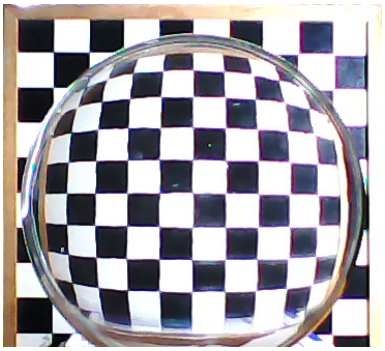
Towards a helioscope based on a tunable ball lens



THESIS

submitted in partial fulfillment of the
requirements for the degree of

BACHELOR OF SCIENCE
in
PHYSICS

Author :	Florine de Villeneuve
Student ID :	1858815
Supervisor :	Dr. Wolfgang Löffler
Second corrector :	Prof.dr. Michel A.G.J. Orrit

Leiden, The Netherlands, January 5, 2021

Towards a helioscope based on a tunable ball lens

Florine de Villeneuve

Huygens-Kamerlingh Onnes Laboratory, Leiden University
P.O. Box 9500, 2300 RA Leiden, The Netherlands

January 5, 2021

Abstract

So far, most well-known helioscopes have to be aligned manually with the ever-changing position of the sun in the sky. In this project, it was assessed whether a gas- or liquid-filled transparent sphere has adequate potential to function as a helioscope. If this would be possible, this helioscope would allow for omnidirectional imaging, solving the problem of constant manual adjustment. The numerical assessment of this gas- or liquid-filled ball lens was done with the help of a ray-tracing simulation, which was tested and confirmed by comparison with analytical as well as experimental results. Then, the suitability of available materials for the experimental assessment was evaluated. The final experimental assessment of the ball lens functioning as a helioscope was done with the help of a miniature version of the envisioned helioscope. It was found that the resolution limit of the ball lens used for the experiments should be sufficient to image a sunspot of average diameter.

Contents

Chapter 1

Introduction: Building a helioscope: using spherical shells for long-distance imaging

For centuries, scientists have been trying to observe the surface of the sun. Even though the star appears big enough in the sky to be directly seen by an observer located on Earth, this is not possible, or at least very damaging to our eyes. Consequently, devices have been invented through which a sharp, yet safely visible image of the sun could be obtained. These devices are called helioscopes.

Christoph Scheiner (1573-1650), a Jesuit priest, physicist and astronomer located in Ingolstadt, Germany [1] was the first to design a helioscope, or, as he termed it, a "machina helioscopica" [2] (see Fig. 1.1). He positioned his helioscope manually, according to the current positions of the sun.

With his helioscope, Scheiner was, from time to time, able to observe spots on the surface of the sun that were darker than the surrounding areas. These dark spots were so-called sunspots (see Fig. 1.2). Sunspots are visible at regions on the surface of the sun where convection has been made impossible by concentrations of magnetic field flux; this results in the surface temperature of the sun being lower in these regions. The diameters of sunspots vary between 16 km and 160 000 km. An observer on earth can, when the dangers this causes for his eyesight are not taken into account, see the larger sunspots without the need of a telescope.

Version of January 5, 2021– Created January 5, 2021 - 11:41

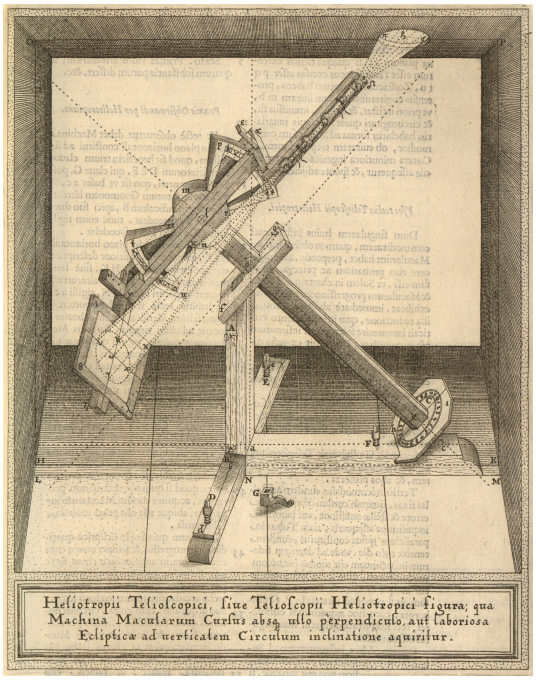


Figure 1.1: *A schematic view of the helioscope as designed by Christoph Scheiner. On the left, the imaging screen can be identified[1].*

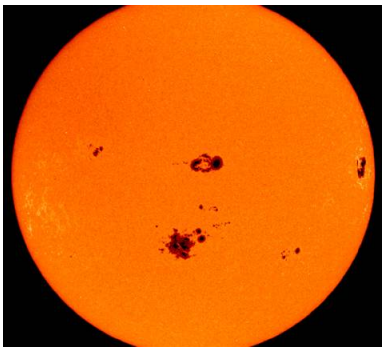


Figure 1.2: *An image of the surface of the sun, on which multiple sunspots are visible [3].*

Yet, it is very rare for a sunspot to appear that is large enough to observe directly. Even the medium-sized and smaller sunspots do not appear reasonably regularly, due to the 11-year long solar cycle [4].

In order to make optimal use of the occasions when sunspots are visible, and to let as many people as possible have a chance of observing and being fascinated by them, a helioscope that would allow for omnidirectional imaging would be ideal. This way, the sun would be observable at every time of day, without the need to realign and readjust any optical equipment.

The goal of this project is to assess the possibility of building a helioscope by using a gas-filled transparent sphere as a large-scale lens that would allow for omnidirectional imaging. The focal length of this lens could be in the order of 100 m, as the refractive index of gases is close to that of air. If placed on a rooftop, it could be used for imaging the surface of the sun onto the ground or a suitable screen. If the resolution of the lens would be sufficient, it might even be possible to image sunspots on the ground in this way. For the best imaging quality of such a ball lens, a shell of at least 1 m in diameter would be most preferable. The bigger the radius of the ball lens, the more sunlight it collects will be focussed properly at the focus of the lens. If the ball lens is too small, spherical aberrations could severely influence the sharpness of the image. Something that also should be taken into account, is the concentration of light at the image. If this is too high, the image will be so bright that there is again a risk of damaging eyes of observers or burning objects located in at the focus.

In this project, it will be assessed whether a gas- or liquid-filled sphere has adequate potential to function as a helioscope. The numerical assessment will be done in Chapter 2, with the help of a ray-tracing simulation. Then, in Chapter 3, the available materials that are best suited for the experimental assessment will be sought after. In Chapter 4, the numerical results of the ray-tracing simulation will be compared to results found experimentally. The final experimental assessment of the ball lens functioning as a helioscope will be done in Chapter 5, with the help of a miniature version of the envisioned helioscope described above.

In order to assess whether such a gas- or liquid-filled lens would be able to image a sunspot, the resolution limit of the lens should be assessed. This gives its minimum resolvable distance. If the diameter of the sunspot is (much) bigger than the minimum resolvable distance, we know that it

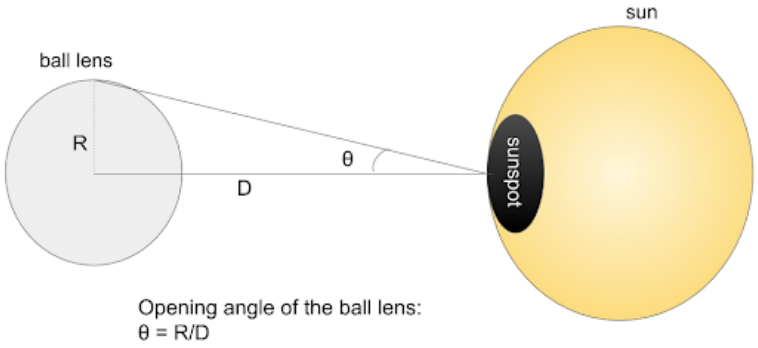


Figure 1.3: *A visualisation of the significance of the so-called "half opening angle" of a ball lens.*

should be possible to image it with this lens. The resolution limit is given by the following Equation, common referred to as the Abbe limit:

$$d_{res} = \frac{\lambda}{2\,n\,\sin\theta} \tag{1.1}$$

where d_{res} is the resolution limit in meters, λ is the wavelength of the incoming light, n is the refractive index of the medium between the lens and the light source and imaging plane [5], and θ is the so-called "half opening angle".

We assume $\lambda = 580$ nm, the wavelength of yellow light. n is in this case equal to 1.0, the refractive index of vacuum. To compute the half opening angle, we compute the following:

$$\theta = \frac{R}{D} \tag{1.2}$$

Where R is the radius of the lens used to image the sunspot, and D is the distance between the lens and the sunspot. The location of this angle is visualised in Fig. 1.3. D is equal to one "Astronomical unit", which is equal to $1.495978707 \cdot 10^{11}$ m [6]. For this example, let us evaluate a hypothetical ball lens with a radius of 0.5 m. If we now calculate the minimum resolvable distance via Eq. 1.1, we find that for a ball lens of this particular size, the minimum resolvable distance is 86.77 km. Thus, a medium-sized sunspot should be easily resolvable by an ideal lens with this diameter.

Chapter 2

Simulations of filled spherical shells

In order to numerically simulate the experiments envisioned in the introduction, a digital ray-tracing simulation was made. This simulation visualises the path of individual rays of light that emerge from a source and travel through an optical system.

The sign convention of the simulation has been decided upon as follows:

- The line all optical components (source, lens, screen) are on is taken to be the z-axis, the x-axis will be the horizontal line perpendicular to the z-axis, and the y-axis will be the vertical line perpendicular to the z-axis and the x-axis;
- All rays will travel from left to right;
- The center of the spherical shell will be at the origin of the coordinate system;
- The radius of curvature will be positive for a convergent surface and vice versa;
- The incoming rays emerge from a source at the negative part of the z-axis. In case of a Huygens source, this source is a point source from which rays expand radially, and in case of a parallel source, each ray emerges at different heights from a line segment left of the lens, and stays parallel to the z-axis before hitting any optical element;

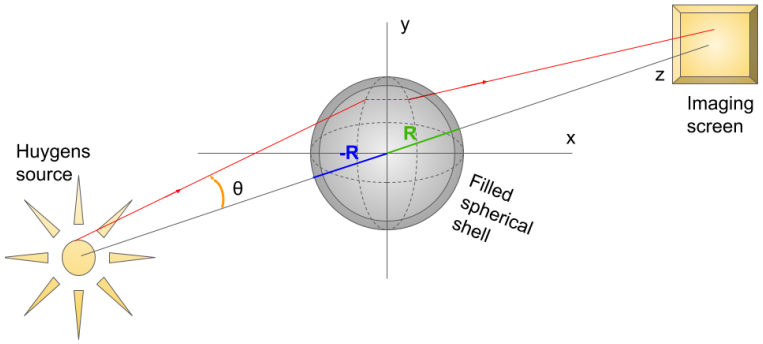


Figure 2.1: *A visualisation of the sign convention in a schematic diagram. A ray (depicted by the red line) emerges from a Huygens source, travels through a ball lens with shell, and finally hits an imaging screen.*

- The ray will be traced up until a variable z-coordinate at the positive part of the z-axis, coinciding with the position of the imaging screen in real space;
- For simplicity, the angle the ray makes with the z-axis only has to be defined in the (z, y)-plane for this simulation. This angle is positive if the z-axis has to be rotated in the counter-clockwise direction in the (z, y)-plane in order to coincide with the ray, and vice versa.

In Fig. 2.1, a visual representation of the sign convention mentioned above is given.

2.1 Analytical description of the refraction in a ball lens with shell

This project focuses on experiments in which all imaging components (light source, ball lens, imaging screen, camera) are aligned in real space, and all optical components are rotationally symmetric around the z-axis. Therefore, when simulating these experimental setups, it suffices to implement a two-dimensional ray tracing model.

If required, the values obtained through ray-tracing could then always be expanded to a three-dimensional equivalent by rotating the simulation around the z-axis.

Furthermore, it suffices if the simulation neglects the rays that do enter the shell, but never enter the space inside the spherical shell. As will be discussed in chapter 3 of the thesis, the spherical shells that were available for this project both had a relatively thin shell thickness compared to their outer radii. Therefore, were the balls to be evenly irradiated, the portion of rays that would only travel through the shell, and not through the space encapsulated by the shell, would be negligibly small.

In Section 2.1.1 and 2.1.2, two possible methods of ray tracing will be assessed. In the third Section, the most adequate method will be implemented for the specific case of a spherical shell.

2.1.1 Analysis using ray transfer matrices

The first technique considered in order to perform ray-tracing on a spherical shell, was ray transfer matrix analysis. This is a very common mathematical ray tracing method. With this method, one is able to describe complex optical systems within one concise ray transfer matrix.

In ray transfer matrix analysis, a 2x2-ray transfer matrix is used to describe each optical element of an optical system, i.e. a surface, interface, mirror or beam travel. Each of these matrices then operates on a vector describing an incoming light ray, in order to calculate a vector describing the outgoing ray. A single, concise ray transfer matrix is obtained by multiplying the matrices in consecutive order.

In order to know whether this method can be used to perform comprehensive ray-tracing on a spherical shell, the constraints of the method must be considered. Because optical objects are usually assumed to be thin, the method of ray transfer matrix analysis is only valid for paraxial rays. Paraxial rays are defined as rays that make a small angle θ to the optical axis of the system, such that the approximation $\sin\theta = \theta$ holds. An implication of a small θ is that, upon entering the ball lens, each ray has a transverse extent that is small compared to the length of the optical system.

In imaging systems where not all rays are paraxial, ray transfer matrix analysis may still describe the magnifications and positions of the focal planes of the system properly, because in decent imaging systems, the paraxial rays must still be focussed correctly. However, ray transfer matrix analysis does not suffice to evaluate aberrations. For this, full ray-tracing techniques are needed [7].

On the basis of this last argument, the ray transfer matrix method has not been used to derive ray-tracing equations for the project, as it would not have given sufficient information on the aberrations of the lens, while these are considerably big in the case of a (thick) ball lens.

2.1.2 Analysis using the law of Snell and the circle-line intersection theorem

In this project, the ray-tracing technique ultimately used was based on two concepts: The law of Snell and the circle-line intersection theorem. Via the resulting ray-tracing technique, all aberrations could be accounted for, and thus a proper ray-tracing analysis was made possible. The two concepts used will be briefly discussed in the next two Sections.

2.1.2.1 The law of Snell

The law of Snell describes the relationship between the incoming and outgoing angle of a ray that passes through a boundary between two different isotropic media (e.g. glass, water, air). It can be stated as follows:

$$\frac{\sin\theta_2}{\sin\theta_1} = \frac{v_2}{v_1} = \frac{n_1}{n_2} \tag{2.1}$$

where, for the respective media, θ is the angle the ray makes with the normal of the interface, v is the velocity of light in the respective medium, and n is defined as the refractive index, a unitless quantity representing the refracting properties of the respective medium [8]. In Fig. 2.2, a visual representation of the law of Snell is shown.

From now on, the incoming and outgoing angles as defined by the law of Snell, θ_1 and θ_2, will be referred to as α and β respectively.

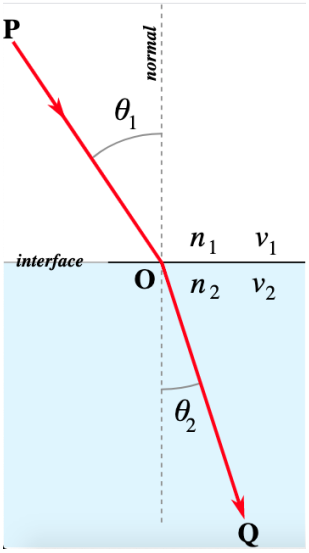


Figure 2.2: *This Figure shows the refraction of a ray of light at the boundary (interface) between medium 1 and medium 2 [8]. When applying the law of Snell, it becomes clear that if the refractive index of the second medium is larger than that of the first medium, the ray in the second medium is closer to the normal.*

2.1.2.2 The circle-line intersection theorem

Suppose one has a line determined by two point coordinates (x_1, y_1) and (x_2, y_2). To find the intersection of this (infinite) line with a circle of radius r and center $(0, 0)$, the following method can be used.

First, we define

$$d_x = x_2 - x_1 \tag{2.2}$$

$$d_y = y_2 - y_1 \tag{2.3}$$

$$d_r = \sqrt{d_x^2 + d_y^2} \tag{2.4}$$

$$D = \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} \tag{2.5}$$

The points of intersection of the circle and the line will now be given by the coordinates

$$x = \frac{D\, d_y \pm \operatorname{sgn}^*(d_y) d_x \sqrt{r^2 d_r^2 - D^2}}{d_r^2} \tag{2.6}$$

$$y = \frac{-Dd_x \pm |d_y| \sqrt{r^2 d_r^2 - D^2}}{d_r^2} \tag{2.7}$$

with the function $\mathrm{sgn}^*(x)$ defined as

$$\mathrm{sgn}^*(x) \equiv \begin{cases} -1 & \text{for } x < 0 \\ 1 & \text{otherwise.} \end{cases} \tag{2.8}$$

The incidence of the circle and the line is therefore determined by the discriminant

$$\Delta \equiv r^2 d_r^2 - D^2 \tag{2.9}$$

for which the following three cases may occur [9]:

Δ	incidence
$\Delta < 0$	no intersection
$\Delta = 0$	tangent
$\Delta > 0$	intersection

Table 2.1: *Different cases for the discriminant.*

2.1.3 Ray-tracing through a spherical shell

In Fig. 2.3, a possible trajectory of a ray (depicted in red) through a ball lens with shell is shown. For the refractive indices of the three different media depicted, we assume that $n_2 > n_3 > n_1$. The (z, y)-coordinates of the source and the angle of incidence are input values for the ray-tracing system, which have to be translated to the corresponding output values: the (z, y)-coordinates of the point where the ray exits the lens and the angle of departure of the ray with respect to the z-axis.

During its trajectory, the ray hits an interface four times in total, before exiting the spherical shell. At each interface, by appying the law of Snell, the outgoing angle of the ray can be calculated for each corresponding incident angle. Then, the coordinates of the following trajectory can be obtained by implementing the circle-line intersection theorem. In Fig. 2.3, the three different parts of the trajectory of the ray in the ball lens are depicted by the line segments KL, LM and MN.

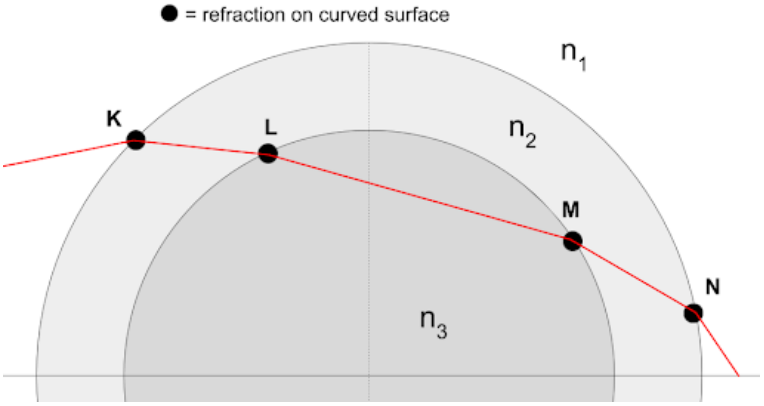


Figure 2.3: *Possible trajectory of a ray through a ball lens with shell.*

2.1.3.1 Calculating the angles of refraction

Fig. 2.4 shows the direct application of the law of Snell on the first interface. The incoming and outgoing angles of the ray with respect to the normal of the interface are labeled α and β respectively.

Each time the ray refracts at an interface of the sphere shell, the α and β that correspond to that refraction are angles that are relative to the surface normal, which is dependent on the height at which the ray hits the interface. These angles correspond to the angles θ_{in} and θ_{out}, the angles the ray makes with the z-axis before and after the interface. The key to translating θ_{in} to α and β to θ_{out} is the angle depicted in Fig. 2.4 as γ, which is the angle the normal of the interface makes with the horizontal axis.

The value of γ is dependent on the transversal coordinate of the point of incidence. This is visible in Fig. 2.5, where it can clearly be seen that the value of γ is larger for the point L (see Fig. 2.5) than for the point K (see Fig. 2.4).

The angle γ is defined as a function of the z- and y-coordinates of the point of incidence:

$$\gamma = \frac{z_{in}}{|z_{in}|} \sin^{-1}\left(\frac{y_{in}}{R}\right) \tag{2.10}$$

where the argument $(\frac{z_{in}}{|z_{in}|})$ determines the sign of gamma, based on whether the point of incidence of the ray is to the left or to the right of the center of the ball.

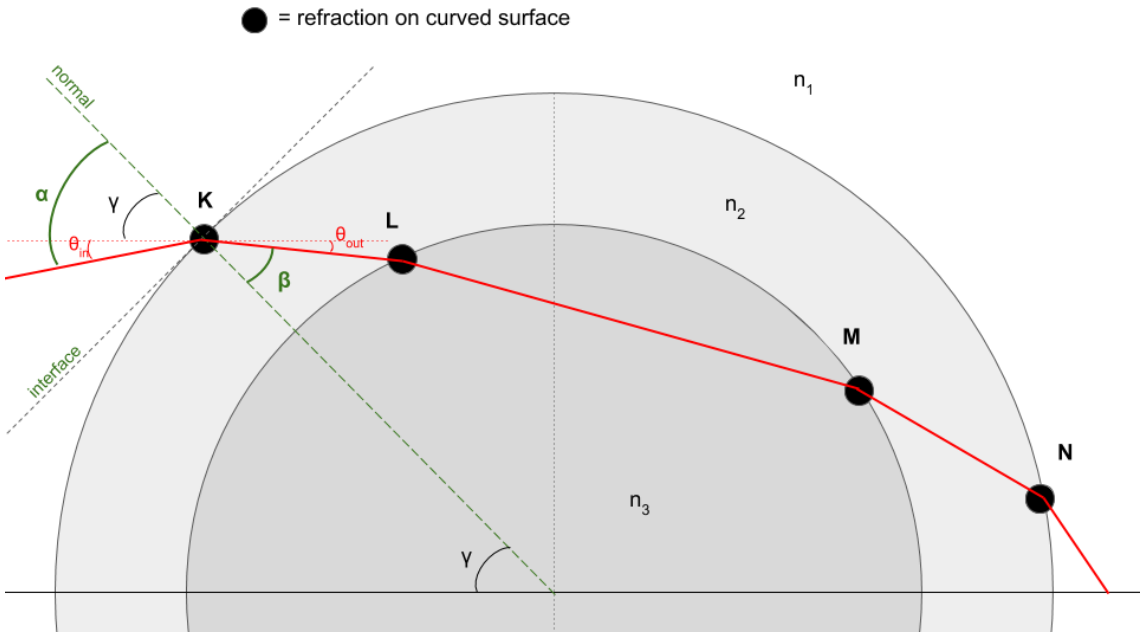


Figure 2.4: *Refraction at point K via the law of Snell.*

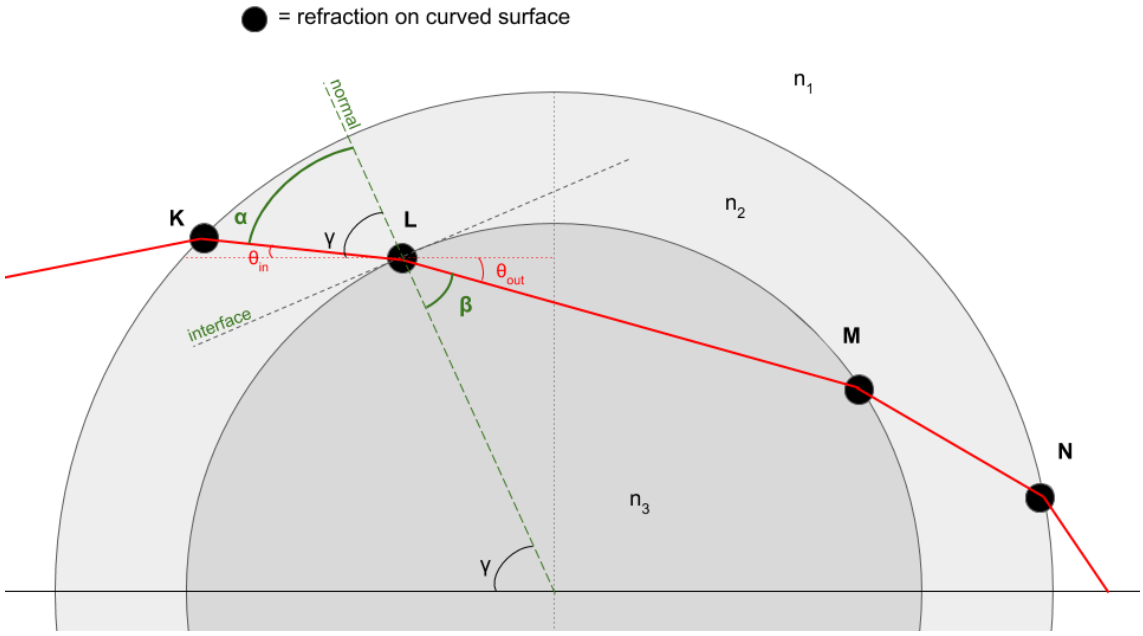


Figure 2.5: *Refraction at point L via the law of Snell.*

The calculation of θ_{out} using the law of Snell can be done for each point according to the following protocol. First, we take an incident ray with a transversal coordinate smaller than the outer radius of the spherical shell. We define

$$\alpha = \theta_{in} - \gamma \tag{2.11}$$

then, using the law of Snell (Eq. 2.1) we can calculate β:

$$\beta = \arcsin\left(\frac{n_1}{n_2}\sin\alpha\right) \tag{2.12}$$

and by translating back to γ we find the value for θ_{out}:

$$\theta_{out} = \beta + \gamma \tag{2.13}$$

2.1.3.2 Calculating the coordinates of the points of incidence

Each time the coordinates have to be calculated of the next point of incidence of the ray at an interface of the ball lens, the line-circle intersection theorem is used. This theorem, as it is outlined in Section 2.1.2, needs a few input variables. Specifically, the radius of the lens interface the ray will hit, as well as the coordinates of two points on the line coinciding with the trajectory the ray followed right before hitting this interface are needed as input. One of the two points on this line is always known already, namely, the former point of incidence of the ray at an interface.

For example, to calculate the coordinates of point L as shown in Fig. 2.3, apart from the radius of the circle point L is on, the coordinates of two points on the line that coincides with line segment KL are also needed as input variables. As explained above, one can use the coordinates of point K as the first of the pair of points. If, conversely, the coordinates of point K are to be calculated, the coordinates of the origin of the ray (the light source) can be used as the first input point in the line-circle intersection theorem.

The second point on the line that coincides with the previous trajectory of the ray can be calculated using the most recent value for the angle the ray makes with the horizontal axis. This angle, along with the coordinates of the first point on the line, can be used to generate coordinates of a second

point on the line. To do this, we note the mathematical definition of the slope of a straight line:

$$\tan\theta = m \tag{2.14}$$

where θ is the angle that line makes with the horizontal axis, and m is the slope of the line [10]:

$$m = \frac{y_2 - y_1}{z_2 - z_1} \tag{2.15}$$

Here, z_1 and y_1 are the coordinates of a known point on the line.

Thus, a second point on the line can be defined as follows:

$$y_2 = (z_2 - z_1)\tan\theta + y_1 \tag{2.16}$$

Where the choice for z_2 is arbitrary. Thus, one can also generate a second point on the line once one has the coordinates of a first point on the line and the angle the line makes with the horizontal axis. This way, two points on the line that coincides with the trajectory of the ray, which are needed input variables for the circle-line intersection theorem, can be generated each time the coordinates of the point where the ray was refracted at the previous interface, as well as its angle of departure, are known.

The line-circle intersection theorem then outputs either the coordinates of two intersection points, the coordinates of a single point if the line is tangent to the sphere, and two imaginary sets of coordinates if the line does not intersect the sphere.

If the calculation outputs the coordinates of two real intersection points of the line and the sphere, the right set of coordinates needs to be chosen, since there is no use for the coordinates of the intersection point that doesn't correspond with the coordinates of the next point of incidence in the trajectory of the ray.

To pick the right coordinates, one needs to consider which of the four possible points of incidence of the ray is being calculated:

1. The point where the ray hits the outer shell when entering the spherical shell for the first time: point K in Fig. 2.3;
2. The point where the ray hits the inner shell after having travelled through the shell for the first time: point L in Fig. 2.3;

3. The point where the ray hits the inner shell again after having traveled through the inner sphere of the spherical shell: point M in Fig. 2.3;

4. The point where the ray hits the outer shell again after having travelled through the shell again: point N in Fig. 2.3.

Assuming the ray travels from left to right, in the first two cases, the point with the lower longitudinal coordinate needs to be chosen. In cases 3 and 4, the point with the higher longitudinal coordinate then needs to be chosen.

2.2 Digital implementation of ray-tracing programme

The digital ray-tracing simulation was made in Python using iPython Notebook. The main part was created using the following functions, of which some are used as subroutines:

`refract`

This subroutine is the numerical equivalent of the process explained under Section 2.1.3.1. It can apply the law of Snell to an incoming ray, to find the angle the outgoing ray makes with the horizontal axis.

`circle_line_intersection`

This subroutine is the numerical execution of the process explained under Section 2.1.2.2. It can apply the circle-line intersection theorem to a circle with given radius and a line with a given angle relative to the horizontal axis and one given set of coordinates of a point on the line.

`ball_with_shell`

This subroutine traces the path of a single ray through the ball lens with shell, by alternately performing the `refract` and `circle_line_intersection` subroutines until the coordinates of the point where the ray departs from

the spherical shell and the angle the ray makes with the horizontal axis have been calculated, which will then be the values this subroutine returns.

In the following list, the process of finding and matching the right coordinates for the next point of refraction is outlined, for each of the four possible refraction points discussed in the list under Section 2.1.3.2.

1. The point where the ray hits the outer shell when entering the spherical shell for the first time: point K in Fig. 2.3: take the smallest z-coordinate. If θ_{in} is positive, take the smallest y-coordinate, and vice versa;
2. The point where the ray hits the inner shell after having travelled through the shell for the first time: point L in Fig. 2.3: take the smallest z-coordinate. If θ_{in} is positive, take the smallest y-coordinate, and vice versa;
3. The point where the ray hits the inner shell again after having traveled through the inner sphere of the spherical shell: point M in Fig. 2.3: take the biggest z-coordinate. If θ_{in} is positive, take the biggest y-coordinate, and vice versa;
4. The point where the ray hits the outer shell again after having travelled through the shell again: point N in Fig. 2.3: take the biggest z-coordinate. If θ_{in} is positive, take the biggest y-coordinate, and vice versa.

`Huygens_source`

This function collects the output values of the `ball_with_shell` subroutine for a given number of rays that make evenly increasing angles with the horizontal axis and are all emerging from a variable point. This variable point thus emits rays like a Huygens source.

`parallel_source`

This function collects the output values of the `ball_with_shell` subroutine for a given number of evenly spaced incoming rays that are parallel to the horizontal axis.

The source code of the program, along with explanatory comments, can be found under Appendix 1.

2.2.1 Limitations of digital implementation

The ray-tracing simulation does not work for rays that exit the inner circle of the simulated circular lens on the left of the y-axis. This is because it has not been worked out properly how the angles α and β are defined in terms of γ, θ_{in} and θ_{out} in this case. Also, the `ball_with_shell` function does not always pick the right z- and y-coordinates of the next refraction point in the above named circumstances.

The consequence of this is that these rays will not be fully traced, just like the rays that only travel through the shell that are also left out of the simulation. This proved not to be a problem, because of the same reason why it was not a problem to leave out the rays only travelling through the shell. As will be discussed in chapter 3 of the thesis, the available spherical shells both had a relatively thin shell thickness compared to their outer radii. Therefore, were the balls to be evenly irradiated, the portion of rays that would exit the ball before passing the (x, y)-plane dissecting the ball at $z = 0$, would be negligibly small for realistic thin-shell spheres.

2.3 Ray-tracing simulation results

Now, to see if the simulation works correctly, we study five distinct cases of different combinations of refractive indices. In each of the cases, the refractive indices of the inner ball, the ball shell and the space outside the ball (from now on, these will be referred to as n_3, n_2 and n_1 respectively), will take on different values. A plot for each of the five cases can be found in Fig. 2.6-2.10. In Section 2.3.3, the simulation will be compared numerically to an analytical formula that was found in a 2019 study by Elagha, which describes the refraction of rays in ball lenses with shells.

The invariable conditions under which each of the five cases were studied, were as follows.

- $R_s = 0.1$ m, $R_b = 0.07$ m, $n_1 = 1.0$.
- The number of incoming rays was 30; for case 1 and 2, the height of incoming rays varied between $-R_b$ and R_b; for the rest of the cases, the height of incoming rays varied between -0.06 m and 0.06 m.

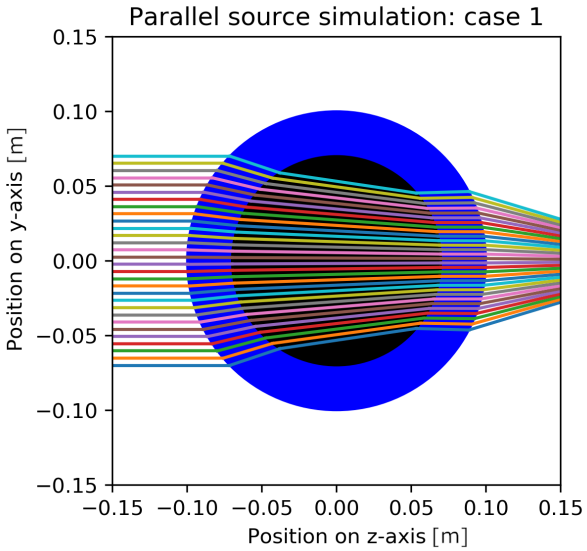


Figure 2.6: *Result of ray-tracing simulation for case 1: $n_2 > n_3 > n_1$. Here the case of a spherical shell ($n_2 = 1.6$) in air ($n_1 = 1.0$), filled with water ($n_3 = 1.33$) is studied. It is visible that the rays converge after having travelled through the ball lens.Thus, the ball lens acts as a convergent lens in this situation.*

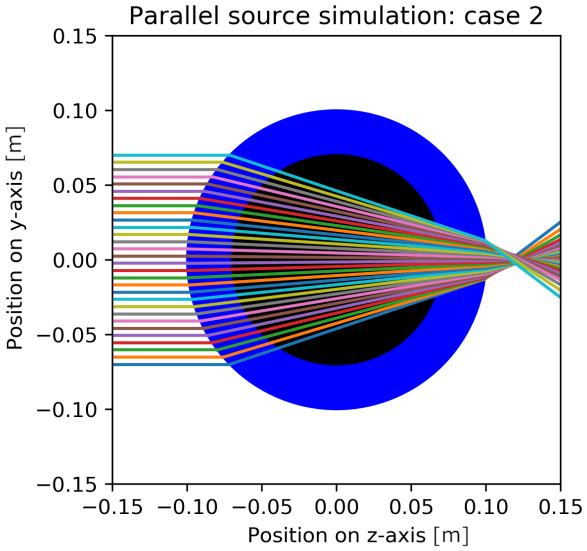


Figure 2.7: *Result of ray-tracing simulation for case 2: $n_2, n_3 > n_1$. In this case, $n_2 = n_3 = 1.6$. the ball lens refracts the rays in the same way as a ball lens of radius R_s without a shell, a "homogenous ball lens" that has the same refractive index everywhere would. It is clearly a convergent lens.*

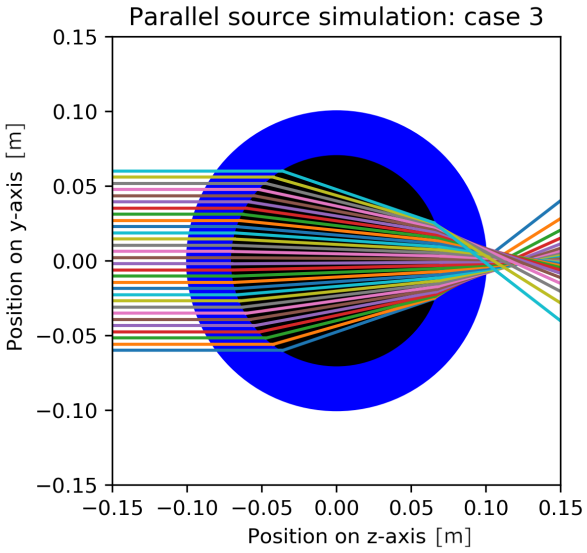


Figure 2.8: *Result of ray-tracing simulation for case 3: $n_3 > n_1, n_2$. In this case, $n_2 = n_1 = 1.0$. The ball lens refracts the rays in the same way as a ball lens of radius R_b without a shell, a "homogenous ball lens" that has the same refractive index everywhere. This refractive index is $n_3 = 1.33$. It is clearly a convergent lens.*

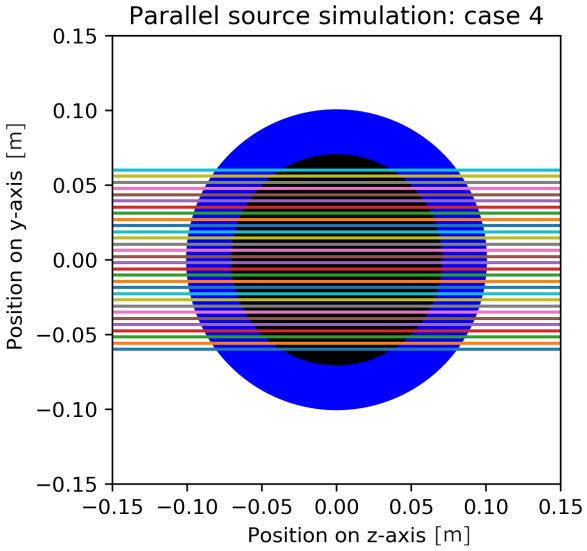


Figure 2.9: *Result of ray-tracing simulation for case 4: $n_1 = n_2 = n_3 = 1.0$. In this case, none of the rays are refracted, and thus all rays traverse in the same way as when there would be no ball lens. The ball lens is neither convergent nor divergent.*

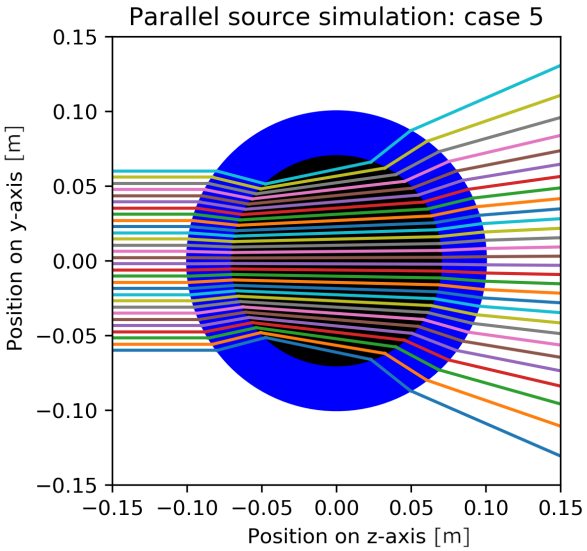


Figure 2.10: *Result of ray-tracing simulation for case 5: $n_2 > n_1, n_3$. Here the case of an empty (or air-filled; $n_1 = n_3 = 1.0$) spherical shell ($n_2 = 1.6$) in air is studied. It is visible that the rays diverge after having travelled through the ball lens. Thus, the ball lens is a divergent lens in this case.*

Note that the empty spherical shell in air is a negative lens (case 5; Fig. 2.10), whereas the spherical shell in air filled with a substance with a refractive index of 1.33 (case 1; Fig. 2.6), e.g. water, acts as a positive lens. In Section 2.4, this finding will be studied in a more in-depth way. An analytical formula will be derived for the focal length of the ball lens for varying values of R_s, R_b, n_1, n_2 and n_3.

2.3.1 Determination of focal length and spherical aberrations

In order to assess the spherical aberrations of the simulated ball lens, the points where parallel incoming rays intersect the z-axis after travelling through a homogenous ball lens (a ball lens for which the refractive index of the inner ball is the same as that of the "shell") have been compared to the effective focal point of that ball lens. The effective focal length has been defined as follows for a homogenous ball lens (a ball lens that has the same refractive index everywhere).

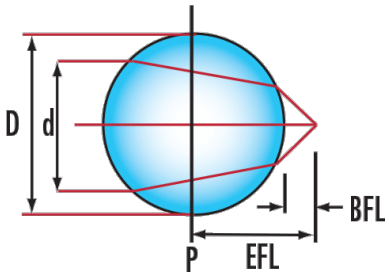


Figure 2.11: *Visual representation of the effective focal length (EFL) and back focal length (BFL) of a ball lens [11]. The lens diameter is given by D, the diameter of the incoming beam of light is labeled as d, and the cross-section of the ball lens itself is labeled P, it being an optical element with lens power P.*

$$EFL = \frac{n\,D}{4(n-1)} \tag{2.17}$$

where the effective focal length (EFL) is the distance between the center of the ball lens and its focal point, n is defined as $\frac{n_{ball\,lens}}{n_{outside}}$, and D is the diameter of the ball lens.

In Chapter 3, the so-called back focal length (BFL) will be used. This has been defined as follows for any ball lens with a known EFL and diameter D[11]:

$$BFL = EFL - \frac{D}{2} \tag{2.18}$$

A visual representation of the EFL and BFL of a ball lens can be found in Fig. 2.11.

In Fig. 2.12, the z-coordinate of the point where the ray intersects the z-axis after passing through a homogenous ball lens with a refractive index of 1.5 is plotted against the initial height of a set of equally spaced incident rays. The red dot represents the effective focal point calculated using Eq. 2.17. The values for the simulation variables have been chosen as follows:

- The size of the polystyrene ball (see Chapter 3) has been used: $R_s = 0.1$ m, $R_b = 0.0977$ m. For the ball lens, the following refractive indices have been chosen in order to make it homogenous: $n_1 = 1.0$, $n_2 = 1.5$, $n_3 = 1.5$.
- For the Huygens source simulation, the distance between the Huygens source and the center of the ball lens has been chosen as 1.0 m.

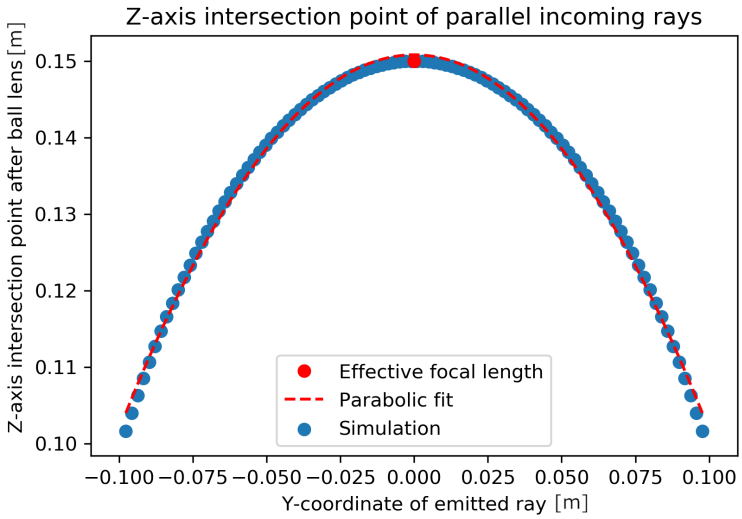


Figure 2.12: *Simulation of effective focal length and spherical aberrations. The "focal point" of a single ray (the point where it intersects the z-axis after travelling through the ball lens) is found to be closer to the ball lens the higher a ray enters the ball lens. This is due to spherical aberrations.*

The incoming angle of rays emitted by the Huygens source (labeled θ_{in} and θ_0 in the constructed simulation and Eq. 2.20 respectively) have been varied between 0.1 rad and -0.1 rad;

- In the ray-tracing simulation, 100 data points have been generated.

In Fig. 2.12, it is clearly visible that the further from the z-axis the incoming parallel ray enters the ball, the greater the distance will be between the point of intersection with the z-axis and the effective focal point of the ball lens. This difference is due to spherical aberrations. The rays in the paraxial approximation intersect the z-axis at points that are relatively very close (almost identical) to the effective focal point. The parabolic fit shows that the dislocation of the focal point for each of the rays due to aberrations can be described by a second-order polynomial. The Equation for the parabola was found as follows (note: due to convention, the horizontal coordinate is labelled as z and the vertical coordinate as y):

$$\frac{1}{m}y = \frac{-4.906}{m^2}z^2 + 0.1508$$

2.3.2 Determination of virtual image and spherical aberrations

The fact that the empty spherical shell functions as a diverging lens (see Fig. 2.10), implies that there will be no actual focus points where rays intersect after they have traveled through the lens. However, if the diverging ray paths that emerge from the back of the lens are extended to the front of the lens, we do find "imaginary" focus points where rays do intersect. The collection of these "imaginary" focus points of a diverging lens is called its virtual image [12].

Fig. 2.13 shows the position of the virtual image of the empty ball lens with shell, for varying shell thicknesses. Each coloured scatter plot represents the simulation results for a ball lens with a different shell thickness. In the legend, the value for the inner radius of the spherical shell (R_b) is given; the outer radius of the spherical shell (R_s) is, in this simulation, always equal to 10 cm. Rays that have a virtual image positioned inside of the ball lens are excluded. Hence, results in this plot do not include y-values higher than -10 cm (which is equal to $-R_s$). An aperture with a radius of $\frac{\pi}{10}$ in front of the ball lens was used in the simulation. The minimum of each scatter plot represents the effective focal length of the virtual image for a certain Huygens source.

In all cases, a parabola fits the simulation results very well. This agrees with the finding that all spherical aberrations of the ball lens with shell are parabolic.

2.3.3 Comparison of the numerical simulation to the analytical result of Elagha

In a 2019 paper by Elagha, the following Equations for ball lenses with shells have been found to describe a ray emerging from a Huygens source on the z-axis and a parallel incident ray travelling through a ball lens with shell. The Equation for a ray emerging from a Huygens source on the z-axis, located at a distance K_0 from the center C at $(0, 0)$, of a symmetric monocentric lens having N spherical shells centered at C, is stated as fol-

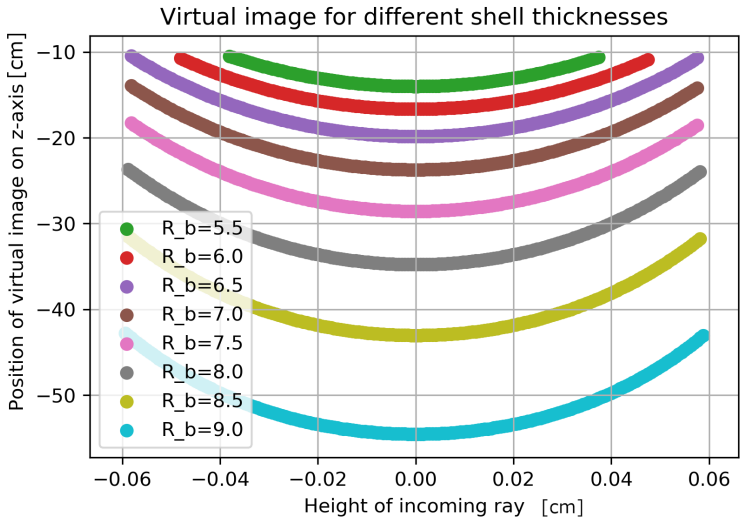


Figure 2.13: *Virtual image for different shell thicknesses.*

lows:

$$\frac{1}{B_1} = \frac{1}{r_1} \sin\left[2\sum_{i=1}^{N}\left(\sin^{-1}\frac{r_1}{R_i\prod_{p=1}^{i}\mu_{p-1}} - \sin^{-1}\frac{r_1}{R_i\prod_{p=1}^{i}\mu_p}\right) - \theta_0\right] \tag{2.19}$$

where:

- The incident ray makes an angle θ_0 with the optical axis and a final image I is formed at a distance B_1 from C;
- r_1 is given as follows: $r_1 = K_0 \sin\theta_0$, where K_0 is the distance of the Huygens source from the center of the ball lens;
- N is the number of spherical shells of the lens centered at $(0,0)$;
- R_i is the radius of the i^{th} shell encountered when travelling out from the center of the lens;
- μ_p is is the relative refractive index at the pth surface encountered when travelling out from the center of the lens, i.e., $\mu_p = \frac{n_p}{n_{p-1}}$, where $\mu_1 = \frac{n_1}{n_0} = \frac{n_1}{1} = n_1$.

In this project, a ball lens with two spherical shell interfaces is studied. For this case, Eq. 2.19 reduces to

$$\frac{1}{B_1} = \frac{1}{r_1} \sin\left[2\left(\sin^{-1}\frac{r_1}{R_1} - \sin^{-1}\frac{r_1}{R_1\mu_1} + \sin^{-1}\frac{r_1}{R_2\mu_1} - \sin^{-1}\frac{r_1}{R_2\mu_2\mu_1}\right) - \theta_0\right] \tag{2.20}$$

For the Equation of a parallel incoming ray at a height h, set the angle $\theta_0 = 0$ and $r_1 = h$; Eq. 2.20 then reduces to [13]:

$$\frac{1}{B_1} = \frac{1}{h} \sin\left[2\left(\sin^{-1}\frac{h}{R_1} - \sin^{-1}\frac{h}{R_1\mu_1} + \sin^{-1}\frac{h}{R_2\mu_1} - \sin^{-1}\frac{h}{R_2\mu_2\mu_1}\right)\right] \tag{2.21}$$

In Fig. 2.14 and Fig. 2.15, values of z-axis intersection points of rays obtained through the simulation that was constructed for this project are compared with values obtained through the application of Eq. 2.19 to the specific case of a ball lens with two spherical shell interfaces. In Fig. 2.14, the angle that a ray emitted from a simulated Huygens source makes with the z-axis is plotted against the z-coordinate where those rays intersect the z-axis after passing through the ball lens. In Fig. 2.15, the y-coordinate of parallel incoming rays is plotted against the z-coordinate where the rays intersect the z-axis after passing through the ball lens.

The simulation parameters have been set as follows:

- The properties of the polystyrene ball (see Chapter 3) filled with water have been used: $R_s = 0.1$ m, $R_b = 0.0977$ m, $n_1 = 1.0$, $n_2 = 1.6$, $n_3 = 1.33$.
- For the Huygens source simulation, the distance between the Huygens source and the center of the ball lens has been chosen as 1.0 m. The incoming angle of rays emitted by the Huygens source (labeled θ_{in} and θ_0 in the constructed simulation and Eq. 2.20 respectively) have been varied between 0.1 rad and -0.1 rad;
- For the parallel source simulation, the incoming height of parallel rays (labeled y_{in} and h in the constructed simulation and Eq. 2.21 respectively) have been varied between -0.0977 m and 0.0977 m;
- In both ray-tracing simulations, 100 data points have been generated.

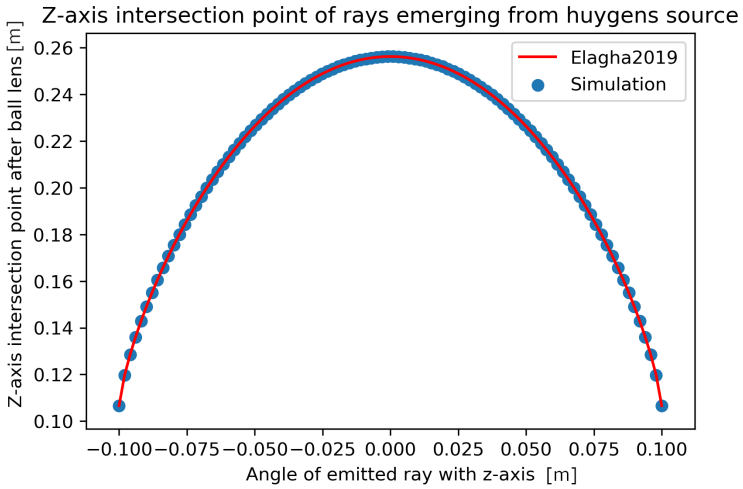


Figure 2.14: *z-axis intersection points found by simulation and found by Elagha for a Huygens source.*

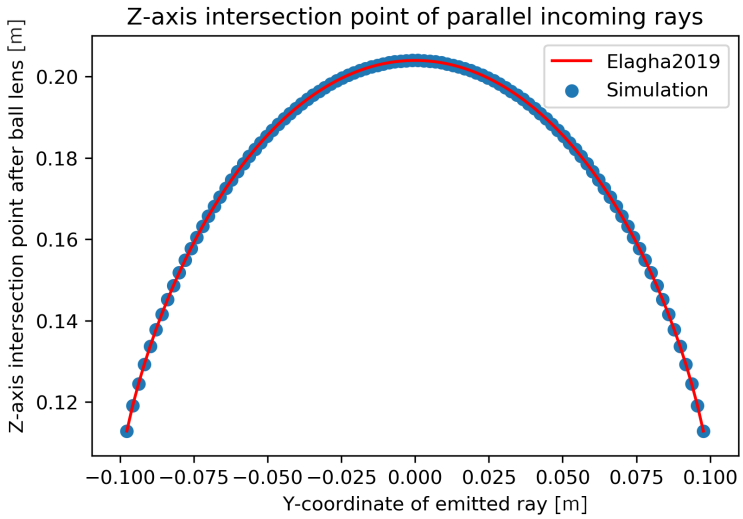


Figure 2.15: *z-axis intersection points found by simulation and found by Elagha for a parallel source.*

It was found that the values obtained through the ray-tracing simulation were equal within numerical accuracy to those obtained through the application of Eq. 2.19 to the specific case of a ball lens with two spherical shell interfaces.

Version of January 5, 2021– Created January 5, 2021 - 11:41

Chapter 3

Discussion of available materials for imaging experiments

In the introduction, the idea of the helioscope in the form of a gas-filled glass sphere that would allow for omnidirectional imaging was introduced. In order to be able to image real sunspots, a high imaging quality of the ball lens would be needed. For the best imaging quality of the spherical shell, a polished glass shell of at least 1 m in diameter would be most preferable, consisting of one piece and containing two holes for supply and drainage of its contents. Yet, in reality, no spherical shell that met all these requirements could be found. It was especially hard to find a spherical shell with a matching diameter; it was easier to find spherical shells with smaller diameters. Thus, it was decided to aim for experimenting with a miniature version of the helioscope described in the introduction.

Another practical constraint was the room available for proper imaging. In order to test the lens power (as would be done in the experiment discussed in chapter 5), it was preferable to work in a dark space, and let the light source the ball would image be the only light in this space. A suitable room, being about 8 m in length, was found. Yet, this meant that the focal length of the lens should be shrunk from 100 m (for the helioscope described in the introduction) to about 1.5 m, maximum 2.0 m. With a focal length of such a magnitude, imaging experiments could be done where the lens could function as a relay lens, imaging an object located at $2f$ on an image plane located at $2f$ on the opposite side of the lens [14].

In Section 3.1, the ball lenses that were available for this project will be introduced. In order to find the most suitable spherical shell and substance encapsulated by it, a formula will then be derived in Section 3.1.1 that relates the focal length of a ball lens with shell to its size, thickness and the refractive indices of the shell, the filling and the space outside the ball lens. The possible combinations of ball lens and filling will then be assessed numerically using this formula. It was found that putting the ball lens in an aquarium would be beneficial, as this would make it possible to switch from using a gas to fill the ball lens, to using a liquid for this, which proved to be much easier. Finally, it was found to be beneficial to use an aperture in front of the aquarium.

3.1 Consideration of different available ball lenses and lens components

Two different spherical shells were available for the project. Both consisted of two halves, which were initially not yet attached to one another. The first spherical shell was made of polystyrene and will from now on be referred to as the "polystyrene ball lens". The second was made of BK7 glass and will from now on be referred to as the "BK7 ball lens".

The polystyrene ball lens (see Fig. 3.1) had a diameter of 20 cm and a shell thickness of 2.3 mm. The material was well polished, which would make for relatively sharp images. Furthermore, the two halves, made of polystyrene, a very lightweight material, could easily be joined together by means of epoxy glue. Drilling two holes in the shell for water supply and drainage would also be relatively easy, as the material was fairly flexible.

The BK7 ball lens (see Fig. 3.2) had a diameter of 43.3 cm and a shell thickness of 14.6 mm. It was a lens that had previously been in use at the bottom of the Mediterranean sea, as part of the KM3NeT neutrino telescope [15]. The material was not well polished, which meant that imaging through this lens would make for relatively less sharp images. Glueing the two halves together would be quite a challenge, as the material (thick shell of BK7 glass) was quite heavy, and maybe the shell would not be able to withstand the pressure of a liquid or gas inside the ball lens. There was also a risk of shattering the shell if two holes were to be drilled in the shell for water supply and drainage.

Figure 3.1: *The polystyrene ball lens in some different sizes [16].*

Figure 3.2: *The BK7 ball lens.*

3.1.1 Derivation of focal length formula via evaluation of vergence

In this Section, a formula for the focal length of any ball lens with shell will be derived via so-called vergence tracing. The found formula for the focal length will then be used to assess what refractive index of the substance inside the spherical shell would be desirable for each of the available ball lenses. This information can then be used to further investigate which of the two ball lenses would be most suitable for the project, as this also depends upon how easy it is to find or make a substance with the refractive indices that are desirable for each of the ball lenses.

The vergence of a lens situated in air is defined as follows:

$$V = \frac{n}{f} \tag{3.1}$$

where V is the vergence, measured in dioptres (m^{-1}), n is the refractive index of the lens and f is the focal length of the lens in m, where we define $V > 0$ when the focus is to the right of the source location. A parallel ray has a vergence of 0 m^{-1}, since its focus is located at infinity. In this derivation, spherical wavefronts are thus assumed. For thick lenses (e.g. the ball lens with shell), vergence tracing is only valid for paraxial rays, in this case that means rays with a small angle θ to the optical axis.

In order to use vergence tracing, two actions have to be defined.

1. Propagation of the ray over a distance d through a medium with a refractive index n. This action corresponds to translating the distance to the focus. Mathematically, it is defined as follows:

$$V' = n \left(\frac{n}{V} - d \right)^{-1} \tag{3.2}$$

or, in comparable form,

$$V' = \left(\frac{1}{V} - \frac{d}{n} \right)^{-1} \tag{3.3}$$

Where V is the vergence before the propagation, V' is the vergence after the propagation, n is the refractive index of the medium, and d is the propagation distance. The sign of d is negative if the ray moves closer to the focus, and vice versa.

2. Refraction through an optical element with lens power P. This action changes the vergence as follows:

$$V' = V + P \tag{3.4}$$

The lens power of a thin spherical lens can be deduced from the lensmaker's Equation, which is known as the following [17]:

$$\frac{1}{f} = \left(\frac{n_1}{n_2}\right)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \tag{3.5}$$

In this formula, $\frac{1}{f}$ is actually equal to the lens power P of the lens. For a spherical lens, $R_1 = -R_2 = R$; the first radius of (convex) curvature is equal to the second radius of (concave) curvature for a perfectly round ball lens. Thus, Eq. 3.5 can be simplified as follows for a thin spherical lens:

$$P = \frac{1}{f} = \frac{2(n_2 - n_1)}{R} \tag{3.6}$$

with $R = R_{ball\ lens}$.

for a thick spherical lens like the ball lens of this project, the convex and concave spherical interfaces of the lens have to be evaluated separately, and propagation through the ball lens should also be taken into account.

For a single convex spherical interface between n_1 and n_2, the lensmaker's Equation reduces to

$$P = \frac{n_1 - n_2}{R} \tag{3.7}$$

as R_2 in Eq. 3.5 is set to infinity.

For a single concave spherical interface between n_2 and n_1, set R_1 to infinity in Eq. 3.5; the Equation then reduces to

$$P = (n_1 - n_2)\frac{-1}{-R} = \frac{n_1 - n_2}{R} \tag{3.8}$$

Note this expression is equal to Eq. 3.7.

3.1.1.1 Ball lens with vergence

The vergence trajectory through a ball lens with radius R and refractive index n is as follows:

$V_0 = 0 \rightarrow P_1 : V_1 \rightarrow$ propagation over $2R$ in $n : V_2 \rightarrow P_2 : V_3$

where $V_i \rightarrow x : V_{i+1}$ means that V_{i+1} is the new vergence calculated by performing action x on V_i. P_1 and P_2 are equal to Eq. 3.7, as seen in the previous Section. Thus, via Eq. 3.4, V_1 and V_3 can be calculated from V_0 and V_2 respectively. In the case of a homogenous ball lens in air, n_1 is equal to 1.0, and n_2 is equal to n, the refractive index of the ball lens. The propagation through the ball lens with refractive index n and radius R makes for the following operation on the vergence V_1:

$$V_2 = \left(\frac{1}{V_1} - \frac{2R}{n} \right)^{-1} \tag{3.9}$$

To transform the resulting vergence V_3 back to a focal distance, calculate $f = \frac{1}{V_3}$. The procedure outlined above results in the following expression:

$$f = \frac{(-2+n)R}{2(n-1)} \tag{3.10}$$

This proves to be equal to Eq. 2.18, the back focal length of a ball lens. This makes sense, as the f calculated through this vergence tracing procedure is also measured from the second spherical interface ("back") of the ball lens.

3.1.1.2 Sphere shell lens with vergence

Now, the vergence trajectory through a ball lens with shell, with inner ball radius R_b and shell radius R_s, will be calculated. The refractive indices of the inner ball, the ball shell and the space outside the ball are labeled n_3, n_2 and n_1 respectively. The vergence trajectory looks as follows:

$$V_0 = 0 \rightarrow P_s : V_1 \rightarrow \text{propagation over } R_s - R_b \text{ in } n_2 : V_2 \rightarrow$$

$$P_b : V_3 \rightarrow \text{propagation over } 2R_b \text{ in } n_3 : V_4 \rightarrow P_b : V_5 \rightarrow$$

$$\text{propagation over } R_s - R_b \text{ in } n_2 : V_6 \rightarrow P_s : V_7$$

It is assumed that $R_s > R_b$, and all variables are larger than zero. From Eq. 3.7, it follows that:

$$P_s = \frac{n_2 - n_1}{R_s} \tag{3.11}$$

and

$$P_b = \frac{n_3 - n_2}{R_b} \tag{3.12}$$

The change in vergence due to propagation over the mentioned distances in the corresponding media can be found through Eq. 3.3.

Via the vergence trajectory outlined above, V_7 was calculated, and the back focal distance was found by calculating

$$f_{bfl} = \frac{n_1}{V_7} \tag{3.13}$$

To translate this to the effective focal distance, calculate

$$f = f_{bfl} + R_s \tag{3.14}$$

This results, after simplification, in the following expression for f:

$$f(R_s, R_b, n_1, n_2, n_3) = \frac{-n_2 n_3 R_b R_s}{2(-n_2 n_3 R_b + n_1 n_3 (R_b - R_s) + n_1 n_2 R_s)} \tag{3.15}$$

Eq. 3.15 will be used in Section 3.1.1.3 to characterize the focal length of the polystyrene and BK7 ball lens filled with substances with different refractive indices n_3.

3.1.1.3 Focal length characterisation of both available lenses

In Fig. 3.3 and Fig. 3.4, the effective focal length, as calculated through the steps under Section 3.1.1.2, is plotted against the refractive index of the substance in the polystyrene and BK7 ball lens respectively.

The data points that are to the left of the asymptote correspond to negative values for f, meaning that a ball filled with a substance of these refractive indices will produce a virtual image and is thus a diverging lens. Thus, one can clearly see that an empty spherical shell (n_3 = 1.0) is a diverging lens.

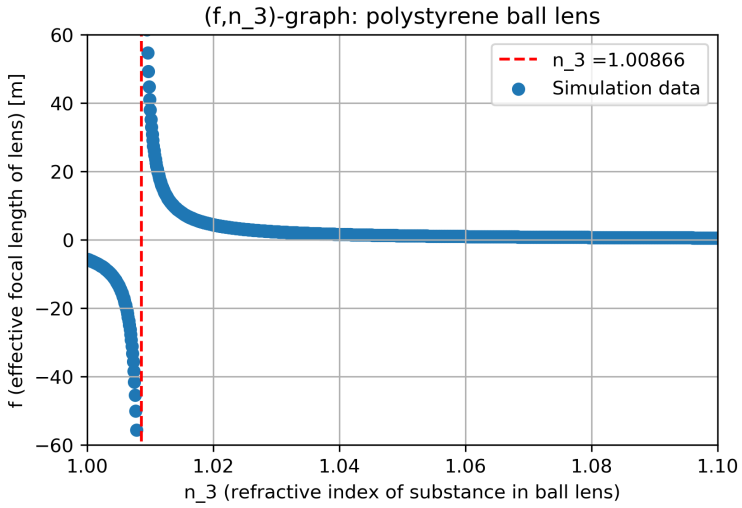


Figure 3.3: *The focal length f of the polystyrene ball lens for different values of n_3, the refractive index of the substance in the ball lens.*

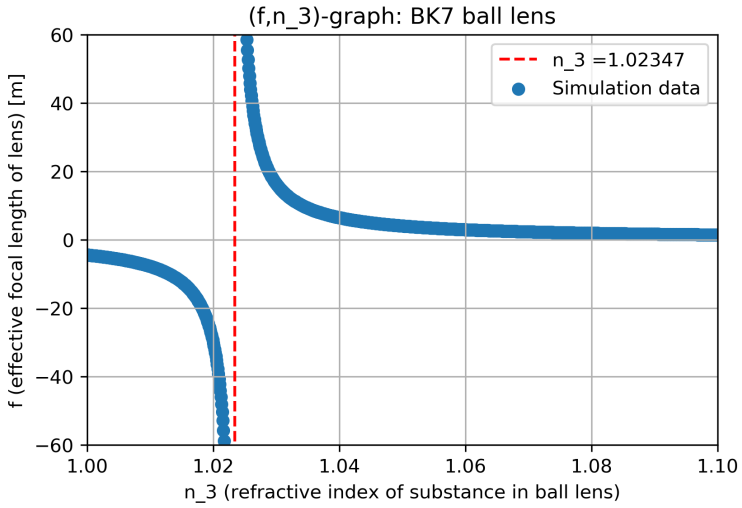


Figure 3.4: *The focal length f of the BK7 ball lens for different values of n_3, the refractive index of the substance in the ball lens.*

At values for n_3 where f goes to $\pm\infty$, paraxial rays entering the ball lens are approximately parallel to the z-axis when exiting the ball lens.

It is also visible that the curve flattens faster for the polystyrene ball lens than for the BK7 ball lens. This can be explained by the fact that the BK7 ball lens has a bigger radius; as seen in Eq. 2.17, for a homogenous ball lens, the focal distance scales linearly with the diameter of the ball lens. To prove that this is also the case for nonhomogenous ball lenses, a simulation has been made where the size of the outer radius of the polystyrene ball lens varies between 0 and 1 m. The inner radius is scaled up by a scaling constant, equal to the actual outer radius of the polystyrene ball lens divided by its actual inner radius. Three cases for n_3 have been plotted. The result of the simulation can be found in Fig. 3.5. This graph shows that the focal distance scales linearly with the radius of the ball lens, not only for a homogenous ball lens (this has been shown in Eq. 2.17), but also for a filled spherical shell that functions as a converging lens.

Futhermore, the scaling constant for the inner radius of the polystyrene ball lens is almost the same as that for the inner radius of the BK7 ball lens:

$$\text{BK7 ball lens}: \quad R_b/R_s = 0.966$$

$$\text{Polystyrene ball lens}: \quad R_b/R_s = 0.977$$

Thus, in approximation, one could say that the BK7 ball lens is a scaled-up version of the polystyrene ball lens, with the substance the shell is made of as the only difference.

The implication of these two findings is that the focal lengths for varying substances of n_3 inside the (bigger) BK7 ball lens will be relatively big compared to those of the (smaller) polystyrene ball lens. For the BK7 lens, one thus has to use substances of relatively high n_3 to get the same focal length as the polystyrene ball lens.

3.1.1.4 Final choice of ball lens

The polystyrene ball lens in the end became the lens both the experiment described in chapter 4 and that described in chapter 5 would be done with. The two halves were glued together using epoxy glue, and two holes were drilled in the shell, to enable the supply and drainage of water (see Fig. 3.6).

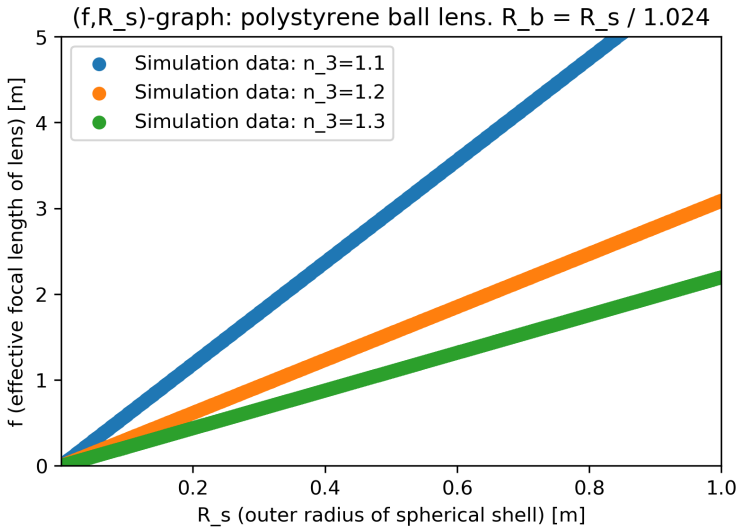


Figure 3.5: *The focal length f of the polystyrene ball lens for different values of R_s, the outer radius of the spherical shell, for three different cases of n_3, the refractive index of the substance in the ball lens. R_b, the inner radius of the spherical shell, is scaled by a scaling constant equal to* 1.024*.*

Figure 3.6: *The glued-together polystyrene ball lens, filled with water.*

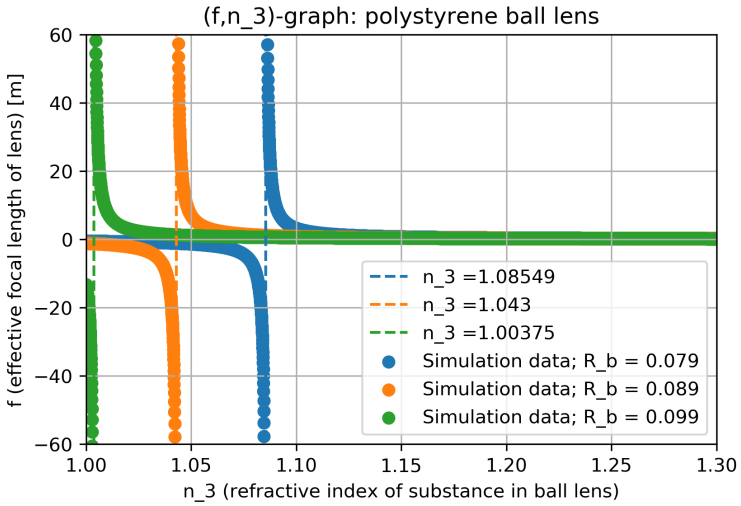


Figure 3.7: *The focal length f of the polystyrene ball lens for different values of R_b, the inner radius of the spherical shell, for three different cases of n_3, the refractive index of the substance in the ball lens. R_s, the outer radius of the spherical shell, is kept at a constant value of* 0.1 *m.*

3.1.2 Consideration of an aquarium as element of compound lens

As was shown in Section 3.1.1.1 and Fig. 2.10, the empty sphere shell acts as a diverging lens. Thus, to make it a converging lens, a high-index substance is needed to fill the spherical shell with. In Fig. 3.3, it is visible that the substance should have a refractive index of approximately 1.02 in order to be able to perform small-scale long distance imaging experiments ($f = 1.5$ m and such). Unfortunately, there were no safe gases available that had a refractive index that matched this value. Another solution would be to make the shell thinner: in Fig. 3.7, it is visible that the shell thickness scales approximately linearly with the position of the asymptote. However, making the shell thinner was not a practical solution for this short project. Therefore, the decision was made to "switch to liquids" and achieve the desired fraction of $\frac{n_3}{n_1}$ not by using some gas for n_3 but by putting the ball in an aquarium filled with pure water, thus making $n_1 = 1.33$. A new (f, n_3)-graph was made that accounted for this situation (see Fig. 3.8). The solution the ball would be filled with now had to have a refractive index in the range of $1.378 - 1.382$ in order to achieve small focal lengths ($f = 1.5\,\mathrm{m} \pm 0.05\,\mathrm{m}$). This is confirmed in Fig. 3.9.

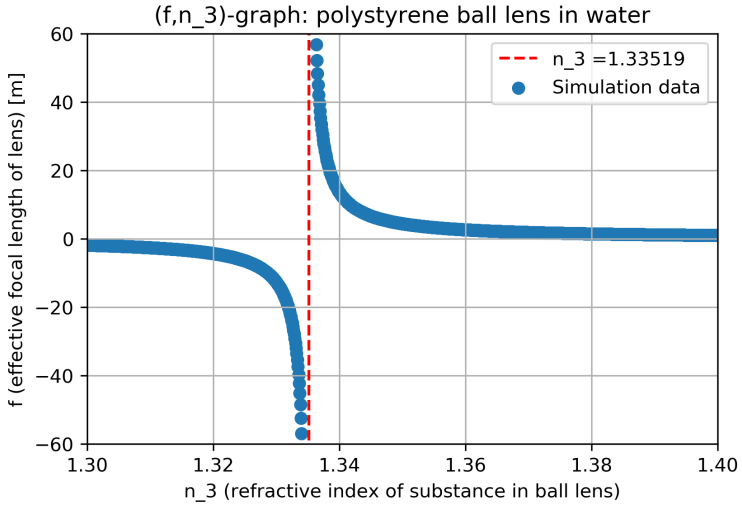


Figure 3.8: *The focal length f of the polystyrene ball lens in water ($n_1 = 1.33$), for different values of n_3, the refractive index of the substance in the ball lens. R_b, the inner radius of the spherical shell, is scaled by a scaling constant of* 1.024*.*

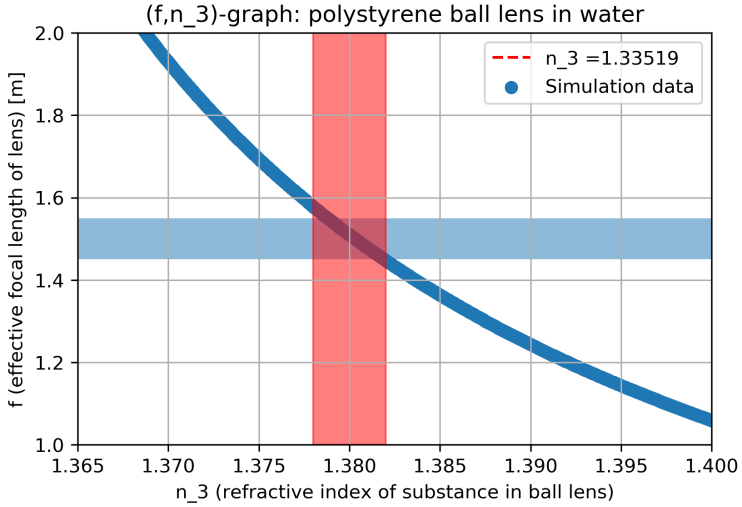


Figure 3.9: *Detail of the graph in Fig. 3.8, values of f ranging from* 1.45 *to* 1.55 *m. are marked in blue, the corresponding values for n_3, lying between* 1.378 *and* 1.382*, are marked in red.*

Figure 3.10: *The polystyrene ball lens, here filled with sugar solution, inside the aquarium.*

3.1.2.1 Choice of aquarium

An aquarium of 25x25x40 cm was bought and filled with pure water, and the polystyrene ball was put inside in order to create the compound lens shown in Fig. 3.10.

3.1.2.2 Simulating an aquarium

In order to incorporate an aquarium in the simulation, the refractions circled in yellow in Fig. 3.11 were added to the simulation. These refractions were easy to calculate, as the angle the incoming ray made with the z-axis could immediately be plugged into Eq. 2.1 as θ_{in}, as well as the refractive indices $n_1 = 1.0$ and $n_{aquarium} = 1.33$ to get θ_{out} which was the new angle the ray made with the z-axis. The air-aquarium refraction locations were taken to be -0.2 m and 0.2 m respectively, which meant the approximation was made that the ball was located in the middle of the aquarium. The position of the aquarium on the y-axis was between -0.1 m and 0.1 m.

An example simulation can be found in Fig. 3.12, where the properties of the polystyrene ball filled with a substance of $n_3 = 1.38$ have been simulated. A Huygens source is, in this case, located 4.0 m from the middle of the ball lens.

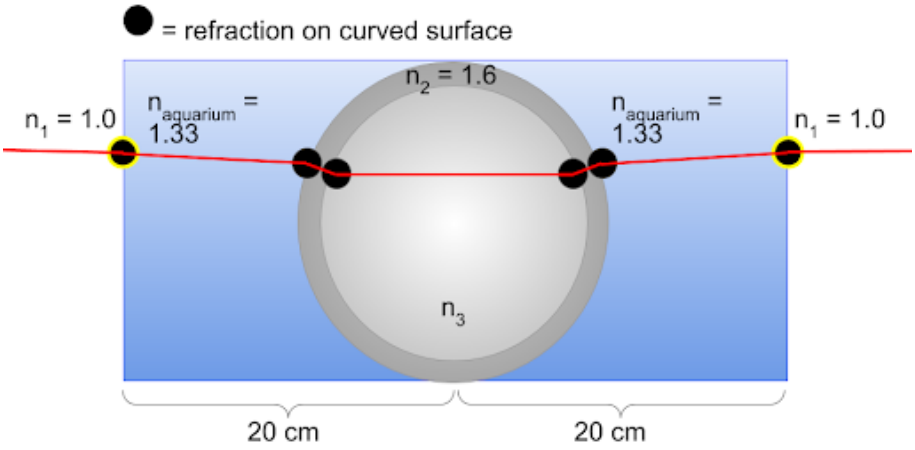


Figure 3.11: *Schematic overview of trajectory of ray in aquarium.*

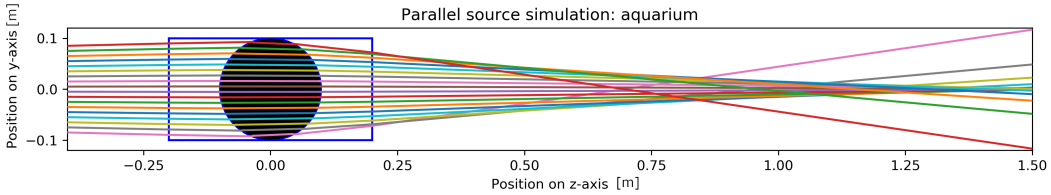


Figure 3.12: *Simulation of trajectory of rays through the compound lens.*

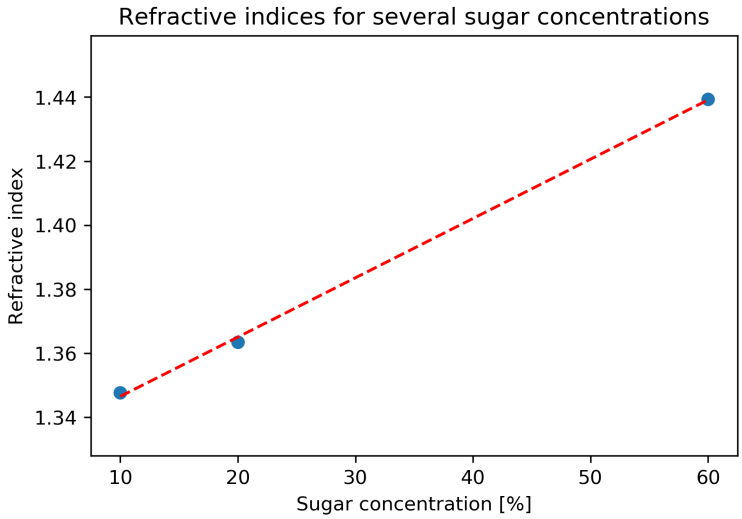


Figure 3.13: *A few sugar concentrations and their corresponding refractive indices [18]. The linear fit has the equation* $n = 0.0019\ C + 1.3280$*, with* n *the refractive index and* C *the sugar concentration in mass-percent (%).*

3.2 Consideration of different solutions as ball lens filling

For the experiment which will be discussed in chapter 5, a compound lens is needed which has a focal distance of around 1.5 m. As seen in Section 3.1.2, the refractive index of the substance inside the ball lens that corresponds with such a focal distance lies between 1.378 and 1.382. To find a substance with such a refractive index, two different methods will be considered: using a sugar solution and a salt solution.

3.2.1 Imaging using sugar solution

In Fig. 3.13, it becomes clear that a sugar solution can be prepared to obtain the desired refractive index for the small-scale long-distance imaging experiment. The refractive index of the solution grows linearly with the sugar concentration, following the expression stated in the caption. Using this expression, the sugar concentration can be directly plotted against the focal length. This is done in Fig. 3.14.

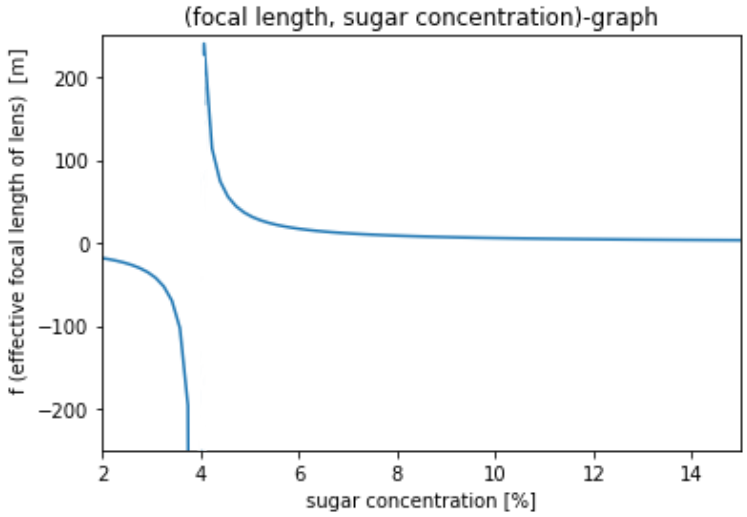


Figure 3.14: *A graph of the focal length of the polystyrene ball lens in water, filled with a sugar solution of varying sugar concentration. The point where the lens is neither converging nor diverging, marked by the characteristic asymptote, is situated at a sugar concentration of* 4%.

At first, the long-distance imaging experiment was thought to be done in a long hallway, thus allowing for focal lengths of around 5.00 m. Thus, the amount of sugar that needed to be dissolved was estimated for this focal length. The result of this can be found in Fig. 3.15.

This process was executed and the result, at first, was a clear sugar solution in the ball, enabling for sharp imaging inside the aquarium filled with pure water. Yet, cases began occurring where the sugar solution got cloudy in the ball lens. This process went according to the steps outlined below.

1. There was no problem with the imaging quality of the ball or solution, until the ball was accidentally dropped, causing a small crack in the shell. This crack was then sealed using 5-minute epoxy glue of the brand Griffon.

2. After this, the sugar solution inside the ball started to get cloudy each time it was left overnight or over the weekend. White big flakes were visible in the cloudy solution.

3. At some point, brown flakes had also appeared in the ball, they could already be seen just when a new sugar solution was poured into the

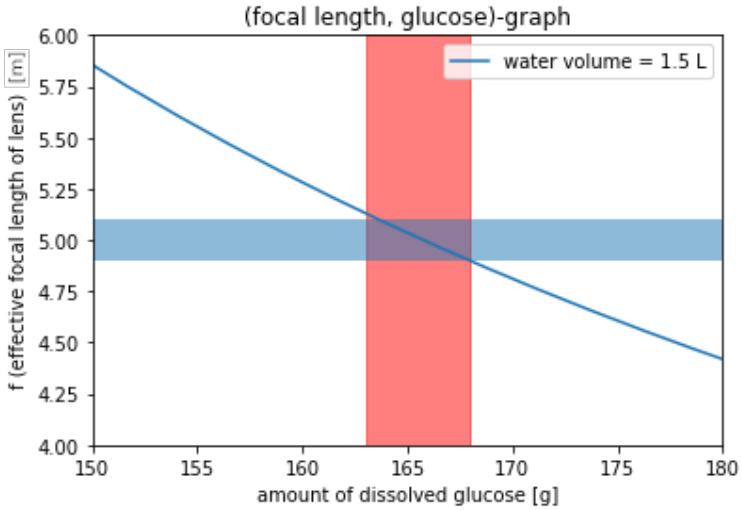


Figure 3.15: *Detail of the focal length, glucose-graph that was used to determine the amount of sugar that should be dissolved to create the right solution. The full volume of the ball was filled in two separate servings of 1.5 L sugar solution.*

ball. Part of the newly prepared solution was then kept in the bottle that was used for mixing and pouring the solution, in order to investigate the cause of the problem.

4. The solution in the ball got cloudy again overnight, and also, the solution in the ball smelled musty. It would, therefore, be possible that some form of microbial life had developed on the inner shell wall and feeded itself off of the sugar in the solution. However, the solution in the mixing bottle had stayed clear, which opposes this hypothesis. This clear sample sugar solution was then poured into a glass bottle and a bit of the hardened epoxy that had been used for fixing the small crack in the ball lens was put into this glass bottle, to investigate whether the solution would get cloudy due to being exposed to this kind of epoxy.

5. The next morning, the solution in the glass bottle had indeed begun to get cloudy. This solution got more and more cloudy as time passed. Thus, it is probable that the used 5-minute epoxy was (partly) responsible for the sugar solution getting cloudy. Some reaction could for example have taken place between the sugar and the epoxy whereby new polymers were formed.

The aquarium water had in the meantime also gotten cloudy; it could be

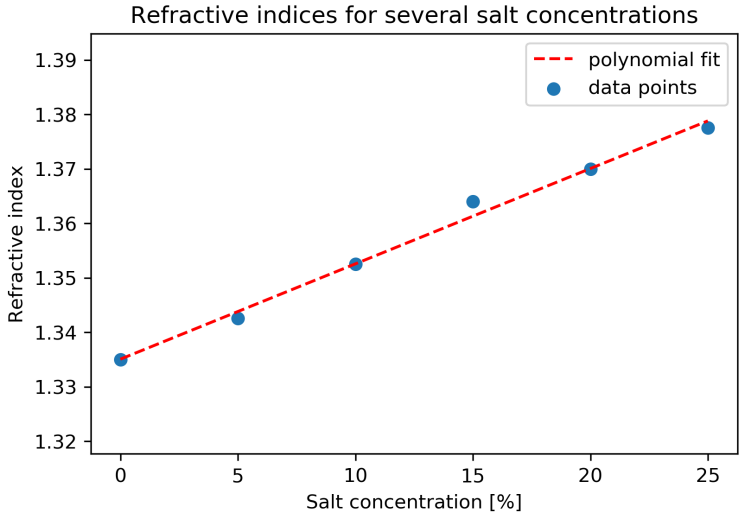


Figure 3.16: *A few (NaCl) salt concentrations and their corresponding refractive indices [19]. The linear fit has the Equation* $n = 0.0017\ \mathrm{C} + 1.3351$ *, with* n *the refractive index and* C *the sugar concentration in mass-percent (%).*

possible that this is due to some bacteria or fungi infiltrating and feeding off of some small amount of sugar solution that must have somehow gotten into the aquarium water, for example via the crack in the ball that was glued shut with epoxy. This suspicion is based on what was found when the aquarium was emptied completely: a smelly, slushy residue was left at the bottom of the aquarium.

In the end, it was decided to discard sugar solutions as a substance to use in the ball lens for this imaging experiment.

3.2.2 Imaging using salt solution

In Fig. 3.16, it becomes clear that a sodium chloride (NaCl) salt solution can also be prepared to obtain the desired refractive index for the small-scale long-distance imaging experiment. The refractive index of the solution grows approximately linearly with the salt concentration, following the expression stated in the caption. Using this expression, the salt concentration can be directly plotted against the focal length. This is done in Fig. 3.17.

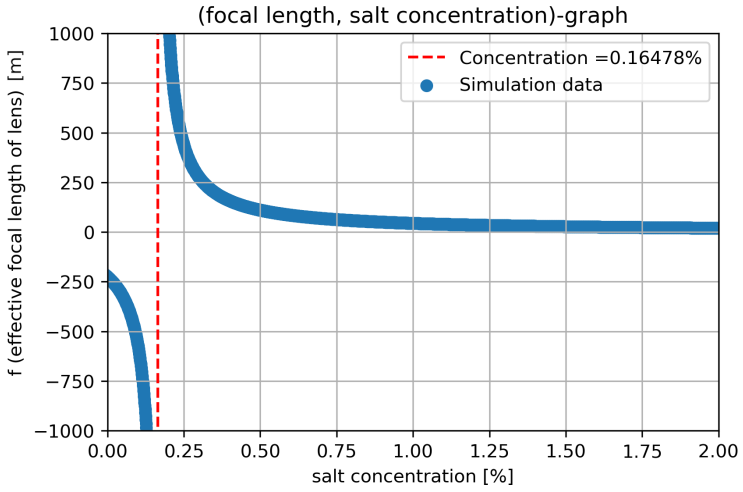


Figure 3.17: *A graph of the focal length of the polystyrene ball lens in water, filled with a sugar solution of varying sugar concentration. The point where the lens is neither converging nor diverging, marked by the characteristic asymptote, is situated at a sugar concentration of* 0.165%.

The suitable salt concentrations in order to achieve refractive indices varying from 1.45 to 1.55 turned out to range from 24.75% to 26.25%, as can be seen in Fig. 3.18.

At room temperature, one liter of water can dissolve about 357 g of salt, which is equal to a concentration of 26.3% [20]. The aim in the experiment was to reach a concentration of 25%.

Considering this information, it was decided to use salt to create a solution and fill the space encapsulated by the spherical shell with this instead.

3.3 Consideration of apertures

As can be seen in Fig. 3.12, The focal point of the compound (ball lens in aquarium) lens is not very clearly bound to a single spot. However, this sharpness of the focus can be improved by adding an aperture before the compound lens (that is, before the first aquarium wall). By using an aperture, a part of the spherically aberrated rays that do not travel neatly through the focus can be excluded from the image. Fig. 3.19 shows a histogram of the transversal coordinates of rays at the focal plane of the

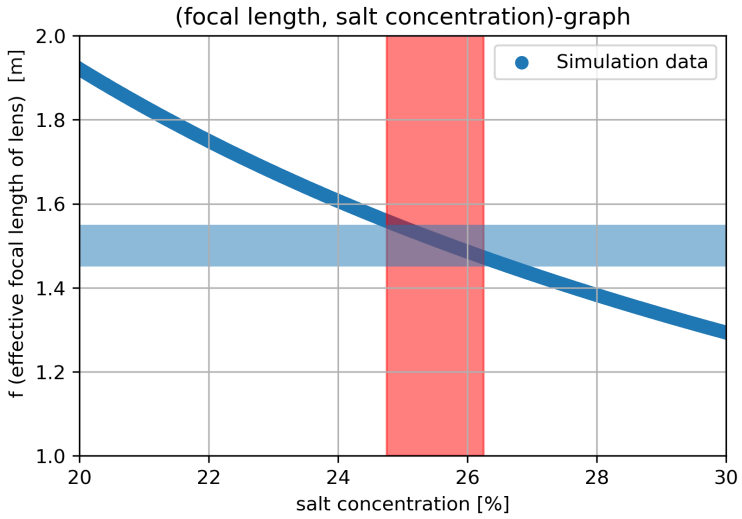


Figure 3.18: *Detail of Fig. 3.17 that was used to determine the amount of salt that should be dissolved to create the right solution.*

lens, after they have passed through an aperture of variable size and the filled ball lens. The effective focal length was determined by using Eq. 3.15.

Note that the individual histograms have been normalized, but with different bin sizes.

The width of the gaussian fit on each of the histograms is depicted by the transparent box on each of the fit curves. In the end, the aperture with a 10 cm diameter was used for the experiment discussed in chapter 5. It was placed in front of the aquarium wall. This aperture caused an image made by the lens of a beam of parallel rays to have a maximum width of 1.39 cm on the imaging screen.

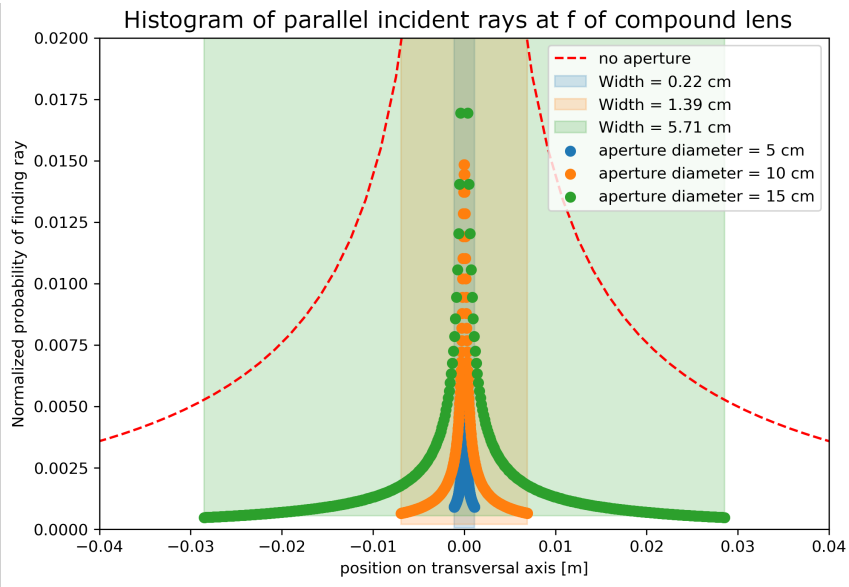


Figure 3.19: *Comparison of the effect of three different apertures on the width of the image of a parallel source at the effective focal point of the lens.*

Chapter 4

First experiment: imaging through empty polystyrene ball lens

In order to test the quality of the simulation and determine the order of magnitude of the mean squared error of the simulation, the following experiment was conducted. Two photos were made of a checkerboard, one with and one without the polystyrene ball lens between the camera and the checkerboard. The pixel locations of the grid points on the checkerboard were then recorded using photo editing software. A digital version of the grid was then plotted by importing these pixel locations in an iPython Notebook.

An equivalent of this experimental "grid point plot" was made using the ray-tracing simulation discussed in Chapter 1. Then, after having obtained two experimental grids and two simulated equivalents, the respective grids were fitted onto each other, and the mean squared error of the location of the grid points was calculated.

4.1 Experiment in real space

In order to assess the displacement of the locations of the grid points in real space due to the lens, a photo has to be taken by a camera. In this experiment, a webcam was used for this. Webcams have a CCD chip on

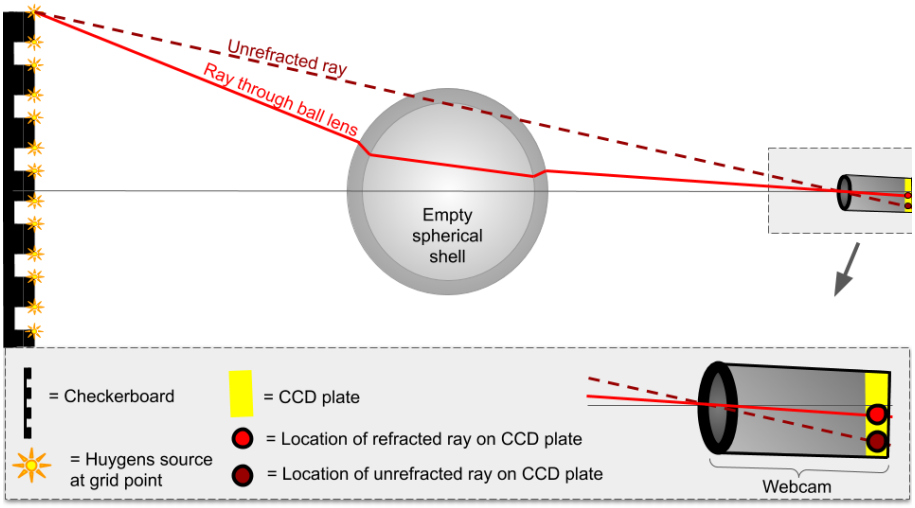


Figure 4.1: *Schematic overview of the first experiment.*

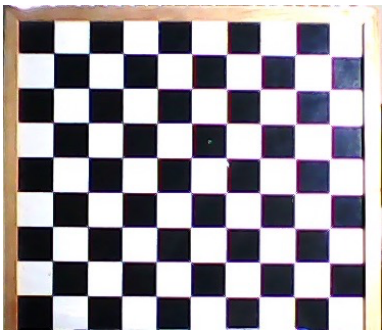


Figure 4.2: *Resulting photo of experiment 1. In this case, there is no lens between the checkerboard and the camera.*

which all rays passing through the camera lens fall. This is where the resulting photo is formed.

It is important to align the checkerboard, ball lens and camera. A schematic overview of the experiment can be found in Fig. 4.1.

The checkers on the checkerboard used had a width and height of 3.5 cm. The distance between the checkerboard and the lens was 59 cm. The distance between the lens and the camera was 72 cm. The resulting photos can be found in Fig. 4.2 and Fig. 4.3.

These photos were then opened in a photo editing programme to trace the pixel locations (x- and y-coordinates) of each of the gridpoints. In the case

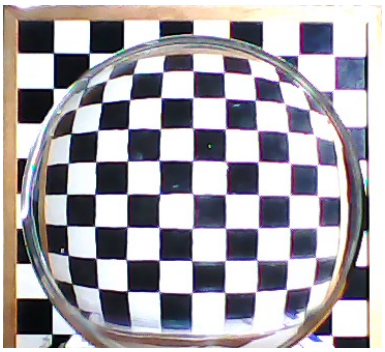


Figure 4.3: *Resulting photo of experiment 1. In this case, the empty polystyrene ball lens is in front of the checkerboard.*

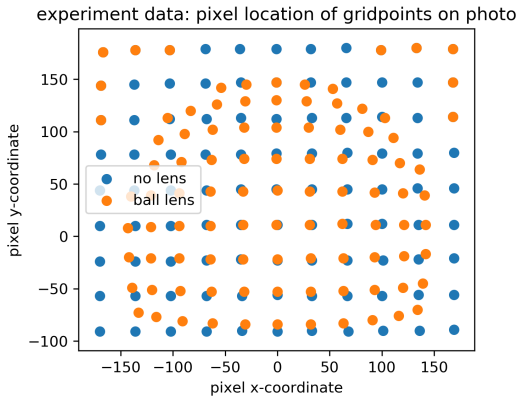


Figure 4.4: *Resulting scatter plot of the grid point locations recorded from Fig. 4.2 and Fig. 4.3.*

of this project "Gimp" was the photo editing programme used for this. Finally, the pixel locations had to be visualised in a scatter plot where each grid point is a separate dot located at the location of its pixel coordinates, relative to the origin, which is taken to be the middlemost pixel of the photo of the checkerboard. The scatter plot can be found in Fig. 4.4.

4.2 Simulated experiment

In order to simulate the experiment, all grid points were (two-dimensionally) simulated in the plane coinciding with the checkerboard location. The offset of the ball relative to the checkerboard was taken into account as well;

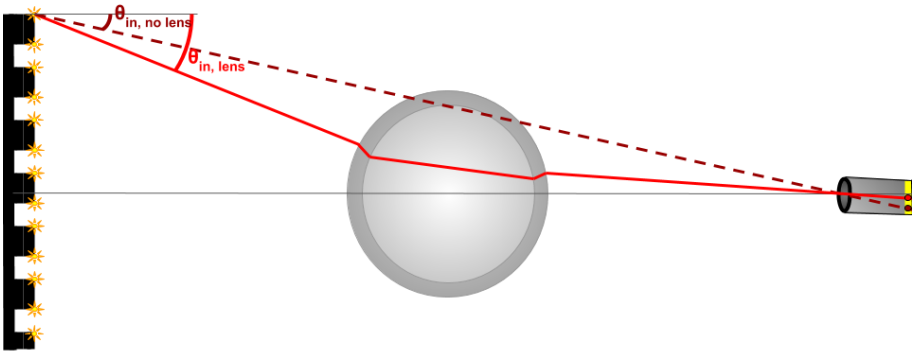


Figure 4.5: *Location of the two incoming angles used to simulate the rays going through the ball lens ($\theta_{in,lens}$) and the rays not going through the ball lens($\theta_{in,\,no\,lens}$).*

in Fig. 4.3, it can be seen that the ball has a small offset to the right and to the lower checkerboard border.

Now that the grid points had been correctly simulated, each of the grid points was separately taken as a Huygens source in the ray-tracing simulation that was designed for this project. The distance between the grid point and the center of the checkerboard was taken as the transversal coordinate (the y-axis in the ray-tracing simulation).

The right incoming angle of the ray emerging from the Huygens source (grid point) needed to be found; for the ray passing through the ball lens, this is the incoming angle corresponding to the certain ray trajectory that goes first through the ball lens, and afterwards, through the lens of the camera. For the unrefracted ray this angle corresponds simply to the angle the ray that directly hits the camera makes with the z-axis. A schematic overview of these two angles is given in Fig. 4.5.

When $\theta_{in,lens}$ and $\theta_{in,no\,lens}$ had been found, the location where the refracted and unrefracted rays hit the CCD plate was calculated (the location of the CCD plate was taken to be 55 mm from the lens of the camera). For this, the one-dimensional transversal coordinate was translated back to a two-dimensional location using the angle the original grid point made with the center of the checkerboard.

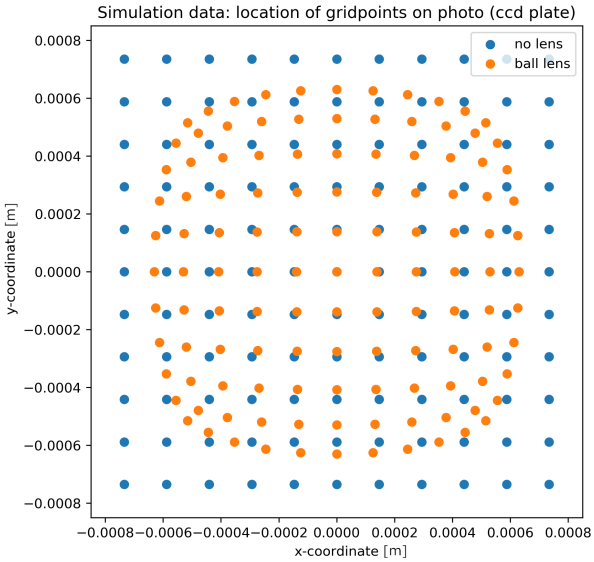


Figure 4.6: *Locations of the grid points on the CCD plate; the orange grid points have travelled through the ball lens, the blue grid points have not.*

Finally, all locations of the grid points on the CCD plate were plotted in a single graph; this graph can be found in Fig. 4.6.

4.3 Comparison

In order to compare the simulation results to the experimental results, the simulated grid without lens was firstly linearly fitted onto the experimental grid (see Fig. 4.7). The found fit parameters were also used to fit the simulated grid with lens over its experimental counterpart (see Fig. 4.8). The simulated and experimental positions of the grid points on the unrefracted grid (see Fig. 4.7) were found to coincide within numerical accuracy.

For the refracted grid (see Fig. 4.8), the uppermost row and the outermost columns of grid points were discarded in the comparison, in order to compare the two grids effectively; in the experiment, some of the grid points in the upper corners of the checkerboard had been measured "outside" of the ball lens, which made it difficult to compare the entire experimentally found grid with the simulated grid. The resulting grid can be found in Fig. 4.9. The mean squared error for the x-coordinates of the refracted grid

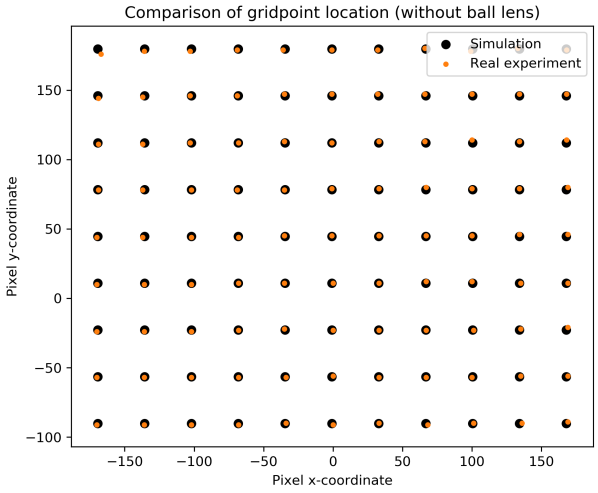


Figure 4.7: *Comparison of the measured and (fitted) simulated grid points, in the situation where there is no lens between the checkerboard and the camera.*

points was found to be equal to 3.1 pixel^2, and the mean squared error for the y-coordinates of the refracted grid points was found to be 3.7 pixel^2. The mean squared error is relatively low, which proves that the simulation accurately predicts the refraction of rays through the ball lens.

4.4 Discussion of experiment

A few points of discussion regarding the experiment suggest themselves:

- The polystyrene ball is made up of two loose ends, which are connected by tightening the two sides onto each other. this might cause the unexpected displacement of the outermost gridpoints.
- Also, the camera quality was not optimal, making it hard to locate the exact pixels corresponding with the gridpoints.
- It was especially hard to locate the pixel locations of the outermost grid points that were refracted through the lens, as the checkers were heavily bent at these points. This can be seen clearly in Fig. 4.3.

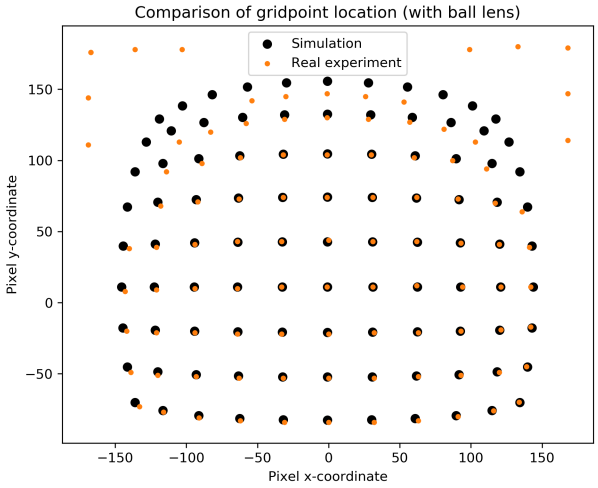


Figure 4.8: *Comparison of the measured and (fitted) simulated grid points, in the situation where there is a ball lens between the checkerboard and the camera. In the experiment, some of the grid points in the upper corners of the checkerboard had been measured "outside" of the ball lens; hence the inconstant location of the orange grid points in the upper corners.*

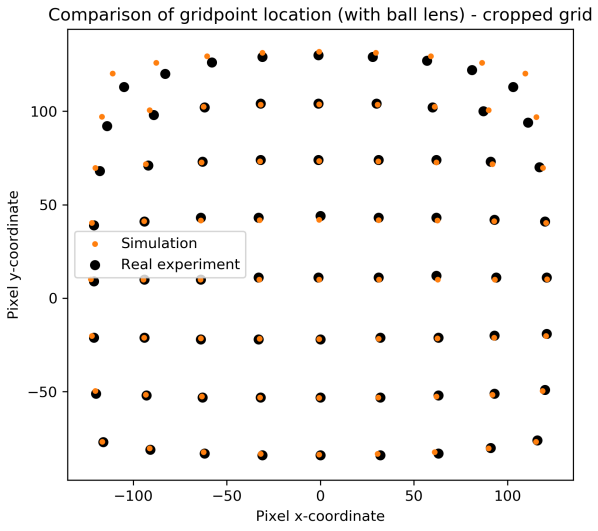


Figure 4.9: *Comparison of the measured and (fitted) simulated grid points, in the situation where there is a ball lens between the checkerboard and the camera. Cropped version of Fig. 4.8; the uppermost row and the two outermost columns have been removed, in order to make a realistic estimation of the mean squared error.*

Chapter 5

Second experiment: long-distance imaging through filled polystyrene ball lens in aquarium

In order to assess what the quality of the compound lens is, we calculate its diffraction limit by solving Eq. 1.1 for this lens. The half opening angle θ, for this lens, is equal to the arctangent of the aperture radius $R_{aperture} =$ 0.05 m divided by $2f - d_{aquarium}$. This is visualised in Fig. 5.1.

The expected focal length of the lens is 1.5 m. The minimum and maximum wavelengths of visible light are approximately 400 nm and 600 nm respectively [21]. With the help of this data, the resolution limit for "$2f$-$2f$-imaging" (see Fig. 5.1) was estimated to be 11.2 μm for light of 400 nm, and 16.8 μm for light of 600 nm.

In order to test the lens quality experimentally, a graphical form of the Rayleigh criterion was used to assess the results of an imaging experiment. Namely, according to the Rayleigh criterion, two images are just resolvable when the center of the diffraction pattern of one is directly over the first minimum of the diffraction pattern of the other [22]. The first and second diffraction pattern intersect at a height that is equal to half of each of the maxima.

Thus, when an image is made by the compound lens of two light sources that are a certain distance apart from each other, an intensity plot can give information on whether the diffraction limit of the lens has been reached.

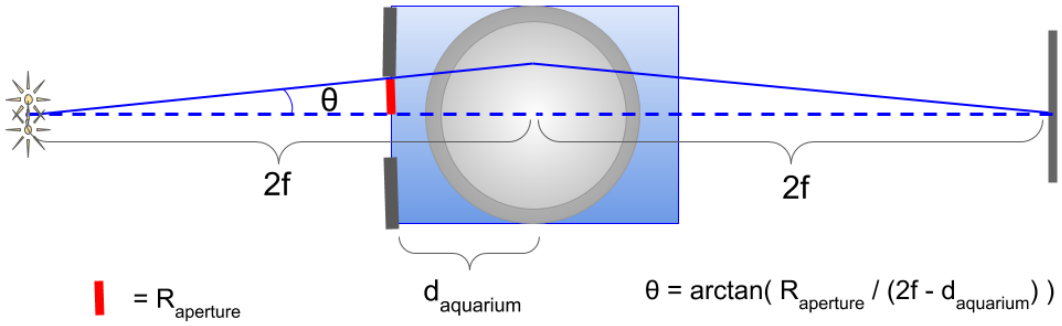


Figure 5.1: *Visualisation of the "half opening angle"* $\theta = \arctan\left(\frac{R_{aperture}}{2f - d_{aquarium}}\right)$ *of the ball lens for imaging a source located at a distance of* $2f$ *before the ball lens on an image plane located at a distance of* $2f$ *behind the ball lens.*

5.1 Experimental procedure

First, a salt solution of 25% salt was prepared. As seen in chapter 3, this concentration of salt in solution corresponds to a refractive index of 1.38. This solution was then put into the polystyrene ball lens. An aquarium of 25x25x40 cm was filled with demineralized water and the filled ball lens. An aperture with a diameter of 10 cm was put in front of the aquarium wall. A flashlight consisting of a strip of LED lights was used as the light source. Two small holes were pricked into a piece of aluminium foil using the sharp point of a pair of compasses, one sheet of aluminium foil contained two small holes that were 3 mm apart, and one sheet of foil contained two small holes that were 6 mm apart. These two aluminium foil pieces were alternately wrapped around the strip of light in order to create two separate sources of light, at 3 and 6 mm spacing. This light source was placed at 3 m distance of the compound lens, corresponding to a length of two times the estimated focal distance of the compound lens. On the other side of the compound lens, at 3 m distance, a briefcase stood upright, with a sheet of paper attached to it. This sheet of paper was used as the imaging plane. A schematic overview of the experiment can be found in Fig. 5.1.

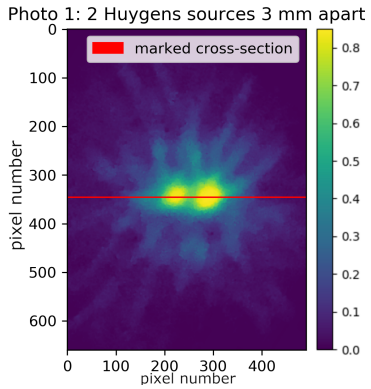


Figure 5.2: *Photo of the imaged two sources at 3 mm distance.*

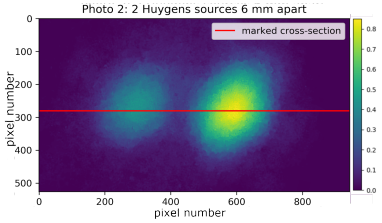


Figure 5.3: *Photo of the imaged two sources at 6 mm distance.*

5.2 Results

The found focal length differed slightly from the calculated focal length: a focal length of 1.43 m was found. The resulting photos of the illuminated imaging screen can be found in Fig. 5.3 and Fig. 5.2.

Fig. 5.4 shows an intensity plot of the photo of the image of the sources at 6 m spacing. In this plot, it is visible that the resolution limit has not yet been reached, as the intersection point of the gaussian fits of the two sources is lower than the half maximum of each of the gaussians. In Fig. 5.5, the intensity plot of the photo of the image of the sources at 3 mm, it is visible that the sources are not completely resolvable; the intersection point of the gaussian fits of the two sources is higher than the half maximum of each of the gaussians.

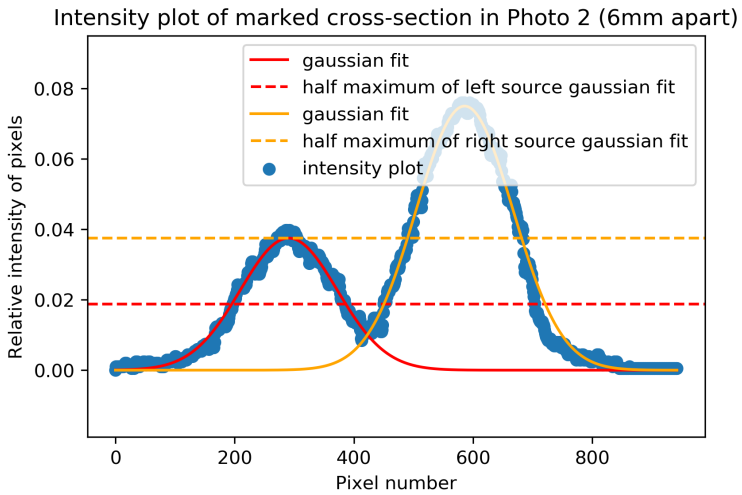


Figure 5.4: *Intensity plot of the imaged two sources at* 6 *mm distance.*

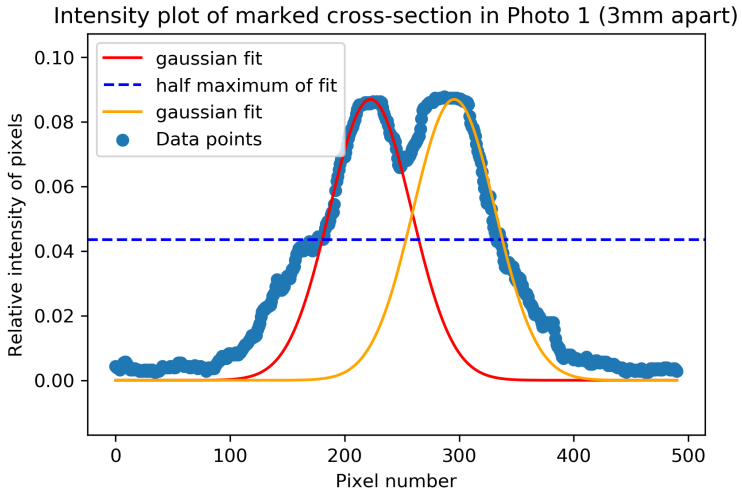


Figure 5.5: *Intensity plot of the imaged two sources at* 3 *mm distance.*

Chapter 6

Discussion

In this thesis, the idea of using a ball lens with shell as a helioscope has been explored. The possibility of imaging sunspots with this lens was assessed.

In the introduction, it was shown that a ball lens with a radius of 0.5 m could image sunspots that were over 87 km in diameter. If we calculate the resolution limit of the polystyrene ball lens via Eq. 1.1, plugging in its radius of 0.1 m and a distance of 1 astronomical unit into 1.2 to find the right half opening angle, and choosing $\lambda = 580$ nm (yellow light) and $n = 1.0$ (vacuum of empty space), we find the minimum diameter of a sunspot it could image, which is 433.83 km. Imaging a medium-sized sunspot with the polystyrene ball lens should thus in theory definitely be possible.

Yet, as shown in Chapter 5, the resolution limit found experimentally for $2f$-$2f$-imaging is a lot bigger than the value found through theory. Thus, the found image is probably highly influenced by lens errors, and does not give accurate information on the diffraction limit of the lens.

The most plausible explanation for the lens errors is that the water might have become a bit more cloudy in the period that the experimental setup was completely ready, but the experiment was not conducted yet; this period lasted for four weeks. Another plausible cause of the lens errors is that the surface of the polystyrene ball was not perfectly round, as well as slightly bruised in some areas, which might also impact the overall lens resolution.

Overall, the ball lens with shell was assessed as a device suitable for imaging the sun, however, the clarity and durability of the substance the ball lens is to be filled with should be taken into consideration when building a helioscope using a ball lens with shell, as this may cause lens errors that are very large and that might gravely blur an image.

Chapter 7

Acknowledgements

On a final note, I would like to thank some people that made invaluable contributions to my research during these past months. First of all, I would like to thank my primary supervisor, Dr. Wolfgang Löffler, for all his support during my months of research. My investigation into the functionality of ball lenses could not have been possible without the guidance I received from him. Whether it concerned effective feedback on whether I was still on the right track in my calculations, suggestions on what materials might be most worth considering for experiments, or even disaster relief by retrieving the near-final version of my thesis from a crashed laptop of which only a very outdated backup was available, I could always count on him for help and advice.

I would also like to thank my second supervisor, Prof.dr. Michel A.G.J. Orrit, for his persistent support. He was always ready to offer various suggestions on which routes I could take next, practical advice on and supply of materials, and inspirational side investigations, such as the birefringence of the polystyrene shell and the possibility of making use of an aerogel as the ball lens filling.

Furthermore, I would like to thank Dorothea Samtleben and Rene de Boer, who work at NIKHEF on the KM3NeT project, for the honour of providing me with a BK7-glass ball lens, which was one of the glass balls that had previously been used as a part of the KM3NeT neutrino telescope at the bottom of the Mediterranean Sea. Also, I would like to express gratitude to Harmen van der Meer, who works at the Fine Mechanics Department

of Leiden University, for making the polystyrene ball lens into a single, workable unit by glueing the two halves together and inserting two tubes for water supply and drainage on opposite sides of the ball.

I would also very much like to thank Dr. Löffler and Prof.dr. Orrit for suggesting this project at the bachelor project fair; it was a project which was not quite in line with their main fields of research, but an idea they had developed on the side. This made it a bit different from most other projects, in a way that very much spoke to me. I was, and still am, very glad to have found it; it was a project through which I was able to combine my interests in both the natural sciences and the (optical) arts. This has been very inspiring to me. Furthermore, through the ray-tracing simulation part, the project gave me a chance to get some practical programming experience, and create my own research tool, which was something I had also hoped to find in a bachelor research project. I look back on the past months very fondly, and I think I will keep on doing so for a long time. This project was truly one of a kind!

Bibliography

[1] Wikipedia. Christoph Scheiner — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Christoph%20Scheiner&oldid=989531057 (2020). [Online; accessed 07-December-2020].

[2] Wikipedia. Helioscope — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Helioscope&oldid=969600110 (2020). [Online; accessed 07-December-2020].

[3] US Department of Commerce, N. The Sun and Sunspots. https://www.weather.gov/fsd/sunspots (2020). [Online; accessed 17-December-2020].

[4] Wikipedia. Sunspot — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Sunspot&oldid=992645098 (2020). [Online; accessed 07-December-2020].

[5] Wikipedia. Angular resolution — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Angular%20resolution&oldid=990691127 (2020). [Online; accessed 30-November-2020].

[6] Wikipedia. Sun — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Sun&oldid=991596684 (2020). [Online; accessed 03-December-2020].

[7] Wikipedia. Ray transfer matrix analysis — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Ray%20transfer%20matrix%20analysis&oldid=963876222 (2020). [Online; accessed 07-November-2020].

[8] Wikipedia. Snell's law — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Snell's%20law&oldid=987502304 (2020). [Online; accessed 07-November-2020].

[9] Weisstein, E. W. "Circle-Line Intersection." From MathWorld–A Wolfram Web Resource. https://mathworld.wolfram.com/Circle-LineIntersection.html (2020). [Online; accessed 07-November-2020].

[10] Wikipedia. Slope — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Slope&oldid=987405659 (2020). [Online; accessed 07-November-2020].

[11] Edmund Optics Ltd. Understanding Ball Lenses. https://www.edmundoptics.eu/knowledge-center/application-notes/optics/understanding-ball-lenses/ (2020). [Online; accessed 07-November-2020].

[12] Wikipedia. Virtual image — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Virtual%20image&oldid=991669387 (2020). [Online; accessed 17-December-2020].

[13] Elagha, H. A. Ray tracing in monocentric and ball lenses by a general exact formula. *J. Opt. Soc. Am. A* **36**, 1117 (2019).

[14] Wikipedia. Relay lens — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Relay%20lens&oldid=917456572 (2020). [Online; accessed 30-November-2020].

[15] Collaboration, K. KM3NeT. https://www.km3net.org/ (2020). [Online; accessed 17-December-2020].

[16] Archidelis. Archidelis: Polystyrene ball Ãž = 50 (2020).

[17] Wikipedia. Lens — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Lens&oldid=990954981 (2020). [Online; accessed 30-November-2020].

[18] Wikipedia. List of refractive indices — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=List%20of%20refractive%20indices&oldid=986790185 (2020). [Online; accessed 29-November-2020].

[19] Stupar, D. *et al.* Remote monitoring of water salinity by using side-polished fiber-optic U-shaped sensor. LS4c.4–1 (2012).

[20] Wikipedia. Saline water — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Saline%20water&oldid=990971936 (2020). [Online; accessed 30-November-2020].

[21] Wikipedia. Visible spectrum — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Visible%20spectrum&oldid=991542186 (2020). [Online; accessed 30-November-2020].

[22] Urone, P. P., Hinrichs, R., Dirks, K. & Sharma, M. *27. Wave Optics*, 1178 (OpenStax, 2012).

Chapter 8

Appendix 1

Listing 8.1: *The Python code describing the five main functions used in the ray-tracing simulation.*

```
# ray tracing through curved surface
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

def refract(inc_z, inc_y, inc_theta, R, n_1, n_2):
    #inc = incoming

    if abs(inc_y)>=R:
        return False    #the rays that only travel
           through the shell are not incorporated in
           this ray-tracing program
    else:
        alpha = -np.sign(inc_z)*np.arcsin(inc_y/R) +
           inc_theta            # alpha = angle of
           incident ray with respect to surface normal
        if abs((n_1/n_2)*np.sin(alpha)) > 1.0:
            return False

        else:
            beta  = np.arcsin((n_1/n_2)*np.sin(alpha))
                # calculated via snell's law
```


Version of January 5, 2021– Created January 5, 2021 - 11:41

```
            out_theta = beta + np.sign((inc_z))*np.
                arcsin(inc_y/R)
            return out_theta

def circle_line_intersection(inc_z,inc_y,R,theta):
        #two points on the line:
        z_in_1 = inc_z
        y_in_1 = inc_y
        y_in_2 = R*np.tan(theta)+inc_y-inc_z*np.tan(
            theta)
        z_in_2 = R

        #properties for the line-sphere intersection:
        d_z = z_in_2-z_in_1
        d_y = y_in_2-y_in_1
        d_r = np.sqrt((d_z)**2+(d_y)**2)
        D = z_in_1*y_in_2-z_in_2*y_in_1

        discr = R**2*d_r**2-D**2          #discriminant

        if discr <=0.0:
            return False    #the rays that only travel
                through the shell are not incorporated
                in this ray-tracing program

        else:
            #results of the circle-line intersection:
            z_1 = (D*d_y - d_z*np.sqrt(R**2*d_r**2-D
                **2))/d_r**2
            z_2 = (D*d_y + d_z*np.sqrt(R**2*d_r**2-D
                **2))/d_r**2
            y_1 = (-D*d_z + np.abs(d_y)*np.sqrt(R**2*
                d_r**2-D**2))/d_r**2
            y_2 = (-D*d_z - np.abs(d_y)*np.sqrt(R**2*
                d_r**2-D**2))/d_r**2

            #match the right result to the right point
                (point of incidence and point of
                departure)
            return (np.array([z_1,z_2]),np.array([y_1,
                y_2]))
```

```
def ball_with_shell(R_s,R_b,n_1,n_2,n_3,source_z,
    source_y,source_theta,stoppoint,raycolor):

    inc_theta = source_theta #angle of the emitted ray
        = incoming angle

    #finding the left intersection point of the ray
       with the outer shell
    if not circle_line_intersection(source_z,source_y,
       R_s,inc_theta):
        return
    zvalues_os,yvalues_os =circle_line_intersection(
       source_z,source_y,R_s,inc_theta)
    #match the right values for z and y
    inc_z = np.amin(zvalues_os)
    if inc_theta <0.0:
        inc_y = np.amax(yvalues_os)
    else:
        inc_y = np.amin(yvalues_os)

    #first refraction; outside to outer shell
       interface
    out_theta_s1 = refract(inc_z, inc_y, inc_theta,
       R_s, n_1, n_2) #outgoing angle after first
       shell refraction

    #if not out_theta_s1:
        #return
    #finding the intersection point of the ray with
       the inner ball (after first refraction)
    #ray position: left from lens center
    if not circle_line_intersection(inc_z,inc_y,R_b,
       out_theta_s1):
        return

    zvalues_sb,yvalues_sb = circle_line_intersection(
       inc_z,inc_y,R_b,out_theta_s1)  #intersection
       shell to ball (sb)
    #match the right values for z and y
```

```
inc_z_b1 = np.amin(zvalues_sb)  #incoming z-
    coordinate at inner ball (left from lens center
    )
if out_theta_s1 < 0.0:
    inc_y_b1 = np.amax(yvalues_sb) #incoming y-
        coordinate at inner ball (left from lens
        center)
else:
    inc_y_b1 = np.amin(yvalues_sb)

#second refraction; shell to ball interface
#ray position: left from lens center
out_theta_b = refract(inc_z_b1, inc_y_b1,
    out_theta_s1, R_b, n_2, n_3) #outgoing angle
    after ball refraction

#if not out_theta_b:
    #return
#finding the intersection point of the ray with
    the inner ball (after second refraction)
#ray position: right from lens center

if not circle_line_intersection(inc_z_b1,inc_y_b1,
    R_b,out_theta_b):
    return

zvalues_bb,yvalues_bb = circle_line_intersection(
    inc_z_b1,inc_y_b1,R_b,out_theta_b) #
    intersection ball to ball (bb)
#match the right values for z and y
inc_z_b2 = np.amax(zvalues_bb) #second incoming z-
    coordinate at inner ball (right from lens
    center)
if out_theta_b > 0.0:
    inc_y_b2 = np.amax(yvalues_bb) #second
        incoming y-coordinate at inner ball (right
        from lens center)
else:
    inc_y_b2 = np.amin(yvalues_bb)
```

```
#third refraction; ball to shell interface
#ray position: right from lens center
out_theta_s2 = refract(inc_z_b2, inc_y_b2,
   out_theta_b, R_b, n_3, n_2) #outgoing angle
   after second shell refraction

if inc_z_b2<0.0:    #the programme doesn't work
   properly for rays with a z-coordinate smaller
   than zero
                    #for the point where the ray
                       comes out of the inner ball
                        and enters the shell.
    return

#finding the intersection point of the ray with
   the outer shell (after third refraction)
#ray position: right from lens center
if not circle_line_intersection(inc_z_b2,inc_y_b2,
   R_s,out_theta_s2):
    return

zvalues_bs,yvalues_bs = circle_line_intersection(
   inc_z_b2,inc_y_b2,R_s,out_theta_s2) #
   intersection ball to shell (bs)
#match the right values for z and y
inc_z_s2 = np.amax(zvalues_bs) #second incoming z-
   coordinate at outer shell (right from lens
   center)

if abs(out_theta_s2)>np.pi/2.:
    inc_z_s2 = np.amin(zvalues_bs)
if out_theta_s2 > 0.0:
    inc_y_s2 = np.amax(yvalues_bs) #second
       incoming y-coordinate at outer shell (right
        from lens center)
else:
    inc_y_s2 = np.amin(yvalues_bs)

#fourth refraction; shell to outside interface
#ray position: right from lens center
```

```
    out_theta = refract(inc_z_s2, inc_y_s2,
       out_theta_s2, R_s, n_2, n_1) #outgoing angle
       after last refraction
    if inc_z_s2<0.0:     #the programme doesn't work
       properly for rays with a z-coordinate smaller
       than zero
                         #for the point where the ray
                            comes out of the shell and
                            enters the area outside the
                             lens.
        return

    return inc_z_s2, inc_y_s2, out_theta

def huygens_source(R_s,R_b,n_1,n_2,n_3,source_z,
   source_y,source_theta,stoppoint, R_h,aperture,
   density):

    # the function "ball_with_shell_plot"
       automatically picks out only those rays emitted
    # by a huygens source that have positive z-
       coordinates at the point of leaving
    # the inner ball to enter the outer shell (third
       refraction) and at the
    # point of leaving the outer shell to enter the
       outside area (fourth refraction).
    result_array = np.zeros((1,4))
    result = np.zeros((1,4))
    #/source_theta // out_z // out_y // out_theta /

    for source_theta in np.linspace(-np.arcsin(
       source_y/R_h)-aperture,-np.arcsin(source_y/R_h)
       +aperture,density):
        if ball_with_shell(R_s,R_b,n_1,n_2,n_3,
           source_z,source_y,source_theta,stoppoint,
           raycolor):
            out_z,out_y,out_theta = ball_with_shell(
               R_s,R_b,n_1,n_2,n_3,source_z,source_y,
               source_theta,stoppoint,raycolor)
            result[0,:] = np.array([source_theta,
               out_z,out_y,out_theta])
```

```
            result_array = np.append(result_array,
                result,axis = 0)

    return result_array[1:,:]

def parallel_source(R_s,R_b,n_1,n_2,n_3,source_z,
    source_y,source_theta,stoppoint, R_h,aperture,
    density):

    # the function "ball_with_shell_plot"
        automatically picks out only those rays emitted
    # by a huygens source that have positive z-
        coordinates at the point of leaving
    # the inner ball to enter the outer shell (third
        refraction) and at the
    # point of leaving the outer shell to enter the
        outside area (fourth refraction).
    result_array = np.zeros((1,4))
    result = np.zeros((1,4))
    source_theta = 0
    #/source_theta // out_z // out_y // out_theta /

    for source_y in np.linspace(-aperture,aperture,
        density):
        if ball_with_shell(R_s,R_b,n_1,n_2,n_3,
            source_z,source_y,source_theta,stoppoint,
            raycolor):
            out_z,out_y,out_theta = ball_with_shell(
                R_s,R_b,n_1,n_2,n_3,source_z,source_y,
                source_theta,stoppoint,raycolor)
            result[0,:] = np.array([source_y, out_z,
                out_y,out_theta])
            result_array = np.append(result_array,
                result,axis = 0)

    return result_array[1:,:]
```