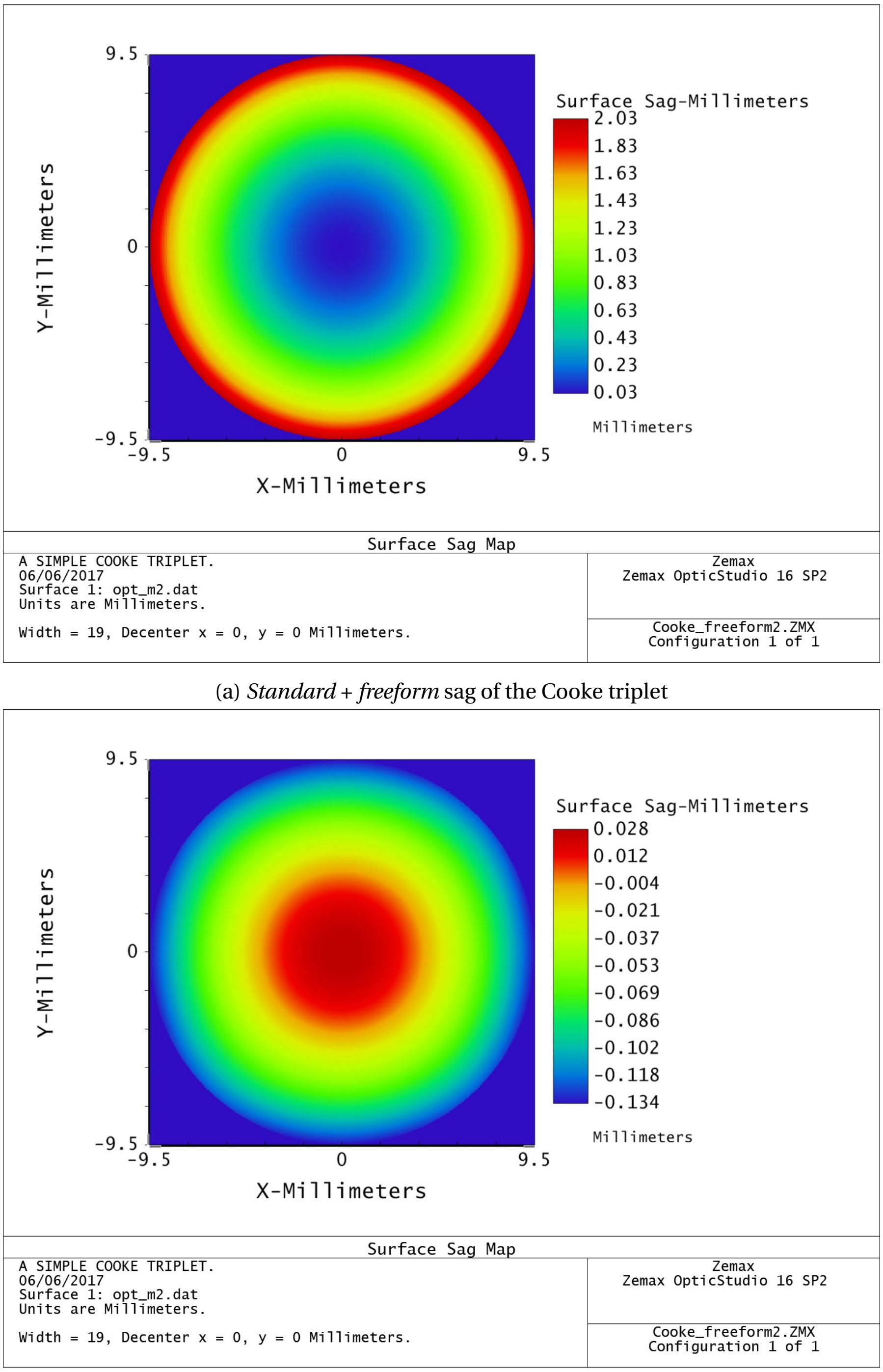
Figure 3.6: Final RMS spot radius vs field of the *freeform* Cooke triplet

Table 3.3: Distortion results for the *freeform* Cooke triplet

Wavelength [nm]	**Initial value**	**Final value**	**Improvement [%]**
480	0.0637 %	0.0038 %	94.0
550	0.0614 %	0.0007 %	98.9
650	0.0610 %	0.0021 %	96.6

Table 3.4: RMS spot radius results for the *freeform* Cooke triplet

Wavelength [nm]	**Initial average value [um]**	**Final average value [um]**	**Average improvement [%]**
480	7.10	3.78	30.08
550	7.43	3.85	45.32
650	9.61	3.86	57.44



(a) *Standard* + *freeform* sag of the Cooke triplet

(b) Isolated *freeform* sag of the Cooke triplet

Figure 3.7: A comparison between the *standard* and *freeform* contributions to the sag of the first surface in the Cooke triplet

than 95%, and the spot radius by almost 50%. This means that freeform optics can dramatically improve the performance of an optical system with only slight changes on the surfaces.

Despite the fact that the optimization results are quite illustrative, the Cooke triplet is by no means the ideal system to demonstrate the full potential of GDRT. Freeform optics are better suited for more complex systems with plenty of off-axis surfaces and significant aberrations, where the lack of symmetry becomes a powerful asset rather than a disadvantage.

Before jumping towards the next optical system, we will dedicate a section of this report to an important point about numerical optimization.

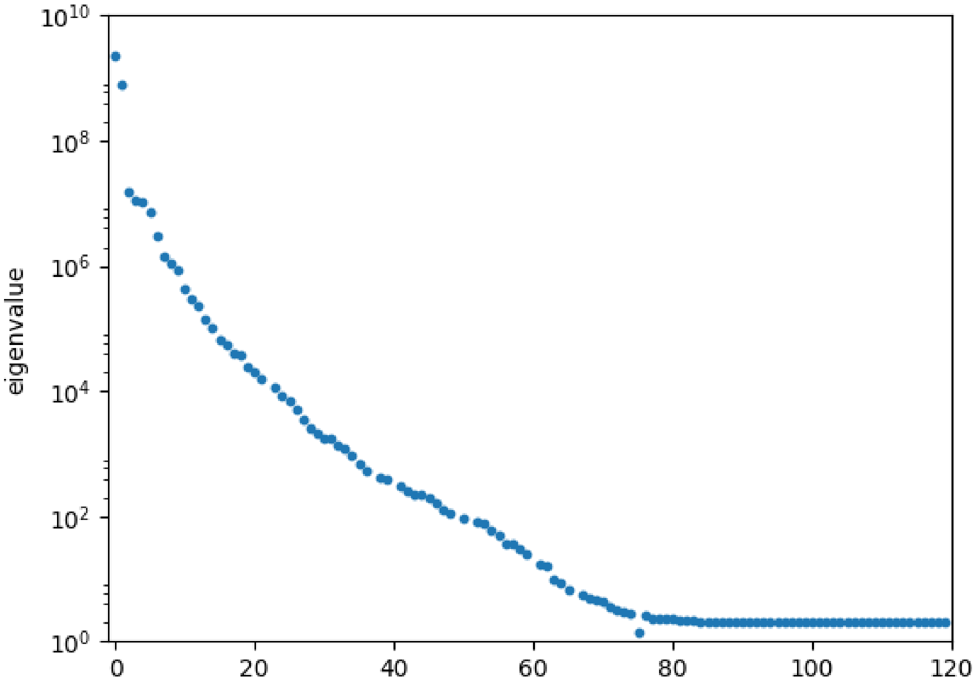


Figure 3.8: Eigenvalue spectrum of the freeform Cooke triplet

3.1.3. A NOTE ON HESSIAN-BASED OPTIMIZATION

In most software packages for optical design just as Zemax it is common to only use gradient-based numerical algorithms for the optimization tasks. One of the reasons behind this choice has already been covered: computing second order derivatives (such as the Hessian) with finite-differences and discrete ray tracing operations can be remarkably time-consuming. Another reason is the fact that for most the optimization of simple optical systems, gradient-based algorithms perform well enough to simply not justify the change towards second-order.

In this section, we provide experimental evidence which supports the claim that algorithms based on second order derivative information are needed for an effective optimization of complex freeform optical systems.

We will use the freeform Cooke triplet as an example. Based on the standard Cooke triplet, we defined a freeform version with a total of 120 degrees of freedom (20 per surface) using a *global* + *RBF* description. At the starting point, the Hessian matrix as provided by GDRT is readily available. This allows us to compute its eigenvalue decomposition which is shown in Figure 3.8.

The eigenvalue decomposition of the Hessian matrix provides valuable information regarding the merit function landscape, and the potential behaviour of optimization algorithms. In light of the results, we can conclude that there is a wide disparity in the magnitude of the eigenvalues, varying from almost 10^{10} to around 1. Consequently, the *condition number* $\kappa(H)$ of the Hessian matrix, which is given by the ratio of the maximum to minimum eigenvalues, is remarkably large.

$$\kappa(H) = \|H\| \|H^{-1}\| \tag{3.3}$$

$$\kappa(H) = \frac{|\lambda_{max}(H)|}{|\lambda_{min}(H)|} = \mathcal{O}(10^{10}) \tag{3.4}$$

This means that the optimization problem is severely ill-conditioned, and the merit function landscape is extremely elongated along particular directions around the minima. This is exactly the type of situation in which gradient-based algorithms perform badly, showing extremely slow convergence. That is the main reason why one needs to use algorithms based on second order derivative information, which take into account information about the eigenvalues of the Hessian (or an approximation of it) to properly rescale the descent direction, mitigating convergence issues.

Even if the algorithm used for optimization exploits second-derivative information (like the Trust Region Newton - Conjugate Gradient) at each iteration the state update needs to be computed by solving a linear system of equations $Ax = b$ of the form:

$$H \triangle \mathbf{x} = -\nabla f \tag{3.5}$$

or a damped version, in case of the Levenberg-Marquadt (Damped Least-Squares):

$$[H + \mu \cdot diag(H)] \triangle \mathbf{x} = -\nabla f \tag{3.6}$$

which is usually solved via iterative methods like Conjugate Gradient. The convergence properties of these iterative methods can be severely hindered when the matrix A (in our case, the Hessian) has a large condition number [63], increasing the number of iterations needed to compute the update $\triangle \mathbf{x}$ and therefore slowing down the overall speed of the optimization. As the condition number provides information about the sensitivity of the solution to perturbations in the data [64], an ill-conditioned Hessian will be sensitive to inaccurate estimations of the gradient, possibly leading to erroneous state updates $\triangle \mathbf{x}$.

Therefore, it is always beneficial to adapt the problem from the beginning to reduce the condition number in favour of a more robust and numerically stable version of itself. This is usually referred as *preconditioning* and it is commonly used to enhance the convergence properties of the iterative methods for solving linear systems of equations [65].

For illustrative purposes we defined a simple Jacobi-like preconditioner [63], which consists of a diagonal matrix stored in the form of a single vector $\mathbf{p}$. The preconditioner $\mathbf{p}$ acts as a rescaling for the state vector $\mathbf{x}$ and all its associated derivatives (gradient, Hessian-vector product...) so that the optimizer works with a preconditioned state $\mathbf{x}^*$ of approximately constant order of magnitude for each component; the desired situation for most optimization algorithms.

$$\mathbf{x}^* = \mathbf{p} * \mathbf{x} \tag{3.7}$$

$$\mathbf{x}^* \simeq \{\mathcal{O}(1), \cdots, \mathcal{O}(1)\} \tag{3.8}$$

Where $*$ represents the element-wise vector product. For the sake of simplicity, the preconditioner only uses a set of two values which are linked to the mean eigenvalues of the inverse Hessian matrix for *global* and *local* freeform terms in the RBF description, $\overline{\lambda}_{global}$ and $\overline{\lambda}_{local}$. This are computed only once from a spectral decomposition of the Hessian for a particular optical system and reduced each type a new optimization is performed.

$$\mathbf{p} \propto \{\overline{\lambda}_{global}, \cdots, \overline{\lambda}_{global}, \overline{\lambda}_{local}, \cdots, \overline{\lambda}_{local}\} \tag{3.9}$$

We compared the convergence behaviour of two optimization runs of Trust Region Newton - Conjugate Gradient with the same conditions in terms of initial state and algorithm parameters. The first run corresponds to the nominal freeform Cooke triplet and the second one was its preconditioned version using $\mathbf{p}$. The results are presented in Figure 3.9 and reveal that even with such a simple preconditioner, the number of iterations required to reach convergence can be reduced dramatically.

In addition, we checked the impact of $\mathbf{p}$ on the condition number and the spectrum of the Hessian matrix of the preconditioned problem. The use of a preconditioner $\mathbf{p}$ has two major effects. First of all, it shifts

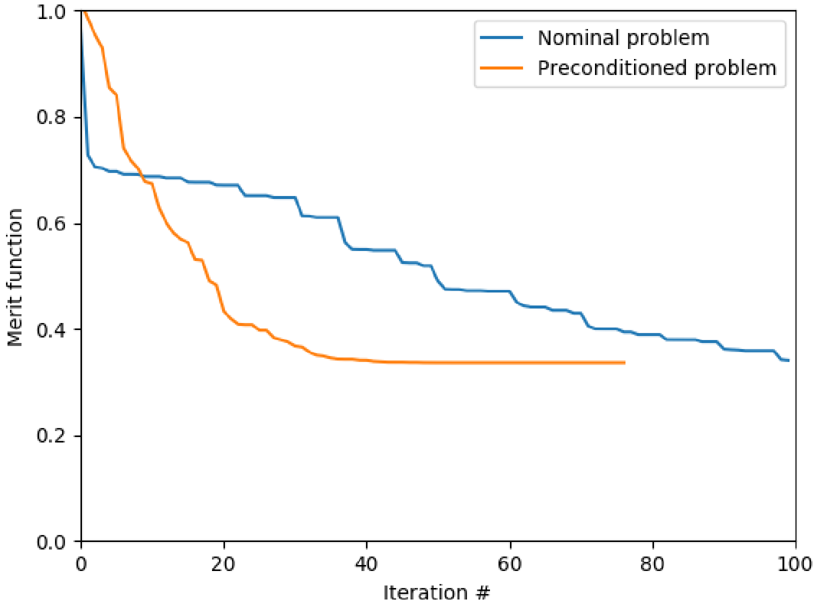


Figure 3.9: Merit function vs number of iterations for the freeform Cooke triplet in its nominal and preconditioned version. Algorithm: Trust Region Newton - Conjugate Gradient

the spectrum bringing down the maximum eigenvalue to $\mathcal{O}(1)$. Secondly, it tends to cluster the eigenvalues reducing the condition number [66]. If the initial condition number was around $\mathcal{O}(10^{10})$, the preconditioned version of the optimization problem exhibited a value of $\mathcal{O}(10^{7})$, which is quite an improvement for such a crude preconditioner.

When dealing with stiffer problems with extremely large condition numbers, it might be advisable to construct a more elaborate preconditioner, but for the applications we usually deal with, this simple preconditioner is more than enough.

3.2. BENCHMARK OF OPTIMIZATION ALGORITHMS

The use of a preconditioner to rescale the different types of variables in the state vector to approximately order one had other beneficial effects apart from speeding up the convergence of the Trust Region algorithm. It opened up the possibility of using other optimization algorithms which could not be used before, simply because the initial problem was too ill-posed to avoid divergence.

Once the problem was sufficiently regularized for most algorithms to be applicable, we decided to perform a benchmark study to characterize the behaviour of the different algorithms as far as convergence profile, number of iterations, speed and final solution are concerned.

In order to characterize how each algorithm scales with the dimensionality of the problem, the same study was done for two different configurations of the *freeform* Cooke triplet: a low-dimensional optimization with 24 degrees of freedom, and a high-dimensional optimization with a total of 120 degrees of freedom

3.2.1. LOW-DIMENSION BENCHMARK

The first part of the study relates to how the algorithms perform when dealing with a freeform optical system of relatively low dimensionality (a Cooke triplet with only 24 degrees of freedom). This number is substantially smaller than what we usually employ in freeform optics design in GDRT (between 100 and 500), but it allows us to check the behaviour of algorithms which are known to scale poorly with dimensionality.

Table 3.5: Performance results of different optimization algorithms for the *freeform* Cooke triplet with 24 degrees of freedom

Algorithm	**Iterations**	**Time [min]**	**Speed [sec / iter]**	**Merit Function**	**Stop criterion**
Conjugate Gradient	16	0.5	1.9	0.716045	*Early stop*
BFGS	260	5.0	1.2	0.370274	*Early stop*
L-BFGS	1542	30.0	1.2	0.345386	*Converged*
Newton - CG	68	15.0	13.2	0.336785	*Converged*
Trust Region N-CG	42	7.7	11.0	0.336776	*converged*

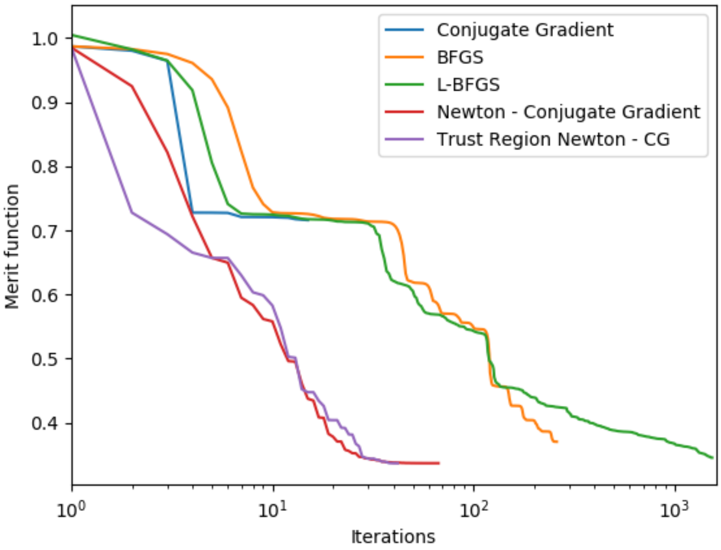


Figure 3.10: Convergence behaviour of several optimization algorithms for the low-dimensional freeform Cooke triplet

For all the algorithms we used a starting point of $\mathbf{x}^0 = \overline{0}$ which corresponds to the standard system as given by Zemax; and we defined a stop criterion of $\|\nabla f\| \leq 10^{-5}$ for the convergence. All algorithms were taken from the SciPy module.

The results for this benchmark are summarized in Table 3.5 and Figure 3.10. Despite the preconditioning, Conjugate Gradient (**CG**) performs poorly and quickly stops achieving merit function improvements. The Broyden-Fletcher–Goldfarb-Shanno algorithm, both in its standard version (**BFGS**) and its low-memory version (**L-BFGS**) perform significantly better with a similar convergence profile. The former stopped prematurely but very close to the local minimum, while the latter managed to fulfil the stop criterion with a slightly better merit function but considerably more iterations and computational time.

In contrast, the Newton - Conjugate Gradient (**N-CG**) which takes full advantage of the Hessian-vector product provided by GDRT manages to reach convergence to the local minimum in around 70 iterations. The trust region version of the algorithm, the Trust Region Newton - Conjugate Gradient (**T-N-CG**) performs even better, reaching full convergence in less than 50 iteration in under 8 minutes. Although they exhibit almost identical convergence profile, the trust region version of the **N-CG** algorithm has the advantage of being more robust, as it only "trusts" the second-order expansion of the merit function up to a certain radius, which is adapted during optimization depending on the success of the iterations. This avoids failures in the updates due to the unreliability of the expansion far away from the point of interest.

It is interesting to note that although the iterations for second-order optimization algorithms are one order of magnitude more expensive than the gradient-based or BFGS iterations, as we are only interested in global computational time t_c (which is n_{iter} times $t_{periter}$), it really pays off to use this type of algorithm. The only drawback is that the overhead, in the form of compilation time for Theano, increases when second-order is used.

Table 3.6: Performance results of different optimization algorithms for the *freeform* Cooke triplet with 120 degrees of freedom

Algorithm	**Iterations**	**Time [hours]**	**Speed [sec / iter]**	**Merit Function**	**Stop criterion**
Conjugate Gradient	2000	10	18	0.134268	*Max iter*
BFGS	594	2.1	12.8	0.113598	*Early stop*
L-BFGS	2500	9.1	13.1	0.115485	*Max iter*
Newton - CG	73	2.5	123.3	0.113692	*Converged*
Trust Region N-CG	35	1.3	133.7	0.113578	*Converged*

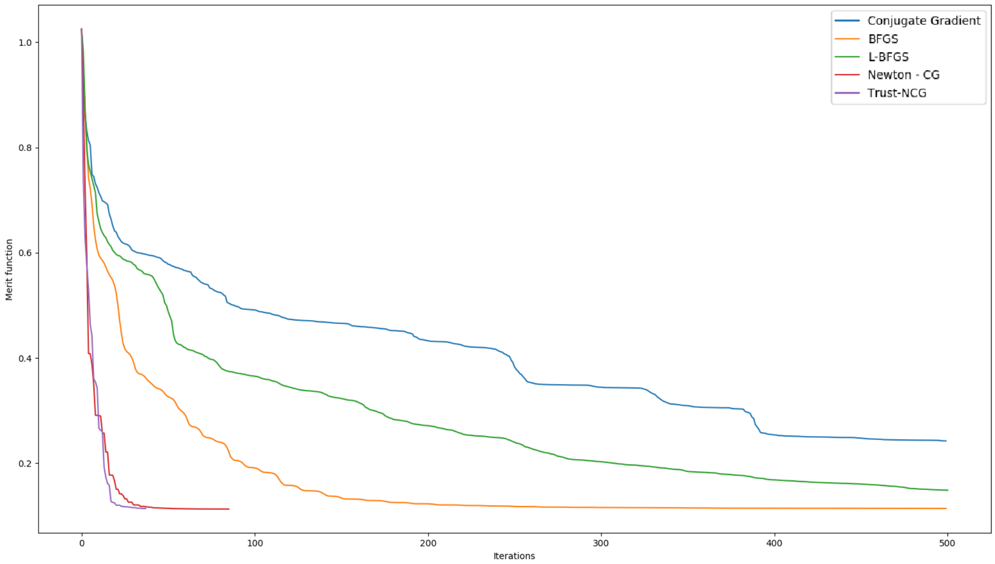


Figure 3.11: Convergence behaviour of several optimization algorithms for the high-dimensional freeform Cooke triplet

3.2.2. HIGH-DIMENSION BENCHMARK

The same analysis was repeated for a freeform Cooke triplet with a total of 120 degrees of freedom, with the same conditions as before. The results are summarized in Table 3.6. It is important to note that as the low and high dimension cases correspond to entirely different systems (in the sense of number of degrees of freedom), the local minimum occurs at different values of the merit function: around 0.336775 for the 24 dof case and around 0.1.

The results of this analysis show that there is a great difference in the convergence behaviour between first-order algorithms (Conjugate Gradient and (L)BFGS) and those which exploit the second-order information from GDRT (Newton-CG and Trust Region NCG). The former require a large number of iterations which tend to be rather cheap (around 15 seconds), while the latter manage to converge in less than 100 iterations, but each being around 10 times as costly. The low-memory version of BFGS, the L-BFGS, certainly performs quite badly and it is not recommended, taking close to 10 hours of computation.

BFGS, and Newton CG take approximately the same amount of time to converge, although the convergence profile is quite different (see Figure 3.11). Due to the disparity on the number of iterations, instead of using a logarithmic scale which would obscure the area around the Newton CG and the Trust Region NCG, Figure 3.11 has been cropped to 500 iterations.

Out of all algorithms, the Trust Region NCG is the one that provides the best performance, being the fastest both in total number of iterations and total time needed for convergence. This algorithm has proven to be the best choice in all the optimization studies we have done with GDRT so far, and thus has become the standard. Its used of second-order derivative information ensures a remarkably fast convergence, quickly improving the value of merit functions.

Table 3.7: Technical specifications of the *freeform* spectrometer

Requirement	**Value**	**Unit**
Physical requirements		
Max. volume	150	l
Numerical aperture (NA)	0.18	
Diffraction grating diameter	70	mm
Slit half-size	25	mm
Magnification	1.0	
Pixel size	20	um
Field number of pixels	2500	
Spectral requirements		
Min. wavelength	400	nm
Max. wavelength	1225	nm
Spectral sampling	7.5	nm
Number of wavelengths	110	
Dispersion	375	nm / mm
Performance requirements		
RMS spot radius	10	um
Keystone	2	um
Dispersion error	6	um
Magnification error	2	um

3.3. FREEFORM SPECTROMETER

In this section we show how GDRT can be used to optimize a realistic example of an optical system for space applications: a spectrometer. This example complements the Cooke triplet in the sense that it contains the two types of surfaces which have not been analysed yet: *reflective* and *refractive*.

3.3.1. TECHNICAL SPECIFICATIONS

We begin by presenting the requirements associated with this optical system (see Table 3.7). Some of them represent physical constraints for the system such as the maximum volume, the size of the diffraction grating (limited by manufacturing capabilities), the field of view (given by the slit dimensions) and physical characteristics of the detector.

The system is expected to operate in the Visible - Near Infrared regime and comply with the following performance requirements:

1. **RMS spot radius** shall be less than 10 μm for all wavelengths and across the whole field of view.
2. **Keystone** shall be less than 2 μm across the whole field of view
3. **Dispersion** errors shall be less than 6 μm across the whole field of view
4. **Magnification** errors shall be less than 2 μm at the end of the field of view (25 mm)

In case some clarifications regarding the definition of the requirements are needed, a thorough explanation of metrics such as RMS spot radius, keystone, dispersion and magnification can be found in section 2.8.

3.3.2. INITIAL DESIGN

The first step was to come up with an initial design entirely based on standard conic surfaces and optimize it with state-of-the-art techniques, in this case Zemax Optics Studio.

The solution was a double TMA design [67] with a diffraction grating, see 3.12. The mirrors M4 and M6 are designed in such a way that they share the same substrate which allows them to be manufactured in one

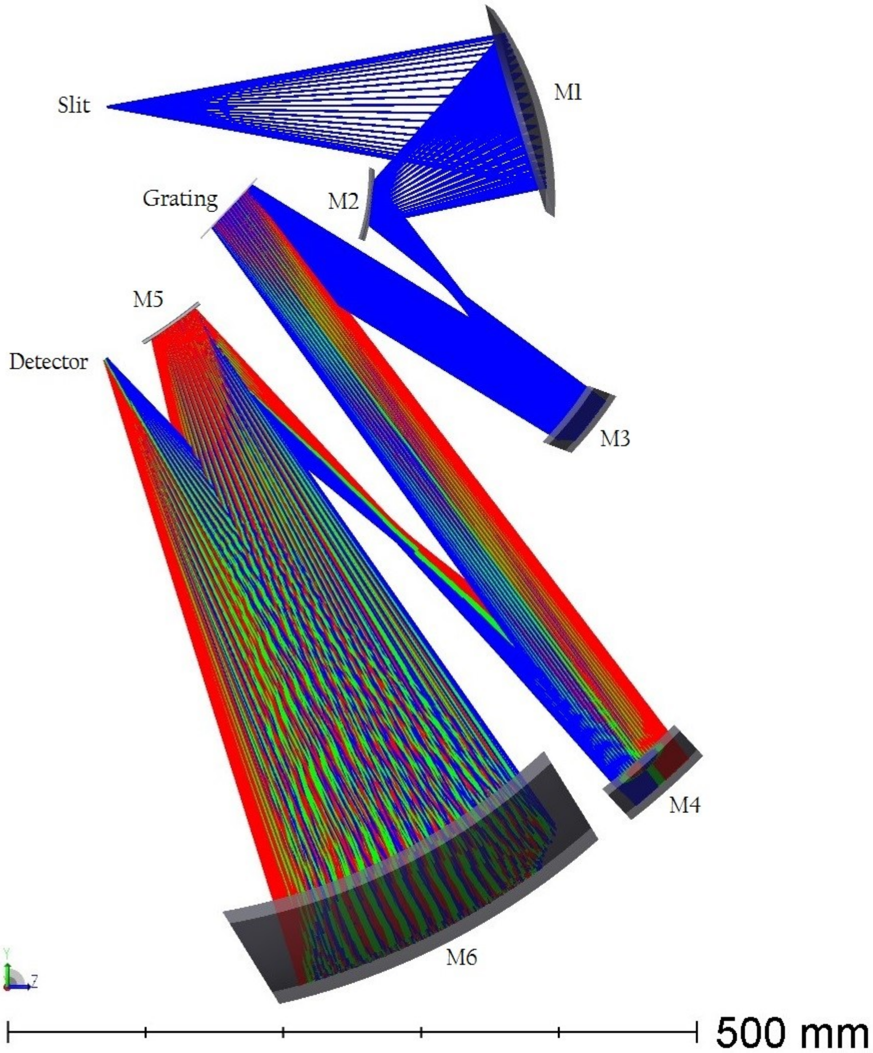


Figure 3.12: 3D layout of the spectrometer

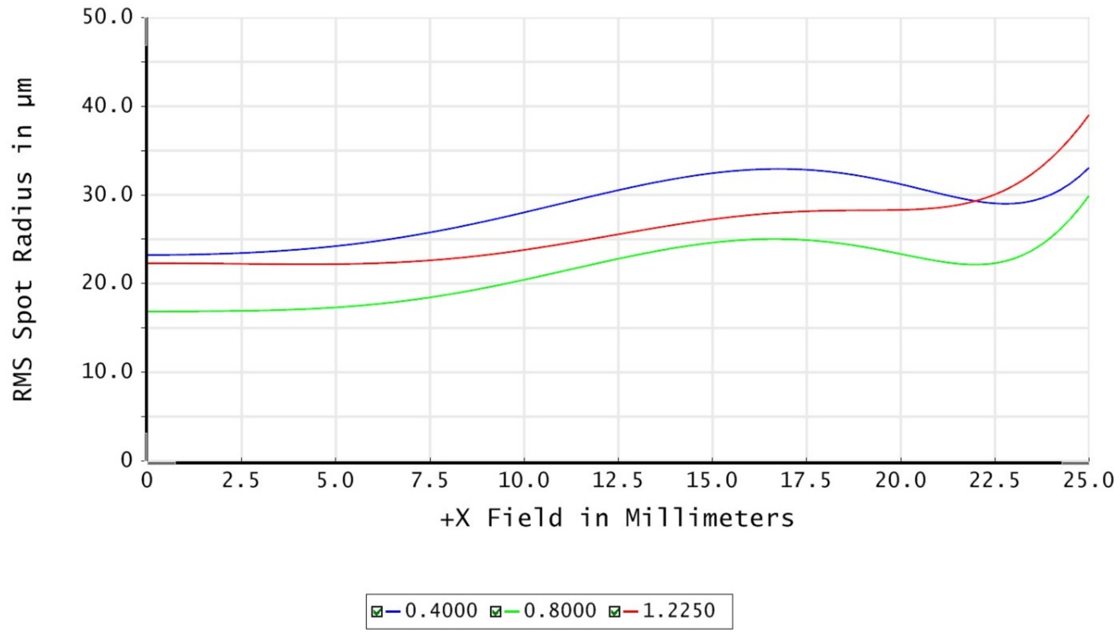


Figure 3.13: RMS spot radius as a function of field for the spectrometer

single piece with the same tool configuration. A similar design can be found in [68]. This not only reduces manufacturing costs, but also it reduces assembly, integration and alignment effort.

This design was optimized with state-of-the-art techniques in Zemax for spot size, keystone and dispersion until no further improvement in optical performance could be obtained. The final results are summarized below.

3.13 shows the evolution of RMS spot radius as a function of field of view for the three main wavelengths (minimum, central and maximum). With an average value of around 25 μm, the design does not fulfil the requirement of 10 μm by a wide margin. The situation is not better when other requirements are considered. On average, keystone is close to 4.5 μm, still way off the nominal 2.0 μm, while dispersion error is around 8 μm also higher than the intended value of 6 μm.

It is important to note that further optimization runs with Zemax no longer result in performance enhancements. With the amount of degrees of freedom considered, which correspond to the typical ones for standard conic surfaces and constant groove density diffraction gratings (curvatures, conic constants, tilts, distances, groove frequency, etc) the system seems to have reached its limit capabilities. It is at this point that the introduction of freeform optics to the system can unlock the situation, allowing the system to further improve its performance due to the addition of new degrees of freedom

3.3.3. FREEFORM DESIGN

Once we made sure the standard Zemax design could no longer be improved, we used it as starting point for a freeform optimization with GDRT. Here, we will present two distinct cases that we analysed. The first one corresponds to a design which employs **Cartesian freeform** surfaces for the mirrors and a freeform grating. The second design uses a combination of the two types of freeform mirrors: M1, M2 and M3 are defined as **polar freeforms**, while M4, M5 and M6 are left as **Cartesian**. The grating is also freeform. For details regarding these types of freeform surfaces see subsection 2.7.2.

A) CARTESIAN FREEFORM DESIGN

As mentioned above, this system uses a Cartesian freeform surface representation on top of the standard conic sags, which includes a series of global terms (in the form of XY polynomials) and local terms given by

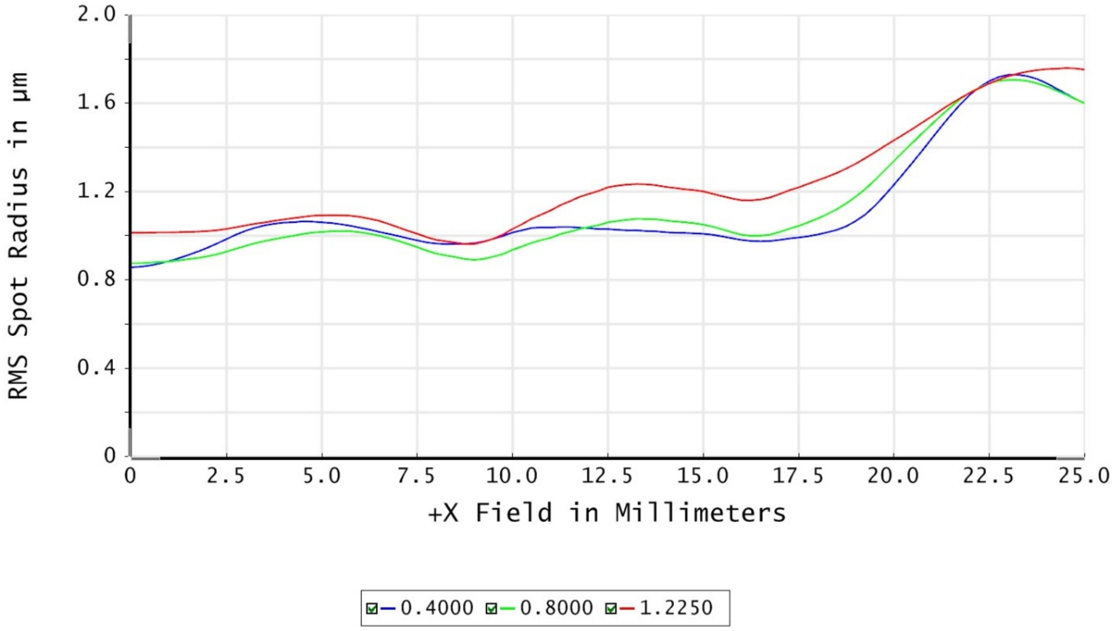


Figure 3.14: RMS spot radius as a function of field for the *freeform* spectrometer

Table 3.8: Performance comparison between the *initial* spectrometer and its optimized *freeform* variant

Metric	**Initial value [um]**	**Final value [um]**
RMS spot radius	22	1.2
Keystone	5.5	0.15
Dispersion	8	0.20
Magnification	10	0.14

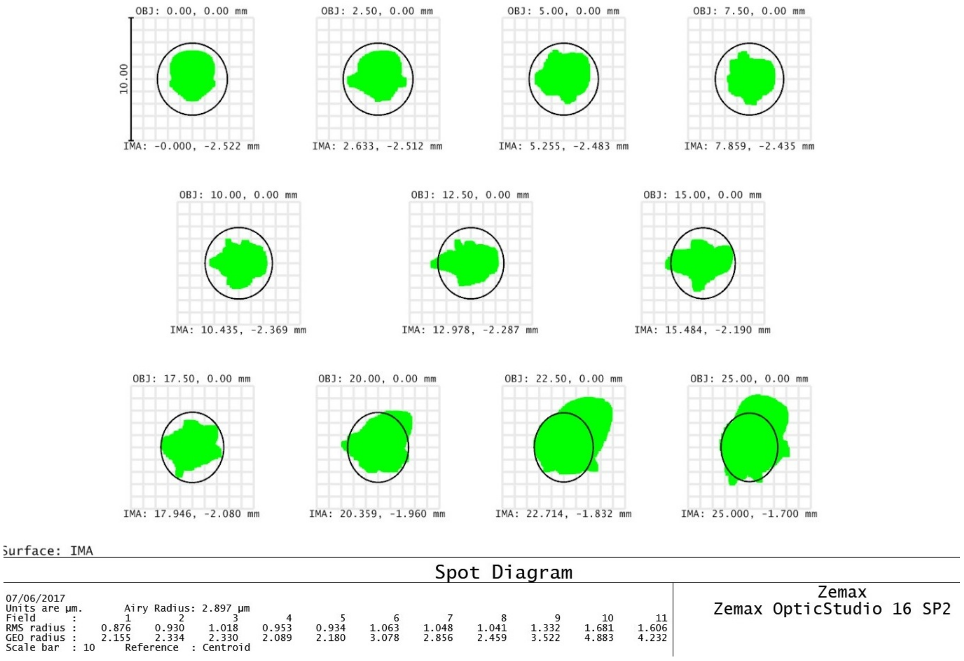
a Radial Basis Function approach. The optimizer defined in GDRT uses a trust-region Newton-Conjugate Gradient algorithm to optimize the system with respect to the weights of the freeform terms (both global an local). The input to the optimizer is, at each iteration, the **gradient** and the **Hessian-vector product** of the merit function which are obtained via differential ray tracing. A full-Hessian optimization is also possible, but it requires longer computational times and memory, providing little to no gain in total speed.

The results in terms of optical performance of the final freeform system (as given by Zemax after optimization) are presented here. Figure 3.14 shows that the RMS spot radius varies between a maximum of 1.8 and a minimum of 0.8 μm. This constitutes and average improvement with respect to the initial design of around 94% over the complete field of view and wavelength range. And a final optical performance in terms of RMS spot radius which is approximately one order of magnitude better than the nominal requirement.

It is important to note that, with these values of RMS spot radius, the system is close to being **diffraction-limited**. Figure 3.15 shows the actual spot sizes for the central wavelength (800nm) with the associated Airy disk, for different field points. It appears that the spots fall within, or are very close to, the theoretical diffraction limit marked by the Airy disk for the wide majority of the field range. Obviously, the situation is even better for the upper limit wavelength (1225nm) in the NIR regime where the system is certainly diffraction-limited; and worse at the lower limit wavelength (400nm) in the boundary of the visible regime.

In terms of keystone, dispersion and magnification the results are also quite remarkable. Table 3.8 provides a comparison between the standard Zemax design and the Cartesian freeform version of the spectrometer design.

Probably, the most illustrative thing to show is the actual shape of the **freeform mirrors** which makes this level of optical performance possible. In Figures 3.16 and 3.17 the contribution from the *global* terms of



```
07/06/2017
Units are µm.        Airy Radius: 2.897 µm
Field       :      1       2       3       4       5       6       7       8       9      10      11
RMS radius :   0.876   0.930   1.018   0.953   0.934   1.063   1.048   1.041   1.332   1.681   1.606
GEO radius :   2.155   2.334   2.330   2.089   2.180   3.078   2.856   2.459   3.522   4.883   4.232
Scale bar   : 10     Reference   : Centroid
```

Figure 3.15: RMS spot radius diagram for the central wavelength of the *freeform* spectrometer

the freeform description to the total sag of the mirrors is shown (only the values inside the actual footprint are shown). To make comparisons more meaningful, the same color bar scale was used for all mirrors. This allows us to identify which mirrors experiment the greatest changes in shape. It can be observed, that the variation in height is in the order of magnitude of 100 μm for most mirrors, which is a substantial amount for a freeform surface.

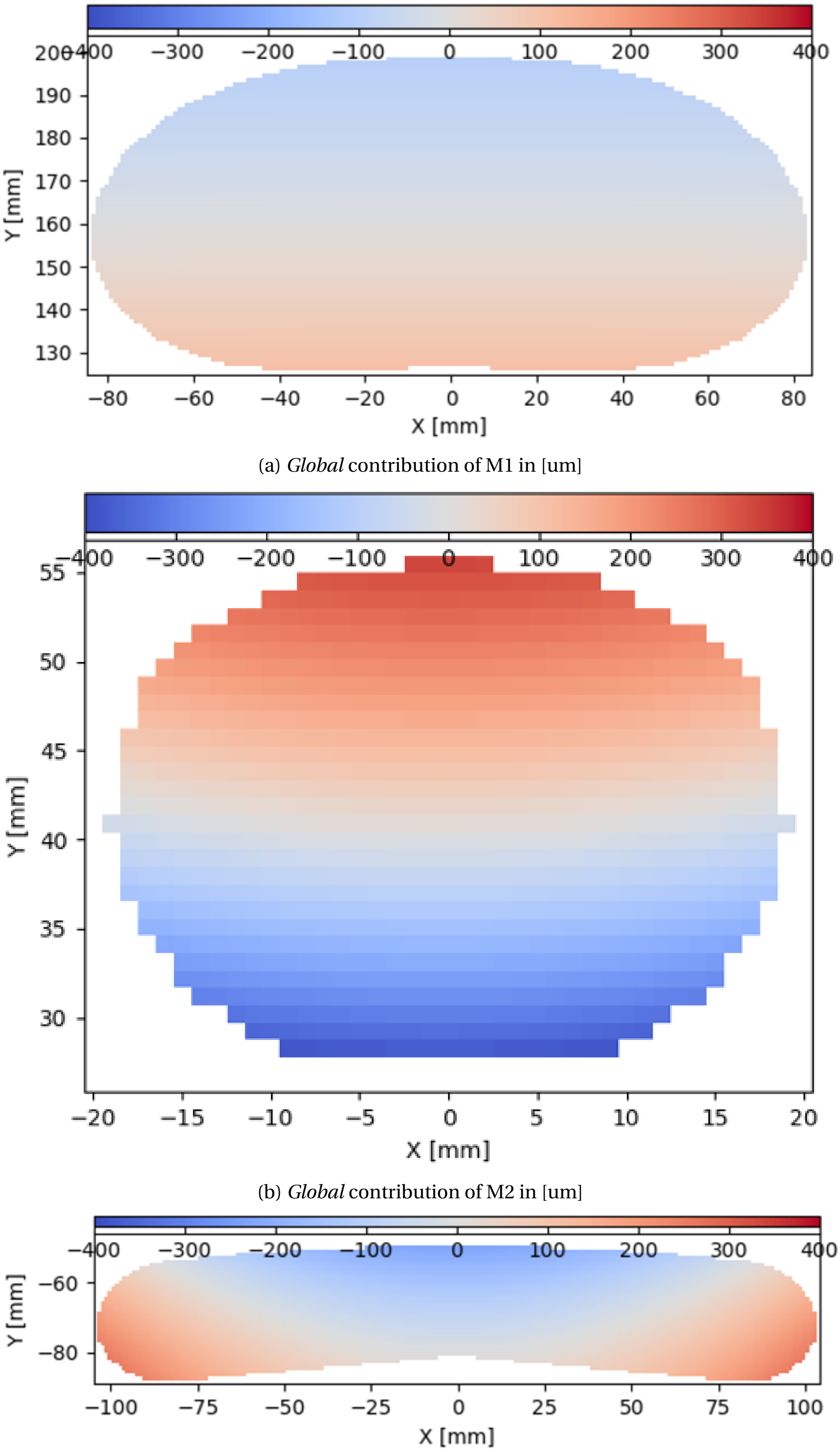
In a similar fashion, Figures 3.18 and 3.19, represent the contribution from the *local* RBF terms of the freeform model. In this case, the surface departure can be as small as a couple μm in most cases, although it can sometimes reach more than 10 μm. A 'global mirror' to 'local mirror' comparison for each surface reveals that although most of the freeform effect is handled by the global terms, there are very local behaviours which the global terms cannot model. It is at these scales that the RBF description plays a major role, imparting very specific departures in localized areas of the mirrors.

An equivalent approach can be used to study the **freeform diffraction grating**. In Figure 3.20, the contour lines of the phase function are shown for each situation: only the *global* terms, only the *local* terms and the *complete* shape. This is essentially equivalent to showing the actual grooves of the manufactured grating. Nevertheless, the distance between lines is not set to the true scale, simply for illustrative purposes.

The most important feature is that the freeform character of the grating, in the form of a variable line spacing, results in groove lines that are visibly bent along the x-direction. Obviously, just like when freeform mirrors are used, the departure from the ideal case of a constant line spacing diffraction grating is fairly small, but noticeable. The magnitude of the bending seems to be around 0.5 mm along the complete length of the grating (60 mm), which is well under 1%.

It is hard to tell at this stage if this poses a great challenge in terms of manufacturing; as that will depend a lot in the actual capabilities of the chosen provider. In any case, a recent report [58] shows that the European Space Agency (ESA) is currently involved in R& D activities to develop spectrometers based on freeform gratings; and demonstrates that it is currently possible to manufacture and characterize freeform gratings without any rotational symmetry.

At this point, it seems reasonable to address certain considerations regarding the **number of freeform surfaces** used for this case study. We decided to initially use all surfaces as freeforms in order to investigate

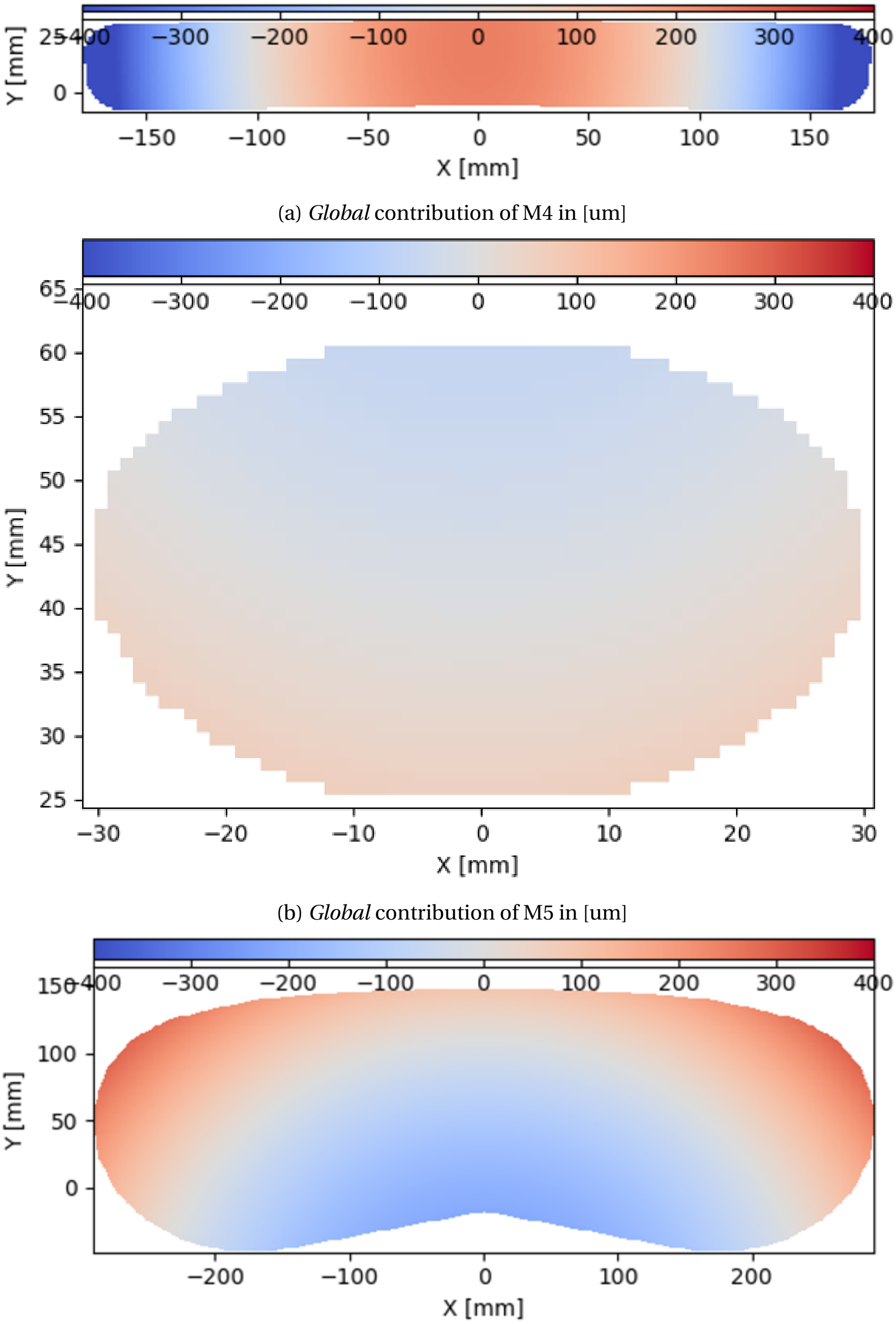


(a) *Global* contribution of M1 in [um]

(b) *Global* contribution of M2 in [um]

(c) *Global* contribution of M3 in [um]

Figure 3.16: *Global* contribution of M1, M2, M3 in [um]

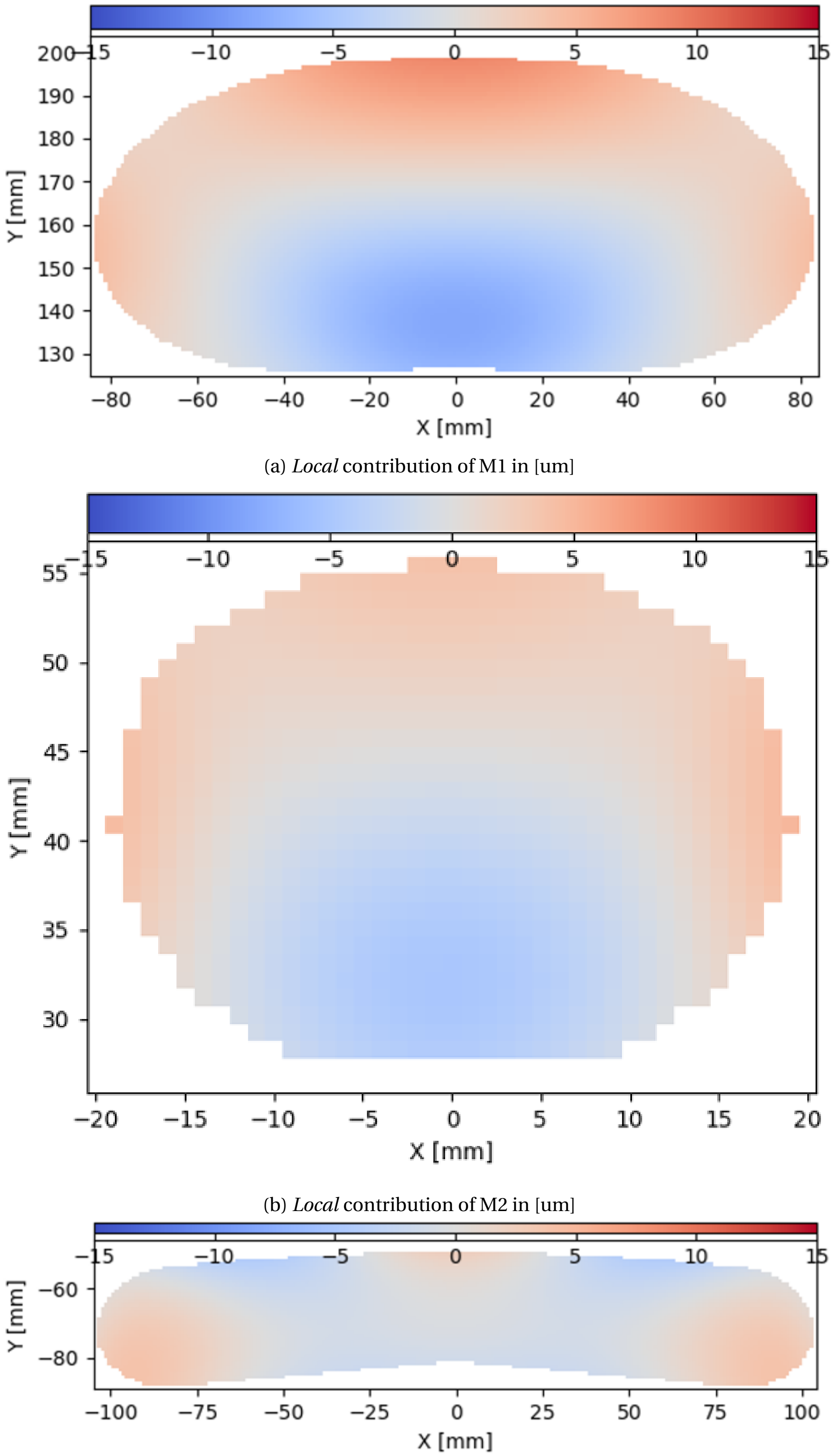


(a) *Global* contribution of M4 in [um]

(b) *Global* contribution of M5 in [um]

(c) *Global* contribution of M6 in [um]

Figure 3.17: *Global* contribution of M4, M5, M6 in [um]

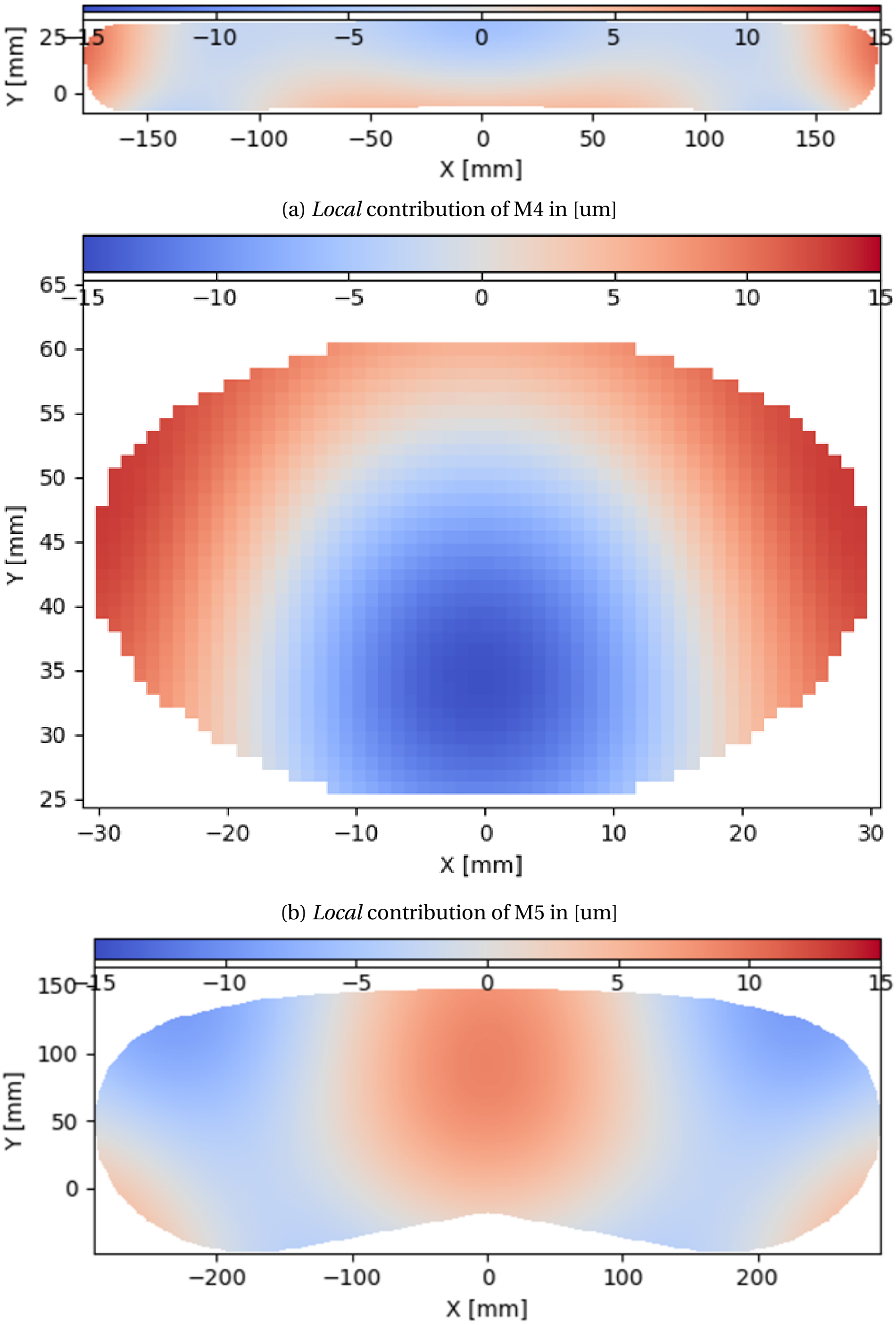


(a) *Local* contribution of M1 in [um]

(b) *Local* contribution of M2 in [um]

(c) *Local* contribution of M3 in [um]

Figure 3.18: *Local* contribution of M1, M2, M3 in [um]

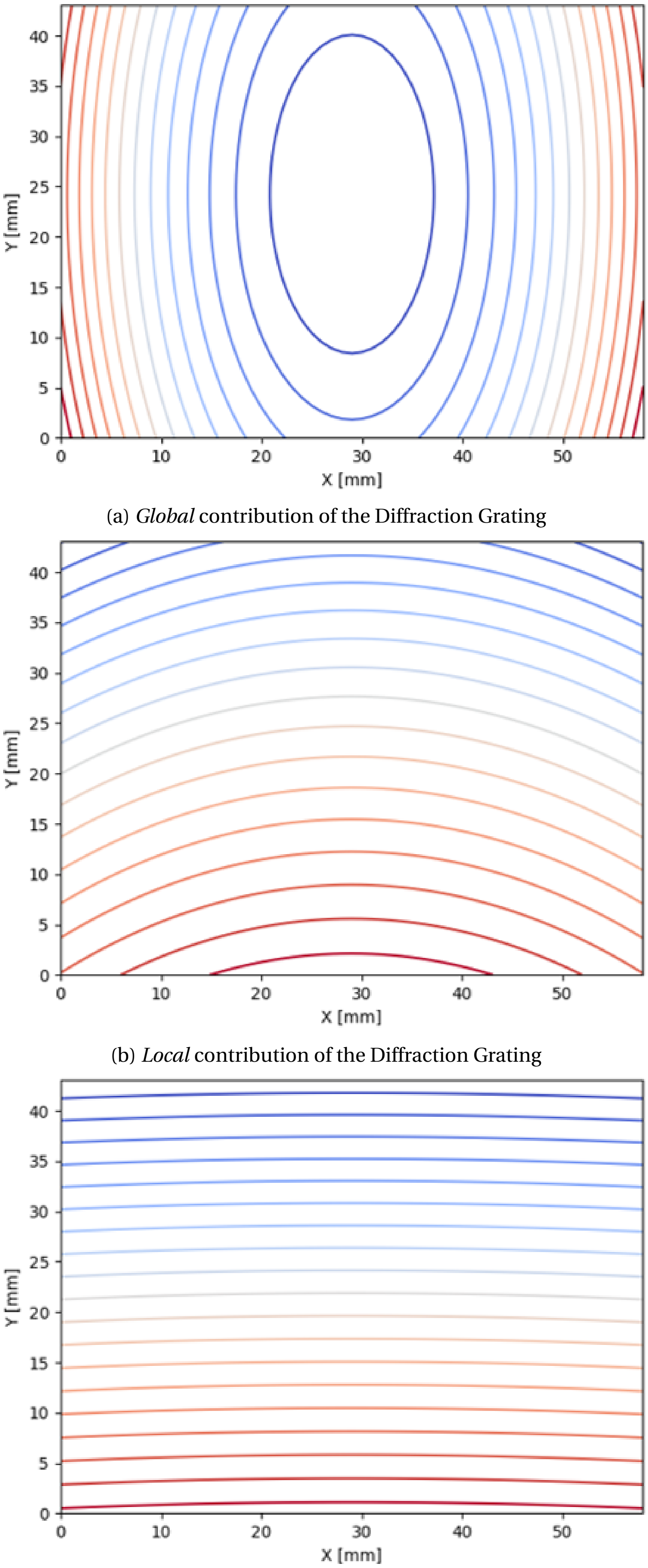


(a) *Local* contribution of M4 in [um]

(b) *Local* contribution of M5 in [um]

(c) *Local* contribution of M6 in [um]

Figure 3.19: *Local* contribution of M4, M5, M6 in [um]



(a) *Global* contribution of the Diffraction Grating

(b) *Local* contribution of the Diffraction Grating

(c) Total *freeform* contribution of the Diffraction Grating

Figure 3.20: Components of the *freeform* diffraction Grating

what a fully freeform system is capable of achieving. Nevertheless, in a realistic scenario it might not be feasible to use freeform optics on all elements of the system, for a variety of reasons:

1. **Cost** is the main driver of freeform optics. And this increase in cost with respect to their standard surfaces counterparts comes from all stages of the engineering process:
   (a) **Design & Development**. Designing freeform surfaces is significantly more time-consuming than simple conic or aspheric surfaces, which can be modelled quickly with a few parameters.
   (b) **Manufacturing**. High precision tools and longer times are usually needed to polish the surfaces to the desired level of accuracy [69].
   (c) **Metrology**. To ensure the as-manufactured surface complies with the requirements metrology is needed; but the metrology of freeform surfaces is remarkably complicated. Conventional interferometers designed for spherical surfaces are not always suitable for freeform systems because their dynamic range is insufficient to measure the departure between the spherical reference wavefront and the test wavefront [70]. More often than not, custom-made nulling subsystems are need to properly measure the freeform surface. And those systems have to be specifically designed, optimized, manufactured and assembled for each freeform surface that one wants to test.
   (d) **Assembly, Integration & Alignment**. The lack of rotational symmetry of freeform optics complicates the AIT process [11]. In addition, freeform optics can be more sensitive to performance degradation due to mechanical tolerances, increasing the demands for a fine alignment.
2. **Risk** in both the form of *technological* and *financial* risk is a key factor to take into account. A design completely based on freeform optics carries more risks than a state-of-the-art design with a single freeform surface.

Consequently, in some situations it may be necessary to make a trade-off and limit the number of freeform surfaces to only the strictly necessary. At that point an interesting question arises. It seems obvious that not all surfaces will have the same impact on the performance of the system. Therefore, we could ask ourselves whether it is possible to achieve comparable levels of performance by using only a partially freeform system. In other words, a system which combines standard conic surfaces and a few freeform surfaces where they are most needed. The results of this investigation are shown below.

Partially freeform system.

The choice of which surfaces to use is by no means straightforward. There is no general rule which will tell you how many and what surfaces will have the biggest impact on your performance because that depends on how you define performance and the particular system you are dealing with.

Some optical insight and rules of thumb suggest that the first surface after the slit (M1) and the last surface before the image plane (M6) are likely to be reasonable choices. As we mentioned earlier that M6 and M4 will be manufactured in a common substrate, there is no cost/effort penalty on also selecting M4 as freeform. The diffraction grating was also considered as freeform for this analysis.

This leaves us with a spectrometer based on 3 standard conic surfaces (M2, M3 and M5), 3 freeform surfaces (M1, M4 and M6) and a freeform diffraction grating. For the freeform surfaces, we used the same configuration in terms of number of freeform parameters as before. Several optimization runs with GDRT revealed that it is possible to achieve almost the same level of optical performance with this configuration. As far as RMS spot radius is concerned, the behaviour is essentially identical as a function of field of view, with maximum deviations with respect to the full-freeform system of around 1-2 μm. Independent studies [16] reported a similar situation in which the desired performance improvements could be obtained by only applying freeform optics to those surfaces of greatest influence on the system.

Taking into account the fact that the *full-freeform* performance was already well within the specifications, this performance degradation is considered acceptable and a wide margin for tolerances is still available. Not only is the optical performance almost the same when only the crucial surfaces are set to be *freeform*, but the time required for optimization is substantially smaller, due to the reduced amount of degrees of freedom.

In conclusion, we have shown that a *partially-freeform* design with almost the same performance as the *full-freeform* alternative, but smaller associated cost, risk and development effort is perfectly feasible. Thanks to the remaining margin in optical performance, further trade-offs could be made by restricting the system to even less freeform surfaces (possibly discard the diffraction grating due to high technological risk), if needed.

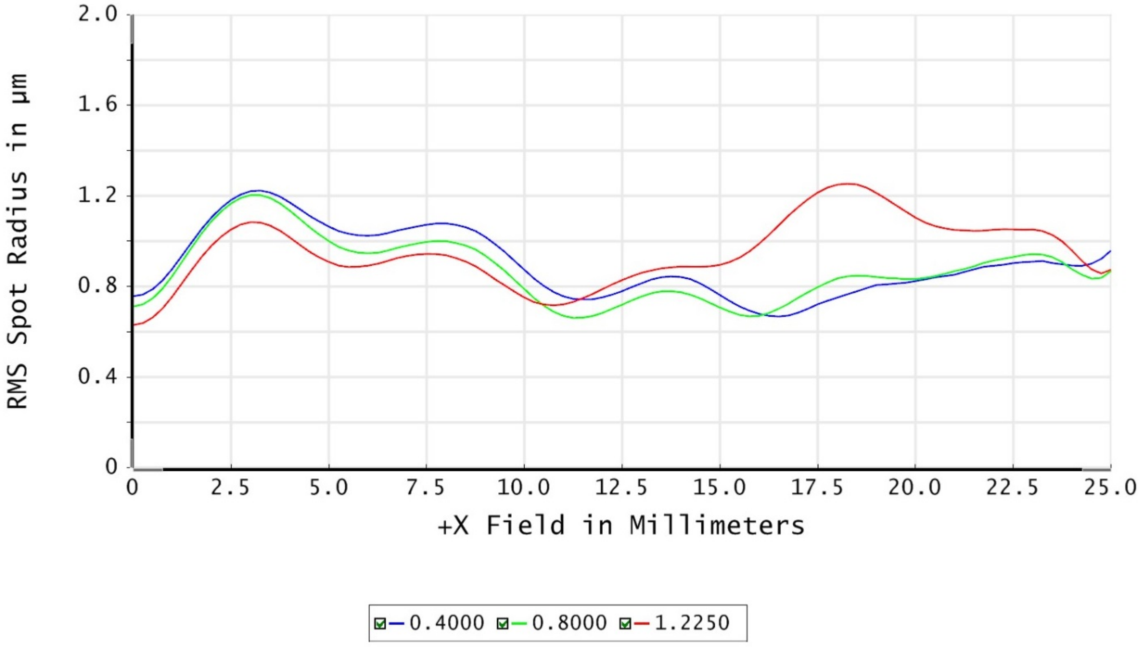


Figure 3.21: RMS spot radius as a function of field for the *polar freeform* spectrometer

Table 3.9: Performance comparison between the *initial* design, the *Cartesian* and the *polar* freeform spectrometer

Metric	**Initial [um]**	**Cartesian [um]**	**Polar [um]**
RMS spot radius	22	1.2	0.91
Keystone	5.5	0.15	0.51
Dispersion	8	0.20	0.60
Magnification	10	0.14	0.37

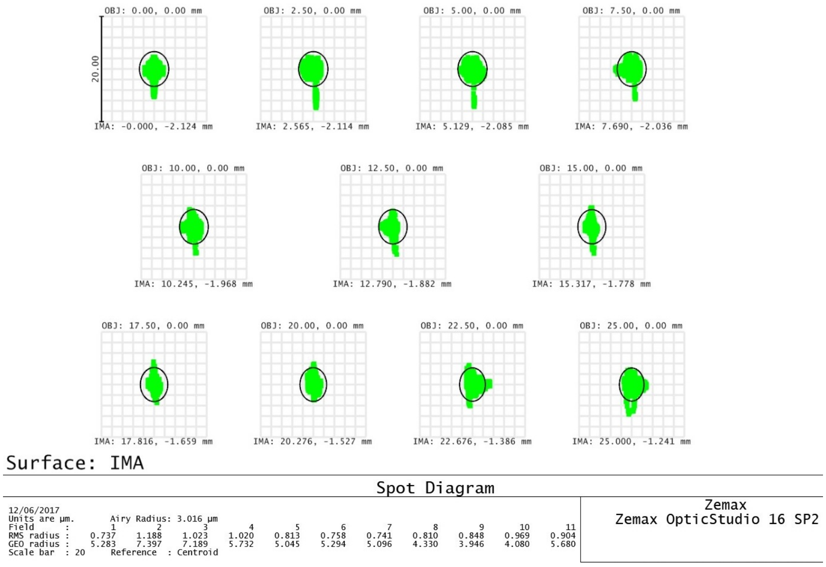
B) POLAR FREEFORM

This case study is quite similar to the *Cartesian* version of the freeform spectrometer. The only difference is that this time, the first three mirrors M1, M2 and M3 are defined using a *polar* freeform description. The remaining surfaces: the diffraction grating, M4, M5 and M6 follow the Cartesian approach for freeform surface definition used in the previous example. The results which will be presented in this section indicate that freeform surfaces with a certain polar behaviour (i.e. a certain degree of rotational symmetry) are also capable of producing high-performance designs.

As far as optical performance is concerned, the results of RMS spot radius vs field are similar to those of the Cartesian version (see Figure 3.21), varying between 0.6 and 1.2 μm for all wavelengths. Figure 3.22 shows the spot diagram for the central wavelength. Once again, the system is close to being diffraction limited at that wavelength. The only thing to note is the fact that this time the spots exhibit a slight protrusion along the y-axis whose origin could not be identified. Despite this, the RMS spot radius values are still well within the specification.

In terms of keystone, dispersion and magnification the results are similar to the Cartesian version. A complete summary of the performance levels of the *initial*, the *Cartesian* and the *polar* versions of the spectrometer is shown in Table 3.9. Both freeform versions, perfectly comply with the specifications of the spectrometer design.

For the sake of completeness, the *freeform* sags of the polar mirrors are shown in Figure 3.23. This includes both *global* and *local* terms. One should note that although the freeform surfaces are perfectly polar along the radial coordinate, as the apertures are decentered with respect to the vertex of the mirrors, the surface plots appear decentered as well.



```
12/06/2017
Units are µm.        Airy Radius: 3.016 µm
Field       :        1          2          3          4          5          6          7          8          9         10         11
RMS radius :     0.737      1.188      1.023      1.020      0.813      0.758      0.741      0.810      0.848      0.969      0.904
GEO radius :     5.283      7.397      7.189      5.732      5.045      5.294      5.096      4.330      3.946      4.080      5.680
Scale bar   : 20     Reference   : Centroid
```

Figure 3.22: Spot diagram for several field points at the central wavelength (800 nm) for the *polar freeform* spectrometer

To conclude, we present below some considerations regarding *polar* freeform surfaces:

One of the possible advantages of *polar* freeform optics is its **manufacturability**. Even though, this type of surface has a marked freeform character along the radial coordinate, it exhibits a rotational symmetry equivalent to that of standard conic surfaces. This symmetry could be exploited during the manufacturing and polishing stage to simplify the process and reduce the cost. In fact, this potential has already been identified in a recent study from this year [71], where the so-called "Near Rotational Freeform Surfaces" (NRFS) are used to facilitate the machining of freeform surfaces on brittle infra-red materials. This benefit is not applicable to Cartesian freeform surfaces, which cannot rely on symmetry.

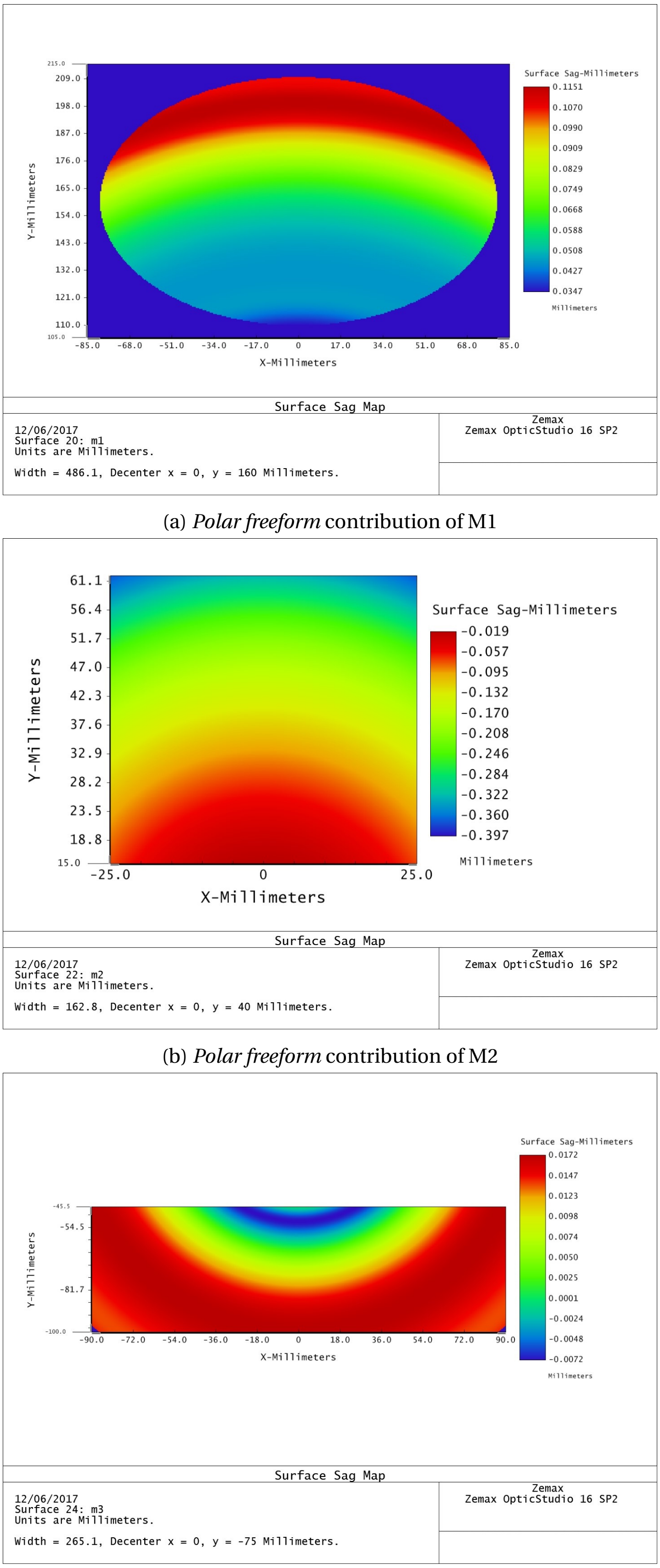
It is important to note that, just like the freeform diffraction grating, all these considerations greatly depend on the actual capabilities of a chosen manufacturer. Therefore, it is extremely useful to know what kind of surface you can expect to get easily from your particular supplier. Not all of them use the same tools and techniques and thus the results might vary. Establishing a close relationship with the manufacturer and polisher of your freeform surfaces is a key asset for any project.

Fortunately, GDRT and its freeform surface descriptor has sufficient generality to allow the implementation of the particular type of surface that best suits what some supplier might provide. Having access to that information allows you to incorporate that into the optimization process at an early stage of the project. In the end, this results in a robust design which does not suffer from a mismatch between the surfaces you have modelled, and the surfaces you are going to get. In reality this is a complicated thing to achieve for several reasons:

1. The optical designer might not fully know what the capabilities of the supplier are in terms of freeform manufacturing. Thus, it is difficult for him to incorporate that into the design.

2. It is difficult to agree upon a common surface definition. The parameters a designer uses in the optimization can be different to those a manufacturer uses during its work. If you do not define a consistent and unambiguous description, you will end up with a surface which the manufacturer says complies with the specification, but that it is completely different to what you wanted in the first place.

3. Designers normally would like to know what the supplier can manufacture before asking for a specific surface; but the supplier usually wants designers to tell them what they want to get and then answer them if that can actually be manufactured. In other words, each one expects the other to share their information first; which inevitably results in no information being shared at all.



12/06/2017
Surface 20: m1
Units are Millimeters.

Width = 486.1, Decenter x = 0, y = 160 Millimeters.

(a) *Polar freeform* contribution of M1

12/06/2017
Surface 22: m2
Units are Millimeters.

Width = 162.8, Decenter x = 0, y = 40 Millimeters.

(b) *Polar freeform* contribution of M2

12/06/2017
Surface 24: m3
Units are Millimeters.

Width = 265.1, Decenter x = 0, y = -75 Millimeters.

(c) *Polar freeform* contribution of M3

Figure 3.23: *Polar freeform* contribution of M1, M2 and M3

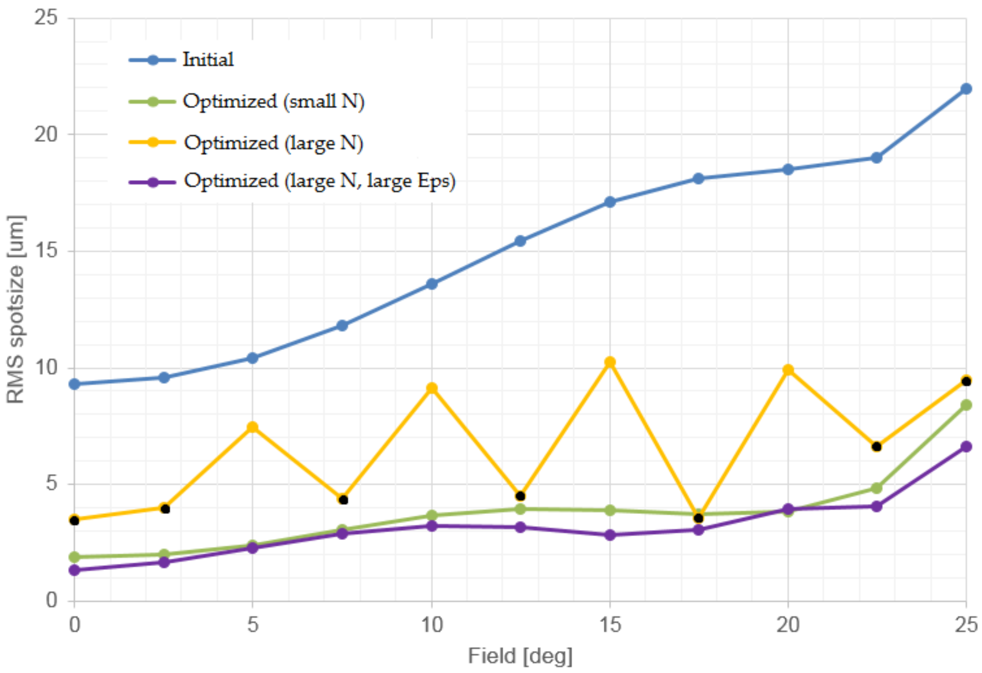


Figure 3.24: Performance results of several optimization runs of the freeform spectrometer showing some *overfitting* issues

3.3.4. THE PROBLEM OF *oversampling*

In this section we analyse an issue called *oversampling* which can appear during the optimization of freeform optics based on local methods (like RBF), and present some recommendations on how to avoid it.

Optical systems are physical systems for which their operating parameters (wavelength range, field of view, etc) are obviously continuous intervals on the real numbers. In other words, if the field of view is 20 degrees, the optical system has to maintain the desired level of performance over the infinite set of angles between 0 and 20.

But *infinite-dimensional* optimization is extremely challenging so instead, designers always optimize optical systems over a *finite* set of parameters; i.e. only a small number of discrete wavelengths and field points are used as control parameters. In most cases, no major issues will arise and the final system (which was optimized only for a discrete set) will perform well over the complete range of parameters.

Nevertheless, in certain occasions one observes that the final performance is indeed better for the discrete set of parameters used for optimization, but it is quite poor outside those points (see Figure 3.24), with substantial oscillations between control points (black dots are the control field points). This phenomenon is generally referred to as *overfitting* and is a well-known issue in many scientific disciplines including interpolation [72], statistics [73] and machine learning [74]. As seen in Figure 3.24, the existence of *overfitting* in the freeform spectrometer directly depends on N (the number of RBF nodes used per surface) and ϵ (the shape parameter of the Gaussian RBF).

This *overfitting* behaviour was observed several times during the early investigations of the freeform spectrometer optimization. The merit function representing optical performance in GDRT would be substantially reduced during optimization, suggesting significant improvements; however, once exported back to Zemax, performance analysis over the complete field of view revealed severe degradation for all field points outside the control set. The reasons behind this problem are related to the way RBF representation for optical surfaces works, and are quite easy to understand.

RBF representation in GDRT works by setting up a grid of nodes separated by a Δ which we referred to as *sampling*. At each node a Radial Basis Function is centred, whose value depends inversely on the distance to the node (the further away, the lower the value). When the RBF is a Gaussian, the area of influence of each is proportional to the shape parameter of the Gaussian ϵ. The degrees of freedom of the optimization are essentially the *weights* of the RBF nodes, meaning that the intensity of each function is adjusted according to a measure of optical performance. But that metric of performance is constructed based on the results of tracing a discrete set of rays through the different surfaces. Those rays being traced come from different positions on the *pupil*, different *field points* and different *wavelengths*. Therefore, it is obvious that for the optimizer to properly choose the values of the RBF *weights*, each RBF node needs to influence a sufficiently dense and diverse set of rays from different field points, pupil positions and wavelengths.

To better understand this, let us refer to Figure 3.25. When the sampling distance between nodes Δ is sufficiently large, the area of influence of each node (assuming the shape parameter is adjusted accordingly) is wide enough to cover the footprints of multiple rays from different field points, wavelengths and pupil positions. Consequently, when the optimizer tries to change the value of one RBF weight, it will consider the impact of that weight on the behaviour of all those rays, and no *overfitting* will appear.

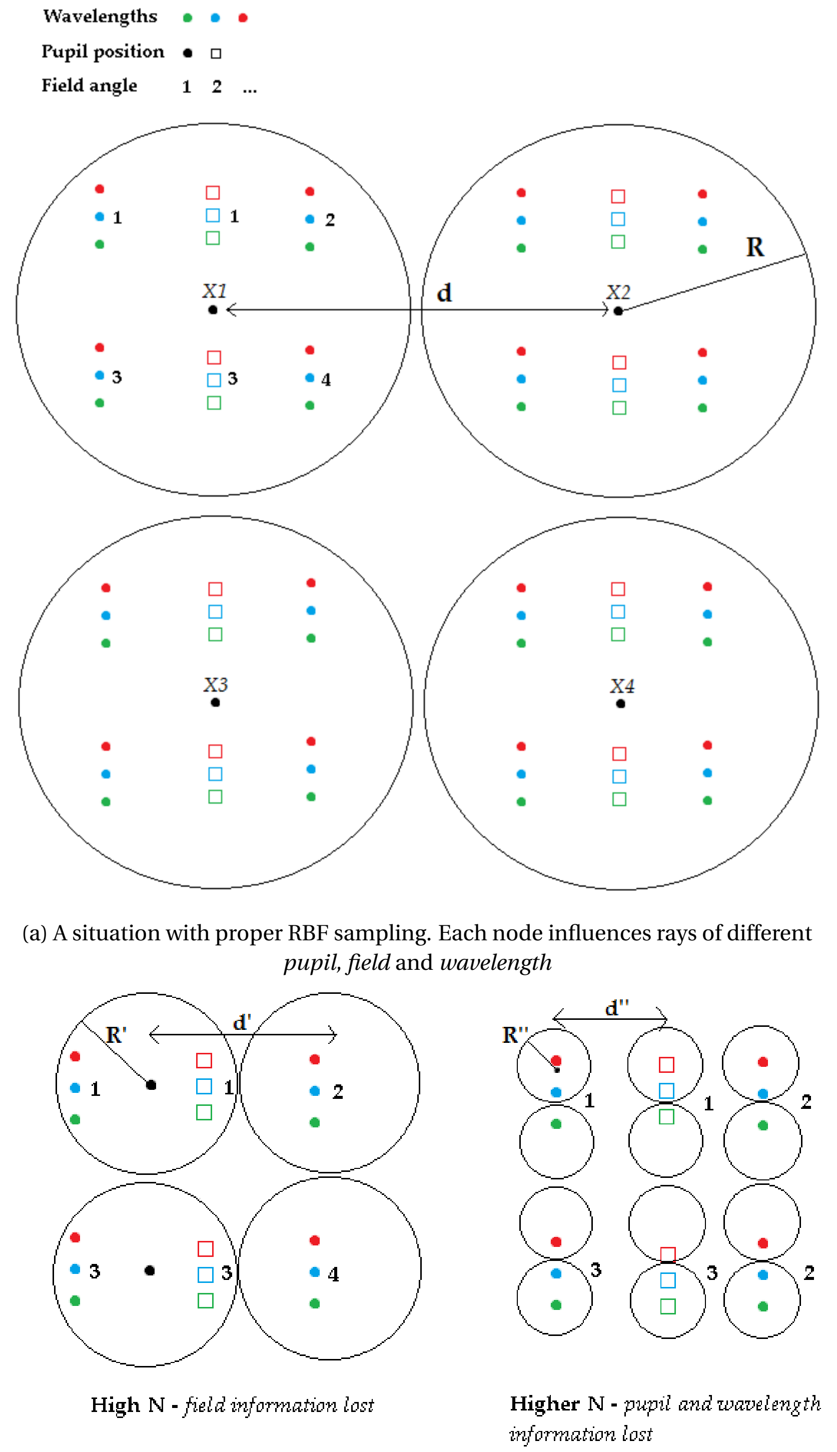
Nevertheless, when optimizing a freeform system like the freeform spectrometer it is easy to fall into the mistake of "increasing the amount of RBF nodes" by reducing Δ in an attempt to enhance the optical performance, but without modifying the ray sampling. This has the immediate effect of reducing the amount of rays influenced by each node, which can lead to a situation in which some nodes only influence rays from one particular field point, pupil position and wavelength; and in some extreme cases, no rays at all. Thus, during optimization the value of those weights will be chosen regardless of their possible impact on other rays outside the region of influence; but as we said before, the field of view, wavelength range and pupil are intrinsically continuous independently of the amount of rays traced. So in the end when the optical performance is evaluated over the continuous range, there will be *overfitting* for certain field points outside the control points.

This situation of using an excessive amount of RBF nodes without re-adjusting the pupil-field-wavelength sampling and suffering from *overfitting* is what we refer to as *oversampling* the surfaces. An immediate solution to this problem is to adjust the amount of rays being traced (increasing the number of field points, wavelengths, and pupil points) so that the density of *rays per node* is kept approximately constant and sufficiently high to avoid *overfitting*. However, this raises the issue of computational speed because it requires increasing the amount of degrees of freedom and rays being traced at the same time, dramatically increasing the time needed for optimization. Consider a ray bundle as a tensor with dimensions the pupil coordinates, the field coordinates and wavelength $R \equiv R[u, v, \theta_x, \theta_y, \lambda]$. Doubling the ray sampling in all dimensions increases the pupil points by 4, the field points by 4 and the wavelengths by 2, leading to 32 times more rays being traced. In reality, the effect is not so dramatic because wavelength overfitting is almost never an issue and a couple of wavelengths are sufficient; pupil sampling is also not so critical, in spectrometers the θ_y is not even considered and usually the planar symmetries of the system allow for reduced ray sampling. The only real threat is normally field overfitting.

We briefly mentioned earlier that the shape factor ϵ of the Gaussian RBF depends on the sampling Δ. Our investigations revealed that ϵ also has an impact on the onset of *overfitting* and has to be chosen carefully. For two Gaussian RBF nodes separated by Δ, the value of the first RBF at the second node is given by:

$$f_1(x = \Delta) = \exp(-\frac{\Delta^2}{\epsilon^2}) \tag{3.10}$$

This means that the ratio f_1/f_2 is equal to $1/2$ when the shape parameter ϵ is set to $\Delta/\sqrt{\ln 2}$ which is roughly 1.2Δ. Empiric results from our research suggest that setting the value of at least $\epsilon = 2\Delta$ is sufficient to avoid *overfitting*. The reason is that, just as having several rays influenced by each node was important, having each surface point influenced by several nodes ensures a proper choice of the RBF weights. For our particular 'rule of thumb' value of $\epsilon = 2\Delta$, the overlap between nodes leads to a ratio of $f_1/f_2 = \exp(-1/4)$ which is roughly 0.78 for adjacent nodes, and $f_1/f_3 = \exp(-1)$ which is around 0.36 second closest node located at 2Δ.


Wavelengths
Pupil position
Field angle
1 2 ...
1
1
2
3
3
4
X1
X2
X3
X4
d
R
R'
d'
R''
d''
High N - field information lost
Higher N - pupil and wavelength information lost

(a) A situation with proper RBF sampling. Each node influences rays of different *pupil, field* and *wavelength*

(b) Increasing the amount of nodes leads to some of them losing their influence over different *field* points (left) and even over different *pupil* position and *wavelengths* (right)

Figure 3.25: The impact of RBF sampling on the onset of *overfitting*

Table 3.10: Requirements for the Monolithic Thermal Infra-red imager concept

Requirement	Value	Unit
Max. volume	100 x 100 x 100	[mm] x [mm] x [mm]
Material	Germanium	
Entrance pupil diameter	90	mm
Focal length	200	mm
F number	2.2	
Field of view (fx, fy)	±2.8 x ±2.2	[deg] x [deg]
Spectral range	8 - 12	um
Pixel size	15	um
RMS spot radius	15	um

In conclusion, a careless choice of the amount of rays being sampled, the amount of RBF nodes and the shape parameters of the functions can lead to the issue of *overfitting*. We have identified the causes of this problem and we proposed strategies for avoiding it.

3.4. COMPACT THERMAL INFRA-RED TELESCOPE

In this section we present a rather different case study to show the versatility of GDRT. The freeform spectrometer was a purely reflective-diffractive system of significant size, with a spectral range in the visible and near infra-red regime. This time the system of interest is a thermal infra-red imager based on a monolithic, purely refractive concept with strict constraints in volume.

3.4.1. TECHNICAL SPECIFICATIONS

The idea for this concept was a system which could approximately fit inside a CubeSat platform, therefore the total volume is constraint to be around $100x100x100$ mm. Also, the system would be monolithic, i.e. constructed out of a single block of infra-red material; its faces acting as optical surfaces. The most relevant requirements and physical constraints for this case study are summarized in Table 3.10.

3.4.2. INITIAL DESIGN

Based on those requirements a baseline design was set up using rotationally symmetric surfaces in Zemax Optics Studio; in this case we decided to use biconics which are equivalent to a standard conic sag but with two curvatures c_x, c_y and two conic constants k_x, k_y.

$$z(x,y) = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x c_x^2 x^2 + k_y c_y^2 y^2)}} \tag{3.11}$$

The chosen optical layout can be seen in Figure 3.26. Light enters the through the pupil on the left side and travels through the block of germanium up to the first face $R1$, where it bounce back towards the lower face. After a second internal reflection at $R2$, light reaches the final face (at the top) and leaves the block while being focused onto the image plane. Thus, the optical system consists of a total of 4 active surfaces: two boundaries of the block $(S1, S2)$ and two internal reflections $(R1, R2)$.

With that in mind, the system was optimized with Zemax until the optical performance could no longer be improved. Despite extensive efforts, due to the limited amount of degrees of freedom available and the strict constraints on the overall size of the system, the performance levels could only be brought down to values above the requirements (see Figure 3.27), with a minimum of 21 microns and a maximum of more than 91 microns, when the desired value was 15 microns.

Therefore, the system was transferred into GDRT and optimized using our freeform optimization capabilities. The results are presented in the following section.

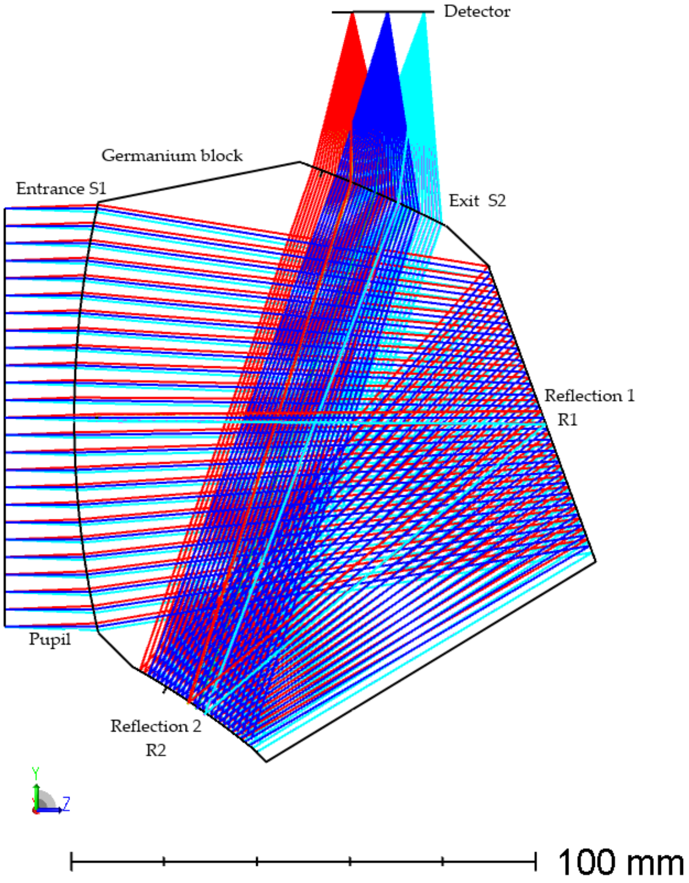


Figure 3.26: Optical layout of the Monolithic Thermal Infra-red imager concept

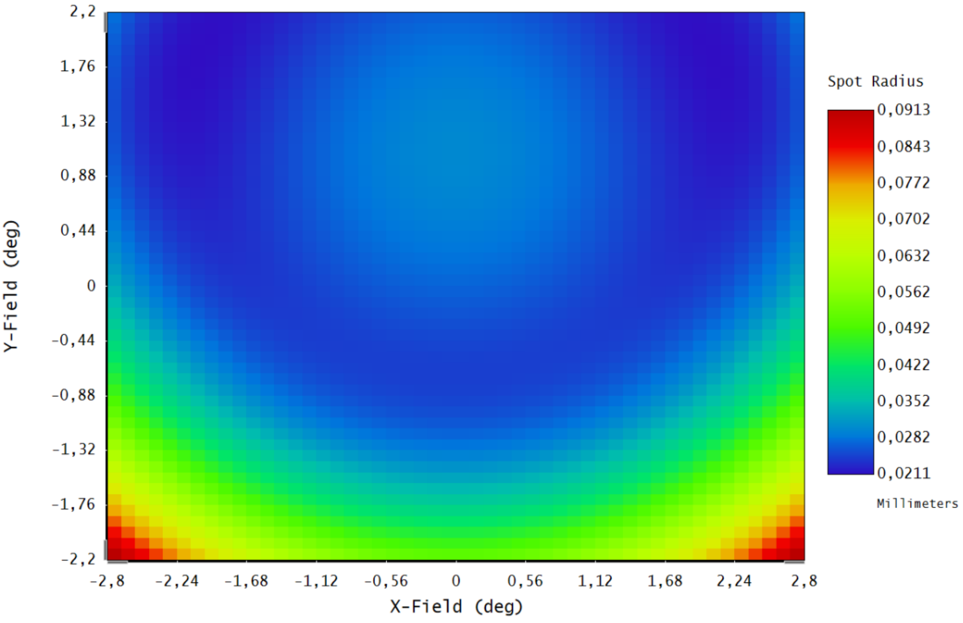


Figure 3.27: RMS spot radius field map for the rotationally symmetric version of the Monolithic Thermal Infra-red imager concept

Table 3.11: RMS spot radius comparison between the *initial* and the *freeform* version of the Compact Thermal Infra-red imager concept

Field point [deg, deg]	**(0, 0)**	**(2.8, 0.0)**	**(0.0, 2.2)**	**(2.8, 2.2)**	**(2.8, -2.2)**	**(0.0, -2.2)**
Initial system	27.18	33.80	27.02	28.24	91.94	51.06
Freeform system	12.41	13.73	11.81	17.70	17.95	14.81

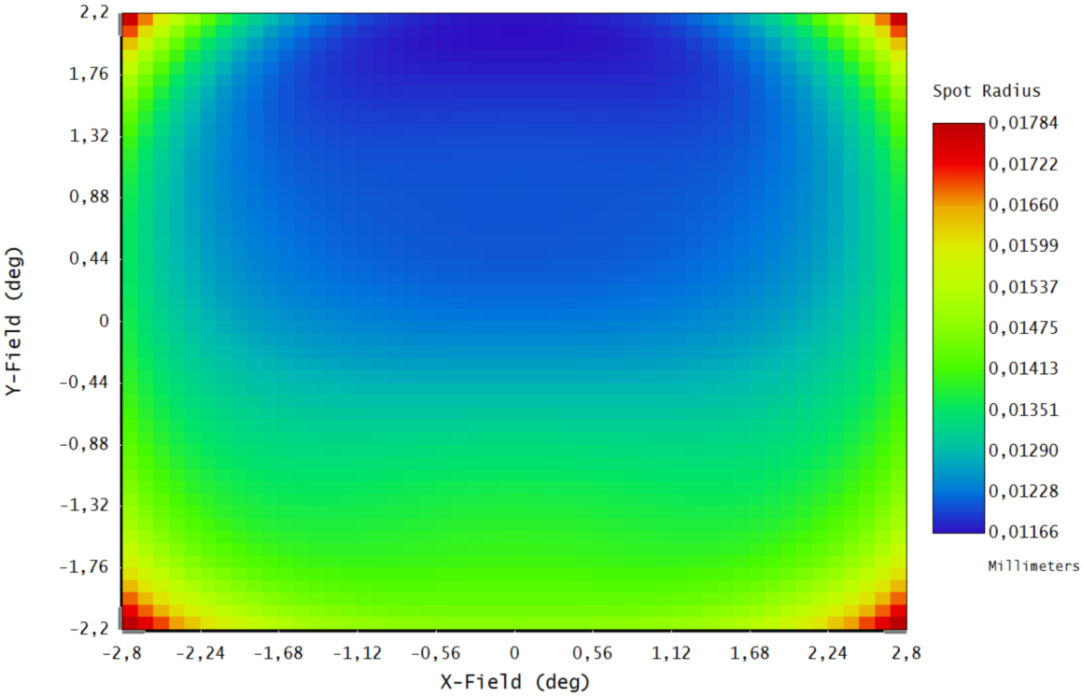


Figure 3.28: RMS spot radius field map for the freeform version of the Monolithic Thermal Infra-red imager concept

3.4.3. Freeform design

The main metric of performance for the freeform version of the Monolithic Thermal Infra-red imager is RMS spot radius. However, in contrast with the previous case studies (Cooke triplet and spectrometer) we now have constraints on the focal length and working F-number. If this is not accounted for in GDRT, the optimizer will most certainly generate a design which does not comply with those constraints. The way we solved that issue is as follows. In GDRT it is easy to constraint the distance between points on the image plane; thus we can control image size. As the field of view is given, controlling image size automatically allows for control of the focal length. In addition, as the pupil diameter is fixed, controlling focal length implies controlling F-number.

Once we included the metrics of performance and constraints into the merit function, the system was optimized inside GDRT, and then exported to Zemax for final performance evaluation. The freeform surface representation chosen in this case is the base **biconic** from the initial design plus a freeform departure consisting of **global** polynomial terms and the usual **RBF**. The amount of RBF nodes per surface is around 100 which leads to a total number of degrees of freedom close to 500. Once again due to the inherent planar symmetry of the system, the surface representation is forced to be symmetric along the x-axis.

The results after optimization are summarized in Table 3.11. Significant improvements in RMS spot radius were achieved for all field points. The final field map can be seen in Figure 3.28. Recalling the requirement of 15 μm, we can conclude that for most of the field of view the freeform system falls within the desired specifications, with slight deviations at the edges. Undoubtedly, additional efforts will be needed to bring up the optical performance to higher levels and thus allow sufficient margin for tolerances. Nevertheless, as a first order solution, this freeform system demonstrates that GDRT is a powerful tool to improve a standard design whose performance is stagnated above the requirements.

We conclude this section presenting the freeform departure of each of the surfaces in the monolithic imager, see Figure 3.29 and Figure 3.30. All surfaces exhibit a smooth freeform departure (strong influence of the global terms) in the order of 100 μm, except for the second reflection R2 which only has around 25 μm. Overall, the first two shapes resemble a standard rotationally symmetric sag; while R2 looks very much like an decentered astigmatic surface.

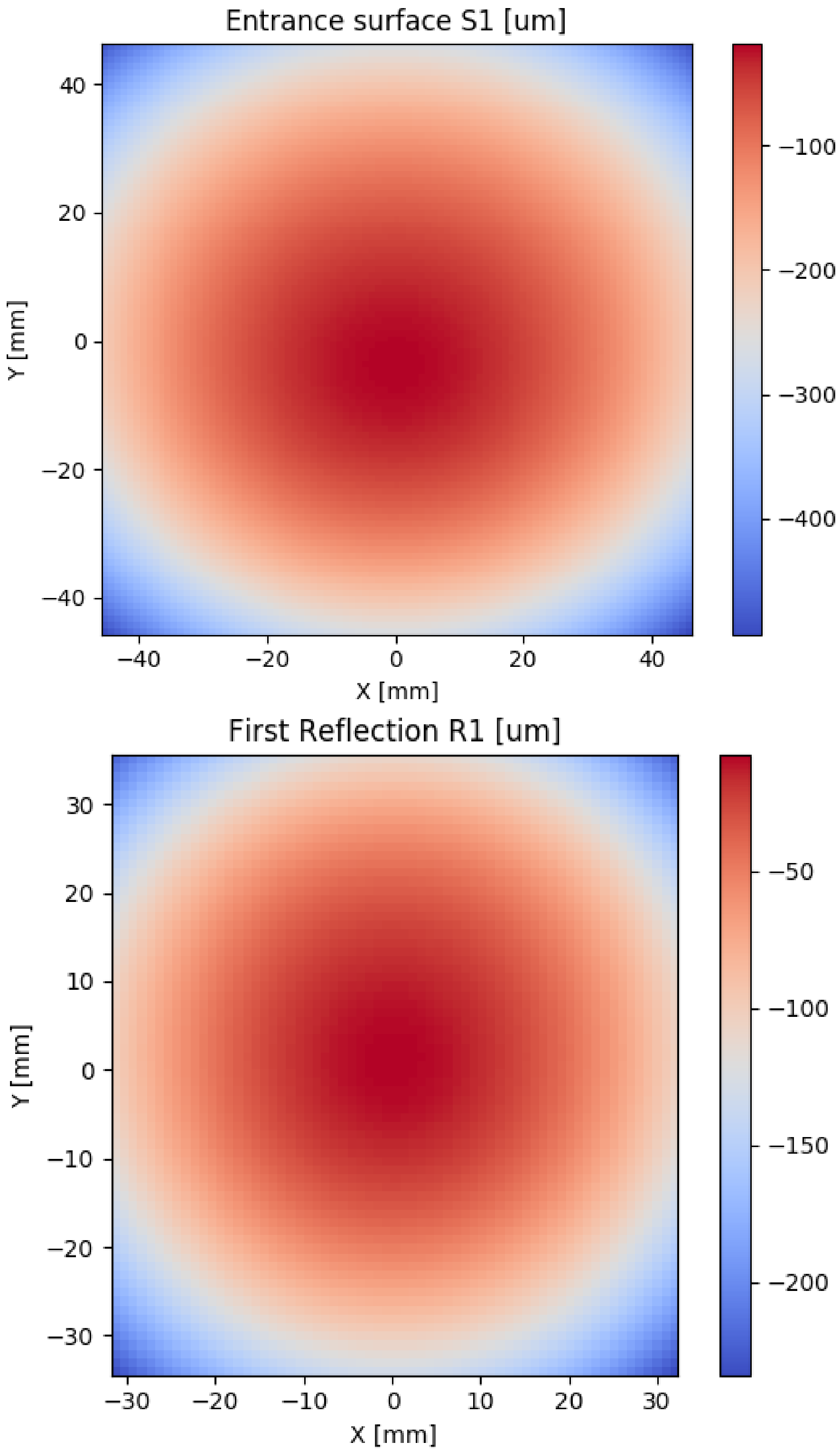


Figure 3.29: Freeform departure (**global** + **RBF**) for the first two surfaces of the *freeform* Monolithic Thermal Infra-red imager concept

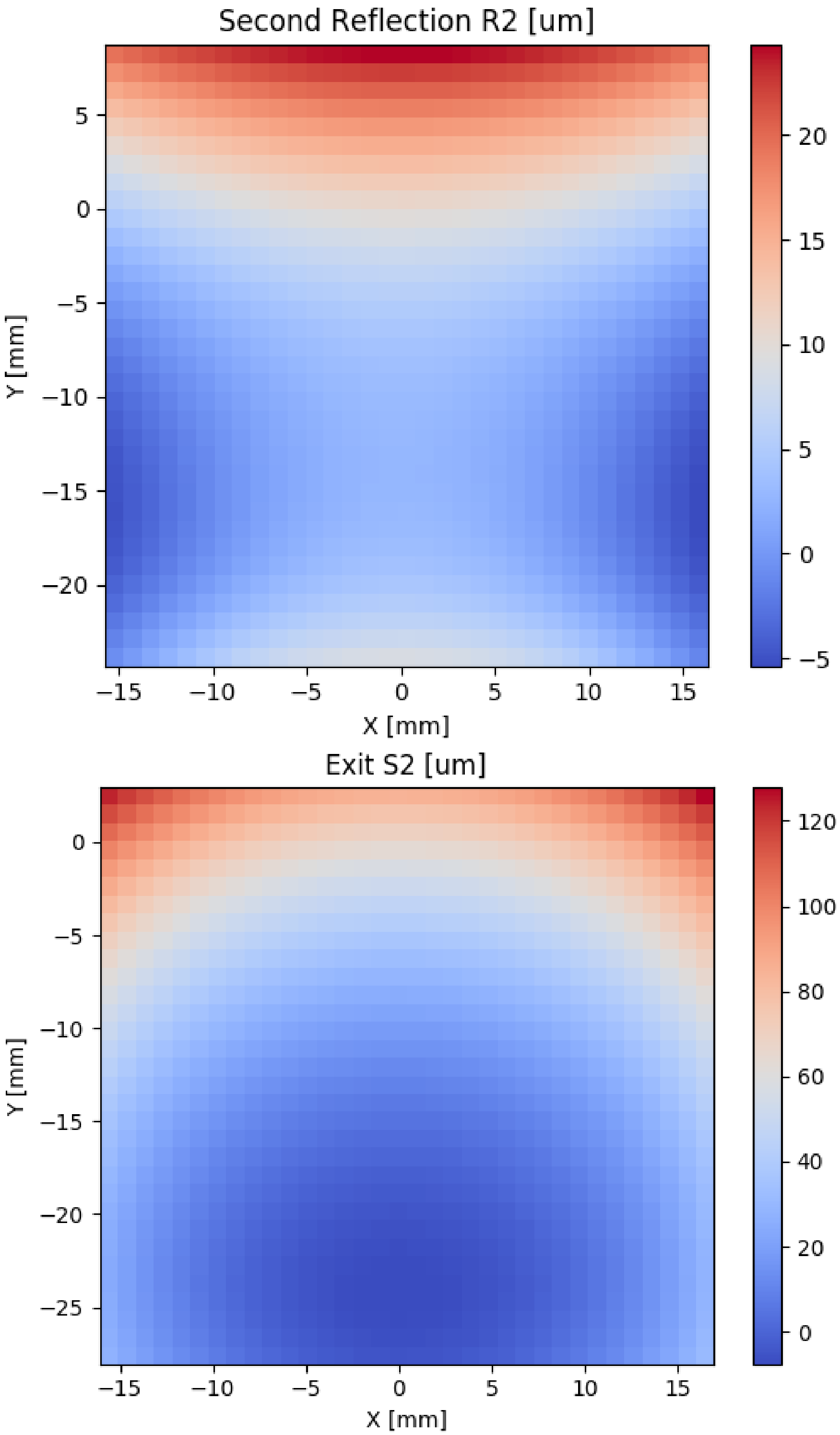


Figure 3.30: Freeform departure (**global** + **RBF**) for the last two surfaces of the *freeform* Monolithic Thermal Infra-red imager concept

Table 3.12: Main requirements and physical constraints of the LWIR imager

Requirement	Value	Unit
Entrance pupil diameter	30	mm
Field of View	10	degrees
F numbed	1.9	
Spectral range	8 - 12	um
Wavefront error @10 um	1/100	waves

3.5. Long-wave infra-red reflective imager

We conclude this chapter with another system optimized with GDRT, but from a slightly different perspective. The system in this case is a long-wave infra-red (LWIR) reflective imager which operates in the 10 μm wavelength range and follows the one presented in the landmark paper of freeform design [28]. In that study, Fuerschbach, Rolland and Thompson used nodal aberration theory and Zernike polynomials to design a freeform system substantially superior to its conic-only counterpart.

At early stages of this research (in fact, even before this Master Thesis started) we decided to use this paper as a benchmark for what it ended up being a primitive version of GDRT. This much simpler tool, despite being based on the use of Fermat's path principle for the computation of derivatives, it did not make use of the Implicit Function Theorem and differential ray tracing as GDRT now does and lacked most of its main features and capabilities. Nevertheless, after several months of research and thanks to the use of RBF surface representation, it was advanced enough to generate a freeform system which outperformed the one presented in [28].

In order to gain some insight into how the latest version of GDRT has improved and how powerful the novel approach based on differential ray tracing can be, we decided to bring back this study and re-optimize it with GDRT. The results of this investigation are summarised in the following sections.

3.5.1. Technical specifications

The LWIR system can be regarded as the reflective equivalent of the Compact Thermal Infra-red imager from the previous section, with a similar size and wavelength range but using only mirrors. The operating requirements are summarised in Table 3.12, the main metric of performance being RMS wavefront error. This is a novelty with respect to the previous systems presented in this report, where RMS spot radius was used as the main metric. Nevertheless, this is not an issue as GDRT currently includes wavefront error as a merit function available for freeform optimization.

The requirement on wavefront error of $1/100\lambda$ at 10 μm is a rather demanding one, considering the fact that the diffraction limit for this system is approximately 0.07 λ, but it is not beyond the capabilities of freeform optics.

3.5.2. Freeform design

We will not go into detail about the surface representation used by Fuerschbach as it is readily available in his paper, but in essence it is the well-known approach of combining a base conic with a series expansion of Zernike polynomials. The particular choice of which Zernike terms to use and on which surface they should be included is skilfully done using nodal aberration theory and it is well explained in [28]. An equivalent approach was followed for our particular version of the system (which we will refer to as *pre-GDRT*) during the early stages of research. This step can be done easily in Zemax to obtain a system which fulfils the wavefront error requirement of 1/100 waves at 10 mum.

After that, the **pre-GDRT** system (which includes *conic* and *Zernike* terms) was optimized outside of Zemax adding a layer of local surface representation in the form of RBF. This brought down the wavefront error even further to maximum values of less than $1/200\lambda$.

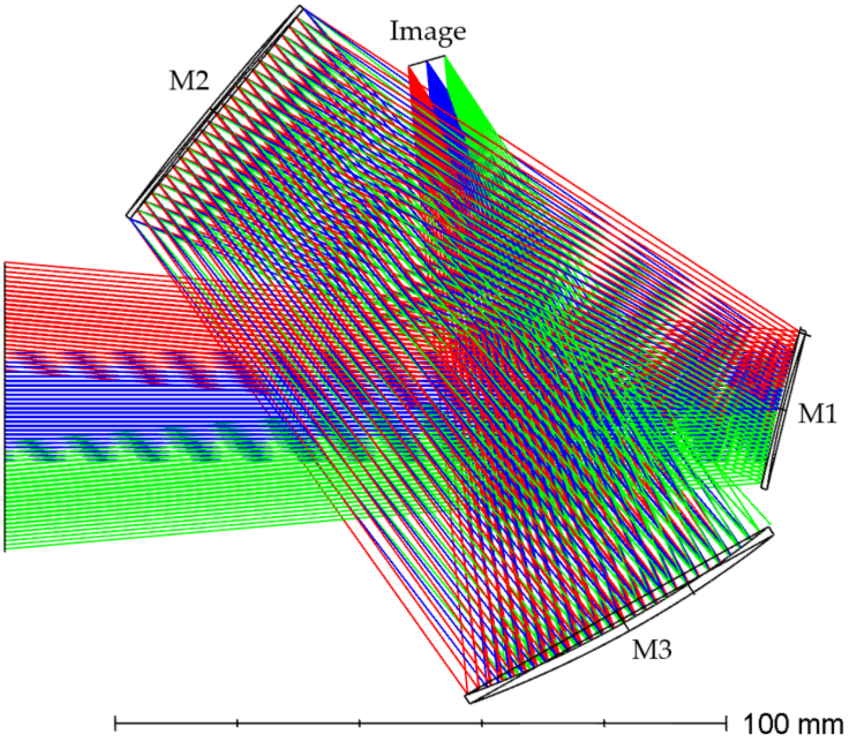


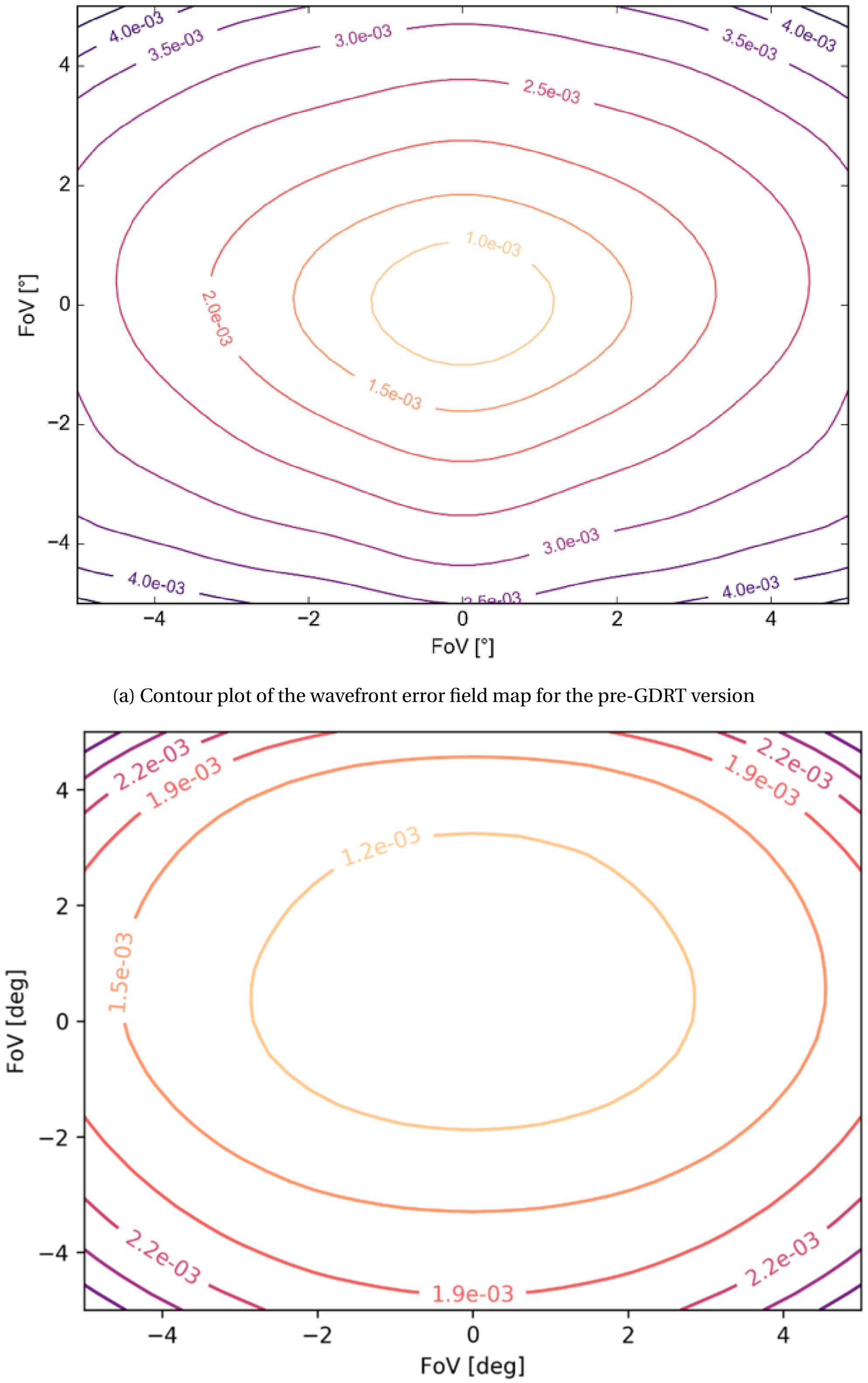
Figure 3.31: Optical layout of the *freeform* LWIR imager

The same procedure was followed but this time using the full capabilities of **GDRT** to re-optimize the Zemax freeform system. It is important to note that both optimizations (pre-GDRT and GDRT) are done only considering the RBF surface representation as degrees of freedom. The conventional parameters of the Zemax freeform system (curvatures, tilts and also the Zernike coefficients) are kept fixed. GDRT managed to bring the wavefront error down again with respect to the pre-GDRT version, to values below $1/300\lambda$ over the complete field of view. A comparison between the wavefront error field maps for both the pre-GDRT and GDRT version is shown in Figure 3.32.

To summarise these findings, during an early investigation previous to the development of GDRT we managed to improve the wavefront error performance of an optical system similar to the one shown in [28], from $1/100\lambda$ to $1/200\lambda$. In order to benchmark the capabilities of the full GDRT suite, we decided to recover this study and re-do the optimization using our differential ray tracing approach. This led to a further improvement in optical performance up to $1/300\lambda$ wavefront error.

However, the benefits of GDRT compared to the early investigations are not simply limited to optical performance. First of all, the pre-GDRT version required an extremely large amount of RBF nodes per surface to achieve its goals (in the order of 1000). This did not pose a serious problem in itself because the pre-GDRT software was considerably more primitive than the current GDRT and lacked many functionalities, which means it had more spare time to spend in dull tasks (such as RBF evaluation) before the optimization becomes too slow to be useful. Nevertheless, it is still a significant disadvantage as the size of the export surface files scales with the amount of RBF weights; and too large a file significantly slows down the evaluation of performance in Zemax.

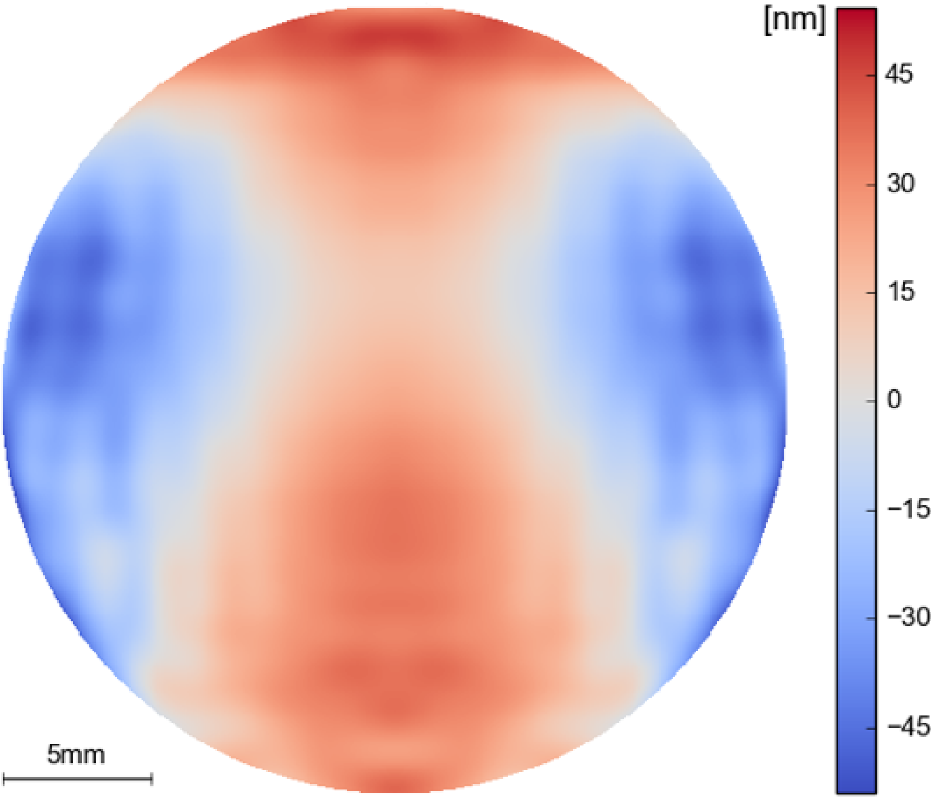
In addition, the pre-GDRT version suffered from certain *texture* issues on the optical surfaces (see Figure 3.33) which means that the freeform departures were not as smooth as expected and exhibited medium-frequency features. This is certainly a problem because medium and high frequency surface features imply complications for the manufacturing and polishing processes; and can also affect the optical performance of the system in a negative way. Fortunately, GDRT does not suffer from these issues and always generates perfectly smooth freeform departures with a reasonable amount of RBF nodes (around 100).



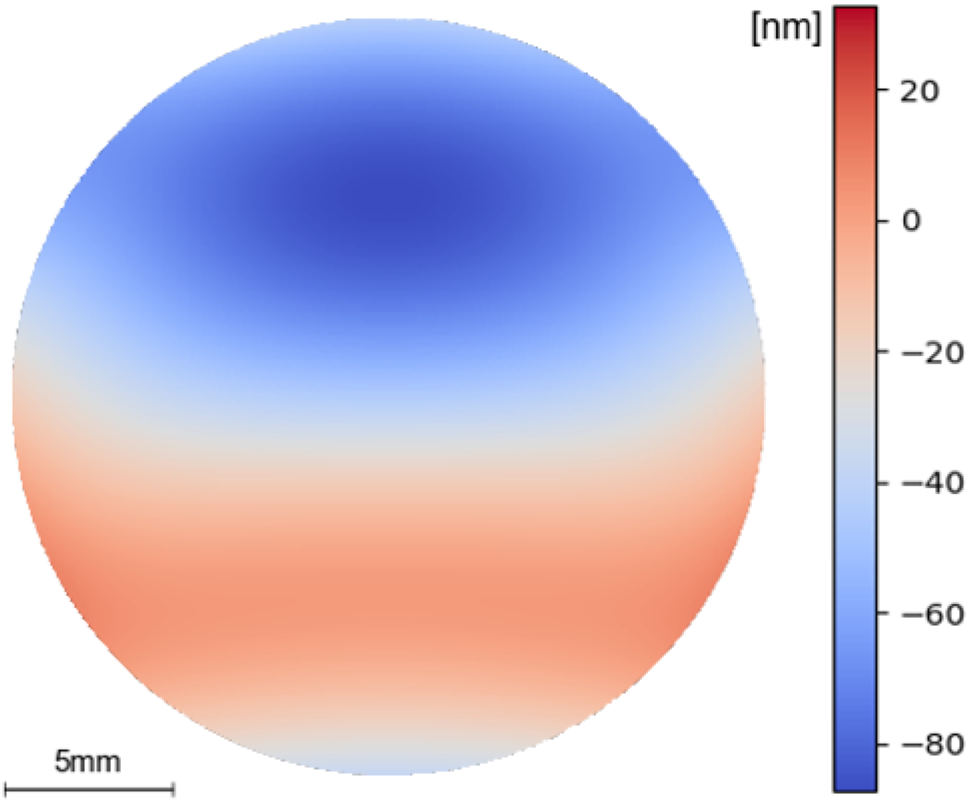
(a) Contour plot of the wavefront error field map for the pre-GDRT version

(b) Contour plot of the wavefront error field map for the GDRT version

Figure 3.32: Comparison of wavefront error between the pre-GDRT and GDRT version of the LWIR imager



(a) Freeform departure of M1 for the pre-GDRT version showing signs of non-smooth surface features



(b) Freeform departure of M1 for the GDRT version showing a perfectly smooth surface

Figure 3.33: Comparison of pre-GDRT and GDRT freeform surface departures

4

Global minima

4.1. Motivation

All our investigations so far have been centred around the search for local minima; i.e starting from a fully optimized system from Zemax, GDRT employs local optimization techniques to obtain one freeform system of enhanced optical performance. But one should note that optical design is an inherently non-linear optimization problem of high dimensionality when freeform surfaces are considered. Therefore, a wide variety of local minima are expected to exist within the merit function landscape, representing different optical systems of locally-optimum performance.

Solely from an optical designer perspective, finding a single one of those minima which fulfils the technical requirements is usually sufficient; provided that the optical system is meaningful in physical terms. But in order to gain a deeper insight into the subject of freeform optics optimization it is important to characterize the merit function landscape, at least from a qualitative point of view.

For this reason, we carried out an analysis of the merit function around the neighbourhood of a local minimum for a freeform system. We used a method to sample the merit function landscape based on the curvature information from the Hessian matrix.

4.2. Methodology

The main goal of this investigation was to characterize the landscape of a typical merit function around a freeform minimum and try to locate and study alternative minima in its surroundings. At first glance, one could think of using a *global optimization* strategy to identify the different minima. But, for the systems we have studied with GDRT, the amount of degrees of freedom n usually varies between 100 and 1000, which is well beyond the capabilities of state-of-the-art global optimizers (at least in a reasonable amount of time).

An alternative approach is to use some sort of sampling method combined with local optimization to study only a discretized set of the state space. Even so, when sampling a high dimensional space $\mathbf{x} \in \mathbb{R}^n$, in order to keep a constant sampling density, the number of sample points N scales with 2^n which grows quickly to infeasible computational times. Consequently, one needs to devise a way of making the most out of its samples by incorporating as much information as possible from the merit function landscape into its sampling strategy.

Thankfully, as we are interested in investigating the neighbourhood of a local minimum $\mathbf{x}^*$, the Hessian matrix which contains information about the curvatures of the merit function can be used to construct a *smart* sampling method. The idea is to draw a set of p samples $S = \{\mathbf{x}_1, \cdots, \mathbf{x}_p\}$ from a multivariate Gaussian distribution $\mathcal{N}_n$ with mean equal to the local minimum and covariance matrix proportional to the inverse of the Hessian at the local minimum:

$$\mu = \mathbf{x}^* \tag{4.1}$$

$$C = \sigma H^{-1}(\mathbf{x}^*) \tag{4.2}$$

$$S \backsim \mathcal{N}_n(\mu, C) \tag{4.3}$$

The use of the inverse Hessian in the covariance matrix results in sample sets elongated along directions where the curvature of the merit function is low. This has two beneficial effects:

1. Samples along those regions of low curvature are more likely to be attracted by alternative minima, increasing the chance of discovery
2. The regions of high curvature, where a local optimizer would tend to bring the state quickly back to $\mathbf{x}^*$ tend to be ignored with this method. This translates into a more efficient use of the samples, allowing to reduce the number of them and consequently saving computational time

Care should be taken to ensure that the evaluation point $\mathbf{x}^*$ is a local minimum, or at least sufficiently close to one, for the Hessian and its inverse to be positive semi-definite matrices. By definition, the covariance matrix must be positive semi-definite; otherwise one would end up trying to draw samples with negative variance.

The parameter σ allows for control of the overall size of the sample set. One should note that not all generated states might represent meaningful optical systems. In other words, although the merit function is usually defined in the complete domain of $\mathbb{R}^n$, it is restricted to states which fulfil the natural constraints imposed by Fermat's path principle. As a result of this, some states (if far away from $\mathbf{x}^*$) may cause the ray trace operation to diverge as they represent non-physical systems.

In the end, the choice of σ is driven by a trade-off. Small values lead to samples initially close to $\mathbf{x}^*$, lowering the chances of escaping its basin of attraction and of finding new minima; while large values increase the probability of generating non-physical system (i.e. wasted samples).

Once the samples have been generated, the procedure is quite straightforward. For each sample, a local optimization run is performed until a new optimum state is reached. A post-processing stage will take care of analysing how the sample set has evolved in order to answer the following questions:

1. Have alternative minima been found?
2. If so, how do they look like when compared to the nominal minimum $\mathbf{x}^*$?
3. If not, how does the merit function around $\mathbf{x}^*$ look like?

4.3. HESSIAN INVERSE

Although GDRT has second order derivative capabilities, obtaining the Hessian and inverting it can be too computationally challenging in terms of time and memory. In contrast with a standard optimization task where accuracy plays a key role, for this sort of investigation we only need the inverse Hessian to get a broad idea of the curvatures and generate sample sets. Having the exact values up to machine precision will not ensure that the samples we draw will lead to new minima, and the computation of p local optimization runs already takes a significant amount of time. Therefore, for this task we are content with using the following fast approximation of the inverse Hessian.

For a generic Linear Operator A in $\mathbb{R}^n \to \mathbb{R}^n$, the Python module SciPy can reconstruct a partial eigendecomposition up to $(n-2)$ if provided with its right and left hand side vector products methods $A \cdot v$ and $v^T \cdot A$. For the case of the Hessian matrix, GDRT has fast and easy access to the Hessian vector product $H \cdot v$ so we can obtain the $(n-2)$ highest eigenvalues $[\Lambda]_*$ and their associated eigenvectors $[U]_*$.

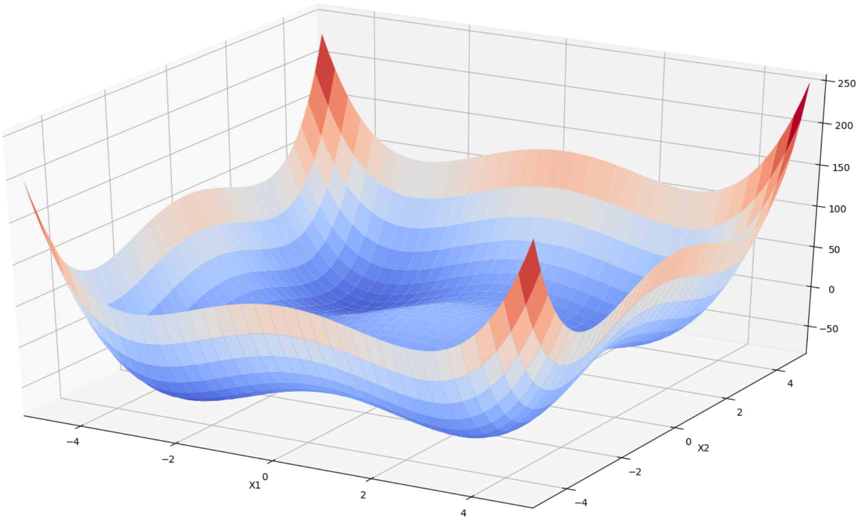


Figure 4.1: Styblinski-Tang function in 2 dimensions

$$H \equiv LinearOperator(H \cdot v) \tag{4.4}$$

$$[\Lambda]_*, [U]_* = eigen(H) \tag{4.5}$$

So that:

$$Hu_i = \lambda_i u_i \tag{4.6}$$

The only drawback of this decomposition is that it is incomplete, meaning that the last two eigenvalues and eigenvectors are missing. However, based on previous knowledge about the Hessian spectrum we know that those two eigenvalues will be many orders of magnitude smaller than λ_1 and very close to the last one available λ_{n-3}. Consequently, we can complete the decomposition by cloning λ_{n-3} and adding two additional linearly independent eigenvectors without degrading the accuracy of the approximation. A Gram-Schmidt orthonormalization process can be used to generate the additional eigenvectors.

Now, based on the extended decomposition $[\Lambda], [U]$ the approximate inverse Hessian can be recovered using the following formula. As $[\Lambda]$ is a diagonal matrix, its inverse can be computed immediately by taking the reciprocal of the eigenvalues λ_i:

$$H^{-1} \simeq U\Lambda^{-1}U^T \tag{4.7}$$

4.4. 2D EXAMPLE

In order to illustrate the idea behind this method, before studying a high-dimensional scenario, we will present first a simple 2D example which allows for direct visualization. Consider the Styblinski-Tang function for a generic dimensionality d:

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i}^{d}(x_i^4 - 16x_i^2 + 5x_i) \tag{4.8}$$

When restricted to 2 dimensions this function has 4 local minima in the $[-5,5] \times [-5,5]$ domain, one of them constitutes the global minimum located at $x^* = (-2.903534, -2.903534)$. The function landscape can be seen in Figure 4.1.

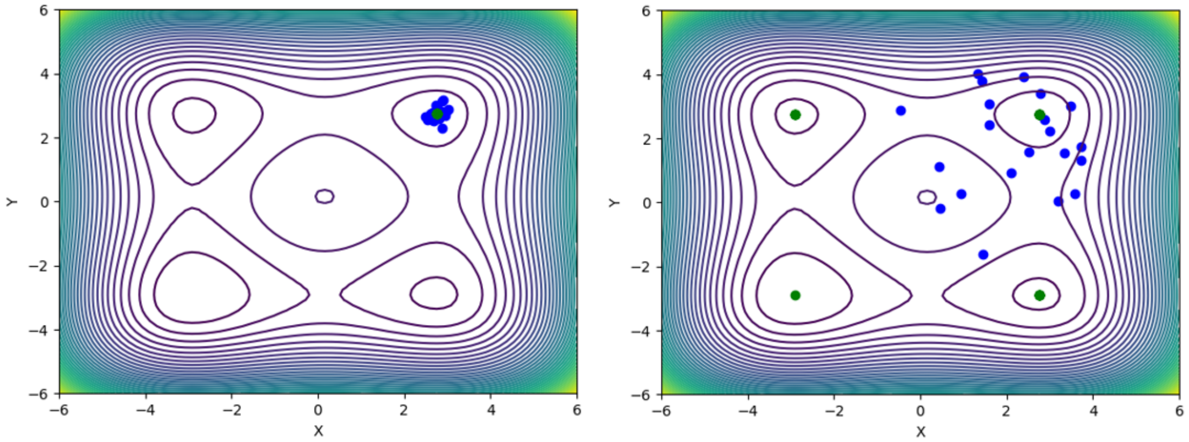


Figure 4.2: Results of the global minima method for $\sigma = 1$ (*left*) and $\sigma = 100$ (*right*) for the Styblinski-Tang function

Judging from the figure, it is easy to understand why without knowing anything about the function landscape (the natural situation for a high dimensional optimization problem), one might end up always find the same local minimum, if the starting points are within the basin of attraction of that particular minimum.

For this example problem we used the inverse Hessian matrix, which can be computed analytically, to set up the covariance matrix and draw 25 samples around the local minimum opposite to the global one. We ran local optimization procedures for each sample and reported the final optimum states. The results are shown in Figure 4.2 for two different values of initial sample set size. The starting sample states are shown in colour *blue* while the final optimized results are shown in *green*.

At this point several conclusions can be drawn:

1. The method we proposed is capable of identifying new local minima in the neighbourhood of the nominal one used to draw the samples (see the green points in Figure 4.2)
2. Incorporating curvature information from the Hessian into the covariance matrix enhances the effectiveness of the sampling. The steep areas of high curvature need not be sampled because there is no chance of not falling back towards the original minimum. Therefore, the samples tend be shifted towards more interesting areas of low curvature where bifurcations towards other minima might exist
3. The success of this method in finding additional minima depends on a proper choice of the sample size σ. When σ is too small, all the samples will start on the basin of attraction of the nominal local minimum and will not produce new findings.

4.5. RESULTS

For this investigation, we used a relatively low-dimensional version of the *freeform* Cooke triplet with 24 degrees of freedom to reduce the total computational time down to a reasonable value (p local optimization runs in a high dimensional space can be quite demanding). The system was *preconditioned* in accordance with the method shown in subsection 3.1.3. Based on the procedure described in section 4.2, the search space was sampled using a set of $p = 100$ randomly generated samples with covariance matrix proportional to the Hessian inverse at the local minimum $C = \sigma H^{-1}(\mathbf{x}^*)$. An initial investigation for $\sigma = 0.1$ did not yield new local minima. All the samples either failed to convergence during local optimization, or converged to the initial minimum $\mathbf{x}^*$.

Therefore, the same procedure was repeated for a higher value of set size $\sigma = 1.0$. This time, careful analysis of the convergence behaviour for each sample revealed that, although most samples converged back to $\mathbf{x}^*$, some appeared to have converged to states of slightly higher merit function then the nominal one,

Table 4.1: RMS spot radius performance and GDRT merit function values for the freeform Cooke triplet associated with $\mathbf{x}^*$ and the two minima candidates

	System of interest		
Field Point [deg]	**Nominal Minimum**	**Candidate 1**	**Candidate 2**
[0, 0]	4.729	5.059	4.185
[0, 14]	6.182	6.354	7.645
[14, 14]	9.266	7.793	15.777
GDRT Merit Function	0.336775	0.547606	0.723034

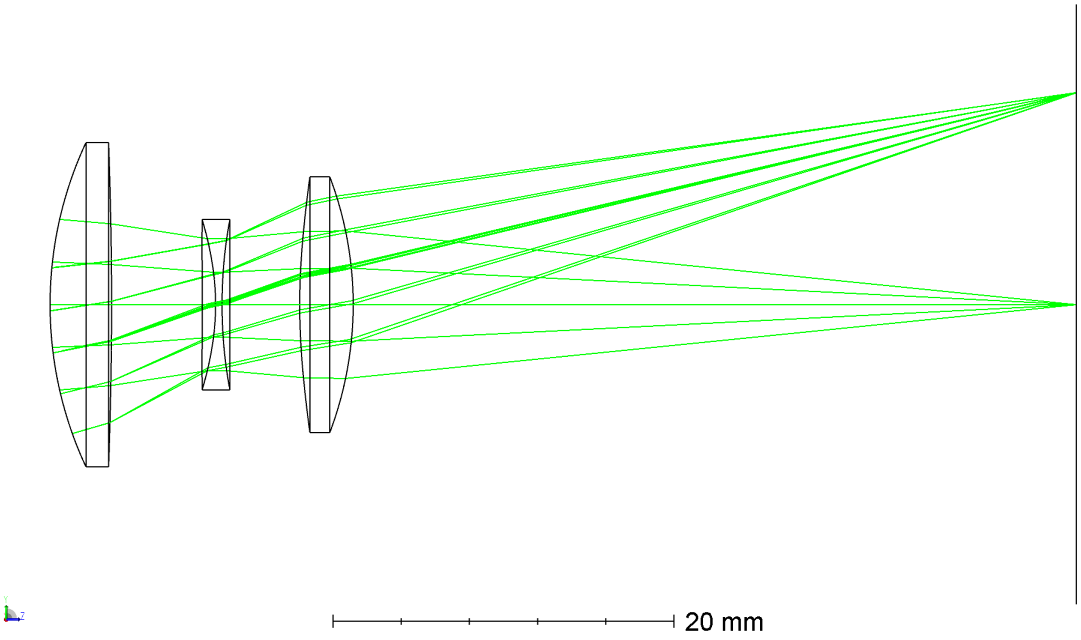


Figure 4.3: 3D layout of the **Nominal Minimum** freeform Cooke triplet

which suggested the existence of new local minima. Two samples of interest were selected, we will refer to each sample as "Candidate 1" $\mathbf{x}_1$ and "Candidate 2" $\mathbf{x}_2$.

Both candidate states were used to generate freeform systems which we exported to Zemax Optics Studio for characterization and to compare them with the nominal system linked to $\mathbf{x}^*$. We were interested in understanding how the different values in GDRT merit function translate into differences in optical performance between the systems, and the overall layout in terms of curvatures, thickness and surface shape.

The results in terms of optical performance for the three systems of interest are summarized in Table 4.1. It can be observed that the values of RMS spot radius for several field points are very similar for all the systems, the major changes happening close to the boundary of the field of view. We will now analyse each candidate in detail.

4.5.1. CANDIDATE 1

After loading the necessary sag files into Zemax, the system candidate 1 was compared with the one associated with the local minimum $\mathbf{x}^*$. Judging from Figure 4.3 and Figure 4.4, there are substantial differences in terms of optical layout between the two systems. The first and last lenses of the new candidate show increased thickness with respect to the nominal design. More importantly, the curvatures of the second lens have increased up to the point of almost contacting each other at the center. This obviously means that the system is close to being physically unrealistic, a situation which will become clear in a moment.

We decided to explore the merit function landscape along the n-dimensional line $\mathbf{v}$ connecting the two

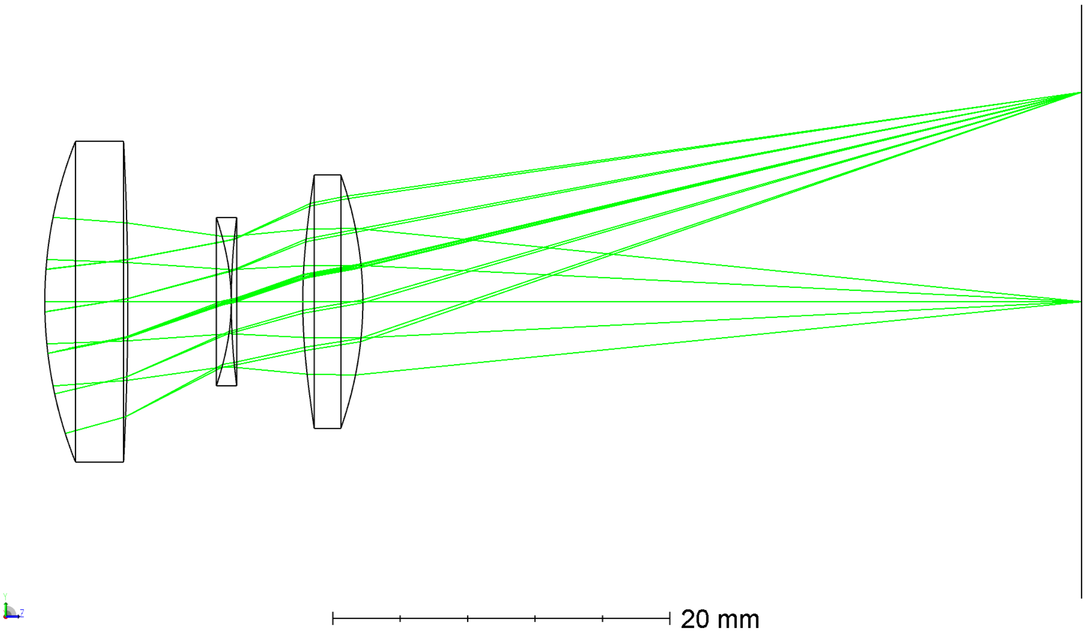


Figure 4.4: 3D layout of the **Candidate 1** freeform Cooke triplet

states $\mathbf{x}^*$ and $\mathbf{x}_1$. This allows for direct visualization of the merit function, even though the state space is inherently high dimensional.

$$\Delta\mathbf{x} = \mathbf{x}_1 - \mathbf{x}^* \tag{4.9}$$

$$\mathbf{v} = \mathbf{x}^* + \tau\Delta\mathbf{x} \qquad \forall\tau\in\mathbb{R} \tag{4.10}$$

The merit function f was evaluated along the interval $\tau = [-0.5, 1.5]$ which covers both sides of the pair of minima. The results shown in Figure 4.5 are quite revealing. Both points $\mathbf{x}^*$ and $\mathbf{x}_1$ appear to be two distinct local minima separated by an small 'energy barrier' of $\mathcal{O}(0.1)$. However, for values of τ slightly higher than 1.0, i.e. further away from $\mathbf{x}_1$, the merit function suddenly diverges. This is perfectly in accordance with the observation that **Candidate 1** is close to being a non-physical system due to the excessive curvatures of the second lens (let us recall Figure 4.4). When τ increases beyond the point where the two surfaces of the second lens auto-intersect each other, the ray tracing is no longer valid; thus the merit function loses its meaning.

This is an inherent disadvantage of generating the samples randomly in this *global minima* method. Even when taking the inverse Hessian information into account, for sufficiently large values of σ, the probability of drawing sample states which represent non-physical systems (with negative thickness, wrong curvature sign at the entrance surface, self-intersecting surfaces, etc) starts to become significant.

4.5.2. CANDIDATE 2

The same process was repeated for the Candidate 2 system, which exhibited slightly worse optical performance than Candidate 1. Interestingly, the optical layout in this case (see Figure 4.6) is opposite to that of Candidate 1. This time, the thickness of the first and second lenses are smaller compared to the nominal system, while the thickness of the second surface is larger. That last feature, combined with milder curvatures for the second lens, ensures that Candidate 2 appears to be a well-behaved system from a physical point of view. There is no danger of self-intersection between surfaces.

Once again, the merit function was analysed along the line connecting $\mathbf{x}^*$ and $\mathbf{x}_2$. As expected, the apparently 'more physical' behaviour of Candidate 2 translates into a much more robust merit function (see

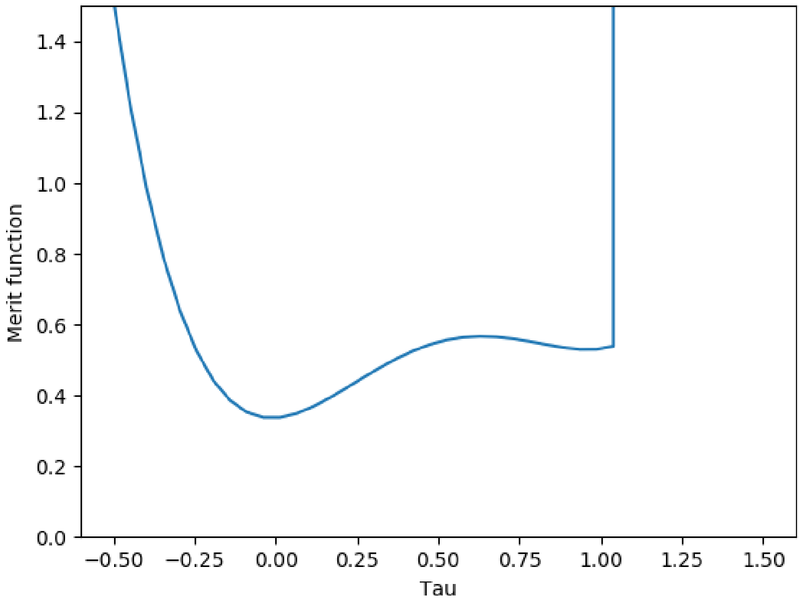


Figure 4.5: Merit function along the line covering $\mathbf{x}^*$ and $\mathbf{x}_1$

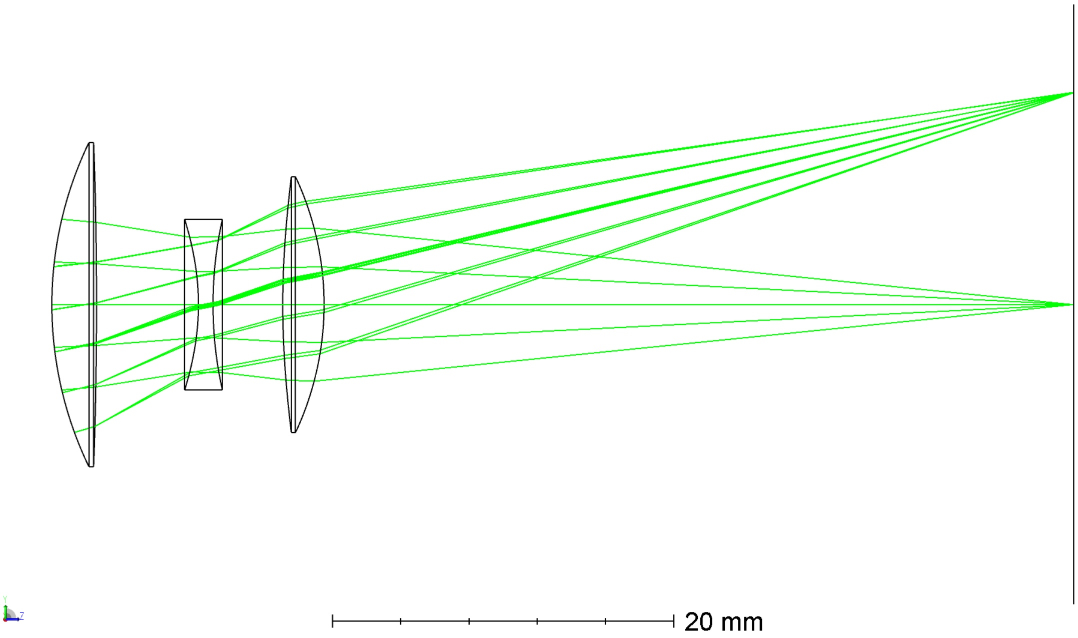


Figure 4.6: 3D layout of the **Candidate 2** freeform Cooke triplet

Figure 4.7). This time, the landscape is continuous and differentiable at both sides of the pair of minima. The merit function 'energy barrier' between the two minima is higher than in the previous case; around $\mathcal{O}(1)$.

A thorough analysis of the merit function in the vicinity of $\mathbf{x}_2$ revealed that the candidate point does not constitute a local minimum in the strict sense. Despite the apparent convergence behaviour observed when the sample was drawn and optimized, the magnitude of the gradient (around 10^{-3}) remains quite large in comparison with the value for $\mathbf{x}^*$ (around 10^{-7}). Even if Figure 4.7 shows that the point is indeed a local minimum along the $\mathbf{v}$ direction, in the complete n-dimensional state space, there are directions along which the merit function still improves.

The spectrum of the Hessian matrix evaluated at $\mathbf{x}_2$ is composed by 23 positive eigenvalues of large magnitude (some as high as 10^8) and only 1 negative eigenvalue with the smallest magnitude out of the whole set (10^{-1}). The result indicates that $\mathbf{x}_2$ corresponds to a saddle point of the merit function; but one which is almost completely convex except for a direction of mild negative curvature. This explains the remarkably slow convergence rate reported for this sample and helps emphasize the statement that second-order derivative information are really necessary in this type of system. With the merit function landscape as it is (in terms of curvature mismatch), gradient-based algorithms would struggle to progress in the neighbourhood of $\mathbf{x}_2$. And these results are for the *preconditioned* Cooke triplet; the situation for the un-preconditioned system could be even more dramatic.

If $\mathbf{x}_2$ is used as a new starting point for a high dimensional optimization with a very demanding tolerance on the norm of the gradient, one will observe that it eventually leads back to the true local minimum $\mathbf{x}^*$. Nevertheless, $\mathbf{x}_2$ still constitutes an important point of interest because it illustrates one of the fundamental problems of local optimization: how an unlucky starting point can lead the optimizer to become temporarily trapped in a poor state which appears to be a local minimum, simply because the convergence has slowed down significantly for many iterations. In a situation like this, the designer might end up looking at an optical system which is by no means optimum, as there is another local minimum in the vicinity which the optimization did not find.

Another important thing to note is that, as the number of dimensions n grows larger, the threshold δ on the norm of the gradient $\mathbf{g} = \nabla f$ (which is usually defined as the stop criterion for the optimization algorithm) becomes increasingly harder to fulfil. Let us assume that the gradient has p components equal to 0 and $(n-p)$ components equal to a non-zero value ϵ:

$$\mathbf{g} = [g_1, \cdots, g_p, \cdots, g_n] \qquad g_j = \epsilon \quad \forall j > p \tag{4.11}$$

The norm of the gradient is thus

$$\|\mathbf{g}\| = \sqrt{\sum_1^p 0^2 + \sum_{p+1}^n \epsilon^2} = \epsilon\sqrt{n-p} \tag{4.12}$$

Which only depends on the amount of non-zero components, rather than on the dimension of the search space. At low dimensionality ($n = 2, p = 1$ for instance), only 1 component needs to be equal to 0 for the gradient to have norm ϵ. In contrast, at high dimensionality (such as $n = 50, p = 1$), 49 components out of the total of 50 must be identically equal to 0 for the gradient to have the exact same norm as the low dimensional case. This makes the search for local minima more challenging as n grows. If one decides to relax the threshold δ to a higher value when optimizing for large n, the possibilities of encountering a situation like that of Candidate 2 increase. In contrast, if one keeps the δ fixed, the amount of iterations required to fulfil the gradient threshold will inevitably grow as n grows; iterations that at the same time grow in computational cost with n.

$$\|\mathbf{g}\|_{max} = max(|g_1|, \cdots, |g_p|, \cdots, |g_n|) \tag{4.13}$$

An alternative to this problem is to use the maximum norm defined as the maximum absolute value of all components of the gradient. But even so, the changes of fulfilling the threshold decrease as the dimension of the gradient grows larger.

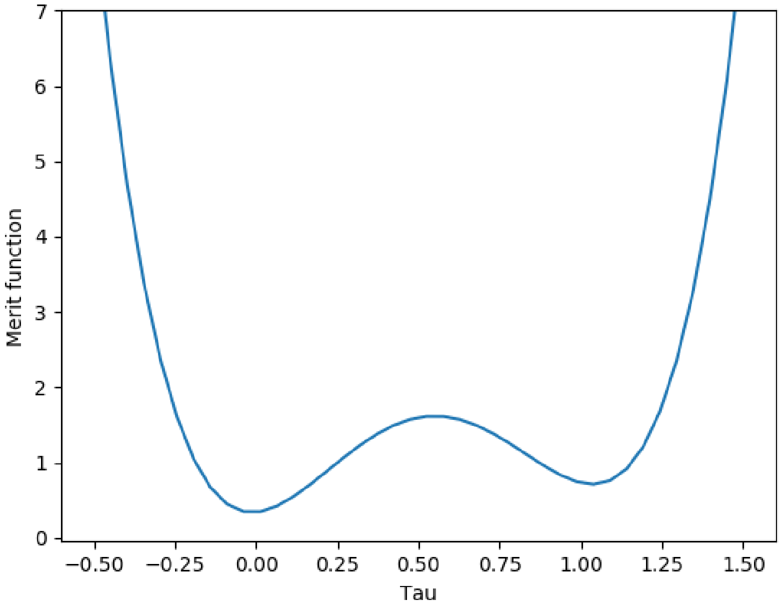


Figure 4.7: Merit function along the line covering $\mathbf{x}^*$ and $\mathbf{x}_2$

5

Alternating Direction Method of Multipliers

5.1. Motivation

The approach to freeform optics design proposed in this project usually leads to high-dimensional optimization problems with hundreds of variables. Most conventional optimization algorithms tend to scale poorly with dimensionality; therefore we decided to investigate the possibility of using algorithms specifically designed for, or at least well-suited, for large-scale optimization tasks.

The *Alternating Direction Method of Multipliers* (ADMM), an algorithm which is quickly gaining popularity, is particularly powerful for large-scale optimization problems which exhibit some inherent parallelism. In this chapter, we demonstrate via a simple proof-of-concept that the ADMM algorithm can be successfully applied to freeform optics optimization, in a way which opens the doors for future implementations of parallel computing techniques.

5.2. Alternating Direction Method of Multipliers

The *Alternating Direction Method of Multipliers* [5] is an optimization algorithm developed around the 1970s which is becoming increasingly popular for a wide range of tasks such as large-scale optimization [75], image processing [76] and machine learning [77]. This algorithm attempts to blend the beneficial characteristics of two different precursor algorithms: the decomposability of the dual ascent method and the superior convergence properties of the method of multipliers [5]. We will begin by considering a general optimization problem of the form:

$$\underset{x,z}{\text{minimize}} \quad f(x) + g(z) \tag{5.1}$$

$$\text{subject to} \quad Ax + Bz = c \tag{5.2}$$

With $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ and $c \in \mathbb{R}^p$. We can now introduce the augmented Lagrangian L_ρ of the function, with the Lagrange multipliers λ and penalty parameter ρ:

$$L_\rho(x, z, \lambda) = f(x) + g(z) + \lambda^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|^2 \tag{5.3}$$

The ADMM algorithm applied to this problem gives rise the following iterative update structure:

$$x^{k+1} = \underset{x}{\operatorname{argmin}} \quad L_\rho(x^k, z^k, \lambda^k) \tag{5.4}$$

$$z^{k+1} = \underset{z}{\operatorname{argmin}} \quad L_\rho(x^{k+1}, z^k, \lambda^k) \tag{5.5}$$

$$\lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \tag{5.6}$$

Two subsequent minimization steps in x and z followed by an update of the multipliers λ. The key features that makes ADMM special is the fact that the x and z variables are updated in an alternating fashion (thus its name of Alternating Directions Method of Multipliers).

It is often convenient to rescale the definition of the variables in the ADMM based on the following expressions for the *primal residual* r and the *dual variable* u:

$$r^k = Ax^k + Bz^k - c \tag{5.7}$$

$$u^k = \frac{1}{\rho}\lambda^k \tag{5.8}$$

The resulting ADMM structure can be written as follows:

$$x^{k+1} = \underset{x}{\operatorname{argmin}} \quad f(x) + \frac{\rho}{2}\|Ax + Bz^k - c + u^k\|^2 \tag{5.9}$$

$$z^{k+1} = \underset{z}{\operatorname{argmin}} \quad g(z) + \frac{\rho}{2}\|Ax^{k+1} + Bz - c + u^k\|^2 \tag{5.10}$$

$$u^{k+1} = u^k + (Ax^{k+1} + Bz^{k+1} - c) \tag{5.11}$$

The stop criteria for the ADMM is usually linked to the current value of two residuals: the *primal residual* r and the *dual residual* s:

$$r^k = Ax^k + Bz^k - c \tag{5.12}$$

$$\|r^k\|^2 \leq \epsilon_{primal} \tag{5.13}$$

$$s^k = \rho A^T B(z^{k+1} - z^k) \tag{5.14}$$

$$\|s^k\|^2 \leq \epsilon_{dual} \tag{5.15}$$

In light of the general structure of the algorithm presented above, one can conclude that the philosophy of ADMM is to decompose the solution to large global problem into small local subproblems which can be solved independently.

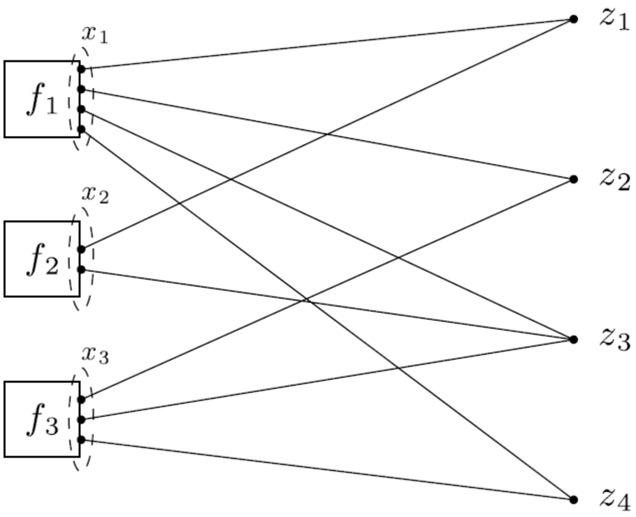


Figure 5.1: General form consensus optimization. Local objective terms are on the left; global variable components are on the right. Each edge in the bipartite graph is a consistency constraint, linking a local variable and a global variable component. Source: [5]

5.3. CONSENSUS OPTIMIZATION

One common application of ADMM, *consensus optimization* [5], is particularly suitable for the kind problem we deal in freeform optics optimization. Consensus optimization basically aims at optimizing an objective function f, separable into subfunctions f_i which depend on local variables $x_i \in \mathbb{R}^n$ under the constraint that all local variables should agree and be equal to a global variable z. We can express this idea as:

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{N} f_i(x_i) \tag{5.16}$$

$$\text{subject to} \quad x_i - z = 0 \quad i = 1, \cdots, N \tag{5.17}$$

Which is a particular case of the general problem from section 5.2. The idea behind consensus optimization can be easily understood by looking at Figure 5.1. Each subfunction to minimize depends on a particular subset x_i which acts as a local copy of the global variable z. The framework of ADMM consensus optimization ensures that after minimization of each subfunction, the different local copies are *in consensus*, i.e. are equal, with their corresponding component of the global z.

When the ADMM is applied to this type of problem, the resulting structure adopts the following form:

$$x_i^{k+1} = \underset{x}{\text{argmin}} \quad f_i(x_i) + \lambda_i^k \cdot x_i + \frac{\rho}{2} \|x_i - z^k\|^2 \tag{5.18}$$

$$z^{k+1} = \underset{z}{\text{argmin}} \quad \sum_{i=1}^{N} -\lambda_i^k \cdot z + \frac{\rho}{2} \|x_i^{k+1} - z\|^2 \tag{5.19}$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - z^{k+1}) \tag{5.20}$$

Thanks to the decomposability of the merit function f each x_i minimization can be solved independently over N different processors, which allows for full use of parallel computing capabilities to speed up ADMM. Then, the z minimization steps aggregates the results from the different processors to ensure consensus among the local variables.

In order to further understand how consensus optimization works, we will present an example of ADMM application to freeform optics design. Let us consider the optimization of a freeform Cooke triplet. It is quite common to define the merit function for an optical system as a combination of certain metrics of performance evaluated over a discrete set of field points, pupil coordinates or wavelengths. Let us assume that this time the merit function is the sum of the spot size over a total of N field points:

$$f = \sum_{i=1}^{N} f_i(x_i) \tag{5.21}$$

Were f_i is the spot size as field point i and x_i represents a local copy of the system parameters z. The framework of ADMM consensus optimization allows for parallelization of the field into a set of N partitions which can be optimized independently, while ensuring that all subproblems maintain consensus of the system parameters. The price one has to pay is that the number of variables n is essentially multiplied by N. Nevertheless the negative impact of that is quite low, as each x-minimization subproblem f_i is substantially simpler than the global form and can be efficiently solved in parallel using simple optimization algorithms. The field partitioning also has the beneficial effect of bringing down the size of the ray tracing operations as the total amount of rays being traced n_{rays} is distributed among the different processors to n_{rays}/N. In addition, the complexity of the z-minimization subproblem remains almost constant as the number of partitions increases, as it usually boils down to a simple weighted average of the local variables.

5.4. Algorithm implementation

In contrast with conventional algorithms such as BFGS or the Newton method, ADMM is not readily available in common Python packages such as Scipy. Fortunately, very recently SParse Optimization Research COde (SPORCO), an open-source Python package for solving optimization problems with sparsity-inducing regularization has been released [78].

SPORCO includes a generic implementation of the ADMM algorithm in the form of Python objects, which provides the basic skeleton to construct a customized implementation for an specific problem of interest. Thanks to its Python interface and open-source characteristics, SPORCO allows for a fast and simple blend with the existing functionalities of GDRT.

The generic ADMM from SPORCO contains basic functionalities transparent to the type of problem being solved such as residual computation, universal update structures, stopping criteria, etc. The main task of the user is to override the methods of the ADMM class to adapt them to the specific problem. This includes all the x, z-minimization steps, u-update, as well as constraint enforcements A, B, c. Therefore, there is complete freedom of choice for the algorithms used in the intermediate minimization steps, freedom to construct the different partitions and parallelize the solution those subproblems. In addition, the fact that the basic internal structure of the ADMM algorithm is already taken care of by SPORCO simplifies the development stage and minimizes verification and validation efforts.

5.5. Results

In this section we present the results for a proof-of-concept ADMM optimization of the freeform Cooke triplet which allows for parallel implementation. As already explained in subsection 3.1.2, the freeform modelling for the Cooke triplet made use of the existing symmetries to simplify the system. The RBF weights and field points are symmetrized in a octant fashion; thus we only consider the following field points to describe the complete field of view (see Table 5.1). This corresponds to the 6 independent field points for a 3×3 square grid with diagonal 20 degrees, which give rise to 6 field partitions.

This time the x_i variables ($\forall i = [1, \cdots, 6]$) correspond to local copies of the degrees of freedom ($n = 24$) associated with each field point. The merit function f was defined as the sum of the RMS spot size evaluated at each field point.

Table 5.1: Field of view partitions for the ADMM implementation of the freeform Cooke triplet

Field point	1	2	3	4	5	6
X [deg]	0	$10/\sqrt{2}$	$20/\sqrt{2}$	$10/\sqrt{2}$	$20/\sqrt{2}$	$20/\sqrt{2}$
Y[deg]	0	0	0	$10/\sqrt{2}$	$10/\sqrt{2}$	$20/\sqrt{2}$

Table 5.2: RMS spot radius comparison for the non-freeform and the freeform version of the Cooke triplet optimized with ADMM

Field point	1	2	3	4	5	6
Initial [um]	5.086	6.739	15.001	9.764	15.980	11.699
Final [um]	2.157	3.182	7.676	6.240	9.065	4.640

$$f(\{x_1,\cdots,x_N\}) = \sum_{i=1}^{N=6} f_i(x_i) \tag{5.22}$$

Even though the definition of each x-minimization step is entirely independent of the rest of local copies and can be solved in parallel over N processors, we started with a purely sequential computing approach for the sake of simplicity. For the x-minimizations we used the BFGS algorithm, while a Trust Region Newton-CG was used for the z-minimization because the Hessian of that subproblem is simple and readily available (it does not depend on the freeform system definition).

The ADMM algorithm fulfilled the stop criteria for the primal and dual residuals after less than 3500 iterations, with a total computational time of around 7 hours. This means that the approximate cost per ADMM iteration is around 7 seconds, which includes 6 x-minimization subproblems (the main drivers of computational time), a z-minimization subproblem (which converges most of the times in a single step) and the Lagrange multipliers update. A parallel implementation would definitely be more competitive.

The merit function evolution with iterations and the convergence behaviour of the residuals is shown in Figure 5.2 and Figure 5.3, respectively. A comparison of optical performance between the non-freeform starting point and the ADMM-optimized system is presented in Table 5.2, in the form of RMS spot radius evaluated at each field partition.

5.6. PARALLELIZATION

As already mentioned, the independent nature of the x_i minimization allows for parallelization of the subproblems. A very simple and easy method to gain some performance with Python is to use *multi-threading* to distribute the computation of x_i^k over several threads.

To illustrate this, we performed a quick experiment in which we compare two versions of the ADMM code: a purely sequential version and one which applies multi-threading to the x_i minimization. The standard sequential version of the code was benchmarked for the first 50 iterations, leading to a total time spent on the x-minimization step of 417 seconds (an average of 8.34 seconds per step). Then, the multi-threaded version of the code was also benchmarked for the same amount of iterations, taking a total of 346 seconds (average of 6.92 seconds per step).

$$speedup = \frac{t_{sequential} - t_{multi-thread}}{t_{sequential}} 100 \tag{5.23}$$

This constitutes an approximate speedup of 17%; which may not seem extraordinary but, for a large number of iterations (let us recall that the sequential ADMM took around 3500 iterations to converge), it can build up to a significant improvement in overall speed.

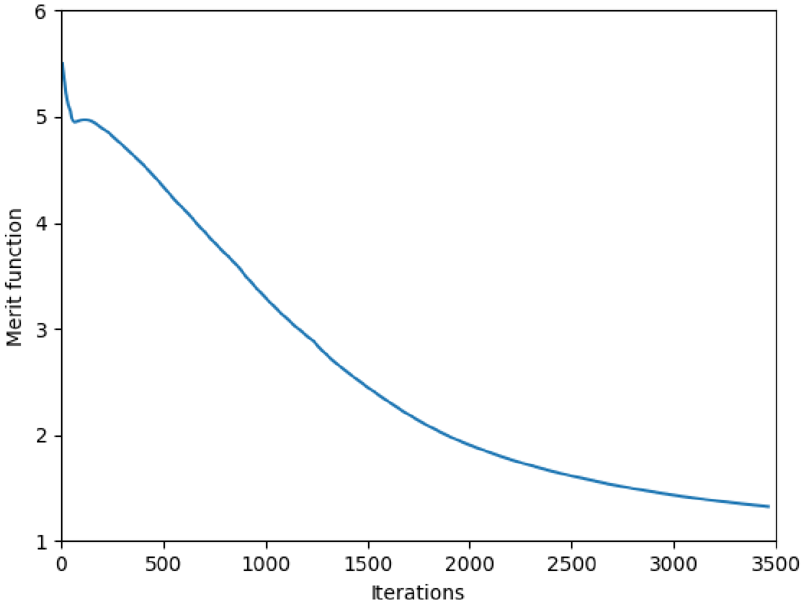


Figure 5.2: Merit function evolution with iterations for the ADMM algorithm applied to the freeform Cooke triplet

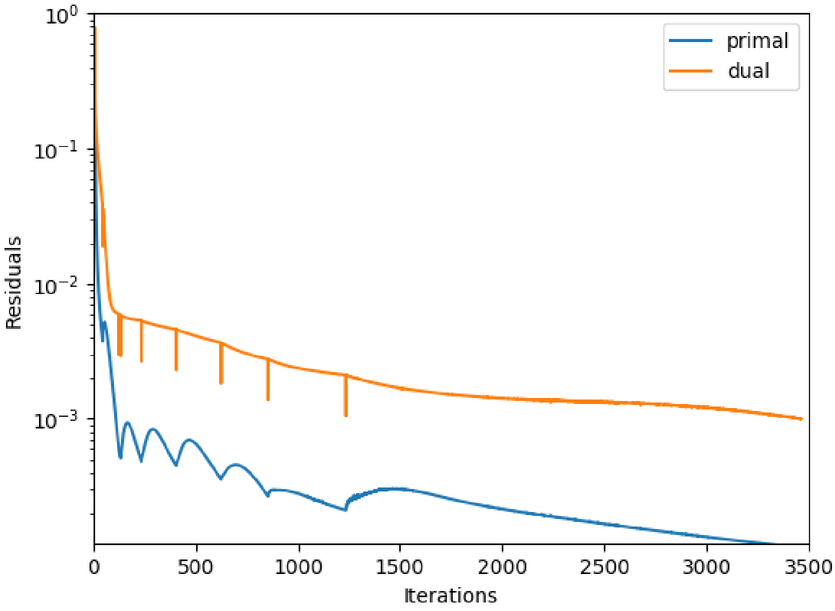
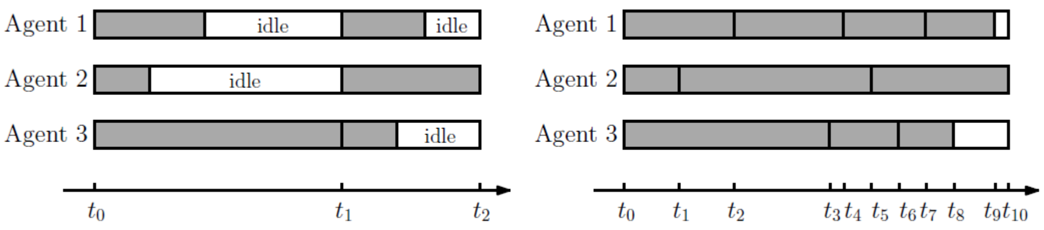


Figure 5.3: Evolution of the *primal* r^k and *dual* s^k residuals with iterations for the ADMM algorithm applied to the freeform Cooke triplet



(a) Sync-parallel computing

(b) Async-parallel computing

Figure 5.4: A comparison between *synchronous* and *asynchronous* parallel computing. Source: [6]

5.7. Asynchronous ADMM

5.7.1. Synchronous vs. asynchronous

In a usual parallel implementation of ADMM, several processors (*workers*) will be in charge of computing the x_i-minimization substeps independently. Nevertheless, for the z-minimization step the aggregated information of $x = \{x_1, \cdots, x_N$ is required, which means that in principle, the processors in charge of the z-minimization (*master*) would have to wait until all N have finished their respective task. This approach to parallel computing is usually referred to as *synchronous*, as synchronization events take place so that processors can update the variables and share data needed for the upcoming iteration. The obvious consequence of this method is that the overall speed of the computation is governed by the speed of the slowest of all processors; and situations in which most of the processors are in an idle state waiting for the rest to finish can become quite common.

In contrast, *asynchronous* parallel computing [77] tries to minimize idle times and thus speed up iterations by eliminating synchronization steps. Each processor operates asynchronously, without waiting for the rest to finish; consequently iterations get disordered as the fastest processors start their next iteration while others are still computing. A quick way to visualize the fundamental differences between *synchronous* and *asynchronous* parallel computing is shown in Figure 5.4.

This behaviour has a detrimental effect on the convergence properties of the algorithm due to phenomena such as uncordinated memory access and inconsistent read (details in [6]), which is easy to understand: when a fast processor starts the next iteration without waiting for some of the slowest ones to properly update the variables they are in charge of, that minimization substep will be done on the basis of outdated information and thus the result will not be truly optimum. As a result, additional iterations will be needed to reach convergence. The essential idea behind the *asynchronous* approach is that the detrimental convergence effects of asynchrony are expected to be compensated by a substantial improvement in speed of the iterations, leading to an overall improvement in computational speed (*'more less-meaningful iterations, but much cheaper'*).

5.7.2. Load balance

Asynchronous parallel computing introduces an additional issue to consider: *load balance*. An important assumption to guarantee convergence of an async-ADMM is that all updates from the*workers* should have the same probability of arriving at the *master* [77]. In other words, the speed of each *worker*, on average throughout the iterations, should be approximately the same for all N. In reality, this is difficult to ensure because the tasks sent to the different processor might differ in complexity.

In problems with significant load imbalance (one task being more complex than the rest), the *worker* in charge of the most complex task will repeatedly need more time than his peers. This will inevitably lead to his work being ignored by the rest of the fast *workers* quite frequently due to asynchrony. Therefore, having a consistent *load balance* is key to ensuring convergence of an asynchronous algorithm.

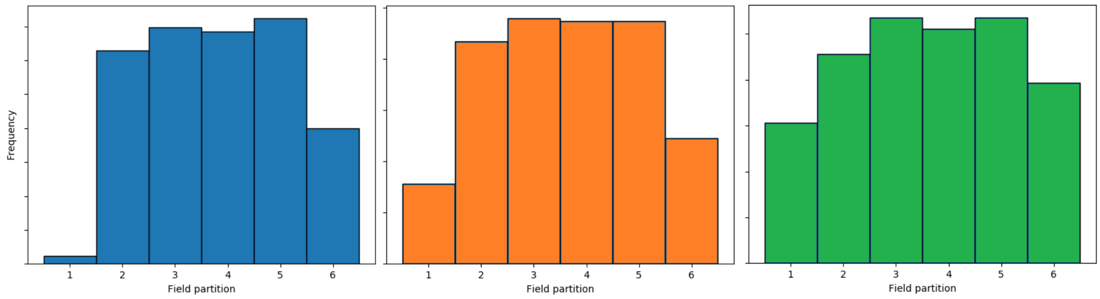


Figure 5.5: Relative frequency of each field partition being considered for the p^k fastest updates for different values of p_{min}. *Left*: $p_{min} = 3$, *centre*: $p_{min} = 4$ and *right*: $p_{min} = 5$

5.7.3. MIMICKING ASYNCHRONY

Implementing a fully functioning asynchronous parallel ADMM requires an expertise and effort which is far beyond the scope of this project. Nevertheless, the effects of asynchrony can be easily mimicked by a couple of modifications introduced in a sequential version of the code. This mimicry does not allow us to study the speed advantages of an asynchronous ADMM, but readily provides valuable information regarding the convergence properties.

The idea is quite simple. Let us assume the existence of a *master* processor in charge of the z-minimization step, and N *workers* each in charge of an x_i-minimization substep. These steps will be solved sequentially at every iteration, as in the standard ADMM implementation, each taking a certain time τ_i. At every iteration, we will assume the *master* "waits" an arbitrary amount of time τ for the *workers* to do their tasks. After τ seconds, he will aggregate the x_i^{k+1} updates from the p^k fastest processors and the x_i^k from the slowest ones and perform the z-minimization; where p^k is the random number of processors which have finished in $\tau_i < \tau$. This can be summarized as follows:

At iteration k do:

1. Compute all the x_i-minimization substeps and register each speed τ_i^k
2. Draw a random number p^k between p_{min} and N
3. Select the p^k fastest substeps and combine their new results x_i^{k+1} with the old ones from the $N - p^k$ slowest substeps x_i^k. In other words, ignore the results from the $N - p^k$ slowest updates.
4. Use the aggregated variable $\hat{x}^{k+1}$ to solve the z-minimization

We analysed the behaviour of the asynch-ADMM for several cases of p_{min}, according to the scheme described above. The relative frequency with which each field partition was considered among the p^k fastest is shown in Figure 5.5. It is evident that fields 2 through 5 require approximately the same time τ_i^k for their minimization steps and thus have almost the same probability of their updates being considered by the *master*. This set is an example of proper load balance. However, the results also indicate that as the number of minimum processors p_{min} being accepted decreases, the frequency with which updates for fields 1 and 6 are accepted also decreases. This reveals that those updates tend to be slower than the rest and are usually under-represented in the global update $\hat{x}^{k+1}$, which can lead to a degradation on the convergence properties of the ADMM when p_{min} is low.

The detrimental effects of this substantial load imbalance can be partially alleviated by introducing a certain modification to the schedule of the *master*. Instead of just waiting for the p^k fastest, no matter which ones they are, the *master* could keep a counter c_i^k of how many iterations have passed since receiving an update from worker i. If after C iterations no information from worker i has been received, the *master* will wait for it and reset c_i^k. This reduces the gains in speed because it forces a certain kind of synchronization, but otherwise the convergence behaviour of the algorithm can be quite poor.

6

Tolerance analysis

When an optical system is designed and optimized using software tools like Zemax or GDRT, the value of the parameters is known up to machine precision and the shape of the surfaces is assumed to be ideal without manufacturing errors. But when the system is actually manufactured, assembled and integrated there will be inherent uncertainties and inaccuracies which will affect the ultimate performance. In other words, the final *as-built* performance of the optical system will differ from the *nominal* performance given by the optical model. In some cases, the performance degradation can be so severe that the as-built system no longer meets the requirements. Therefore, after the design and optimization stage it is of paramount importance to analyze the impact of manufacturing errors, mechanical tolerances, assembly uncertainties and alignment errors on the optical performance of the system, to first of all ensure that the as-built system will comply, but also to design a reasonable assembly and alignment strategy which will guarantee that compliance. This analysis is usually referred to as a *tolerance analysis*.

In this section, the results of the tolerance analysis for the freeform spectrometer analysed in chapter 3 are presented. For this study, we used the so called *polar freeform* version of the system which combines both Cartesian and polar freeform surfaces. This tolerance analysis was entirely done using the state-of-the-art techniques included in Zemax because at the time of the analysis, GDRT did not have its own capabilities for tolerancing properly implemented. The results shown in this section will serve as an illustrative example of how limited the Zemax tolerancing capabilities are when freeform optics are involved, and how necessary it is to have a tolerance submodule for GDRT.

6.1. Nominal performance

Before jumping directly into the results of the tolerance analysis it is useful to first recall the requirements for the spectrometer system and the nominal performance of its freeform design. This will help understand what is the available margin for performance degradation, and will allow us to put the effect of mechanical tolerances into perspective.

Table 6.1 shows a summary of all relevant metrics of performance for the freeform spectrometer and compares them to the actual requirement. In light of the results we can conclude that a margin of approximately one order of magnitude is available for tolerances. This might seem like a healthy margin, but the *freeform* spectrometer design is a complex system containing a wide number of optical surfaces, which translates into a lot of degrees of freedom to be controlled accurately. In addition, errors in manufacturing can easily build up when many surfaces are involved, leading to severe performance degradation. Therefore, a careful tolerance analysis is essential to ensure the final as-built system will comply with the requirements.

6.2. Tolerance set-up

The tolerance analysis for the freeform spectrometer was performed twice, according to two different integration and alignment philosophies:

Table 6.1: Nominal performance of the *freeform* spectrometer compared to the requirements

RMS Spot Radius			
Min [um]	Max [um]	Average [um]	Requirement [um]
0.63	1.25	0.91	10
Dispersion			
Field [mm]	Value [mm]	Deviation [um]	Requirement [um]
0.0	2.19948	0.52	6
12.5	2.19965	0.35	
25.0	2.19932	0.68	
Keystone			
Field [mm]	Value [mm]	Deviation [um]	Requirement [um]
25.0	25.00081 @ 400 nm 25.00046 @ 1225 nm	0.35	2
Magnification			
Wavelength [nm]	Value [mm]	Deviation [um]	Requirement [um]
400	25.00081	0.81	2
800	25.00048	0.48	
1225	25.00046	0.46	

1. **Passive alignment**. For this analysis it was assumed that a passive alignment strategy would be followed. This means that all surfaces will be mounted without any intermediate control of the optical performance and without using some of the degrees of freedom of the system as compensators. This is the simplest approach, but also the most sensitive to alignment errors, as performance cannot be compensated in-situ during alignment.

2. **Active alignment**. For this analysis, some of the degrees of freedom of the optical system were assumed to act as compensators during the alignment process. This means that intermediate evaluations of the optical performance and active corrections would be carried out during alignment.

For the sake of consistency and completeness, the tolerance analysis for both alignment procedures was done via a step-by-step approach. At each stage of the process, a different type of mechanical tolerances was added to the model; subsequently increasing both the complexity and realism of the analysis. The sequence of tolerances is shown below:

1. **Decenters**. This includes decenters of the optical surfaces along their local XY-plane and changes in distances between surfaces, i.e. along the Z-axis.

2. **Tilts around the vertex**. This includes rotations of the optical surfaces with respect to their vertex.

3. **Tilts around the aperture**. In some cases, the vertex of the mirrors in the spectrometer design is outside the manufactured region, which means that the tilts around that point have little relevance to tolerances. In those cases, tilts around the actual aperture of the mirror represent more realistic tolerances. It is at this point that the issues regarding Zemax appear. Zemax does not have specific tolerance operands for tilts around apertures. To circumvent this problem one must define additional dummy surfaces to trick Zemax into doing the correct tilt. This is a very tedious operation.

4. **Surface irregularities**. This also poses a serious problem in Zemax. Tolerance operands on surface irregularities are only implemented for certain freeform surface types. Therefore, in a general case such as the freeform spectrometer the conventional operands no longer apply. As a first order solution, one can account for surface irregularity as errors in the radius of curvature of the surfaces, something that Zemax actually allows.

For each of the criteria shown above, both *sensitivity* and *Monte Carlo* analyses were performed.

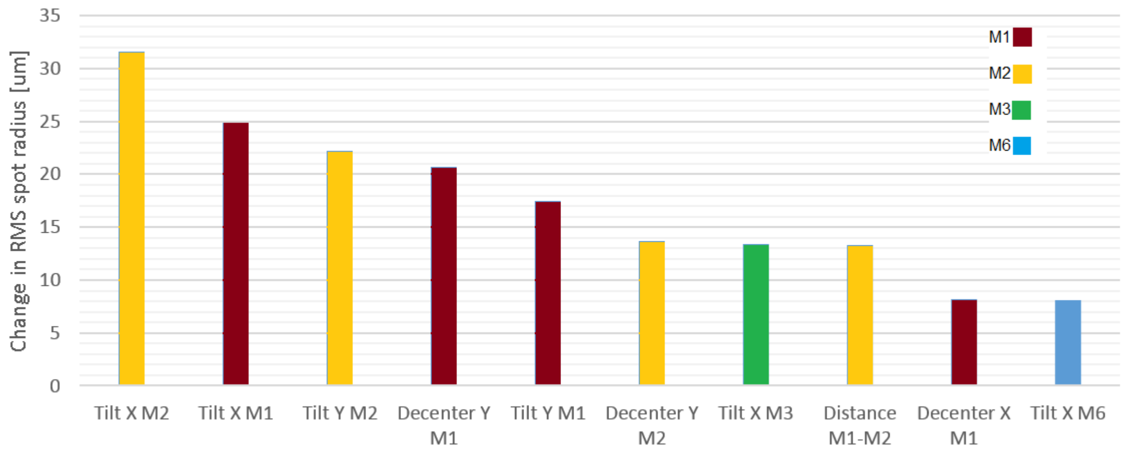


Figure 6.1: Impact of the top 10 tolerances on RMS spot radius (coarse tolerances of 50 μm)

As far the merit function is concerned, all tolerance analyses were done for each component of the merit function of the freeform spectrometer. This includes **RMS spot radius**, **keystone**, **dispersion** and **magnification**. This provides a better insight into the tolerance behaviour of the system and allows us to answer the following questions:

1. *What performance metric degrades more rapidly when mechanical tolerances are taken into account?*. Some metrics such as RMS spot radius might change significantly while others might be fairly insensitive. In the ideal case one would like all the metrics to be as insensitive as possible. In reality, one might be content with only the less important metrics being the most sensitive.
2. *What type of tolerance has the biggest impact on each performance metric?*. Not all metrics might respond in the same way to each kind of tolerance. Some might be very sensitive to tilts, while others might depend more on surface errors. In the ideal case, one would like all the metrics to behave similarly, so that a small number of degrees of freedom could be used to compensate all metrics at the same time.

The values of mechanical tolerances used for this study were based upon the rules of thumb for optomechanics reported in [79].

6.3. RESULTS

6.3.1. PASSIVE ALIGNMENT

In this section the results for the tolerance analysis considering a passive alignment strategy are presented. We followed the procedure described in section 6.2 of increasing the complexity of the model progressively until all major effects are considered. A very coarse value of 50 μm (and its respective tilt equivalent) was used for all tolerances and surfaces; and and 5 μm peak-to-valley for surface irregularity.

This allows us to clearly identify what particular parameters are the ones driving the performance degradation. These will then require tightening of their associated tolerances to more precise values so that the as-built performance complies with the requirements.

The top 10 most influential tolerances on RMS spot radius are shown in Figure Figure 6.1. In light of the results we can conclude that the first mirrors of the optical system M1 and M2 are the most sensitive. This is something to be expected, as the effects of alignment errors on surfaces at the beginning of the optical train have a longer optical path to travel, and thus more opportunities for amplification and build-up.

The actual values of performance estimated via a sensitivity analysis (with a Root Sum Square method) are shown in Table 6.2. First of all, it seems clear that with such coarse tolerances, all relevant metrics of

Table 6.2: Sensitivity analysis - performance after coarse tolerances (50 μm) for the freeform spectrometer

Metric	**Initial [um]**	**Final [um]**	**Requirement [um]**
RMS spot radius	0.81	62.5	10.0
Keystone	0.35	2.8	2.0
Dispersion	0.53	8.0	6.0
Magnification	0.61	64.3	2.0

Table 6.3: Monte Carlo analysis - performance statistics after coarse tolerances (50 μm) for the freeform spectrometer

Metric	**Initial [um]**	**Mean [um]**	**Standard Deviation [um]**	**Requirement [um]**
RMS spot radius	0.81	25.7	10.6	10.0
Keystone	0.35	1.2	0.6	2.0
Dispersion	0.53	2.7	2.1	6.0
Magnification	0.61	22.4	16.1	2.0

performance suffer so much degradation that the values fall outside the specification. The effect is specially severe in terms of RMS spot radius, the main metric of optical performance for our system. Fortunately, the results also show that the spectral metrics like keystone and dispersion are fairly insensitive to mechanical tolerances. This suggests that they will not pose a challenge when more precise tolerances are considered.

In addition, Monte Carlo analyses with 250 systems were done for each metric, the results summarized in Table 6.3. The performance estimation via Monte Carlo is less conservative than a RSS approach, and more reliable. If one pays attention to the statistics for both keystone and dispersion (assuming normal distributions), a 2σ approach reveals that in 95.45 % of the cases the metrics will be quite close to the requirement: 2.4 (2.0) μm for keystone and 6.9 (6.0) μm for dispersion.

The next step was to subsequently tighten the most critical tolerances to more precise values to reduce the performance degradation. This requires repeating the analysis several times and identifying the *killer* tolerances. The final values of tolerances used are summarized in Table 6.4. The tolerances not mentioned in that table remained at the initial value of 50 μm for decenters and tilts; and 5.0 μm PV for surface irregularity. One might observe that the most critical tolerances tend to be associated with tilts around the x-axis of the system. This is the only rotation that does not break the planar symmetry of the optical system.

The performance results for this configuration, which one should remember corresponds to simple passive alignment (i.e. no compensation during integration), are summarized here. In Table 6.5 the estimated performance via a sensitivity analysis are presented. Both keystone and dispersion are will within the specification. RMS spot radius appears to be off slightly, while magnification still remains a serious issue. Monte Carlo results (Table 6.6) support this trend and suggest that in the majority of the alignment scenarios RMS spot radius would be within acceptable values.

A very important point to analyse here is which mechanical tolerances are the main drivers of performance degradation after the tightening to more precise values. This will answer the question of whether the metrics that are still above the requirement could be further improved by tightening those tolerances. The truth is that, the tolerances for M1 and M2 still remain the performance *killers* as shown in Figure 6.2. But

Table 6.4: Final configuration of mechanical tolerances for passive alignment of the freeform spectrometer

Surface	**Tolerance type**	**Value [um]**
M1	Tilts & Decenters	10
M2	Tilts & Decenters	10
M3	Tilt X	15
M4	Tilt X	25
M5	Tilt X & Surface irregularity	25 & 2.5 PV
M6	Tilt X	25

Table 6.5: Sensitivity analysis - performance after tight tolerances (see Table 6.4) for the freeform spectrometer

Metric	Initial [um]	Final [um]	Requirement [um]
RMS spot radius	0.81	13.8	10.0
Keystone	0.35	1.5	2.0
Dispersion	0.53	3.5	6.0
Magnification	0.61	23.4	2.0

Table 6.6: Monte Carlo analysis - performance statistics after tight tolerances (see Table 6.4) for the freeform spectrometer

Metric	Initial [um]	Mean [um]	Standard Deviation [um]	Requirement [um]
RMS spot radius	0.81	6.6	2.4	10.0
Keystone	0.35	0.8	0.4	2.0
Dispersion	0.53	1.4	1.0	6.0
Magnification	0.61	8.8	6.6	2.0

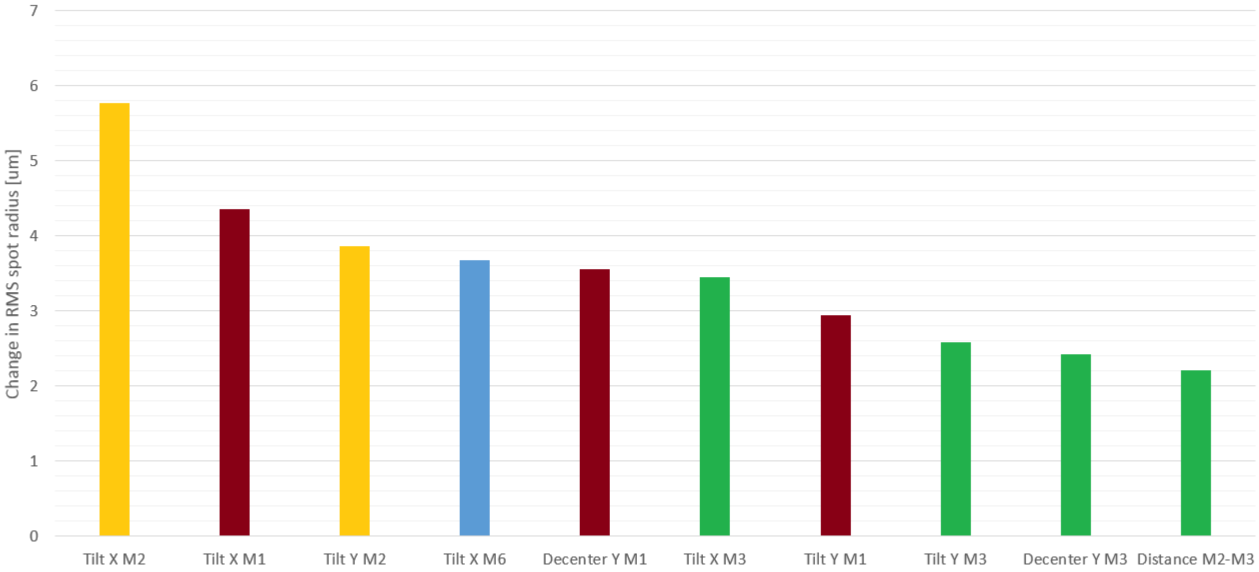


Figure 6.2: Impact of the top 10 tolerances on RMS spot radius (tight tolerances see Table 6.4)

Table 6.7: Comparison of RMS spot size for *passive* and *active* alignment strategies

Analysis type	**Passive aligment [um]**	**Active alignment [um]**
Sensitivity	13.8	10.1
Monte Carlo	N(6.6, 2.4)	N(5.1, 2.2)

Table 6.8: Effect of *active* alignment in the other performance metrics

Metric	**Passive aligment [um]**	**Active alignment [um]**
Keystone	1.5	2.0
Dispersion	3.5	3.7
Magnification	23.4	42.0

those tolerances are already set to quite demanding values (10 μm) just like other key tolerances like the tilt X of M6 and M3 (25 and 15 μm respectively).

Consequently, the most sensible thing to do at this point is to establish an *active* alignment strategy that could bring the performance down to reasonable values by using compensators, instead of demanding unrealistic tolerances.

6.3.2. ACTIVE ALIGNMENT

Briefly speaking, an active alignment strategy requires the following things: a *compensator*, *measurements* and some sort of *optimization*. The way it works is as follows. Some of the adjustable parameters of the system (such as the tilt of a mirror) are defined as *compensators* meaning that their value will be actively adjusted during the integration of the optical system to compensate for performance degradation. In order to select the proper value for the compensators, an AIT engineer must make some *measurements* of optical performance (the RMS spot radius at some point of the detector, for instance); then an *optimization* will decide what is the ideal value of compensation to reduce the performance degradation.

The result of an active alignment strategy is that with the same values of mechanical tolerances as in the previous passive scenario, the final performance of the optical system can be better, thanks to the effect of the compensators. In reality, an active alignment strategy need not be complicated or involve exotic compensators to achieve decent performance enhancements.

For the freeform spectrometer we decided to start the analysis with a simple compensation strategy in which only the position and orientation of the detector plane is used as compensator. The z-position of the detector can be used as a simple focus compensator, while the tilts around x and y can be used to correct the RMS spot radius. These are corrections that, in principle, can easily be applied in a realistic AIT scenario.

The results in terms of RMS spot radius for an active alignment scenario are summarized in Table 6.7. One can easily observe a substantial improvement thanks to the use of compensators. The estimated performance via sensitivity analysis (RSS method) is just above the requirement, but this estimation tends to be on the conservative side. The Monte Carlo results indicate an average performance at approximately 5.1 μm (half the value of the requirement) with a standard deviation of 2.2 μm. This means that with a confidence of 95.45 % the RMS spot radius in a realistic scenario is expected to be below 9.5 μm. Obviously, this analysis from Zemax is not totally realistic as it assumes no uncertainty in the value of the compensator during its optimization. In reality, there will be an uncertainty on the value of the parameters used as compensators; but as a first order analysis this model might suffice.

Due to the limitations of Zemax, the optimization used to compute the proper values of the compensator does not show how the other metrics are affected by these changes. But one can take the nominal values of the compensators as produced by Zemax and evaluate the keystone, dispersion and magnification in separate systems. This is a crude solution, but it gives some insight into how the compensators alter the metrics that are not considered during active aligment.

The results for this analysis are shown in Table 6.8. Evidently, the metrics that are not taken into account for the choice of the compensators suffer some degradation during the active alignment process. One cannot expect an AIT engineer to measure every single metric of performance during integration and use it to compute the proper compensation. In some cases, such measurement might even be impossible. In addition, there is no guarantee that metric will not counteract each other, meaning that one improves with a certain parameter (like the tilt X of the detector) while others get worse. In such scenario, compensation of all metrics at the same time becomes infeasible.

In any case, these results indicate that the three main metrics of optical performance of the freeform spectrometer (RMS spot radius, keystone and dispersion) will comply with their associated requirements once realistic mechanical tolerances are considered.

6.4. Freeform sensitivity

It was already mentioned in section 1.2, that one of the many factors slowing down the inclusion of freeform surfaces in optical payloads for space applications is the widespread assumption that freeform systems are more *sensitive* to assembly tolerances than their rotationally symmetric counterparts. What this notion implies is that even though freeform surfaces might have superior *nominal* performance, the rate of performance degradation when mechanical tolerances are considered is expected to be higher, leading to a significant loss of the performance advantage when the system is assembled and integrated. Expressed in mathematical terms, the performance f can be modelled around the nominal point θ as a series expansion on a particular system tolerance $\Delta\theta$ (such as the change on the tilt of one mirror):

$$f(\theta+\Delta\theta) = f(\theta) + \frac{1}{2}\frac{\partial^2 f}{\partial\theta^2}\Delta\theta^2 + \mathcal{O}(\Delta\theta^3) \tag{6.1}$$

Where $f(\theta+\Delta\theta)$ is the *as-built* performance which depends on the nominal performance $f(\theta)$, $\frac{\partial^2 f}{\partial\theta^2}$ is the *sensitivity* of the performance and $\Delta\theta$ is the mechanical tolerance. As the system has been optimized, the gradient of the performance $\frac{\partial f}{\partial\theta}$ is zero.

In light of this, the assumption states that $\frac{\partial^2 f_{free}}{\partial\theta^2} > \frac{\partial^2 f_{conv}}{\partial\theta^2}$ such that:

$$f_{free}(\theta+\Delta\theta_c) \geq f_{conv}(\theta+\Delta\theta_c) \tag{6.2}$$

For a particular value of tolerance $\Delta\theta_c$, even if the nominal performance was better:

$$f_{free}(\theta) \ll f_{conv}(\theta) \tag{6.3}$$

At this point two questions arise: *is the sensitivity of freeform systems actually higher?* and if so, *for what value of mechanical tolerance $\Delta\theta_c$ is the performance advantage of freeform systems lost?*. As far as the first question is concerned, there is no actual agreement in the literature, with some studies claiming that, contrary to the general assumption, *freeform* systems are actually less sensitive to assembly tolerances [7] while other study from this year claims that sensitivities are not really influenced by the freeform character, but by the degree of packaging, size and form of the optical surfaces [80]. Therefore, we decided to tackle this question for one of the freeform systems designed with GDRT.

For this investigation we extended the tolerance analysis presented in this chapter and compared the evolution of optical performance as a function of mechanical tolerances for both the *freeform* spectrometer system and its *non-freeform* baseline design. We started with the first two surfaces which had already been identified as the most sensitive to mechanical tolerances: M1 and M2.

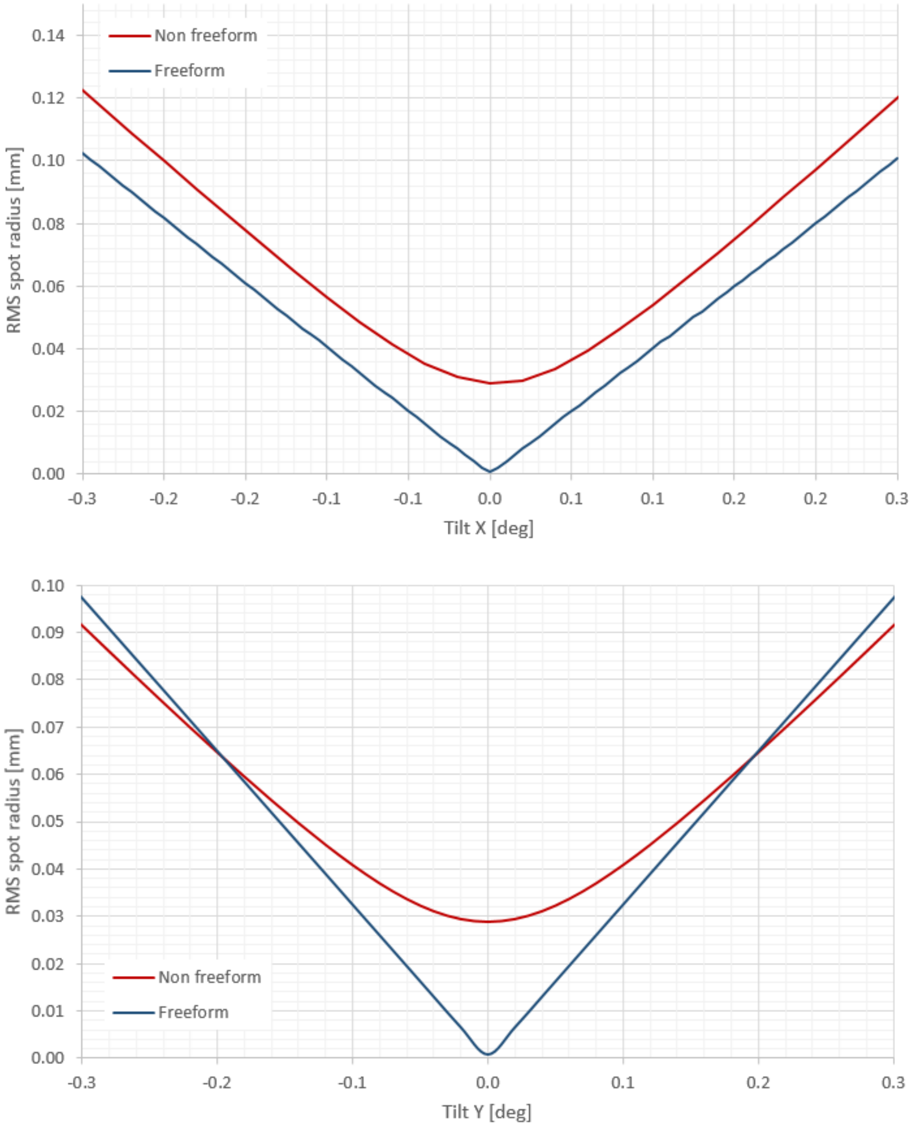


Figure 6.3: Evolution of RMS spot radius as a function of mechanical tolerances of M1 for the *non-freeform* and *freeform* spectrometer

The results for M1 are shown in Figure 6.3. Let us first recall quickly what each tilt implies: tilt around X does not break the plane symmetry of the system, whereas both tilts around Y and Z do. Tilt Z (usually referred to as *clocking* was not included in the plot as it is quite similar to tilt Y. It appears that around the nominal point the freeform system tends to have a higher *sensitivity* to the tilts of M1. But even if the performance degradation rate is higher, the crossover occurs for $\Delta\theta_c \simeq 0.2$ degrees, which corresponds to 7200 arcseconds. One should note that the sensitivity comparisons presented in [7] were done for a tolerance of only 30 arc-seconds.

In fact, when the aperture size is taking into account, 0.2 degrees of tilt Y for M1 translate into more than 250 microns of relative misalignment; 5 times larger than the coarsest value used for the tolerance analysis and obviously way too coarse of any realistic assembly scenario. When only the 30 arc-second interval is considered (see Figure 6.4) the performance degradation of the freeform system is almost negligible, and system remains substantially superior to the conventional one.

A similar situation was observed for the tolerances of M2. As already mentioned, these are the two most sensitive surfaces of the spectrometer as they are the closest to the entrance slit; but we were also interested in the sensitivity behaviour of surfaces further down the optical train. Thus we repeated the analysis for the

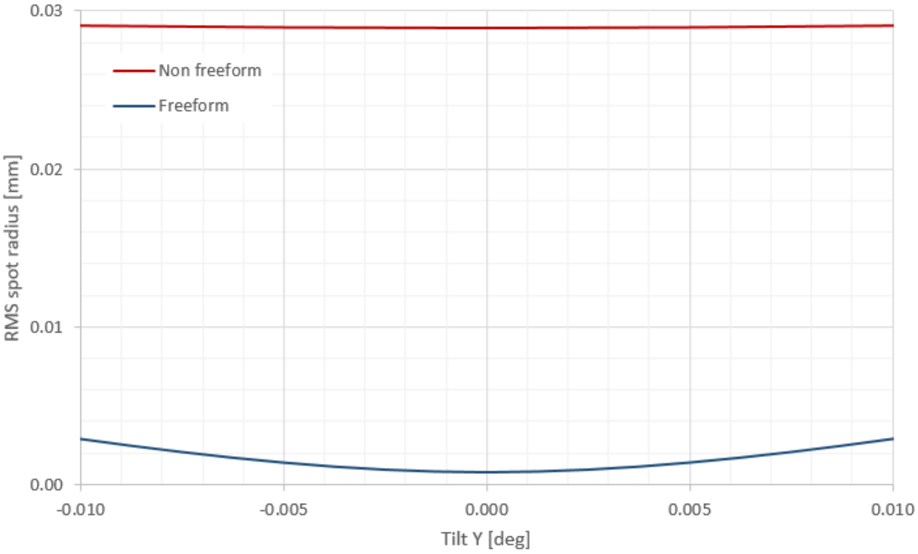


Figure 6.4: Close-up of tilt Y from Figure 6.3 for the range of tolerances analysed in [7]

last mirror before the detector: M6.

The results in this case revealed that although the sensitivity (formally $\frac{\partial^2 f}{\partial\theta^2}$) around the nominal point is indeed higher for the freeform system, the performance profile is such that no actual crossover $\Delta\theta_c$ exists. In other words, even for mechanical tolerances as coarse as 1 degree of tilt on all the axes of M6, the performance of the freeform system always remained superior to that of the conventional system.

To summarize, in accordance with other studies presented elsewhere [7], the assumption that freeform systems are inherently more sensitive to tolerances than conventional designs is, at the least, misleading. It is true that for the freeform spectrometer presented in this study, at the limit $\Delta\theta \longrightarrow 0$ the *sensitivity* in its formal definition $\frac{\partial^2 f}{\partial\theta^2}$ can be higher. But it is no less true that the difference in degradation rate is so low that even for the most initially-sensitive surfaces, the freeform performance advantage only vanishes for values of mechanical tolerances which are far outside what would be considered a reasonable assembly scenario. And for less sensitive surfaces, the advantage is actually never lost.

Consequently, for the range of mechanical tolerances already needed to meet the requirements, virtually no effect would be noted in the freeform system. What's more, the fact that the nominal performance of the freeform system is substantially superior to that of the conventional design allows for relaxation of the tolerances. A conventional design based on rotational symmetric surfaces would only barely comply with the requirements (the one shown here does not even reach that level), leaving almost no room for performance degradation due to tolerances. And in that case, it would not even matter which one is "more sensitive".

7

Future work

Despite the success of GDRT as a toolbox for optimization of freeform optics, a lot of potential still remains to be exploited. In this chapter we present some ideas for future extensions, novel functionalities and various improvements which could greatly enhance GDRT.

7.1. Extension of GDRT

7.1.1. Tolerance analysis

The next step after the development of an optical design concept which fulfils the requirements is to evaluate the impact of mechanical tolerances on the final performance. The optical model used for optimization relies on the assumptions that any parameter of the system is known to infinite accuracy (or to the machine precision, to be exact) and that the optical surfaces are perfect. In reality, when the optical surfaces are manufactured there will be errors and inaccuracies (for instance, in the form of surface roughness). In addition, the optical surfaces will be mounted, assembled and aligned within a certain level of uncertainty.

As a result of this, some performance degradation will inevitably occur once the system is integrated. It is the role of the optical engineer to model the effect of these mechanical tolerances right from the beginning, to ensure that despite the uncertainties, the final performance of the as-built system will comply with the specifications.

As already mentioned during the tolerance analysis of the freeform spectrometer, the capabilities and functionalities of Zemax for tolerance analysis are quite scarce as far as freeform surfaces are concerned. Most of the functionalities commonly used in standard surfaces are not even implemented for freeform surfaces in Zemax. This complicates the analysis of tolerances in freeform systems so that tedious workarounds are needed to compute useful information. Sometimes the issue can be so serious that a thorough and realistic tolerance analysis with Zemax is no longer an option.

Fortunately, the information required for analysing the effect of mechanical tolerances: *how the optical performance is influenced by changes in the parameters of the system?* is essentially the same as the differential ray tracing information which GDRT already employs for optimization. Consequently, it seems a natural choice to extend the functionalities of GDRT so that it can provide tolerance analysis capabilities. This could potentially avoid the issue of not having the right tools to do a proper analysis in Zemax.

7.1.2. Stray light analysis

The degradation of optical performance due to *stray light* (light that arrives at the detector in an unintended way) is a key concern for space instruments. Stray light can come in various forms such as light scattered from the different surfaces and structure of the instrument, unintended reflections on lenses, ghost orders from diffraction gratings or simply light from an outer source which follows a path not considered during the design stage.

A great deal of effort has to be devoted to analysis during the development of an instrument to ensure no major stray light issues will compromise its operation and to guarantee that the requirements will be met. Some of those analyses involve a *brute-force* approach, tracing thousands upon thousands of rays through the optical system to evaluate stray light, consuming a lot of time. Monte Carlo methods such as importance sampling are usually used to reduce the amount of rays needed to draw meaningful conclusions, but they are far from optimal.

As already mentioned, differential ray tracing contains information on how the path of a ray changes when either the system parameters or the ray data are changed. This information can be quite useful for stray light analysis, a topic which is mainly concerned about the paths of rays going through the optical system. In fact, some years ago a study [81] acknowledged the potential of differential ray tracing for stray light analysis. It argued that the derivatives could be used to aim rays at target points, resulting in faster convergence and less noise compared to traditional Monte Carlo methods. Therefore, this is a topic of interest for possible future applications of GDRT.

7.1.3. MERIT FUNCTIONS AND SURFACE MODELS

Despite that GDRT has a suite of merit functions and surface models wide enough for most applications, there is still room for further extension. It is always beneficial to have as many merit functions as possible available and once implemented for a particular case study they can easily be recycled for other optical systems. Currently, efforts are being devoted to the implementation of merit functions based on Point Spread Function (PSF) and Modulation Transfer Function (MTF) as it is common for optical systems to have some requirements linked to those metrics.

As fas as surfaces are concerned, the selection of surfaces for optical systems is almost endless both in the realm of conventional and freeform systems. Extending GDRT so that it can cope with a range of optical surfaces (Zernike polynomials, XY polynomials, Q-polynomials, etc) will allow for more thorough research into freeform systems. Moreover, the implementation of new surface models in GDRT is a relatively simple and easy task which does not require extensive knowledge of the inner working of the toolbox, and can be done by almost any user with a programming background.

7.2. IMPROVEMENTS OF GDRT

7.2.1. SPEED IMPROVEMENTS OF AUTOMATIC DIFFERENTIATION

The choice of using Theano for the automatic differentiation operations was mainly motivated by the ease of use of its Python environment which favours fast development and experimentation, as well as the existence of some practical experience prior to the start of this project. Nevertheless, the actual computational speed of Theano is by no means competitive to what other AD tools can offer, specially does developed in C or C++.

This lack of speed is the price one has to pay when dealing with freeform systems with many degrees of freedom. Although it is not critical, the slow character of Theano can sometimes lead to optimization runs which take several hours to compile and tens of hours to run; a highly undesirable situation. Therefore it could be extremely beneficial to dedicate some efforts to improve the speed of the computations. This can basically from within, by using GPU computing when possible to gain performance with Theano, or from without, meaning that other AD tool with intrinsically better performance could substitute Theano itself.

7.2.2. STARTING POINT GENERATION

The way GDRT usually operates is by taking a conventional design from Zemax whose performance can be no longer optimized and use it as a starting point for freeform optimization. One should note that the main feature of GDRT is *local* optimization meaning that the numerical algorithm looks for a system with enhanced performance, a *local minimum* in the neighbourhood of the starting point. But optical design is in itself a highly non-linear optimization problem which means that a multitude of local minima can exist with the merit function landscape. Thus, there is no guarantee that the candidate found by a local optimization method will be the *global minimum*. What's more, the result of the optimization (what minimum is found) directly depends on the choice of the starting point.

In light of this, one can quickly identify a weak point of GDRT: the result provided by the toolbox (when using local methods) depends on how well someone has done the task of optimizing the starting point in Zemax. If the initial system is inherently bad, GDRT will be able to improve it, but it is fairly possible that the result ends up being worse than what GDRT could have achieved if provided with a more suitable starting point. In other words, at the moment GDRT "works with what it gets". It is important to note that this is not an exclusive weakness of GDRT, but that Zemax also suffers from it. That is the reason why a good optical designer is essential for ensuring the success of an optimization with Zemax. One can not expect Zemax to take a terrible design which does not resemble at all what we want to achieve and output a perfectly reasonable design with optimized performance.

Nonetheless, a future improvement of great importance to GDRT would be to devise a way of generating a set of starting points which are *good* candidates for freeform optimization, instead of just relying on a single one. Another possibility would be to implement some sort of *global optimization* which would allow for analysis of the complete search space. However, global optimization is in itself much more challenging and time consuming than local optimization, and it does not scale well with dimensionality. Therefore, for the freeform systems studied with GDRT (with hundreds of degrees of freedom), global optimization if not implemented smartly could take an unfathomable amount of time.

8

CONCLUSIONS

In light of what has been presented in this report, we can at this point draw the following conclusions:

1. First of all, we have shown that the technique called *differential ray tracing* can be successfully applied to general optical systems containing freeform surfaces. The main drawback of differential ray tracing, the tediousness of deriving and hand-coding the expressions, was effectively circumvented by the use of *automatic differentiation* tools. This allows the computation of differential ray tracing results of arbitrary order up to machine precision, without resorting to finite-difference approximations. The method presented in this report (which is based on the application of *Fermat's path principle* and the *implicit function theorem*) is independent of any assumption regarding the rotational symmetry of the system and the type of surfaces it contains.

2. Based on those findings, a Python toolbox called ***Generalized Differential Ray Tracing*** (**GDRT**) was developed. The main goal of the **GDRT** toolbox is the design and optimization of optical systems based on freeform optics. **GDRT** has been extended during this research project to cope with a wide spectrum of optical system, ranging from *reflective* (containing mirrors) to *refractive* (lenses and prisms) and *diffractive* (diffraction gratings), or combinations of them. In addition, different surface description models for both *freeform* and *conventional* optical surfaces have been implemented.

3. A wide range of merit functions which are commonly used in optical design to quantify optical performance and optimize designs have been implemented in **GDRT**, including spot radius, wavefront error, distortion and spectral metrics for spectrometers. In addition, the versatility of **GDRT** allows for fast and easy extension of the merit function suite to deal with the particular requirements of the optical system of interest.

4. One of the key features of **GDRT** is that it takes advantage of differential ray tracing techniques to compute derivative information essential for effective optimization, including the gradient, Hessian matrix and Hessian-vector product evaluated up to machine precision. These derivatives can be computed for any type of merit function containing different metrics of optical performance and physical constraints, with respect to any variable of the optical system.

5. Based on these derivatives, **GDRT** can perform numerical optimization of optical systems in an autonomous way without directly relying on other software packages for optical design such as Zemax. Moreover, **GDRT** can easily handle freeform systems containing a large number of degrees of freedom (in the order of magnitude of 100 and 1000).

6. Several optimization case studies were used to test and analyze **GDRT**. The results indicate that **GDRT** is perfectly capable of taking stagnated designs based on rotational symmetric surfaces (created with state-of-the-art techniques) and optimize them into freeform systems with significant enhancements on optical performance. These case studies included systems from a wide range of configurations (including on-axis and off-axis), types of surfaces (reflective, refractive and diffractive), applications (spectroscopy, imaging), spectral range (visible, near infra-red and thermal infra-red) and size (compact and large). In all cases, substantial improvements in optical performance were reported.

7. A complete tolerance analysis was performed for one of the freeform systems optimized with **GDRT** (the freeform spectrometer), showing that it is possible to reach the desired levels of optical performance when reasonable mechanical tolerances are taken into consideration. In addition, remarks regarding the sensitivity of freeform systems to mechanical tolerances have been discussed.

8. **GDRT** is mainly based on *local* optimization techniques. Nevertheless, a method to characterize the parameter search space and look for global minima was demonstrated in this report. This method uses Hessian information to construct a high-dimensional sampling tailored to the landscape of the merit function.

9. A proof of concept regarding the use of parallel computing with **GDRT** was also presented. This study made use of the Alternating Direction Method of Multipliers (ADMM) to demonstrate how consensus optimization can be used to parallelize the optimization of freeform systems.

10. Potential for further extension of **GDRT** has been identified. Tolerance analysis of freeform optics is, at this day, the most promising application. The tolerancing capabilities of state-of-the-art software tools such as Zemax are very limited when freeform systems are considered. Other ideas include the use of **GDRT** for stray light analysis.

BIBLIOGRAPHY

[1] R. Geyl, H. Leplan, and E. R. Ruch, *Advanced space optics development in freeform optics design and polishing,* in Optical Design and Fabrication 2017 (Freeform, IODC, OFT) (Optical Society of America, 2017) p. JTh2B.6.

[2] W. Holland, D. Bintley, E. Chapin, A. Chrysostomou, G. Davis, J. Dempsey, W. Duncan, M. Fich, P. Friberg, M. Halpern, *et al.*, *Scuba-2: the 10 000 pixel bolometer camera on the james clerk maxwell telescope,* Monthly Notices of the Royal Astronomical Society **430**, 2513 (2013).

[3] DeIngenieur, *Vervuilingsmeter is klaar voor vertrek - pollution-meter is ready for departure,* (Accessed: August 2017).

[4] T. Peschel, C. Damm, M. Beier, A. Gebhardt, S. Risse, I. Walter, I. Sebastian, and D. Krutz, *Design of an imaging spectrometer for earth observation using freeform mirrors,* International Conference on Space Optics (2016).

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers,* Foundations and Trends in Machine Learning **3**, 1 (2011).

[6] Z. Peng, Y. Xu, M. Yan, and W. Yin, *Arock: An algorithmic framework for asynchronous parallel coordinate updates,* SIAM Journal of Scientific Computing **38**, A2851 (2016).

[7] K. P. Thompson, E. Schiesser, and J. P. Rolland, *Why are freeform telescopes less alignment sensitive than a traditional unobscured tma,* Proc. SPIE **9633**, 963317 (2015).

[8] F. Duerr, Y. Meuret, and H. Thienpont, *Potential benefits of free-form optics in on-axis imaging applications with high aspect ratio,* Optics express **21**, 31072 (2013).

[9] J. Reimers, K. Thompson, J. Troutman, J. Owen, A. M. Bauer, J. C. Papa, K. Whiteaker, D. Yates, M. Farsad, P. Marasco, M. Davies, and J. P. Rolland, *Increased compactness of an imaging spectrometer enabled by freeform surfaces,* in Optical Design and Fabrication 2017 (Freeform, IODC, OFT) (Optical Society of America, 2017) p. JW2C.5.

[10] K. Takahashi, *Development of ultrawide-angle compact camera using free-form optics,* Optical Review **18**, 55 (2011).

[11] F. Fang, X. Zhang, A. Weckenmann, G. Zhang, and C. Evans, *Manufacturing and measurement of freeform optics,* CIRP Annals - Manufacturing Technology **62**, 823 (2013).

[12] J. A. Connelly, R. G. Ohl, J. E. Mentzell, T. J. Madison, J. E. Hylan, R. G. Mink, T. T. Saha, J. L. Tveekrem, L. M. Sparr, V. J. Chambers, D. Fitzgerald, M. A. Greenhouse, and J. W. MacKenty, *Subsystem Imaging Performance and Modeling of the Infrared Multi-Object Spectrograph*, NASA Technical Reports Server (NTRS) (NASA Goddard Space Flight Center, 2004).

[13] K. Garrard, T. Bruegge, J. Hoffman, T. Dow, and A. Sohn, *Design tools for freeform optics, Proc. SPIE,* **5874** (2005).

[14] E. Atad-Ettedgui, T. Peacocke, D. Montgomery, D. Gostick, H. McGregor, M. Cliff, I. J. Saunders, L. Ploeg, M. Dorrepaal, and B. van Venrooij, *Opto-mechanical design of scuba-2,* Proc. SPIE **6273**, 62732H (2006).

[15] L. Summerer, *Specifics of innovation mechanisms in the space sector,* (2009).

[16] J. M. Howard and S. Wolbach, *Improving the performance of three-mirror imaging systems with freeform optics,* in *Renewable Energy and the Environment* (Optical Society of America, 2013) p. FT2B.6.

[17] *Freeform Optics enabling CubeSat Missions Project*, Tech. Rep. (Center Independent Research & Developments: GSFC IRAD Program | Mission Support Directorate, 2017).

[18] *2016 ROSES A.42 Solicitation NNH16ZDA001N-IIP Research Opportunities in Space and Earth Sciences* (NASA Earth Science Technology Office, 2016).

[19] K. Fletcher, ed., *Sentinel-5 Precursor: ESA Atmospheric Chemistry and Pollution-Monitoring Mission* (ESA Communications, 2016).

[20] K. Thompson, *Description of the third-order optical aberrations of near-circular pupil optical systems without symmetry,* J. Opt. Soc. Am. A **22**, 1389 (2005).

[21] K. Fuerschbach, J. P. Rolland, and K. P. Thompson, *Theory of aberration fields for general optical systems with freeform surfaces,* Opt. Express **22**, 26585 (2014).

[22] A. Y. Yi and L. Li, *Design and fabrication of a microlens array by use of a slow tool servo,* Opt. Lett. **30**, 1707 (2005).

[23] L. Li and A. Y. Yi, *Design and fabrication of a freeform microlens array for a compact large-field-of-view compound-eye camera,* Appl. Opt. **51**, 1843 (2012).

[24] E. Brinksmeier, Y. Mutlugunes, F. Klocke, J. C. Aurich, P. Shore, and H. Ohmori, *Ultra-precision grinding, CIRP Annals - Manufacturing Technology,* **59**, 652 (2010).

[25] X. Zhang, Z. Zeng, X. Liu, and F. Fang, *Compensation strategy for machining optical freeform surfaces by the combined on- and off-machine measurement,* Opt. Express **23**, 24800 (2015).

[26] J. McGuire, *A fast, wide-field of view, freeform tma: Design and tolerance analysis,* in *Imaging and Applied Optics 2015* (Optical Society of America, 2015).

[27] E. Hugot, X. Wang, D. Valls-Gabaud, G. Lemaitre, T. Agocs, R. Shu, and J. Wang, *A freeform-based, fast, wide-field, and distortion-free camera for ultralow surface brightness surveys,* Proc. SPIE **9143**, 91434X (2014).

[28] K. Fuerschbach, J. P. Rolland, and K. P. Thompson, *A new family of optical systems employing ϕ-polynomial surfaces,* Opt. Express **19**, 21919 (2011).

[29] E. Muslimov, E. Hugot, W. Jahn, S. Vives, M. Ferrari, B. Chambion, D. Henry, and C. Gaschet, *Combining freeform optics and curved detectors for wide field imaging: a polynomial approach over squared aperture,* Opt. Express **25**, 14598 (2017).

[30] R. Shi and J. Kross, *Differential ray tracing for optical design,* in *EUROPTO Conference on Design and Engineering* (1999).

[31] D. P. Feder, *Differentiation of ray-tracing equations with respect to construction parameters of rotationally symmetric optics,* J. Opt. Soc. Am. **58**, 1494 (1968).

[32] F.-W. Oertmann, *Differential ray tracing formulae; applications especially to aspheric optical systems,* Proc. SPIE **1013**, 20 (1989).

[33] P. D. Lin, *Determination of first-order derivative matrix of wavefront aberration with respect to system variables,* Appl. Opt. **51**, 486 (2012).

[34] T. B. Andersen, *Optical aberration functions: derivatives with respect to surface parameters for symmetrical systems,* Appl. Opt. **24**, 1122 (1985).

[35] P. D. Lin and W. Wu, *Determination of second-order derivatives of a skew ray with respect to the variables of its source ray in optical prism systems,* J. Opt. Soc. Am. A **28**, 1600 (2011).

[36] P. D. Lin, *Second-order derivatives of a ray with respect to the variables of its source ray in optical systems containing spherical boundary surfaces,* J. Opt. Soc. Am. A **28**, 1995 (2011).

[37] P. D. Lin and C.-S. Liu, *Jacobian and hessian matrices of optical path length for computing the wavefront shape, irradiance, and caustics in optical systems,* J. Opt. Soc. Am. A **29**, 2272 (2012).

[38] P. D. Lin, *Derivatives of optical path length: from mathematical formulation to applications,* J. Opt. Soc. Am. A **32**, 710 (2015).

[39] B. D. Stone, *Differential ray tracing in code v,* (2012).

[40] R. P. Feynman, R. B. Leighton, and M. L. Sands, *The Feynman lectures on physics* (1963) Chap. 26: Optics: The Principle of Least Time.

[41] A. L. Dontchev and R. T. Rockafellar, *Implicit functions and solution mappings,* Springer Monogr. Math. (2009).

[42] C. A. Floudas and P. M. Pardalos, *Encyclopedia of optimization* (Springer Science & Business Media, 2008) Chap. Bilevel Programming: Implicit Function Approach.

[43] H. Dixon, *Surfing Economics: essays for the inquiring economist* (Palgrave MacMillan, 2006) Chap. 6.

[44] J. Werner, M. Hillenbrand, A. Hoffmann, and S. Sinzinger, *Automatic differentiation in the optimization of imaging optical systems,* Schedae Informaticae **21**, 169 (2012).

[45] GitHub, *Cppad - a package for differentiation of c++ algorithms,* (Accessed: August 2017).

[46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, *Tensorflow: Large-scale machine learning on heterogeneous distributed systems,* arXiv preprint arXiv:1603.04467 (2016).

[47] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, *Theano: new features and speed improvements,* arXiv preprint arXiv:1211.5590 (2012).

[48] C. Menke and G. Forbes, *Optical design with orthogonal representations of rotationally symmetric and freeform aspheres,* Advanced Optical Technologies, Retrieved **2**, 97 (2013).

[49] Q. Meng, W. Wang, H. Ma, and J. Dong, *Easy-aligned off-axis three-mirror system with wide field of view using freeform surface based on integration of primary and tertiary mirror,* Appl. Opt **53**, 3028 (2014).

[50] G. W. Forbes, *Characterizing the shape of freeform optics,* Optics Express **20** (2012).

[51] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials* (CRC Press, 2002).

[52] C. C. de Visser and M. Verhaegen, *Wavefront reconstruction in adaptive optics systems using nonlinear multivariate splines,* JOSA A **30**, 82 (2013).

[53] C. S. Chen, Y. C. Hon, and R. A. Schaback, *Scientific computing with radial basis functions,* (2005).

[54] C. A. Palmer and E. G. Loewen, *Diffraction grating handbook (p. 15,* (2005).

[55] M. C. Hettrick and S. Bowyer, *Variable line-space gratings: new designs for use in grazing incidence spectrometers,* Applied optics **22**, 3921 (1983).

[56] W. R. McKinney, *Varied line-space gratings and applications,* Review of Scientific Instruments **63**, 1410 (1992), http://dx.doi.org/10.1063/1.1143030 .

[57] A. R. A. Manaf, T. Sugiyama, and J. Yan, *Design and fabrication of si-hdpe hybrid fresnel lenses for infrared imaging systems,* Opt. Express **25**, 1202 (2017).

[58] A. Z. Marchi and B. Borguet, *Freeform grating spectrometers for hyperspectral space applications: Status of esa programs,* in *Optical Design and Fabrication 2017 (Freeform, IODC, OFT)* (Optical Society of America, 2017) p. JTh2B.5.

[59] M. J. Kidger, *Fundamental optical design,* (SPIE Press, 2001).

[60] D. M. Vasiljevic, *Optimization of the cooke triplet with various evolution strategies and damped least squares,* in *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation* (1999) pp. 207–215.

[61] J. W. Gooch, *Sellmeier equation,* in *Encyclopedic Dictionary of Polymers*, edited by J. W. Gooch (Springer New York, New York, NY, 2011) pp. 653–654.

[62] Y.-B. Chen and P. D. Lin, *Second-order derivatives of optical path length of ray with respect to variable vector of source ray,* Appl. Opt **51**, 5552 (2012).

[63] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods* (SIAM, 1994) Chap. 2.

[64] N. J. Higham, *A survey of condition number estimation for triangular matrices,* SIAM Review **29**, 575 (1987), https://doi.org/10.1137/1029112 .

[65] M. Benzi, *Preconditioning techniques for large linear systems: A survey,* Journal of Computational Physics **182**, 418 (2002).

[66] G. Alleon, M. Benzi, and L. Giraud, *Sparse approximate inverse preconditioning for dense linear systems arising in computational electromagnetics,* Numerical Algorithms **16**, 1 (1997).

[67] L. Cook, *Three mirror anastigmatic optical system,* (1981), uS Patent 4,265,510.

[68] T. Yang, J. Zhu, and G. Jin, *Compact freeform off-axis three-mirror imaging system based on the integration of primary and tertiary mirrors on one single surface,* Chin. Opt. Lett. **14**, 060801 (2016).

[69] L. Zhang, D. Huang, W. Zhou, C. Fan, S. Ji, and J. Zhao, *Corrective polishing of freeform optical surfaces in an off-axis three-mirror imaging system,* The International Journal of Advanced Manufacturing Technology **88**, 2861 (2017).

[70] K. Fuerschbach, K. P. Thompson, and J. P. Rolland, *Interferometric measurement of a concave, phi-polynomial, zernike mirror,* Opt. Lett. **39**, 18 (2014).

[71] Z. Li, F. Fang, J. Chen, and X. Zhang, *Machining approach of freeform optics on infrared materials via ultra-precision turning,* Opt. Express **25**, 2051 (2017).

[72] T. Rendall and C. Allen, *Multi-dimensional aircraft surface pressure interpolation using radial basis functions,* Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering **222**, 483 (2008).

[73] M. A. Babyak, *What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models,* Psychosomatic medicine **66**, 411 (2004).

[74] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting.* Journal of machine learning research **15**, 1929 (2014).

[75] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, *An admm algorithm for a class of total variation regularized estimation problems,* IFAC Proceedings Volumes **45**, 83 (2012).

[76] B. Wohlberg, *Efficient algorithms for convolutional sparse representations,* IEEE Transactions on Image Processing **25**, 301 (2016).

[77] R. Zhang and J. Kwok, *Asynchronous distributed admm for consensus optimization,* in *International Conference on Machine Learning* (2014) pp. 1701–1709.

[78] Brendt Wohlberg, *SPORCO: A Python package for standard and convolutional sparse representations,* in *Proceedings of the 15th Python in Science Conference*, edited by Katy Huff, David Lippa, Dillon Niederhut, and M. Pacer (2017) pp. 1 – 8.

[79] K. Schwertz, *Useful estimations and rules of thumb for optomechanics,* (2010).

[80] I. B. Murray, *Optimizing reflective systems using aspheric and freeform surfaces,* in *Optical Design and Fabrication 2017 (Freeform, IODC, OFT)* (Optical Society of America, 2017) p. JTu1C.6.

[81] D. F. Rock, *Using differential ray tracing in stray light analysis,* in *SPIE Optical Engineering+ Applications* (International Society for Optics and Photonics, 2012) pp. 84950X–84950X.