Technical University of Denmark

DTU Space

National Space Institute

Bachelor Project

McXtrace simulation of x-ray telescope and mirror deformation

Toke Bue Bjørner s144111

June 16, 2017

Contents

Abstract

The topic of this thesis is ”McXtrace simulation of x-rat telescope and mirror deformation”. The overall aim of this project is provide a model capable of simulating a deformation on reflecting surfaces in the ray-tracing program McXtrace, for use mainly in optical apertures as those seen in x-ray telescopes such as the ATHENA satellite.

The underlying context is to provide an usable application to determine the effect of a level magnitude of deformation upon a high precision x-ray reflector module. To answer this question I present a simple algorithm to account for the effect of a large scale deformation and a similar more complicated algorithm, which rely on different intersection methods. The deformation is divided into 2 different scales, which are handled by separate routines.

In order to properly test the implemented algorithm I conduct extensive simulation using both models of the intermediate deformation using various reflective tables with different roughness values.

Following a successful implementation of the described algorithm, I simulate a more advanced instrument which approximates the shape of the ATHENA telescope in order to determine the effect of the deformation upon the light collecting efficiency of the optical module.

From this ATHENA model I determine that the photon distribution follows a linear pattern of cohesion with respect to the level of deformation, which is unaffected by the level of roughness.

Chapter 1

Introduction

Properly studying the cosmos requires the possibility to view the universe throughout the entire range of the electromagnetic spectrum. Designing instruments that are capable of operating in the high energy section of the electromagnetic spectrum is an arduous task.

1.1 X-ray telescopes

X-ray telescopes require a great amount of resources and development, and it is therefore essential for us to properly consider the possible flaws that might have an impact on the overall accuracy of the instrument before entering the construction phase.

A great deal of the precision of an x-ray telescope hinges on the accompanying optical module which focuses the incident x-ray photons onto the narrow diameter of the detector module.

This thesis will focus on the composition of the most popular variant, the Wolter type I model design, which consists of two series of circular con-focal mirrors mounted around a central spoke. The shape of the first series of mirrors is a parabolic shape and the second a hyperbolic shape. Photons incident on both surfaces will be directed to the focal point of the instrument which is separated from the mirrors at a distance known as the focal length.

To focus the higher energy x-ray photons it is necessary to employ a special sort of multilayered coating on the surface of the reflector (mirror) to match the specific wavelength of the photon and the incident grazing angle.

If we consider only the optical module we can determine several different possible errors that may impact the instrument: most important are the mirrors in the instrument. The Wolter I model requires that all mirrors are focused onto a single point and thus any deviation upon the surface of the mirrors will change the direction of the incident photons.

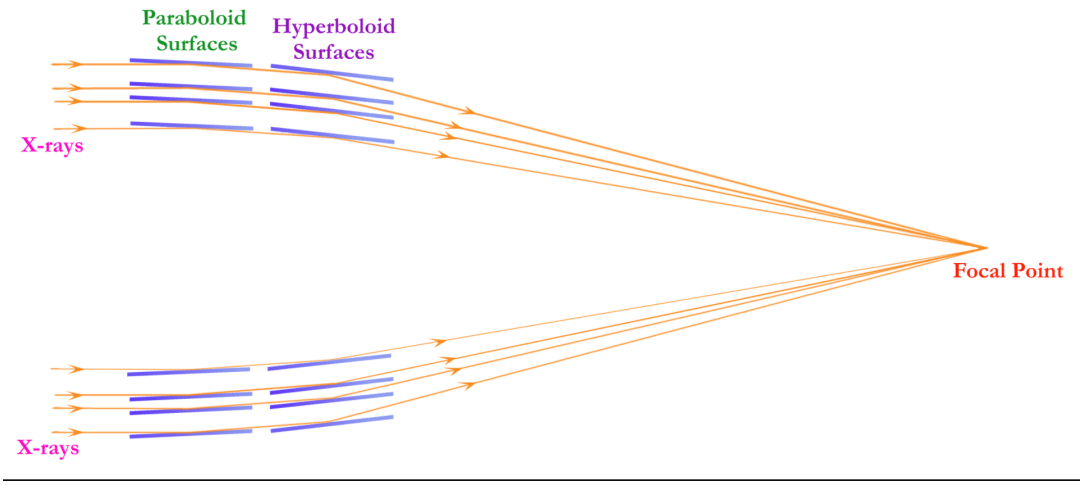

Figure 1.1: Wolter type I model, where parallel travellingphotons reflect off a set of paraboloid and hyperboloid mirrors towards a common focal point (1)

1.2 The ATHENA satellite

In this project I choose to work specifically with the ATHENA mission. The ATHENA satellite is an x-ray telescope which implements the commonly used model of the Wolter Type I geometry in order to focus the incident photons.

The accuracy of a x-ray optics determines the resolution of the telescope in general. Any inaccuracies in the reflected photons will transmit a blur to the viewed image which determines how efficiently the telescope can discriminate between different x-ray sources in stellar space.

1.2.1 Goals

The primary goal of the project is to build a stable model for the deformation aspect of the x-ray mirror which is both universally applicable and computationally efficient. This will allow it to be used for the myriad instrument configurations used to model the x-ray satellite ATHENA or similar projects. The second goal of the project is a thorough study of the results that are obtained by the use of the previously described model, I wish to investigate whether a relation between the reflection intensity and the level of deformation can be determined. Further I wish to study at which level of deformation a critical error margin for the instrument is reached, which can then be compared with figures for the actual deformation on the ATHENA optical module.

It is important to clarify that this project focuses on modelling the effect of perturbations only upon the model shape. Therefore, there will be no considerations of other focusing disruptive effects such as dirt materials on the lens or flaws inside the material volume. The analysis will be restricted to the surface interface.

In order to test this I will need to construct a detailed instrument space with controlled variables, which will approximate the expected shape of the ATHENA optical module in order to properly replicate the working parameters of the mission. The developed algorithm will be implemented in this instrument which simulates the geometry of the ATHENA telescope and in such a way that the direct scattering effect of the deformation is examined.

Chapter 2

Theory

In this chapter I will discuss the underlying theory behind the general x-ray photon propagation and the mechanics in x-ray reflection. In addition to this I will extrapolate the principal mechanics of the x-ray focusing telescope so that the governing parameters of the ATHENA telescope can be better understood. The ray tracing software is based on the laws of reflection and electromagnetism. Therefore the electromagnetic equations directly effects the functions of the software.

2.1 Photon scattering

The most fundamental structure is of course the photon, which can be described in several different terms. The direction of the photon is described as being perpendicular to the electric field E. When dealing with advanced x-ray interactions the photon will mainly be described in terms of the wave equation. The regular plane wave equation of a photon is as follows.(2)

$$E(r,t) = E_0 e^{-} i(\omega t - k * r) \tag{2.1}$$

In terms of quantum mechanics I would describe a monochromatic beam of electromagnetic radiation as being composed of photons which have an intrinsic energy $\hbar\omega$ with an velocity of $\hbar k$, with the intensity of the beam being described by the density of the photons contained in it, or rather the amount of photons passing through an area per unit of time. The relationship between the photon energy and wavelength is given by the following equation.

$$\lambda[\mathring{A}] = \frac{hc}{\varepsilon} = \frac{12.398}{\varepsilon[KeV]} \tag{2.2}$$

When the photon interacts with matter there are 2 possible outcomes, the photon is either scattered or absorbed by the atom.(2)

There are 2 different conditions under which an x-ray photon may be reflected from a surface, the first is if the photon strikes the surface at a sufficiently shallow grazing incidence angle and the second is if constructive interference occurs as a result of several layers interacting as described in by Bragg's law.

For x-ray wavelengths the real part of the index of refraction will always be less than one, thus I can describe the index of refraction as following.(3)

$$n = 1 - \delta - i\beta \tag{2.3}$$

$$\delta = \frac{N_0 \rho r_e}{A 2\pi} \lambda^2 f_1 \tag{2.4}$$

$$\beta = \frac{N_0 \rho r_e}{A 2\pi} \lambda^2 f_2 \tag{2.5}$$

Where N_0 is avogadros number, r_e is the classic electron radius, Z is the atomic number and A is the atomic weight and ρ is the mass density for the material. $f_1 and f_2 are the complex atomics$

where f_1 and f_2 are the atomic scattering factors for the material, ρ is the material density, A_0 is Avogadro's number, r_0 is the electron radius, and λ is the photon wavelength.

There is a certain limit of the grazing angle where after the reflectivity of the surface drops rapidly, this value is known as the critical angle. The critical angle is derived from Snell's law. (4)

$$\alpha_{crit} = \sqrt{2\delta} \tag{2.6}$$

2.1.1 Refraction

Refraction of light can be viewed as an event which is dependent both upon the properties of the photon and the properties of the surface material of the interface where the material intersects the photon path of propagation.

First consider the refractive index of the material. The refractive index is a dimensionless value which is used to describe the propagation of electromagnetic waves through the medium. The refractive index is defined by equation 2.6.

When dealing with the high energy portion of the electromagnetic spectrum, where we find x-ray photons, we experience that the refractive index is less than one. The refractive index of the material is used to calculate whether or not the photons are refracted or reflected off the surface.

The refraction angle of a photon into a material of a different optical density is given as a function of the incident angle and the 2 indexes of refraction for the materials

related by Snell's law

$$\frac{sin\theta_1}{sin\theta_2} = \frac{n_2}{n_1} \tag{2.7}$$

Where θ_1, θ_2 are the angles of the incident and the refracted photon respectively and n_1 and n_2 are the corresponding indices of refraction.

The most essential parameters for the x-ray telescope is the focal length and the grazing incidence angle. At a certain radius from the center point these two parameters are related by the following equation. (4)

$$\alpha = \frac{1}{4} tan^{-1} \frac{r_0}{Z_0} \tag{2.8}$$

Where α is the graze angle, r_0 is the radius of the point from center and Z_0 is the focal length.

2.1.2 Fresnel equations

The reflectivity of the surface is determined based on a set of physical rules called the Fresnel equations. The Fresnel diffraction equations are an approximation of the Kirchhoff-Fresnel diffraction, which is vital a function in the area of x-ray spectroscopy.

The Wave vector of the incident photon is denoted as K_i and its amplitude as a_i, conversely the wave vector and the the amplitude of the reflected photon are k_r and a_r and for the transmitted photon k_T and a_T. The Fresnel equations are given as. (2)

$$r = \frac{a_R}{a_T} = \frac{\alpha - \alpha'}{\alpha + \alpha'} \tag{2.9}$$

$$t = \frac{a_T}{a_I} = \frac{2\alpha}{\alpha + \alpha'} \tag{2.10}$$

$$R = |r|^2 \tag{2.11}$$

$$T = |t|^2 \tag{2.12}$$

Where r is the amplitude reflectivity and t is the transmitivity. R is the intensity reflectivity. α is the grazing incidence angle.

2.1.3 Absorption

If the photon that interacts with an atom is absorbed then the energy is transferred to the atom and it becomes ionized. This event is referred to as photoelectric absorption. The

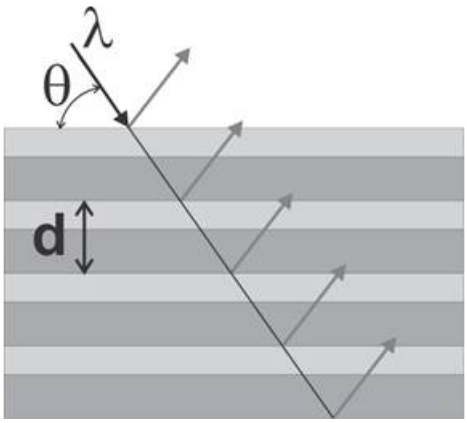

Figure 2.1: Parameters of a periodic multilayer coating: (5)

value of the absorption for a material is given by the linear absorption coefficient μ. I define the value of μ by the equation $\mu * dz$ which is the attenuation of a photon through an infinitesimal sheet of thickness dz at a distance of z from the material surface.

μ can be determined from the absorption cross section σ_a from the following equation. (2)

$$\mu = \rho\sigma_a = \frac{\rho_m N_A}{A}\sigma_a \tag{2.13}$$

Where ρ_a and ρ_m are the atomic number density and the mass density respectively, N_A is Avogadro's number and A is the atomic mass number. The absorption cross section is a variable which is dependant on the wavelength of the incident photon and is proportional to $1/\varepsilon^3$. The photoelectric cross section varies with the atomic number Z of the absorbent material roughly equivalent to Z^4.

2.1.4 Bragg's law

As was specified in equation 2.8, the critical angle is a function of the x-ray wavelength and the refractive index of the surface material, but there is a special method that can be used to construct positive interference on the incident x-ray photons.

Bragg's law is a result of Laue diffraction, where the angles of the coherent and incoherent scattering are given. For the purposes of deriving Bragg's law I consider The rays of the incident photons to always be in phase, and their wave vectors to be parallel to each other.

The θ physical equation representing Bragg's law is very simple and is given by the following. (3)

$$n * \lambda = 2 * d * sin\theta \tag{2.14}$$

where λ is the wavelength of the photon, n is a variable integer, d is the uniform distance between each of the multilayer apertures as shown in figure 2.1 and θ is the angle of the incident photon.

2.2 Raytracing

A test of the optical module of the x-ray telescope, can be done by performing simulations of the x-ray interactions in a virtual space, through a type of physics simulation software which is referred to as a ray-tracing program.

Ray tracing is following the path of a set number of moving particles which interact with a particular scene. The interactions of the particles in the scene are generally described in the form of intersections, which is to say that the coordinates of the particle coincide with the location of a object boundary contained within the scene.

Chapter 3

Method

This chapter describes the methods with which I created the model to account for the surface deformation.

3.0.1 Monte Carlo

Monte Carlo is a type of simulation in which the program collects data by repeated random sampling, thus in effect using the random nature of those simulations to obtain an empirical result. Monte Carlo methods are generally used in relatively complex systems with a large amount of variables that cannot be completely accounted for. Thus the solution is found by sampling enough times over randomly distributed simulations so that a consistent result is determined.

3.1 McXtrace

The program I used in this project is the x-ray raytracing package McXtrace. The difference between McXtrace and other similar ray tracers is the added concept of weight for each photon. This is an added variable for each photon which changes with the incidence of the photon. The weight of each photon starts at a specified value and if the photon does not intersect with any geometry in the instrument the weight will remain unchanged. If the photon reflects off a surface, the current weight of the photon will be multiplied by the calculated reflectivity of the mirror. (6)

Physically the weight factor P is the probability that the ray is transmitted through the instrument. Thus I can append this to the chance of the ray being sampled by the Monte Carlo method and I can focus only on the most relevant rays in the instrument.

Once the photon has completed a full propagation of the instrument space then I can summarize the weight factor of the photon as the product of all incident contribu-

tions, where the multiplication factor of the specified component is denoted by π_j.

$$P = p_n = p_0 \prod_{j=1}^{n} \pi_j$$

Since I know that $\pi_j \subset [0,1]$ then the weight factor must decrease after each incidence event.

3.2 Deformation

This section is focused on the development of the deformation algorithm which is used in the thesis. The algorithm development has been one of the core project goals and constitutes a majority of the theoretical and practical work in the thesis.

The most integral function of the deformation algorithm is the ability to accurately replicate the level and shape of deformation that I might expect to find in the reflector surface, thus altering the light reflecting properties of the surface to align with the deformation parameters.

What is meant by the deformation in this context is the distance of the mirror at any specific point from the ideal shape of the surface. In reality I expect that no mirror surface will be perfectly ideal, and thus it becomes necessary to take into account the perturbation of the surface and what effect it will have for the entire instrument.

An approximation which is made in this project, concerns how the shape of this deformation is estimated. Since there is no complete knowledge regarding all cause of the deformation, it will not be possible to create an accurate geometric model of the shape. Thus this project will approximate that the deformation is randomly distributed within a certain height difference range from the ideal surface plane. The other geometric properties of the deformation will be discussed later.

3.2.1 Deformation shape

In order to model the deformation, it becomes necessary to make a number of assumptions regarding its shape. However the algorithm will be designed so that it will be adaptable to any level of deformation size.

To describe this deformation I detirmened that the simplest option would be to randomly generate points within a range of maximum deformation from the surface. This would allow for quick generation of grids to describe the surface. I would therefore be able to investigate which range of perturbation gives a physically cohesive result of the surface reflectivity and scattering angles.

Another reason behind this choice was that proper simulation requires an aggregate amount of simulations performed with shapes and orders of magnitude. Therefore it is necessary for me to be able to generate a large amount of deformation grids in a short amount of time.

The grid consists of a set of points of equal distance in the x and z plane, the only variable parameter is therefore the height in the y plane. The surface height is generated by first choosing a maximum deformation height difference (Δh_{max}) between each point. I start generating the height for one point in the corner of the grid Δh_{max}

$$\begin{aligned} &\Delta h \subset [-\Delta h_{max}\ ;\ \Delta h_{max}] \\ &P_{0,0}(x) = 0 \\ &P_{0,0}(z) = 0 \\ &P_{0,0}(y) = 0 \pm \Delta h \end{aligned} \tag{3.1}$$

The system then iterates through the points in the grid, I denote the horizon position as i and the vertical position as j. The deformation height of the next point is found as a randomly generated number within a range of maximum Δh_{max} from the previously generated points.

$$\begin{aligned} &P_{i,j}(x) = i \\ &P_{i,j}(z) = j \\ &P_{i,j}(y) \subset [\ ||P_{i,j}(y) - P_{i-1,j}(y)|| < \Delta h_{max}\ ;\ ||P_{i,j}(y) - P_{i,j-1}(y)|| < \Delta h_{max}\ ] \end{aligned} \tag{3.2}$$

Figure 3.1 shows an example of the deformed surface, the size of the indices being shown is in terms of meter. There are several different orders of size that I can choose to work with and it can be difficult to relate the correlation length of the grid to an actual physical parameter. Therefore I have decided a range of deformation sizes based on tests of which brought a noticeable change to the scattering and the reflectivity of a test instrument, without completely obstructing the reflected photons.

An unavoidable hazard of using a randomly generated shape of deformation is that certain areas attain unreasonable geometric shapes compared to the geometry that would result from actual perturbation of the surface. Some of this is alleviated by the inclusion of $< \Delta h_{max}$ check in the grid generation, but it is not enough to completely removing the risk of such a deviation.

3.2.2 The instrument

As previously explained the McXtrace software operates by loading a main instrument file with all components. The instrument contains all the information that needs to be

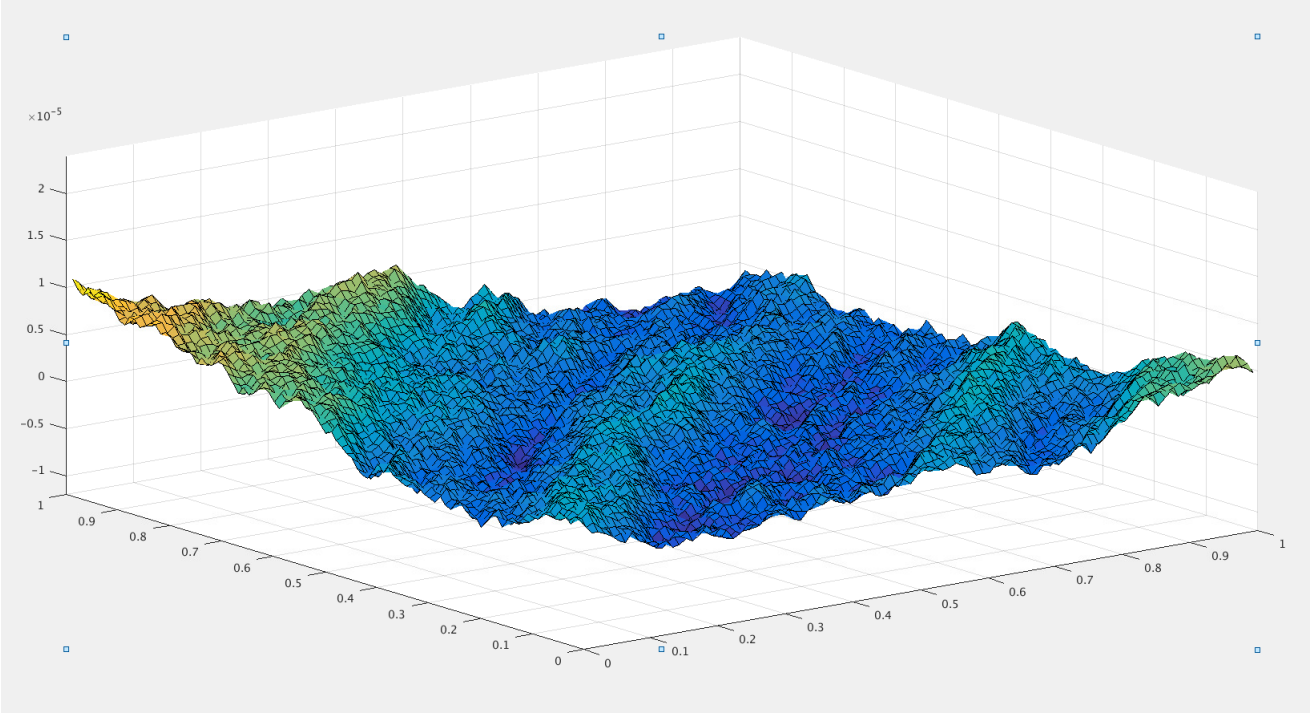

Figure 3.1: Surface grid generated by the described by the proces described in equation 3.2. Generated using a 100 x 100 point grid with Δh_{max} = 1e-6

processed in the experiment. Designing a instrument file that accurately relates the experimental information is vital to ensure that the developed algorithms are functioning properly.

In figure 3.2 we can observe the structure of a simple instrument file. The photon travels in along the z-axis and reflects first of the green mirror component and the blue, finally being recorded in the teal monitor surface. The photon interacts with each component in the order they are listed. For the instrument in figure 3.2 it would look approximately like the following:

1. Source of the x-rays: at the origin point of the instrument
2. Green mirror in the xz plane: placed at distance from origin along z-axis, rotated around x-axis.
3. Blue mirror in the xz plane : placed at distance from green mirror along z-axis compared to green mirror rotated around x-axis.
4. Teal monitor in the xy plane: placed at distance from blue mirror along z-axis rotation same as blue mirror

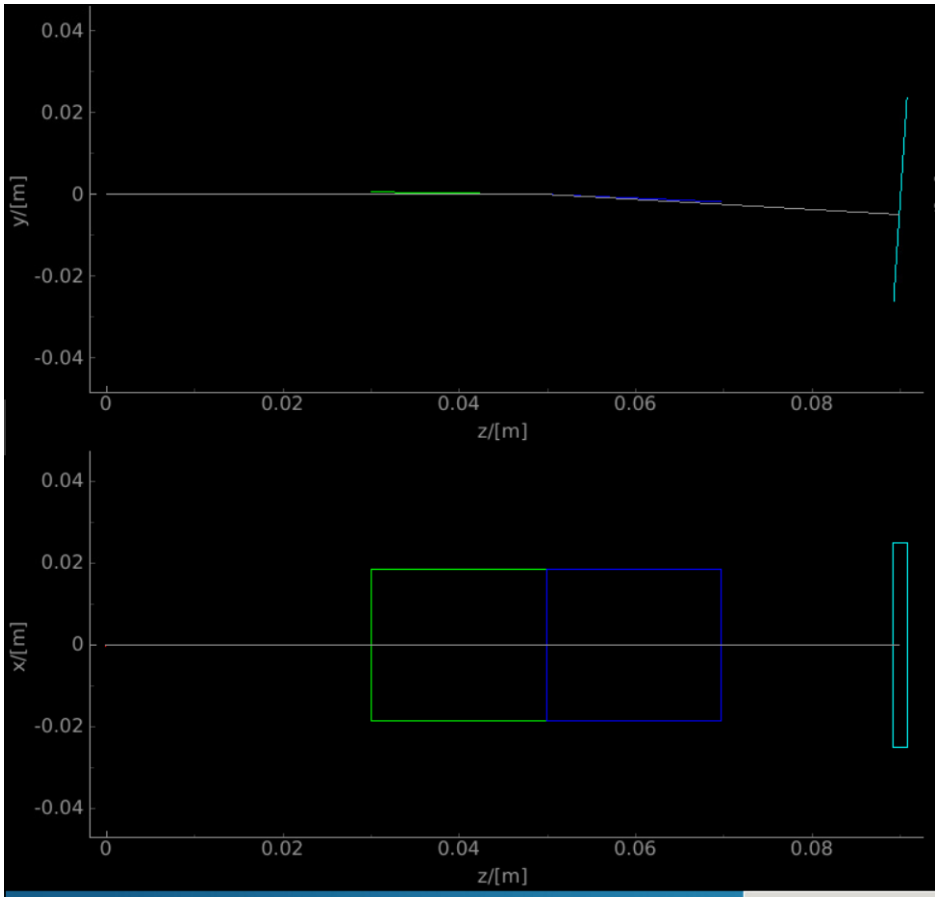

Figure 3.2: Visualization of an instrument containing 2 mirror (green and blue) compontns reflecting a photon unto a monitor surface (teal). viewed from the y and z axis (above) and the x and z axis (below)

3.3 Surface normal perturbation

A main part of the algorithm for any reflector component is determining the reflected wave-vector of the photon and determining the reflectance of the incident. The implemented algorithm shares the same method of reflection calculation as the standard reflector surface in McXtrace. The main difference lies in determining the normal vector to the surface.

First the intersection between the flat surface plane and the photon is determined. If an intersection occurs the photon is propagated to the surface of the reflector plane. Then the nearest comparable point in the grid is determined with regards to the position on the mirror. The program determines 2 points that are adjacent to the first point. Then it creates two vectors spanning between the first and second and the first and third point. The cross product of these two vector gives the values for the new surface normal.

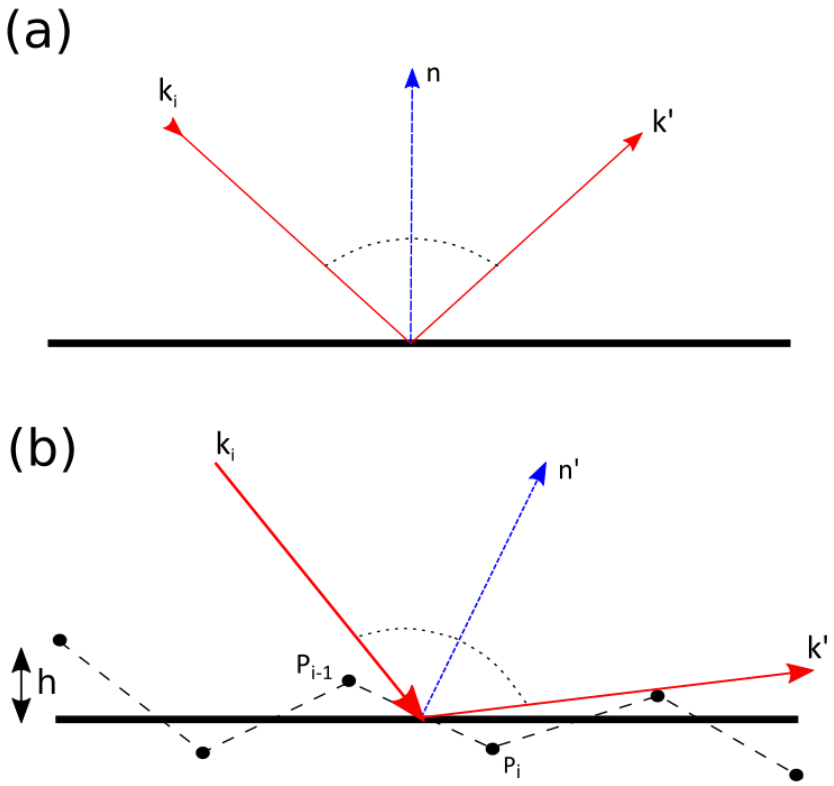

Figure 3.3: Photon wavevector k_i reflects off nondeformed surface (a) and k_i reflecting off a deformed surface with surface normal perturbation (b)

$$\begin{aligned} v_1 &= P_1 - P_2 \\ v_2 &= P_1 - P_3 \\ n' &= v_1 \times v_2 \end{aligned} \tag{3.3}$$

Once the surface normal has been determined then the wave vector k' of the reflected photon can be determined, with the follwing method.

$$k' = k_i - 2(k_i \cdot n')\, n' \tag{3.4}$$

Using the original and reflected wave-vector the angle of the incident is determined and using the angle and the energy of the photon as a reference frame I can determine the reflectivity by the nearest corresponding point on the generated reflectivity grid.

3.3.1 .OFF file

The .off file format is a specialized format for computing 3d objects as a sum of points and vertices interconnecting the defined points.

Implementing such a solution requires a script which is capable of automatically reading the surface deformation matrix into the .off file format as it would be unfeasible to do the task by hand in a reasonable time frame. The points themselves would be relatively easy to implement as both the x and the z positions are distributed evenly, leaving only the height on the y-axis as the unknown.

My algorithm computes the random grid into an .off file by splitting the the surface of the grid into a number of triangles, in this way the calculation is reminiscent of the previous method of surface normal calculation. I thus have a grid defined by set amount of smaller triangular surfaces.

In this algorithm I determine the point in the grid nearest to the photon intersection with the mirror plane, and then determine two points which are adjacent to the first point. By doing this I can get a rough estimate of the normal of the surface at that point by taking the vector cross product of the two vectors that connect these points. This cross product determines the surface normal which I can use to calculate the reflected angle of the photon.

3.3.2 Perturbation size

One of my main goals for the program was to establish a relationship between the level of deformation on the mirror surface and the intensity measured in the instrument. However, accurately establishing this relationship requires the aggregate of a large number of test runs with the instrument for different levels of deformation, since the surface is stochastically generated.

The first step to accurately model the deformation was to determine the order of magnitude for the deformation where the greatest variation in the intensity was found. This can be found analytically by applying the simulation algorithm over different deformation grids and seeing which δh_{max} in the grid resulted in the greatest change in intensity. Thus I also had to scale down the level of δh_{max} height for each deformation in the order that made sense.

3.4 Roughness Modelling

The first order of magnitude that I wish to simulate is the micro scale level, which I model by the roughness level of the surface . This process is handled separately from the other areas of the algorithm due to the fineness of the operation.

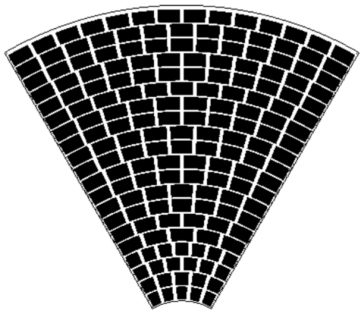

Figure 3.4: Example of a singe petal of the ATHENA telescope, showing the rows of mirrors with corresponding mirror modules

3.4.1 IMD

The tool which will be used for modelling the micro scale deformation as roughness is the software IMD, which is capable of modelling the optical properties of a multilayer film.

In IMD I describe the surface as a series of layers composed of different materials with varying depths. More relevant to the project is the fact that I can also model the material roughness of each layer of material. Using this information IMD can process a reflectivity table which gives the reflectivity value for an angle and a corresponding photon energy.

In order to calculate the reflectivity I need to construct a model of the surface as depth graded multilayer. I will attempt to replicate the coding of some of the ATHENA satellite, therefore the materials of the surface will be composed of a substrate of SiO, a layer of B4C, and a layer of Ir with depths of 100 Å and 80 Å respectively.

The IMD program uses the Fresnel equations from earlier in the theory section and also applies the knowledge from the Bragg equation, in order to calculate the reflectivity of the surface.

3.5 ATHENA instrument

Following the successful implementation of the deformation algorithm, with 2 different variants I develop a more complex simulation environment. For the practical approach to the subject I implemented the use of said algorithms for a more advanced instrument file. As outlined in the project goals, the background for this thesis is to simulate the effect of the deformation upon the ATHENA satellite, therefore I need to create an instrument which can simulate the functionality that we expect in its optical formation.

In order to simplify the process, I made a number of approximations towards

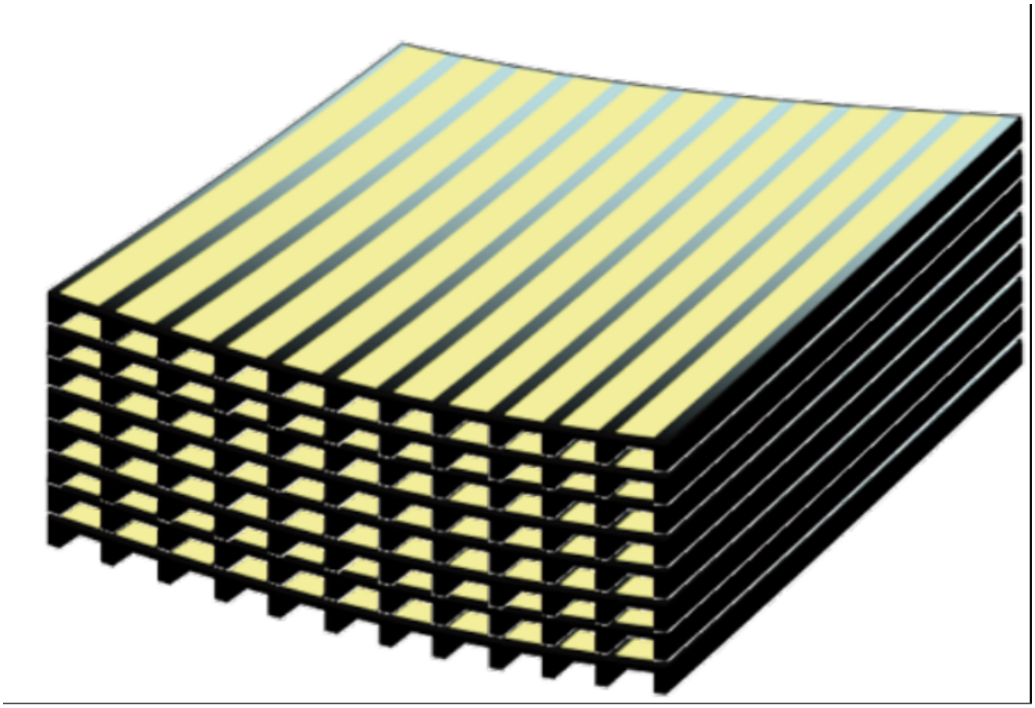

Figure 3.5: SIngle mirror module of the ATHENA telescope, with 70 layers of mirrors divided chamber walls into pores (7)

the shape of the ATHENA telescope. The ATHENA telescope is based upon the optical model of the Wolter I type x-ray telescope, meaning that it consists of layers of parabolic mirrors followed by layers of hyperbolic mirrors which are all curved around a central spoke. Since my algorithms are developed exclusively as an extension of a flat plane mirror I will be basing my model upon the conical approximation of the Wolter I model, where the mirror curves are replaced by a flat mirror layer, sharing the same general angle as the previous curved component.

In a Wolter design we designate the angle of the parabolic mirror as equal to α defined by Eq. 2.9 in chapter 2. The hyperbolic mirror is equal to 3 * α.

3.5.1 ATHENA design

The general optical layout of the ATHENA instrument is that of several thousand optical pores that focus reflect the light towards the focal point. These optical pores are divided into blocks of mirror modules containing each 2 times 34 layers of mirrors, which are then divided into pores by separating chamber walls. Simulating the entire instrument is therefore unreasonable. Since the instrument is divided into 6 petals which are individually symmetric to another the simulation can be restricted to a portion of the entire instrument size without any significant effect upon the photon distribution.

Each line of mirror modules is divided into rows and the entire instrument consists of 20 such rows. The total number of mirror layers constitute ca. 2800 layers. I opt

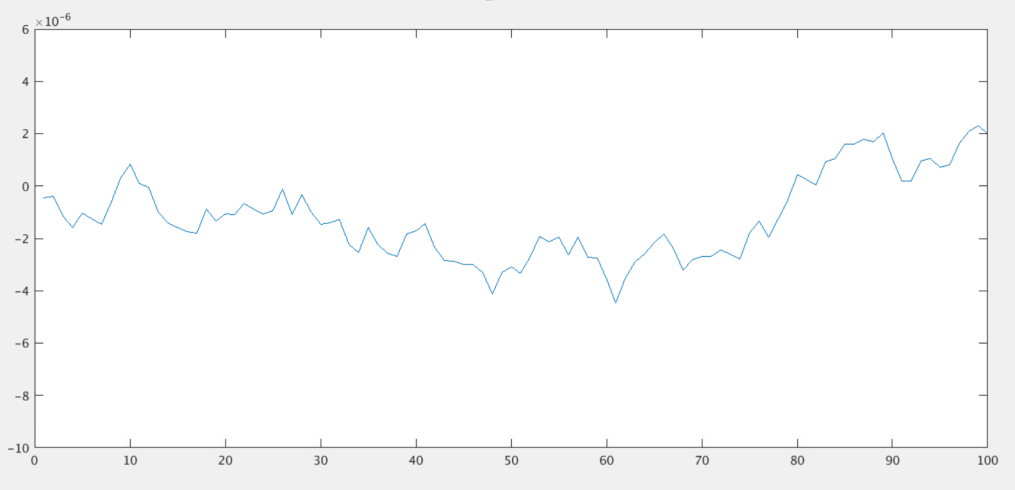

Figure 3.6: Surface grid for a deformation which is only along the z-axis

to create an approximate instrument that represents each module of the row with a single mirror, In doing this, exact knowledge of the positions and rotations is still retained from the instrument specifications.(8)

The exact rotation of each component is modelled according to the laws of reflection from the Wolter Type I telescope. Using the conical approximation I can infer that the angle of the parabolic mirror with respect to the axis of propagation is equal to α defined in Eq. 2.9, while the angle of the hyperbolic mirror is defined as 3 * α. The positioning of the mirror can then be easily obtained from knowledge of the mirror dimensions and radius of the mirror component, since the distance to the focal point is a known constant. I impose a restriction on the simulation of the instrument to a 2d plane which is again due to the fact that my algorithm is based on a flat mirror and thus cannot be curved around a central spoke. Even with these approximations the behavior of the instrument should still match with those of the ATHENA telescope.

By incurring this restriction to the dimensions of the instrument, I can opt to use a deformation grid, which is deformed along one axis. Therefore I can alter the form of equation 3.3 to the following shape.

$$\begin{aligned} P_{i,j}(x) &= i \\ P_{i,j}(z) &= j \\ P_{i,j}(y) &= P_{i-1,j}(y) \pm \Delta h \end{aligned} \tag{3.5}$$

The combination of large amount of mirrors allows the use of differently generated deformation grids with the same parameters, for each of the components used in the instrument and thus limiting the uncertainty that is relevant to the use of stochastically generated surfaces.

Chapter 4

Results

This chapter will be split into 3 different parts that cover different iterations of the program.

4.1 ATHENA instrument results

Results obtained in this section are derived from simulations through a simplified model of the ATHENA telescope described earlier. The instrument contains 40 different mirror components which will each use a differently generated surface to ensure that the effect of the deformation is random.

4.1.1 Roughness

The first aspect of the deformation that I wish to model is the micro scale. I will model the deformation which is in the microscale level as roughness. This is implemented through the IMD program to generate tables for the reflectivity.

For the ATHENA instrument I would expect to find a roughness average of 4.5 Å based on current knowledge of produced mirrors. In order to test the more extreme cases I will vary the roughness in the range of $\subset [0-20\mathring{A}]$ and determine if there is a linear correlation between the roughness and the resulting reflectivity.

The first result shows the intensity measured at the focal point of the ATHENA instrument varying only the levels of roughness. The mirror components used in this test are non deformed so $\Delta h_{max} = 0$. From figure 4.1 it can be seen that the intensity decreases linearly as a function of the roughness magnitude.

Next I test whether the association still applies in the case of a deformed mirror. Therefore I will repeat the previous test with a uniform $\Delta h_{max} = 1e-6$.

Figure 4.1: Intensity of the instrument given at different levels of surface roughness, performed from flat non-deformed mirror components

From figure 4.2 I can see that the shape of the intensity variance is almost identical to that of Figure 4.1. There is a slight change in the slope of the curve at different levels of deformation. Therefore I make the assumption that I can vary the levels of roughness and deformation independently of each other throughout the tests carried out with the ATHENA instrument.

While varying the Δh_{max} for the simulations, I keep the level of the roughness at 4.5 Å as this would be the expected average of the roughness for the ATHENA mirrors.

4.1.2 Intensity from deformation height

In figure 4.3 I have performed extensive testing of the ATHENA instrument with various surface deformation heights. These simulations were done with a monitor of the size 5 * 5 cm. As can be observed, the loss in intensity follows the level of deformation and the change in intensity follows a potential function given as.

$$Intensity \propto 0.00003 * \Delta h_{max}^{-0.2} 1e-7m \tag{4.1}$$

which is in terms of meters can be rewritten to.

$$Intensity \propto 300 * \Delta h_{max}^{-0.2} m \tag{4.2}$$

I assume that the drop in intensity variation may be caused by the limited size of

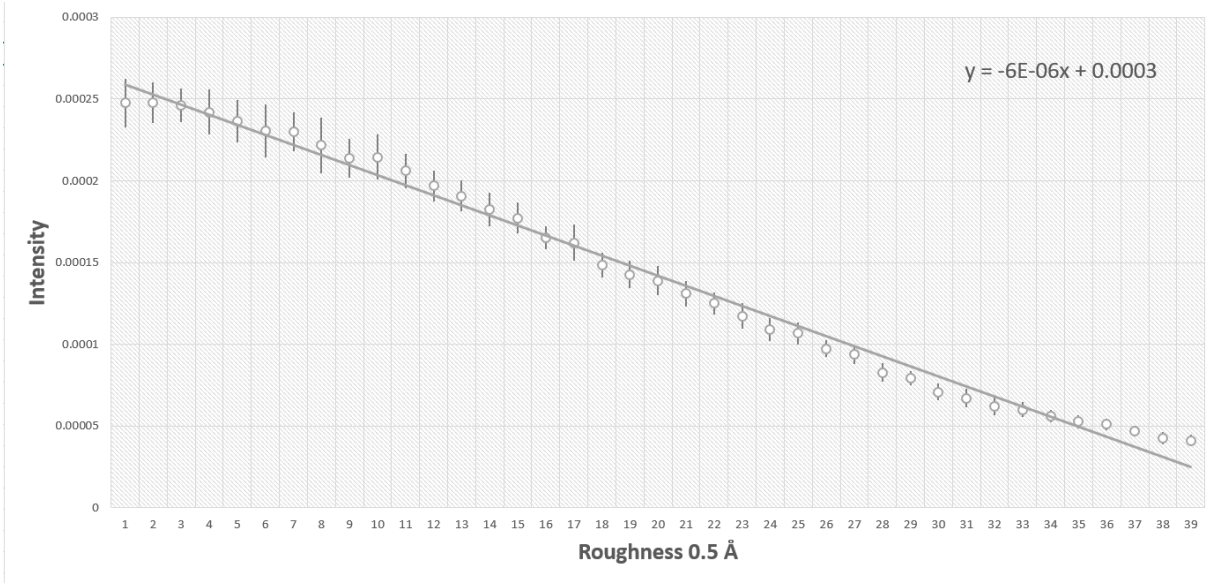

Figure 4.2: Intensity of the instrument given at different levels of surface roughness, performed from an ATHENA insturment with a $\Delta h_{max} = 1e-6$ size mirror deformation

the detector area which doesn't account for the total amount of reflected photons.

4.2 Scattering

The method by which we measure the angular resolution of an x-ray telescope is the Half Power Diameter (HPD) of the instrument. HPD designates the size of the circular area in which half of the focused photons are located. In the ATHENA instrument the focal point is the zero point of the y-axis.

I assume that due to the nature of roughness and the method by which I generate the roughness in this instrument the value of the roughness will not have any impact upon the scattering angle of the photons. Based on this, it becomes meaningless to include different values of roughness when determining the scatter for different levels of deformation.

I measure the scattering of the photons in terms of the HPD, which is defined as the area with half the total photon density.

Similar to the result seen for the intensity we see an association between the scattering that approaches a linear level which proceeds to flatten out (Figure 4.4). In this case however, the drop is more sudden and for the first half of the tests the relationship can be seen as appriximately linear.

From Figure 4.5 i can observe that the HPD follows the height deformation by the folllowing equation

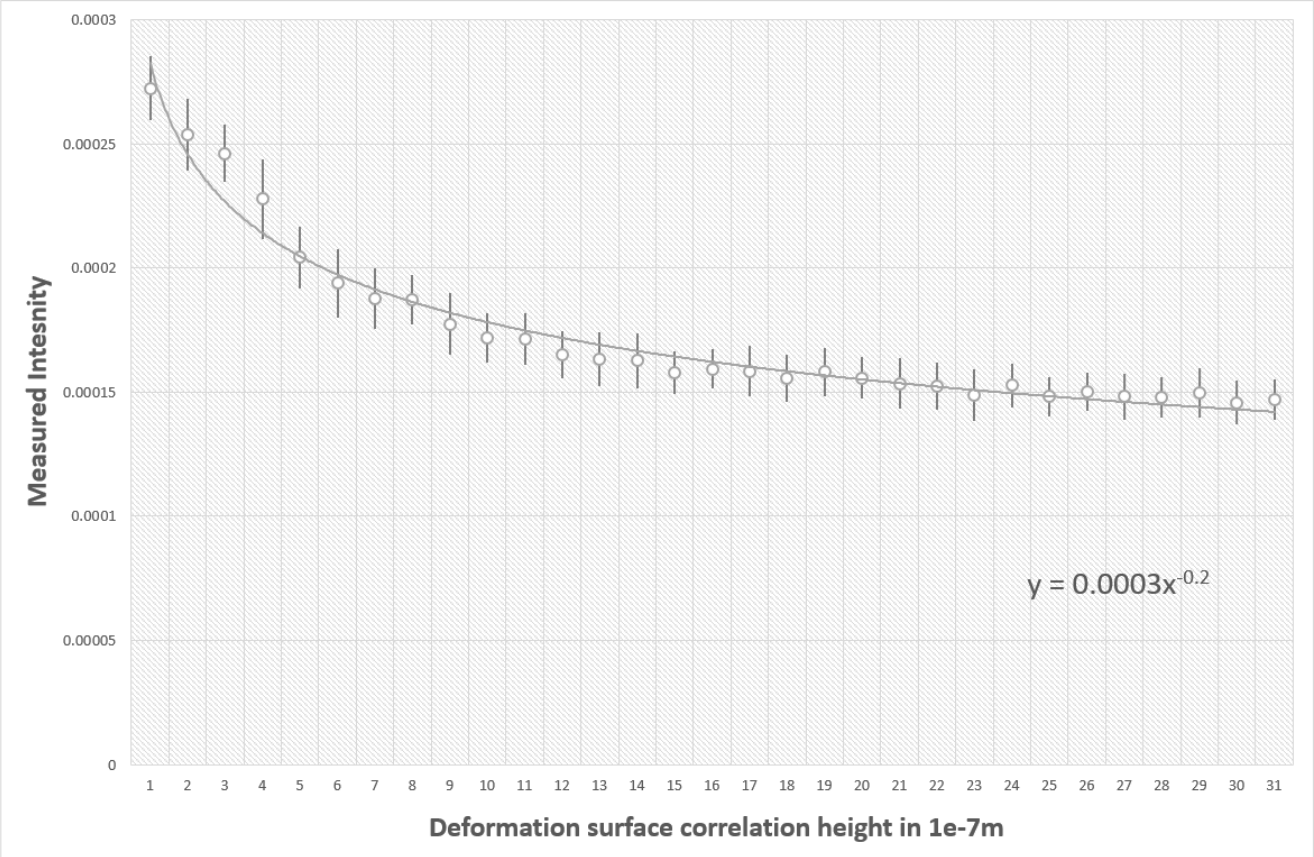

Figure 4.3: Intensity as a function of roughness for simple instrument performed with a roughness of 4.5 Å

$$HPD(m) \propto 0.0032 * \Delta h_{max}^{-0.2} 1e-7m \tag{4.3}$$

I choose to investigate the scattering in more detail by viewing the distribution of the photons along the y-axis.

Figure 4.6 shows that for an instrument with no deformation the photons are focused in the center point of the detector, which is the focal point of the instrument.

Figure 4.7 shows that a minimal level of deformation creates an area of scattered photons closely along the center point of the detector, decreasing the height of the central peak slightly

Figure 4.8 shows that for a greater level of deformation there is still a central peak, and the rest of the photons are scattered in a wide area around the focal point, this lends credence to the theory that the measured HPD stops increasing at a certain point due to the photons being scattered outside of the capturing area of the detector.

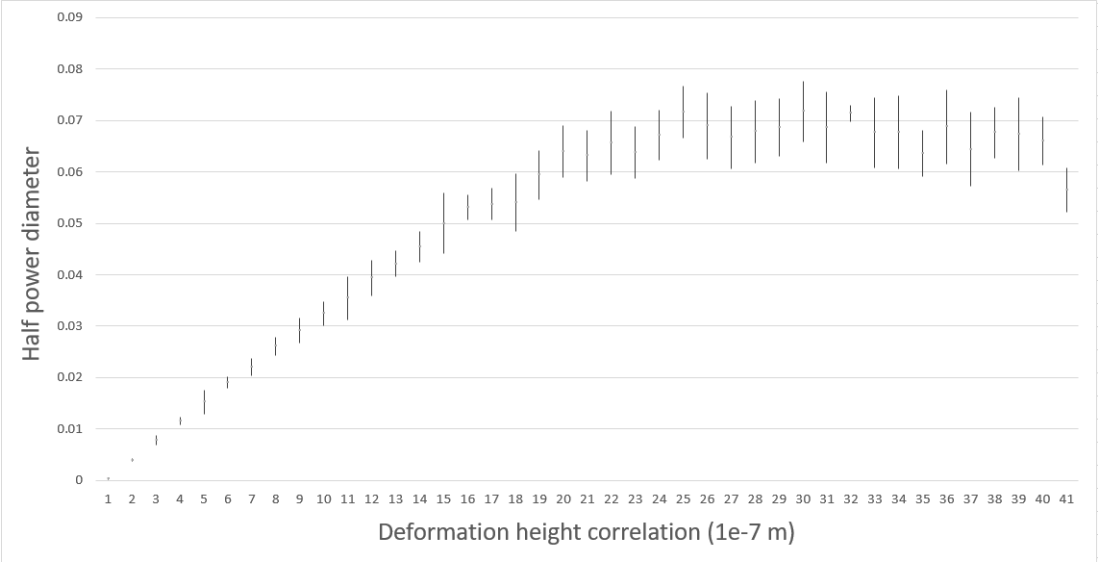

Figure 4.4: Scattering of the photons as given by HPD in the ATHENA telescope with a detector size of 0.5 * 0.5 m

4.2.1 OFF structure intersection

As I would expect from initial testing of the off intersection algorithm, it is very time consuming when applied to an advanced simulation. The ATHENA test instrument includes a total of 40 different mirror components and for the simulation of the instrument to run the intersection needs to be checked for all of the components. I found that the average run time for a regular single incidence instrument averaged around 2 minutes, which would mean that I expect the run time of the ATHENA instrument to be about 80 minutes for the off intersection algorithm.

After testing the simulation I found that the necessary time for the simulation is ca. 90 minutes, so I will use only a small sample size of OFF intersection routines for comparison as the time frame of this project does not allow for averaging values of this algorithm.

I perform a limited number of sampled testing to compare the result of the off structure intersection and the surface normal change tests to see whether the values are similar. There will very likely be a large amount of uncertainty for the off intersection results as I can only apply them once for each value, as such they will be used as a point of comparison for the normal surface algorithm rather than a full fledged description of the surface.

One of the most interesting results of the simulation with the off intersection is the scattering of the peaks. At Δh_{max} = 1e-7 m I observe a distribution of photons that is somewhat similar to the one found in figure 4.6

Figure 4.9 shows a central peak at the center point of the monitor and 2 symmetrically scattered peaks of lower intensity, which is different from the scattering found

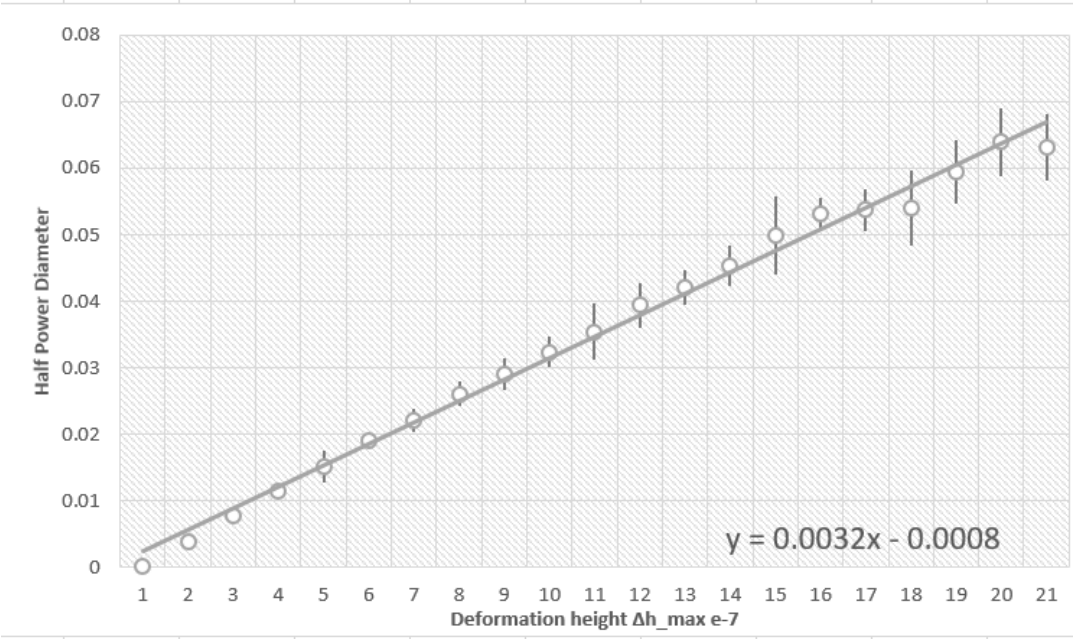

Figure 4.5: Scattering of the photons as given by HPD in the ATHENA telescope with a detector size of 0.5 * 0.5 m, from $\Delta h_{max} = 0$ to $\Delta h_{max} = 2e-6$

for the similar instrument as the scattered photons are not distributed smoothly along the y-axis.

At a level of Δh_{max} = 1e-6 m I find the following result(Figure 4.1). Similar to figure 4.9 I find a that the position of the symmetrical scattered peaks remain the same while increasing in intensity and the central peak conversely losing intensity. Very few of the photons are spread beyond these peaks.

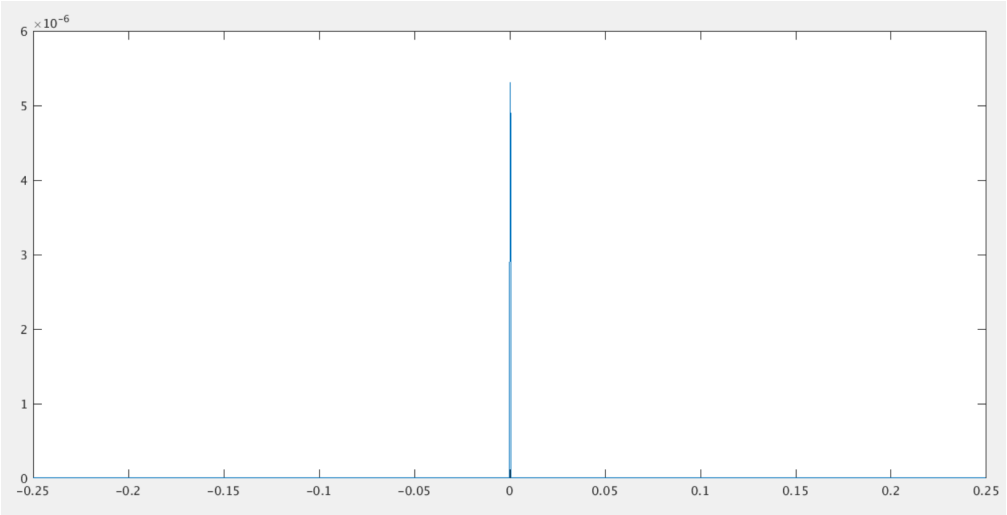

Figure 4.6: Positions and photon weight of the photons on the detector at focal point measured along the y-axis for a global deformation level of 0

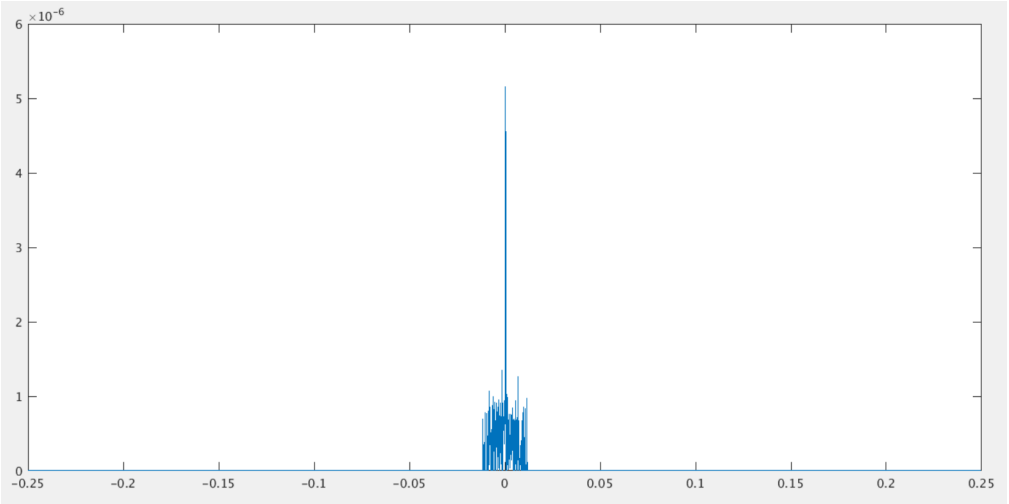

Figure 4.7: Positions and photon weight of the photons on the detector at focal point measured along the y-axis for a global deformation level of $\Delta h_{max} = 1e-6$ 1e-7

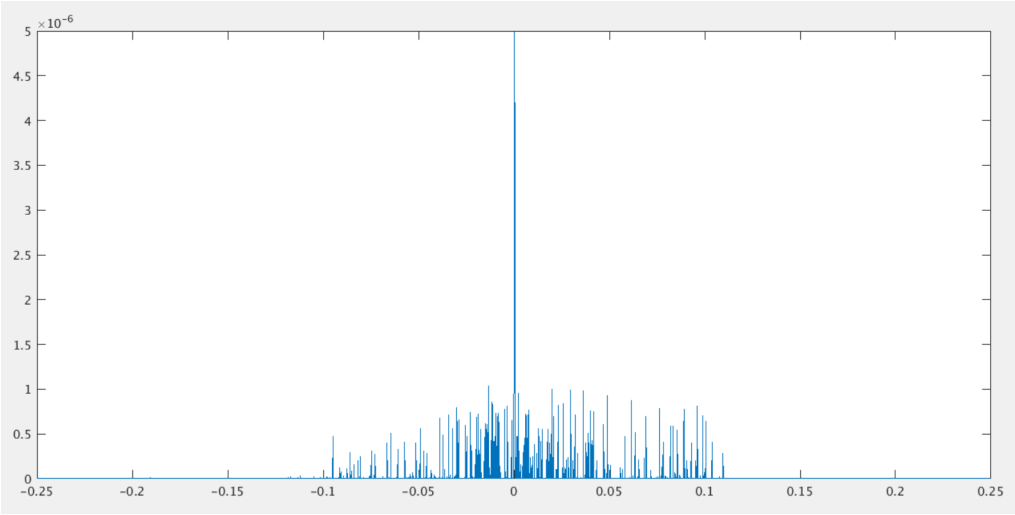

Figure 4.8: Positions and photon weight of the photons on the detector at focal point measured along the y-axis for a global deformation level of $\Delta h_{max} = 1e-6$ = 1e-6

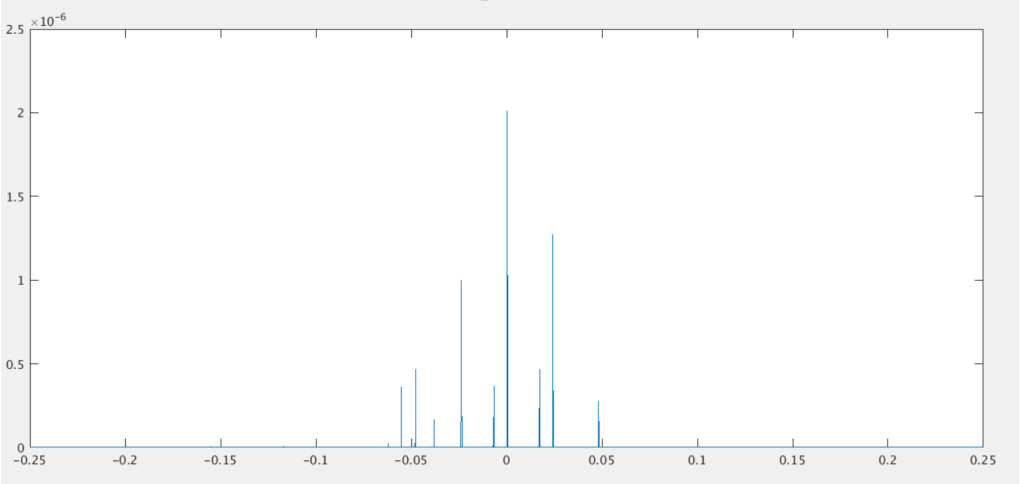

Figure 4.9: Scattering results from a position sensitive monitor of size 0.5 *0.5 m measuring the amount of incident photons along the y-axis Δh_{max} = 1e-6 m

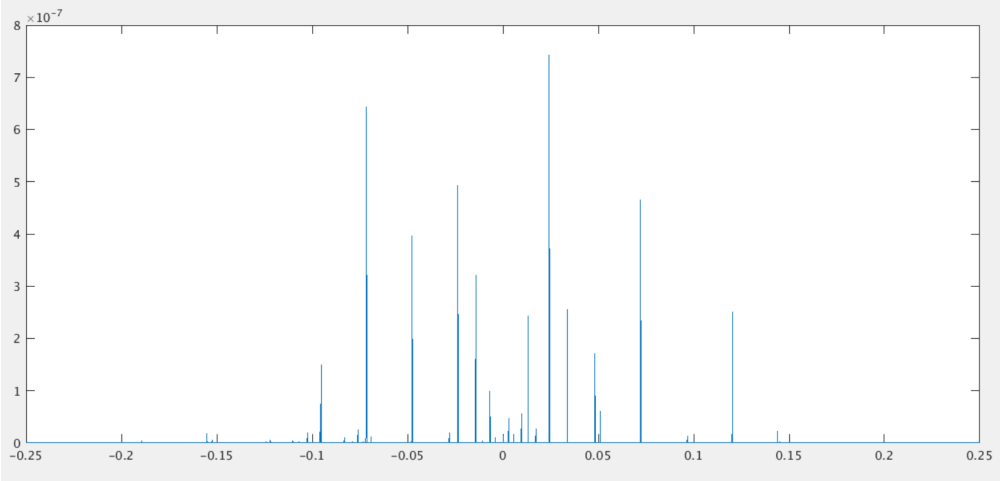

Figure 4.10: Scattering results from a position sensitive monitor of size 0.5 *0.5 m measuring the amount of incident photons along the y-axis Δh_{max} = 4e-6m

Chapter 5

Discussion

This chapter concerns the interpretation of the results, and discussion of the potential flaws of the developed methods.

5.1 Deformation algorithms

The development of the algorithms to model the deformation was a key part of the goals for this thesis and analyzing the results it is clear that the difference in how they function greatly impact the simulation of the surface deformation

5.1.1 Normal incidence

The normal incidence angle algorithm was the primary code used to model the deformation in the project due to its efficiency compared to other tested alterations.

The scattering of the results from this algorithm provided a smooth scattering along an area that expanded with the size of the deformation. Unexpectedly, the detector measured a central peak that suffered little aberration with the increasing deformation size.

A possible source of errors for the algorithm is the fact that the manner of which the deformation grid is generated is not a true random surface but rather generated in a manner that closely approximates the shape of such a surface, with the specified surface correlation between each point, with each point in the 2d grid being correlated with two previous points and each point in the 1D version being generated from the previous.

5.1.2 off intersection

This algorithm required 2 parts in order to function: the first part to convert the grid to a 3 dimensional .off file and a function to calculate intersection with each off vertices.

However checking the intersection for each triangular surface in the deformation grid is a very inefficient process, which is the reason that it was only used sparingly in the simulation of the ATHENA instrument.

The results of the off intersection provided very different results compared to the simulations performed with the changed surface normal algorithm. This is likely due to the manner in which the surface is treated, which causes the off intersection to condense the stream of scattered photons into key areas of higher density compared to the smooth scattering found in the other result.

5.2 ATHENA instrument

The work done on constructing a virtual segment of the ATHENA instrument constitutes the application of developed software to a relevant scenario. This has provided a better grasp at the efficiency of which the algorithm can be applied outside of test cases.

There is a significant distance in terms of work when making an instrument for simulation of physically relevant scenario and a test instrument for investigating the functionality of specific system components. The ATHENA instrument proved that algorithms that were applicable in test cases are significantly less effective in a increasingly advanced instrument simulation. This is the reason that the off intersection algorithm could only be used sparingly for the instrument as the intersections would take too long to determine, to be properly effective.

In test results for the ATHENA instrument I have managed to find associations between the levels of deformation and the reflectivity and scattering. At higher levels of deformation the effect of the deformation becomes less pronounced which may be caused by the fact that the monitor area is only able to detect a subset of the scattered photons due to its limited size.

The scattering results for the surface normal perturbation and the off intersection model show a large degree of difference. It is possible that the measured scattering image may be a result of how the program samples over the monitor results.

5.2.1 Future work

In the case of a longer time frame for the project I would have liked to create a more complex version of the current algorithm and in the course of the project I have consid-

ered several possibilities for how such a thing might be done. The first possibility which was most strongly considered was modelling the deformed surface according to a point set surface model.

In addition I would have liked to be able to run a larger number of tests especially for the off intersection algorithm so I could better compare the results of the 2 simulation models.

Given additional time it might have been possible for me to compare the results of my simulations with physical test results from the ATHENA mirror modules, and thereby test the accuracy of my results and determine which of the two deformation models was the superior.

Chapter 6

Conclusion

The main goal of this project was to provide a working model for the deformation upon a reflector surface for use with the McXtrace software, and through this project I have developed a working program. While the program does makes several approximations on the behavior of the photon it does provide results that are within the expected parameters.

What has been developed within the time frame of this Bachelor project is a model which approximates the behavior of the deformed surface. However, the program makes a large amount of approximations regarding the reflection interaction, most prominently the assumption that no shadowing occurs for the surface normal perturbation algorithm. The off intersection does however take the shadowing into account but the application of the algorithm is too cumbersome for advanced instrument simulations.

The ATHENA telescope simulation has proven the efficiency of the algorithms in an advanced simulation and has shown how the different deformation parameters will effect the accuracy of the instrument.

The project found a relation between the level of deformation and the increased spread and decreased intensity of the reflected photons, in the case of the ATHENA instrument. I have expressed both of these relations in terms of of the following mathematical equations.

For the intensity as a function of Δh_{max} I find the following relation

$$Intensity \propto 0.00003 * \Delta h_{max}^{-0.2} \tag{6.1}$$

ANd for the HPD I found a similar linear relation.

$$HPD(m) \propto 0.0032 * \Delta h_{max}^{-0.2} 1e-7m \tag{6.2}$$

One of the most significant result of the project was the difference between the simulation for the two deformation models, it is expected that the reflected photons of the off intersection model would follow a less smooth scattering patterns, however instead

all of the reflected photons are condensed into symmetric peaks despite all 40 mirrors having different surfaces. The Surface normal perturbation scattered the photons over an area that increases with the deformation size as would be expected, and yet it has a central peak which is far too prominent over higher orders of magnitude for the deformation.

I conclude that that the developed methods provide a functioning yet approximated means of measuring the effect of deformation upon x-ray reflector, specifically in the case of multilayered Wolter I telescope optics.

.

Bibliography

[1] H. B. B. Aschenbach, "Grazing incidence telescopes for esa's x-ray astronomy mission xmm," *X-Ray Instrumentation in Astronomy II*, 1988.

[2] J. A. Nielsen and D. McMorrow, *Elements of Modern X-ray physics*. JOHN WILEY SONS, LTD, 1993.

[3] B. Aschenbach, "X-ray telescopes," *Reports on Progress in Physics*, 1985.

[4] P. B. Reid, "X-ray telescopes," *ENCYCLOPEDIA OF ASTRONOMY AND ASTROPHYSICS*, 2001.

[5] "Periodic multilayers." `http://www.rxollc.com/technology/periodics.html`, 2013. [Online; accessed 16-June-2017].

[6] P. W. E. B. Knudse and, E. Farhi, K. Lefmann, and S. Schmidt, *User and Programmers' Guide to the X Ray-Tracing Package McXtrace version 1.2*. Physics Physics Department, Technical University of Denmark, 2800 Kongens Lyngby, Denmark, April 2014.

[7]

[8] T. Oosterbroek, "Athena telescope reference design and effective area estimates," tech. rep., European Space Research and Technology Centre, 2201 AZ Noordqijk, The Netherlands, 2014.

Appendix A

Appendix

The appendix includes the codes used for the project. They are split between matlab and C code

A.1 Matlab

A.1.1 1D surface generation

.

```
size1=100;
rel=0.0000001*count;
g=zeros(size1,size1);
for j=1:size1
%          This handles the first point generated int the
    matrix
          if   j == 1;
             g(:,j)= -rel+2*rel*rand();
             while g(:,j) > rel
             g(:,j) =-rel+2*rel*rand();
             end;
          else
              g(:,j)=g(:,j-1)-rel+2*rel*rand();
             while abs(g(:,j)-g(:,j-1)) > rel
             g(:,j) =g(:,j-1)-rel+2*rel*rand();
             end;
          end
end;
filename = sprintf('%ddg.txt',count);
dlmwrite(filename,g);
```

A.1.2 2D surface generation

```
size1=100;
rel=0.000001;

g=zeros(size1,size1);


for j=1:size1;
    for i=1:size1;
%         This handles the first point generated int the
    matrix
          if  i == 1 && j == 1;
             g(j,i)= -rel+2*rel*rand();
             while g(j,i) > rel
             g(j,i) =-rel+2*rel*rand();
             end;
%          This handles the the tpo row of the matrix
          elseif i > 1 && j==1;
             g(j,i) =g(j,i-1)-rel+2*rel*rand();
             while abs(g(j,i)-g(j,i-1)) > rel
             g(j,i) =g(j,i-1)-rel+2*rel*rand();
             end;
%          This handles the first column of each row
          elseif i == 1 && j > 1;
              g(j,i)= g(j-1,i)-rel+2*rel*rand();
              while abs(g(j,i)-g(j-1,i)) > rel
              g(j,i)=g(j-1,i)-rel+2*rel*rand();
              end
%          This handles the remaining positions
         else
             g(j,i) =g(j,i-1)-rel+2*rel*rand();
              while abs(g(j,i)-g(j-1,i)) > rel || abs(g(j,i
                  )-g(j,i-1)) > rel
                  g(j,i) =g(j,i-1)-rel+2*rel*rand();

              end;
          end;
    end;
end;
filename = sprintf('%ddg.txt',count);
dlmwrite(filename,g);
```

A.1.3 off file converter

This code was used for converting a surface grid into an .off file

```
% This is the script to convert the deformation files to
    off files
% this is a necessary step if we wish to use
    off_intersection to properly
% model the shadow behavior of the incident photons
for count=1:200
deform_name = sprintf('%ddg.txt',count);
off_name =sprintf('deform_surf%d.off',count);
def_file = fopen(off_name,'w');
def = dlmread(deform_name);
% def_file = fopen('5x5.OFF','w');
% def = dlmread(deform_name);


[length,width] = size(def);
format long;

% Header of the off file
fprintf(def_file,'OFF\n');

fprintf(def_file,'%d %d %d \n',length*width,((width-1))
    *2*(length-1),0);

% First part of the off file with the point postions
     for j=1:width;
          for i=1:length;
          ind=def(j,i);
          fprintf(def_file,'%f %f %f\n',(j-width/2)/(width
              *10),ind,(i-length*2)/(length*10));
          end
     end

% second part of the off file with the surface definitions
     for j=1:width-1;
          for i=1:length-1;
          current=(j-1)*width+i;
          fprintf(def_file,'%d %d %d %d \n',3,current-1,
              current,current+width-1);
          fprintf(def_file,'%d %d %d %d \n',3,current,
              current+width-1,current+width);
          end
     end
fclose(def_file);
end
```

HPD calculate This code was used to calculate the HPD from the output of a position

sensitive monitor

```
file = fopen('HPD_data.txt','w');
format long;

fileread =sprintf('sincurv2/final_psd.dat',count);
c = textread(fileread,'',-1, 'commentstyle','shell');
tot = sum(c(:,3));

start = length(c(:,1))/2;
i = 0;
tin = 0;

while tin < tot/2
    tin = tin + c(start-i,3)+c(start+1+i,3);
    i=i+1;
end
HPD = abs(c(1,1)-c(i,1));

fprintf(file,'%d\n',HPD);

fclose(file);
```

A.2 C code

A.2.1 ATHENA surface normal perturbation component

```
/*******************************************************************************

* Component: new_deform
*
* %I
*
* Written by: Toke bjorner
* Date: MAY 2017
* Version: 1.3
* Release: McXtrace 1.4
* Origin: DTU Physics
*
* Deformed flat mirror
*
* %D
* alternative version of deform mirror xz plane
*
```

```
* %P
* Input parameters:++
* width [m] The width of the mirror
* length [m] The length of the mirror
* R0 [ ]  Constant reflecticity
* reflec [ ] text containing reflectivities in a 2D matrix
    block parameterized by energy and glancing angle.
* deform [ ] text file containing the heights of the
   deformed surface
* gridn: number of deformation grid that the user wishes
   to use
* reflecn: number of reflectivity table that the user
   wishes to use
*
* This is an alternative version of the new_deform.comp
   which is able to read an ordered list of deformation
   grids or reflectivity tables
*
* (none)
* %E
*******************************************************************************

DEFINE COMPONENT ATHENA_deform
DEFINITION PARAMETERS (double gridn=0,double reflecn=0)
SETTING PARAMETERS (length=1, width=1, R0=1)
OUTPUT PARAMETERS (reflec_table)
/* X-ray parameters: (x,y,z,kx,ky,kz,phi,t,Ex,Ey,Ez,p) */
SHARE
%{
//#include "read_table-lib"
//#include "interoff-lib"
%}

DECLARE
%{
      double e_min,e_max,e_step,theta_min,theta_max,
         theta_step,length_step,width_step;
      t_Table reflec_table, deform_table, deform_table2;
      int use_reflec_table, use_def_table;
%}

INITIALIZE%{
      if (reflecn) {
      char ref[100];
      snprintf(ref,sizeof(char)*100,"ReflecRoughnes%d.txt",
```

```
        reflecn);

      Table_Read(&reflec_table,ref,1);
      t_Table *tp=&reflec_table;
      e_min=0.5;
      e_max=20;
      e_step=0.5;
      theta_min=0.1;
      theta_max=5;
      theta_step=0.1;
      use_reflec_table=1;
    }else{
      use_reflec_table=0;
    }

      // second table read function to interpret the
         deformation matrix into a table file
      if (gridn) {
      char def[100];
      snprintf(def,sizeof(char)*100,"%ddg.txt",gridn);

      Table_Read(&deform_table,def,1);
      t_Table *tp2=&deform_table;

       //compute length and width step for deformation
          matrix
      length_step=length/(tp2->rows);
      width_step=width/(tp2->columns);
       use_def_table=1;
    }else{
      use_def_table=0;
    }

%}

TRACE
%{
    double l_small;
    double kl;

    l_small=DBL_MAX;

    plane_intersect(&kl,x,y,z,kx,ky,kz,0,1,0,0,0,0);

    PROP_DL(kl);
```

```
    if(x<-width/2.0 || x>width/2.0 || z<-length/2.0 || z>
       length/2.0){
       RESTORE_XRAY(INDEX_CURRENT_COMP, x,y,z, kx,ky,kz,
          phi,t, Ex,Ey,Ez, p);
    }else{

    double nx,ny,nz,R;

    if (use_def_table){
// Read the height values for the incident position and 2
   adjacent points
    double height1=Table_Value2d(deform_table, (x+(width
       /2))/width_step, (z+(length/2))/length_step);
    double height2=Table_Value2d(deform_table, ((x+(width
       /2)-width_step)/width_step), (z+(length/2))/
       length_step);
    double height3=Table_Value2d(deform_table, (x+(width
       /2))/width_step, ((z+(length/2)-length_step)/
       length_step));
// Define the 3 points
    double p1[] = {x, height1, z};
    double p2[] = {x-(width_step), height2, z};
    double p3[] = {x, height3, z-(length_step)};

// Define 2 vectors that span the 3 points
    double vec1[3], vec2[3];
    vec1[0] = p1[0]-p2[0];
    vec1[1] = p1[1]-p2[1];
    vec1[2] = p1[2]-p2[2];

    vec2[0] = p1[0]-p3[0];
    vec2[1] = p1[1]-p3[1];
    vec2[2] = p1[2]-p3[2];


// Determine the surface normal by calculating the cross/
   product of the 2 vectors
    vec_prod(nx,ny,nz,vec1[0],vec1[1],vec1[2],vec2[0],vec2
       [1],vec2[2]);
    NORM(nx,ny,nz);
    }else{
        nx=0;
        ny=1;
        nz=0;
    }
```

```
            double kix=kx,kiy=ky,kiz=kz;
            double s=scalar_prod(kx,ky,kz,nx,ny,nz);
             kx-=2*s*nx;
             ky-=2*s*ny;
             kz-=2*s*nz;
             SCATTER;
        if(use_reflec_table){
                double k=sqrt(kx*kx+ ky*ky + kz*kz);
                double theta=RAD2DEG*0.5*acos(scalar_prod(kx,
                   ky,kz,kix,kiy,kiz)/k/k);
                double e=K2E*k;
                R=Table_Value2d(reflec_table,(e-e_min)/e_step,
                     (theta-theta_min)/theta_step);
            }else{
            R=R0;
        }
          p*=R;
       }

%}

MCDISPLAY
%{

magnify("");
   line(-width/2.0,0,-length/2.0, width/2.0,0,-length/2.0);
   line(-width/2.0,0, length/2.0, width/2.0,0, length/2.0);
   line(-width/2.0,0,-length/2.0,-width/2.0,0, length/2.0);
   line( width/2.0,0,-length/2.0, width/2.0,0, length/2.0);

%}


END
```

A.2.2 ATHENA off intersection component

```
/*******************************************************************

* Component: off_deform
*
* %I
*
* Written by: Toke bjorner
* Date: MAY 2017
* Version: 1.3
```

```
* Release: McXtrace 1.4
* Origin: DTU Physics
*
* Deformed flat mirror
*
* %D
* mirror for detailing the relfection from .off file
*
* %P
* Input parameters:
* width [m] The width of the mirror
* length [m] The length of the mirror
* R0 [ ]  Constant reflecticity
* reflec [ ] text containing reflectivities in a 2D matrix
    block parameterized by energy and glancing angle.
* deform [ ] text file containing the heights of the
   deformed surface
*
* This model is based largely upon the Mirror.com model of
    the flat mirror in the xz plane
*
* (none)
* %E
*******************************************************************************


DEFINE COMPONENT ATHENA_off
DEFINITION PARAMETERS (double reflecn=4.5,double offn=0)
SETTING PARAMETERS (length=1, width=1, R0=1)
OUTPUT PARAMETERS (reflec_table)
/* X-ray parameters: (x,y,z,kx,ky,kz,phi,t,Ex,Ey,Ez,p) */
SHARE
%{
//#include "read_table-lib"
//#include "interoff-lib"
#ifndef INTEROFF_LIB_C
#include "interoff-lib.c"
#endif

#ifndef INTEROFF_LIB_H
#include "interoff-lib.h"
#endif
%}

DECLARE
%{
```

```
      double e_min,e_max,e_step,theta_min,theta_max,
         theta_step,length_step,width_step;
      t_Table reflec_table;
      off_struct deform_off;
%}

INITIALIZE%{
      char ref[100];
      snprintf(ref,sizeof(char)*100,"ReflecRoughnes%d.txt",
         reflecn);
      Table_Read(&reflec_table,ref,1);
      t_Table *tp=&reflec_table;
       e_min=0.5;
       e_max=20;
       e_step=0.5;
       theta_min=0.1;
       theta_max=5;
       theta_step=0.1;


      // read the off deform file into the component
      char off[100];
      snprintf(off,sizeof(char)*100,"deform_surf%d.off",
         offn);
      off_init(off,width,0,length,0,&deform_off);
%}

TRACE
%{
     double l_small;
     double kl,xn,yn,zn;
     Coords n0,n3;
     double l0,l3,k2,l;

// need to propagate to the off_intersection therefore i
   need to find the intersection length
// off_intersect only gives the time of intersect

     off_x_intersect(&l0,&l3,&n0,&n3,x,y,z,kx,ky,kz,
        deform_off);
     PROP_DL(l0);

    if(x<-width/2.0 || x>width/2.0 || z<-length/2.0 || z>
       length/2.0){
         //ABSORB;
         RESTORE_XRAY(INDEX_CURRENT_COMP, x,y,z, kx,ky,kz,
```

```
            phi,t, Ex,Ey,Ez, p);
    }else{

    coords_get(n0,&xn,&yn,&zn);
    NORM(xn,yn,zn);
    double nx,ny,nz,R;

//This action is to flip the surface normal corresponding
   to the direction of the ray
    if (kz>0 && zn<0 || kz<0 && zn>0){
        nx=xn;
        ny=yn;
        nz=zn;
    }else{
        nx=-xn;
        ny=-yn;
        nz=-zn;
    }

        double kix=kx,kiy=ky,kiz=kz;
        double s=scalar_prod(kx,ky,kz,nx,ny,nz);

         kx-=2*s*nx;
         ky-=2*s*ny;
         kz-=2*s*nz;

         SCATTER;

            double k=sqrt(kx*kx+ ky*ky + kz*kz);
            double theta=RAD2DEG*0.5*acos(scalar_prod(kx,
               ky,kz,kix,kiy,kiz)/k/k);
            double e=K2E*k;
            R=Table_Value2d(reflec_table,(e-e_min)/e_step,
                (theta-theta_min)/theta_step);
            p*=R;

    }
%}

MCDISPLAY
%{

magnify("");
  line(-width/2.0,0,-length/2.0, width/2.0,0,-length/2.0);
  line(-width/2.0,0, length/2.0, width/2.0,0, length/2.0);
  line(-width/2.0,0,-length/2.0,-width/2.0,0, length/2.0);
```

```
  line( width/2.0,0,-length/2.0, width/2.0,0, length/2.0);

%}


END
```

A.2.3 ATHENA instrument file

```
// model of single row of athena satelitte optical module
DEFINE INSTRUMENT ATHENA_test(int N_deform=0,int N_reflec
   =9,int mirror_distance=5)
DECLARE
%{
    int reflecs;
%}

INITIALIZE
%{

%}
TRACE
COMPONENT Origin = Progress_bar()
  AT (0,0,0) ABSOLUTE
EXTEND
%{
    reflecs=0;
%}

COMPONENT source = Source_div(
xwidth=0, yheight=3, E0=10, dE = 2, gauss = 1,focus_aw=0,
   focus_ah=0)
AT (0,0,0) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin
// split the optical modules into 20 rows consisting each
   of 2 aligned mirrors
// the angle of the first mirror is detemrined by alpha
// the angle of the second mirror is determined by 3 *
   alpha

//define an arm component for each row to act as the
   centerpoint of the row, the mirror planes will then be
   placed according to this

//1 row of the optical module:
```

```
COMPONENT Row1 = Arm()
AT (0, 0.286, mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//2 row of the optical module:

COMPONENT Row2 = Arm()
AT (0, 0.348,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//3 row of the optical module:

COMPONENT Row3 = Arm()
AT (0, 0.411,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//4 row of the optical module:

COMPONENT Row4 = Arm()
AT (0, 0.473 ,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//5 row of the optical module:

COMPONENT Row5 = Arm()
AT (0, 0.535,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//6 row of the optical module:

COMPONENT Row6 = Arm()
AT (0, 0.597,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//7 row of the optical module:

COMPONENT Row7 = Arm()
AT (0, 0.659,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//8 row of the optical module:

COMPONENT Row8 = Arm()
AT (0, 0.722,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin
```

```
//9 row of the optical module:


COMPONENT Row9 = Arm()
AT (0, 0.784 ,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//10 row of the optical module:

COMPONENT Row10 = Arm()
AT (0, 0.846,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//11 row of the optical module:

COMPONENT Row11 = Arm()
AT (0, 0.908,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//12 row of the optical module:

COMPONENT Row12 = Arm()
AT (0, 0.970,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//13 row of the optical module:

COMPONENT Row13 = Arm()
AT (0, 1.032 ,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//14 row of the optical module:

COMPONENT Row14 = Arm()
AT (0, 1.095,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//15 row of the optical module:

COMPONENT Row15 = Arm()
AT (0, 1.157,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//16 row of the optical module:

COMPONENT Row16 = Arm()
```

```
AT (0, 1.219,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//17 row of the optical module:

COMPONENT Row17 = Arm()
AT (0, 1.281,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//18 row of the optical module:

COMPONENT Row18 = Arm()
AT (0, 1.344,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//19 row of the optical module:

COMPONENT Row19 = Arm()
AT (0, 1.406,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//20 row of the optical module:
COMPONENT Row20 = Arm()
AT (0, 1.468 ,mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin


COMPONENT para1 = ATHENA_deform(
width = 0.037096, length = 0.101504, gridn= N_deform,
   reflecn= N_reflec)
AT (0,0.000604636338/2,-0.101502199/2) RELATIVE Row1
ROTATED (0.3413,0,0) RELATIVE Row1
GROUP parabolic EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}

COMPONENT para2= ATHENA_deform(
width = 0.050158, length = 0.083388, gridn= N_deform+1,
   reflecn= N_reflec)
AT (0,0.00060442/2,-0.0833858/2) RELATIVE Row2
ROTATED ( 0.4153,0,0) RELATIVE Row2
GROUP parabolic
EXTEND
```

```
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para3 = ATHENA_deform(
width = 0.049838, length = 0.070762, gridn= N_deform+2,
   reflecn= N_reflec)
AT (0,0.000605655/2,-0.0707594/2) RELATIVE Row3
ROTATED (0.4904034,0,0) RELATIVE Row3
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para4 = ATHENA_deform(
width = 0.049613, length = 0.061460, gridn= N_deform+3,
   reflecn= N_reflec)
AT (0,0.000605314/2,-0.061457/2) RELATIVE Row4
ROTATED (0.564310030402610,0,0) RELATIVE Row4
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para5 = ATHENA_deform(
width = 0.089636, length = 0.054321, gridn= N_deform+4,
   reflecn= N_reflec)
AT (0,0.00060504/2,-0.0543176/2) RELATIVE Row5
ROTATED (0.638186,0,0) RELATIVE Row5
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para6 = ATHENA_deform(
```

```
width = 0.082746, length = 0.048671, gridn= N_deform+5,
   reflecn= N_reflec)
AT (0,0.000604831/2,-0.0486672/2) RELATIVE Row6
ROTATED (0.7120292,0,0) RELATIVE Row6
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}



COMPONENT para7 = ATHENA_deform(
width = 0.07751, length = 0.044087, gridn= N_deform+6,
   reflecn= N_reflec)
AT (0,0.000604651/2,-0.0440829/2) RELATIVE Row7
ROTATED ( 0.785833957320171,0,0) RELATIVE Row7
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para8 = ATHENA_deform(
width = 0.086892, length = 0.040294, gridn= N_deform+7,
   reflecn= N_reflec)
AT (0,0.000605336/2,-0.0402895/2) RELATIVE Row8
ROTATED (0.860786,0,0) RELATIVE Row8
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para9 = ATHENA_deform(
width = 0.082053, length = 0.037104, gridn= N_deform+8,
   reflecn= N_reflec)
AT (0,0.0006051452/2,-0.0370991/2) RELATIVE Row9
ROTATED (0.934502950810648,0,0) RELATIVE Row9
GROUP parabolic
```

```
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para10 = ATHENA_deform(
width = 0.090205, length = 0.034383, gridn= N_deform+9,
   reflecn= N_reflec)
AT (0,0.000587373/2,-0.03337782/2) RELATIVE Row10
ROTATED (1.008170036262145,0,0) RELATIVE Row10
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para11 = ATHENA_deform(
width = 0.037096, length = 0.032036, gridn= N_deform+10,
   reflecn= N_reflec)
AT (0,0.000604826/2,-0.0320303/2) RELATIVE Row11
ROTATED (1.081783740946868,0,0) RELATIVE Row11
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para12 = ATHENA_deform(
width = 0.092782, length = 0.02999, gridn= N_deform+11,
   reflecn= N_reflec)
AT (0,0.000604692/2,-0.0299839/2) RELATIVE Row12
ROTATED (1.155340238805293,0,0) RELATIVE Row12
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}
```

```
COMPONENT para13 = ATHENA_deform(
width = 0.088326, length = 0.028191, gridn= N_deform+12,
    reflecn= N_reflec)
AT (0,0.000604572/2,-0.0281845/2) RELATIVE Row13
ROTATED (1.228835721789498,0,0) RELATIVE Row13
GROUP parabolic
EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}

COMPONENT para14 = ATHENA_deform(
width = 0.088326, length = 0.026597, gridn= N_deform+13,
    reflecn= N_reflec)
AT (0,0.000605016/2,-0.0265901/2) RELATIVE Row14
ROTATED (1.303450214987052,0,0) RELATIVE Row14
GROUP parabolic
EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}

COMPONENT para15 = ATHENA_deform(
width = 0.088326, length = 0.025175, gridn= N_deform+14,
    reflecn= N_reflec)
AT (0,0.000604894/2,-0.0251677/2) RELATIVE Row15
ROTATED (1.376811185102159,0,0) RELATIVE Row15
GROUP parabolic
EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}

COMPONENT para16 = ATHENA_deform(
width = 0.088326, length = 0.023898, gridn= N_deform+15,
    reflecn= N_reflec)
AT (0,0.00060477/2,-0.0238903/2) RELATIVE Row16
ROTATED (1.450099775462344,0,0) RELATIVE Row16
GROUP parabolic
EXTEND
```

```
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para17 = ATHENA_deform(
width = 0.088326, length = 0.022738, gridn= N_deform+16,
   reflecn= N_reflec)
AT (0,0.000604672/2,-0.022738/2) RELATIVE Row17
ROTATED (1.523312261810409,0,0) RELATIVE Row17
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para18 = ATHENA_deform(
width = 0.037096, length = 0.0217, gridn= N_deform+17,
   reflecn= N_reflec)
AT (0,0.000605/2,-0.0216916/2) RELATIVE Row18
ROTATED ( 1.597623828661746,0,0) RELATIVE Row18
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para19 = ATHENA_deform(
width = 0.037096, length = 0.020748, gridn= N_deform+18,
   reflecn= N_reflec)
AT (0,0.000604899/2,-0.0207392/2) RELATIVE Row19
ROTATED ( 1.670671653378206,0,0) RELATIVE Row19
GROUP parabolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}

COMPONENT para20 = ATHENA_deform(
```

```
width = 0.037096, length = 0.019876, gridn= N_deform+19,
   reflecn= N_reflec)
AT (0,0.000604776/2,-0.0198668/2) RELATIVE Row20
ROTATED ( 1.743632288719842,0,0) RELATIVE Row20
GROUP parabolic
EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}


COMPONENT hyper1 = ATHENA_deform(
width = 0.037096, length = 0.101504, gridn= N_deform+20,
   reflecn= N_reflec)
AT (0,-0.00181400033/2,0.1014879/2) RELATIVE Row1
ROTATED (1.024,0,0) RELATIVE Row1
GROUP hyperbolic EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}

COMPONENT hyper2 = ATHENA_deform(
width = 0.050158, length = 0.083388, gridn= N_deform+21,
   reflecn= N_reflec)
AT (0,-0.00181304/2,0.0833683/2) RELATIVE Row2
ROTATED (1.24583403,0,0) RELATIVE Row2
GROUP hyperbolic
EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}

COMPONENT hyper3 = ATHENA_deform(
width = 0.049838, length = 0.070762, gridn= N_deform+22,
   reflecn= N_reflec)
AT (0,-0.00181679/2,0.0707387/2) RELATIVE Row3
ROTATED (1.47121024,0,0) RELATIVE Row3
GROUP hyperbolic
EXTEND
%{
```

```
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper4 = ATHENA_deform(
width = 0.049613, length = 0.061460, gridn= N_deform+23,
   reflecn= N_reflec)
AT (0,-0.00181571/2,0.0614332/2) RELATIVE Row4
ROTATED (1.69293,0,0) RELATIVE Row4
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper5 = ATHENA_deform(
width = 0.089636, length = 0.054321, gridn= N_deform+24,
   reflecn= N_reflec)
AT (0,-0.00181482/2,0.0542907/2) RELATIVE Row5
ROTATED (1.91455979,0,0) RELATIVE Row5
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper6 = ATHENA_deform(
width = 0.082746, length = 0.048671, gridn= N_deform+25,
   reflecn= N_reflec)
AT (0,-0.00181412 /2,0.0486372/2) RELATIVE Row6
ROTATED (2.136087617948129,0,0) RELATIVE Row6
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
```

```
%}

COMPONENT hyper7 = ATHENA_deform(
width = 0.07751, length = 0.044087, gridn= N_deform+26,
    reflecn= N_reflec)
AT (0,-0.0018135/2,0.0440497/2) RELATIVE Row7
ROTATED ( 2.357501871960515,0,0) RELATIVE Row7
GROUP hyperbolic
EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}


COMPONENT hyper8 = ATHENA_deform(
width = 0.086892, length = 0.040294, gridn= N_deform+27,
    reflecn= N_reflec)
AT (0,-0.00181546/2,0.0402531/2) RELATIVE Row8
ROTATED ( 2.582358982490732,0,0) RELATIVE Row8
GROUP hyperbolic
EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}

COMPONENT hyper9 = ATHENA_deform(
width = 0.082053, length = 0.037104, gridn= N_deform+28,
    reflecn= N_reflec)
AT (0,-0.00181479/2,0.0370596/2) RELATIVE Row9
ROTATED ( 2.803508852431944,0,0) RELATIVE Row9
GROUP hyperbolic
EXTEND
%{
     if (SCATTERED) {
          reflecs+=mcScattered;
     }
%}


COMPONENT hyper10 = ATHENA_deform(
width = 0.090205, length = 0.034383, gridn= N_deform+29,
    reflecn= N_reflec)
```

```
AT (0,-0.00181416/2,0.03433512/2) RELATIVE Row10
ROTATED (3.024510108786434,0,0) RELATIVE Row10
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper11 = ATHENA_deform(
width = 0.037096, length = 0.032036, gridn= N_deform+30,
   reflecn= N_reflec)
AT (0,-0.00181362/2,0.0319846/2) RELATIVE Row11
ROTATED (3.245351222840604,0,0) RELATIVE Row11
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}



COMPONENT hyper12 = ATHENA_deform(
width = 0.092782, length = 0.02999, gridn= N_deform+31,
   reflecn= N_reflec)
AT (0,-0.00181309/2,0.0299351/2) RELATIVE Row12
ROTATED (3.466020716415880,0,0) RELATIVE Row12
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper13 = ATHENA_deform(
width = 0.092782, length = 0.028191, gridn= N_deform+32,
   reflecn= N_reflec)
AT (0,-0.00181261/2,0.0281327/2) RELATIVE Row13
ROTATED (3.686507165368494,0,0) RELATIVE Row13
GROUP hyperbolic
```

```
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper14 = ATHENA_deform(
width = 0.088326, length = 0.026597, gridn= N_deform+33,
   reflecn= N_reflec)
AT (0,-0.0018138/2,0.0265351/2) RELATIVE Row14
ROTATED (3.910350644961155,0,0) RELATIVE Row14
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}



COMPONENT hyper15 = ATHENA_deform(
width = 0.088326, length = 0.025175, gridn= N_deform+34,
   reflecn= N_reflec)
AT (0,-0.00181329/2,0.0251096/2) RELATIVE Row15
ROTATED (4.130433555306476,0,0) RELATIVE Row15
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}



COMPONENT hyper16 = ATHENA_deform(
width = 0.088326, length = 0.023898,gridn= N_deform+35,
   reflecn= N_reflec)
AT (0,-0.00181276/2,0.0238291/2) RELATIVE Row16
ROTATED (4.350299326387032,0,0) RELATIVE Row16
GROUP hyperbolic
EXTEND
%{
```

```
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper17 = ATHENA_deform(
width = 0.088326, length = 0.022738, gridn= N_deform+36,
   reflecn= N_reflec)
AT (0,-0.00181167/2,0.0226657/2) RELATIVE Row17
ROTATED (4.569936785431228,0,0) RELATIVE Row17
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper18 = ATHENA_deform(
width = 0.037096, length = 0.0217, gridn= N_deform+37,
   reflecn= N_reflec)
AT (0,-0.00181312/2,0.0216241/2) RELATIVE Row18
ROTATED (4.792871485985237,0,0) RELATIVE Row18
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


COMPONENT hyper19 = ATHENA_deform(
width = 0.037096, length = 0.020748, gridn= N_deform+38,
   reflecn= N_reflec)
AT (0,-0.00181264/2,0.0206687/2) RELATIVE Row19
ROTATED ( 5.012014960134620,0,0) RELATIVE Row19
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
```

```
    }
%}

COMPONENT hyper20 = ATHENA_deform(
width = 0.037096, length = 0.019876, gridn= N_deform+39,
    reflecn= N_reflec)
AT (0,-0.00181209/2,0.0197932/2) RELATIVE Row20
ROTATED ( 5.230896866159527,0,0) RELATIVE Row20
GROUP hyperbolic
EXTEND
%{
    if (SCATTERED) {
        reflecs+=mcScattered;
    }
%}


//this is the part of the instrument where the monitors
    are inserted. first we use a PSD monitor to record the
    locations of the photons
// this monitor only chacks for photons which scattered
    off a surface
COMPONENT final_monitor = PSD_monitor(
xwidth=0.5, yheight= 0.5, ny=500000,nx=1,filename="
    final_psd.dat")
WHEN(reflecs=2)AT (0,0,12+mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin

//monitor_nd is intended to monitor the divergence of the
    incident photons
//monitor_nD has trouble working for some reason
/*
COMPONENT div_mon = Divergence_monitor(nh=5000, nv=1,
    filename="Output.pos",
xwidth=1, yheight=1,
maxdiv_h=2, maxdiv_v=2, restore_xray=1)
WHEN(reflecs>0)AT (0,0,12+mirror_distance) RELATIVE Origin
ROTATED (0,0,0) RELATIVE Origin
*/
FINALLY
%{

%}
END
```