Optimization methods for the design of progressive lenses

Glòria Casanellas Peñalver

Optimization methods for the design of progressive lenses

Author:
Glòria Casanellas Peñalver

A thesis submitted for the degree of
Doctor of Philosophy

Supervisor: **Jordi Castro Pérez**

Department of Statistics and Operations Research
Universitat Politècnica de Catalunya - BarcelonaTech
Barcelona
2020

Amb el suport de la Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya.

Abstract

This work is the result of an Industrial Doctorate developed through a partnership agreement between the Universitat Politècnica de Catalunya and the company Horizons Optical. This thesis solves the complex design of progressive lenses for eyeglasses, which is a real problem in the industry. The lens is the transparent part of the eye behind the pupil that helps humans to see clearly by focusing light onto the retina. Over time, the lens loses some of its elasticity and therefore can no longer accommodate clearly for near vision. This phenomenon is called presbyopia and explains why people need reading glasses as they become older. Progressive lenses correct presbyopia and have a complex design: they have an upper region for far vision, a low region for near vision (reading), and a corridor that connects these areas which allows clearly vision at an intermediate distance, for example, when looking at a computer screen. The surface of the progressive lens designed in this thesis is the surface that is farthest from the eye, thus the power in the near region is bigger than the power in the far region.

In geometrical terms, power and astigmatism are calculated using the principal curvatures of the lens surface. When the power changes vertically, unwanted lateral astigmatism (aberrations) appear as a result of the Minkwitz theorem. The focus of this thesis is the use of optimization methods in order to design progressive lenses minimizing the lateral aberrations (astigmatism) and providing the power required in each zone.

This thesis presents two different models for computing progressive lens. Both models are highly nonlinear, nonconvex and continuous and were solved using the AMPL modeling language and the interior point solvers IPOPT, LOQO and KNITRO. Both models have approximately 900 variables (the coefficients of a third-degree B-spline basis). The first model has about 7000 constraints, while the second model has about 15000 constraints. Each constraint corresponds to a property of power or astigmatism at a point on the grid that defines the lens surface.

The first model uses Cartesian coordinates and is an improved version of a previous model by the same author, published in a master's thesis. The CPU time in the master thesis was between 10 and 33 minutes, and in this thesis it has been reduced to less than 3 minutes using the same machine and the LOQO solver. In this thesis, all of the proposed instances converged using the LOQO solver and the Cartesian coordinate model, which was not the case in the master's thesis. However, with other solvers some of the instances did not converge using the Cartesian coordinate model of this thesis.

The second model uses spherical coordinates and exhibits better convexity properties than the previous one based on Cartesian coordinates. All of the problem instances converged using all the proposed solvers, and the quality of the solution was improved. CPU time for spherical coordinates increased in relation to the Cartesian coordinate model, due to large calculations involved, but the number of iterations needed to converge decreased considerably (for example, from a maximum of 192 iterations using the Cartesian coordinate model to a maximum of 84 iterations using the spherical coordinate model and

the same LOQO solver).

These models resulted in two publications. The first one is a patent for an invention that uses the Cartesian coordinate model and orients the astigmatism gradient, which is useful when personalizing progressive lenses for real users. The second publication is a scientific article published in *Optimization and Engineering* that proposes the spherical coordinate model.

Resum

Aquest document és el resultat d'un Doctorat Industrial desenvolupat a través d'un acord entre la Universitat Politècnica de Catalunya i l'empresa Horizons Optical. Aquesta tesi resol el disseny complex de les lents progressives per ulleres, que és un problema real de la indústria. El cristal·lí és la part transparent de l'ull, situada darrera la pupil·la, que ens permet veure-hi nítidament enfocant la llum a la retina. Amb el pas del temps, el cristal·lí perd la seva elasticitat i disminueix la seva capacitat d'acomodació a la visió de prop. Aquest fenomen s'anomena presbícia i explica per què necessitem ulleres quan ens fem grans. Les lents progressives corregeixen la presbícia i tenen un disseny complex: la zona superior s'utilitza per a la visió de lluny, la zona inferior per a la visió de prop (lectura) i el corredor que connecta aquestes dues zones permet una visió nítida per a distàncies intermèdies, per exemple per a mirar la pantalla d'un ordinador. La superfície de la lent progressiva dissenyada en aquesta tesi és la superfície més distant de l'ull, és a dir, la potència de la zona de prop és més gran que la potència de la zona de lluny.

En termes geomètrics, la potència i l'astigmatisme es calculen utilitzant les curvatures principals de la superfície de la lent. En augmentar la potència verticalment, apareixen aberracions laterals en forma d'astigmatisme com a conseqüència del teorema de Minkwitz. L'objectiu d'aquesta tesi és utilitzar mètodes d'optimització per a dissenyar lents progressives minimitzant les aberracions laterals (astigmatisme) i proporcionant la potència demanada per a cada zona de la lent.

Aquesta tesi presenta dos models diferents per a calcular lents progressives. Tots dos models són altament no lineals, no convexos i continus i han estat resolts utilitzant el llenguatge de modelització AMPL i els *solvers* de punt interior IPOPT, LOQO i KNITRO. Tots dos models tenen aproximadament 900 variables (les variables són els coeficients d'una base de B-splines de grau tres). El primer model té unes 7000 restriccions, mentre que el segon model té unes 15000 restriccions. Cada restricció correspon a una propietat de potència o astigmatisme d'un punt de la malla que defineix la superfície de la lent.

El primer model utilitza coordenades cartesianes i és una versió millorada d'un model previ de la mateixa autora, publicat en un treball final de màster. El temps de CPU en el treball final de màster era entre 10 i 33 minuts, i en aquesta tesi s'ha reduït a menys de 3 minuts utilitzant el mateix ordinador i el *solver* LOQO. En aquesta tesi, totes les instàncies han convergit utilitzant el *solver* LOQO i el model amb coordenades cartesianes, la qual cosa no passava en el treball final de màster. Tanmateix, en aquesta tesi algunes de les instàncies no han convergit utilitzant el model amb coordenades cartesianes i altres *solvers*.

El segon model utilitza coordenades esfèriques i presenta millor convexitat que l'anterior model de coordenades cartesianes. Totes les instàncies han convergit utilitzant qualsevol dels *solvers*, i la qualitat de la solució ha millorat. El temps de CPU utilitzant coordenades esfèriques ha estat superior que el temps del model de coordenades cartesianes, a causa dels llargs càlculs, tot i que el nombre d'iteracions ha disminuït considera-

blement (per exemple, d'un màxim de 192 iteracions utilitzant coordenades cartesianes a un màxim de 84 iteracions utilitzant coordenades esfèriques i el mateix *solver* LOQO).

Aquests dos models han tingut com a resultat dues publicacions. La primera és la patent d'invenció que utilitza coordenades cartesianes i orienta el gradient d'astigmatisme, la qual cosa és útil a l'hora de personalitzar les lents progressives per als usuaris finals. La segona publicació és un article científic publicat a la revista *Optimization and Engineering*, que presenta el model amb coordenades esfèriques.

Agraïments

Em fa especial il·lusió escriure aquestes ratlles d'agraïment, precisament ara, en plena recta final de la redacció de la tesi, per tal de correspondre a totes aquelles persones que d'una manera o altra m'han ajudat a tirar endavant en aquest projecte. Fa uns anys que vaig emprendre el camí d'aquest Doctorat Industrial realitzat entre l'empresa Horizons Optical i el Departament d'Estadística i Investigació Operativa de la UPC. Després d'uns quants anys dedicada al treball aplicat i a les exigències del món empresarial, la realització del doctorat m'ha permès el retrobament amb el món universitari. Dos mons sovint massa distants que aquest programa de Doctorat Industrial, impulsat per la Generalitat de Catalunya i l'AGAUR, encertadament té el ferm propòsit de fer convergir. Gràcies a l'existència d'aquest tipus de doctorat he pogut compaginar feina i recerca universitària. Dedico els següents paràgrafs a totes aquelles persones que m'heu ajudat a fer-ho possible i que m'heu donat l'escalf i el suport necessaris per a mantenir vives les ganes d'aprendre, d'investigar, de millorar i de trobar sentit a un projecte que m'ha pres hores, dies i temps, a la vegada que m'omplia d'exigència i satisfacció. Per a tots vosaltres van aquestes línies.

En primer lloc, vull agrair de tot cor que el Dr. Jordi Castro acceptés dirigir-me la tesi; sense la seva comprensió, entesa i dedicació no hauria estat possible lliurar-la ni en condicions ni a temps. Les anades i vingudes entre l'empresa i la universitat han esdevingut constants en aquests anys de camí. És per això que faig extensibles els meus sincers agraïments a tot el Departament d'Estadística i Investigació Operativa de la UPC, als companys doctorands, al personal d'administració i serveis i a la resta de professors, que m'han ajudat sempre que ha calgut. Moltes gràcies a tots, i en concret a la Paula de la Lama, per haver compartit hores de recerca des de l'empatia i complicitat de qui està vivint un procés vital similar.

En segon lloc, vull agrair amb la mateixa sinceritat i convicció a tot l'equip humà d'Horizons Optical i INDO, però de manera significativa vull agrair al Dr. Pau Artús, el responsable de la tesi a l'empresa Horizons Optical, el seu suport i entesa, així com la confiança necessària per haver cregut en el projecte des de bon inici i haver-me donat el suport i el temps necessaris per a dedicar-m'hi. Al llarg del procés d'elaboració de la tesi hi han hagut temps més difícils que d'altres i el Departament de Recerca i Desenvolupament d'INDO on vaig començar la tesi es va escindir per a crear l'empresa Horizons Optical. No sempre ha estat fàcil donar prioritat a la investigació, ni compaginar els interessos acadèmics amb els empresarials. Per això també vull agrair l'esforç de tots els companys amb qui he compartit projectes laborals durant aquests cinc anys. Alguns estan a INDO, d'altres a Horizons Optical, d'altres aporten el seu bon fer en altres empreses.

Em fa il·lusió agrair al Rafa Perpiñan la seva col·laboració i el tracte dispensat en aquests últims anys que hem compartit i compartim equip i feina en el camp del disseny de lents. Així mateix dono gràcies al Javier Vegas i a la Blanca Fuster per les hores passades discutint sobre optimització, mètodes numèrics, informàtica, òptica i mil temes més. Ha estat sovint en l'ambient d'aquestes discussions on han sorgit molt bones idees. Faig extensible l'agraïment al Dr. Juan Carlos Dürsteler, antic cap del Departament de Recerca i Desenvolupament d'INDO, per haver transmès que la investigació és necessària per a treballar millor. També al Dr. Manel Espínola, antic cap de l'equip de Disseny de Lents, amb qui vaig compartir hores de feina i plantejaments inicials de la tesi, i de qui he après molt. I encara dins de l'àmbit empresarial, finalment agraeixo al Santi Soler, actual CEO d'Horizons Optical, haver vetllat pel bon ambient i el treball en equip en aquesta empresa.

Voldria esmentar també els amics de tota la vida, que amb més o menys distància han donat suport a la tesi mentre caminaven decidits en les seves vides. També la gent de Sant Feliu, per haver-me acollit en aquesta nova ciutat. I evidentment, vull agrair i ho faig plenament convençuda, tota l'estima i suport que he rebut de la meva família. A la meva mare li agreixo els bons consells de valor incalculable. Al meu pare, la seva paciència, assessorament i exigència en les correccions lingüístiques. I als meus germans Albert i Pau, la comprensió i estima incondicional.

Finalment vull mencionar el Jordi i el Marçal, per tots aquells vespres i caps de setmana dedicats a la tesi, allò que sembla que mai acaba. Espero que a partir d'ara puguem tornar a gaudir de més estones en família.

Moltes gràcies!

Contents

Chapter 1

Introduction

The lens is the transparent part of the eye behind the pupil that helps humans to see clearly by focusing light onto the retina. The parts of the eye and the lens can be seen in Figure 1.1. In young people the lens of the eye is soft and flexible, allowing it be easily reshaped by the thin muscles inside the eye to focus on near and far objects. Over time, the lens loses some of its elasticity and therefore can no longer accommodate clearly for near vision. This phenomenon is called *presbyopia* and explains why people need reading glasses as they become older. The simplest eyeglasses to correct presbyopia are monofocal lenses that can only be used for near vision. A more complex type of eyeglasses are bifocal lenses, allowing clear vision at two different distances: far vision and near vision. Finally, the third type of eyeglasses are progressive lenses (also called progressive addition lenses) that have a complex design: they have an upper region for far vision (*far region*), the *corridor* for middle vision and the lower region for near vision (*near region*). The different parts of a progressive lens surface are shown in Figure 1.2 (left). The far region is used to focus objects that are at a distance of 2 meters or more, near region is used to focus objects at around 40 cm, and the corridor changes gradually its power in order to be used to focus objects between 2 m and 40 cm.

This thesis will consider progressive lenses used for eyeglasses. In this thesis, only one of the two surfaces of the lens will be calculated, and the other surface of the lens will be a spherical cap. That means that we will consider progressive lenses with no center thickness. The two main properties of a progressive lens are the power and the astigmatism (formulas for which will be shown in Chapter 3) that are defined for each point of the lens. From now on, by term *power* will refer to *surface power*, or, more precisely, to *mean surface power*. Similarly, the term *astigmatism* will refer to *surface astigmatism*. In geometrical terms, power is the product of the mean principal curvatures of the lens surface multiplied by a constant; and the astigmatism is the product of the principal curvatures difference multiplied by the same constant. When the power changes vertically, unwanted lateral astigmatism (aberrations) appears because of the Minkwitz theorem [29]. For more information about the Minkwitz theorem, see [12].

In this thesis, the progressive lens surface being considered is the surface of the lens that is farthest from the eye. The lateral view of this progressive lens surface is shown in Figure 1.2 (right). The near zone must have greater power (P_N—power in the near zone—) than the far region (P_F—power in the far zone—), and the corridor exhibits a gradual increase in power. Optimization techniques are used to design these types of surfaces, while minimizing lateral aberrations that appear in form of unwanted astigmatism. The aim of this thesis is to obtain a surface of a progressive lens, with some desired

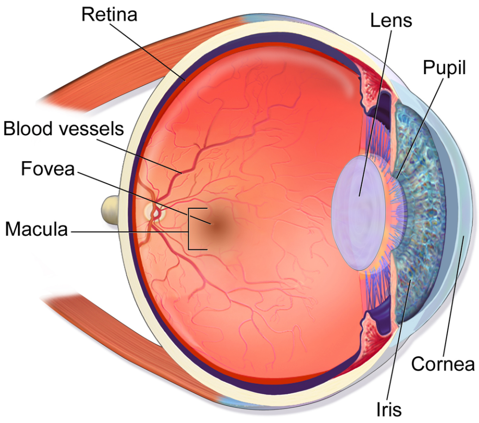


Figure 1.1: Parts of the eye.

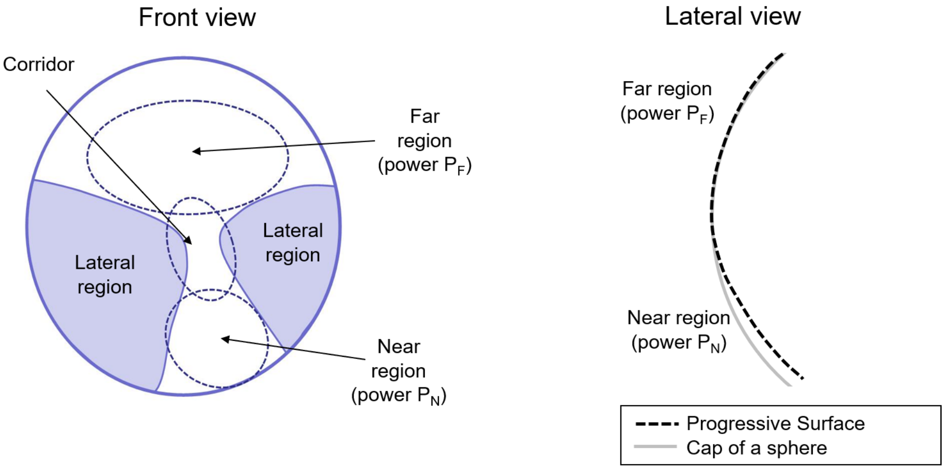


Figure 1.2: Parts of a progressive lens.

power and minimum astigmatism using optimization methods. The optimization methods used in this thesis are interior point methods. The next Chapter 2 presents a review of optimization methods.

This thesis is a continuation of the Master's thesis [47] by the same author. That work developed a model of the progressive lens, and that model has been enhanced and further enlarged here. We may observe that the problem of designing progressive lenses is a nonlinear and nonconvex problem. Both theses use Cartesian coordinates with a surface model of B-splines. This thesis has also tested different degrees of B-splines. In addition, spherical coordinates have also been used to enhance the convexity of the problem.

Contributions of this thesis in the form of publications are listed in Section 2.3. There are three main contributions of this thesis. Firstly, the Cartesian coordinate model presented in [47] has been improved, reducing the number of iterations and the time required by solvers to converge. Secondly, an application of the Cartesian coordinate model has been patented, orientating the direction of power and astigmatism gradients. Finally, a new model with spherical coordinates is presented, being “more convex” that the model using Cartesian coordinates.

The quality of the solution obtained using spherical coordinates will, in general, be the same as that using the Cartesian coordinates. The main advantage of using spherical coordinates is that, theoretically, the problem is “less nonconvex” (as will be shown), and, consequently, the optimization problem is (ideally) expected to converge faster (i.e., with fewer iterations). The data provided in the spherical coordinates model must be referenced in angles (radians), and the data provided in the model using Cartesian coordinates must be referenced in distances (mm) on the projected x–y plane.

The structure of this thesis is as follows. The next Chapter 2 describes the state of the art of nonlinear optimization methods and the design of progressive lenses. The detailed contributions in form of published articles, conferences and other miscellaneous publications are described at the end of Chapter 2. The definition and computation of the power and astigmatism are given in Chapter 3, which also introduces the optimization model. The detailed optic model for this problem is given in Chapter 4 for Cartesian coordinates, and in Chapter 5 for spherical coordinates. These two sections also discuss convexity issues. A new application of this optimization model is presented at the end of Chapter 4 that allows the power and astigmatism gradients to be orientated in different directions. Chapters 4 and 5 also give examples of progressive lenses computed using the two different models. These two Chapters also provide numerical results using the interior point solvers LOQO, IPOPT and KNITRO hooked to AMPL. Finally, future work, conclusions and contributions are detailed in Chapter 6. Tables of numerical results using the Cartesian coordinate model are shown in Appendix (page 106).

Chapter 2

State of the art

2.1 Nonlinear optimization

In this thesis, optimization methods have been used to calculate the design of progressive lenses. However, in other works different authors have used alternative methods to solve the issue of designing progressive lenses, for example variational methods. This chapter will describe nonlinear optimization and other methods used to calculate progressive lenses.

Due to the fact that the expressions of power and astigmatism are nonconvex and nonlinear, our problem is neither convex nor linear. Consequently, this section will detail the optimization methods that can be used to solve a nonconvex and nonlinear problem.

The practical methods of nonlinear and nonconvex optimization are:

1. Penalty and augmented Lagrangian methods (implemented in the solvers LANCELOT, MINOS)

2. Sequential quadratic programming (implemented in the solver SNOPT)

3. Interior point methods (implemented in the solvers LOQO, KNITRO/direct, IPOPT)

Various solvers are now available which can apply these methods. The names of commercial solvers are shown in parentheses. In the case of KNITRO, there exists different algorithms of the same solver.

All of these methods are iterative methods. Depending on the way solvers iterate from one point to the next, they may use a line search or trust region. For more information about these techniques, see the book [7, page 19]. This book also describes the optimization problems cited above (pp. 497–597).

We have used IPOPT, LOQO and KNITRO solvers that use interior point methods to solve the progressive lenses problem. In [6, pp. 487–491] there is a short description of interior point methods. For more information about the solver LOQO, see [9] and [10], and for more information about the IPOPT solver, see [11]. The interior point algorithm implemented in the KNITRO software is described in the articles [2] and [4].

We have not yet succeed in solving our problem with method that were not interior point methods. Using KNITRO with the active set algorithm (algorithm 3) and the SQP algorithm (algorithm 4) the problems did not converge. We also, unsuccessfully, tried other active set codes, such as MINOS (which implements a projected Lagrangian method for nonlinearly constrained problems): in 10 million iterations (3 hours of CPU time) on server 2, MINOS did not converge, and the surface obtained—although it looked like a progressive lens—was very rough.

2.2 Progressive lenses

The literature defining the progressive lens is quite extensive (e.g., [27, 28]). However, literature on how to calculate progressive lenses is scarce, mainly for reasons of confidentiality, since this problem is generally addressed by private industry. However, a few references are available.

2.2.1 History of the progressive lens

A detailed history of the progressive lens from 1907 to 1986 can be found in [17]. [14] traces the development of progressive lenses from 1951 to 2000; it should be noted that this book is written from the prespective of the company Essilor. We remark that the free-form technology developed from 1983 to 2000. Free-form technology allows progressive lenses to be manufacturated using digitally controlled machines instead of mechanical ones. In 2000, Zeiss and Rodenstock launched their first free-form PALs, and soon afterwards other companies also started to manufacture progressive lenses using free-form technology. The importance of free-form technology is explained [26]. Some of the progressive lenses designed in this thesis have been cut using free-form machines.

2.2.2 Evaluating the quality of a progressive lens

As stated in the Introduction (Chapter 1), a progressive lens has different zones (*far vision zone*, *corridor* and *near vision zone*); the properties being studied in these areas are power and astigmatism. Depending on the shape and properties (i.e. power and astigmatism values) of these zones, different types of progressive lenses can be designed, to suit different users and different activities.

The different types and features of progressive lenses are explained in the reviews [27] and [28]. These publications describe the limits of astigmatism values and define what makes a good quality progressive lens. The article [21] also gives a broad description of progressive lenses.

However, depending on the user and how the progressive lenses are to be used different distributions of the aberrations (lateral astigmatism) are better than others. For example, we can classify the users as *head movers* or *eye movers*, and wearers will tend to have a preference for one type of progressive lenses. For more information about *head movers* or *eye movers* see [28] and [20].

Progressive lenses can be designed to suit the different activities being carried while they are worn. Examples of activities include working in an office, doing outdoor sports or driving a car. Depending on the main purpose of the lens, the progressive lens will have the far, near or corridor zone bigger or smaller and the power will be also modified. The Minkwitz theorem means that it is not possible to maximize all of these zones within the same lens, so one zone must be prioritized depending on the user's main activities. For example, in an office progressive lens, the distance vision zone will be smaller than the near vision zone. This is because the lens is specifically intended for near and intermediate vision (intermediate vision is used to look at the computer or laptop screen). However, in a multipurpose lens or an outdoor sports lens, the distance vision zone will be larger than the other zones. Moreover, the power will differ depending on the user's prescription and on the distance from the eye of these various different activities. For example, when driving a car the near zone will be used to focus on objects at the same distance as the car dashboard, whereas the near zone in an office will be used for focusing to the objects located at normal reading distance (usually 40 cm).

The goal of this thesis is to present a versatile method for designing any type of progressive lenses. By modifying the different parameters in our model (described in Section 4.5, page 27), we can calculate different types of progressive lenses.

2.2.3 Modeling lenses

In some articles and patents, progressive lenses surfaces are modeled using B-splines, whereas other publications use Zernike polynomials or other type of polynomials. Reference will be made to these publications below.

Splines

Many publications use splines when modeling progressive lenses. This is the case in the articles [25] and [37] and the theses [36], [19] and [17]. Furthermore, the article [25] also specifies the properties of the splines used. Finally, the patent [13] uses splines of five–degree with control points spaced 10 mm or 20 mm. We may observe that in Section 5.6 (page 91) splines of three–degree will work better than splines of five–degree. We also remark that in this thesis control points will be spaced 4 mm (see 4.6.1, page 29).

Zernike polynomials

In the review [28] and in some other publications, such as the thesis [22], it is said that Zernike polynomials can be used to model progressive surfaces.

Linearisation and Taylor series

The article [25] proposes an approximation with the Taylor series. Finally, the thesis [32] uses polynomials of degree 20 to model the surface of the progressive lens.

2.2.4 Designing progressive lenses

There are several methods used for designing progressive lenses. This thesis focuses on our research into optimization methods, but we would like to describe, or at least to mention, others methods used for designing lenses. These methods are:

- Variational methods
- Constructive methods
- Optimization

Variational methods

The article [25] describes the use of a variational method, where the strategy is to minimize an objective function without constraints, by using numeric approximations. The objective function is the sum of the astigmatism plus the difference between the lens power and the objective lens power.

Other articles that use variational methods are, for example, [35] and [34]. In addition, the article [30] explains the variational method and [35] reviews it.

In order to solve variational methods, a discretization is used. This normally involves the use of numerical algorithms and finite element methods. See, for example, [34].

Constructive methods

In the article [31], a constructive method is described. Instead of considering the whole surface of the lens, this method considers the different regions of the lens.

Optimization methods

Many articles and patents cite optimization methods in order to solve the problem of progressive lenses, but few give many details about the optimization method used.

In the reviews [27] and [28] numerical optimization methods are cited. However, they state that the “finite element method seeks to minimize” the merit function. They add that the merit function includes power errors, unwanted astigmatism and gradients of power. Strangely, this review does not explicitly cite the minimization of the astigmatism gradient in the merit function, although it does suggest that some users “may benefit from designs with softer gradients”. To sum up, this review cites both finite element methods and numerical optimization methods without giving details of any of them.

However, there are two theses that merit special attention because they solve the problem of progressive lenses by applying optimization methods, so they are quite relevant to our work. One is the thesis [32] that solves multilevel optimization problems using a trust-region algorithm. This thesis makes use of the GALAHAD library. An application of this thesis is the design of progressive lenses. The bibliography includes the article [1], which covers interior point methods, but the thesis develops multilevel algorithms that are a type of trust-region methods. In 2009 thesis [32] resolves a progressive lens model without constraints in 20 minutes using polynomials of degree 20 in fewer than 10000 iterations. However, during this project we solved a progressive lens model in less than 5 minutes using B-splines and the LOQO solver in fewer than 200 iterations (see Section 4.6, page 29). Another difference between [32] and our work is that in [32] the model has no constraints (the only set of constraints have been moved to the objective function and our work has different sets of constraints. Consequently, we do have more flexibility in the objective function. The complete model in [32] is formed only by the following objective function:

$$\min \sum_{i=1}^{M} \alpha_i Ast_i^2 + \beta_i \Big(Pow_i - PowRef_i \Big)^2 \tag{2.1}$$

where α_i and β_i are given weights, $PowRef_i$ is the reference value for the optical power over the whole surface of the lens and M is the number of points. The optimization problem to be solved in our model is described below in Chapter 4 and involves the minimization of (4.26), subject to constraints (4.27), (4.28), (4.29), (4.30) and (4.25) using variables (4.24) (see page 28). That means that in both works the astigmatism is minimized and the requested lens power is fixed in three regions for our work and in the whole surface for [32]. This is: our model does not require the power values for the whole surface of the lens to be specified, whereas [32] requires it. Having a degree of freedom with the power value in the lateral zones is a clear advantage. In addition, we specify a threshold astigmatism value for the far region, corridor and near regions (producing higher quality lenses) and the power and astigmatism gradients can also be minimized.

To sum up, our work allows more freedom with the lateral power, produces smoother surfaces (minimizing the gradients of power and astigmatism), imposes a minimum level of quality by fixing the astigmatism threshold values, and the method converges in fewer iterations. Computational time cannot be compared because these projects date from different years and use different computers. The concept of using different weights applies in both theses. We may note that the thesis [32] cites the company INDO, because it is said that INDO have approved this progressive lens design.

The other thesis that is quite related to our work is [22]. This thesis used Zernike polynomials and a B-spline basis and developed different interior points methods (trust-region methods) and also conjugate gradient methods in order to solve the progressive lens problem. It did not use any commercial software (although it cites KNITRO). The optimization problem is as follows:

$$\begin{aligned} \min\ & \frac{1}{2}\sum_{i=1}^{N_p} w_i^{Sph}\left(Sph_i(u)-\widehat{Sph_i}\right)^2 + w_i^{Cyl}\left(Cyl_i(u)-\widehat{Cyl_i}\right)^2 \\ s.t.\ & c_k(k)=\widehat{c_k},\ k=1..m_k \\ & \widehat{c_k^{inf}(k)} \le c_k(k) \le \widehat{c_k^{sup}(k)},\ k=m_{k+1}..m_l \end{aligned} \tag{2.2}$$

where N_p is the number of points which is about 2000 to 3000. The $\widehat{Sph_i}$ and $\widehat{Cyl_i}$ are the objective power and the objective cylinder (parameters) for all points, and the variables $Sph_i(u)$ and $Cyl_i(u)$ are the power (sphere) and the astigmatism (cylinder). We may note that *cylinder* is synonymous with *astigmatism*. In the objective function, weights w_i^{Sph} and w_i^{Cyl} are used to prioritize certain zones, for example the corridor, over others. Concerning the constraints, it states that at certain points (usually only 2 or 4 points)—the distance and the near points—some values of power and astigmatism are given. These values are $c_k(k)$ and are limited by $\widehat{c_k^{inf}(k)}$ and $\widehat{c_k^{sup}(k)}$ or are fixed to another parameter $\widehat{c_k}$. Therefore, in this thesis power and astigmatism values must be specified for the whole surface of the lens surface ($\widehat{Sph_i}$ and $\widehat{Cyl_i}$), but 2 o 4 points are considered more important. This is due to ISO specifications, that impose strict tolerance in near and distance points; see [23] and [24]).

The number of variables depends on the definition of the surface using Zernike polynomials or B-splines and is said to be 150 or 250. The number of iterations and time to convergence depends on the optimization method implemented and, in general, the number of iterations is approximately twice the number of variables (i.e. twice 150 or 250). The time to convergence, which also depends on the optimization method being implemented, is between 7 minutes and 44 seconds in a Windows NT, Pentium III 666 MhZ machine. Again, this optimization modelization does not allow to minimize the astigmatism or power gradients (as we have achieved in this new project), and likewise does not allow the sum of astigmatism to be minimized (as in [32]). The main drawbacks of this model are the need to generate all power and astigmatism data (parameters) for the whole lens surface and the fact that smoother lenses, which minimize astigmatism and power gradients, cannot be produced. The optimization code developed used Matlab and Fortran 77 and solved a non-lineal and nonconvex optimization problem with (few) constraints.

Interior point methods:
To date, we have found few publications relating progressive lenses with interior point

optimization methods, aside from INDO or Horizons Optical publications. The only other major publication (aside from INDO and Horizons Optical publications) is the above thesis [22].

The master's theses produced at INDO, [16] and [47], cite the use of interior point methods. This thesis continues the work achieved in these master's theses.

2.2.5 Work carried out at INDO and at Horizons Optical

Considering that this thesis has grown out of a collaboration between Horizons Optical and UPC, it is important to reference certain publications. We must also note that, the Research and Development department at INDO company separated from INDO to form the company Horizons Optical in 2017. For these reasons, it is important to describe publications related to the design of progressive lenses by both companies.

Two theses on progressive lenses have come out of INDO during recent decades: [17] and [19]. Both of these theses tackled the problem of progressive lenses using optimization methods and splines, but neither used AMPL software or interior point algorithms.

The master's thesis [16] introduces the use of AMPL modeling language used with interior point solvers. The proposed model uses polynomials. Although splines are cited in the conclusions, they were not implemented yet.

The master's thesis [47] forms the groundwork of this thesis. It uses splines and interior point methods, and is from the same author of this thesis. The master's thesis [47] is described in more detail in the Introduction, see Chapter 1 and below, in Section 2.2.6.

INDO has produced two patents, [18] and [15], that are particularly relevant to this thesis. Patent [15] describes a new method for distributing the astigmatism aberrations accross the surface of the lens in order to decrease the aberrations inside the frame. Specifically, this method increases aberrations (astigmatism) below the bottom part of the frame allowing lateral aberrations of the lens (lateral astigmatism) to be decreased. The author of this thesis, Casanellas, contributed to the elaboration of the patent [15] before starting the research of this thesis.

Patent [18] describes a method for designing progressive lenses that consists of defining the progressive surface on the back surface (convex) of a lens, and moving it to the front surface while keeping the same optical properties. With free-form technology, the progressive surface must be on the front surface (concave). The method described in this patent allows this thesis to develop and define solely the back (rear) surface of the progressive lens. Patent [18] is described in more detail in Section 3.1 (11).

2.2.6 Previous work of this thesis

The master's thesis [47] upon which this thesis is based is by the same author. In this publication splines and interior point methods were used. Using a set of 9 problem instances in Cartesian coordinates, only 5 out 9 problems converged with fewer than 3000 iterations. LOQO solver took between 10 minutes and 33 minutes to converge. In this new thesis, the Cartesian coordinate model has been improved considerably so that all problems calculated using LOQO converged in fewer than 200 iterations and less than 3 minutes (using the same computer and a new Cartesian coordinate model). It also presents a new spherical coordinate model.

2.3 Contributions

The author's contributions of this thesis are:

- Articles
  - G. Casanellas, J. Castro, Using interior point solvers for optimizing progressive lens models with spherical coordinates, *Optimization and Engineering*, 2019. https://doi.org/10.1007/s11081-019-09480-z.
- Patents
  - G.Casanellas, P.Artús, T.Vilajona, Method for Optimising a Progressive Ophthalmic lens and Method for Producing Same, PCT ES2018070321, WO2018193147, Assignee: Horizons Optical S.L.U, 2017.
- Scientific conferences:
  - G. Casanellas, J. Castro, J.C. Dürsteler, Modelling and optimization of progressive addition lenses, in The XXXV Congreso Nacional de Estadística e Investigación Operativa (SEIO), Pamplona, Spain, May 2015.
  - G. Casanellas , J. Castro, Modelling and Optimization of Progressive Lenses, in the 28th European Conference on Operational Research (EURO), Poznan, Poland, July 2016. Invited presentation.
- Informative conferences
  - G. Casanellas, J. Castro, P. Artús, Design of progressive lenses by optimization techniques, BGSMath Workshop Maths for Industry 4.0, Universitat Pompeu Fabra, 19 February 2018. Invited presentation.
- Competitions
  - G. Casanellas. Selected candidate for the institutional final of the UPC "Present your thesis in four minutes". UPC and Catalan Foundation for Research and Innovation (FCRi). May 17h 2018.
- Informative article
  - Do the progressive lenses really satisfy the Minkwitz theorem? Strategies to go beyond the Minkwitz theorem. Innovation Department of Horizons Optical. MAFO - Ophthalmic Labs & Industry, Volume 15, pp 10-17, 1/2019. ISSN 1614-1598 66527. (*)
    (*)Although they are not mentioned in the publication, the authors of this article are G.Casanellas and P.Artús.

Chapter 3

Model description

3.1 Problem description

As stated in the introduction, the aim of this thesis is to calculate a progressive surface, where the upper part has less power than the lower part (see Figure 1.2 right, page 2). In the calculations for this surface, only the surface power and the surface astigmatism are considered. However, in a later step (which is confidential and does not form part of this thesis) the wavefront, optical power and optical astigmatism are included.

The *front surface* of a lens is the surface that is farther from the eye, while the *back surface* is the surface of the lens that is closer to the eye. In this research project, the progressive surface of the lens is the front surface, while the back surface is spherical (see Figure 3.1 left). Using wavefront tracing, the progressive surface is moved from the front to the back surface of the lens while retaining the same optical properties, and adding optical astigmatism and optical power calculations.

The calculations of wavefront tracing are not part of this thesis. The process described in Figure 3.1 is registered in patent [18], which was filed some years before work began on this thesis.

When this new progressive lens is calculated, see the dashed line in Figure 3.1 (right), a *semi-finished spherical lens* will be cut and polished using free-form generating and polishing commercial equipment. A *semi-finished spherical lens* is a lens with a spherical surface on the front side and a rough, as yet uncut, back surface. Information about professional free-form equipment can be found at [38] and [39].

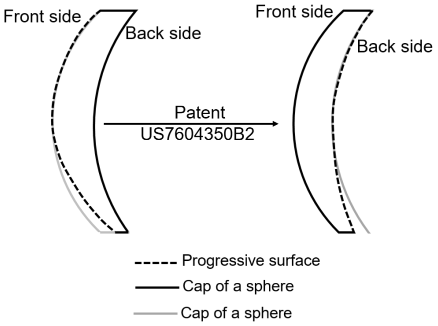


Figure 3.1: The progressive lens calculated in this thesis (left) and the progressive lens calculated in a subsequent step (right).

Although the lenses are rather rounded, free-form generators require a description of a square surface, that is bigger than the (rounded) lens and is expressed in Cartesian coordinates. Therefore, the optimization problem will compute a square surface (not a rounded one, as might be expected).

From this point forward, the progressive surface that will be calculated is the front side of the left image in Figure 3.1, where the upper part is the *far region* and the lower part is the *near region.*

3.2 Features of the progressive surface

In order to have stable far and near regions, these zones must have stable power, and near zero astigmatism. The only surface that has these two optical properties is a sphere, and consequently, far and near regions should be similar to the surfaces of two spheres, where the radius of the sphere of the near zone is smaller than that of the top part. The right image in Figure 1.2 shows that the upper part of the lens is very similar to a spherical cap. However, the lower part of the lens is more curved to increase its power (in the near region). Locally it is also similar to the another spherical cap of a smaller radius. The orientation in the space of the upper sphere is fixed (by the constraints of the problem), and the orientation of the lower spherical cap is such that the surface is continuous in the corridor with the minimum level of astigmatism.

The goal of this thesis is to build a progressive surface having (1) the minimum astigmatism (aberrations); (2) the requested power in the far region; (3) the requested power in the near region; (4) and such that the astigmatism in the corridor, in the far, and in the near regions must be less than a certain threshold value. As stated in the introduction, the astigmatism is an aberration, and it must be minimized. By power we mean the optical power of the lens, which must be equal to a predefined value for the far region and another greater value for the near region. The main parameters for the model are thus the powers of the far and near regions, the shape of these regions, and the maximum permitted levels of astigmatism in the far region, corridor and near region.

3.3 Defining astigmatism and power

The power and the astigmatism of a lens are defined as follows:

Definition 1. *Let be k_1 and k_2 the two principal curvatures at a given point of a surface, then*

$$\begin{aligned} Astigmatism &= (\mu - 1)|k_1 - k_2|\ , \\ Power &= (\mu - 1)\frac{k_1 + k_2}{2}\ , \end{aligned} \tag{3.1}$$

where μ is the refractive index of the material.

If we express the curvatures in m^{-1} (inverse of meters), then the astigmatism and the power are expressed in diopters ($1D = 1m^{-1}$).

The principal curvatures k_1 and k_2 at a given point of the lens surface will be computed considering the following general parametrization of the lens:

$$\begin{aligned} \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ (u, v) &\longmapsto \overrightarrow{p}(u, v) = (x(u, v), y(u, v), z(u, v)). \end{aligned} \tag{3.2}$$

From differential geometry [40] and [42], k_1 and k_2 are the eigenvalues of the Weingarten application defined by the matrix

$$A = \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1}, \tag{3.3}$$

where E, F, G and e, f, g are the coefficients of the first and second fundamental forms. They are expressed as

$$\begin{array}{lll} E = \overrightarrow{p_u} \cdot \overrightarrow{p_u} & F = \overrightarrow{p_u} \cdot \overrightarrow{p_v} & G = \overrightarrow{p_v} \cdot \overrightarrow{p_v} \\ e = \overrightarrow{n} \cdot \overrightarrow{p_{uu}} & f = \overrightarrow{n} \cdot \overrightarrow{p_{uv}} & g = \overrightarrow{n} \cdot \overrightarrow{p_{vv}} \ , \end{array} \tag{3.4}$$

where $\cdot$ is the dot product, $\overrightarrow{n}$, defined as

$$\overrightarrow{n} = \frac{\overrightarrow{p_u} \times \overrightarrow{p_v}}{|\overrightarrow{p_u} \times \overrightarrow{p_v}|}, \tag{3.5}$$

is the normal vector to the surface (where $\times$ is the cross or vector product of two vectors in $\mathbb{R}^3$), and the subscripts u, v denote the partial derivative with respect to u or v, that is,

$$\overrightarrow{p_u} = \left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right) \qquad \overrightarrow{p_v} = \left(\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right). \tag{3.6}$$

The second-order partial derivatives are thus

$$\overrightarrow{p_{uu}} = \left(\frac{\partial^2 x}{\partial u^2}, \frac{\partial^2 y}{\partial u^2}, \frac{\partial^2 z}{\partial u^2} \right), \ \overrightarrow{p_{uv}} = \left(\frac{\partial^2 x}{\partial u \partial v}, \frac{\partial^2 y}{\partial u \partial v}, \frac{\partial^2 z}{\partial u \partial v} \right), \ \overrightarrow{p_{vv}} = \left(\frac{\partial^2 x}{\partial v^2}, \frac{\partial^2 y}{\partial v^2}, \frac{\partial^2 z}{\partial v^2} \right). \tag{3.7}$$

From (3.3), A is a 2×2 matrix, which can be rewritten as

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix},$$

and then its eigenvalues can be computed by

$$P(\lambda) = \det(A - \lambda I) = \begin{vmatrix} a_{1,1} - \lambda & a_{1,2} \\ a_{2,1} & a_{2,2} - \lambda \end{vmatrix} = \lambda^2 - \mathrm{tr}(A)\lambda + \det(A), \tag{3.8}$$

where

$$\begin{aligned} \mathrm{tr}(A) =& a_{1,1} + a_{2,2} \\ \det(A) =& a_{1,1}a_{2,2} - a_{1,2}a_{2,1} \end{aligned} \tag{3.9}$$

are respectively the trace and the determinant of the matrix A. The solutions to $P(\lambda) = 0$ are

$$\lambda = \frac{\mathrm{tr}(A) \pm \sqrt{\mathrm{tr}(A)^2 - 4\det(A)}}{2} = \frac{\mathrm{tr}(A)}{2} \pm \sqrt{\left(\frac{\mathrm{tr}(A)}{2} \right)^2 - \det(A)}. \tag{3.10}$$

Denoting $K = \det(A)$ and $H = \frac{1}{2}\mathrm{tr}(A)$, k_1 and k_2 are defined as

$$k_1, k_2 = H \pm \sqrt{(H^2 - K)} \tag{3.11}$$

where H and K are named the *mean curvature* and the *Gaussian curvature*, respectively. From (3.3) we obtain

$$\begin{aligned} K =& \det(A) = \frac{eg - f^2}{EG - F^2} \\ H =& \frac{1}{2}\mathrm{tr}(A) = \frac{eG - 2fF + gE}{2(EG - F^2)} \end{aligned} \tag{3.12}$$

Using (3.1) and (3.11) we obtain:

$$Pow(x,y) = (\mu - 1) \cdot H(x,y) \tag{3.13}$$

$$Ast(x,y) = -2 \cdot (\mu - 1) \cdot \sqrt{H(x,y)^2 - K(x,y)}\ . \tag{3.14}$$

Thus, we have a method for calculating the power and astigmatism at each point on the lens.

3.4 The inverse problem

In the previous section we showed that given a surface described in Cartesian coordinates, the values for power and astigmatism can be computed for the entire surface.

The inverse problem [41] (that is, reconstructing the entire surface from some given values for power and astigmatism) is not trivial, and a solution cannot be guaranteed. That is, given a distribution of power and astigmatism, in most cases there is no surface that fits them and, even if there were, it would not be easy to calculate. Indeed, the goal of this research work is to solve this inverse problem formulated as an optimization problem. To guarantee the feasibility of the optimization problem, values of power and astigmatism are not given for the entire surface, but only for some regions. In particular, the values for target power are given for the far and near regions and a threshold for astigmatism is given in most regions of the lens. This optimization model will be fully detailed in Chapter 4. 4.5 (page 27).

Note that this inverse problem is not often explained for reasons of confidentiality within private companies. Among the few available references we find [27, page 248].

3.5 Modeling the progressive lenses surface using B-splines

In order to calculate the power and the astigmatism, we need to calculate the partial derivatives of the surface. Using B-splines basis allows us to calculate partial derivatives analytically. For this reason, we will define the surface of the lens as a cubic B-spline. Later, in Section 5.6 (page 91) we will see that B-splines of degrees 4, 5 and 6 do not offer any benefit.

In this section we have assumed that we have a Cartesian coordinate model, but the B-spline basis can be similarly used with spherical coordinates.

In Cartesian coordinates, the expression (3.2) (page 12) is expressed as:

$$\begin{aligned} \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ (u,v) &\longmapsto \overrightarrow{p}(u,v) = (x(u,v),\ y(u,v),\ z(u,v)) = (u,\ v,\ z(u,v)) \end{aligned} \tag{3.15}$$

that came from the definition of a surface as the function:

$$\begin{aligned} z : \mathbb{R}^2 &\longrightarrow \mathbb{R} \\ (x, y) &\rightarrow z(x, y). \end{aligned} \tag{3.16}$$

We have assumed a square lens surface with each side of $2 \cdot r$ length. We have also assumed the following parameters:

- $$(x_i, y_j) \in [-r, r] \times [-r, r], (i, j) \in \mathcal{G} = \{1, \ldots, n\} \times \{1, \ldots, n\} \tag{3.17}$$

  , is a grid of points (in mm) used to define the lens in Cartesian coordinates, where n denotes the number of points for each dimension of the grid. The grid is defined such that $(x_{\frac{1+n}{2}}, y_{\frac{1+n}{2}}) = (0, 0)$.

- $$(x'_{i'}, y'_{j'}) \in [-r, r] \times [-r, r], \quad (i', j') \in \mathcal{G}' = \{1, \ldots, o\} \times \{1, \ldots, o\}, \tag{3.18}$$

  is another grid of points (mm), with $\mathcal{G}'$ much coarser than $\mathcal{G}$ (i.e., $o \ll n$), where o is the number of points used in the definition of a B-spline whose coefficients are the variables of the optimization model (see equations 3.19 and 3.20). Usually the $(x'_{i'}, y'_{j'})$ are called *knots*.

The variables of the optimization problem are the coefficients of a third-degree B-spline surface, as defined in [44, page 100]. These coefficients are denoted as

$$\mathbb{R} \ni c(x'_{i'}, y'_{j'}) \geq 0, \qquad (i', j') \in \mathcal{G}'. \tag{3.19}$$

Using the B-spline we define the z of the surface for the grid $\mathcal{G}$ as

$$z(x_i, y_j) = \sum_{i'=1}^{o} \sum_{j'=1}^{o} c(x'_{i'}, y'_{j'}) B^3_{i'}(x_i) B^3_{j'}(y_j), \quad (i, j) \in \mathcal{G}, \tag{3.20}$$

where $B^3_{i'}(x_i)$ and $B^3_{j'}(y_j), (i, j) \in \mathcal{G}, (i', j') \in \mathcal{G}'$, are the 1-dimensional third-degree B-splines basis defined in [44, page 100].

The optimization problem has only o^2 variables, but the surface can be evaluated in n^2 points (where $o \ll n$). Indeed, this is the second reason for using a B-spline in the model. The first reason is to provide a way of calculating its partial derivatives.

Properties of the cubic B-splines:

- Nonnegativity (a lens surface is always positive)
- Convex hull property
- Smoothness
- $z(i, j)$ is a C^2 class function

3.6 Detailing the definition of the B-splines

In this section the basis of third-degree B-splines is explained in depth. The surface is modeled using the expression (3.18), (3.19) and (3.20) (page 15). The parameters used for the grid of *knots* defined at (3.18) are:

$$x'_{i'} = y'_{i'} = -72 + 4 \cdot i, \quad i = 1, .., o, \;\; o = 35 \tag{3.21}$$

i.e., from -68 mm to 68 mm each 4 mm step. Note that this grid will be used later to calculate square lens surfaces. The progressive lenses calculated will be square surfaces $[-54mm, 54mm] \times [-54mm, 54mm]$. Points from 54 mm to 68 mm are used to force the border to be smooth.

The $B^3_{i'}(x_i)$ and $B^3_{j'}(y_j)$ are the third-degree polynomials constructed piecewise, as a product of second-degree polynomials, that are defined using first-degree polynomials. Similarly first-degree piecewise functions are defined using zero-degree piecewise functions.

The B^0, B^1, B^2 and B^3, that is to say the zero-degree, first-degree, two-degree and third-degree B-spline 1-dimensional basis are defined as:

$$B^0_{i'}(x_i) = \begin{cases} 1 & if \quad x'_{i'-1} \leq (x_i) \leq x'_{i'} \\ 0 & else \end{cases} \tag{3.22}$$

$$B^0_{j'}(y_i) = \begin{cases} 1 & if \quad y'_{j'-1} \leq (y_j) \leq y'_{j'} \\ 0 & else \end{cases} \tag{3.23}$$

$$B^1_{i'}(x_i) = \frac{x_i - x'_{i'-1}}{x'_{i'} - x'_{i'-1}} \cdot B^0_{i'}(x_i) + \frac{x'_{i'+1} - x_i}{x'_{i'+1} - x'_{i'}} \cdot B^0_{i'+1}(x_i). \tag{3.24}$$

$$B^1_{j'}(y_j) = \frac{y_j - y'_{j'-1}}{y'_{j'} - y'_{j'-1}} \cdot B^0_{j'}(y_j) + \frac{y'_{j'+1} - y_j}{y'_{j'+1} - y'_{j'}} \cdot B^0_{j'+1}(y_j). \tag{3.25}$$

$$B^2_{i'}(x_i) = \frac{x_i - x'_{i'-1}}{x'_{i'+1} - x'_{i'-1}} \cdot B^1_{i'}(x_i) + \frac{x'_{i'+2} - x_i}{x'_{i'+2} - x'_{i'}} \cdot B^1_{i'+1}(x_i). \tag{3.26}$$

$$B^2_{j'}(y_j) = \frac{y_j - y'_{j'-1}}{y'_{j'+1} - y'_{j'-1}} \cdot B^1_{j'}(y_j) + \frac{y'_{j'+2} - y_j}{y'_{j'+2} - y'_{j'}} \cdot B^1_{j'+1}(y_j). \tag{3.27}$$

$$B^3_{i'}(x_i) = \frac{x_i - x'_{i'-1}}{x'_{i'+2} - x'_{i'-1}} \cdot B^2_{i'}(x_i) + \frac{x'_{i'+3} - x_i}{x'_{i'+3} - x'_{i'}} \cdot B^2_{i'+1}(x_i). \tag{3.28}$$

$$B^3_{j'}(y_j) = \frac{y_j - y'_{j'-1}}{y'_{j'+2} - y'_{j'-1}} \cdot B^2_{j'}(y_j) + \frac{y'_{j'+3} - y_j}{y'_{j'+3} - y'_{j'}} \cdot B^2_{j'+1}(y_j). \tag{3.29}$$

Over the next pages, $x'_{i'}$ with $i' = 10, 11, 12$ will be used. We may note that, according to (3.28), to calculate B^3_{10} (using $i' = 10$), not only x'_{10} will be used but also other points of the grid of *knots* such as $x'_{i'-1}$, $x'_{i'}$, $x'_{i'+2}$ and $x'_{i'+3}$.

Some points are calculated as an example:

$$\begin{aligned} x'_9 &= -72 + (4 \cdot 9) = -36 \; mm \\ x'_{10} &= -72 + (4 \cdot 10) = -32 \; mm \\ x'_{11} &= -72 + (4 \cdot 11) = -28 \; mm \\ x'_{12} &= -72 + (4 \cdot 12) = -24 \; mm \\ x'_{13} &= -72 + (4 \cdot 13) = -20 \; mm. \end{aligned}$$

In Figures 3.2, 3.3, 3.4 and 3.5 the $B^0_{i'}(x_i)$, $B^1_{i'}(x_i)$, $B^2_{i'}(x_i)$ and $B^3_{i'}(x_i)$ B-splines functions are shown for $i' = 10, 11, 12$, and $x'_{11} = -28\ mm, x'_{12} = -24\ mm, x'_{13} = -20\ mm$. We remark that $B^0_{i'}(x_i)$ is 1 in one interval and 0 in the others.

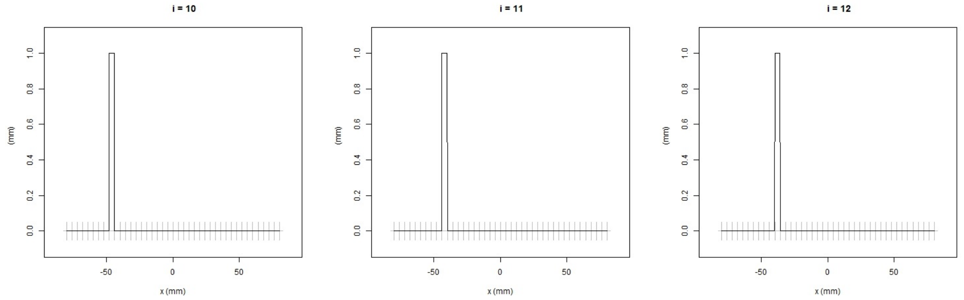


Figure 3.2: $B^0_{i'}$ for $i' = 10, 11, 12$, $x'_{10} = -32\ mm, x'_{11} = -28\ mm, x'_{12} = -24\ mm$.

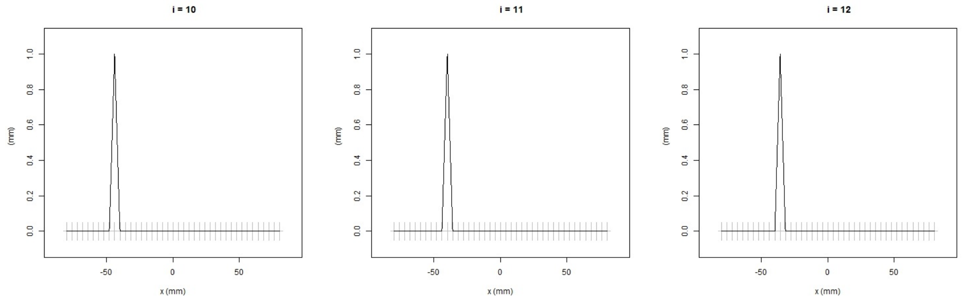


Figure 3.3: $B^1_{i'}$ for $i' = 10, 11, 12$, $x'_{10} = -32\ mm, x'_{11} = -28\ mm, x'_{12} = -24\ mm$.

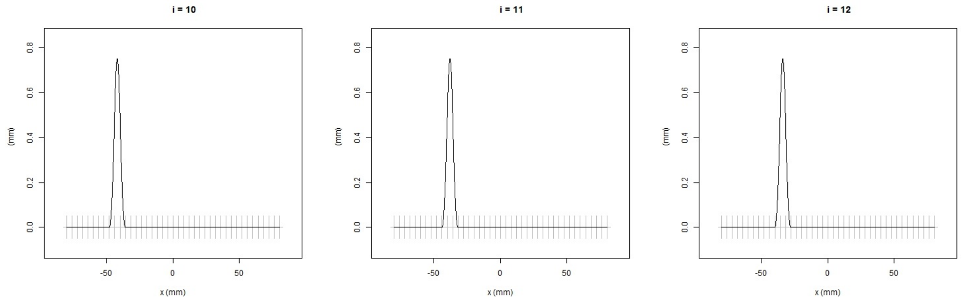


Figure 3.4: $B^2_{i'}$ for $i' = 10, 11, 12$, $x'_{10} = -32\ mm, x'_{11} = -28\ mm, x'_{12} = -24\ mm$.

3.7 Calculating the derivatives of the B-splines

In order to calculate the power and the astigmatism as defined in equation (3.1) and using (3.2) - (3.12) we need to calculate the second order partial derivatives of $z(x_i, y_j)$ as can be seen in (3.4). In addition, (in the next section) we will seek to minimize the astigmatism

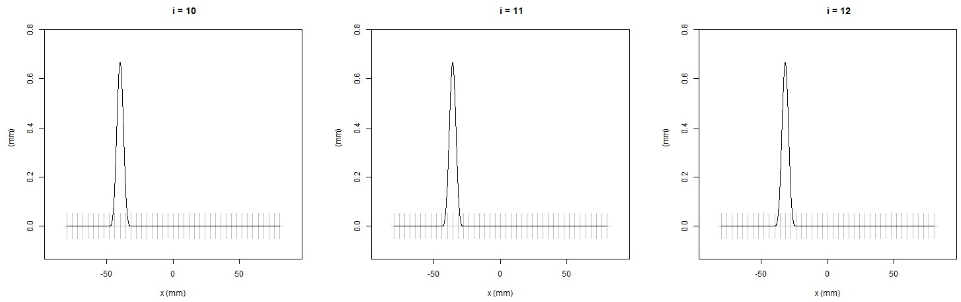


Figure 3.5: $B^3_{i'}$ for $i' = 10, 11, 12$, $x'_{10} = -32\ mm, x'_{11} = -28\ mm, x'_{12} = -24\ mm$.

gradient and the power gradient. Consequently, we will need to calculate the third order partial derivatives of $z(x_i, y_j)$.

The definition of $z(x_i, y_j)$ depends on $B^3_{i'}(x_i)$, which is a recursive formula: the calculations of $B^3_{i'}(x_i)$ depends on the calculations of $B^2_{i'}(x_i)$, the calculations of $B^2_{i'}(x_i)$ depends on the calculations of $B^1_{i'}(x_i)$, which depends on the calculations of $B^0_{i'}(x_i)$. Consequently we must calculate the first derivative of $B^1_{i'}(x_i)$, $B^2_{i'}(x_i)$ and $B^3_{i'}(x_i)$, the second derivative of $B^2_{i'}(x_i)$ and $B^3_{i'}(x_i)$ and the third derivative of $B^3_{i'}(x_i)$, for $i = 1, ..., n$. The same applies to the other dimension $y_1, ..., y_n$.

Here are the calculations for these derivatives. The first derivative of $B^1_{i'}$ is a direct calculation:

$$\frac{\partial B^1_{i'}}{\partial x}(x_i) = \frac{1}{x'_{i'} - x'_{i'-1}} \cdot B^0_{i'}(x_i) - \frac{1}{x'_{i'+1} - x'_{i'}} \cdot B^0_{i'+1}(x_i). \tag{3.30}$$

$$\frac{\partial B^1_{j'}}{\partial y}(y_j) = \frac{1}{y'_{j'} - y'_{j'-1}} \cdot B^0_{j'}(y_j) - \frac{1}{y'_{j'+1} - y'_{j'}} \cdot B^0_{j'+1}(y_j). \tag{3.31}$$

The other derivatives require some (not very extensive) calculations. The other first derivatives are:

$$\frac{\partial B^2_{i'}}{\partial x}(x_i) = \frac{2}{x'_{i'+1} - x'_{i'-1}} \cdot B^1_{i'}(x_i) - \frac{2}{x'_{i'+2} - x'_{i'}} \cdot B^1_{i'+1}(x_i). \tag{3.32}$$

$$\frac{\partial B^2_{j'}}{\partial y}(y_j) = \frac{2}{y'_{j'+1} - y'_{j'-1}} \cdot B^1_{j'}(y_j) - \frac{2}{y'_{j'+2} - y'_{j'}} \cdot B^1_{j'+1}(y_j). \tag{3.33}$$

$$\frac{\partial B^3_{i'}}{\partial x}(x_i) = \frac{3}{x'_{i'+2} - x'_{i'-1}} \cdot B^2_{i'}(x_i) - \frac{3}{x'_{i'+3} - x'_{i'}} \cdot B^2_{i'+1}(x_i). \tag{3.34}$$

$$\frac{\partial B^3_{j'}}{\partial y}(y_j) = \frac{3}{y'_{j'+2} - y'_{j'-1}} \cdot B^2_{j'}(y_j) - \frac{3}{y'_{j'+3} - y'_{j'}} \cdot B^2_{j'+1}(y_j). \tag{3.35}$$

The second derivatives are:

$$\frac{\partial^2 B^2_{i'}}{\partial x^2}(x_i) = \frac{2}{x'_{i'+1} - x'_{i'-1}} \cdot \frac{\partial B^1_{i'}}{\partial x}(x_i) - \frac{2}{x'_{i'+2} - x'_{i'}} \cdot \frac{\partial B^1_{i'+1}}{\partial x}(x_i). \tag{3.36}$$

$$\frac{\partial^2 B^2_{j'}}{\partial y^2}(y_j) = \frac{2}{y'_{j'+1} - y'_{j'-1}} \cdot \frac{\partial B^1_{j'}}{\partial y}(y_j) - \frac{2}{y'_{j'+2} - y'_{j'}} \cdot \frac{\partial B^1_{j'+1}}{\partial y}(y_j). \tag{3.37}$$

$$\frac{\partial^2 B^3_{i'}}{\partial x^2}(x_i) = \frac{3}{x'_{i'+2} - x'_{i'-1}} \cdot \frac{\partial B^2_{i'}}{\partial x}(x_i) - \frac{3}{x'_{i'+3} - x'_{i'}} \cdot \frac{\partial B^2_{i'+1}}{\partial x}(x_i). \tag{3.38}$$

$$\frac{\partial^2 B^3_{j'}}{\partial y^2}(y_j) = \frac{3}{y'_{j'+2} - y'_{j'-1}} \cdot \frac{\partial B^2_{j'}}{\partial y}(y_j) - \frac{3}{y'_{j'+3} - y'_{j'}} \cdot \frac{\partial B^2_{j'+1}}{\partial y}(y_j). \tag{3.39}$$

And the third derivatives are:

$$\frac{\partial^3 B^3_{i'}}{\partial x^3}(x_i) = \frac{3}{x'_{i'+2} - x'_{i'-1}} \cdot \frac{\partial^2 B^2_{i'}}{\partial x^2}(x_i) - \frac{3}{x'_{i'+3} - x'_{i'}} \cdot \frac{\partial^2 B^2_{i'+1}}{\partial x^2}(x_i). \tag{3.40}$$

$$\frac{\partial^3 B^3_{j'}}{\partial y^3}(y_j) = \frac{3}{y'_{j'+2} - y'_{j'-1}} \cdot \frac{\partial^2 B^2_{j'}}{\partial y^2}(y_j) - \frac{3}{y'_{j'+3} - y'_{j'}} \cdot \frac{\partial^2 B^2_{j'+1}}{\partial y^2}(y_j). \tag{3.41}$$

We may observe that some derivatives as, such as, $\frac{\partial B^0_{i'}}{\partial x}(x_i)$, $\frac{\partial B^0_{i'}}{\partial y}(y_i)$, $\frac{\partial^2 B^1_{i'}}{\partial x^2}(x_i)$, $\frac{\partial^2 B^1_{i'}}{\partial y^2}(y_i)$, $\frac{\partial^3 B^2_{i'}}{\partial x^3}(x_i)$ and $\frac{\partial^3 B^2_{i'}}{\partial y^3}(y_i)$, as well as the cross derivatives, are zero. The cross derivatives are, for example: $\frac{\partial B^0_{i'}}{\partial y}(x_i)$ and $\frac{\partial^2 B^2_{j'}}{\partial x \partial y}(y_j)$

The first, second and third order derivatives appear in Figures 3.6, 3.7, 3.8 for $i = 11$. Recall that $x'_{11} = -28\ mm$.

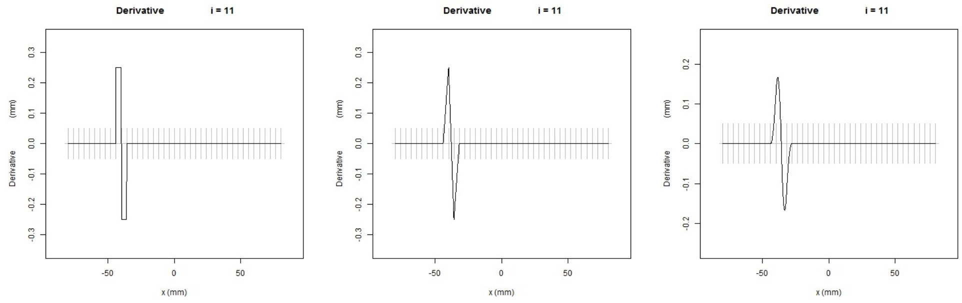


Figure 3.6: First derivatives $\frac{\partial B^1_{i'}}{\partial x}(x_i)$, $\frac{\partial B^2_{i'}}{\partial x}(x_i)$ and $\frac{\partial B^3_{i'}}{\partial x}(x_i)$ for $i = 11$ ($x'_{11} = -28\ mm$).

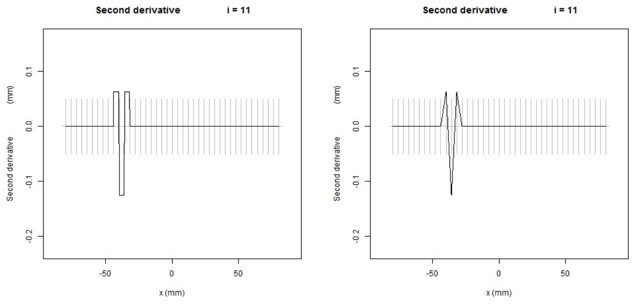


Figure 3.7: Second derivatives $\frac{\partial^2 B^2_{i'}}{\partial x^2}(x_i)$ and $\frac{\partial^2 B^3_{i'}}{\partial x^2}(x_i)$ for $i = 11$ ($x'_{11} = -28\ mm$).

The third-degree B-spline basis functions are two times continuously differentiable and three times differentiable (the third derivative is not continuous, and the fourth derivative is zero). This means that the surface defined in (3.20) allows us to compute its first, second

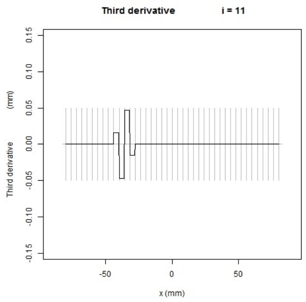


Figure 3.8: Third derivatives $\frac{\partial^3 B^3_{i'}}{\partial x^3}(x_i)$ for $i = 11$ ($x'_{11} = -28\ mm$).

and third derivatives. The first and second derivatives are needed to calculate the power and the astigmatism, for all the n^2 points of the grid $\mathcal{G}$. The power and astigmatism are thus continuous, although they are only evaluated in a grid of n^2 points. The third degree of differentiability allows us to compute the gradients of the astigmatism and power, which will be needed later in the calculation of the objective function (4.26) (page 28).

The next chapter will present the detailed calculations of power and astigmatism in Cartesian coordinates and also some computational results using Cartesian coordinates. The definition of the surface in Cartesian coordinates (and also in spherical coordinates) uses the third-degree B-spline basis defined in this chapter.

Chapter 4

Cartesian coordinates

4.1 Computing power and astigmatism using Cartesian coordinates

From (3.2) (page 12), considering $x(u,v)=u$, $y(u,v)=v$ and using Cartesian coordinates, the surface of a lens can be defined as a function

$$\begin{aligned} z : \mathbb{R}^2 &\longrightarrow \mathbb{R} \\ (x,y) &\rightarrow z(x,y). \end{aligned} \tag{4.1}$$

The main advantage of using Cartesian coordinates is that the expressions $\overrightarrow{p_u}$, $\overrightarrow{p_v}$, $\overrightarrow{p_{uu}}$, $\overrightarrow{p_{uv}}$ and $\overrightarrow{p_{vv}}$ are simplified. The first-order partial derivatives at a given point in Cartesian coordinates are

$$\overrightarrow{p_u} = \left(1, 0, \frac{\partial z(x,y)}{\partial x}\right),\ \overrightarrow{p_v} = \left(0, 1, \frac{\partial z(x,y)}{\partial y}\right)\ , \tag{4.2}$$

while the second-order partial derivatives are given by

$$\overrightarrow{p_{uu}} = \left(0, 0, \frac{\partial^2 z(x,y)}{\partial x^2}\right),\ \overrightarrow{p_{uv}} = \left(0, 0, \frac{\partial^2 z(x,y)}{\partial x \partial y}\right),\ \overrightarrow{p_{vv}} = \left(0, 0, \frac{\partial^2 z(x,y)}{\partial y^2}\right)\ . \tag{4.3}$$

The normal vector to the surface is then

$$\overrightarrow{n} = \frac{\overrightarrow{p_u} \times \overrightarrow{p_v}}{|\overrightarrow{p_u} \times \overrightarrow{p_v}|} = \frac{1}{\sqrt{1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2}}. \tag{4.4}$$

Replacing (4.2) (4.3) (4.4) in the fundamental form (3.4) (page 13) we get:

$$\begin{array}{ll} e = \dfrac{\frac{\partial^2 z}{\partial x^2}}{\sqrt{1+\left(\frac{\partial z}{\partial x}\right)^2+\left(\frac{\partial z}{\partial y}\right)^2}} & E = 1 + \frac{\partial z}{\partial x}\frac{\partial z}{\partial x} \\ f = \dfrac{\frac{\partial^2 z}{\partial xy}}{\sqrt{1+\left(\frac{\partial z}{\partial x}\right)^2+\left(\frac{\partial z}{\partial y}\right)^2}} & F = \frac{\partial z}{\partial x}\frac{\partial z}{\partial y} \\ g = \dfrac{\frac{\partial^2 z}{\partial y^2}}{\sqrt{1+\left(\frac{\partial z}{\partial x}\right)^2+\left(\frac{\partial z}{\partial y}\right)^2}} & G = 1 + \frac{\partial z}{\partial y}\frac{\partial z}{\partial y}. \end{array}$$

From (3.12) (page 14), we can express the *mean curvature* as:

$$H(x,y) = \frac{\frac{\partial^2 z}{\partial x^2} \cdot (1 + (\frac{\partial z}{\partial y})^2) - 2 \cdot \frac{\partial^2 z}{\partial xy} \cdot \frac{\partial z}{\partial x} \cdot \frac{\partial z}{\partial y} + \frac{\partial^2 z}{\partial y^2} \cdot (1 + (\frac{\partial z}{\partial x})^2)}{2 \cdot \left(1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2\right)^{3/2}} = \frac{1}{2}(k_1 + k_2) \tag{4.5}$$

and the *Gaussian curvature* as:

$$K(x,y) = \frac{\frac{\partial^2 z}{\partial x^2} \cdot \frac{\partial^2 z}{\partial y^2} - \left(\frac{\partial^2 z}{\partial xy}\right)^2}{\left(1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2\right)^2} = k_1 \cdot k_2 \ . \tag{4.6}$$

Using (3.13) and (3.14) (page 14) and (4.5) and (4.6) we can rewrite in a single formula:

$$Pow(x,y) = (\mu - 1) \cdot \frac{\frac{\partial^2 z}{\partial x^2} \cdot (1 + (\frac{\partial z}{\partial y})^2) - 2 \cdot \frac{\partial^2 z}{\partial xy} \cdot \frac{\partial z}{\partial x} \cdot \frac{\partial z}{\partial y} + \frac{\partial^2 z}{\partial y^2} \cdot (1 + (\frac{\partial z}{\partial x})^2)}{2 \cdot \left(1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2\right)^{3/2}} \tag{4.7}$$

and

$$Ast(x,y) = -2 \cdot (\mu - 1) \cdot$$
$$\sqrt{\left[\frac{\frac{\partial^2 z}{\partial x^2} \cdot (1 + (\frac{\partial z}{\partial y})^2) - 2 \cdot \frac{\partial^2 z}{\partial xy} \cdot \frac{\partial z}{\partial x} \cdot \frac{\partial z}{\partial y} + \frac{\partial^2 z}{\partial y^2} \cdot (1 + (\frac{\partial z}{\partial x})^2)}{2 \cdot \left(1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2\right)^{3/2}}\right]^2 - \frac{\frac{\partial^2 z}{\partial x^2} \cdot \frac{\partial^2 z}{\partial y^2} - \left(\frac{\partial^2 z}{\partial xy}\right)^2}{\left(1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2\right)^2}} \ . \tag{4.8}$$

And then

$$\begin{aligned} Ast(x,y) = -2 \cdot \sqrt{\left[Pow(x,y)\right]^2 - (\mu - 1)^2 \cdot K(x,y)} = \\ -2 \cdot (\mu - 1) \cdot \sqrt{\left[\frac{Pow(x,y)}{\mu - 1}\right]^2 - K(x,y)} \ . \end{aligned} \tag{4.9}$$

We have thus calculated the power and astigmatism using Cartesian coordinates.

4.2 Simplifying the expressions of the astigmatism

The expression of $Ast(x,y)$ in (4.8) can be simplified by making two transformations. The first arrangement uses $Ast(x,y)^2$ instead of $Ast(x,y)$ in order to eliminate the square root. We will see later in Section 4.5 in equations (4.26) and (4.29) (page 29) that the expression of $Ast(x,y)$ is only used in our model squared. The new expression is therefore:

$$Ast(x,y)^2 = 4 \cdot (\mu - 1)^2 \cdot$$
$$\left[\frac{\left(\frac{\partial^2 z}{\partial x^2} \cdot \left(1 + \left(\frac{\partial z}{\partial y}\right)^2\right) - 2 \cdot \frac{\partial^2 z}{\partial xy} \cdot \frac{\partial z}{\partial x} \cdot \frac{\partial z}{\partial y} + \frac{\partial^2 z}{\partial y^2} \cdot \left(1 + \left(\frac{\partial z}{\partial x}\right)^2\right)\right)^2}{4 \cdot \left(1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2\right)^3} - \frac{\frac{\partial^2 z}{\partial x^2} \cdot \frac{\partial^2 z}{\partial y^2} - \left(\frac{\partial^2 z}{\partial xy}\right)^2}{\left(1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2\right)^2}\right] . \tag{4.10}$$

The second arrangement is to consider the greatest common divisor (gcd) in order to have a division between polynomials. The expression (4.10) can be simplified as:

$$Ast(x,y)^2 = 4\cdot(\mu-1)^2\cdot\left[\frac{Aux(x,y)}{4\cdot\left(1+\left(\frac{\partial z}{\partial x}\right)^2+\left(\frac{\partial z}{\partial y}\right)^2\right)^3}\right]. \tag{4.11}$$

where:

$$\begin{aligned}Aux(x,y) =&\left(\frac{\partial^2 z}{\partial x^2}\cdot(1+(\frac{\partial z}{\partial y})^2)-2\cdot\frac{\partial^2 z}{\partial xy}\cdot\frac{\partial z}{\partial x}\cdot\frac{\partial z}{\partial y}+\frac{\partial^2 z}{\partial y^2}\cdot(1+(\frac{\partial z}{\partial x})^2)\right)^2-\\ &4\cdot\left(\frac{\partial^2 z}{\partial x^2}\cdot\frac{\partial^2 z}{\partial y^2}-(\frac{\partial^2 z}{\partial xy})^2\right)\cdot\left(1+(\frac{\partial z}{\partial x})^2+(\frac{\partial z}{\partial y})\right)\end{aligned} \tag{4.12}$$

We thus have a new expression of $Ast(x,y)^2$.

4.3 Example: a sphere

Let us consider a thin lens (without center thickness) whose surface is part of a sphere of radius R. We will calculate the power and the astigmatism of the sphere using the preceding formulas.

The parametrization in Cartesian coordinates is:

$$\begin{aligned}&z:\mathbb{R}^2\longrightarrow\mathbb{R}\\ &(x,y)\rightarrow z(x,y)=-\sqrt{R^2-x^2-y^2}\end{aligned} \tag{4.13}$$

The previous formulation corresponds to a sphere of radius R centered at $(0,\ 0,\ 0)$. As we consider the negative root, we take into account the inferior part of the sphere shown in blue in Figure 4.1.

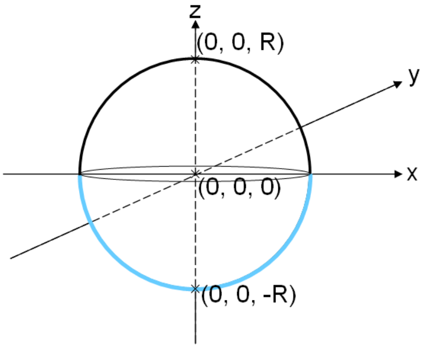


Figure 4.1: Sphere of radius R, centered at $(0,\ 0,\ 0)$. In blue, the inferior part of the sphere modeled in (4.13).

In order to calculate the astigmatism and the power, we need to calculate the first and second derivatives of (4.13):

$$\begin{aligned}\frac{\partial z(x,\ y)}{\partial x} =& \frac{x}{\sqrt{R^2-x^2-y^2}}\\ \frac{\partial z(x,\ y)}{\partial y} =& \frac{y}{\sqrt{R^2-x^2-y^2}},\end{aligned} \tag{4.14}$$

$$\begin{aligned}\frac{\partial^2 z(x,\ y)}{\partial x^2} =& \frac{R^2-y^2}{(R^2-x^2-y^2)^{3/2}}\\ \frac{\partial^2 z(x,\ y)}{\partial xy} =& \frac{x\cdot y}{(R^2-x^2-y^2)^{3/2}}\\ \frac{\partial^2 z(x,\ y)}{\partial y^2} =& \frac{R^2-x^2}{(R^2-x^2-y^2)^{3/2}}.\end{aligned} \tag{4.15}$$

Using the expressions (4.7) and (4.8) and also (4.14) and (4.15) and doing large calculations we finally obtain:

$$\begin{aligned}Pow(x,\ y) =& \frac{\mu-1}{R} \qquad \forall(x,y) \text{ where } x^2-y^2 \leq R^2\\ Ast(x,\ y) =& 0.0 \qquad \forall(x,y) \text{ where } x^2-y^2 \leq R^2.\end{aligned} \tag{4.16}$$

This means that a sphere of radius R has zero astigmatism and constant power $\frac{\mu-1}{R}$ for the entire surface. For example, if we want to create a sphere of power 5 D using a 1.6 refractive index, we will need a radius of $(\mu-1)/5\ D = (1.6-1)/5\ D = 0.12m = 120\ mm$.

4.4 Nonconvexity in Cartesian coordinates

The expressions of power and astigmatism expressed in Cartesian coordinates are clearly nonlinear and, when used in the formulation of the constraints of an optimization problem, give rise to a nonconvex set of solutions and a nonconvex feasible region (this was also noted in [33]). To illustrate this fact, let us first consider the following Examples 1 and 2 which correspond to a lens with $P_F = P_N$ (i.e., with the same power in the far and near regions).

Example 1. *Given the function (defined on a grid of points in $\mathbb{R}^2$)*

$$\begin{aligned} z : G \subseteq \mathbb{R}^2 &\longrightarrow \mathbb{R}\\ (x_i, y_i) &\longrightarrow z(x_i, y_i) \qquad i = 1, \ldots, n,\end{aligned}$$

we find the surface $z(x, y)$ solution to the following problem:

$$\begin{array}{lllll} \displaystyle\min_{z(x_i,y_i)} & 1 & & & \\ \text{subject to} & Ast(x_i,y_i) & = 0 & & \forall(x_i,y_i)\in G\\ & Pow(x_i,y_i) & = P & \forall(x_i,y_i)\in G & \\ & z(0,0) & = -\frac{\mu-1}{P} & & \\ & \left.\frac{\partial z(x,y)}{\partial x}\right|_{x=0,y=0} & = 0 & & \\ & \left.\frac{\partial z(x,y)}{\partial y}\right|_{x=0,y=0} & = 0, & & \end{array} \tag{4.17}$$

where $Pow(x_i, y_i)$ *and* $Ast(x_i, y_i)$ *are defined in* (4.7) *and* (4.8)*,* $P = 5D$*,* $\mu = 1.6$*, and the points* (x_i, y_i) *satisfy* $-45mm \le x_i, y_i \le 45mm$*.*

The solution to (4.17) is a sphere of radius $\hat{R}$ *centered at point* $(0, 0, 0)$*, such that*

$$\hat{R} = \frac{\mu - 1}{P} = \frac{1.6 - 1}{5\ D} = 0.120\ m = 120.0\ mm$$

i.e.

$$z(x_i, y_i) = -\sqrt{R^2 - x_i^2 - y_i^2}\ . \tag{4.18}$$

Being more precise, the solution to (4.17) is a spherical cap of the previously defined sphere such that $-45mm \le x_i, y_i \le 45mm$. This sphere is represented in Figure 4.2 and the spherical cap is shown in Figure 4.3.

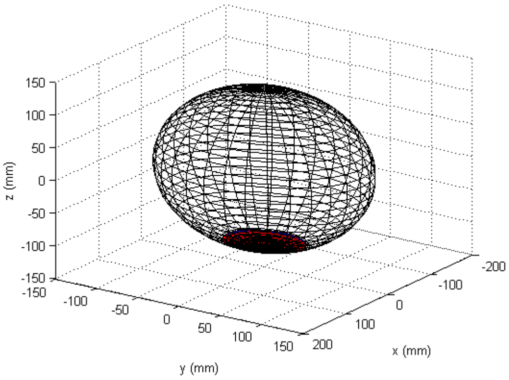


Figure 4.2: In gray color a sphere of radius 120 mm, centered at $(0,\ 0,\ 0)$. In red color, the solution of problem 1, that is the inferior part of the previous sphere where $-45\ mm \le x_i,\ y_i \le 45\ mm$.

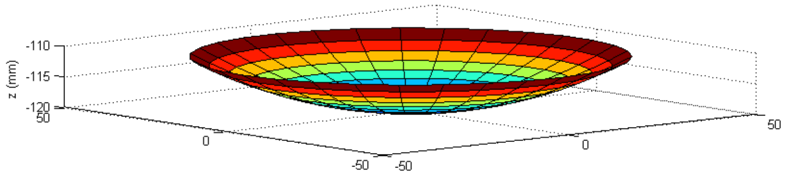


Figure 4.3: Solution of problem 1: the inferior part of a sphere of radius 120 mm, centered at $(0,\ 0,\ 0)$ where $-45\ mm \le x_i,\ y_i \le 45\ mm$.

We remark that the constraint:

$$z(0,\ 0) = -\frac{\mu - 1}{P} = -\hat{R} = \frac{1.6 - 1}{5\ D} = 0.120\ m = 120.0\ mm \tag{4.19}$$

fixes that the sphere goes throw the point $(0,\ 0, -\hat{R})$. In addition, the constraints:

$$\begin{aligned} \left.\frac{\partial z(x,\ y)}{\partial x}\right|_{x=0,\ y=0} &= 0.0 \\ \left.\frac{\partial z(x,\ y)}{\partial y}\right|_{x=0,\ y=0} &= 0.0 \end{aligned} \tag{4.20}$$

indicate that the normal of the surface (lens) at point $(x = 0,\ y = 0)$ must be perpendicular to the axis z. We have added these three constraints in order to obtain a single solution (i.e. to fix the surface being located and orientated in a single way in the space).

Example 2. *Given the function (defined on a grid of points in $\mathbb{R}^2$)*

$$\begin{array}{rcll} z : G \subseteq \mathbb{R}^2 & \longrightarrow & \mathbb{R} & \\ (x_i, y_i) & \longrightarrow & z(x_i, y_i) & \quad i = 1, \dots, n, \end{array}$$

we find the surface $z(x, y)$ solution to the following problem:

$$\begin{array}{lrlll} \min\limits_{z(x_i,y_i)} & 1 & & & \\ \text{subject to} & & Ast(x_i, y_i) & = 0 & \forall (x_i, y_i) \in G \\ & P - \epsilon \leq & Pow(x_i, y_i) & \leq P + \epsilon & \forall (x_i, y_i) \in G \\ & & z(0,0) & = -\frac{\mu - 1}{Pow(0,0)} & \\ & & \left.\frac{\partial z(x,y)}{\partial x}\right|_{x=0,y=0} & = 0 & \\ & & \left.\frac{\partial z(x,y)}{\partial y}\right|_{x=0,y=0} & = 0, & \end{array} \tag{4.21}$$

where $Pow(x_i, y_i)$ and $Ast(x_i, y_i)$ are defined in (4.7) *and* (4.8)*, $P = 5D$, $\epsilon = 0.12D$, $\mu = 1.6$, and the points (x_i, y_i) satisfy $-45mm \leq x_i, y_i \leq 45mm$.*

The solution to (4.21) is a set of spheres of radius $\hat{R}$ centered at point $(0, 0, 0)$*, such that $R_{min} \leq \hat{R} \leq R_{max}$ where*

$$\begin{aligned} R_{min} &= (\mu - 1)/(P + \epsilon) = (1.6 - 1)/(5 + 0.12) = 0.11719m = 117.19mm \\ R_{max} &= (\mu - 1)/(P - \epsilon) = (1.6 - 1)/(5 - 0.12) = 0.12295m = 122.95mm. \end{aligned}$$

Considering two different solutions of (4.21)

$$\begin{aligned} \text{Solution 1} : z^1(x_i, y_i) &= -\sqrt{R_{min}^2 - x_i^2 - y_i^2} \\ \text{Solution 2} : z^2(x_i, y_i) &= -\sqrt{R_{max}^2 - x_i^2 - y_i^2}, \end{aligned} \tag{4.22}$$

we observe that

$$az^1(x_i, y_i) + bz^2(x_i, y_i) \neq -\sqrt{R_k^2 - x_i^2 - y_i^2} \quad \forall R_k \in \mathbb{R}, \text{where } a + b = 1. \tag{4.23}$$

Therefore the solution set (and thus also the feasible region) of (4.21) (in Cartesian coordinates) is not a convex set.

It is worth noting that if the model had been reformulated in terms of z^2 instead of z, the solution set in the example above would become convex. However (as seen in expressions (4.1), (4.5), (4.6), (4.7), (4.8) and (4.9)), in order to calculate the power and the astigmatism, we need the first and second derivatives of z (not z^2) with respect to x and with respect to y. Therefore, although reformulating the model in variables z^2 might improve the convexity properties, it would make it impractical to compute the power and astigmatism. In addition, it may force us to add some square root, creating discontinuities around zero.

However, with another parametrization (in particular, using spherical coordinates) the feasible region of that problem becomes convex. This new parametrization is shown in the next Chapter.

Although the space of solutions in Cartesian coordinates is not a convex set, some progressive lenses are calculated using Cartesian coordinates. In practice, we will see that when calculating progressive lenses from scratch it is faster to use Cartesian coordinates than spherical coordinates because, although the convexity is worse, the calcutations are simpler. In Section 4.6 (page 29) several progressive lenses are calculated using Cartesian coordinates. Firstly, the optimization model is detailed in the next section.

4.5 The optimization model

The goal of the optimization problem is to obtain a square surface with certain optical properties. These optical properties are astigmatism and power, as defined in the previous sections. Predefined values of power and maximum values of astigmatism will be imposed on certain regions of the surface. In addition, the surface must be as smooth as possible, and have minimum astigmatism. These last two conditions will be controlled by the objective function.

The parameters, variables and objective function of the model using Cartesian coordinates are very similar to those in spherical coordinates model. Below, we will present the model in Cartesian coordinates. Details of the spherical coordinates model will be explained in Section 5.4 of Chapter 5 (pag. 75).

Parameters of the model

The main parameters for defining the model are:

- $(x_i, y_j) \in [-r, r] \times [-r, r]$, $(i, j) \in \mathcal{G} = \{1, \ldots, n\} \times \{1, \ldots, n\}$, is a grid of points used for defining the lens in Cartesian coordinates, where n denotes the number of points for each dimension of the grid. The grid is defined such that $(x_{\frac{1+n}{2}}, y_{\frac{1+n}{2}}) = (0, 0)$. The grid $\mathcal{G}$ is partitioned into three subsets: $\mathcal{G} = \mathcal{F} \cup \mathcal{N} \cup \mathcal{B}$. $\mathcal{F}$ and $\mathcal{N}$ are the set of points in the far and near regions of the lens, respectively, where some values of power will be imposed; $\mathcal{B}$ is the set of the remaining points, corresponding to regions of the lens whose power will not be constrained.

- $(x'_{i'}, y'_{j'}) \in [-r, r] \times [-r, r]$, $(i', j') \in \mathcal{G}' = \{1, \ldots, o\} \times \{1, \ldots, o\}$ is another grid of points (mm), with being $\mathcal{G}'$ much coarser than $\mathcal{G}$ (i.e., $o \ll n$), where o is the number of points used in defining of a B-spline whose coefficients are the variables of the optimization model (see next Section).

- P_F and P_N are the requested powers (in diopters) in the far and near regions of the lens, respectively ($P_N > P_F$).

- $\mu \in [1.5, 1.9]$ is the refractive index of the lens material

- The subset $\mathcal{F}$ of the grid's far region points is partitioned into k additional far subregions $\mathcal{F} = \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_k$. For each of these k subregions we consider a tolerance ϵ_h, $h = 1, \ldots, k$, for the soft constraints for the power (in diopters). The total number of far region constraints is then $\sum_{h=1}^{k} |\mathcal{F}_h|$.

- Similarly to the far region, the set of points $\mathcal{N}$ of the near region is partitioned into l near subregions $\mathcal{N} = \mathcal{N}_1 \cup \cdots \cup \mathcal{N}_l$. For each of these l subregions we consider a tolerance δ_h, $h = 1, \ldots, l$, for the soft constraints for the power (in diopters). The total number of near region constraints is $\sum_{h=1}^{l} |\mathcal{N}_h|$.

- The grid $\mathcal{G}$ of points is also partitioned into different m subregions of astigmatism, that is, $\mathcal{G} = \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_m$. An upper bound on β_h, $h = 1, \ldots, m$ will be imposed to the astigmatism (diopters) of points in each subregion. The total number of astigmatism constraints is $\sum_{h=1}^{m} |\mathcal{A}_h|$.
- Finally, w_1, w_2, $w_3 \in [0,1] \subset \mathbb{R}$ are weights of the different parts of the objective function (defined below in (4.26)).

Variables of the model

The variables of the optimization problem are the coefficients of a three-degree B-spline surface, as defined in [44, page 100]. These coefficients are denoted as

$$\mathbb{R} \ni c(x'_{i'}, y'_{j'}) \geq 0, \qquad (i', j') \in \mathcal{G}'. \tag{4.24}$$

Using the B-spline we define the vertical component of the surface for the grid $\mathcal{G}$ as

$$z(x_i, y_j) = \sum_{i'=1}^{o} \sum_{j'=1}^{o} c(x'_{i'}, y'_{j'}) B^3_{i'}(x_i) B^3_{j'}(y_j), \quad (i,j) \in \mathcal{G}, \tag{4.25}$$

where $B^3_{i'}(x_i)$ and $B^3_{j'}(y_j), (i,j) \in \mathcal{G}, (i', j') \in \mathcal{G}'$, are the 1-dimensional three-degree B-splines basis defined in [44, page 100], and detailed in (3.28) and (3.29) using (3.22)-(3.27). Calculations of the derivatives of the B-splines needed to compute the power and the astigmatism are detailed in (3.30)–(3.41).

In Chapter 3, sections 3.5, 3.6 and 3.7 the B-splines are described in more detail calculating its derivatives, explaining its properties and plotting some of them.

Objective function

The objective function (4.26) consists of minimizing the sum of the squared astigmatism and the squared norm of the gradients of power and astigmatism, for all the points on the grid $\mathcal{G}$. These factors are weighted by $w_1, w_2, w_3 \in [0,1]$. The objective function is:

$$\begin{aligned} \min \sum_{(i,j) \in \mathcal{G}} \frac{1}{n^2} \Bigg(& w_1 \Big(Ast(x_i, y_j) \Big)^2 + \\ & w_2 \Bigg(\Big(\frac{\partial Ast(x_i, y_j]}{\partial x} \Big)^2 + \Big(\frac{\partial Ast(x_i, y_j)}{\partial y} \Big)^2 \Bigg) + \\ & w_3 \Bigg(\Big(\frac{\partial Pow(x_i, y_j)}{\partial x} \Big)^2 + \Big(\frac{\partial Pow(x_i, y_j)}{\partial y} \Big)^2 \Bigg) \Bigg), \end{aligned} \tag{4.26}$$

where n is the number of points in each dimension of the grid.

Note that the astigmatism and also its derivatives are squared to avoid discontinuities around zero (the derivatives of the astigmatism may have divisions by zero if not squared).

Constraints

The objective function is minimized subject to the following three groups of constraints:

$$\begin{aligned} P_F - \epsilon_h \leq Pow(x_i, y_j) \leq P_F + \epsilon_h \quad (i,j) \in \mathcal{F}_h,\ h = 1, \ldots, k, \qquad (4.27) \\ P_N - \delta_h \leq Pow(x_i, y_j) \leq P_N + \delta_h \quad (i,j) \in \mathcal{N}_h,\ h = 1, \ldots, l, \qquad (4.28) \\ Ast(x_i, y_j)^2 \leq \beta_h^2 \quad (i,j) \in \mathcal{A}_h,\ h = 1, \ldots, m. \qquad (4.29) \end{aligned}$$

These constraints define the philosophy of the design of the progressive lens. Constraints (4.27) and (4.28) control the power in the different far and near subregions of the lens, while constraints (4.29) fix a maximum of astigmatism in certain regions of the lens. Constraints (4.29) are squared to avoid the square root in the definition of the astigmatism (4.8), thus making the model simpler and avoiding discontinuities near (θ_i, φ_j) when $Ast(\theta_i, \varphi_j) = 0$ (see Section 4.2, pag. 22).

The quality and characteristics of the progressive lens are governed by the values of the parameters ϵ_h, δ_h and β_h, and the sets $\mathcal{F}_h$, $\mathcal{N}_h$ and $\mathcal{A}_h$. In terms of optics, setting the proper values of these parameters is the most difficult part.

A second set of three constraints imposes conditions on the midpoint of the grid $(0, 0)$ (and on the lens surface):

$$\begin{aligned} z\,(0,0) &= \tfrac{1-\mu}{P_F} \\ \left.\tfrac{\partial z(x_i,y_i)}{\partial x}\right|_{x_i=0,y_i=0} &= 0 \\ \left.\tfrac{\partial z(x_i,y_i)}{\partial y}\right|_{x_i=0,y_i=0} &= 0. \end{aligned} \qquad (4.30)$$

The purpose of these three constraints is to center the lens within the three dimensional space: the first one imposes a certain radius, while the other two guarantee that it is perpendicular to the normal of the surface. These are the only equality constraints on the model.

Constraints (4.27), (4.28) and (4.29) refer to the properties of the lens (power and astigmatism). On the other hand, constraints (4.30)refer to the shape of the lens.

Finally, the optimization problem to be solved is the minimization of (4.26), subject to constraints (4.27), (4.28), (4.29), (4.30) and (4.25) using variables (4.24).

The solution

Once the optimization problem has been solved, we obtain the optimal coefficients of the B-spline surface $c(x'_{i'}, y'_{j'}) \geq 0, (i', j') \in \mathcal{G}'$. Using (4.25) we obtain $z(x_i, y_j), (i, j) \in \mathcal{G}$. The obtained points $(x_{ij}, y_{ij}, z_{ij}) \in \mathbb{R}^3,\ (i, j) \in \mathcal{G}$, where (x_{ij}, y_{ij}) define a square surface in $\mathbb{R}^2$ are sent to the free-form generator to physically cut the lens.

4.6 Numerical results

4.6.1 Problem instances

We generated a set of 12 families of problem instances, denoted as F1, F2,...,F12. Each family of problems have 6 different types denoted as T1, T2,...,T6. This produces a total number of problem instances of 72 (12 families · 6 types = 72). Some parameters that are common for all 72 problem instances are:

- $P_F = 5$D (power in the far region).
- P_N =7D (power in the near region).
- $\mu = 1.6$ (refractive index of the lens material).
- $n = 59$ (n^2 is the number of points in grid $\mathcal{G}$).
- $o = 35$ (o^2 is the number of points in grid $\mathcal{G}'$, the grid used to define the B-splines). Remember that the B-splines can be evaluated at any point, not only for the list of o^2 points. In particular, for each problem we have $n^2 = 59^2 = 3481$ points where the B-splines can be evaluated.
- $(x_i, y_j), (i, j) \in \mathcal{G}$ (the particular points used in grid $\mathcal{G}$).
- $(x'_{i'}, y'_{j'}), (i', j') \in \mathcal{G}'$ (the particular points used in grid $\mathcal{G}'$).

  Note that:

  – the subset $\mathcal{F}$ of far region points and its partition in k far subregions $\mathcal{F} = \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_k$,

  – the subset $\mathcal{N}$ of near region points and its partition in l near subregions $\mathcal{N} = \mathcal{N}_1 \cup \cdots \cup \mathcal{N}_l$,

  – and the partition of m subregions of astigmatism $\mathcal{G} = \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_m$

are different for each family.

The grid of points $\mathcal{G}'$ is a regular grid computed by the formula $x'_{i'} = y'_{i'} = -72+4\dot{i}, \ i = 1, ..., o$ i.e., from -68 mm to 68 mm each 4 mm step. Note that only from -54 mm to 54 mm is used.

Among the parameters that differ for each problem we find $\epsilon_h \ h = 1, \ldots, k$, $\delta_h \ h = 1, \ldots, l$, and $\beta_h \ h = 1, \ldots, m$.

Each family of problems differs from every other family of problems because each one has a different objective function and/or a different shape for the regions (far subregions, near subregions and subregions of astigmatism).

Each problem of one family differs from every other problem of the same family and different type because the near region is located in a different position, having three different horizontal positions and two different vertical positions. The position of the center of the near region is described in Table 4.1 for each type of problem.

Type	x	y
T1	1.0	-12.5
T2	2.5	-12.5
T3	4.0	-12.5
T4	1.0	-14.0
T5	2.5	-14.0
T6	4.0	-14.0

Table 4.1: Horizontal and vertical coordinates in mm for the center of the near region for each type of problem.

4.6.2 Example of zones for a particular family

The type T2 of family F1 is considered to represent the different subregions of the far region points, near regions points and astigmatism regions points. For this problem we have $k = 2$ far regions, $l = 2$ near regions, and $m = 9$ astigmatism regions. Type T2 has the center of the near zone in the position ($x = 2.5\ mm, y = -12.5\ mm$) (as described previously in Table 4.1).

The regions corresponding, respectively, to far, near and astigmatism (defined by sets $\mathcal{F}_1, \mathcal{F}_2$, $\mathcal{N}_1, \mathcal{N}_2$, and $\mathcal{A}_1, ..., \mathcal{A}_9$) are shown in Figures 4.4, 4.5 and 4.6, and are illustrated using different colors for each subregion.

In Figure 4.6 the near and far astigmatism regions are concentric; this fact, together with the values of δ_h and the objective function, guarantee that the change in astigmatism will be gradual, thus obtaining a smoother lens. This same behavior also applies to the far and near regions in Figures 4.4 and 4.5. In Figure 4.6 we may also observe that some astigmatism regions correspond to the part of the lens named *corridor*, which connects the far and near regions.

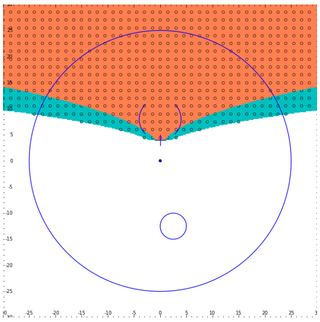


Figure 4.4: The two far regions of problem F1 and type T2, each shown in a different color.

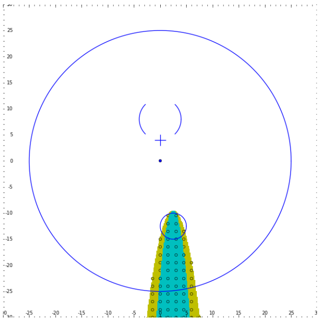


Figure 4.5: The two near regions of problem F1 and type T2, each shown in a different color.

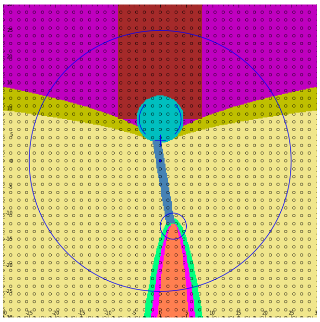


Figure 4.6: The nine astigmatism regions of problem F1 and type T2, each shown in a different color.

The tolerances ϵ_1, ϵ_2, of far region constraints (4.27) are, respectively, 0.11 and 0.16. For near region constraints (4.28), tolerances δ_1, δ_2 are 0.01 and 0.06. Finally, the 9 tolerances $\beta_h, h = 1, \dots, 9$ for astigmatism constraints (4.29) are 0.10, 0.11, 0.14 and 0.25 (for the far astigmatism zones); 0.09, 0.12 and 0.25 (for the near astigmatism zones); 0.12 for the corridor astigmatism zone, and 2.5 for the global zone (yellow color). These tolerances are expressed in diopters (D). For this family, the 9 astigmatism subregions are partitioned into four far astigmatism subregions, three near astigmatism subregions, one corridor astigmatism subregion and one global region (with the rest of points). In all families, there is always a global zone of astigmatism with the rest of points of the lens. In this way, all the points have exactly one constraint of astigmatism (see constraint (4.29), page 29).

Note that this family has a lot of subregions for astigmatism and only two subregions for far regions and near regions of power. For the other problem instances of family F1 and other types (T1, T3, T4, T5 and T6), the distribution of regions is similar except that the near zone is shifted horizontally and vertically according to Table 4.1. The other zones for the other families are not described in detail. The number of zones of the other families can be different (i.e. different parameters k, l and m), different shapes of the zones having bigger or smaller regions with different tolerances.

4.6.3 Number of constraints and variables

Once the shape of the regions have been detailed in previous section using an example, we can easily describe and comprehend the number of constraints. The number of constraints is the addition of:

- the sum of the number of points of sets $\mathcal{F}_h$, $h = 1, \dots, k$, that are the points in the far regions, usually much smaller than n^2. See constraints (4.27), page 29.

- the sum of the number of points of sets $\mathcal{N}_h$, $h = 1, \dots, h$, that are the points in the near regions, usually much smaller than n^2. See constraints (4.28), page 29.

- the sum of the number of points of sets $\mathcal{A}_h$, $h = 1, \dots, l$, that are the points in the astigmatism regions. In our problem instances there are exactly n^2 points in the astigmatism region. See constraints (4.29), page 29, and the example in Figure 4.6.

- 3 constraints. See constraints (4.30) (page 29).

Family/Type	T1	T2	T3	T4	T5	T6
F1	7811	7815	7811	7792	7792	7792
F2	7878	7878	7878	7864	7865	7864
F3	7811	7815	7811	7792	7792	7792
F4	7878	7878	7878	7864	7865	7864
F5	6289	6293	6289	6270	6270	6270
F6	6289	6293	6289	6270	6270	6270
F7	6289	6293	6289	6270	6270	6270
F8	6289	6293	6289	6270	6270	6270
F9	7878	7878	7878	7864	7865	7864
F10	7878	7878	7878	7864	7865	7864
F11	7878	7878	7878	7864	7865	7864
F12	6289	6293	6289	6270	6270	6270

Table 4.2: Number of constraints for each problem instance, i.e., for each familiy and type. The number of variables is always 961.

Table 4.2 reports the number of constraints for all families and types. For a fixed family (for example F1) the number of constraints is very similar for all the types (for example for F1 the number of constraints varies from 7811 to 7792). The reason for this is that, once fixed a family, one type differs from another type because the near region has been shifted (horizontally and/or vertically), but the shape of the regions remains the same. When shifting the near regions to the bottom (for example from T1, T2 and T3 to T4, T5 and T6, respectively), the number of constraints decreases, because the near regions goes from a certain point to the bottom of the square grid (so the regions are a lightly smaller for T4, T5 and T6 than for T1, T2 and T3, respectively). When shifting the near regions horizontally, the number of constraints vary very little (between 0 and 4 constraints), only due to the fact that there are rounded areas in a discrete grid of points.

The number of variables is always the same $o^2 = 31^2 = 961$, which is the number of points in the grid $\mathcal{G}'$.

In general, we have between 6000 to 8000 (nonlinear) constraints, and most of these are inequalities. There are only three equality constraints defined in (4.30) (page 29).

We may note that there are $n^2 = 59^2 = 3481$ points where power and astigmatism can be evaluated. Constraints (4.27) refer to the power values in the points of the far subregions; constraints (4.28) refer to the power values in the points of the near subregions; and finally constraints (4.29) refer to the astigmatism values in the points of the astigmatism subregions. In our instances, the number of points of the astigmatism subregions is always n^2, as illustrated in Figure 4.6. However, the number of points in the far and near regions is much smaller than n^2. That means that the power constraints do not apply to the whole lens surface (see Figures 4.4 and 4.5). Consequently, there are always fewer than $3 \cdot n^2 + 3 = 3 \cdot 3481 + 3 = 10446$ constraints, as illustrated in Table 4.2.

Objective function

The objective function (4.26) (page 28) was used for all the 15 families, with the different weights w_1, w_2, and w_3:

- $w_1 = 1, w_2 = 0, w_3 = 0$, for families of problems F1 to F4.

- $w_1 = 1, w_2 = p, w_3 = 0$, for families of problems F5 to F8.
- $w_1 = 0, w_2 = 0, w_3 = 1$, for families of problems F9 to F11.
- $w_1 = 1, w_2 = 0, w_3 = 1$, for families of problems F12.

For families of problems from F5 to F8, we added a parameter p whose values are between 2500 and 10000 in order to balance the minimization of the square of the astigmatism and the minimization of the gradients of power astigmatism.

The units of the objective function are D^2 for families F1 to F4; $D^2 + \frac{D^2}{\text{mm}^2}$ for families F5 to F8 and F12; and $\frac{D^2}{\text{mm}^2}$ for families F9 to F11.

4.6.4 Computational environment and stopping criteria

The optimization model was implemented using the AMPL modeling language [5] linked with three different interior points solvers: LOQO [8], KNITRO [3] and IPOPT [11]. Due to licensing agreements (with AMPL, LOQO and KNITRO, which are commercial products), two different servers were used. The first one has eight 2.7GHz AMD Opteron 8384 Shanghay CPUs, with 32 cores and 128GB RAM. The AMPL modeling language, the LOQO 6.0.6 solver and the IPOPT 3.8.1 solver were installed on this computer. The second machine was a Fujitsu Primergy RX300, with two 3.33 GHz Intel Xeon X5680 CPUs, with 24 cores and 144GB RAM. The modeling language AMPL, and the solvers KNITRO 10.1.0, IPOPT 3.9.3 and IPOPT 3.12.8 were installed on this second server. Both servers will be referred to as “server 1” and “server 2” in the following sections.

From this point forward, we will consider six different combinations of solvers and servers:

- LOQO 6.0.6, server 1.
- IPOPT 3.8.1, server 1.
- IPOPT 3.9.3, server 2.
- IPOPT 3.12.8, server 2.
- KNITRO 10.1.0 with interior point / direct algorithm (algorithm 1), server 2.
- KNITRO 10.1.0 with interior point / conjugate gradient algorithm (algorithm 2), server 2.

The two variants of KNITRO differ in how Newton’s equation is solved at each iteration of the interior point algorithm; either by a direct method (factorization); or through an iterative conjugate gradient [3]. IPOPT and LOQO use a direct method for this step [8, 11].

Recall that when using KNITRO with the active set algorithm (algorithm 3) and the SQP algorithm (algorithm 4) the problems did not converge.

The stopping criteria chosen provides good quality lenses without extra iterations or refinements. In order to determine whether there are extra iterations, some problems have been calculated several times with five different stopping criteria. We appreciate that, from one point, the obtained lenses are practically the same. To evaluate the quality of the progressive lenses (or to determine whether they can be considered the same solution), we plot the astigmatism map and the power map and appreciate that they are in fact very similar. Note that when observing the values of the variables (the coefficients of the B-splines) it is not easy to know whether two lenses are different.

The stopping criteria used for each solver will be described in Section 5.5.3 (page 82) for spherical coordinates, and the same stopping criteria have been used for Cartesian coordinates.

The stopping conditions used for each solver within AMPL were:

- LOQO 6.0.6: sigfig= 4, inftol= 10^{-6}
- IPOPT (versions 3.8.1, 3.9.3 and 3.12.8): tol= 10^{-2}
- KNITRO 10.1.0 (both algorithms 1 and 2): opttol= 10^{-3}

with a maximum of 2000 iterations.

4.6.5 Comparison of solvers

Over the next pages, the following aspects of the solutions will be studied:

- the number of iterations
- the CPU time
- the velocity ($\frac{\text{number of iterations}}{\text{CPU time}}$)
- the value of the objective function
- the relative error with respect to LOQO (the absolute value)
- the relative error with respect to LOQO (with the sign)
- the quality of the solution (in optical terms)

The number of iterations is the number of iterations required to converge, and the CPU time is reported in seconds. The “velocity” is defined as:

$$\text{velocity} = \frac{\text{number of iterations}}{\text{number of seconds}} = \frac{\text{number of iterations}}{\text{CPU time}}. \tag{4.31}$$

Note that considering the expression of the objective function (4.26), problems F1 to F4 are expected to be faster in terms of velocity (iterations/seconds), because the calculations for the power and astigmatism derivatives are not being considered. Note that calculating the power and astigmatism gradients for each iteration requires large calculations (and large CPU time).

In order to evaluate the quality of the objective function, it is worth calculating the relative error. The objective function value used as reference has been the objective function obtained using LOQO. The relative error is thus:

$$\text{absolute relative error} = \frac{|o.f_{LOQO} - o.f_{\text{other solver}}|}{o.f_{LOQO}}. \tag{4.32}$$

It is worth calculating the relative error without the absolute value:

$$\text{relative error} = \frac{o.f_{LOQO} - o.f_{\text{other solver}}}{o.f_{LOQO}} \tag{4.33}$$

because the sign indicates which objective function is bigger (either the objective function obtained using LOQO or the other one).

Number of iterations and CPU time

In most of the 72 problem instances, all the solvers reported solutions obtained as "optimal" (with LOQO and IPOPT) or either "locally optimal" or as "satisfactory solution" (with KNITRO). However, there are some cases that reached the maximum number of iterations (2000 iterations). It was the case for 2 problem instances using IPOPT version 3.8.1, 3 problem instances using IPOPT versions 3.12.8 and 3.9.3 and 13 problem instances using KNITRO with the conjugate gradient algorithm 2.

Using LOQO all of the problem instances reported solutions obtained as "optimal". Using KNITRO with the direct algorithm 1, all the problem instances reported the solutions obtained as "locally optimal" or as "satisfactory solution".

Problem	LOQO	IPOPT	IPOPT	IPOPT	KNITRO	KNITRO
	6.0.6	3.8.1	3.9.3	3.12.8	10.1.0 alg 1	10.1.0 alg 2
Optimal or locally optimal	72	70	69	69	72	59
Maximum iterations exceeded	0	2	3	3	0	13

Table 4.3: Number of optimal solutions found using the six different solvers.

Table 4.3 summarizes the number of problems that reached the optimum using the different solvers for the 72 problem instances (12 families and 6 types). It does not make sense to increase the number of maximum iterations: using LOQO, all the problems converged using fewer than or equal to 192 iterations. Using IPOPT 3.8.1 all the cases that converged used fewer than or equal to 1269 iterations, and one case converged using 72 iterations. Using KNITRO with the direct algorithm 1, all the problems converged using fewer than or equal to 1371 iterations, and one instance was solved with 27 iterations.

The Appendix (page 106) includes a table for each solver which reports the number of iterations (Tables 6.1 to 6.6), the CPU time (Tables 6.7 to 6.12), the velocity (number of iterations / CPU time) (Tables 6.13 to 6.18) and objective value (Tables 6.19 to 6.24). As these tables are rather dens, some of the tables are represented in the figures below in this section.

Figure 4.7 (page 38) summarizes in 6 different bar charts the number of iterations for the 72 problems and 6 solvers. The number of iterations for each solver was:

- LOQO: between 42 and 192 iterations
- IPOPT 3.8.1: between 72 and 1269 iterations (and 2 instances reached 2000 iterations without converging)
- IPOPT 3.9.3: between 151 and 957 iterations (and 3 instances reached 2000 iterations without converging)

- IPOPT 3.12.8: between 151 and 968 iterations (and 3 instances reached 2000 iterations without converging)
- KNITRO with the direct algorithm 1: between 27 and 1371 iterations
- KNITRO with the conjugate gradient algorithm 2: between 46 and 1727 iterations (and 12 instances reached 2000 iterations without converging)

LOQO was the solver with the smallest "maximum number of iterations" (192), so it was the faster solver in terms of the number of iterations. In some cases, KNITRO with the direct algorithm 1 was faster than LOQO, but exhibited large variability than LOQO. The three versions of IPOPT used more iterations than LOQO and KNITRO with the direct algorithm 1. Finally, KNITRO with the conjugate gradient algorithm 2 was the worst performing solver: 13 instances did not reach the optimal solution in fewer than 2000 iterations and had high variability.

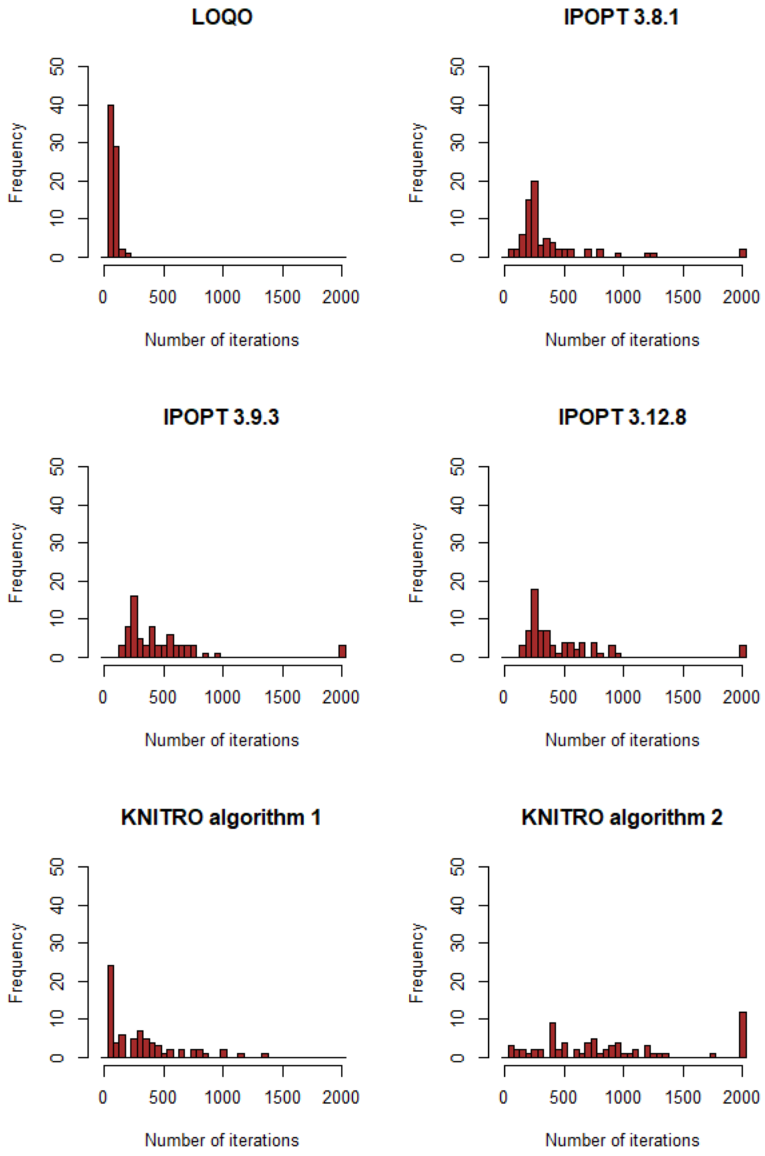


Figure 4.7: Number of iterations for each solver and problem instance (for 12 families and 6 types) using Cartesian coordinates. A total of 72 problem instances. The maximum number of iterations allowed was 2000.

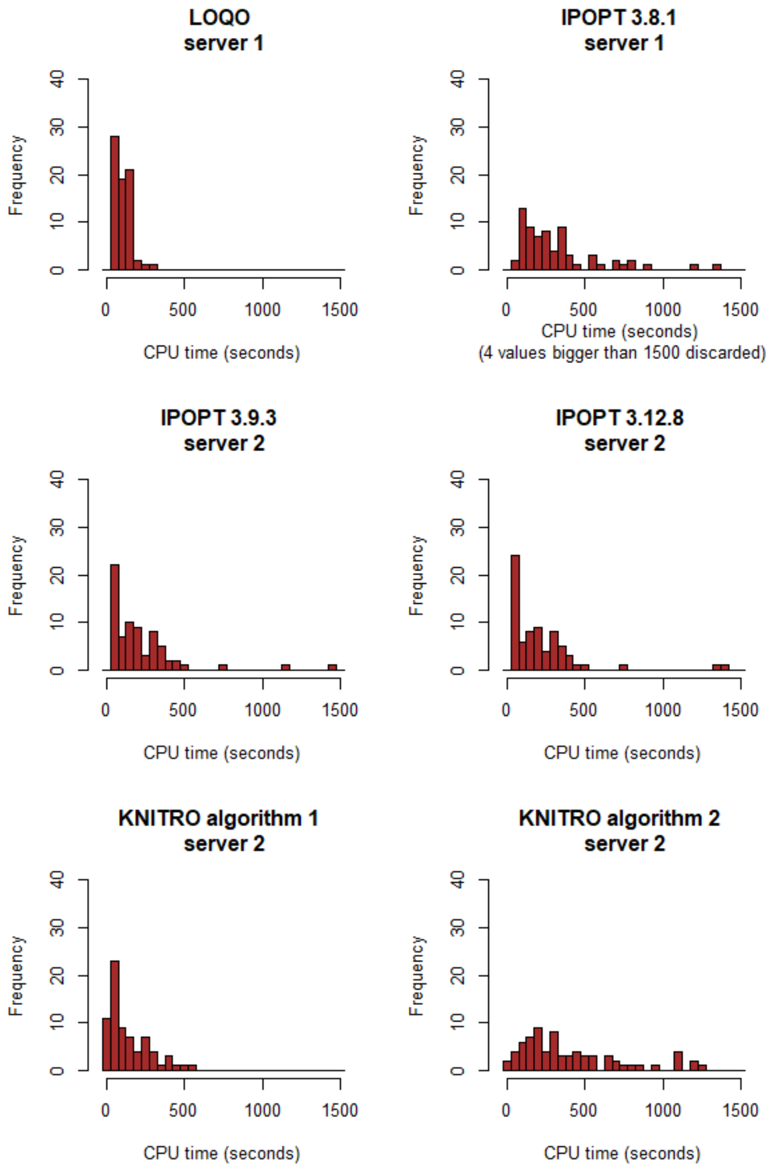


Figure 4.8: CPU time (in seconds) for each solver and problem instance (for 12 families and 6 types) using Cartesian coordinates. A total of 72 problem instances. Note that server 2 is faster than server 1.

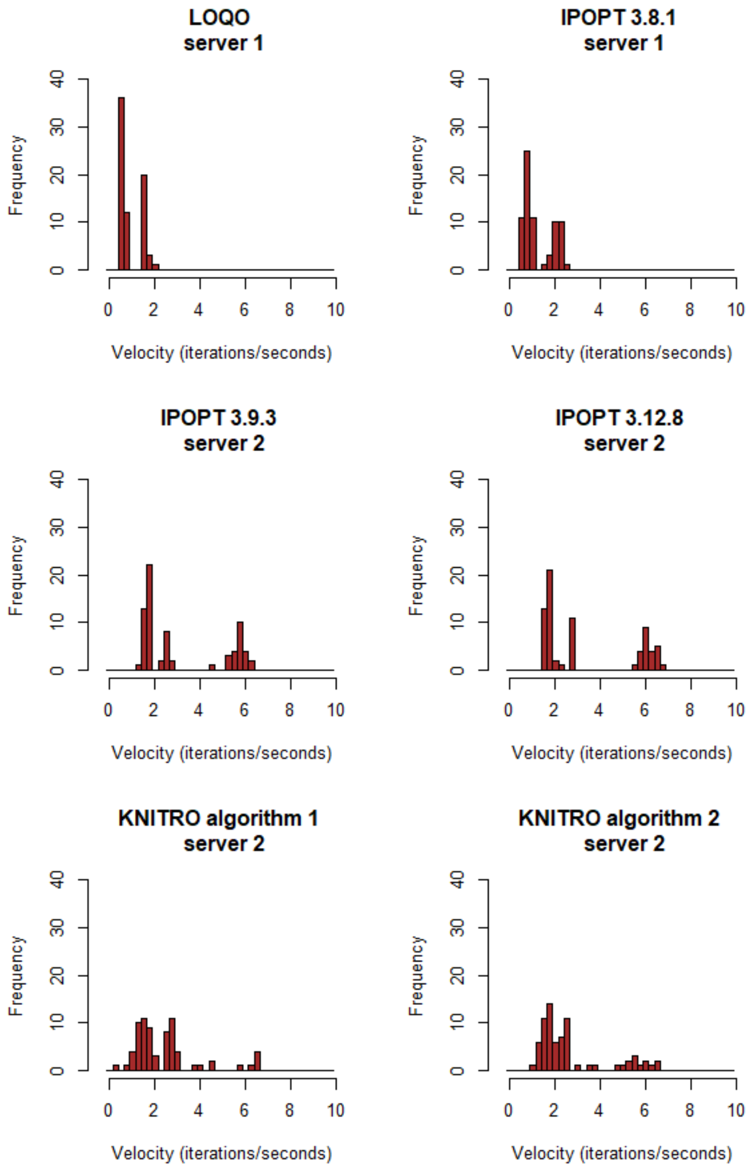


Figure 4.9: Velocity (iteration/seconds) for each solver and problem instance (for 12 families and 6 types) using Cartesian coordinates. A total of 72 problem instances. Note that server 2 is faster than server 1.

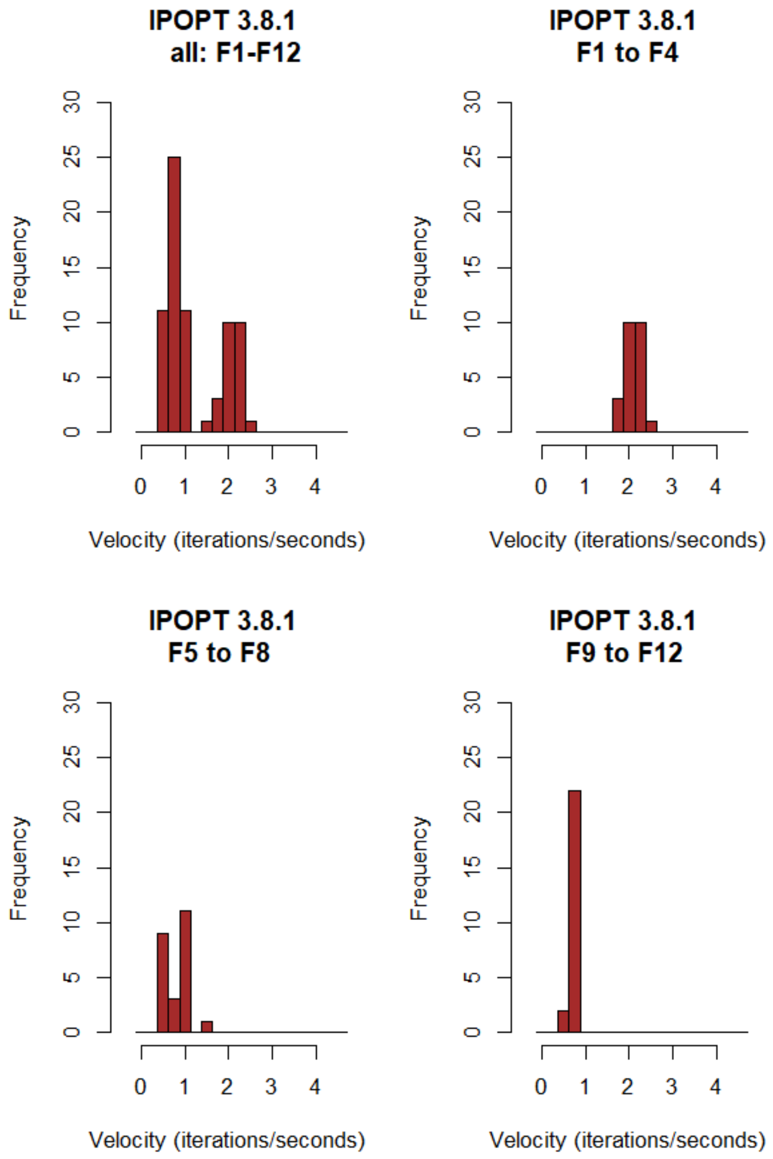


Figure 4.10: Velocity (iteration/seconds) for each solver and problem instance (for 12 families and 6 types) using Cartesian coordinates using IPOPT 3.8.1 on server 1. Graphics considering different objective functions.

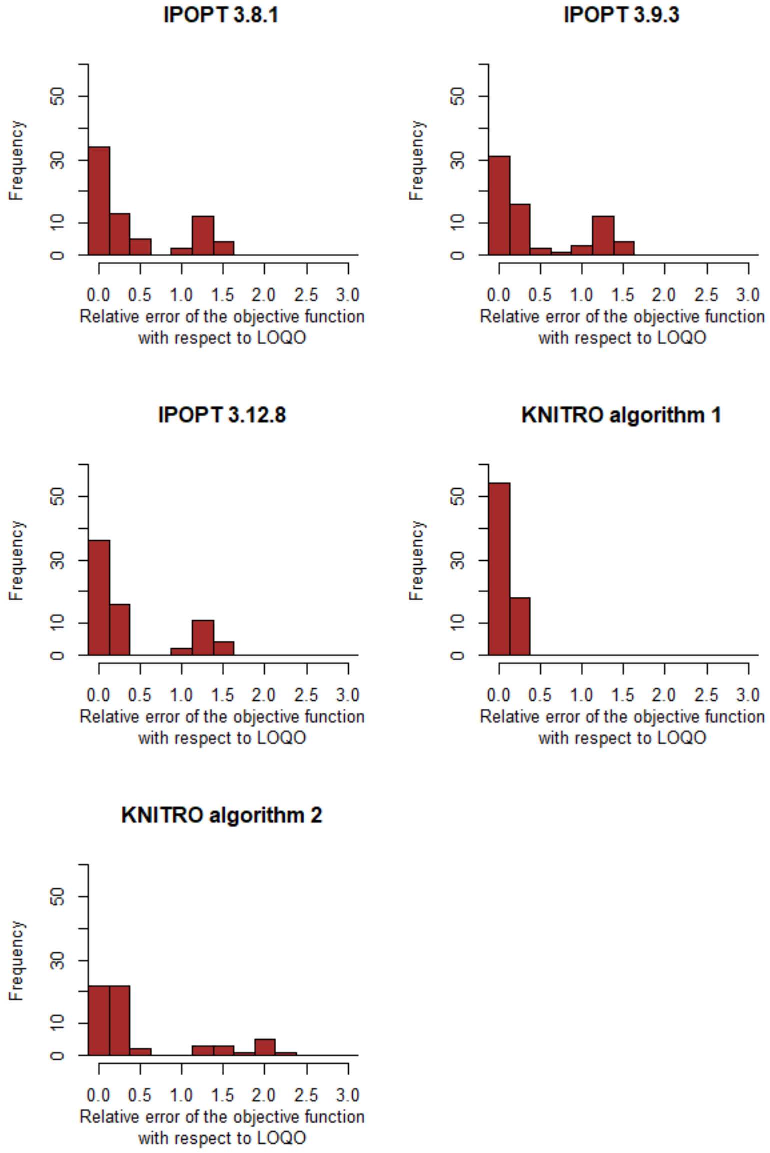
The CPU time, reported in Figure 4.8 (page 39), is directly related to the number of iterations (more iterations, more seconds) taking into account two additional aspects. One aspect is that two solvers (LOQO and IPOPT 3.8.1) were executed on server 1 and the other four on server 2; this explains the different times between the first IPOPT version and the other two, while all of them had similar number of iterations (server 2 was on average 2.65 times faster than server 1). The other aspect is that the objective function (and also the modeling problem) of families F1 to F4 is much simpler than the objective function (and modeling problems) of families F5 to F12, because it does not consider the power and astigmatism gradients (recall the objective function defined in (4.26) and detailed in Section 4.6.3 for each family).

Figure 4.9 (page 40) reports the velocity (number of iterations / number of seconds) (see (4.31)) for each solver for the 72 instances. Considering only IPOPT 3.8.1, Figure 4.10 (page 41) represents the velocity in three different bar charts for three different groups of families. It is appreciated that the velocity for families F1 to F4 is much bigger than the velocity for families F5 to F12. For IPOPT 3.8.1 the velocity is between 1.8 and 2.4 iterations/second for families F1 to F4 and between 0.5 and 1.6 for families F5 to F12 (see also Table 6.14 in the Appendix). That means that calculating the power and astigmatism gradients requires large calculations for each iteration (as was expected).

In order to calculate how many times faster server 2 is than server 1, we have divided the velocity (number of iterations / number of seconds) see (4.31)) of the 72 instances of IPOPT 3.8.1 (on server 1) by the velocity of the 72 instances of IPOPT 3.9.3 (on server 2). The resulting mean of these 72 values was 2.6. When repeating the same calculations using IPOPT 3.8.1 (on server 1) and IPOPT 3.12.8 (on server 2), the mean was 2.7. We considered the mean of these two values. The next chapter will demonstrate that using spherical coordinates, server 2 will be shown to be 2.8 times faster than server 1.

Quality of the objective function and the solutions obtained

The relative error with respect to LOQO has been calculated for all the problem instances and the other 5 solvers. See Tables 6.25 to 6.29 in the Appendix (page 119). Figure 4.11 summarizes theses tables in 5 different bar charts with the relative error with respect to LOQO for the 72 problem instances and the other 5 solvers.



IPOPT 3.8.1

IPOPT 3.9.3

IPOPT 3.12.8

KNITRO algorithm 1

KNITRO algorithm 2

Figure 4.11: Relative error of the objective function for each solver and problem instance with respect to LOQO (for 15 families and 6 types) using Cartesian coordinates. A total of 72 problem instances.

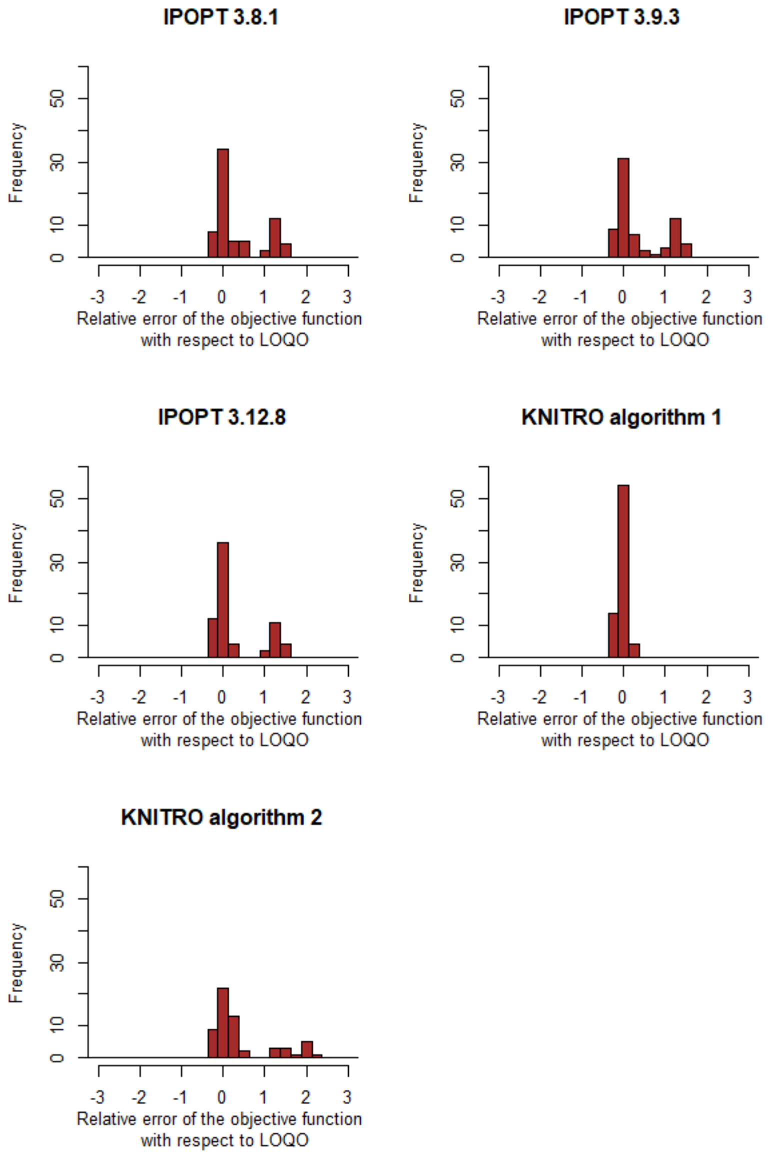


Figure 4.12: Relative error without the absolute value of the objective function for each solver and problem instance with respect to LOQO (for 15 families and 6 types) using Cartesian coordinates. A total of 72 problem instances.

The relative error with respect to LOQO for each solver was:

- between 0.0 and 1.59 for IPOPT (3 versions)
- between 0.0 and 0.29 for KNITRO with the direct algorithm 1
- between 0.0 and 2.17 for KNITRO with the conjugate gradient algorithm 2

The problems that did not reach the optimum have been omitted from the ranges above. Note that the relative error of the objective function with respect to LOQO for all the others solvers should be near zero. A relative error of 1.0 means that the objective function of one solver is twice the objective function of the other solver (taking as a reference the solver with the smallest objective function). A relative error of 1.5 means that the objective function of one solver is 2.5 times the objective function of the other solver (also taking as a reference the solver with the smallest objective function).

In order to determine which solver is better (if LOQO or one of the others) we may examine which objective function is bigger or the sign of the relative error without the absolute value:

$$\frac{o.f_{LOQO} - o.f_{\text{other solver}}}{o.f_{LOQO}}$$

This is illustrated in 4.12 (page 44) and the limit of the relative error without the absolute value was:

- between -0.33 and 1.59 for IPOPT 3.8.1
- between -0.36 and 1.59 for IPOPT 3.9.3 and IPOPT 3.12.8
- between -0.29 and 0.18 for KNITRO with the direct algorithm 1
- between -0.36 and 2.17 for KNITRO with the conjugate gradient algorithm 2

Considering Table 6.33 (page 123) without rounding to 2 decimals, there are 60 problem instances where the minimum of KNITRO with the direct algorithm 1 was smaller than the objective function of LOQO; and 12 problem instances where the minimum of KNITRO with the direct algorithm 1 was bigger than the objective function of LOQO. However, considering LOQO as reference is reasonable, since the solutions produced by both of these solvers are very similar.

We want to determine the maximum relative error from which two problem instances can be considered the same. Two solutions are the same, if the astigmatism map and the power map look the same using isolines of 0.25D.

In Figures 4.13 and 4.14, it is appreciated that a relative errors of 1.59 and 0.42, respectively, produce different power and astigmatism maps. It is appreciated that the astigmatism and power maps using IPOPT 3.8.1 of these Figures are rougher than the astigmatism and power maps produced using LOQO. Consequently, the solution using IPOPT 3.8.1 is not of sufficient quality, and relative errors of 1.59 and 0.42 are much bigger (as one would expect).

We continued to investigate what the biggest relative error could be that would produce a lens of sufficient quality that was also similar to the original. In Figure 4.15 it is appreciated that both lenses look the same, and they have a relative error of 0.29. The maximum biggest value of the relative errors of all the 72 instances with KNITRO with the direct algorithm 1 (with respect to LOQO) is 0.29. We may thereby conclude that all lenses calculated using LOQO and KNITRO with the direct algorithm 1 look the same, and have similar objective functions.

Finally, Figure 4.16 shows that the two lenses with a relative error of 0.33 look slightly, but no significantly, different. One illustrative example is the astigmatism map around the point ($x = -1.5\ mm, y = -10.0\ mm$) and the power map around the point ($x = 20.0\ mm, y = 2.5\ mm$).

In conclusion, a relative error below 0.30, produces lenses of similar quality, an a relative error greater than 0.30 produces lenses of different qualities. The greater the relative error, the greater the difference in the the lenses (as it is evident).

All the problem instances calculated with LOQO and KNITRO using direct algorithm 1 have a relative error of fewer than 0.30, and we may therefore consider that these two solvers can produce a correct optimal lens for all of the 72 problem instances computed.

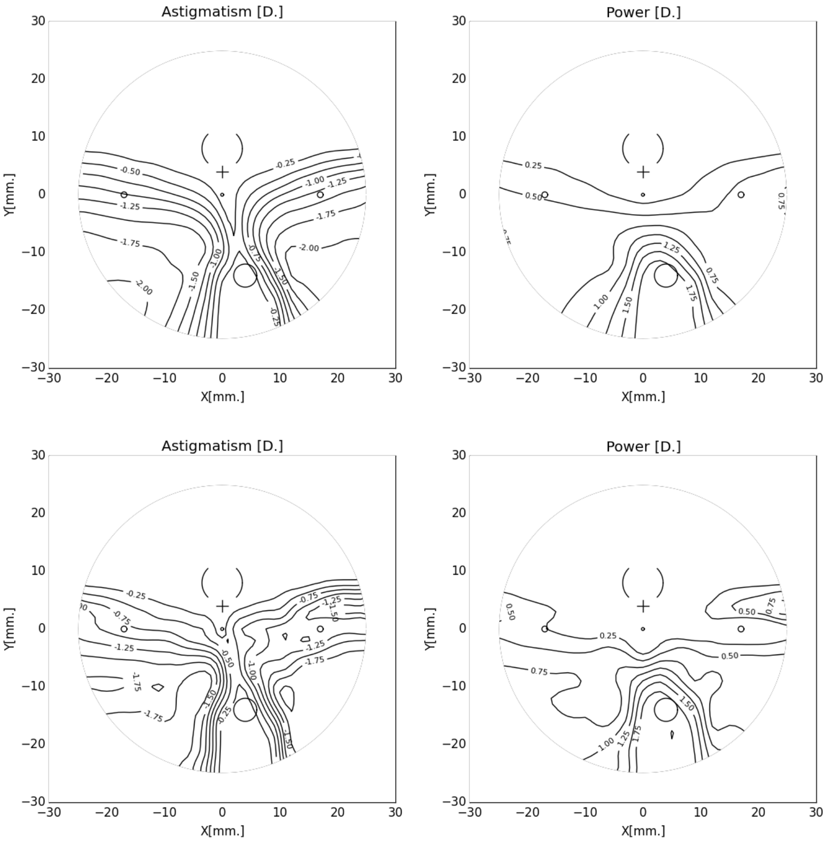


Figure 4.13: Astigmatism (left) and power (right) of the lens of family F10 and type T6 using LOQO 6.0.6 (top) and IPOPT 3.8.1 (bottom), with a relative error of 1.59. Maps using IPOPT 3.8.1 are rougher than maps using LOQO.

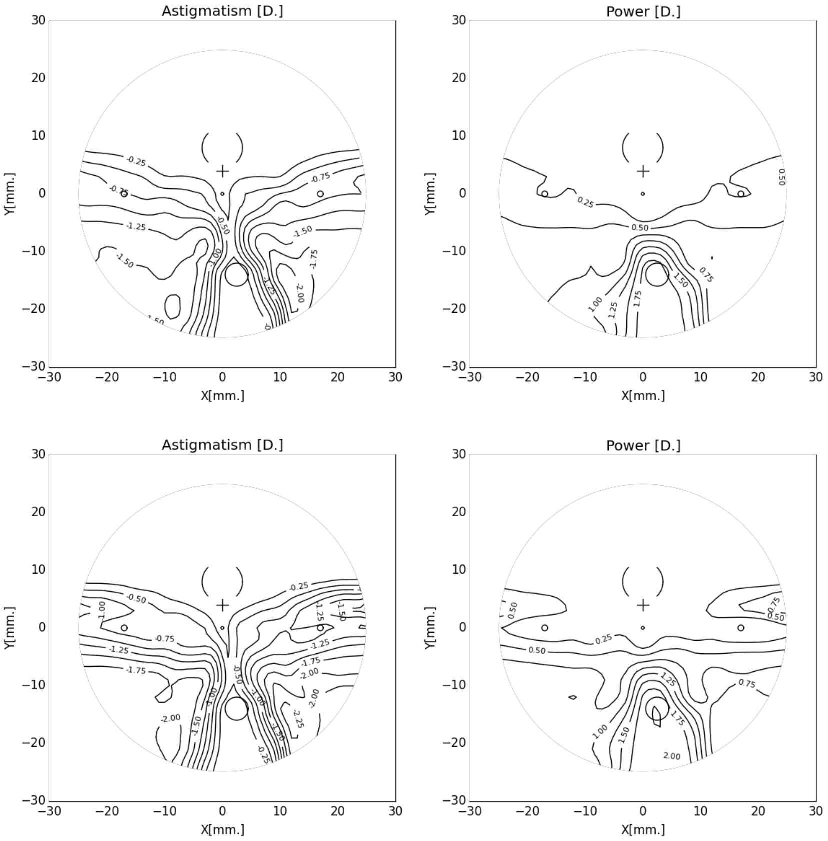


Figure 4.14: Astigmatism (left) and power (right) of the lens of family F1 and type T5 using LOQO 6.0.6 (top) and IPOPT 3.8.1 (bottom), with a relative error of 0.42. Maps using IPOPT 3.8.1 are rougher than maps using LOQO.

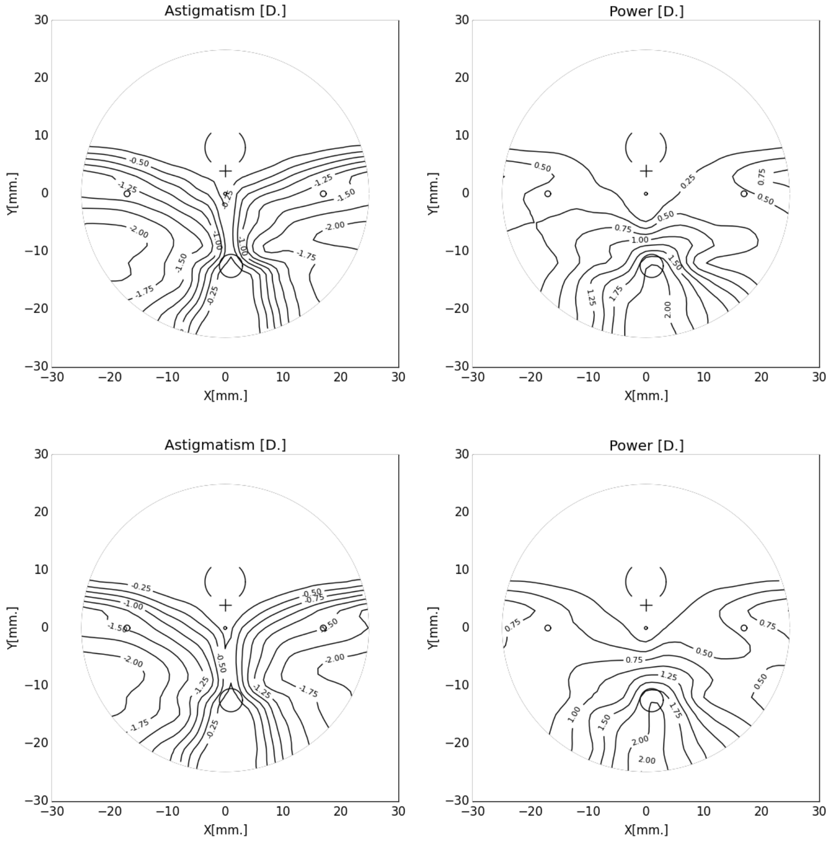


Figure 4.15: Astigmatism (left) and power (right) of the lens of family F7 and type T1 using LOQO 6.0.6 (top) and KNITRO with the direct algorithm 1 (bottom), with a relative error of 0.29.

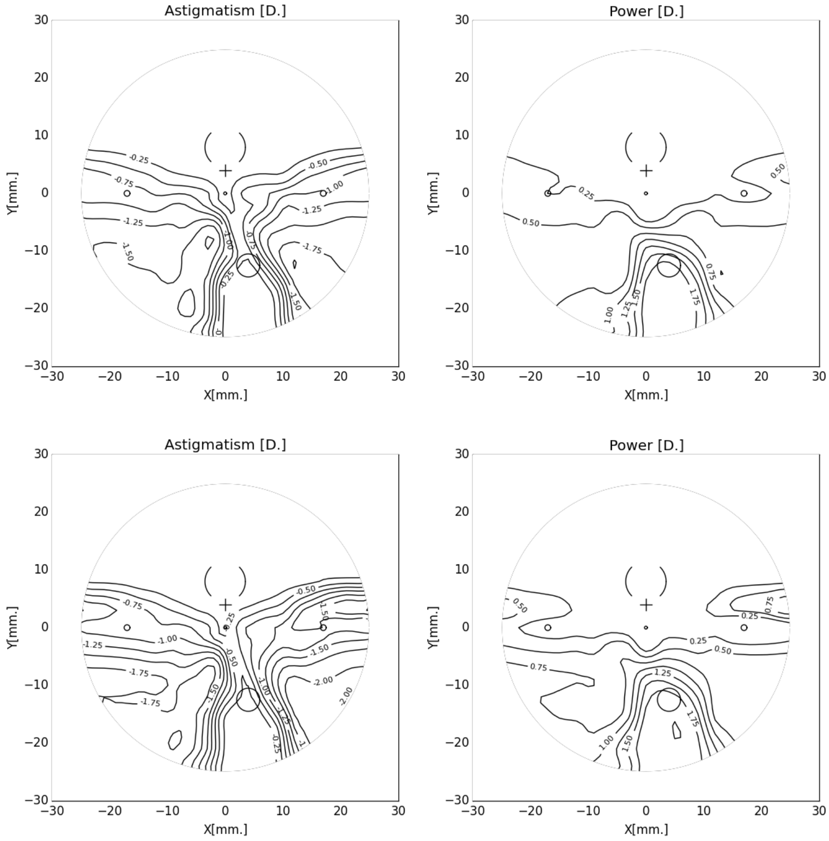


Figure 4.16: Astigmatism (left) and power (right) of the lens of family F2 and type T3 using LOQO 6.0.6 (top) and IPOPT 3.12.8 (bottom), with a relative error of 0.33.

The percentage of problem instances for each case having a relative error of fewer than 0.30 is shown in the following Table 4.4:

Problem	LOQO	IPOPT	IPOPT	IPOPT	KNITRO	KNITRO
	6.0.6	3.8.1	3.9.3	3.12.8	10.1.0 alg 1	10.1.0 alg 2
Optimal or locally optimal with a rel. error < 0.30	72 (100%)	43 (59.7%)	42 (58.3%)	48(66.6%)	72 (100%)	38 (52.8%)
Optimal or locally optimal with a rel. error > 0.30	0	27 (37.5%)	27 (37.5%)	21 (29.2%)	0	21 (29.2%)
Maximum iterations exceeded	0	2 (2.8%)	3 (4.2%)	3 (4.2%)	0	13 (18.0%)

Table 4.4: Number of optimal solutions found using the six different solvers, indicating the cases with relative error with respect to LOQO inferior to 0.30.

Note than in Tables 4.5 to 4.9 there are some values with the relative error bigger than 0.30. Note that all of the problem instances for families F9, F10 and F11 produce bad results. The objective function for F9, F10 and F11 is:

$$\min \sum_{(i,j)\in\mathcal{G}} \frac{1}{n^2}\left(w_3 \left(\left(\frac{\partial Pow(x_i, y_j)}{\partial x} \right)^2 + \left(\frac{\partial Pow(x_i, y_j)}{\partial y} \right)^2 \right) \right), \tag{4.34}$$

without considering the minimization of the sum of the astigmatism. If we discard these problem families (F9, F10 and F11), the number of instance problems with a relative error bigger than 0.30 is still not zero (but decreases significantly).

Problem instances for families F9, F10 and F11 can be improved by adding another term with $w_1 > 0$ (4.26) (page 28) to the objective function .

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.41	0.33	0.32	0.13	0.42	0.40
F2	0.12	0.21	0.10	0.12	0.40	0.33
F3	0.05	0.90	0.00	0.07	0.03	0.05
F4	0.61	0.06	0.02	0.02	0.06	0.05
F5	-0.15	-0.01	-0.11	-0.02	-0.11	†
F5	-0.11	0.02	-0.11	-0.33	-0.06	0.00
F7	-0.29	0.00	-0.23	-0.19	-0.02	-0.17
F8	-0.07	-0.10	-0.13	-0.12	-0.05	-0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.50	1.15	1.17	1.59
F11	1.16	1.21	1.18	†	1.09	1.20
F12	-0.02	-0.02	-0.02	0.00	-0.02	-0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 4.5: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.8.1 on "server 1" without absolute value.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.10	0.33	0.32	0.13	0.11	0.33
F2	0.12	0.29	0.33	0.12	0.40	0.39
F3	0.05	0.90	0.00	0.07	0.03	0.05
F4	0.05	0.06	0.02	0.81	0.06	0.89
F5	-0.15	-0.02	-0.11	-0.03	-0.12	†
F5	†	-0.13	-0.14	-0.36	-0.11	-0.16
F7	-0.29	-0.03	-0.24	-0.06	0.14	-0.07
F8	-0.07	-0.10	-0.13	-0.08	-0.05	-0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.55	1.15	1.17	1.59
F11	1.16	1.21	1.18	†	1.09	1.20
F12	-0.02	-0.02	-0.02	0.00	-0.02	-0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 4.6: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.9.3 on "server 2" without absolute value.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.10	0.33	0.07	0.13	0.11	0.33
F2	0.12	0.08	0.33	0.12	0.12	0.10
F3	0.05	0.04	0.00	0.07	0.03	0.05
F4	0.05	0.06	0.02	0.02	0.06	0.05
F5	-0.15	-0.02	-0.11	-0.03	-0.12	†
F5	†	-0.13	-0.14	-0.36	-0.11	-0.16
F7	-0.29	-0.03	-0.24	-0.19	-0.02	-0.18
F8	-0.07	-0.10	-0.13	-0.14	-0.05	-0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.50	1.15	1.17	1.59
F11	1.16	1.06	1.18	†	1.09	1.20
F12	-0.02	-0.02	-0.02	0.00	-0.02	-0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 4.7: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.12.8 on "server 2" without absolute value.

Family/Type	T1	T2	T3	T4	T5	T6
F1	-0.02	-0.02	-0.07	-0.01	-0.01	-0.03
F2	0.00	-0.04	-0.03	-0.01	0.00	-0.02
F3	-0.01	-0.02	-0.06	0.00	-0.03	-0.02
F4	0.00	0.00	-0.03	-0.04	0.00	0.00
F5	-0.15	-0.02	-0.15	0.02	-0.12	-0.06
F5	-0.15	-0.13	-0.15	-0.20	-0.12	-0.17
F7	-0.29	-0.03	-0.27	-0.20	-0.03	-0.20
F8	-0.07	-0.10	-0.17	-0.13	-0.05	-0.13
F9	0.07	0.16	-0.09	0.14	0.18	-0.02
F10	0.07	0.11	-0.06	0.08	0.17	0.01
F11	-0.05	-0.01	-0.05	0.00	0.06	0.02
F12	-0.03	-0.02	-0.02	0.00	-0.02	-0.07

Table 4.8: Relative error of the objective function with respect to LOQO for each family and problem with KNITRO 10.1.0 direct algorithm 1 on "server 2" without absolute value.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.13	0.13	†	†	0.38	0.11
F2	0.14	0.11	0.27	0.26	0.14	†
F3	0.32	0.31	0.02	0.36	0.62	†
F4	0.31	0.07	0.22	0.28	0.33	0.07
F5	-0.15	0.00	-0.11	-0.03	-0.11	-0.03
F5	-0.10	-0.07	-0.09	-0.36	-0.05	-0.13
F7	-0.29	-0.03	†	-0.19	-0.02	-0.19
F8	-0.07	-0.10	-0.13	-0.13	-0.05	-0.13
F9	1.96	1.31	2.03	1.24	1.94	1.58
F10	1.98	1.31	1.54	1.87	1.92	†
F11	†	2.17	†	1.61	†	†
F12	†	0.05	-0.01	†	0.05	†

† No optimal solution was found within the limit of 2000 iterations.

Table 4.9: Relative error of the objective function with respect to LOQO for each family and problem with KNITRO 10.1.0 conjugate gradient algorithm 2 on “server 2” without absolute value.

However, even discarding problem instances of families F9, F10 and F11, there are still some problem instances that have the relative error bigger than 0.30. Using spherical coordinates in next chapter the relative error will be enhanced. This is one of the main advantages of spherical coordinates.

We may therefore conclude that using Cartesian coordinates, LOQO and KNITRO with the direct algorithm 1 produces good quality lenses. However, some of the lenses obtained using IPOPT (all 3 versions) and KNITRO with the conjugate gradient algorithm 2 are poor quality lenses (compared to those lenses obtained with LOQO and KNITRO with the direct algorithm 1). This is due to the fact that the problem is nonconvex. Different solutions were obtained using different solvers, but some solvers (LOQO and KNITRO with the direct algorithm 1) produce better lenses than the other solvers (IPOPT -all 3 versions- and KNITRO with the conjugate gradient algorithm 2).

In the next chapter, it will be seen that using spherical coordinates with a fixed problem instance and the 6 different solvers the same solution will be obtained (unlike for Cartesian coordinates).

4.7 Orienting the power and astigmatism gradients

This Section presents a new way of orienting the power and astigmatism gradients. These orientations are obtained by modifying the objective function. The optimization model defined in Section 4.5 (page 27) is used, replacing the objective function (4.26) (page 28) with a new objective function. The variables, parameters and constraints are the same as in the previous model.

We propose two ways of orienting the power and astigmatism gradients. One way is orienting the gradient in a direction between 0 and $\frac{\pi}{2} rad$. This orientation is obtained with the new objective function:

$$
\begin{aligned}
\min \sum_{(i,j)\in\mathcal{G}} \frac{1}{n^2}\Bigg(& w_1\Big(Ast(x_i,y_j)\Big)^2 + \\
& w_4\Bigg(cos^2(\alpha)\Big(\frac{\partial Ast(x_i,y_j]}{\partial x}\Big)^2 + sin^2(\alpha)\Big(\frac{\partial Ast(x_i,y_j)}{\partial y}\Big)^2\Bigg) + \\
& w_5\Bigg(cos^2(\beta)\Big(\frac{\partial Pow(x_i,y_j)}{\partial x}\Big)^2 + sin^2(\beta)\Big(\frac{\partial Pow(x_i,y_j)}{\partial y}\Big)^2\Bigg)\Bigg),
\end{aligned}
\tag{4.35}
$$

where n is the number of points in each dimension of the grid, $w_1, w_4, w_5 \in [0,1]$, $0 \leq \alpha \leq \frac{\pi}{2}$, and $0 \leq \beta \leq \frac{\pi}{2}$. This objective function produces progressive lenses with the gradient oriented vertically, horizontally or with a direction between $0rad$ and $\frac{\pi}{2}rad$.

The second way to orient the gradients is orienting them at an angle between $0rad$ and πrad. This produces fairly asymmetrical lenses not only due to position of the near zone, but also due to the orientation of the gradients. The objective function is:

$$
\begin{aligned}
\min \sum_{(i,j)\in\mathcal{G}} \frac{1}{n^2}\Bigg(& w_1\Big(Ast(x_i,y_j)\Big)^2 + \\
& w_6\Bigg(cos(\alpha)\frac{\partial Ast(x_i,y_j]}{\partial x} + sin(\alpha)\frac{\partial Ast(x_i,y_j)}{\partial y}\Bigg)^2 + \\
& w_7\Bigg(cos(\beta)\frac{\partial Pow(x_i,y_j)}{\partial x} + sin(\beta)\frac{\partial Pow(x_i,y_j)}{\partial y}\Bigg)^2\Bigg),
\end{aligned}
\tag{4.36}
$$

where n is the number of points in each dimension of the grid, $w_1, w_6, w_7 \in [0,1]$, $0 \leq \alpha \leq \pi$, and $0 \leq \beta \leq \pi$.

The optimization problem solved in this section is the minimization of (4.35) or (4.36), subject to constraints (4.27), (4.28), (4.29), (4.30) and (4.25) using variables (4.24).

When using the objective function (4.35), having $w_4 = 0$ or $w_5 = 0$ is recommended because it is not possible to orient the power and the astigmatism gradients in the same way. Similarly, when using the objective function (4.36), having $w_6 = 0$ or $w_7 = 0$ is recommended.

We generated a set of 9 problem instances, denoted as D1, D2,...,D9. These instances have the parameters defined in Section 4.6.1 (page 29). The far regions, near regions and astigmatism regions are those of the instances of family F10 in Section 4.6.1, and the position of the near zone is that of type T6 of Table 4.1 (page 30). However, the objective function has been replaced by a new objective function. Instances D1, D2 and D3 use the objective function (4.35) with $w_1 = 0$, $w_4 = 1$ and $w_5 = 0$. Instances D4, D5,...,D9 use the objective function (4.36) with $w_1 = 0$, $w_6 = 1$ and $w_7 = 0$. Table 4.10 indicates, for each problem, the objective function and the parameter α used. Since $w_5 = 0$ and $w_7 = 0$, parameter β is not used.

Note that minimizing (4.35) with $w_1 = 0$ or $w_4 = 1$, $w_5 = 0$ and $\alpha = 0$ is the same as minimizing (4.36) with $w1 = 0$, $w_6 = 1$ $w_7 = 0$ and $\alpha = 0$. Consequently, instances D1 and D4 produced the same result. Similarly, D3 and D7 also produced the same result.

Instances D1, D2,...,D9 were solved using the LOQO solver and the same stopping criteria as in Section 4.6.4 (page 34), sigfig= 4, inftol= 10^{-6}, on server 1. The number of iterations, CPU time and the optimal objective function are reported in Table 4.11.

Problem instance	Objective function	α (rad)	α (°)
D1	(4.35)	0 rad	0°
D2	(4.35)	$\frac{\pi}{4}$ *rad*	45°
D3	(4.35)	$\frac{\pi}{2}$ *rad*	90°
D4	(4.36)	0 rad	0°
D5	(4.36)	$\frac{\pi}{6}$ *rad*	30°
D6	(4.36)	$\frac{\pi}{3}$ *rad*	60°
D7	(4.36)	$\frac{\pi}{2}$ *rad*	90°
D8	(4.36)	$\frac{2\pi}{3}$ *rad*	120°
D9	(4.36)	$\frac{5\pi}{6}$ *rad*	150°

Table 4.10: Objective function and value of parameter α for each instance.

Problem instance	Number of iterations	CPU time	Objective function
D1(†)	75	100	0.001165
D2	134	263	0.001662
D3(‡)	87	124	0.000592
D4(†)	75	100	0.001165
D5	85	145	0.001279
D6	99	177	0.001197
D7(‡)	87	124	0.000592
D8	85	155	0.001056
D9	109	203	0.001143

† Note that D1 and D4 problems produce the same lens result.

‡ Note that D3 and D7 problems produce the same lens result.

Table 4.11: Number of iterations, CPU time and optimal value of the objective function for each instance.

Figure 4.17 shows the power and astigmatism maps for instances D1, D2 and D3. It is appreciated that the astigmatism map for D1 (top) is oriented horizontally, the astigmatism map for D3 (bottom) is oriented vertically, and the astigmatism map for D2 (middle) is a mixture of these. Note that the horizontal isolines on the astigmatism map for D1 (for example, around the point $(x = -15\ mm,\ y = 4\ mm)$) are very close together, whereas the isolines on the astigmatism map for D3 around the same point are very separated, and finally for D2 there is a compromise. Similarly, the vertical isolines on the astigmatism map for D3 (for example, around the point $(x = -2\ mm,\ y = -20\ mm)$) are very close together, whereas the vertical isolines for the same point for D1 and D2 are more separated. Figure 4.18 shows the astigmatism maps for instances D4, D5,...,D9. It is appreciated that the astigmatism map for each instance has a different orientation, thus producing asymmetrical maps. For example, we may note that the astigmatism map for D6 (middle left) has horizontal isolines around the line $y = 0\ mm$ that are separated differently, for $x > 10$ and $x < 10$, because $\alpha = 60°$.

Orienting the power and astigmatism gradients is helpful in personalizing progressive lenses according to each user, by using different data. One data is how a user moves their

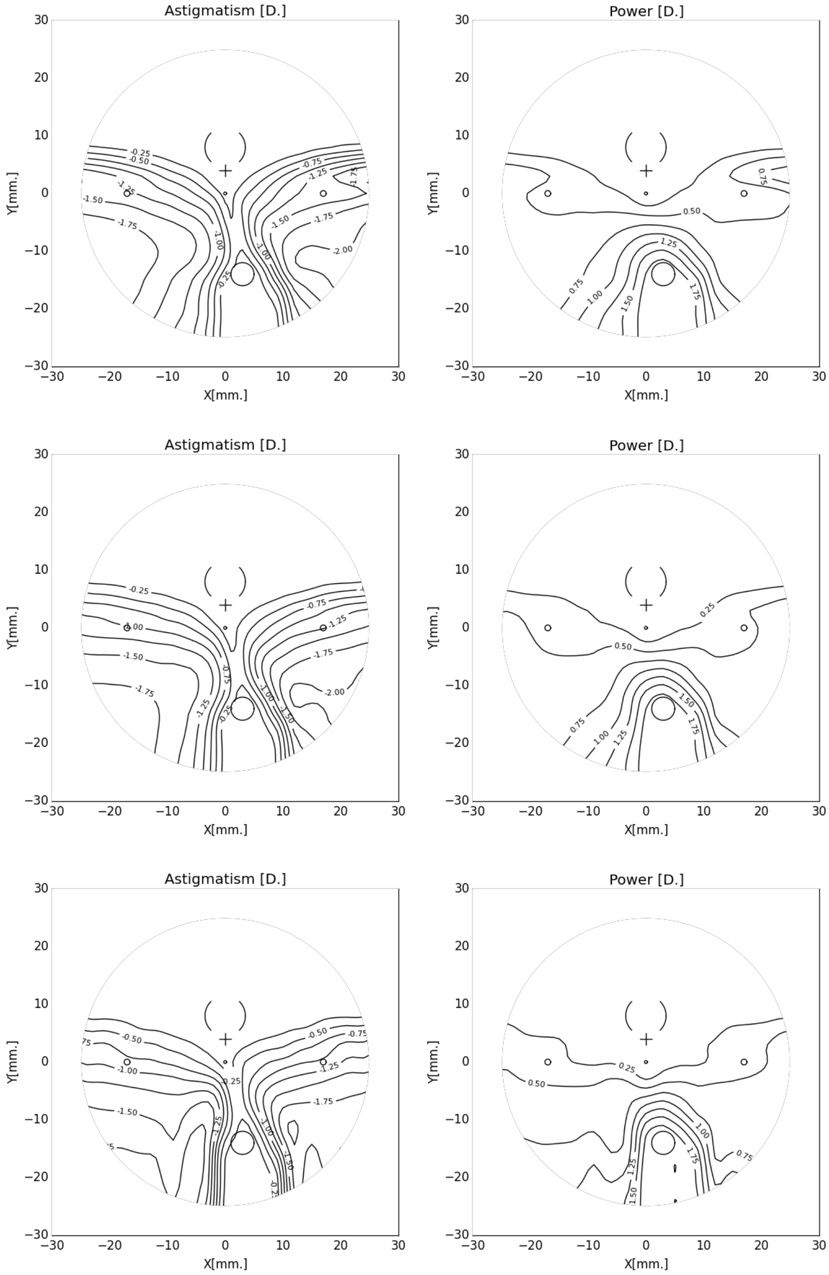


Figure 4.17: Astigmatism (right) and power (left) maps of the problems D1 (top), D2 (middle) and D3 (bottom).

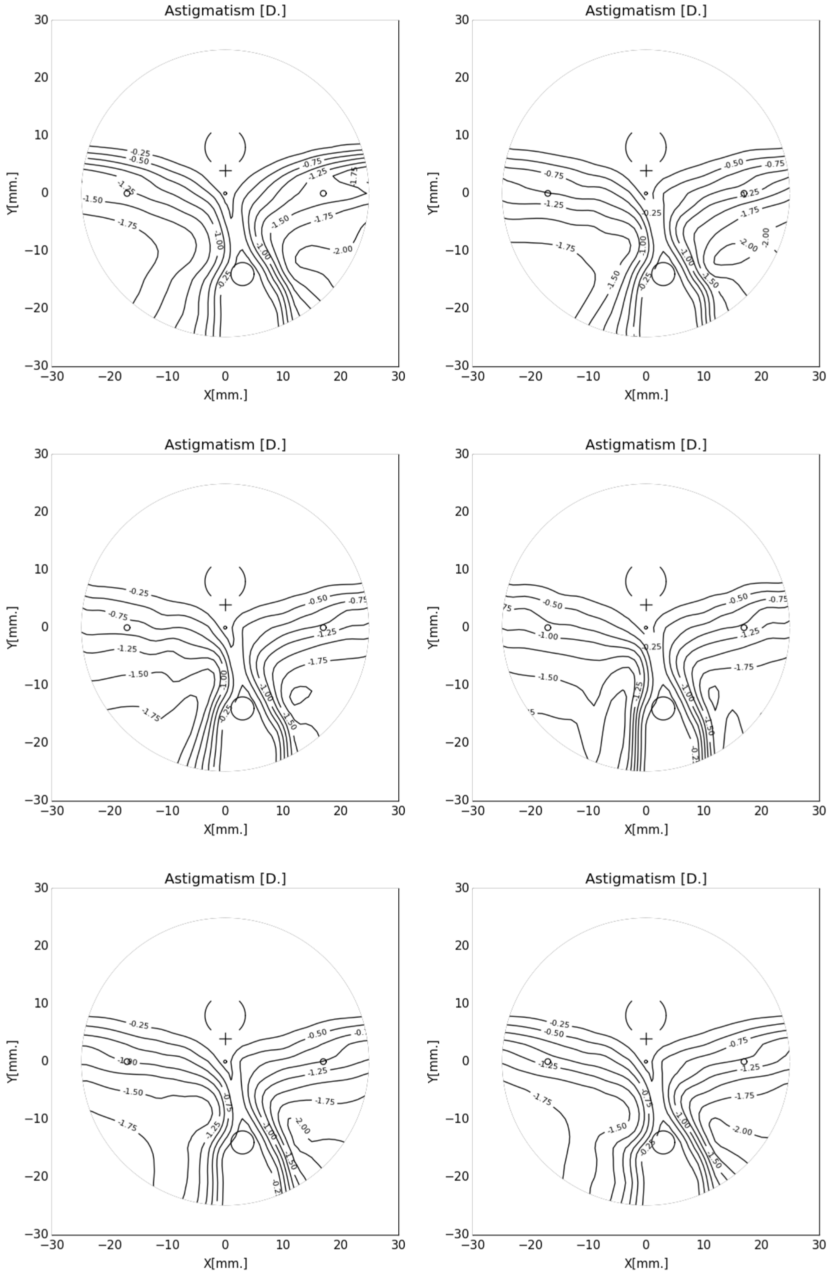


Figure 4.18: Astigmatism map of the problems D4 and D5 (top); D6 and D7 (middle); and D8 and D9 (bottom). From right to left.

eyes. For example, when using progressive lenses, there are *eye movers* and *head movers*, see Section 2.2.2 (page 5). Eye movers tend to move their eyes more vertically or more horizontally. Depending on eye movements, some orientations of power and astigmatism gradients are likely to fit better than others. Another application of orientating these gradients is relating the orientation of the astigmatism gradient to the axis on the cylinder of prescription.

All of the results described in this Section are described in full in the patent of invention [46]. The first author of this patent is the author of this thesis, and the patent was developed during the research for this thesis.

Chapter 5

Spherical coordinates

5.1 Introduction to the spherical coordinates used

When spherical coordinates are used in ophthalmic optics, the center of the coordinates is referenced at the center of the eye. As a simplification, the center of the eye can be considered to be the center of rotation of the eyeball. Using this model, the eye's angle of rotation is the angle of rotation of the model, and the point with radius 0 (with any angle) is the center of the eye. An example of using spherical coordinates in this way is given in Figure 1.21 of [22] and in Figure 5.1. The left part of Figure 5.1 shows the lateral view of a progressive surface with the spherical coordinates centered at the center of the eye. The right part of the same Figure 5.1 shows the same progressive lens in a front view. However, the model proposed here is entirely unrelated to this model.

As stated in Sections 3.2 (page 12) and 4.3 (page 23), the far zone of a progressive lens can be approximated by a sphere because the lens power in this region and, consequently, its curvature is nearly constant. The center of this sphere will be the center of the new proposed model in spherical coordinates. One angle is in the normal direction of the lens, while the other angle is in the perpendicular direction. The main motivation behind this new model is to obtain better convexity properties than previous existing models. The representation of this new model is shown in Figure 5.2. In this image, the radius is near a constant R for all the points on the far region; the radius is smaller for the points on the near region. The radius will vary gradually from the far region to the near region.

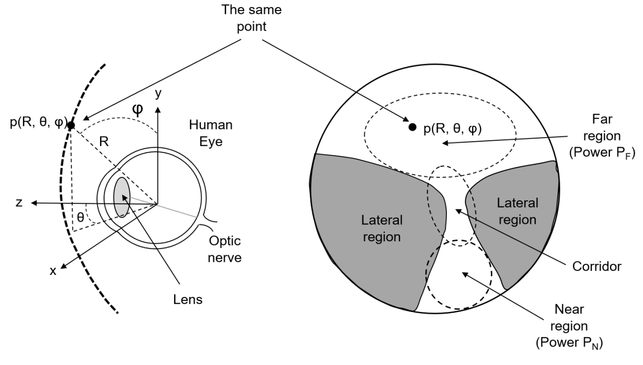
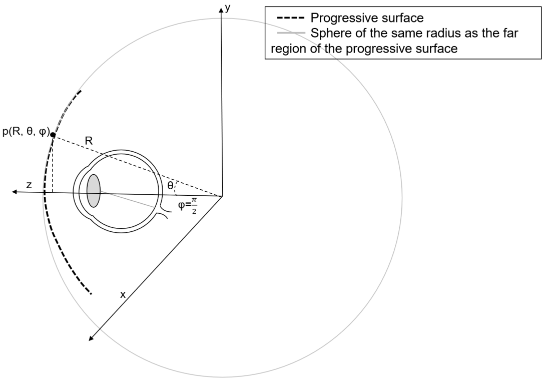


Figure 5.1: Example of spherical coordinates centered at the center of the eye.



y
Progressive surface
Sphere of the same radius as the far region of the progressive surface
p(R, θ, φ)
R
z
θ
φ=π/2
x

Figure 5.2: Example of spherical coordinates centered at the center of the sphere corresponding to the far region.

5.2 First attempt to use spherical coordinates

The first attempt made using spherical coordinates was the following parameterization:

$$\begin{aligned} \mathbb{R}^2 \longrightarrow & \mathbb{R}^3 \\ (\theta,\varphi) \longmapsto & \overrightarrow{p}(\theta,\varphi) = (x(\theta,\varphi),\ y(\theta,\varphi),\ z(\theta,\varphi)) = \\ & (R(\theta,\varphi)\cdot sin(\theta)\cdot cos(\varphi),\ R(\theta,\varphi)\cdot sin(\theta)\cdot sin(\varphi),\ R(\theta,\varphi)\cdot cos(\theta)) \end{aligned} \tag{5.1}$$

such that:

$$\begin{cases} x(\theta,\varphi) = & R(\theta,\varphi)\cdot sin(\theta)\cdot cos(\varphi) \\ y(\theta,\varphi) = & R(\theta,\varphi)\cdot sin(\theta)\cdot sin(\varphi) \\ z(\theta,\varphi) = & R(\theta,\varphi)\cdot cos(\theta) \end{cases} \quad where \quad \begin{cases} R(\theta,\varphi) \geq 0 \\ \theta \in [0,\frac{\pi}{2}] \\ \varphi \in [-\pi,\pi] \end{cases} \tag{5.2}$$

Considering a grid of angles $(\theta_i,\varphi_j),\ i=1,\dots,n,\ j=1,\dots,n$ where $\theta_i \in [0,\frac{\pi}{2}]$ and $\varphi_i \in [-\pi,\pi]$, we obtain:

$$\begin{cases} x_{ij} = & R_{ij}\cdot sin(\theta_i)\cdot cos(\varphi_j) \\ y_{ij} = & R_{ij}\cdot sin(\theta_i)\cdot sin(\varphi_j) \\ z_{ij} = & R_{ij}\cdot cos(\theta_j) \end{cases} \quad where \quad \begin{cases} R_{ij} \geq 0 \\ \theta_i \in [0,\frac{\pi}{2}] \\ \varphi_j \in [-\pi,\pi] \end{cases} \tag{5.3}$$

where R_{ij} are the variables and θ_i and φ_j are the parameters.

5.2.1 Example: A sphere

The solution of Example 1 (page 24) was a sphere of radius $\hat{R}$ centered at point $(x = 0\ mm, y = 0\ mm, z = 0\ mm)$, such that:

$$\hat{R} = \frac{\mu - 1}{P} = \frac{1.6 - 1}{5\ D} = 0.120\ m = 120.0\ mm$$

This sphere expressed in Cartesian coordinates was defined in (4.18) (page 25). The same sphere expressed in the spherical coordinates (5.3) is defined by $R_{ij} = \hat{R} = 120\ mm$ $\forall i, j = 1, \ldots, n$. Figure 5.3 represents the sphere of radius 120 mm using the first model (5.3) with spherical coordinates. In this figure the positive part of the sphere is considered, and the lens is a spherical cap of radius 120 mm. In left image of this figure, the range of angle θ is smaller ($\theta \in [0, \frac{\pi}{4}]$), in order to consider only the part corresponding to a lens. In right image of the same figure, the range or angle θ is bigger ($\theta \in [0, \frac{\pi}{2}]$), in order to consider all the positive part of the sphere. Figure 5.4 shows the ranges of the described angles.

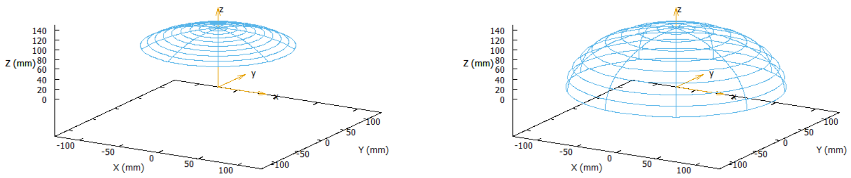


Figure 5.3: Part of a sphere of radius 120 mm centered at (0, 0, 0) using spherical coordinates. The ranges of the angles are different in left and right image. $\varphi \in [-\pi, \pi]$ (left and right) and $\theta \in [0, \frac{\pi}{4}]$ (left) and $\theta \in [0, \frac{\pi}{2}]$ (right). The axes are x, y, z (mm).

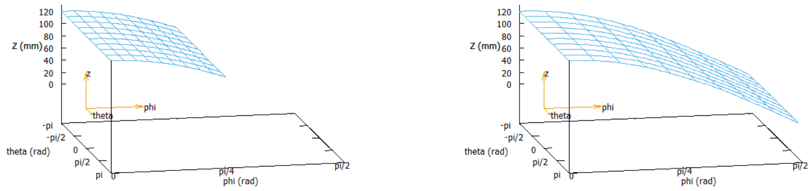


Figure 5.4: Plot of coordinates (θ, φ, R) of a part of a sphere of radius R=120 mm centered at (0, 0, 0) using spherical coordinates. $\varphi \in [-\pi, \pi]$ (left and right) and $\theta \in [0, \frac{\pi}{4}]$ (left) and $\theta \in [0, \frac{\pi}{2}]$ (right). The axes are θ, φ, z (mm).

In the next section, we will see that this parameterization resulted in certain discontinuity problems. Consequently, we moved to a new formulation, as explained in Section 5.3.

5.2.2 Calculation of the power and astigmatism of a surface using spherical coordinates

In this section, we will calculate the power and the astigmatism of a lens surface expressed in the spherical coordinates (5.1) (page 60). In the expression (5.1) (page 60), the coordinates (x, y, z) and R are expressed in mm, the refractive index is expressed in diopters ($\frac{1}{metre}$) and the angles θ and φ are expressed in radians.

Using the equations (3.4) (page 13), the coefficients of the first and second fundamental forms are recalculated as:

$$\begin{matrix} E = \overrightarrow{p_\theta} \cdot \overrightarrow{p_\theta} & F = \overrightarrow{p_\theta} \cdot \overrightarrow{p_\varphi} & G = \overrightarrow{p_\varphi} \cdot \overrightarrow{p_\varphi} \\ e = \overrightarrow{n} \cdot \overrightarrow{p_{\theta\theta}} & f = \overrightarrow{n} \cdot \overrightarrow{p_{\theta\varphi}} & g = \overrightarrow{n} \cdot \overrightarrow{p_{\varphi\varphi}} \end{matrix} \tag{5.4}$$

where $\cdot$ is the dot product, $\overrightarrow{n}$ is the normal vector to the surface:

$$n = \frac{\frac{\partial \overrightarrow{p}}{\partial \theta} \times \frac{\partial \overrightarrow{p}}{\partial \varphi}}{\left| \frac{\partial \overrightarrow{p}}{\partial \theta} \times \frac{\partial \overrightarrow{p}}{\partial \varphi} \right|}, \tag{5.5}$$

and the subscripts θ, φ are the partial derivatives with respect to θ or φ.

The first order partial derivatives at the point being studied can be computed by:

$$\begin{aligned} \overrightarrow{p_\theta} &= \left(\frac{\partial x}{\partial \theta}, \frac{\partial y}{\partial \theta}, \frac{\partial z}{\partial \theta} \right) \\ \overrightarrow{p_\varphi} &= \left(\frac{\partial x}{\partial \varphi}, \frac{\partial y}{\partial \varphi}, \frac{\partial z}{\partial \varphi} \right). \end{aligned} \tag{5.6}$$

Likewise, the second-order partial derivatives at the point being studied are:

$$\begin{aligned} \overrightarrow{p_{\theta\theta}} &= \left(\frac{\partial^2 x}{\partial \theta^2}, \frac{\partial^2 y}{\partial \theta^2}, \frac{\partial^2 z}{\partial \theta^2} \right) \\ \overrightarrow{p_{\theta\varphi}} &= \left(\frac{\partial^2 x}{\partial \theta \partial \varphi}, \frac{\partial^2 y}{\partial \theta \partial \varphi}, \frac{\partial^2 z}{\partial \theta \partial \varphi} \right) \\ \overrightarrow{p_{\varphi\varphi}} &= \left(\frac{\partial^2 x}{\partial \varphi^2}, \frac{\partial^2 y}{\partial \varphi^2}, \frac{\partial^2 z}{\partial \varphi^2} \right). \end{aligned} \tag{5.7}$$

Thus, the first derivatives can be calculated as:

$$\frac{\partial \overrightarrow{p}}{\partial \theta} = \begin{pmatrix} \frac{\partial R(\theta,\varphi)}{\partial \theta} \sin(\theta)\cos(\varphi) + R(\theta,\varphi)\cos(\theta)\cos(\varphi) \\ \frac{\partial R(\theta,\varphi)}{\partial \theta} \sin(\theta)\sin(\varphi) + R(\theta,\varphi)\cos(\theta)\sin(\varphi) \\ \frac{\partial R(\theta,\varphi)}{\partial \theta} \cos(\theta) - R(\theta,\varphi)\sin(\theta) \end{pmatrix}^\top, \tag{5.8}$$

$$\frac{\partial \overrightarrow{p}}{\partial \varphi} = \begin{pmatrix} \frac{\partial R(\theta,\varphi)}{\partial \varphi} \sin(\theta)\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\sin(\varphi) \\ \frac{\partial R(\theta,\varphi)}{\partial \varphi} \sin(\theta)\sin(\varphi) + R(\theta,\varphi)\sin(\theta)\cos(\varphi) \\ \frac{\partial R(\theta,\varphi)}{\partial \varphi} \cos(\theta) \end{pmatrix}^\top. \tag{5.9}$$

The second derivatives are:

$$\frac{\partial^2 \overrightarrow{p}}{\partial \theta \partial \theta} = \begin{pmatrix} \frac{\partial^2 R(\theta,\varphi)}{\partial \theta^2} \sin(\theta)\cos(\varphi) + 2\frac{\partial R(\theta,\varphi)}{\partial \theta} \cos(\theta)\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\cos(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial \theta^2} \sin(\theta)\sin(\varphi) + 2\frac{\partial R(\theta,\varphi)}{\partial \theta} \cos(\theta)\sin(\varphi) - R(\theta,\varphi)\sin(\theta)\sin(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial \theta^2} \cos(\theta) + 2\frac{\partial R(\theta,\varphi)}{\partial \theta} \sin(\varphi) - R(\varphi,\theta)\cos(\varphi) \end{pmatrix}^\top, \tag{5.10}$$

$$\frac{\partial^2 \overrightarrow{p}}{\partial\theta\partial\varphi} =$$
$$\begin{pmatrix} \frac{\partial^2 R(\theta,\varphi)}{\partial\theta\partial\varphi}\sin(\theta)\cos(\varphi) + \frac{\partial R(\theta,\varphi)}{\partial\varphi}\cos(\theta)\cos(\varphi) - \frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\sin(\varphi) - R(\theta,\varphi)\cos(\theta)\sin(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\theta\partial\varphi}\sin(\theta)\sin(\varphi) + \frac{\partial R(\theta,\varphi)}{\partial\varphi}\cos(\theta)\sin(\varphi) + \frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\cos(\varphi) + R(\theta,\varphi)\cos(\theta)\cos(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\theta\partial\varphi}\cos(\theta) - \frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta) \end{pmatrix}^\top, \tag{5.11}$$

$$\frac{\partial^2 \overrightarrow{p}}{\partial\varphi\partial\varphi} = \begin{pmatrix} \frac{\partial^2 R(\theta,\varphi)}{\partial\varphi^2}\sin(\theta)\sin(\varphi) - 2\frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta)\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\cos(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\varphi^2}\sin(\theta)\sin(\varphi) + 2\frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\sin(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\varphi^2}\cos(\theta) \end{pmatrix}^\top. \tag{5.12}$$

Finally, by using (3.12), (3.13) and (3.14) (page 14) we have obtained a method for calculating the power and astigmatism of a surface using spherical coordinates (5.2).

5.2.3 Example: Calculating the power and astigmatism of a sphere using spherical coordinates

Now let us consider a sphere of radius $\hat{R} = 120\ mm$ centered at point $(x = 0\ mm,\ y = 0\ mm,\ z = 0\ mm)$. In this case, we will examine the positive part of the sphere (see Figure 5.5). The purpose of this example is to calculate the power and the astigmatism of the surface (the sphere) and to discuss whether it is calculated correctly. In order to calculate the power and the astigmatism, we must calculate the first and second fundamental forms, as well as the partial derivatives of the surface. We define the radius as a cubic B-spline, in order to be able to obtain analytically the first and second partial derivatives with respect to θ_i and φ_i. The cubic B-spline are first defined in Cartesian coordinates in sections 3.5 (14) and 3.6 (page 16) and also in (4.25) (page 28). In this problem a cubic B-spline is defined in spherical coordinates. The Problem 1 to be solved in this Section is defined by the following parameters, variables, objective function and constraints.

Problem 1. .

1) The parameters are:

- $(\theta_i, \varphi_j) \in [0, \pi] \times [0, \pi]$, $(i, j) \in \mathcal{G} = \{1, \dots, n\} \times \{1, \dots, n\}$, *is a grid of angles (in radians) used for the definition of the lens in spherical coordinates, where* n *denotes the number of angles for each dimension of the grid.*

  n = 61 (n^2 is the number of angles in grid $\mathcal{G}$).

  The grid of angles $\mathcal{G}$ is computed by the formula $\theta_i = \varphi_i = 0.9817477042+$ $0.01963495(i-1)$ *rad, for* $i = 1, \dots, n$ *(where* $n = 61$*).*

- $(\theta'_{i'}, \varphi'_{j'}) \in [0, \pi] \times [0, \pi]$, $(i', j') \in \mathcal{G}' = \{1, \dots, o\} \times \{1, \dots, o\}$, *is another grid of angles (radians), with $\mathcal{G}'$ much coarser than $\mathcal{G}$ (i.e., $o \ll n$), where o is the number of angles used in the definition of a B-spline whose coefficients are the variables of the optimization model.*

- *o = 30 (o^2 is the number of angles in grid $\mathcal{G}'$).*

  The grid of angles $\mathcal{G}'$ is computed by the formula: $\theta'_{i'} = \varphi'_{i'} = 0.0 + 3.0(i-1)/30 = 0.1(i-1)$ rad, for $i' = 1, \ldots, o$ (where $o = 30$).

- *$R_{ij} = 120$ mm is the radius for each point of the lens (a sphere).*

2) The variables are:

$$\mathbb{R} \ni c(\theta'_{i'}, \varphi'_{j'}) \geq 0, \qquad (i', j') \in \mathcal{G}'. \tag{5.13}$$

Using a new B-spline in spherical coordinates we define the radius of the surface for the grid $\mathcal{G}$ as

$$R(\theta_i, \varphi_j) = \sum_{i'=1}^{o} \sum_{j'=1}^{o} c(\theta'_{i'}, \varphi'_{j'}) B^3_{i'}(\theta_i) B^3_{j'}(\varphi_j), \quad (i, j) \in \mathcal{G}, \tag{5.14}$$

where $B^3_{i'}(\theta_i)$ and $B^3_{j'}(\varphi_j), (i, j) \in \mathcal{G}, (i', j') \in \mathcal{G}'$, are the 1-dimensional three-degree B-splines basis defined in [44, page 100].

We remark that the B-splines basis is defined using the grids $(\theta_i, \varphi_j) \in [0, \pi] \times [0, \pi]$ and $(\theta'_{i'}, \varphi'_{j'}) \in [0, \pi] \times [0, \pi]$, $(i', j') \in \mathcal{G}' = \{1, \ldots, o\} \times \{1, \ldots, o\}$. Replacing these grids of points in equations (3.22)–(3.29) (page 16) we obtain the new B-splines basis in spherical coordinates.

3) Objective function: minimize 1

4) Constraints: $R(\theta_i, \varphi_j) = R_{ij}, \quad (i, j) \in \mathcal{G}$

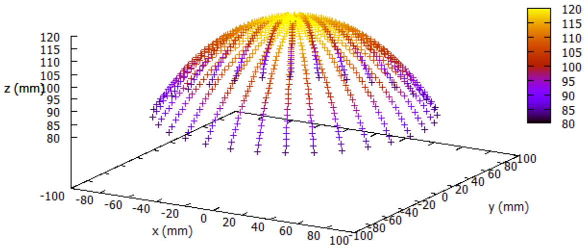


Figure 5.5: A part of a sphere of radius 120 mm, centered at $(0,\ 0,\ 0)$ of Problem 1.

This Problem 1 is modeled using the AMPL and solved using LOQO 6.0.6 on server 1. This problem converges in 4 iterations (it is a simple problem). At the optimum, we calculate the first and second fundamental forms, as well as the power and the astigmatism. The angles are shown in Figure 5.6, for each Cartesian coordinate $(x,\ y)$ and it is appreciated that angle φ has some discontinuity around line $y = 0mm$ and $x \leq 0mm$. The power and astigmatism are shown in Figure 5.7, ant it is appreciated that $Pow = 5\ D$ and $Ast = 0\ D$ for the entire surface. Figure 5.8 shows the values of the first and second fundamental forms. We appreciate that the first and second fundamental forms are continuous.

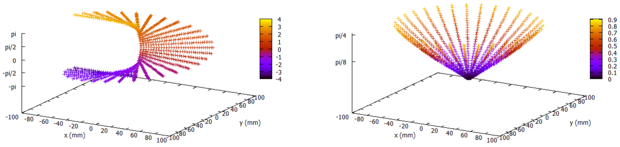


Figure 5.6: Values of the angles φ (left) and θ (right) for each point $(x_i,\ y_i)$.

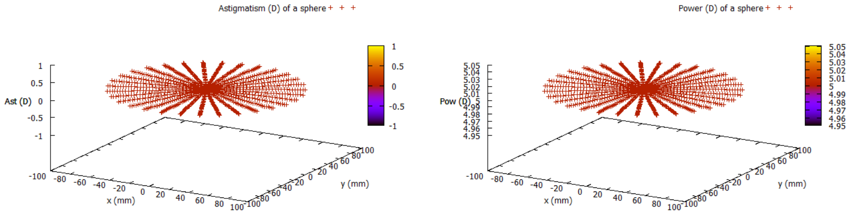


Figure 5.7: Values of the astigmatism (0D) and power (5D) for the Problem 1.

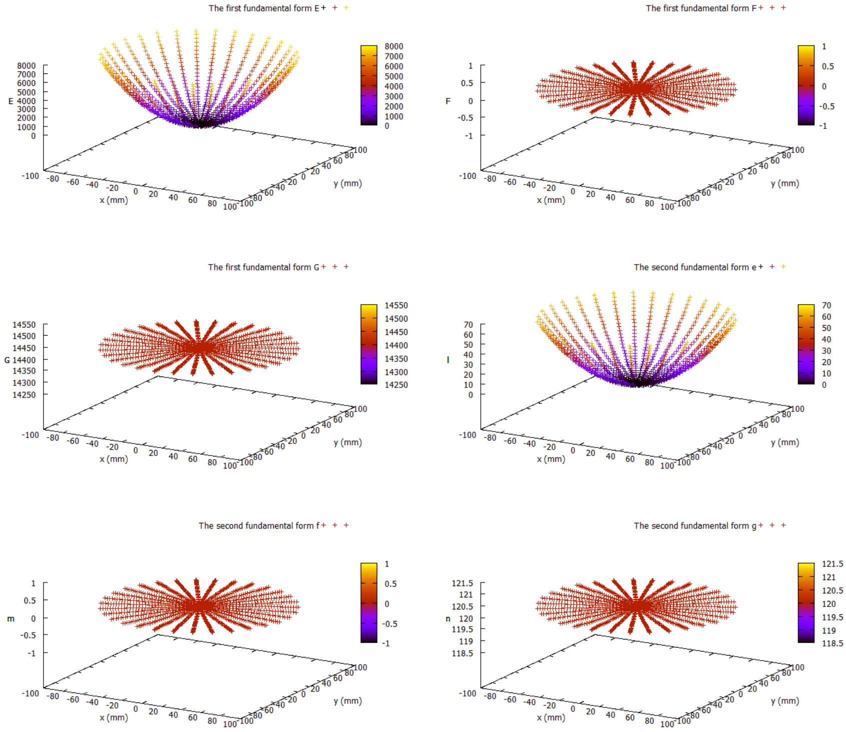


Figure 5.8: Values of the first and second fundamental forms for the Problem 1. The axes are $(x\ (mm),\ y\ (mm),$ the fundamental form).

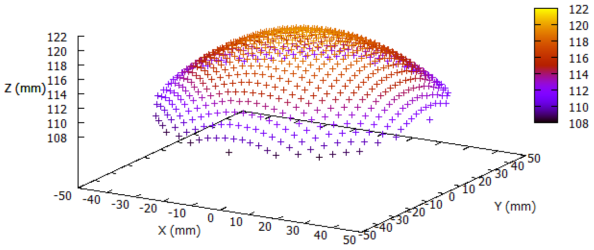


Figure 5.9: The progressive lens adjusted in Problem 2.

5.2.4 Example: Calculating the power and astigmatism of a progressive lens using spherical coordinates

Let us consider a progressive addition lens (see Figure 5.9), and we will calculate the astigmatism and the power for all points, as in the previous example (Problem 1). We will consider the same spherical coordinates and the same grids of angles as in Problem 1. The difference is that instead of considering a constant radius $R(\theta_i, \varphi_j) = 120\,mm$ for all of the points, in this example we will consider the radius that we have previously calculated for a progressive lens for each point of the grid (θ_i, φ_j) (for example a progressive lens calculated previously in Cartesian coordinates). In order to define the radius for each point and its partial derivatives with respect to θ_i and respect to φ_j, we will define the radius as a cubic B-spline, as in the previous Problem 1 (see Section 5.2.3).

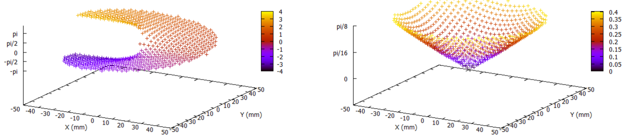


Figure 5.10: Values of the angles φ (left) and θ (right) for each point $(x_i,\ y_i)$.

Problem 2. *The new Problem defined here is the same as Problem 1 from Section 5.2.3 but replacing the parameters R_{ij} as it is explained.*

Parameters:

- $R_{ij} = \tilde{R_{ij}}$ *are the radius for a progressive lens calculated previously in Cartesian coordinates and converted into spherical coordinates using 5.3 (page 60).*

This problem converges in 3 iterations using the AMPL modeling language and the solver LOQO 6.0.6 on server 1. At the optimum, we calculated the first and second fundamental forms, as well as the power and the astigmatism, and the results are shown in Figures 5.11, 5.12 and 5.13.

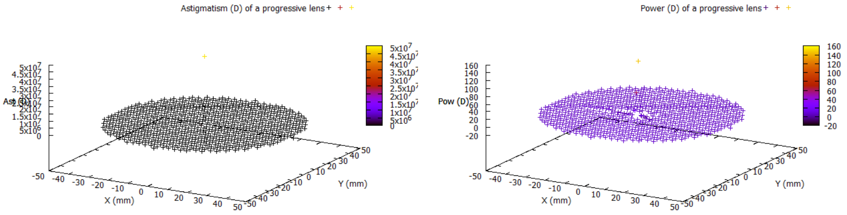


Figure 5.11: Values of the astigmatism (left) and power (right) for the progressive lens of Problem 2. There is a significant discontinuity around the center and around line $y = 0mm$ and $x \leq 0mm$.

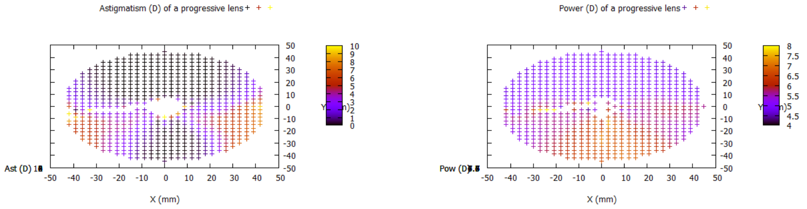


Figure 5.12: Values of the astigmatism (left) and power (right) for the progressive lens of Problem 2, discarding points with $Ast < 10D$ (left image), and discarding points with $Pow \leq 4D$ or $Pow \geq 8D$ (right). There is a significant discontinuity around the center and around line $y = 0$ and $x \leq 0$, where points lie outside of the z zoom axis.

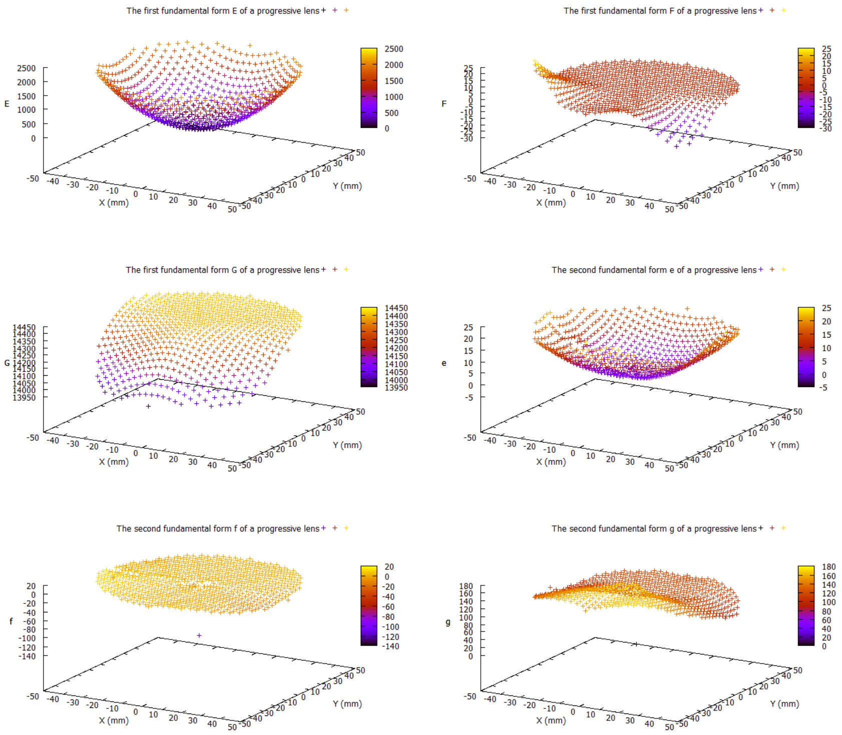


Figure 5.13: Values of the first and second fundamental forms for the Problem 2. The axes are (x (mm), y (mm), the fundamental form). There is a significant discontinuity for the fundamental form f (bottom left) for points $y = 0$ and $x \leq 0$.

For the entire surface, the power and astigmatism should be:

$$\begin{aligned} &P_F = 5\ D \leq Pow \leq P_N = 7\ D \\ &Ast^2 <= (3.0\ D)^2 \end{aligned} \tag{5.15}$$

because the progressive lens used for the input data in this example has the previous values of power and astigmatism. However, in Figure 5.11 it is clear that the power and the astigmatism show significant discontinuity around points $y = 0, x \leq 0$.

In Figure 5.12, the points with $Ast < 10\ D$ are plotted in the left image, and the points with $4\ D \leq Pow \leq 8\ D$ are plotted in the right image. Points outside of these ranges are not plotted. It is clear that some points around the line $y = 0, x \leq 0$ do not appear in these figures, because they lie outside of these ranges, and consequently they have been calculated incorrectly.

Finally, Figure 5.13 shows the first and second fundamental forms. Fundamental form f has a discontinuity around the line $y = 0, x \leq 0$.

In Figure 5.10 we may see that angle θ is not continuous around points $y = 0$ and $x \leq 0$. This is the cause that the calculation of the power and the astigmatism has a significant discontinuity around these points. For this reason, we will propose new spherical coordinates in the next section.

5.3 Second definition of spherical coordinates

The new formulation is:

$$\overrightarrow{p} = \begin{pmatrix} x(\theta,\varphi) \\ y(\theta,\varphi) \\ z(\theta,\varphi) \end{pmatrix} = \begin{pmatrix} R(\theta,\varphi)\cos(\theta) \\ R(\theta,\varphi)\sin(\theta)\cos(\varphi) \\ R(\theta,\varphi)\sin(\theta)\sin(\varphi) \end{pmatrix} \qquad \text{and} \qquad \theta,\varphi \in [0,\pi], \tag{5.16}$$

which is very similar to the previous formulation (5.3) (page 60) but avoids the singularities of the previous formulation. There are two differences. The first difference is that the new x is the old z, the new y is the old x and the new z is the old y. Applying this rotation, the range required for the angles θ, φ is $[0, \pi]$ (second difference) and there are consequently no discontinuities.

Considering a grid $(\theta_i, \varphi_j), i = 1, \ldots, n,\ j = 1, \ldots, n$ where each $\theta_i \in [0, \pi]$ and each $\varphi_j \in [0, \pi]$, we obtain

$$\begin{aligned} x_{ij} =&\ R_{ij}\cos(\theta_j) \\ y_{ij} =&\ R_{ij}\sin(\theta_j)\cos(\varphi_i) \\ z_{ij} =&\ R_{ij}\sin(\theta_j)\sin(\varphi_i), \end{aligned} \tag{5.17}$$

where R_{ij} are the variables and θ_i and φ_j the parameters of the model.

Using the new parameterization (5.16) (and (5.17)) a spherical cap of radius $R_{ij} = 120\ mm$ centered at $(0,\ 0,\ 0)$ is modeled, and we obtain the Figures 5.14 and 5.15. In Figure 5.14 (left), we appreciate a barrel-shaped $(x,\ y,\ z)$ grid.

5.3.1 Calculation of the power and astigmatism using the new spherical coordinate model

This section provides the expressions of the power and astigmatism of the surface defined using the new spherical coordinates (5.16). Firstly, the first and second derivatives of $\overrightarrow{p}(\theta, \varphi)$ need to be calculated:

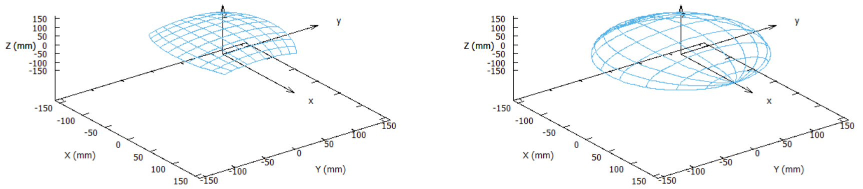


Figure 5.14: Part of a sphere of radius 120 mm using spherical coordinates. $\theta, \varphi \in [\frac{\pi}{4}, \frac{3\cdot\pi}{4}]$ (left) and $\theta, \varphi \in [0, \pi]$ (right). The axes are x, y, z (mm).

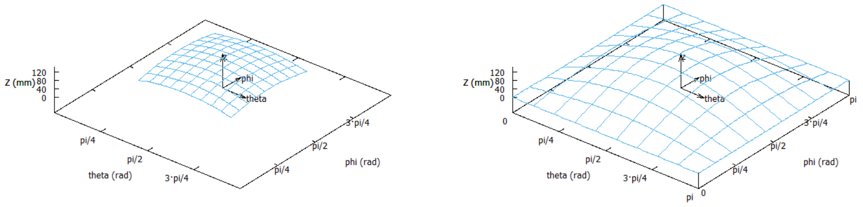


Figure 5.15: Part of a sphere of radius 120 mm using spherical coordinates. $\theta, \varphi \in [\frac{\pi}{4}, \frac{3\cdot\pi}{4}]$ (left) and $\theta, \varphi \in [0, \pi]$ (right). The axes are x, θ, φ (mm and radians).

$$\overrightarrow{p_\theta} = \begin{pmatrix} \frac{\partial R(\theta,\varphi)}{\partial\theta}\cos(\theta) - R(\theta,\varphi)\sin(\theta) \\ \frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\cos(\varphi) + R(\theta,\varphi)\cos(\theta)\cos(\varphi) \\ \frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\sin(\varphi) + R(\theta,\varphi)\cos(\theta)\sin(\varphi) \end{pmatrix}^\top, \tag{5.18}$$

$$\overrightarrow{p_\varphi} = \begin{pmatrix} \frac{\partial R(\theta,\varphi)}{\partial\varphi}\cos(\theta) \\ \frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta)\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\sin(\varphi) \\ \frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta)\sin(\varphi) + R(\theta,\varphi)\sin(\theta)\cos(\varphi) \end{pmatrix}^\top, \tag{5.19}$$

$$\overrightarrow{p_{\theta\theta}} = \begin{pmatrix} \frac{\partial^2 R(\theta,\varphi)}{\partial\theta^2}\cos(\theta) + 2\frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\varphi) - R(\varphi,\theta)\cos(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\theta^2}\sin(\theta)\cos(\varphi) + 2\frac{\partial R(\theta,\varphi)}{\partial\theta}\cos(\theta)\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\cos(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\theta^2}\sin(\theta)\sin(\varphi) + 2\frac{\partial R(\theta,\varphi)}{\partial\theta}\cos(\theta)\sin(\varphi) - R(\theta,\varphi)\sin(\theta)\sin(\varphi) \end{pmatrix}^\top, \tag{5.20}$$

$$\overrightarrow{p_{\theta\varphi}} = \begin{pmatrix} \frac{\partial^2 R(\theta,\varphi)}{\partial\theta\partial\varphi}\cos(\theta) - \frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\theta\partial\varphi}\sin(\theta)\cos(\varphi) + \frac{\partial R(\theta,\varphi)}{\partial\varphi}\cos(\theta)\cos(\varphi) - \frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\sin(\varphi) - R(\theta,\varphi)\cos(\theta)\sin(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\theta\partial\varphi}\sin(\theta)\sin(\varphi) + \frac{\partial R(\theta,\varphi)}{\partial\varphi}\cos(\theta)\sin(\varphi) + \frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\cos(\varphi) + R(\theta,\varphi)\cos(\theta)\cos(\varphi) \end{pmatrix}^\top, \tag{5.21}$$

$$\overrightarrow{p_{\varphi\varphi}} = \begin{pmatrix} \frac{\partial^2 R(\theta,\varphi)}{\partial\varphi^2}\cos(\theta) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\varphi^2}\sin(\theta)\sin(\varphi) - 2\frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta)\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\cos(\varphi) \\ \frac{\partial^2 R(\theta,\varphi)}{\partial\varphi^2}\sin(\theta)\sin(\varphi) + 2\frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\sin(\varphi) \end{pmatrix}^\top. \tag{5.22}$$

The coefficients of the first and second fundamental forms E, F, G and e, f, g are

$$\begin{array}{lll} E = \overrightarrow{p_\theta} \cdot \overrightarrow{p_\theta} & F = \overrightarrow{p_\theta} \cdot \overrightarrow{p_\varphi} & G = \overrightarrow{p_\varphi} \cdot \overrightarrow{p_\varphi} \\ e = \overrightarrow{n} \cdot \overrightarrow{p_{\theta\theta}} & f = \overrightarrow{n} \cdot \overrightarrow{p_{\theta\varphi}} & g = \overrightarrow{n} \cdot \overrightarrow{p_{\varphi\varphi}}, \end{array} \tag{5.23}$$

where

$$\overrightarrow{n} = \frac{\overrightarrow{p_\theta} \times \overrightarrow{p_\varphi}}{|\overrightarrow{p_\theta} \times \overrightarrow{p_\varphi}|}. \tag{5.24}$$

After some rather large calculations we finally obtain:

$$\begin{aligned} E =& \left(\frac{\partial R(\theta,\varphi)}{\partial\theta}\cos(\theta) - R(\theta,\varphi)\sin(\theta)\right)^2 + \\ & \left(\frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\cos(\varphi) + R(\theta,\varphi)\cos(\theta)\cos(\varphi)\right)^2 + \\ & \left(\frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\sin(\varphi) + R(\theta,\varphi)\cos(\theta)\sin(\varphi)\right)^2, \end{aligned} \tag{5.25}$$

$$\begin{aligned} F =& \left(\frac{\partial R(\theta,\varphi)}{\partial\theta}\cos(\theta) - R(\theta,\varphi)\sin(\theta)\right)\frac{\partial R(\theta,\varphi)}{\partial\varphi}\cos(\theta) + \\ & \left(\frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\cos(\varphi) + R(\theta,\varphi)\cos(\theta)\cos(\varphi)\right) \\ & \left(\frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta)\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\sin(\varphi)\right) + \\ & \left(\frac{\partial R(\theta,\varphi)}{\partial\theta}\sin(\theta)\sin(\varphi) + R(\theta,\varphi)\cos(\theta)\sin(\varphi)\right) \\ & \left(\frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta)\sin(\varphi) + R(\theta,\varphi)\sin(\theta)\cos(\varphi)\right), \end{aligned} \tag{5.26}$$

$$\begin{aligned} G =& \left(\frac{\partial R(\theta,\varphi)}{\partial\varphi}\cos(\theta)\right)^2 + \left(\frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta)\cos(\varphi) - R(\theta,\varphi)\sin(\theta)\sin(\varphi)\right)^2 + \\ & \left(\frac{\partial R(\theta,\varphi)}{\partial\varphi}\sin(\theta)\sin(\varphi) + R(\theta,\varphi)\sin(\theta)\cos(\varphi)\right)^2. \end{aligned} \tag{5.27}$$

Formulas for e, f, g are omitted because they are very large (in fact several pages long). Finally we can compute the power and astigmatism as

$$Pow(\theta,\varphi) = (\mu - 1)H(\theta,\varphi) \tag{5.28}$$

$$Ast(\theta,\varphi) = -2(\mu - 1)\sqrt{H(\theta,\varphi)^2 - K(\theta,\varphi)}, \tag{5.29}$$

where

$$\begin{aligned} K(\theta,\varphi) =& \det(A) = \frac{eg - f^2}{EG - F^2} \\ H(\theta,\varphi) =& \frac{1}{2}\mathrm{tr}(A) = \frac{eG - 2fF + gE}{2(EG - F^2)}. \end{aligned} \tag{5.30}$$

From the spherical coordinates equations (5.16), we can compute the power and the astigmatism for the entire lens surface (i.e., for points $(i, j) \in \mathcal{G}$) using the formulas (5.35) and (5.18)–(5.30). In particular, (5.28) and (5.29) provide the definition of power and astigmatism, respectively. We remark that the definition of $R(\theta,\varphi)$ as a B-spline allows us to calculate its derivatives using (5.18)–(5.20).

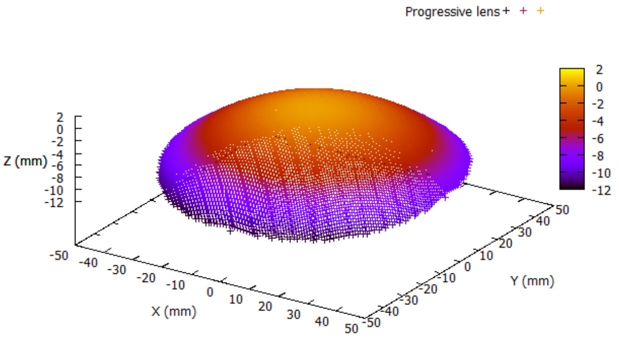


Figure 5.16: The progressive lens adjusted in Problem 3.

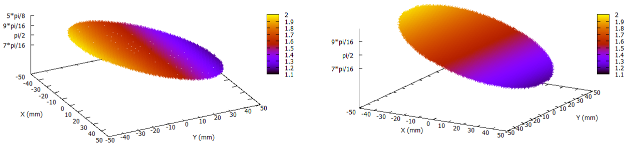


Figure 5.17: Values of the angles θ_i and φ_i for each point $(x_i,\ y_i)$.

5.3.2 Example: Calculating the power and astigmatism of a progressive lens using the new spherical coordinates

Here, we consider the same progressive addition lens as in Problem 2, but the power and the astigmatism will be calculated using the new spherical coordinates (5.16). We will see that using these new coordinates, the main discontinuities (that appeared in Problem 2 for points $y = 0$ and $x \leq 0$) disappear. The progressive lens to be adjusted is represented in Figure 5.16.

Problem 3. *The new Problem 3 defined here is the same as Problem 1 from Section 5.2.3 but we have replaced the parameters R_{ij} as follows:*

Parameters:

- $R_{ij} = \bar{R_{ij}}$ *are the radius for a progressive lens calculated previously in Cartesian coordinates and converted into spherical coordinates using* (5.16) *(page 69).*

This Problem 3 converges in 7 iterations using the AMPL modeling language and the solver LOQO 6.0.6 on server 1. The angles of Problem 3 are shown in Figure 5.17. Note that the angle φ for Problem 2 (see Figure 5.10 left) had a discontinuity for points $y = 0$ and $x \leq 0$, and in Problem 3 this discontinuity has disappeared. This is because the ranges of the angles for the model in new spherical coordinates are different.

The first and second fundamental forms are shown in Figure 5.18. Note that there are some discontinuities on the border of the fundamental forms, but these discontinuities appear on the border of the lens. Instead, the fundamental form f of Problem 2 (see

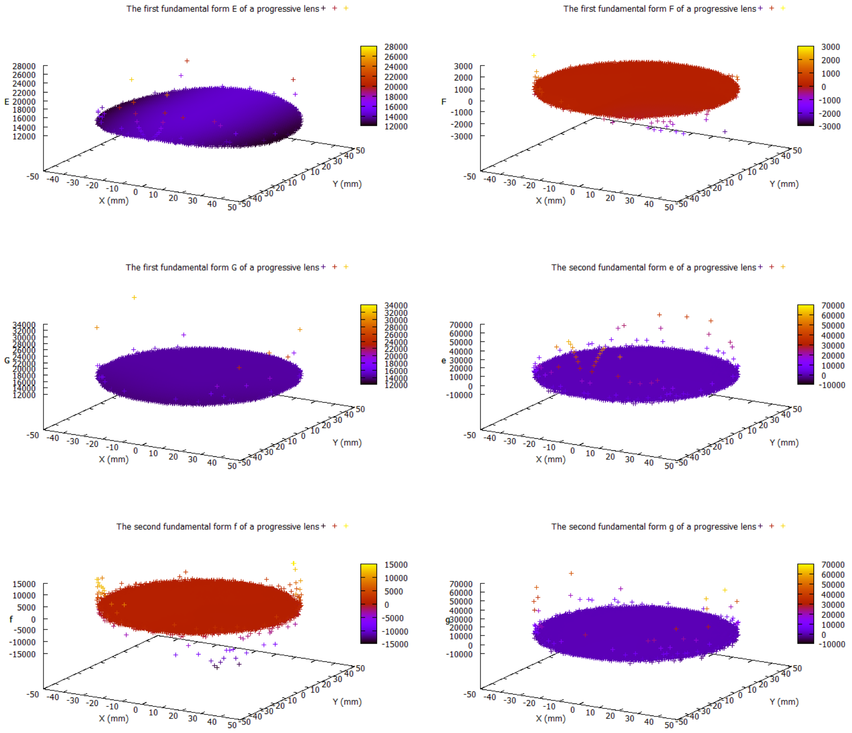


Figure 5.18: Values of the first and second fundamental forms for the Problem 3. The axes are (x (mm), y (mm), the fundamental form). There is a discontinuity for all the fundamentals forms on the border of the lens.

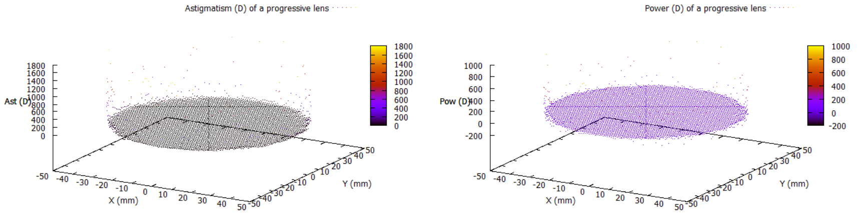


Figure 5.19: Values of the astigmatism and power for the Problem 3. There is a significant discontinuity on the border of the lens.

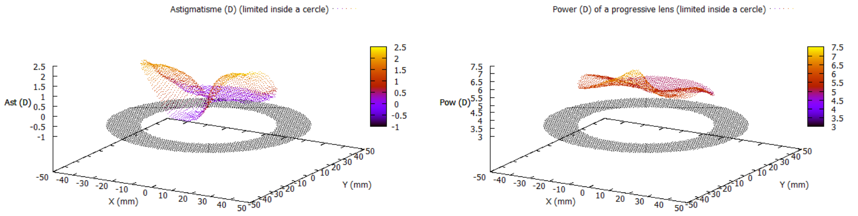


Figure 5.20: Values of the astigmatism and power for the Problem 3. We have only shown the values in the middle of the lens, to avoid discontinuities around the lens.

bottom left of Figure 5.13), showed a significant discontinuity in the middle of the lens, for points $y = 0$ and $x \leq 0$. Now this discontinuity has disappeared.

Finally, Figure 5.19 shows the astigmatism and the power of the adjusted lens. Since there are some discontinuities on the border of the lens, we can not appreciate the real power and astigmatism values. For this reason, in Figure 5.20 we represent the same power and astigmatism of the adjusted lens, but without displaying its border. In this Figure 5.20 it is appreciated that the constraints of the equation (5.15) (page 69, Problem 2) are accomplished, which did not happen in Problem 2.

In conclusion, we may state that by using the new spherical coordinates (5.16) we can fit correctly a progressive lens, except for the border. In next section it will be seen that, in addition, the spherical coordinates exhibit better convexity than Cartesian coordinates.

5.3.3 Convexity in spherical coordinates

As shown by Proposition 1, the model in spherical coordinates exhibits better convexity properties than the Cartesian model in Chapter 4 (page 21).

Proposition 1. *Using the new model in spherical coordinates* (5.16) *(page* (5.16)*), the feasible region of the optimization problem formulated in Example 2 (page 26)—which was not convex—becomes a convex set.*

Proof. Given the function

$$\begin{array}{rcll} R : G \subseteq \mathbb{R}^2 & \longrightarrow & \mathbb{R} & \\ (\theta_i, \varphi_i) & \longrightarrow & R(\theta_i, \varphi_i) & \qquad i = 1, \ldots, n, \end{array}$$

and the spherical coordinates defined by (5.16), we find the radii $R(\theta_i, \varphi_i)$ solution to the following optimization problem:

$$\begin{array}{llll} \min\limits_{R(\theta_i,\varphi_i)} & 1 & & \\ \text{subject to} & Ast(\theta_i,\varphi_i) & = 0 & \forall(\theta_i,\varphi_i) \in G \\ P - \epsilon \leq & Pow(\theta_i,\varphi_i) & \leq P + \epsilon & \forall(\theta_i,\varphi_i) \in G \\ & R\left(\frac{\pi}{2},\frac{\pi}{2}\right) & = \frac{\mu-1}{Pow(\frac{\pi}{2},\frac{\pi}{2})} & \\ & \left.\frac{\partial R(\theta_i,\varphi_i)}{\partial \theta_i}\right|_{\theta_i=\frac{\pi}{2},\varphi_i=\frac{\pi}{2}} & = 0 & \\ & \left.\frac{\partial R(\theta_i,\varphi_i)}{\partial \varphi_i}\right|_{\theta_i=\frac{\pi}{2},\varphi_i=\frac{\pi}{2}} & = 0, & \end{array} \tag{5.31}$$

where $Ast(\theta_i, \varphi_i)$ and $Pow(\theta_i, \varphi_i)$ are defined in (5.28) and (5.29) (page 71).

Using the same data as in (4.21) (from Example 2), that is, $P = 5D$, $\epsilon = 0.12D$, $\mu = 1.6$, and (θ_i, φ_i) is a grid of angles, and where $0 \leq \theta_i, \varphi_i \leq \pi$, the solution of (5.31) is a set of spheres of radius $\hat{R}$ centered at point $R(\cdot,\cdot) = 0$, such that $\hat{R}_{min} \leq \hat{R} \leq \hat{R}_{max}$, where

$$\hat{R}_{min} = (\mu - 1)/(P + \epsilon) = (1.6 - 1)/(5 + 0.12) = 0.11719m = 117.19mm$$
$$\hat{R}_{max} = (\mu - 1)/(P - \epsilon) = (1.6 - 1)/(5 - 0.12) = 0.12295m = 122.95mm.$$

Considering two different solutions of (5.31):

$$\begin{array}{r} \text{Solution 1} : R^1(\theta_i, \varphi_i) = \hat{R}_{min} \quad \forall \theta_i \in [0, \ldots, \pi] \\ \text{Solution 2} : R^2(\theta_i, \varphi_i) = \hat{R}_{max} \quad \forall \varphi_i \in [0, \ldots, \pi], \end{array} \tag{5.32}$$

we have

$$aR^1(\theta_i,\varphi_i) + bR^2(\theta_i,\varphi_i) = R(\theta_i,\varphi_i) \qquad \begin{array}{r} \text{for some } \hat{R} \in [\hat{R}_{min}, \hat{R}_{max}] \\ \text{where } a+b=1 \text{ and } a,b \geq 0, \end{array} \tag{5.33}$$

thus proving that the solution set is convex. □

The model in spherical coordinates will be used in the following sections to compute progressive lenses. In addition, in the next sections progressive lenses will be calculated from scratch, as it is done in an industrial environment.

5.4 The optimization model

The optimization model in Cartesian coordinates is described in Section 4.5 (page 27). In order to convert this optimization model from Cartesian coordinates to spherical coordinates some few changes explained in the following paragraphs must be done.

Parameters of the model

The parameters $(x_i, y_j) \in [-r,r] \times [-r,r]$ and $(x'_{i'}, y'_{j'}) \in [-r,r] \times [-r,r]$ are replaced by the parameters:

- $(\theta_i, \varphi_j) \in [0,\pi] \times [0,\pi]$, $(i,j) \in \mathcal{G} = \{1,\ldots,n\} \times \{1,\ldots,n\}$, is a grid of angles (in radians) used for the definition of the lens in spherical coordinates, where n denotes the number of angles for each dimension of the grid. The grid is defined such that $(\theta_{\frac{1+n}{2}}, \varphi_{\frac{1+n}{2}}) = (\frac{\pi}{2}, \frac{\pi}{2})$. Obviously, in the partition of $\mathcal{G}$ in spherical coordinates $\mathcal{G} = \mathcal{F} \cup \mathcal{N} \cup \mathcal{B}$, the sets $\mathcal{F}$, $\mathcal{N}$ and $\mathcal{B}$ are sets of angles (θ_i, φ_j). In Cartesian coordinates these sets were sets of points (x_i, y_j).
- $(\theta'_{i'}, \varphi'_{j'}) \in [0,\pi] \times [0,\pi]$, $(i',j') \in \mathcal{G}' = \{1,\ldots,o\} \times \{1,\ldots,o\}$, is another grid of angles (radians), with $\mathcal{G}'$ much coarser than $\mathcal{G}$ (i.e., $o \ll n$), where o is the number of angles used in the definition of a B-spline whose coefficients are the variables of the optimization model.

A new parameter is used:

- T is a tolerance expressed in mm that appears when bounding the variable radius (this parameter did not exist in the Cartesian coordinate model).

The other parameters (P_F, P_N, μ, $\mathcal{F}$, $\mathcal{N}$, $\mathcal{G}$ and their partitions, w_1, w_2 and w_3) are the same as the parameters defined in the Cartesian coordinate model in Section 4.5 (page 27).

Variables of the model

The variables (4.24) and (4.25) (page 28) of the model in Cartesian coordinates are replaced by these variables:

$$\mathbb{R} \ni c(\theta'_{i'}, \varphi'_{j'}) \geq 0, \qquad (i',j') \in \mathcal{G}'. \tag{5.34}$$

Using a new B-spline in spherical coordinates we define the radius of the surface for the grid $\mathcal{G}$ as

$$R(\theta_i, \varphi_j) = \sum_{i'=1}^{o} \sum_{j'=1}^{o} c(\theta'_{i'}, \varphi'_{j'}) B^3_{i'}(\theta_i) B^3_{j'}(\varphi_j), \quad (i, j) \in \mathcal{G}, \tag{5.35}$$

where $B^3_{i'}(\theta_i)$ and $B^3_{j'}(\varphi_j), (i, j) \in \mathcal{G}, (i', j') \in \mathcal{G}'$, are the 1-dimensional three-degree B-splines basis defined in [44, page 100].

We remark that the B-splines basis is defined using the grids $(\theta_i, \varphi_j) \in [0, \pi] \times [0, \pi]$ and $(\theta'_{i'}, \varphi'_{j'}) \in [0, \pi] \times [0, \pi]$, $(i', j') \in \mathcal{G}' = \{1, \ldots, o\} \times \{1, \ldots, o\}$. Replacing these grids of points from equations (3.22)–(3.29) (page 16) we obtain the new B-splines basis in spherical coordinates.

Objective function

The objective function (4.26) (page 28) is replaced by the following objective function:

$$\begin{aligned} \min \sum_{(i,j) \in \mathcal{G}} \frac{1}{n^2} \Bigg(& w_1 \Big(Ast(\theta_i, \varphi_j) \Big)^2 + \\ & w_2 \bigg(\Big(\frac{\partial Ast(\theta_i, \varphi_j]}{\partial \theta} \Big)^2 + \Big(\frac{\partial Ast(\theta_i, \varphi_j)}{\partial \varphi} \Big)^2 \bigg) + \\ & w_3 \bigg(\Big(\frac{\partial Pow(\theta_i, \varphi_j)}{\partial \theta} \Big)^2 + \Big(\frac{\partial Pow(\theta_i, \varphi_j)}{\partial \varphi} \Big)^2 \bigg) \Bigg), \end{aligned} \tag{5.36}$$

where n is the number of angles in each dimension of the grid.

Similarly to Cartesian coordinates, the astigmatism and also its derivatives are squared to avoid discontinuities around zero.

We may also note that, in models based on Cartesian coordinates, the partial derivatives of the objective function are with respect to x and y instead of θ and φ, and the objective function has a similar formulation (4.26) (page 28), where $\bar{\mathcal{G}}$ represents a grid of points (not a grid of angles). Expressions (5.36) and (4.26) are not exactly the same in terms of numerical results, but both have the same goal of producing a surface with the minimum astigmatism value and the smoothest possible distribution of power and astigmatism (that is, with isolines of power and astigmatism separated as much as possible).

Constraints

Constraints (4.27), (4.28) and (4.29) (page 29) are replaced by these constraints:

$$P_F - \epsilon_h \leq Pow(\theta_i, \varphi_j) \leq P_F + \epsilon_h \quad (i, j) \in \mathcal{F}_h, \ h = 1, \ldots, k \tag{5.37}$$

$$P_N - \delta_h \leq Pow(\theta_i, \varphi_j) \leq P_N + \delta_h \quad (i, j) \in \mathcal{N}_h, \ h = 1, \ldots, l \tag{5.38}$$

$$Ast(\theta_i, \varphi_j)^2 \leq \beta_h^2 \quad (i, j) \in \mathcal{A}_h, \ h = 1, \ldots, m. \tag{5.39}$$

Similarly, constraints (4.30) (page 29) are replaced by these ones, that impose conditions in the midpoint of the grid $\left(\frac{\pi}{2}, \frac{\pi}{2}\right)$ (and on the position and orientation of the lens surface):

$$\begin{array}{ll} R\left(\frac{\pi}{2},\frac{\pi}{2}\right) & = \frac{1-\mu}{P_F} \\ \left.\frac{\partial R(\theta_i,\varphi_i)}{\partial \theta}\right|_{\theta_i=\frac{\pi}{2},\varphi_i=\frac{\pi}{2}} & = 0 \\ \left.\frac{\partial R(\theta_i,\varphi_i)}{\partial \varphi}\right|_{\theta_i=\frac{\pi}{2},\varphi_i=\frac{\pi}{2}} & = 0. \end{array} \tag{5.40}$$

Finally, the last set of constraints are the bounds of variables $R(\theta_i, \varphi_i)$ and a bound of the power, for the entire surface of the lens:

$$\begin{array}{lll} R(\theta_i,\varphi_i) & \leq -\frac{1-\mu}{P_F} + T & (i,j) \in \mathcal{G} \\ R(\theta_i,\varphi_i) & \geq -\frac{1-\mu}{P_N} - T & (i,j) \in \mathcal{G} \\ Pow(\theta_i,\varphi_j) & \geq P_F & (i,j) \in \mathcal{G}; \end{array} \tag{5.41}$$

The first two groups of constraints of (5.41) bound the feasible region and were helpful for the convergence of the optimization solver (but they are not compulsory and are inactive in the optimal solution). The last group of constraints of (5.41) imposes a minimum value of power in all the points of the lens.

These last constraints (5.41) are not used in Cartesian coordinates and they have been added to the model in spherical coordinates in order to help the problem to converge. As these constraints apply for all n^2 points, the number of constraints in spherical coordinates is much larger than the number of constraints in Cartesian coordinates.

Similarly to Cartesian coordinates, the constraints (5.37), (5.38) and (5.39) and the last group of (5.41) refer to the properties of the lens (power and astigmatism). On the other hand, the constraints (5.40) and the first two groups of constraints (5.41) refer to the shape of the lens.

Finally, the optimization problem to be solved in spherical coordinates is minimizing (5.36), subject to constraints (5.37), (5.38), (5.39), (5.40), (5.41) and (5.35) using the variables (5.34).

The solution

Once the optimization problem has been solved, we obtain the optimal coefficients of the B-spline surface $c(\theta'_{i'}, \varphi'_{j'}) \geq 0, (i', j') \in \mathcal{G}'$. Using (5.35) we obtain $R(\theta_i, \varphi_j), (i, j) \in \mathcal{G}$ and then from (5.17) we obtain the Cartesian coordinates $(x_{ij}, y_{ij}, z_{ij}) \in \mathbb{R}^3$, $(i, j) \in \mathcal{G}$, where (x_{ij}, y_{ij}) define a square surface in $\mathbb{R}^2$, as required by the free-form generator. Finally, a file containing the points (x_{ij}, y_{ij}, z_{ij}) is sent to the free-form generator to physically cut the lens.

5.5 Numerical results

5.5.1 Problem instances

We generated a set of 15 problem instances, denoted as P1, P2,...,P15, which we obtained using different sets of parameters. However, some parameters are common for all 15 problems, such as:

- $P_F = 5$ D (power in the far region).
- P_N =7 D (power in the near region).

- $T = 0.045\ m$ (tolerance of bound (5.41)).
- $\mu = 1.6$ (refractive index of the lens material).
- $n = 61$ (n^2 being the number of angles in grid $\mathcal{G}$).
- $o = 30$ (o^2 being the number of angles in grid $\mathcal{G}'$, the grid used for defining the B-splines). Remember that the B-splines can be evaluated at any angle, not only for the list of o^2 angles. In particular, for each problem we have $n^2 = 61^2 = 3721$ points where the B-splines can be evaluated.
- $(\theta_i, \varphi_j), (i, j) \in \mathcal{G}$ (the particular angles used in grid $\mathcal{G}$).
- $(\theta'_{i'}, \varphi'_{j'}), (i', j') \in \mathcal{G}'$ (the particular angles used in grid $\mathcal{G}'$).

Note that:

- the subset $\mathcal{F}$ of far region angles and its partition in k far subregions $\mathcal{F} = \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_k$,
- the subset $\mathcal{N}$ of near region angles and its partition in l near subregions $\mathcal{N} = \mathcal{N}_1 \cup \cdots \cup \mathcal{N}_l$,
- and the partition of m subregions of astigmatism $\mathcal{G} = \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_m$,

are different for each problem.

The grid of angles $\mathcal{G}$ is computed by the formula $\theta_i = \varphi_i = 0.9817477042 + 0.01963495(i-1)$ rad, for $i = 1, \ldots, n$ (where $n = 61$). Expressing the angles in degrees we have $\theta_i = \varphi_i = 33.75 + 1.125(i-1)^\circ$, for $i = 1, \ldots, n$. For example, $\theta_1 = \varphi_1 = 0.9817477042$ rad (or 33.75°); $\theta_{31} = \varphi_{31} = 1.5707963268 = \frac{\pi}{2}$ rad (or 90°); and $\theta_{61} = \varphi_{61} = 2.1598449493$ rad (or 123.75°). The grid of angles $\mathcal{G}'$ is computed by the formula: $\theta'_{i'} = \varphi'_{i'} = 0.0 + 3.0(i-1)/30 = 0.1(i-1)$ rad, for $i' = 1, \ldots, o$ (where $o = 30$).

Among the parameters that differ for each problem we find $\epsilon_h\ h = 1, \ldots, k$, $\delta_h\ h = 1, \ldots, l$, and $\beta_h\ h = 1, \ldots, m$.

Optimization problem for a particular instance (P5)

Let us consider a particular instance, e.g., P5. For this problem we have $k = 4$ far regions, $l = 3$ near regions, and $m = 11$ astigmatism regions. The tolerances $\epsilon_h, h = 1, \ldots, 4$, of far region constraints (5.37) are, respectively, 0.03, 0.06, 0.12 and 0.25. For near region constraints (5.38), tolerances $\delta_h, h = 1, 2, 3$ are 0.03, 0.12, 0.25. Finally, the 11 tolerances β_h for astigmatism constraints (5.39) were 0.03, 0.12, 0.25, 0.03, 0.06, 0.12, 0.10, 0.15, 0.20, 0.25, 0.06. These tolerances are expressed in diopters (D).

The four, three and 11 regions corresponding, respectively, to far, near and astigmatism regions (defined by sets $\mathcal{F}_h$, $\mathcal{N}_h$, and $\mathcal{A}_h$) are shown in Figures 5.21, 5.22 and 5.23, using different colors and patterns for each subregion. In the same manner as for Cartesian coordinates, the near, far and astigmatism regions are concentric, to guarantee a gradual change in power and a gradual change in astigmatism.

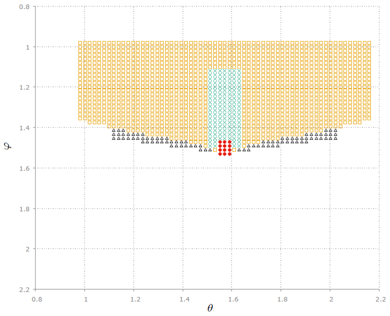


Figure 5.21: The four far regions of problem P5, each displayed in a different color and pattern.

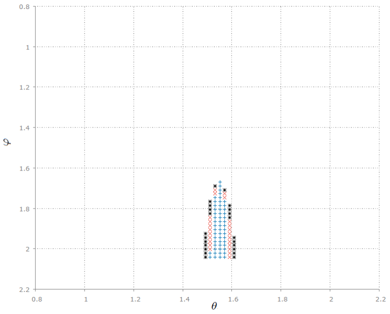


Figure 5.22: The three near regions of problem P5, each displayed in a different color and pattern.

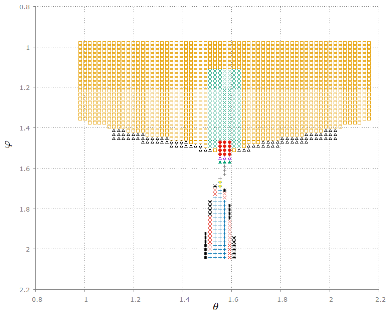


Figure 5.23: The 11 astigmatism regions of problem P5, each displayed in a different color and pattern.

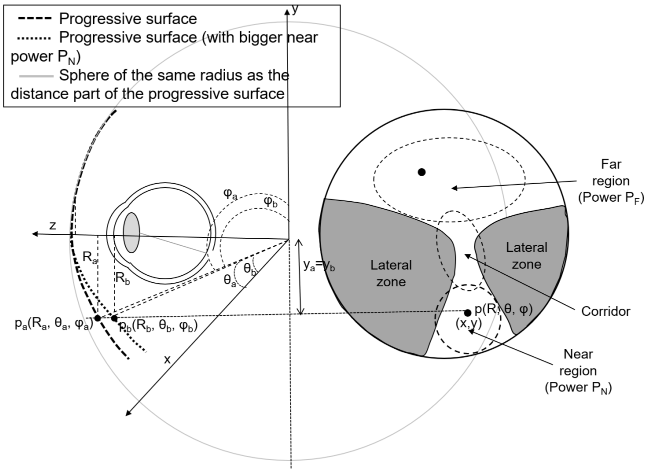

Progressive surface
Progressive surface (with bigger near power P$_N$)
Sphere of the same radius as the distance part of the progressive surface
y
z
x
φ$_a$
φ$_b$
θ$_a$
θ$_b$
R$_a$
R$_b$
y$_a$=y$_b$
p$_a$(R$_a$, θ$_a$, φ$_a$)
p$_b$(R$_b$, θ$_b$, φ$_b$)
Far region (Power P$_F$)
Lateral zone
Lateral zone
Corridor
p(R, θ, φ)
(x,y)
Near region (Power P$_N$)

Figure 5.24: Conversion of (x, y) near point from Cartesian to spherical coordinates.

Expressing the far, near and astigmatism regions in angles

Far, near and astigmatism regions are defined using the grids of angles $\mathcal{F}_h$, $\mathcal{N}_h$, and $\mathcal{A}_h$. Unfortunately, the designers of progressive lenses are limited to having the information about those grids in Cartesian coordinates (in mm). This is not a problem in a Cartesian model, because the constraints can be expressed in points (x, y). But in our spherical model, this information about the regions must be converted to angles (θ, φ). The conversion from (x, y) to (θ, φ) depends on $R(\theta, \varphi)$ as can be seen in (5.16). As $R(\theta, \varphi)$ is the solution of our problem, we have an issue to solve.

For instance, let us consider the near region in Cartesian coordinates (mm) of the right image of Figure 5.24, which corresponds to the set

$$\{(x, y) \in \mathbb{R}^2 : (x-3)^2 + (y+15)^2 \leq 5^2\},$$

that is, a circle of radius 5 mm centered at point (3 $mm, -15$ mm). Converting this region into spherical coordinates is not an easy task. We will focus on the conversion of the particular point (x, y) in the near region of the right image of Figure 5.24. The left image of the figure shows two progressive lenses with the same P_F (power in far region) and different P_N (power in near region). It can be seen in this left image that the conversion of point (x, y) from Cartesian to spherical coordinates depends not only on the shape of the lens, but also on P_F and P_N. In this figure, point (x, y) has two spherical coordinate projections in the $\mathbb{R}^3$ space in spherical coordinates. In the image we can appreciate that $\theta_a \neq \theta_b$ because $\overrightarrow{p}_a \neq \overrightarrow{p}_b$. This means that the grids of angles (for far, near and astigmatism regions), which are parameters of the optimization model, depend on the solution of the problem (the $R(\theta_i, \varphi_j)$). In order to solve this issue, a progressive lens of the same P_F and P_N (and μ) must be pre-calculated to obtain approximate radii values for the far and near regions; the (x, y) points are thus converted from Cartesian to spherical coordinates by using this approximate radius, and the inverse of equation (5.16). Using this technique, we can convert all the (far, near and astigmatism) regions from Cartesian to spherical coordinates.

In the case of this project, P_F, P_N and μ were the same for all 15 instances; therefore, once the radii for the far and near regions had been approximated, these approximations could be used for all 15 problems. The radius considered for the corridor region was approximated by the mean of the radii of the far and near regions. The complete procedure for generating and solving the 15 instances was thus as follows:

- First, compute a single progressive lens by fixing P_F, P_N and μ using Cartesian coordinates.
- Compute approximate radii for the far and near regions.
- Compute the average of these two radii. This new radius will be used for the corridor region, and for the other astigmatism regions that lie outside the far and near regions.
- Using these three radii, compute all the far, near, corridor and astigmatism regions for instances P1, P2, P3,. . . , P15 in spherical coordinates.
- Solve the 15 instances using the model in spherical coordinates.

It may be useful, in future research based upon this project, to improve the instance generator. For instance, one potential improvement would be using for the corridor an interpolated value (instead of an average) of the radii in the far and near regions, depending on the distance of the corridor points from the far and near regions.

Objective function

The objective function (4.26) was used for all 15 instances, using the different weights w_1, w_2, and w_3:

- $w_1 = 0, w_2 = 0, w_3 = 0$, for problems P1 and P2 (that is, the objective function is a constant).
- $w_1 = 1, w_2 = 0, w_3 = 0$, for problems P3, P4 and P5.
- $w_1 = 0, w_2 = 1, w_3 = 0$, for problems P6, P7 and P8.
- $w_1 = 0, w_2 = 0, w_3 = 1$, for problems P9, P10 and P11.
- $w_1 = 1, w_2 = 1, w_3 = 0$, for problems P12, P13.
- $w_1 = 1, w_2 = 0, w_3 = 1$, for problems P14 and P15.

In order to simplify our problem we did not use weights others than 0 or 1. Using other weights in the objective function makes it difficult to comprehend the results in the optimal solution. The units of the objective function are D^2 for problems P3, P4 and P5; $\frac{D^2}{\mathrm{rad}^2}$ for problems P6, P7, P8, P9, P10 and P11; and the sum of D^2 and $\frac{D^2}{\mathrm{rad}^2}$ for problems P12, P13, P14 and P15.

Finally, Table 5.1 reports the number of constraints for the 15 instances generated. The number of variables is always the same, $o^2 = 30^2 = 900$, which is the number of points in the grid $\mathcal{G}'$. In general we have around 16000 (nonlinear) constraints, and most of them are inequalities.

As stated above, the number of constraints in spherical coordinates is much bigger than the number of constraints in Cartesian coordinates due to the group of constraints (5.41) (page 77) (that are not used in Cartesian coordinates). Note that the number of

Problem	n. of constraints
P1	18057
P2	15870
P3	15870
P4	15936
P5	15870
P6	15836
P7	15836
P8	15836
P9	15870
P10	15836
P11	15836
P12	15870
P13	15936
P14	15870
P15	15836

Table 5.1: Number of constraints for each problem. The number of variables was always 900.

variables are very similar (900 variables in spherical coordinates—$o = 30$—and 961 in Cartesian coordinates—$o = 31$—). Note also that the grid of the points is always very similar in both models: there are 3481 points in Cartesian coordinates ($n^2 = 59^2 = 3481$ points) and 3721 points in spherical coordinates ($n^2 = 61^2 = 3721$ points).

5.5.2 Computational environment

As in Chapter 4, the optimization model was implemented using the AMPL modeling language linked with three different interior points solvers: LOQO, IPOPT and KNITRO. The same two servers as in Chapter 4 were used: server 1 and server 2. The servers and the versions of the solvers are fully described in Section 4.6.4 (page 34).

As for Cartesian coordinates, KNITRO with the active set (algorithm 3) and the SQP (algorithm 4) algorithms did not converge.

5.5.3 Stopping criteria

First, we chose one of the solvers (LOQO) with different stopping criteria in order to find tolerances for obtaining a sufficiently good quality progressive lens at the optimum. The quality of the lens must be evaluated in terms of optics, by analyzing the isolines of the optimal lens as well as the value of the objective function.

We ran the 15 problems using five different stopping criteria with LOQO, obtained by adjusting the tolerances sigfig (the number of equal digits in the primal and dual objective functions) and inftol (infeasibility tolerance for the primal and dual problems). Table 5.2 reports the values of the primal and dual objective functions in the last iteration for problem P12. We chose problem P12 because the objective function is affected by the square of the astigmatism as well as its partial derivatives. The relative error (last column of Table 5.2) is defined as: $|\text{primal o.f.} - \text{dual o.f.}| / |\text{primal o.f.}|$.

Stopping criteria	primal o.f.	dual o.f.	relative error
$\text{sigfig} = 2, \text{inftol} = 10^{-3}$	10.14186825	-1257.362782	124.9774321
$\text{sigfig} = 4, \text{inftol} = 10^{-6}$	1.634879469	1.395858085	0.146201227
$\text{sigfig} = 6, \text{inftol} = 10^{-3}$	1.62718076	1.580678331	0.028578527
$\text{sigfig} = 8, \text{inftol} = 10^{-3}$	1.626118778	1.625710862	0.000250853
$\text{sigfig} = 8, \text{inftol} = 10^{-12}$	1.626114492	1.626114214	1.7096E-07

Table 5.2: Primal objective value, dual objective value and relative error at the optimum, using LOQO 6.0.6 and five different stopping criteria for problem P12.

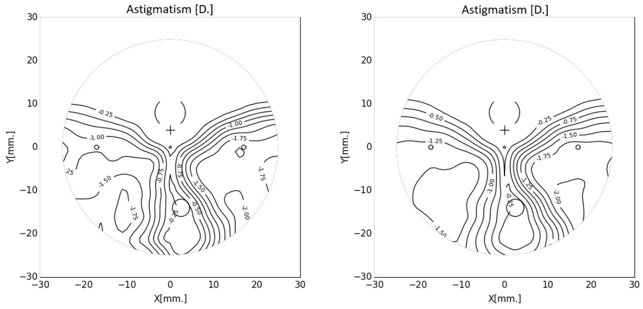


Figure 5.25: Astigmatism of the lens of P12 using LOQO 6.0.6 and two different stopping criteria: sigfig=2, inftol=10^{-3} (left) and sigfig=4, inftol=10^{-6} (right).

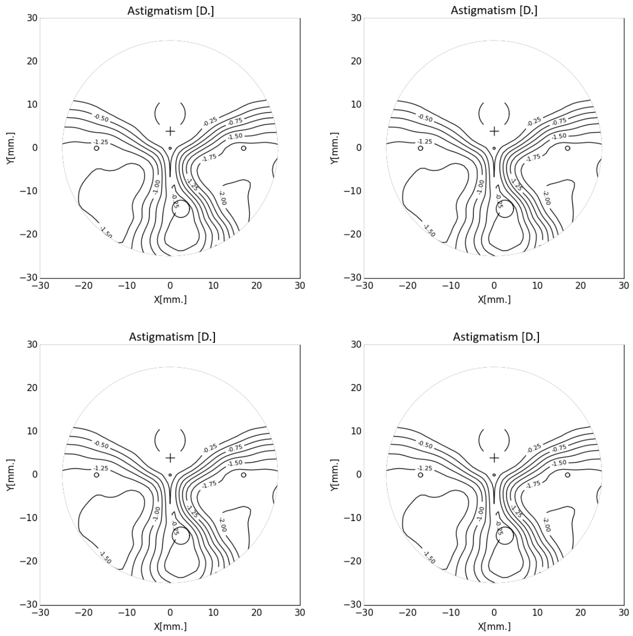


Figure 5.26: Astigmatism of the lens of problem P12 using LOQO 6.0.6 and four different stopping criteria: sigfig=4, inftol=10^{-6} (top left); sigfig=6, inftol=10^{-3} (top right); sigfig=8, inftol=10^{-3} (bottom left); and sigfig=8, inftol=10^{-12} (bottom right).

Problem	sigfig 4	sigfig 6	sigfig 8	sigfig 8
	inftol 10^{-6}	inftol 10^{-3}	inftol 10^{-3}	inftol 10^{-12}
P1	313	34	124	411
P2	385	29	30	461
P3	54	50	69	87
P4	60	53	86	94
P5	54	50	69	87
P6	36	55	109	131
P7	95	62	126	160
P8	36	55	109	131
P9	41	84	105	115
P10	33	49	63	68
P11	41	44	55	60
P12	42	48	63	74
P13	41	44	57	69
P14	36	44	58	65
P15	36	48	64	68

Table 5.3: Number of iterations for each problem with LOQO 6.0.6 and different stopping criteria.

Numerically, from the first line of Table 5.2, we can conclude that the first stopping condition sigfig= 2, inftol= 10^{-3} does not provide a suficiently high-quality solution. To evaluate the solutions in terms of the optical properties of the lens produced, Figure 5.25 shows the astigmatism map for the lens obtained using LOQO and two different stopping criteria; the units of these maps are diopters (D) for the isolines and the axes x and y are displayed in mm. We see that the left image (sigfig=2, inftol=10^{-3}) is blurrier than the right one (sigfig=4, inftol=10^{-6}). From an optics perspective, using sigfig=4, inftol=10^{-6} is prefereable to obtain a good quality lens.

Figure 5.26 shows the lenses obtained with the four last stopping criteria of Table 5.2 (i.e., the tighter ones). We observe that the four lenses obtained are of similar quality, thus, the most preferred will consequently be the stopping criteria that solves the problem faster (in terms of number of iterations, and thus also in terms of seconds). Table 5.3 shows the number of iterations required for all 15 instances and the four stopping criteria. If we were considering only P12, we would choose the stopping condition sigfig=4, inftol=10^{-6}; but for P1 and P2, the objective function is constant, and we see that the fastest executions were obtained with sigfig=6, inftol=10^{-3}. Note that using sigfig=4, inftol=10^{-6} for problems P1 and P2 required a large number of iterations to converge.

We conclude that the most suitable stopping criteria for LOQO 6.06 are sigfig= 6 and inftol=10^{-3}. Since each solver might have different parameters or tolerances, we chose for the remaining solvers those which are closest to sigfig= 6 and inftol=10^{-3}, as shown below.

5.5.4 Comparison of solvers

From this point forward, we will consider the same combination of solvers and servers and the same stopping condition as in Cartesian coordinates. The solvers, servers and stopping condition are described in Section 4.6.4 (page 34).

Problem	LOQO	IPOPT	IPOPT	IPOPT	KNITRO	KNITRO
	6.0.6	3.8.1	3.12.8	3.9.3	10.1.0 alg 1	10.1.0 alg 2
P1	34	80	80	79	150	24
P2	29	39	39	39	21	18
P3	50	44	44	44	36	22
P4	53	56	56	56	62	22
P5	50	44	44	44	36	22
P6	55	51	51	51	198	53
P7	62	46	46	46	376	48
P8	55	51	51	51	198	53
P9	84	49	49	49	53	25
P10	49	54	54	54	49	39
P11	44	51	51	51	53	34
P12	48	52	52	52	261	158
P13	44	66	66	66	262	43
P14	44	50	50	50	47	26
P15	48	51	51	51	48	26

Table 5.4: Number of iterations for each problem using the six different solvers.

All of the solvers reported the solutions obtained as “optimal” (LOQO and IPOPT) or either “locally optimal” or a “satisfactory solution” (KNITRO). It is worth noting the following: although the problem is nonlinear and nonconvex, which means each solver could provide a different local minima, visualizing the obtained lenses allowed us to observe that the six solutions found for each problem were the same (except for some negligible numerical differences).

Table 5.4 shows the number of iterations for the 15 problems and six solvers. The number of iterations for LOQO and IPOPT were between 29 and 84, while KNITRO exhibited greater variability: it performed between 21 and 376 iterations with the direct algorithm 1, and between 18 and 158 with the conjugate gradient algorithm 2. That is, in some cases KNITRO was the best solver (for example for instances P2 and P15) but it was the worst in others (for example, P12).

Figure 5.27 shows the number of iterations for the 15 problems P1–P15 using the six different solvers. Note that the number of iterations for the different versions of IPOPT are similar, and are superposed in the figure. Note that KNITRO with the direct algorithm 1 is the only solver that uses more than 180 iterations for some problems. It is the case of problems P6, P7, P8, P12 and P13, that are the problems with $w_2 \neq 0$. KNITRO with algorithm 2 (conjugate gradient) used fewer iterations than the other solvers (except for P12), but we will see later that KNITRO with this algorithm 2 provided the largest objectives.

Using spherical coordinates all the 15 problems converged with fewer than 2000 iterations with any of the 6 solvers. In fact, all the problems converged with fewer than or equal to 376 iterations (the maximum number of iterations was 376 iterations for problem P7 using KNITRO with the direct algorithm 1). However, using Cartesian coordinates, 2 or 3 problems (out 72) did not converge using IPOPT solver, and 13 problems (out 72) did not converge using KNITRO with the conjugate gradient algorithm 2. (See Table 4.3 (page 36).) This is an indication that the spherical coordinate model converges faster

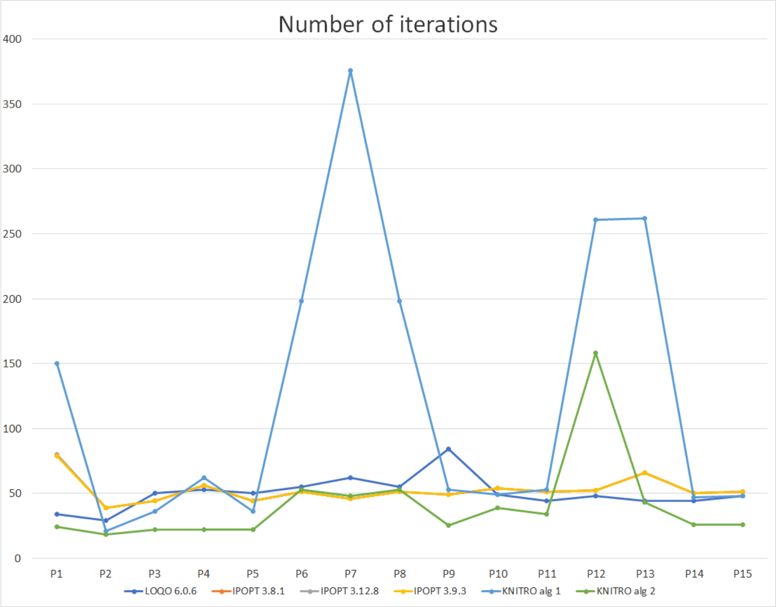


Figure 5.27: Number of iterations for each problem using the six different solvers.

(in terms of number of iterations) than the Cartesian coordinate model. For example, LOQO for the Cartesian coordinate model used between 42 and 192 iterations, while for the spherical coordinate model it required between 29 and 84 iterations. Note that differences with other solvers are larger, because some instances in Cartesian coordinates did not converge.

The CPU time, reported in Table 5.5, was proportional to the number of iterations. Recall that the first two solvers (LOQO and IPOPT 3.8.1) were executed on server 1 and the other four on server 2; this explains the different times between the first IPOPT version and the other two, while all of them had similar number of iterations (server 2 was on average 2.8 times faster than server 1).

In comparing the CPU time for spherical versus Cartesian coordinates, we want to discard the problem instances that did not converge. Recall that some problems in Cartesian coordinates did not converge, and used 2000 iterations (and consequently large CPU time). We chose LOQO solver because all problem instances converged using both, spherical (15 problem instances) and Cartesian coordinates (72 problem instances). Using LOQO, the CPU time with the spherical coordinates was between 112 and 925 seconds, while it was between 42 and 289 seconds with the Cartesian coordinate model, on the same server 1. That means that the Cartesian coordinate model is faster (in CPU time) than the spherical coordinate model. However, depending on the solver, not all the problems converged using the Cartesian coordinate model (see Table 4.3)—and all the problems converged using the spherical coordinate model.

In Section 4.6.5 (page 35) we defined the velocity as the number of iterations divided by the CPU time (seconds). Table 5.6 shows the velocity for the 15 problems and the six different solvers in spherical coordinates. Note that velocity depends on the server (server 2 is faster than server 1), on the objective function (calculating gradients is much slower than not calculating them) and on the coordinates model. In Table 5.6 there is a horizontal line separating 3 groups of problem instances, depending on the objective function: in P1 and P2 the objective function is a constant (very fast); in P3, P4, P5 the objective

Problem	LOQO	IPOPT	IPOPT	IPOPT	KNITRO	KNITRO
	6.0.6	3.8.1	3.12.8	3.9.3	10.1.0 alg 1	10.1.0 alg 2
	(server 1)	(server 1)	(server 2)	(server 2)	(server 2)	(server 2)
P1	112	211	90	93	128	51
P2	115	122	63	73	51	65
P3	165	134	66	75	60	65
P4	163	158	75	78	78	66
P5	165	135	74	68	62	64
P6	688	562	200	169	598	327
P7	769	512	184	158	1094	314
P8	616	562	201	174	601	325
P9	925	522	191	154	184	163
P10	565	539	219	172	179	295
P11	462	512	182	164	192	250
P12	616	578	204	180	852	861
P13	572	718	244	207	774	278
P14	472	510	179	161	170	187
P15	561	549	182	164	174	179

Table 5.5: CPU time (seconds) for each problem using the six different solvers.

Problem	LOQO	IPOPT	IPOPT	IPOPT	KNITRO	KNITRO
	6.0.6	3.8.1	3.12.8	3.9.3	10.1.0 alg 1	10.1.0 alg 2
	(server 1)	(server 1)	(server 2)	(server 2)	(server 2)	(server 2)
P1	0.3	0.4	0.9	0.8	1.2	0.5
P2	0.3	0.3	0.6	0.5	0.4	0.3
P3	0.3	0.3	0.7	0.6	0.6	0.3
P4	0.3	0.4	0.7	0.7	0.8	0.3
P5	0.3	0.3	0.6	0.6	0.6	0.3
P6	0.1	0.1	0.3	0.3	0.3	0.2
P7	0.1	0.1	0.2	0.3	0.3	0.2
P8	0.1	0.1	0.3	0.3	0.3	0.2
P9	0.1	0.1	0.3	0.3	0.3	0.2
P10	0.1	0.1	0.2	0.3	0.3	0.1
P11	0.1	0.1	0.3	0.3	0.3	0.1
P12	0.1	0.1	0.3	0.3	0.3	0.2
P13	0.1	0.1	0.3	0.3	0.3	0.2
P14	0.1	0.1	0.3	0.3	0.3	0.1
P15	0.1	0.1	0.3	0.3	0.3	0.1

Table 5.6: Velocity (iterations/seconds) for each problem using the six different solvers.

function is the calculation of the astigmatism (medium velocity); and finally in P6-P15 the objective function uses the calculation of the power and/or astigmatism gradients (very slow). Comparing the velocity for the spherical coordinate model versus the Cartesian coordinate model, we note that each iteration is much slower in spherical than in Cartesian coordinates. For example, for LOQO, the velocity in Cartesian coordinates is between 0.4 and 1.9 (see Table 6.13 in the Appendix, page 113), while in spherical coordinates the velocity is between 0.1 and 0.3 (see the first column of Table 5.6). That means that each iteration in spherical coordinates takes between 4 and 6 more time than for Cartesian coordinates (0.4/0.1=4, 1.9/0.3=6.3)

In order to evaluate the solutions, not only in terms of the optical properties of the lens produced, but also in terms of optimization, we checked the objective functions and constraints at optimal points. The objective functions are shown in Table 5.7. We see that, in general, KNITRO with algorithm 2 (conjugate gradient) provides the largest objectives (which is not surprising, since the conjugate gradient is meant to approximately solve Newton's equations); the lowest objectives are provided by LOQO and KNITRO with algorithm 1 (direct solver); while the objectives for IPOPT were in between. The last column of Table 5.7 reports for each problem the difference between the minimum and the maximum objective functions obtained, divided by the objective function of LOQO (taken as a baseline). The largest of these ratios was 0.73 for P5 (and P3). Figure 5.28 shows the maps of power and astigmatism for P5 using LOQO; KNITRO with direct algorithm 1; and KNITRO with conjugate gradient algorithm 2. Observe that these three lenses are quite the same in terms of optics.

However, the lens obtained using KNITRO with conjugate gradient algorithm 2 is a little bit different around the point ($x = 22.5\ mm$, $y = 5\ mm$), and it is also the lens with a larger objective function. Note that the differences in the values of the objective functions are not significant in terms of the lens' optical properties. The difference in the isolines around the point ($x = 22.5\ mm$, $y = 5\ mm$) does not affect the quality of the lens. We can conclude that the solutions using any of the six different solvers and spherical coordinates are the same in terms of optics.

Recall that in the Cartesian coordinate model not all the solutions reported as optimal by all the solvers were equivalent. The largest relative error of the objective function with respect to LOQO in spherical coordinates was 0.73, while in Cartesian coordinates was 2.17 (using solver KNITRO with conjugate gradient algorithm 2, for family F11 and type T2). A relative error of 2.17 is enormous because it means that one objective function—KNITRO with conjugate gradient algorithm 2—is more than three times the other objective function—using LOQO.

To check the (primal feasibility of the) constraints, we will focus again on the astigmatism and power maps, for instance P5 in Figure 5.28. Note that the power and the astigmatism in the far, corridor, and near regions, have all the required values. In addition, the maximum and minimum of the whole lens for the astigmatism and power are also in accordance with constraints (4.27)–(4.29). The astigmatism for all the points is smaller than $1.2 \cdot 2.0 = 2.4D$, and the power is between $5D$ and $7D$. Similar behavior was observed for the remaining problems (astigmatism and power maps for these are not reported here, in order to save space).

The values of the objective function (4.26) reported in Table 5.7 are related to the optical quality properties of the solutions. The optical quality depends on the separation of the astigmatism isolines: the larger the isoline separation, the lower the change in astigmatism and the better the lens. For instance, of the two different solutions for

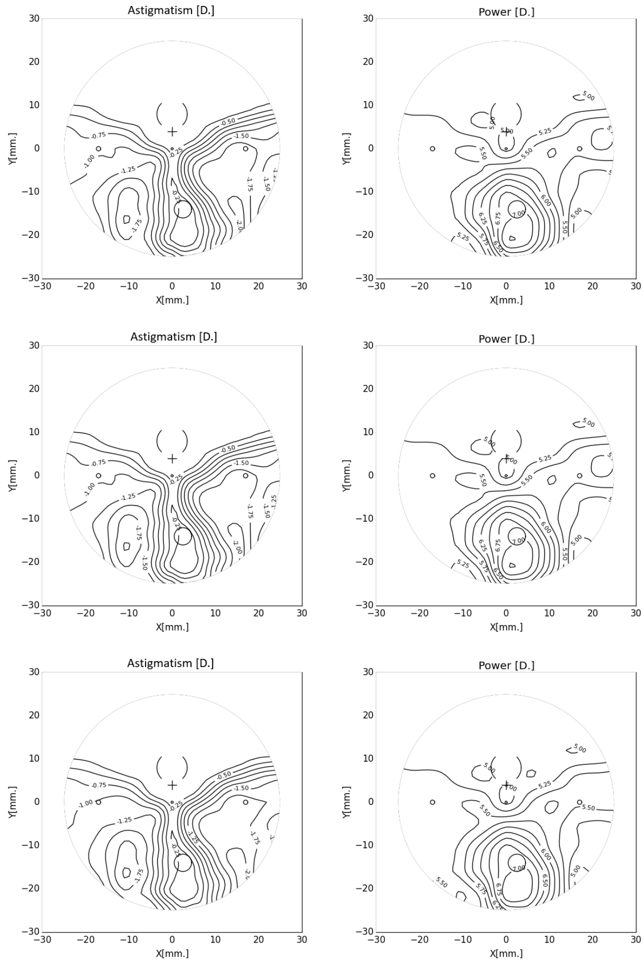


Figure 5.28: Astigmatism (left) and power (right) of the lens of problem P5 using LOQO 6.0.6 (top), KNITRO with direct algorithm 1 (middle) and KNITRO with conjugate gradient algorithm 2 (bottom).

Prob- lem	LOQO 6.0.6	IPOPT 3.8.1	IPOPT 3.12.8	IPOPT 3.9.3	KNITRO 10.1.0 alg 1	KNITRO 10.1.0 alg 2	Max. diff. (relative LOQO)
P1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.00
P2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.00
P3	0.2575	0.2824	0.2824	0.2824	0.2566	0.4455	0.73
P4	0.8872	0.9113	0.9113	0.9113	0.8859	1.0057	0.14
P5	0.2575	0.2824	0.2824	0.2824	0.2566	0.4455	0.73
P6	0.9106	0.9234	0.9234	0.9234	0.9100	1.1671	0.28
P7	0.4348	0.4494	0.4494	0.4494	0.4354	0.6841	0.57
P8	0.9106	0.9234	0.9234	0.9234	0.9100	1.1671	0.28
P9	15.5478	15.5529	15.5529	15.5529	15.5434	16.1208	0.04
P10	11.8692	11.8786	11.8786	11.8786	11.8733	11.8832	0.00
P11	12.1851	12.1957	12.1957	12.1957	12.1815	12.1842	0.00
P12	1.6272	1.6470	1.6470	1.6470	1.6261	1.6304	0.01
P13	2.9763	2.9892	2.9892	2.9892	2.9739	3.3132	0.11
P14	16.9738	16.9763	16.9763	16.9763	16.9709	16.9802	0.00
P15	13.9853	13.9923	13.9923	13.9923	13.9826	14.2659	0.02

Table 5.7: Objective function for each problem using the six different solvers.

problem P12 shown in Figure 5.25, the right one is better, since isolines are more separated. However, such a control on the separation of isolines is already implicitly included in the model: the second term in the objective function (4.26) (the one weighted by w_2) attempts to minimize the astigmatism gradient (similarly, the third term minimizes the power gradient), favoring lens with separated isolines.

We may finally remark that using identical stopping conditions (opttol= 10^{-3}), did not produce the same results when using KNITRO with direct algorithm 1 and conjugate gradient algorithm 2: it performed faster but worse with algorithm 2 than with algorithm 1. Again, this can be explained by the Newton direction being computed approximately by a conjugate gradient at each interior point iteration. Using KNITRO with the active set algorithm (algorithm 3) and the SQP algorithm (algorithm 4) the problems did not converge. We also, unsuccessfully, tried other active set codes, such as MINOS (which implements a projected Lagrangian method for nonlinearly constrained problems): in 10 million iterations (3 hours of CPU time) on server 2, MINOS did not converge, and the surface obtained—although it looked like a progressive lens—was very rough.

In summary, we were able to solve 15 different instances with six different solvers when using the model in spherical coordinates. In all cases, we obtained high-quality progressive lenses. As stated above, all of the lenses obtained were equivalent for the six solvers.

Recall that this did not happen in Cartesian coordinates. Note that not all the problem instances converged for the 6 solvers in Cartesian coordinates and some of the solutions reported as optimal for different solvers were different (the power and astigmatism maps differed).

5.6 Using B-splines of four–, five– and six–degree

At the end of Section 3.7 (page 17) we stated that using B-splines of a degree greater than 3 might be better than using three-degree B-splines due to a major degree of differentiability. In this section we will use B-splines of three–, four–, five– and six–degree. In theory, higher degree B-splines would be better, but in practice we will see in this section that this is not the case.

In this section the optimization model defined in Section 5.4 of this chapter (page 75) is used by replacing the definition of the radius of the surface (5.35) with:

$$R(\theta_i, \varphi_j) = \sum_{i'=1}^{o}\sum_{j'=1}^{o} c(\theta'_{i'}, \varphi'_{j'}) B^4_{i'}(\theta_i) B^4_{j'}(\varphi_j), \quad (i,j) \in \mathcal{G}, \tag{5.42}$$

$$R(\theta_i, \varphi_j) = \sum_{i'=1}^{o}\sum_{j'=1}^{o} c(\theta'_{i'}, \varphi'_{j'}) B^5_{i'}(\theta_i) B^5_{j'}(\varphi_j), \quad (i,j) \in \mathcal{G}, \tag{5.43}$$

or

$$R(\theta_i, \varphi_j) = \sum_{i'=1}^{o}\sum_{j'=1}^{o} c(\theta'_{i'}, \varphi'_{j'}) B^6_{i'}(\theta_i) B^6_{j'}(\varphi_j), \quad (i,j) \in \mathcal{G}, \tag{5.44}$$

where $B^3_{i'}(\theta_i)$, $B^3_{j'}(\varphi_j)$, $B^4_{i'}(\theta_i)$, $B^4_{j'}(\varphi_j)$, $B^5_{i'}(\theta_i)$, $B^5_{j'}(\varphi_j)$, $B^5_{i'}(\theta_i)$, $B^6_{j'}(\varphi_j)$, $(i,j) \in \mathcal{G}$, $(i',j') \in \mathcal{G}'$, are the 1-dimensional four-degree (5.42), five-degree (5.43) and six-degree (5.44) B-splines basis defined in [44, page 100] using a recursive formula. That means that the B-splines defined in (3.22)– (3.29) (page 16) have been increased up to six-degree. In this section the three-degree B-splines defined at (5.35) (page 76) and used in previous sections will also be used.

Once the optimization model is defined, the 15 problems P1–P15 defined in Section 5.5.1 (page 77) are solved using B-splines of degree 3, 4, 5 and 6. In this section, the solver IPOPT 3.8.1 is used on server 1 with the stopping criteria 10^{-2} that was also used in Section 5.5.4 (page 84). Table 5.8 reports the convergence of problems P1–P15 using four-degree B-splines, and only 5 out of 15 problems converged. However, using three–, five– and six-degree B-splines, all the 15 problems converged. Since 5-degree B-splines are calculated using 4-degree B-splines, we discarded a problem of implementation in B-splines of 4-degree (as it could be imagined).

Table 5.9 shows the number of iterations required for all 15 instances considering three-, four–, five– and six-degree B-splines. The last row shows the mean of the number of iterations for the 15 problems and each degree. Considering only the number of iterations, any of the degrees except 4-degree would be more or less equivalent (the mean of the number of iterations is 1.8 times for six-degree compared to three-degree but very similar for four– and five–degree -1.3 and 1.1 the number of iterations of three-degree-).

However, considering the CPU time, the three-degree B-splines are much faster than the higher degree B-splines. Table 5.10 reports the number of seconds for each problem and the degree required to achieve the optimum. The four-degree model was on average 2.5 times slower than the three-degree model; the five-degree model was on average 3.7 times slower than the three-degree model and finally the six-degree model was on average 14.1 times slower than the three-degree model.

Table 5.11 reports the values of the primal objective functions at the optimal solution for each problem and each degree. As the calculation of the objective function is computed

Problem	4-degree
P1	Optimal
P2	Iterations limit (1.000)
P3	Iterations limit (1.000)
P4	Local infeasibility
P5	Iterations limit (1.000)
P6	Optimal
P7	Optimal
P8	Optimal
P9	Iterations limit (1.000)
P10	Optimal
P11	Iterations limit (1.000)
P12	Iterations limit (1.000)
P13	Local infeasibility
P14	Iterations limit (1.000)
P15	Iterations limit (1.000)

Table 5.8: Convergence for each problem with IPOPT 3.8.1 and four–degree B-splines.

Problem	3-degree	4-degree	5-degree	6-degree
P1	80	72	69	87
P2	39	–	42	73
P3	44	–	50	78
P4	56	–	56	78
P5	44	–	50	78
P6	51	74	68	96
P7	46	45	56	93
P8	51	74	68	96
P9	49	–	47	95
P10	54	80	46	107
P11	51	–	46	99
P12	52	–	74	83
P13	66	–	79	123
P14	50	–	48	93
P15	51	–	48	96
mean	52.3	69.0	56.5	91.7

Table 5.9: Number of iterations for each problem with IPOPT 3.8.1 and B-splines of different degrees.

Problem	3-degree	4-degree	5-degree	6-degree
P1	211	327	545	1272.84
P2	122	–	510	1233.53
P3	134	–	572	1465.81
P4	158	–	589	1086.51
P5	135	–	447	1080.46
P6	562	1341	2222	8532.65
P7	512	973	1931	8182.59
P8	562	1313	2246	8734.84
P9	522	–	1665	7719.15
P10	539	1436	1669	8763.61
P11	512	–	1661	8424.46
P12	578	–	2758	7250.05
P13	718	–	2934	9555.73
P14	510	–	2013	7713.33
P15	549	–	1697	7940.07
mean	421.6	1078.0	1563.9	5930.4

Table 5.10: Number of seconds for each problem with IPOPT 3.8.1 and B-splines of different degrees.

Problem	3-degree	4-degree	5-degree	6-degree
P1	1.00	1.00	1.00	1.00
P2	1.00	–	1.00	1.00
P3	0.28	–	0.25	0.24
P4	0.91	–	0.91	0.83
P5	0.28	–	0.25	0.24
P6	0.92	7.23	0.85	0.55
P7	0.45	0.40	0.53	0.25
P8	0.92	7.23	0.85	0.55
P9	15.55	–	21.84	11.30
P10	11.88	53.30	16.74	11.63
P11	12.20	–	16.94	13.04
P12	1.65	–	1.53	0.86
P13	2.99	–	3.75	2.42
P14	16.98	–	23.10	12.59
P15	13.99	–	18.56	14.88

Table 5.11: Primal objective value for each problem with IPOPT 3.8.1 and B-splines of different degrees.

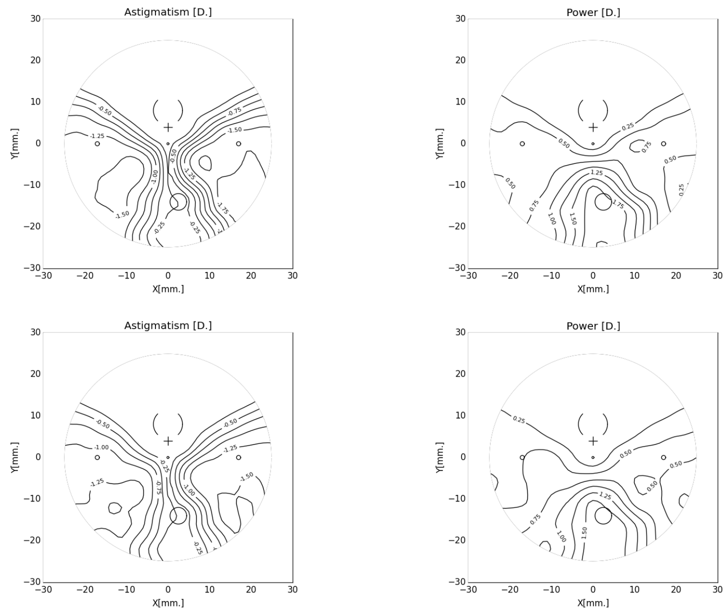


Figure 5.29: Astigmatism (left) and power (right) of the lens of problem P6 using IPOPT and B-splines of three-degree (top) and five-degree (bottom).

using B-splines of different degrees, this value mixes the quality of the solution with the precision of the calculation of the solution for each model. I.e., having a smaller value does not necessarily means having a better solution, because the objective function has been calculated using a different model. It would be interesting to calculate the objective function of the solutions for each problem and three-degree B-spline using the six-degree B-spline model. This can be done as a future task.

The number of iterations, CPU time (number of seconds), and objective function for all problems P1–15, using three-degree B-splines of this section are exactly the same as the results from Section 5.5.4 (page 84) using IPOPT 3.8.1 on server 1. This because the same model, solver and server have been used.

Figures 5.29 and 5.30 show the lenses obtained for problem P6 using B-splines of different degrees. It is appreciated that the three-degree and five-degree power and astigmatism maps are very similar (Figure 5.29). However, the maps using four-degree and six-degree are very rough (Figure 5.30), specially for the four-degree maps. In terms of optics, the quality of the solutions using four-degree and six-degree is not acceptable. Solutions with maps derived from three-degree and five-degree B-splines are very similar but not identical (see, for example, the map of power around the point ($x = 20\ mm, y = -10\ mm$)). In terms of optics, three-degree and five-degree solutions are of equal quality and produce high-quality lenses.

The quality of the lenses obtained using odd degree numbers (three– and five–degree) is much better than the quality obtained using even degree numbers (four– and six–degree). [43] explains that the odd-degree and the even-degree B-splines produce different results, which we have also oberved here.

In order to increase the lens quality obtained using six-degree B-splines, we also ran the P6 problem using tighter tolerances:

- LOQO 6.0.6: sigfig= 4, inftol= 10^{-6}.

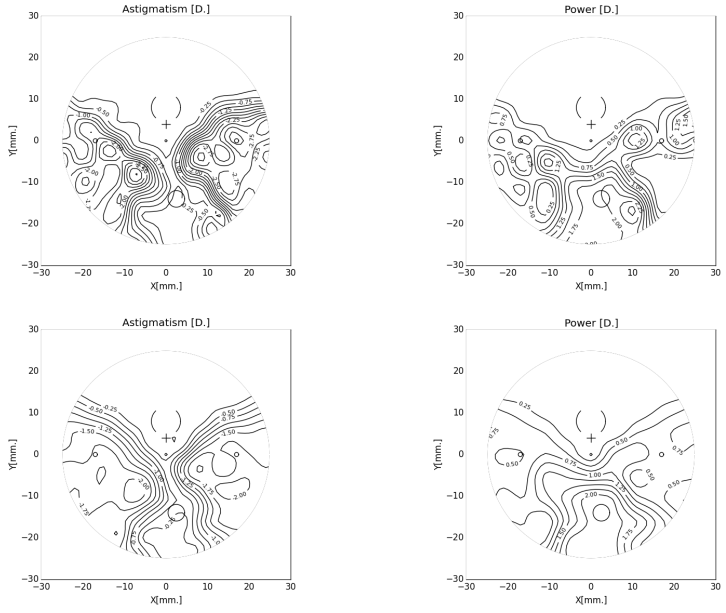


Figure 5.30: Astigmatism (left) and power (right) of the lens of problem P6 using IPOPT and B-splines of four-degree (top) and six-degree (bottom).

Solver	Stopping condition	CPU time (seconds)	Number of iterations	Primal objective values
LOQO	$inftol = 1.0e-6$	215	16860	0.548436563
IPOPT	$tol = 10e-6$	173	12735	0.539232799

Table 5.12: CPU time (seconds) —on server 1—, number of iterations and primal objective value using strict tolerances for problem P6 with six-degree B-splines.

- IPOPT 3.8.1, server 1, tol= 10^{-4}.

However, using these strict tolerances, the lenses we obtained had not improved and the number of iterations and seconds increased significantly, as can be seen in Table 5.12.

Figure 5.31 shows the lenses obtained using the tighter stopping conditions for problem P6 and using the six-degree B-spline. It is appreciated that none of these figures is of a high enough quality because all of them are very rough. In addition, the times required to obtain the optimum is very high, as can be seen in Table 5.12.

Increasing the tolerance of B-splines of four and six-degree did not enhanced the solution. The number of iterations has increased a lot but the solution has not improved.

Note that patent [13] uses splines of five-degree with control points spaced 10 mm or 20 mm, and the results seem to be reasonable. However, for our problem, using B-splines of more than three-degrees does not offer any benefit. In addition, the control points of our problem are spaced 4 mm in the case of Cartesian coordinates and 0.0196 radians in the case of spherical coordinates.

To sum up, considering the quality of the solutions and CPU time, the three-degree is the best choice. Lenses obtained using five-degree B-splines have the same optical quality as lenses obtained using three-degree B-splines, but solving the model with five-degree B-splines was on average 3.7 times slower than the three-degree model. For all of

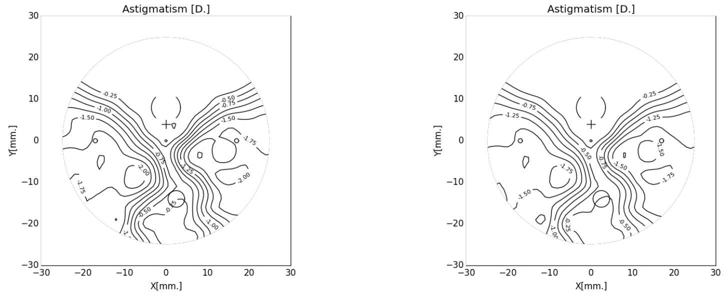


Figure 5.31: Astigmatism of the lens of problem P6 using six-degree B-splines and LOQO $inftol = 1.0e - 6$ (left) and IPOPT $tol = 10e - 6$ (left).

these reasons, the three-degree B-splines model is the best model and is the model used throughout this thesis.

According to [43], placing the knots and nodes in a different way for even degree B-splines might avoid poor quality results. A further task would be to apply the *shifting method* described in [43] to our spherical even degree B-splines, in order to see whether it makes them work correctly.

5.7 Pros and cons of Cartesian and spherical coordinates

Comparing the spherical and Cartesian coordinate models is not an easy task. Theoretically, the spherical model improves convexity, but in practice it requires an extra effort to express the parameters in spherical coordinates. It is not easy to have an identical optimization problem in Cartesian and in spherical coordinates, due to the issue of expressing the far, near and astigmatism regions in angles instead of millimeters. However, as a rule, the spherical coordinate model requires fewer iterations to converge, but each iteration is slower, while the Cartesian coordinate model requires more iterations and each iteration is faster. Depending on the stopping conditions, CPU time and the number of iterations of both models may vary significantly, and thus they are difficult to compare. All of the problems converged with all of the solvers using spherical coordinates, which was not the case for Cartesian coordinates: some problems did not converge in fewer than 3000 iterations. In addition, the solutions obtained using spherical coordinates and different solvers are more similar (the relative error is smaller) than using Cartesian coordinates. We may conclude that the spherical coordinate model is slower in CPU time, but that the results obtained are better.

Both models were implemented using the AMPL modeling language and were solved using interior point methods with solvers LOQO, IPOPT, and KNITRO. Concerning the KNITRO solver, two variants of interior point algorithm were used: one that uses a direct method to solve Newton's equations and another that uses an iterative conjugate gradient method. IPOPT and LOQO use a direct method. In this thesis we solved a set of 72 problem instances using Cartesian coordinates and a set of 15 problem instances using spherical coordinates. As stated above, we did not find a method that achieved equivalence between both methods, and expressing the parameters in spherical coordinates requires an extra effort. However, we will compare some results in terms of the convergence of these two models.

The **Cartesian coordinates** model has 900 variables and between 6000 and 8000 constraints. The problem is solved using the different interior point solvers described above. LOQO solver is installed on a server with eight 2.7GHz AMD Opteron 8384 Shanghay CPUs, with 32 cores and 128GB RAM. Using LOQO solver on this server, all of the problem instances converged with fewer than or equal to 192 iterations within 289 seconds.

Note that in previous work by the same author [47], the Cartesian coordinate model using LOQO on the same server took between 10 and 32 minutes to converge and between 248 and 1098 iterations, and only 5 cases out 9 converged using the same LOQO solver.

In the work of this thesis, some of the proposed instances did not converge using Cartesian coordinates linked to KNITRO with the iterative conjugate gradient method and linked to IPOPT solver, In addition, the relative error of the objective function with respect to LOQO reached up to 2.17, which is enormous. The maximum relative error should be about 0.30, thus some of the problem instances considered to be optimal in fact produced some solutions that were of insufficient quality. The quality of the solution, is one of the drawbacks when using Cartesian coordinates. The solvers that produced the highest-quality lenses were LOQO and KNITRO with the direct method.

On the other hand, the **spherical coordinates** model has a similar number of variables (991), but has between 15000 and 18100 constraints. Some constraints have been added in comparison with the Cartesian coordinate model in order to help the convergence. All of the problems converged using all of the interior point proposed solvers (which was not the case for the Cartesian coordinates) and the maximum relative error (with respect to LOQO) was 0.73 (instead of 2.17 for Cartesian coordinates). Thus, the convergence of the problem has been improved considerably. Again, the best solvers were LOQO and KNITRO with the direct method, but all solvers converged using the spherical coordinate model for all of the problem instances proposed (which was not the case for Cartesian coordinates).

Using the LOQO solver on the same machine, all of the problem instances converged with fewer than or equal to 84 iterations within 925 seconds. That means that the number of iterations with respect to Cartesian coordinates decreased (for example from a maximum of 192 iterations—Cartesian—to 84—spherical—), but the number of seconds increased (from a maximum of 289 seconds—Cartesian—to 925 seconds—spherical—). The reason for this is that the convexity of the problem has improved, but spherical coordinates required large calculations for each iteration, which increased the CPU time. The CPU time for each iteration and the total CPU time increased with respect to Cartesian coordinates.

We thus conclude that the spherical coordinates solved all of the problem instances and produced higher quality lenses than the Cartesian coordinates (the relative error between solvers is much lower), the convexity is better (although the problem is still nonconvex), the number of iterations is lower but the required CPU to converge the problem has increased with respect to Cartesian coordinates.

Given the complexity of both models, it is difficult to obtain theoretical results for a general setting, other than the small example used in the Proposition 1. The objective functions of both the Cartesian and the new spherical models are nonconvex. A (yet open) problem would be to study the convexity of the objective function and the feasible set around the optimal solution. However, this would require and analytical computation and study of a very large Hessian and a multidimensional set in a neighborhood of the optimal point, which has not yet been done.

In addition, we observed that in all instances tested using spherical coordinates, the different solvers always converged to the same solution. This indicates that, near the optimum, the problem (objective function and feasible set) is convex; but even more important, it raises the question of whether this local minimum is indeed a global one. A definitive answer to those questions is part of the further work research to be done in this field.

Chapter 6

Conclusions and contributions

This thesis develops two different methods for designing progressive lenses using interior point solvers. Progressive lenses are used to correct presbyopia and have a complex design. The upper part of the progressive lens is used for far vision, the lower part is used for near vision, and the middle part is used for intermediate distances. This thesis only considers a side of the lens, that which is farthest from the eye. The upper part of this surface has less power than the bottom zone. Unfortunately, modifying the power across the surface lens vertically creates lateral aberrations in form of astigmatism. The main properties of a progressive lens are its power and astigmatism, which are calculated using the principal curvatures of the surface, i.e, using the second partial derivatives of the surface. The power is the product of the mean principal curvatures of the lens surface multiplied by a constant; and the astigmatism is the product of the principal curvatures difference multiplied by the same constant. The optimization models proposed in this thesis use these definitions of power and astigmatism in order to design progressive lenses.

The first model is based on Cartesian coordinates, and improves a previous model described in [47]. The second model uses spherical coordinates and exhibits better convexity properties than previous models. When spherical coordinates are used in ophthalmic optics, the center of the coordinates is referenced at the center of rotation of the eye. However, the spherical coordinate model proposed in this thesis is entirely unrelated to this model. The far zone of a progressive lens can be approximated by a sphere because the power of the lens in this region and, consequently, its curvature is nearly constant. The center of this sphere will be the center of the new proposed model using spherical coordinates. If the center of the spherical coordinates were not placed as explained, the model would not exhibit better convexity properties.

The new Cartesian and spherical coordinate models have been implemented using the AMPL modeling language and solved using interior point solvers. Their implementation has allowed us to successfully compute real-world progressive lenses. The optics of these progressive lenses are of high-quality; not only are they similar to the progressive lenses obtained using previously existing Cartesian coordinate models, but they also are of the same quality as other progressive lenses that are sold on the market. However, we would like to point out that the design of these lenses differs: the corridors of the lenses obtained using spherical coordinates are usually slightly longer, and either the far or the near region may be larger or smaller than those of lenses obtained using Cartesian coordinates. All of these lenses are accepted on the progressive lens market. One limitation worth commenting on is that we were unsuccessful in obtaining equivalence between the two sets of data required for calculating the lenses using Cartesian coordinates and spherical coordinates. Such equivalence is possible only if the lenses are solved in both coordinate systems. This is because the parameters in Cartesian coordinates are the coordinates $(x,\ y)$ of the different zones (far, near, corridor and astigmatism zones), and the same parameters in spherical coordinates are the $(\theta,\ \varphi)$. Converting these points from Cartesian to spherical coordinates requires the value of the radius to each point or the value of the coordinate z for each point, which are the solution of the model in spherical and Cartesian coordinates, respectively.

For both models, we compared different interior point solvers. Both models were solved using the interior points solvers LOQO, IPOPT, and two variants of KNITRO: one that solves the Newton's equation using a direct method (factorization) and the other through an iterative conjugate gradient. Using Cartesian coordinates, the LOQO solver and KNITRO with the direct method worked correctly in all cases in fewer than 3000 iterations, but some instances with IPOPT (3 instances out of 72) and with KNITRO using the iterative conjugate gradient method (13 instances out of 72) did not converge. Using spherical coordinates, all of the proposed solvers converged in all instances.

The new Cartesian coordinate model greatly reduced the CPU time and the number of iterations with respect to the previous Cartesian model [47]. However, the main drawback of the Cartesian coordinate model is that not all the problems converged using all the proposed interior point solvers (with fewer than 3000 iterations) and that the relative error was quite large. In the previous Cartesian coordinate model, the percentage of problems that did not converge was higher.

The main advantage of using spherical coordinates is that the problem exhibits better convexity properties. Improved convexity properties can be proved theoretically on Proposition 1 (page 74), and in three aspects in practice: the number of iterations and the relative error decreased (with respect to Cartesian coordinates), and all the problems converged (which was not the case for the Cartesian coordinate model). However, as each iteration took a lot of time, the CPU time is higher overall for the spherical than for the Cartesian model.

The Cartesian coordinate model allows us to orient the power and astigmatism gradients, thus producing a new particular type of progressive lenses. Theses lenses are used when personalizing lenses for each user. For example when considering the user's eye movements or the axis of the cylinder of the prescription. This result was patented during the research work for this thesis (see [46]).

The main contributions of this research are: a Cartesian coordinate model that improves the model presented by [47]; the ability to orient power and astigmatism gradients (patented in [46]); a new spherical coordinate model published in the article [45] and a comparison of different interior point solvers (given in Sections 4.6.5 and 5.5.4). Several conferences and other publications are reported in Section 2.3 (page 10).

However, there are two aspects of this thesis that could be ameliorated. The first is to improve the third-degree B-spline basis used to represent the surface. Considering that calculating power and astigmatism uses the second partial derivatives of the surface, the degree of the B-splines should be increased. However, five-degree B-splines did not improve the quality of the solutions, and even degrees did not produce correct progressive lenses. The *shifting method* described in [43] might improve the Cartesian and the spherical coordinate models using even degree B-splines.

The second improvement involves studying the relationship between the data sets used in the two models of Cartesian and spherical coordinates. As previously stated, we did not succeed in obtaining the relationship of the parameters in both models without solving one of the models beforehand. One improvement would be to approximate the radius of the different zones using one or two precalculated radii. Enhancing this relationship would improve the quality of the instance generator.

Bibliography

Optimization

[1] R. E. Bank, P. E. Gill, and R. F. Marcia. Interior methods for a class of elliptic variational inequalities. Large-Scale PDE-Constrained Optimization. *Lecture Notes in Computational Science and Engineering*, 30, 218–235, 2003.

[2] R. Byrd, M. E. Hribar, J. Nocedal. An Interior Point Algorigthm for Large Scale Nonlinear Programming. *SIAM J. Optimization*, 9, 877–900, 1999.

[3] R.H. Byrd, J. Nocedal, R.A. Waltz. Feasible interior methods using slacks for nonlinear optimization, *Computational Optimization and Applications*, 26, 35–61, 2003.

[4] R. Byrd, J. Nocedal, R. Waltz. Knitro: An integrated package for nonlinear optimization, Large-Scale Nonlinear Optimization, *Nonconvex Optimization and Its Applications*, 83, 35–59, 2006.

[5] R. Fourer, D.M. Gay, D.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* Duxbury Press, Pacific Grove CA, USA, 2002.

[6] D. G. Luenberger, Y. Ye. *Linear and NonLinear Programming.* Third Edition, Springer, 2008.

[7] J. Nocedal, S. J. Wright. *Numerical Optimization.* Second Edition, Springer, 2006.

[8] R.J. Vanderbei. LOQO user's manual–version 3.10, *Optimization Methods and Software*, 11, 485–514, 1999.

[9] R. Vanderbei. LOQO: an interior point code for quadratic programming. *Optimization Methods and Software*, 11, 451–484, 1999.

[10] R. Vanderbei, D. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Statistics and Operation Research, Computational Optimization and Applications*, 13, 1-3, 231–252, 1997.

[11] A. Wächter, L.T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming, *Mathematical Programming*, 106, 25–57, 2006.

Optics

[12] P.Artús, G.Casanellas (Innovation Department of Horizons Optical). Do the progressive lenses really satisfy the Minkwitz theorem? Strategies to go beyond the Minkwitz theorem. *MAFO, Ophthalmic Labs & Industry*, 15, 10–17, 2019.

[13] P. Allione, L. Calixte, C. Guilloux. Method and apparatus for designing an optical lens. Patent of invention, US2012010500-A2, Assignee: Essilor, 2012.

[14] J. M. Boix Palacián. *Lentes Progresivas. Evolución científica hasta la quinta generación.* Editorial Complutense, 2000.

[15] S. Chamadoira Hermida, G. Casanellas Peñalver, J. C. Dürsteler, J. Vegas Caballero, M. Espínola Estepa. Procedure for designing a progressive ophthalmic lens and corresponding lens. Patent of invention, US8770748-B2, Assignee: Indo Internacional S.A, 2014.

[16] K. Demaseure. *Étude de logiciels d'optimisation non-linéaire de grande taille, avec application et analyse comparative sur un exemple de modélisation de verres de lunettes progressifs.* DEA thesis, Université de Namur, Belgique, 2002.

[17] J.C. Dursteler. *Computer Assisted Design Systems for Progressive Lenses* (in Spanish). PhD thesis, Universitat Politècnica de Catalunya, 1991.

[18] , J. C. Dursteler, J. Vegas, R. Villuela, E. Fontdecaba, J. Palomar. Method for calculating an ophthalmic lens and corresponding lens. Patent of invention, US7604350B2, Assignee: Indo Internacional S.A., 2009.

[19] E. Fontdecaba. *High Performance Algorithms for Progressive Addition Lens Design.* PhD thesis, Universitat Politècnica de Catalunya, 2000.

[20] A .Giménez, C. Prieto, J.C. Dürsteler, J.S. Solaz, B. Mateo, R.M. Porcar, Method and device for determining the visual behaviour of a person and method of customising a spectacle lens. Patent of invention, WO2005107576, Assignee: Indo Internacional S.A., 2005.

[21] W. Haimerl, P. Baumbach. State of the Art. Optimization of spectacle lenses PAL. *Bulgarian Journal of Physics*, 27, 7–12, 2000.

[22] X. Jonsson. *Méthodes de Points Intérieurs et de Régions de Confiance en Optimisation Non Linéaire. Application à la Conceptions de Verrers Ophtalmiques Progressifs.* PhD thesis, Université Paris 6, 2002.

[23] ISO 8980-1, *Ophthalmic optics, Uncut finished spectacle lenses, Part 1: Specifications for single-vision and multifocal lenses*, 2017.

[24] ISO 8980-2, *Ophthalmic optic, Uncut finished spectacle lenses, Part 2: Specifications for power-variation lenses*, 2017.

[25] J. Loos, G. Greiner, Hans-Peter Seidel. A variational approach to progressive lens design, *Computer-Aided Design*, 30, 595–602, 1998.

[26] D. Meister, Free-Form Surfacing Technology Makes Possible New Levels of Optical Sophistication for Spectacles, *Refractive Eyecare for Ophthalmologists*, 9, 1–4, 2005.

[27] D. Meister, S.W. Fisher. Progress in the spectacle correction of presbyopia. Part 1: Design and development of progressive lenses, *Clinical and Experimental Optometry*, 91, 240–250, 2008.

[28] D. Meister, S.W. Fisher. Progress in the spectacle correction of presbyopia. Part 2: Modern progressive lens technologies, *Clinical and Experimental Optometry*, 91, 251–264, 2008.

[29] G. Minkwitz, Uber den flachenastigmatismus bei gewissen symmetrischen aspharen [On the surface astigmatism of a fixed symmetrical aspheric surface], *Optica Acta: International Journal of Optics*, 10, 223–227, 1963.

[30] F. Santosa, Wang, R. Gulliver. Analysis of a Variational Approach to Progressive Lens Design, *SIAM Journal on Applied Mathematics*, 64, 277–296, 2003.

[31] G. Savio, G. Concheri, R. Meneghello. Parametric Modeling of Free-Form Surfaces for Progressive Addition Lens, *Proceedings of the IMProVe 2011, International conference on Innovative Methods in Product Design*, 2011.

[32] D. Tomanos. *Algorithms and Software for Multilevel Nonlinear Optimization.* PhD thesis, University of Namur (FUNDP), Namur, 2009.

[33] N. Gould, Ph. Toint. How Mature is Nonlinear Optimization?, in *Applied Mathematics Entering the 21st Century: Invited Talks from the ICIAM 2003 Congress*, (J. H. Hill and R. Moore, eds.) SIAM, Philadelphia, 141–161, 2004.

[34] Jing Wang, Fadil Santosa. A numerical method for progressive lens design, *Mathematical Models and Methods in Applied Sciences*, 14, 619–640, 2004.

[35] Wei Jiang, Bao Weizhu, Tang Qinglin, Wang Hanquan. A variational-difference numerical method for designing progressive-addition lenses. *Computer-Aided Design*, 48, 17–24, 2014.

[36] Fanhuan Zhou. Design of Progressive Additional Lens with Wavefront Tracing Method. PhD thesis, University of Minesota, 2010.

[37] Wei-Yao Hsu, Yen-Liang Liu, Yuan-Chieh Cheng, Ching-Hsiang Kuo, Chun-Chieh Chen, Guo-Dung Su. Design, fabrication, and metrology of ultra-precision optical freeform surface for progressive addition lens with B-spline description, *International Journal of Advanced Manufacturing Technology*, 63, 225–233, 2012.

[38] https://www.satisloh.com/emea-india/ophthalmic/generating/

[39] https://www.schneider-om.com/ophthalmics/generating.html

Other subjects (Differential geometry, splines and nurbs)

[40] M.P. de Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Inc., 1976, New Jersey, USA.

[41] J. P. Kaipio, E. Somersalo. *Statistical and Computational Inverse Problems*, Springer, 2005, New York.

[42] M. Lipschutz. *Differential Geometry.* Schaum's, McGraw-Hill, 1969, New York.

[43] H. Park. Choosing nodes and knots in closed B-spline curve interpolation to point data, *Computer-Aided Design*, 33, 967–974, 2001.

[44] L. Piegel, W. Tiller. *The Nurbs Book.* Springer, 1997, Berlin.

Work of the author during this thesis

[45] G. Casanellas, J. Castro, Using interior point solvers for optimizing progressive lens models with spherical coordinates, *Optimization and Engineering*, 2019. https://doi.org/10.1007/s11081-019-09480-z.

[46] G.Casanellas, P.Artús, T.Vilajona, Method for Optimising a Progressive Ophthalmic lens and Method for Producing Same. Patent of invention, ES2018070321, WO2018193147, Assignee: Horizons Optical S.L.U, 2017.

Work of the author previous to this thesis

[47] G. Casanellas. Disseny de lents progressives. Master thesis, Universitat Politècnica de Catalunya, 2014.

Appendix

Results of the optimization problems using Cartesian coordinates

Number of iterations for each family and each problem for six different solvers in Cartesian coordinates

Family/Type	T1	T2	T3	T4	T5	T6
F1	62	79	73	72	66	62
F2	69	67	64	71	146	64
F3	116	75	91	75	61	106
F4	83	82	89	85	110	75
F5	43	81	45	60	53	75
F6	55	47	49	42	56	57
F7	55	107	46	55	106	54
F8	71	103	55	45	59	48
F9	89	85	90	81	102	83
F10	89	85	90	81	94	79
F11	69	192	63	115	92	76
F12	54	62	47	90	153	47

Table 6.1: Number of iterations for each family and problem with LOQO 6.0.6 on “server 1”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	136	206	303	207	251	309
F2	123	333	254	420	374	273
F3	124	190	210	249	351	248
F4	254	231	544	276	365	329
F5	1269	436	808	178	168	2000
F6	698	186	698	180	273	169
F7	778	163	1195	208	215	190
F8	521	192	936	227	213	242
F9	72	247	380	443	227	249
F10	72	251	386	491	231	258
F11	406	262	242	2000	229	259
F12	197	201	540	163	158	182

Table 6.2: Number of iterations for each family and problem with IPOPT 3.8.1 on “server 1”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	210	206	269	207	332	272
F2	957	267	376	389	422	297
F3	264	190	271	247	426	296
F4	271	228	388	267	376	325
F5	465	486	685	551	537	2000
F6	2000	575	704	628	650	727
F7	697	731	856	523	527	541
F8	508	527	748	439	617	650
F9	329	251	306	352	225	414
F10	404	252	589	595	231	245
F11	313	252	388	2000	226	264
F12	180	198	151	159	165	195

Table 6.3: Number of iterations for each family and problem with IPOPT 3.9.3 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	210	206	262	207	307	253
F2	349	467	336	395	325	313
F3	337	233	242	270	206	257
F4	272	268	295	263	260	303
F5	482	501	727	568	547	2000
F6	2000	553	748	626	658	792
F7	674	744	968	913	888	924
F8	510	545	745	610	617	659
F9	331	263	503	320	226	341
F10	384	245	402	335	235	247
F11	340	252	325	2000	229	265
F12	179	197	151	164	160	184

Table 6.4: Number of iterations for each family and problem with IPOPT 3.12.8 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	52	293	358	97	50	50
F2	59	257	346	87	57	110
F3	57	269	326	127	55	54
F4	255	572	773	406	368	375
F5	433	843	500	562	1025	645
F6	760	799	1131	1011	1371	646
F7	313	376	791	455	447	398
F8	40	284	319	146	38	45
F9	40	284	319	146	38	45
F10	40	281	397	137	38	45
F11	41	240	264	73	27	32
F12	61	134	154	67	58	86

Table 6.5: Number of iterations for each family and problem with KNITRO 10.1.0 direct algorithm 1 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	859	701	2000	2000	1332	604
F2	746	260	1311	53	110	2000
F3	483	915	1008	46	91	920
F4	385	410	1103	1090	418	1250
F5	281	495	700	812	770	280
F6	739	1026	2000	1727	605	932
F7	772	671	699	696	758	834
F8	421	1222	385	448	513	970
F9	421	1222	385	448	513	970
F10	232	958	1214	48	211	2000
F11	2000	383	2000	883	2000	2000
F12	2000	148	158	2000	379	2000

Table 6.6: Number of iterations for each family and problem with KNITRO 10.1.0 conjugate gradient algorithm 2 on “server 2”.

CPU time for each family and each problem for six different solvers in Cartesian coordinates

Family/Type	T1	T2	T3	T4	T5	T6
F1	42	50	50	49	46	42
F2	44	44	42	45	75	42
F3	65	48	56	48	42	61
F4	51	51	55	52	63	48
F5	102	175	105	136	120	156
F6	78	70	70	65	78	78
F7	84	134	67	78	134	76
F8	143	198	116	99	121	104
F9	155	156	162	143	175	147
F10	150	151	158	133	158	135
F11	116	289	108	181	149	126
F12	95	109	87	149	252	87

Table 6.7: CPU time (seconds) for each family and problem with LOQO 6.0.6 on “server 1”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	76	104	150	103	118	149
F2	67	148	114	173	171	123
F3	67	88	102	110	154	114
F4	131	117	261	126	177	146
F5	2052	264	1342	272	254	3579
F6	707	211	734	197	362	195
F7	785	182	1218	225	243	210
F8	888	364	1510	404	382	438
F9	106	356	564	619	330	374
F10	103	351	532	724	325	359
F11	571	376	341	3224	323	363
F12	319	284	821	234	238	260

Table 6.8: CPU time (seconds) for each family and problem with IPOPT 3.8.1 on “server 1”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	45	39	49	38	57	48
F2	154	47	62	62	69	51
F3	46	36	47	43	82	51
F4	47	42	63	47	68	54
F5	345	306	440	364	330	1463
F6	726	224	320	239	276	335
F7	283	286	319	209	204	211
F8	316	346	498	275	379	471
F9	189	146	171	195	129	227
F10	221	143	323	380	132	136
F11	212	143	213	1145	130	151
F12	108	115	91	95	98	114

Table 6.9: CPU time (seconds) for each family and problem with IPOPT 3.9.3 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	37	36	45	37	50	42
F2	53	72	52	59	52	49
F3	52	39	40	45	36	42
F4	44	44	48	44	43	49
F5	301	318	493	361	358	1405
F6	755	206	273	233	283	288
F7	249	273	359	325	320	339
F8	322	347	461	387	383	405
F9	184	149	276	177	129	188
F10	207	136	217	183	133	138
F11	188	140	178	1369	129	147
F12	102	112	89	95	93	106

Table 6.10: CPU time (seconds) for each family and problem with IPOPT 3.12.8 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	19	45	55	24	18	20
F2	20	44	52	23	21	25
F3	21	43	51	28	21	19
F4	159	311	423	250	224	231
F5	170	286	186	216	366	240
F6	281	288	403	561	476	246
F7	190	229	448	376	270	239
F8	35	108	172	95	33	36
F9	34	108	172	94	34	37
F10	34	106	156	81	34	36
F11	36	146	138	73	81	48
F12	46	80	90	46	44	56

Table 6.11: CPU time (seconds) for each family and problem with KNITRO 10.1.0 direct algorithm 1 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	159	125	341	314	204	115
F2	151	68	225	24	31	320
F3	103	176	185	23	30	151
F4	337	250	629	767	294	686
F5	132	208	279	324	294	141
F6	382	475	833	777	336	443
F7	650	501	516	525	567	655
F8	180	470	240	178	299	562
F9	179	468	239	177	298	558
F10	107	376	713	41	136	1243
F11	1193	174	955	388	1121	1197
F12	1115	106	113	1085	232	1083

Table 6.12: CPU time (seconds) for each family and problem with KNITRO 10.1.0 conjugate gradient algorithm 2 on “server 2”.

Velocity (iterations/seconds) for each family and each problem for six different solvers in Cartesian coordinates

Family/Type	T1	T2	T3	T4	T5	T6
F1	1.5	1.6	1.5	1.5	1.4	1.5
F2	1.6	1.5	1.5	1.6	1.9	1.5
F3	1.8	1.6	1.6	1.6	1.5	1.7
F4	1.6	1.6	1.6	1.6	1.7	1.6
F5	0.4	0.5	0.4	0.4	0.4	0.5
F5	0.7	0.7	0.7	0.6	0.7	0.7
F7	0.7	0.8	0.7	0.7	0.8	0.7
F8	0.5	0.5	0.5	0.5	0.5	0.5
F9	0.6	0.5	0.6	0.6	0.6	0.6
F10	0.6	0.6	0.6	0.6	0.6	0.6
F11	0.6	0.7	0.6	0.6	0.6	0.6
F12	0.6	0.6	0.5	0.6	0.6	0.5

Table 6.13: Velocity (iterations/seconds) for each family and problem with LOQO 6.0.6 on “server 1”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	1.8	2.0	2.0	2.0	2.1	2.1
F2	1.8	2.2	2.2	2.4	2.2	2.2
F3	1.8	2.2	2.1	2.3	2.3	2.2
F4	1.9	2.0	2.1	2.2	2.1	2.3
F5	0.6	1.6	0.6	0.7	0.7	0.6
F5	1.0	0.9	1.0	0.9	0.8	0.9
F7	1.0	0.9	1.0	0.9	0.9	0.9
F8	0.6	0.5	0.6	0.6	0.6	0.6
F9	0.7	0.7	0.7	0.7	0.7	0.7
F10	0.7	0.7	0.7	0.7	0.7	0.7
F11	0.7	0.7	0.7	0.6	0.7	0.7
F12	0.6	0.7	0.7	0.7	0.7	0.7

Table 6.14: Velocity (iterations/seconds) for each family and problem with IPOPT 3.8.1 on “server 1”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	4.6	5.2	5.5	5.4	5.8	5.7
F2	6.2	5.7	6.1	6.2	6.1	5.8
F3	5.7	5.3	5.7	5.7	5.2	5.8
F4	5.8	5.4	6.1	5.7	5.5	6.0
F5	1.3	1.6	1.6	1.5	1.6	1.4
F5	2.8	2.6	2.2	2.6	2.4	2.2
F7	2.5	2.6	2.7	2.5	2.6	2.6
F8	1.6	1.5	1.5	1.6	1.6	1.4
F9	1.7	1.7	1.8	1.8	1.7	1.8
F10	1.8	1.8	1.8	1.6	1.8	1.8
F11	1.5	1.8	1.8	1.7	1.7	1.8
F12	1.7	1.7	1.7	1.7	1.7	1.7

Table 6.15: Velocity (iterations/seconds) for each family and problem with IPOPT 3.9.3 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	5.7	5.7	5.8	5.6	6.1	6.1
F2	6.5	6.5	6.5	6.7	6.2	6.4
F3	6.4	5.9	6.1	6.1	5.7	6.1
F4	6.2	6.1	6.2	5.9	6.1	6.2
F5	1.6	1.6	1.5	1.6	1.5	1.4
F5	2.7	2.7	2.7	2.7	2.3	2.7
F7	2.7	2.7	2.7	2.8	2.8	2.7
F8	1.6	1.6	1.6	1.6	1.6	1.6
F9	1.8	1.8	1.8	1.8	1.8	1.8
F10	1.9	1.8	1.9	1.8	1.8	1.8
F11	1.8	1.8	1.8	1.5	1.8	1.8
F12	1.7	1.8	1.7	1.7	1.7	1.7

Table 6.16: Velocity (iterations/seconds) for each family and problem with IPOPT 3.12.8 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	2.8	6.5	6.6	4.0	2.7	2.5
F2	3.0	5.8	6.6	3.8	2.7	4.4
F3	2.7	6.2	6.4	4.5	2.6	2.9
F4	1.6	1.8	1.8	1.6	1.6	1.6
F5	2.6	2.9	2.7	2.6	2.8	2.7
F5	2.7	2.8	2.8	1.8	2.9	2.6
F7	1.6	1.6	1.8	1.2	1.7	1.7
F8	1.2	2.6	1.9	1.5	1.2	1.2
F9	1.2	2.6	1.9	1.5	1.1	1.2
F10	1.2	2.7	2.5	1.7	1.1	1.3
F11	1.1	1.6	1.9	1.0	0.3	0.7
F12	1.3	1.7	1.7	1.4	1.3	1.5

Table 6.17: Velocity (iterations/seconds) for each family and problem with KNITRO 10.1.0 direct algorithm 1 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	5.4	5.6	5.9	6.4	6.5	5.2
F2	4.9	3.8	5.8	2.2	3.5	6.2
F3	4.7	5.2	5.4	2.0	3.0	6.1
F4	1.1	1.6	1.8	1.4	1.4	1.8
F5	2.1	2.4	2.5	2.5	2.6	2.0
F5	1.9	2.2	2.4	2.2	1.8	2.1
F7	1.2	1.3	1.4	1.3	1.3	1.3
F8	2.3	2.6	1.6	2.5	1.7	1.7
F9	2.4	2.6	1.6	2.5	1.7	1.7
F10	2.2	2.5	1.7	1.2	1.6	1.6
F11	1.7	2.2	2.1	2.3	1.8	1.7
F12	1.8	1.4	1.4	1.8	1.6	1.8

Table 6.18: Velocity (iterations/seconds) for each family and problem with KNITRO 10.1.0 conjugate gradient algorithm 2 on “server 2”.

Objective function for each family and each problem for six different solvers in Cartesian coordinates

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.2961	0.2921	0.3055	0.2772	0.2842	0.2968
F2	0.3017	0.3054	0.3077	0.2923	0.2914	0.3040
F3	0.6334	0.6239	0.6339	0.5656	0.6059	0.6073
F4	0.6764	0.6523	0.6725	0.6578	0.6368	0.6551
F5	9.6047	8.1493	8.9523	7.0738	8.3500	8.1019
F5	2.7204	2.6200	2.6076	3.2013	2.3803	2.6856
F7	8.5879	6.2211	7.7933	6.5990	5.6722	7.3264
F8	16.8198	17.0141	17.5570	15.0898	14.8430	17.3963
F9	0.0017	0.0017	0.0017	0.0015	0.0015	0.0016
F10	0.0017	0.0017	0.0017	0.0015	0.0015	0.0015
F11	0.0016	0.0015	0.0016	0.0014	0.0014	0.0015
F12	8.5024	8.1693	8.0566	7.6308	7.7516	8.2046

Table 6.19: Objective function for each family and problem with LOQO 6.0.6 on "server 1".

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.4183	0.3877	0.4041	0.3128	0.4030	0.4152
F2	0.3380	0.3694	0.3373	0.3275	0.4090	0.4028
F3	0.6631	1.1871	0.6361	0.6028	0.6231	0.6407
F4	1.0916	0.6884	0.6892	0.6710	0.6729	0.6890
F5	8.1843	8.0311	7.9502	6.9352	7.4485	†
F5	2.4116	2.6676	2.3079	2.1399	2.2390	2.6949
F7	6.1199	6.2388	5.9952	5.3535	5.5803	6.0930
F8	15.6667	15.3312	15.2042	13.2347	14.0493	15.0459
F9	0.0038	0.0037	0.0043	0.0033	0.0033	0.0039
F10	0.0038	0.0037	0.0042	0.0033	0.0033	0.0039
F11	0.0035	0.0034	0.0034	†	0.0030	0.0032
F12	8.2950	8.0403	7.8827	7.6177	7.6189	7.7667

† No optimal solution was found within the limit of 2000 iterations.

Table 6.20: Objective function for each family and problem with IPOPT 3.8.1 on "server 1".

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.3270	0.3877	0.4042	0.3128	0.3168	0.3956
F2	0.3380	0.3953	0.4083	0.3275	0.4085	0.4226
F3	0.6631	1.1871	0.6361	0.6028	0.6231	0.6406
F4	0.7117	0.6884	0.6892	1.1891	0.6728	1.2370
F5	8.1485	8.0153	7.9498	6.8675	7.3686	†
F5	†	2.2920	2.2515	2.0368	2.1188	2.2648
F7	6.0962	6.0559	5.9364	6.1949	6.4638	6.8465
F8	15.5619	15.3276	15.2044	13.9573	14.0486	15.0461
F9	0.0038	0.0036	0.0043	0.0033	0.0033	0.0039
F10	0.0038	0.0037	0.0043	0.0033	0.0033	0.0039
F11	0.0035	0.0034	0.0034	†	0.0030	0.0032
F12	8.2948	8.0403	7.8827	7.6178	7.6189	7.7666

† No optimal solution was found within the limit of 2000 iterations.

Table 6.21: Objective function for each family and problem with IPOPT 3.9.3 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.3270	0.3877	0.3272	0.3128	0.3168	0.3955
F2	0.3380	0.3301	0.4101	0.3275	0.3270	0.3343
F3	0.6648	0.6492	0.6361	0.6029	0.6231	0.6407
F4	0.7117	0.6884	0.6892	0.6710	0.6729	0.6890
F5	8.1485	8.0152	7.9497	6.8673	7.3685	†
F5	†	2.2920	2.2515	2.0376	2.1187	2.2648
F7	6.0963	6.0559	5.9363	5.3201	5.5498	5.9825
F8	15.5618	15.3293	15.2093	13.0445	14.0493	15.0459
F9	0.0038	0.0037	0.0043	0.0033	0.0033	0.0039
F10	0.0038	0.0036	0.0042	0.0033	0.0033	0.0039
F11	0.0035	0.0031	0.0034	†	0.0030	0.0032
F12	8.2950	8.0402	7.8827	7.6178	7.6189	7.7667

† No optimal solution was found within the limit of 2000 iterations.

Table 6.22: Objective function for each family and problem with IPOPT 3.12.8 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.2902	0.2863	0.2841	0.2755	0.2807	0.2865
F2	0.6259	0.6114	0.5976	0.5646	0.5854	0.5952
F3	0.6753	0.6511	0.6523	0.6340	0.6352	0.6520
F4	8.1278	8.0054	7.5849	7.2334	7.3651	7.6301
F5	2.3161	2.2683	2.2241	2.5632	2.0929	2.2383
F5	6.0816	6.0360	5.7099	5.2986	5.5286	5.8274
F7	15.5871	15.3910	14.5397	13.0798	14.0272	15.1060
F8	0.0019	0.0019	0.0016	0.0017	0.0018	0.0015
F9	0.0019	0.0019	0.0016	0.0017	0.0018	0.0015
F10	0.0019	0.0018	0.0016	0.0017	0.0018	0.0015
F11	0.0015	0.0015	0.0015	0.0014	0.0015	0.0015
F12	8.2709	8.0174	7.8606	7.5947	7.5962	7.5895

Table 6.23: Objective function for each family and problem with KNITRO 10.1.0 direct algorithm 1 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.3336	0.3287	†	†	0.3915	0.3294
F2	0.8343	0.8169	0.6484	0.7706	0.9795	†
F3	0.8872	0.7005	0.8225	0.8441	0.8448	0.7025
F4	8.1346	8.1228	7.9363	6.8543	7.3921	7.8683
F5	2.4486	2.4272	2.3795	2.0385	2.2571	2.3359
F5	6.0810	6.0395	†	5.3237	5.5564	5.9664
F7	15.5999	15.3448	15.2444	13.0861	14.0376	15.0965
F8	0.0051	0.0038	0.0051	0.0034	0.0045	0.0041
F9	0.0051	0.0038	0.0051	0.0034	0.0045	0.0041
F10	0.0052	0.0038	0.0043	0.0044	0.0045	†
F11	†	0.0048	†	0.0037	†	†
F12	†	8.6063	7.9859	†	8.1753	†

† No optimal solution was found within the limit of 2000 iterations.

Table 6.24: Objective function for each family and problem with KNITRO 10.1.0 conjugate gradient algorithm 2 on “server 2”.

Relative error of the objective function with respect to LOQO for each family and each problem for five different solvers in Cartesian coordinates

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.41	0.33	0.32	0.13	0.42	0.40
F2	0.12	0.21	0.10	0.12	0.40	0.33
F3	0.05	0.90	0.00	0.07	0.03	0.05
F4	0.61	0.06	0.02	0.02	0.06	0.05
F5	0.15	0.01	0.11	0.02	0.11	†
F5	0.11	0.02	0.11	0.33	0.06	0.00
F7	0.29	0.00	0.23	0.19	0.02	0.17
F8	0.07	0.10	0.13	0.12	0.05	0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.50	1.15	1.17	1.59
F11	1.16	1.21	1.18	†	1.09	1.20
F12	0.02	0.02	0.02	0.00	0.02	0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 6.25: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.8.1 on “server 1”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.10	0.33	0.32	0.13	0.11	0.33
F2	0.12	0.29	0.33	0.12	0.40	0.39
F3	0.05	0.90	0.00	0.07	0.03	0.05
F4	0.05	0.06	0.02	0.81	0.06	0.89
F5	0.15	0.02	0.11	0.03	0.12	†
F5	†	0.13	0.14	0.36	0.11	0.16
F7	0.29	0.03	0.24	0.06	0.14	0.07
F8	0.07	0.10	0.13	0.08	0.05	0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.55	1.15	1.17	1.59
F11	1.16	1.21	1.18	†	1.09	1.20
F12	0.02	0.02	0.02	0.00	0.02	0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 6.26: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.9.3 on “server 2”.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.10	0.33	0.07	0.13	0.11	0.33
F2	0.12	0.08	0.33	0.12	0.12	0.10
F3	0.05	0.04	0.00	0.07	0.03	0.05
F4	0.05	0.06	0.02	0.02	0.06	0.05
F5	0.15	0.02	0.11	0.03	0.12	†
F5	†	0.13	0.14	0.36	0.11	0.16
F7	0.29	0.03	0.24	0.19	0.02	0.18
F8	0.07	0.10	0.13	0.14	0.05	0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.50	1.15	1.17	1.59
F11	1.16	1.06	1.18	†	1.09	1.20
F12	0.02	0.02	0.02	0.00	0.02	0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 6.27: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.12.8 on "server 2".

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.02	0.02	0.07	0.01	0.01	0.03
F2	0.00	0.04	0.03	0.01	0.00	0.02
F3	0.01	0.02	0.06	0.00	0.03	0.02
F4	0.00	0.00	0.03	0.04	0.00	0.00
F5	0.15	0.02	0.15	0.02	0.12	0.06
F5	0.15	0.13	0.15	0.20	0.12	0.17
F7	0.29	0.03	0.27	0.20	0.03	0.20
F8	0.07	0.10	0.17	0.13	0.05	0.13
F9	0.07	0.16	0.09	0.14	0.18	0.02
F10	0.07	0.11	0.06	0.08	0.17	0.01
F11	0.05	0.01	0.05	0.00	0.06	0.02
F12	0.03	0.02	0.02	0.00	0.02	0.07

Table 6.28: Relative error of the objective function with respect to LOQO for each family and problem with KNITRO 10.1.0 direct algorithm 1 on "server 2".

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.13	0.13	†	†	0.38	0.11
F2	0.14	0.11	0.27	0.26	0.14	†
F3	0.32	0.31	0.02	0.36	0.62	†
F4	0.31	0.07	0.22	0.28	0.33	0.07
F5	0.15	0.00	0.11	0.03	0.11	0.03
F5	0.10	0.07	0.09	0.36	0.05	0.13
F7	0.29	0.03	†	0.19	0.02	0.19
F8	0.07	0.10	0.13	0.13	0.05	0.13
F9	1.96	1.31	2.03	1.24	1.94	1.58
F10	1.98	1.31	1.54	1.87	1.92	†
F11	†	2.17	†	1.61	†	†
F12	†	0.05	0.01	†	0.05	†

† No optimal solution was found within the limit of 2000 iterations.

Table 6.29: Relative error of the objective function with respect to LOQO for each family and problem with KNITRO 10.1.0 conjugate gradient algorithm 2 on "server 2".

Relative error of the objective function with respect to LOQO for each family and each problem for five different solvers in Cartesian coordinates without absolute value

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.41	0.33	0.32	0.13	0.42	0.40
F2	0.12	0.21	0.10	0.12	0.40	0.33
F3	0.05	0.90	0.00	0.07	0.03	0.05
F4	0.61	0.06	0.02	0.02	0.06	0.05
F5	-0.15	-0.01	-0.11	-0.02	-0.11	†
F5	-0.11	0.02	-0.11	-0.33	-0.06	0.00
F7	-0.29	0.00	-0.23	-0.19	-0.02	-0.17
F8	-0.07	-0.10	-0.13	-0.12	-0.05	-0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.50	1.15	1.17	1.59
F11	1.16	1.21	1.18	†	1.09	1.20
F12	-0.02	-0.02	-0.02	0.00	-0.02	-0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 6.30: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.8.1 on “server 1” without absolute value.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.10	0.33	0.32	0.13	0.11	0.33
F2	0.12	0.29	0.33	0.12	0.40	0.39
F3	0.05	0.90	0.00	0.07	0.03	0.05
F4	0.05	0.06	0.02	0.81	0.06	0.89
F5	-0.15	-0.02	-0.11	-0.03	-0.12	†
F5	†	-0.13	-0.14	-0.36	-0.11	-0.16
F7	-0.29	-0.03	-0.24	-0.06	0.14	-0.07
F8	-0.07	-0.10	-0.13	-0.08	-0.05	-0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.55	1.15	1.17	1.59
F11	1.16	1.21	1.18	†	1.09	1.20
F12	-0.02	-0.02	-0.02	0.00	-0.02	-0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 6.31: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.9.3 on “server 2” without absolute value.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.10	0.33	0.07	0.13	0.11	0.33
F2	0.12	0.08	0.33	0.12	0.12	0.10
F3	0.05	0.04	0.00	0.07	0.03	0.05
F4	0.05	0.06	0.02	0.02	0.06	0.05
F5	-0.15	-0.02	-0.11	-0.03	-0.12	†
F5	†	-0.13	-0.14	-0.36	-0.11	-0.16
F7	-0.29	-0.03	-0.24	-0.19	-0.02	-0.18
F8	-0.07	-0.10	-0.13	-0.14	-0.05	-0.14
F9	1.21	1.20	1.55	1.15	1.18	1.50
F10	1.21	1.20	1.50	1.15	1.17	1.59
F11	1.16	1.06	1.18	†	1.09	1.20
F12	-0.02	-0.02	-0.02	0.00	-0.02	-0.05

† No optimal solution was found within the limit of 2000 iterations.

Table 6.32: Relative error of the objective function with respect to LOQO for each family and problem with IPOPT 3.12.8 on “server 2” without absolute value.

Family/Type	T1	T2	T3	T4	T5	T6
F1	-0.02	-0.02	-0.07	-0.01	-0.01	-0.03
F2	0.00	-0.04	-0.03	-0.01	0.00	-0.02
F3	-0.01	-0.02	-0.06	0.00	-0.03	-0.02
F4	0.00	0.00	-0.03	-0.04	0.00	0.00
F5	-0.15	-0.02	-0.15	0.02	-0.12	-0.06
F5	-0.15	-0.13	-0.15	-0.20	-0.12	-0.17
F7	-0.29	-0.03	-0.27	-0.20	-0.03	-0.20
F8	-0.07	-0.10	-0.17	-0.13	-0.05	-0.13
F9	0.07	0.16	-0.09	0.14	0.18	-0.02
F10	0.07	0.11	-0.06	0.08	0.17	0.01
F11	-0.05	-0.01	-0.05	0.00	0.06	0.02
F12	-0.03	-0.02	-0.02	0.00	-0.02	-0.07

Table 6.33: Relative error of the objective function with respect to LOQO for each family and problem with KNITRO 10.1.0 direct algorithm 1 on “server 2” without absolute value.

Family/Type	T1	T2	T3	T4	T5	T6
F1	0.13	0.13	†	†	0.38	0.11
F2	0.14	0.11	0.27	0.26	0.14	†
F3	0.32	0.31	0.02	0.36	0.62	†
F4	0.31	0.07	0.22	0.28	0.33	0.07
F5	-0.15	0.00	-0.11	-0.03	-0.11	-0.03
F5	-0.10	-0.07	-0.09	-0.36	-0.05	-0.13
F7	-0.29	-0.03	†	-0.19	-0.02	-0.19
F8	-0.07	-0.10	-0.13	-0.13	-0.05	-0.13
F9	1.96	1.31	2.03	1.24	1.94	1.58
F10	1.98	1.31	1.54	1.87	1.92	†
F11	†	2.17	†	1.61	†	†
F12	†	0.05	-0.01	†	0.05	†

† No optimal solution was found within the limit of 2000 iterations.

Table 6.34: Relative error of the objective function with respect to LOQO for each family and problem with KNITRO 10.1.0 conjugate gradient algorithm 2 on “server 2” without absolute value.