Optical and Thermal Radiative Simulation of an Earth Radiation Budget Instrument

Joel Seth Fronk

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Mechanical Engineering

James Robert Mahan. Chair
Vinh Nguyen
Brian Vick

May 6, 2021
Blacksburg, Virginia

Keywords: Earth Radiation Budget, DEMETER, Ray-Tracing
ABSTRACT

Researchers at the NASA Langley Research Center (LaRC) are developing a next-generation instrument for monitoring the Earth radiation budget (ERB) from low Earth orbit. This instrument is called the DEMonstrating the Emerging Technology for measuring the Earth’s Radiation (DEMETER) instrument. DEMETER is a candidate to replace the Clouds and Earth’s Radiant Energy System (CERES) instruments which currently monitor the ERB. LaRC has partnered with the Thermal Radiation Group at Virginia Tech to model and evaluate the thermal and optical design of the DEMETER instrument. The effort reported here deals with the numerical modeling of the optical and thermal radiative performance the DEMETER instrument. The numerical model is based on the Monte Carlo Ray-Trace (MCRT) method. The major optical components of the instrument are incorporated into the ray-trace model using 3-D surface equations. A CAD model of the instrument baffle is imported directly into the ray-trace environment using an STL triangular mesh. The instrument uses a single freeform mirror to focus radiation on the detector. A method for incorporating freeform surfaces into a ray-trace model is described. The development and capabilities of the model are reported. The model is used to run several ray-traces to compare two different quasi-black surface coatings for the DEMETER telescope baffle. Included is a list of future tests the Thermal Radiation Group will use the model to accomplish.
For decades NASA has used satellite-mounted scientific instruments to monitor the Earth radiation budget (ERB). The ERB is the energy balance of the planet Earth with its surroundings. Radiation from the sun is absorbed and reflected by the Earth. The Earth also emits radiation. The balance between these heat transfer components drives the planetary climate. Researchers at the NASA Langley Research Center (LaRC) are developing a new instrument for monitoring the ERB from low Earth orbit. This Earth observing instrument is called the DEMonstrating the Emerging Technology for measuring the Earth’s Radiation (DEMETER) instrument. NASA has partnered with the Thermal Radiation Group at Virginia Tech to model and evaluate the thermal and optical design of the DEMETER instrument. The effort reported here deals with the numerical modeling of radiation heat transfer in the DEMETER instrument. The numerical model uses the Monte Carlo Ray-Trace (MCRT) method to evaluate the thermal and optical behavior of the DEMETER instrument. The development and capabilities of the model are reported. The model is used to run a series of simulations to compare the performance of two different quasi-black surface coatings for the DEMETER telescope baffle. Included is a list of future tasks the Thermal Radiation Group will accomplish using the model.
Acknowledgments

First, I would like to thank my advisor Dr. J. Robert Mahan for his unfailing support and guidance throughout the course of my studies at Virginia Tech. His infectious enthusiasm for academia motivated me to dive deeper into topics. He has given me a greater appreciation for the acquisition of knowledge.

I would also like to thank Dr. Brian Vick and Dr. Vinh Nguyen for serving on my committee. Their perspectives and feedback were invaluable. More than anything, I appreciate that they took genuine interest in my research work.

I would like to thank our research collaborators at the NASA Langley Research Center, Dr. Kory J. Priestly and Dr. Anum Barki Ashraf. I thank them for the opportunities they have given my fellow students and me to work with real-life and impactful Engineering problems.

I would like to thank Adam Phenis of Quartus Engineering, Inc. for his tireless efforts in working with me to correct errors in my numerical model of the DEMETER freeform mirror. I could not have corrected my model in time without his help.

I would like to thank the many students and faculty members at Virginia Tech and other members of the Blacksburg community who selflessly offered me friendship and help with my research work and course work. Finally, I would like to thank my family. Without their encouragement and support, I would never have completed this thesis.
Table of Contents

ABSTRACT ..................................................................................................................... ii
GENERAL AUDIENCE ABSTRACT ........................................................................... iii
Acknowledgments ........................................................................................................... iv
Table of Contents ............................................................................................................. v
List of Figures ............................................................................................................... viii

Chapter 1: Introduction ............................................................................... 1
1.1 Earth Radiation Budget .............................................................................................. 1
1.2 Clouds and Earth’s Radiant Energy System (CERES) Instrument ......................... 3
1.3 DEMonstrating the Emerging Technology for measuring Earth’s Radiation (DEMETER) Instrument .................................................................................................. 4
1.4 The Thermal Radiation Group at Virginia Polytechnic Institute & State University 6

Chapter 2: Modelling DEMETER .............................................................. 8
2.1 DEMETER System Overview ................................................................................... 8
2.2 Model Objectives ....................................................................................................... 9
2.2.1 Visualize the Path of Radiation Passing Through the Instrument ...................... 9
2.2.2 Calculate the Radiation Distribution Factors from the Entrance Aperture to the Baffle Surface Elements .............................................................................................. 10
2.2.3 Evaluate the Effects of Using Different Surface Coatings for the Baffle ............. 10
2.2.4 Collect the Point-Spread Function (PSF) for Beam Radiation on the Focal-Plane Array .............................................................................................................................. 11
2.2.5 Distinguish Between Regular Rays and Stray Rays Collected by the Focal-Plane Array .............................................................................................................................. 12
2.3 Monte Carlo Ray-Trace Method .............................................................................. 12
2.4 DEMETER 3-D Model ............................................................................................ 12

Chapter 3: DEMETER Component Surface Equations ............................. 15
3.1 Surface Equations .................................................................................................... 15
3.2 Entrance Aperture ................................................................................................... 15
3.3 The Baffle ................................................................................................................ 19
3.3.1 CAD Model Development .................................................................................... 19
3.3.2 Model Meshing ..................................................................................................... 21
3.3.3 Reading the Model ................................................................................................ 29
List of Figures

Figure 1.1 Earth radiation budget [2] ........................................................................................................... 1

Figure 1.2 Typical ERB data products, (a) showing shortwave radiation, and (b) longwave radiation [3]. ................................................................................................................................. 3

Figure 1.3 CERES instrument in a scanning orbit [3] ................................................................................ 4

Figure 1.4 CERES and DEMETER operational comparison [6] .................................................................... 5

Figure 2.1 DEMETER instrument (a) full model and (b) cut-away view showing the optical path [8] ........................................................................................................................................ 8

Figure 2.2 Demeter with components numbered [8] ................................................................................ 9

Figure 2.3 DEMETER instrument ray-trace model ..................................................................................... 13

Figure 3.1 Entrance aperture ...................................................................................................................... 16

Figure 3.2 Aperture divided into two shapes with rays distributed by relative area. .............................. 17

Figure 3.3 Rays generated in aperture using (a) $10^2$ rays (b) $10^3$ rays (c) $10^4$ rays (d) $10^5$ rays (e) $10^6$ rays. ............................................................................................................................................... 18

Figure 3.4 CAD model of the DEMETER baffle ..................................................................................... 20

Figure 3.5 Side elevation cut-away view of the baffle .............................................................................. 20

Figure 3.6 Plan cut-away view of the baffle ............................................................................................... 20

Figure 3.7 Several views of a baffle triangular mesh example. Panels (a) and (b) show the same view without and with the mesh. Panel (c) is an isometric view of the baffle mesh. This mesh consists of 84,774 elements. ................................................................................... 23

Figure 3.8 Several views of a baffle triangular mesh example. Panels (a) and (b) show the same view without and with the mesh. Panel (c) is an isometric view of the baffle mesh. This mesh consists of 2,304 elements. ........................................................................................................ 25
Figure 3.9 Meshing scheme from with a single element highlighted in red ...................... 27

Figure 3.10 Selected baffle mesh. Panels (a) and (b) show the same view without and with the mesh. Panels (c) and (d) are different isometric views of the baffle mesh. Panel (e) is a cut-away view showing the meshing on the inside of the baffle. This mesh consists 10,590 elements .......................................................... 28

Figure 3.11 Baffle mesh in Matlab .......................................................... 31

Figure 3.12 Baffle mesh in Matlab. This is the same Mesh as in Fig. 3.11, but the mesh edges have been suppressed and the mesh fill set to transparent so that all baffle vanes are clearly visible .......................................................... 32

Figure 3.13 Ray intersection point inside a triangular element defined by three vectors whose sum is zero .......................................................... 33

Figure 3.14 Block of material with sag profile shown ...................................... 34

Figure 3.15 Cut-away view of the spherical mirror trimmed to DEMETER size .......... 37

Figure 3.16 Cut-away view of Spherical mirror with even aspheric corrections ........ 39

Figure 3.17 Cut-away view of the freeform mirror used in the DEMETER instrument . 45

Figure 3.18 Ray-trace model with chief ray shown in red .................................. 47

Figure 3.19 Comparison of PSF plots for a beam angle of 0 deg. Panel (a) is the PSF obtained by Quartus Engineering and has units of micrometers [24]. Panel (b) is the PSF from the ray-trace code and has units of millimeters .......................................................... 48

Figure 3.20 Comparison of PSF plots for a beam angle of 8 deg. Panel (a) is the PSF obtained by Quartus Engineering and has units of micrometers [24]. Panel (b) is the PSF from the ray-trace code and has units of millimeters .......................................................... 49
Figure 3.21 Comparison of PSF plots for a beam angle of 16 deg. Panel (a) is the PSF obtained by Quartus Engineering and has units of micrometers [24]. Panel (b) is the PSF from the ray-trace code and has units of millimeters.

Figure 4.1 Ray-trace logic flow chart.

Figure 4.2 Ray-trace 3-D model with 1,000 rays. No additional graphical components are activated.

Figure 4.3 Ray-traces for a beam angle of 0 deg with (a) 10 rays, (b) 100 rays, and (c) 1,000 rays. Ray paths are visible. The figure is in the mirror local coordinate system.

Figure 4.4 Cut-away view of a ray-trace for a beam angle of 0 deg with 1,000,000 rays. Power flux highlights are visible.

Figure 4.5 Ray-trace for a beam angle of 0 deg with 1,000 rays. Both power flux highlights and ray paths are visible.

Figure 4.6 Focal-plane point-spread function (PSF) plot for a beam angle of 0 deg with 100,000 rays.

Figure 5.1 Ray paths with 10,000 rays and a beam angle of 0 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$.

Figure 5.2 Ray paths with 10,000 rays and a beam angle of 8 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$.

Figure 5.3 Ray paths with 10,000 rays and a beam angle of 16 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$.

Figure 5.4 Power flux highlights with 100,000 rays and a beam angle of 0 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$.

Figure 5.5 Cut-away views of Fig. 5.4. The beam angle is 0 deg.
Figure 5.6 Front elevation view of panels (a) and (b) of Fig. 5.4. The beam angle is 0 deg.  
A color bar has been added to indicate power flux. The units for the color bar are rays per  
square millimeter. ................................................................. 76  

Figure 5.7 Side elevation cut-away view of Fig. 5.4 (b). The beam angle is 0 deg........... 78  

Figure 5.8 Power flux highlights with 100,000 rays and a beam angle of 8 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. ......................... 79  

Figure 5.9 Cut-away views of Fig. 5.8. The beam angle is 8 deg......................... 80  

Figure 5.10 Power flux highlights with 100,000 rays and a beam angle of 16 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. ......................... 81  

Figure 5.11 Cut-away views of Fig. 5.10. The beam angle is 16 deg.......................... 82  

Figure 5.12 Point-spread function plots for a ray-trace with 100,000 rays and a beam  
angle of 0 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. ... 84  

Figure 5.13 Point-spread function plots for a ray-trace with 100,000 rays and a beam  
angle of 8 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. ... 85  

Figure 5.14 Point-spread function plots for a ray-trace with 100,000 rays and a beam  
angle of 16 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. . 86  

Figure 5.15 Power flux highlights with 1,000,000 rays and a beam angle of 0 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. ......................... 89  

Figure 5.16 Cut-away views of Fig. 5.15. The beam angle is 0 deg.......................... 90  

Figure 5.17 Front elevation view of panels (a) and (b) of Fig. 5.15. The beam angle is 0  
deg. A color bar has been added to indicate power flux................................. 92  

Figure 5.18 An isometric view of the 3-D model in Fig. 5.15 panel (a). The beam angle is  
0 deg. A color bar has been added to indicate power flux................................. 93
Figure 5.19 Point-spread function plots for a ray-trace with 1000,000 rays and a beam angle of 0 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. ... 94
Chapter 1: Introduction

1.1 Earth Radiation Budget

The motivation for this work is the need to monitor the Earth radiation budget (ERB). The ERB is the energy balance for the planet as a body in space [1]. The only way energy leaves and enters the Earth from space is through radiation heat transfer. This is why the energy balance is referred to as the radiation budget. Figure 1.1 illustrates the components of the ERB.

![Earth Radiation Components](image)

Figure 1.1 Earth radiation budget [2].
The sun radiates energy to the Earth. Most of this radiant energy lies in the shortwave part of the electromagnetic spectrum. Some of that shortwave radiation is reflected off the Earth’s surface and atmosphere. The rest is absorbed by the Earth. The Earth also loses energy through emission. This emission is mostly longwave radiation [1]. In order to maintain a consistent climate and habitable living conditions, a balance between these major components must prevail.

To quantitatively evaluate the ERB it must be monitored over time. NASA has monitored the ERB for decades. Since the 1980s, NASA has been using satellite-based instruments to measure the amount of radiation reflected and emitted from the Earth [3]. NASA is able to track and categorize changes in the ERB by detecting the radiation emitted and reflected from the Earth.

While the design and capabilities of these ERB instruments have evolved over the years, the basic method has remained constant. The ERB instrument behaves like a camera pointed at the Earth. As the instrument orbits the Earth, the camera faces different places on the planetary surface. The light emitted and reflected from the Earth enters the ERB instrument and is focused on to a detector array. When radiation is absorbed on a pixel of the array, an electrical signal is created which can be interpreted as an image pixel.

As the instrument passes over the entire Earth, it generates a thermal image, created from processing the radiation absorbed by the detector. Figure 1.2 shows two images, short wavelength and long wavelength, generated by orbiting ERB instruments. The detector is filtered so that it can distinguish between these two wavelength bands of radiation. This permits the radiation to be separated into reflected and emitted components.
Figure 1.2 Typical ERB data products, (a) showing shortwave radiation, and (b) longwave radiation [3].

1.2 Clouds and Earth’s Radiant Energy System (CERES) Instrument

Most recently, NASA has been operating an ERB instrument design called the Clouds and Earth’s Radiant Energy System (CERES). The CERES program has been instrumental in providing Earth radiation emission and reflection data as well as indicating the role of atmospheric clouds in the ERB. Since 1997 NASA has always had at least one CERES
instrument in orbit collecting data [3]. Each CERES instrument consists of a telescope and
detector system mounted on a satellite shared by other scientific instruments. The instrument
uses a two-mirror system to focus and direct radiation on to the instrument detector [4].

The CERES instrument is a scanning instrument, meaning the detector sweeps back and
forth to collect radiation data as it orbits the Earth. Figure 1.3 shows an example of the scanning
operation of the CERES.

![Cross-track Scanning](image)

Figure 1.3 CERES instrument in a scanning orbit [3].

By scanning along its orbit track, the instrument is able to measure the radiation over a wider
view angle than it would if it stared in one direction.

1.3 DEMonstrating the Emerging Technology for measuring Earth’s Radiation

(DEMETER) Instrument

NASA will continue to monitor the ERB from orbit in the decades to come. Experience
and emerging technology allow for design upgrades to ERB monitoring systems. In 2019, Dr.
Anum Ashraf and her team at the NASA Langley Research Center (LaRC) received funding
under the Instrument Incubator Program (IIP) to demonstrate new technology for an improved ERB instrument. The instrument would be called DEMETER, which stands for DEMonstrating the Emerging Technology for measuring the Earth’s Radiation [4, 5].

The proposed DEMETER instrument would improve upon the existing technology by decreasing instrument package size, mass, and cost, while increasing sensing quality and instrument durability [6, 7].

Unlike CERES, DEMETER would be a non-scanning, or staring, telescope. Instead of sweeping back and forth as it orbits, it would point in a fixed direction, as illustrated in Fig. 1.4. The DEMETER instrument would feature a wide view angle to eliminate the need to scan back and forth.

![Figure 1.4 CERES and DEMETER operational comparison [6].](image)

This would reduce the power consumption of the instrument since it would not need to spend power on the scanning motion. This would also reduce the risk of mechanical failure inherent to systems with moving parts. While previous ERB instruments had to be embarked on larger host satellites with other scientific instruments, DEMETER would be carried by its own
small free-flying satellite. The use of this small satellite would decrease cost and size of the project. It also would free the project from the limitations related to sharing space on a larger satellite with other instruments with different objectives [6].

While maintaining continuity with observations from CERES, DEMETER would improve on its predecessors by increasing the spatial resolution of ERB measurements by a factor of 10 [5]. DEMETER would also make use of a 2-D detector array (multiple rows of pixels) to collect unique information (spectral and polarization state) in a single pass [6].

In order to create the proposed DEMETER instrument, NASA LaRC partnered with Quartus Engineering Inc., NovaWurks Inc., Science Systems and Applications Inc., and The Thermal Radiation Group at Virginia Polytechnic Institute & State University [5]. The combined experience and expertise of these partners would assure the DEMETER project’s success.

1.4 The Thermal Radiation Group at Virginia Polytechnic Institute & State University

The Thermal Radiation Group at Virginia Polytechnic Institute & State University (Virginia Tech TRG) has been involved with NASA in their effort to monitor the ERB for decades [4]. The TRG has contributed end-to-end numerical models of the CERES instrument and other ERB instruments and conducted research on different elements of the ERB detecting process. Their numerical models and research have been useful in performance evaluation of actual instruments before their implementation and has led to recommendations for new solutions and concepts for solving problems related to the ERB [4].

The TRG continues to participate in the effort to monitor the ERB by assisting NASA LaRC in the DEMETER project. This thesis documents a part of the effort undertaken by the TRG to support DEMETER; namely, the development of a numerical model of the DEMETER instrument system to help evaluate the optical and thermal radiative performance of its design.
The TRG worked with Quartus Engineering to understand and model the DEMETER design. The TRG then used its numerical modelling expertise to develop a performance model. The TRG will continue to expand the DEMETER model capability beyond this thesis, and also use the methods developed here to assist in future ERB activities.
Chapter 2: Modelling DEMETER

2.1 DEMETER System Overview

Figure 2.1 (a) shows a CAD model of the evolving DEMETER instrument. Figure 2.1 (b) is a cut-away image revealing the optical path inside the instrument. Note that this preliminary baffle design is expected to evolve as the DEMETER concept matures.

![Figure 2.1 DEMETER instrument (a) full model and (b) cut-away view showing the optical path](image)

Figure 2.1 DEMETER instrument (a) full model and (b) cut-away view showing the optical path [8].

Figure 2.2 is a cut-away side view of the instrument with key components numbered. Radiation enters the detector though the slot shaped aperture (1). Entering radiation is formed into a beam by a baffle system (2). Radiation that passes through the baffle exit (3) is reflected and focused by a freeform mirror (4) and further formed by a smaller set of baffle vanes and an
array of precision apertures (5) before it is ultimately absorbed by a focal-plane detector array (6).

Figure 2.2 Demeter with components numbered [8].

2.2 Model Objectives

Modelling of the DEMETER instrument concept was guided by several design objectives. These objectives determined the components incorporated and the assumptions made. The objectives were as follows:

- Visualize the path of radiation passing through the instrument.
- Calculate the radiation distribution factors from the entrance aperture to the baffle surface elements.
- Evaluate the effects of using different surface coatings for the baffle.
- Collect the point-spread function (PSF) for beam radiation on the focal-plane array (FPA).
- Distinguish regular rays and stray rays collected by the focal-plane array.

2.2.1 Visualize the Path of Radiation Passing Through the Instrument

By visualizing the path of radiation, the soundness of the design geometry can be evaluated. Visualizing the ray paths would show the effectiveness of the baffle in eliminating stray light incident to the detector array. It would also help determine if it was possible for rays
representing the scene to exit the baffle without reaching the focal-plane. That is, the optical efficiency of the instrument could be evaluated.

2.2.2 Calculate the Radiation Distribution Factors from the Entrance Aperture to the Baffle Surface Elements

Radiation distribution factors indicate the optical sensitivity of all surfaces to a given surface [9]. Eventually it would be useful to know the sensitivity of all surfaces in the system to all other surfaces. This would be essential for a conduction heat transfer analysis. For the present, the scope of the thesis is limited to evaluating the sensitivity of all surfaces to radiation entering through the aperture.

2.2.3 Evaluate the Effects of Using Different Surface Coatings for the Baffle

The purpose of the baffle is to form the incoming radiation into a beam. With each vane of the baffle, more radiation is blocked, until the beam is formed into the desired shape. Ideally, all of the surfaces of the baffle would be perfectly black, meaning they absorb all radiation that comes in contact with them. In reality, the baffle surfaces will reflect some of the radiation that strikes them. This radiation will be reflected both diffusely and speculally. The amount of radiation that is absorbed by the surfaces and the amount that is reflected is determined by the coating used on the baffle surfaces.

2.3.3.1 Surface Coatings

In optical systems, surface coatings can be used to control the absorptive and reflective behavior of surfaces [10]. The surface coating used for the baffle of the DEMETER instrument will affect the performance of the instrument. The reflective and absorptive behavior of a surface coating can be quantified using two properties: absorptivity and specularity ratio.
2.3.3.2 Absorptivity and Specularity Ratio

The absorptivity of the surface indicates the degree to which the surface absorbs radiation at a given wavelength. If the absorptivity is zero for a given wavelength, the surface does not absorb any radiation. If the absorptivity is unity, it will absorb all incident radiation at that wavelength [9].

The specularity ratio of a surface indicates the degree to which reflections are specular at a given wavelength. A specularity ratio of zero means the surface reflections at that wavelength are completely diffuse. In other words, the radiation is scattered following Lambert’s cosine law [9]. A specularity ratio of unity means that all reflections at that wavelength are completely specular. Any value between zero and unity would imply a combination of diffuse and specular reflection [9].

The specularity ratio and absorptivity of the baffle surfaces are adjustable in the model developed here so the user can easily study the effect of different surface finishes on system performance. Clear graphical representations of the optical performance are needed in order to effectively evaluate these differences.

2.2.4 Collect the Point-Spread Function (PSF) for Beam Radiation on the Focal-Plane Array

A point-spread function is the response of an optical system to a point source of radiation [11]. The response is characterized by the spatial distribution of radiation in the focal plane. In the context of the DEMETER instrument, the point-spread function is the distribution of radiation collected on the focal-plane detector due to a collimated beam of radiation entering the instrument aperture at a given angle. By collecting the point-spread function at different angles, the optical performance of the freeform mirror can be evaluated. The point-spread function
indicates the degree of focus achieved by the mirror. The angle of radiation entering the aperture is also adjustable so that optical performance can be evaluated at different discrete entrance angles.

2.2.5 Distinguish Between Regular Rays and Stray Rays Collected by the Focal-Plane Array

With regard to the PSF, it would be important to distinguish between regular rays and stray rays. In context of this thesis, regular rays are those that enter the aperture and reach the focal-plane directly without first being reflected from a baffle surface. Stray rays are rays that would have been absorbed by the baffle had they been perfectly black but have escaped and reached the focal-plane. Stray rays add noise to the detector response and, if too prevalent, can skew results. By highlighting the stray rays in the point-spread function, the effectiveness of the baffle geometry and surface coating can be evaluated. This also allows scattered stray rays to be distinguished from poor focus.

2.3 Monte Carlo Ray-Trace Method

Based on these objectives, the Monte Carlo ray-trace (MCRT) method has been used to evaluate the DEMETER detector. All programming for the model was completed in the MathWorks® Matlab coding environment. The Matlab program written in the course of this project is included in Appendix C. A basic understanding of the MCRT method is assumed in subsequent sections. For more details on the MCRT method, refer to The Monte Carlo Ray-Trace Method in Radiation Heat Transfer and Applied Optics by Mahan [9].

2.4 DEMETER 3-D Model

Figure 2.3 shows a 3-D representation of the model of the DEMETER instrument used in the ray-trace. As previously stated, the details of the design are likely to evolve as the instrument
matures. Therefore, the modeling approach had to be sufficiently flexible to allow design changes to be easily accommodated. As seen in the figure, the detector was divided into four main components: the entrance aperture (far right entrance), the baffle (transparent blue), the mirror (magenta), and the focal-plane array (transparent yellow).

Small baffle vanes and a precision aperture array may be included between the mirror and the focal-plane, but they are not included in the model. The current version of the model is more focused on evaluating the effectiveness of the baffle in front of the mirror. Therefore, it was assumed that any radiation that reaches the mirror also reaches the focal-plane. Additionally, the enclosure surrounding the mirror has not been included in this model. Instead, the entire model is surrounded by an imaginary black cubic enclosure that collects any rays that escape the baffle but also miss the mirror. The baffle and mirror are oriented such that only stray rays can reach the walls of the imaginary enclosure.
The rays are traced using the same basic algorithm outlined in Mahan [9] for diffuse-specular gray surfaces. Ray sources are uniformly distributed in the entrance aperture of the baffle. They enter the baffle as a collimated beam. The rays can be reflected or absorbed in the baffle. If a ray is absorbed inside the baffle, the distribution factor is incremented for the baffle element containing the point where the ray is absorbed. While only collimated beam radiation is allowed to enter the instrument in the current study, any radiation distribution, including diffuse, can be introduced.

If a ray is not absorbed in the baffle, it either exits the baffle through the aperture or through the baffle exit at the mirror end of the baffle. If a ray leaves through the aperture, it is no longer traced and the distribution factor for the aperture is incremented. A ray that leaves through the baffle exit either strikes the mirror and is reflected to the focal-plane or misses the mirror completely and is absorbed by the imaginary black enclosure surrounding the instrument. A ray intercepted by the mirror is reflected to the focal-plane. Stray rays that reach the focal-plane are tagged so they can be distinguished from regular rays.
Chapter 3: DEMETER Component Surface Equations

3.1 Surface Equations

To incorporate the DEMETER system into a ray-trace, a surface equation (or equations) is required for each component. A surface equation is an expression of the form,

\[ f(x, y, z) = 0. \] 

(3.1)

In this expression, \( f \) is a function of \( x, y, \) and \( z \) which is defined for all points on the surface. Once such an expression is defined for a surface, the surface can be incorporated into a ray-trace.

Both the aperture and the focal-plane array are defined as planar surfaces. The baffle is defined as a mesh of triangular plane elements. The mirror is the only component defined with a non-planar surface equation. A freeform polynomial summation is used for the mirror. The derivation and features of the surface equations of each of these components are now described in detail.

3.2 Entrance Aperture

Radiation enters DEMETER through a slot-shaped entrance aperture at the front of the baffle. The aperture differs from the other components in that it is not actually a physical surface
but rather the absence of a surface. However, for the purposes of the model, it behaves like a solid surface in that it emits and absorbs rays.

Figure 3.1 shows the shape of the slot aperture. The length of the rectangular midsection is $L_{\text{rec}}$. The width of the rectangular midsection (also the diameter of the rounded ends) is $W_{\text{rec}}$. The aperture is centered on the coordinate system origin and lies in the $x$-$y$ plane.

![Figure 3.1 Entrance aperture.](image)

For the ray-trace, $N_{\text{ray}}$ rays are generated inside the entrance aperture. The rays need to be uniformly distributed throughout the whole slot. To accomplish this, the aperture is divided into two shapes: a $W_{\text{rec}}$-by-$L_{\text{rec}}$ rectangle, and a circle with a diameter of $W_{\text{rec}}$, as illustrated in Fig. 3.2. The number of rays generated in each shape is proportional to its area. If $A_{\text{circ}}$ is the area of the circle, and $A_{\text{rec}}$ is the area of the rectangle, then the number of rays generated in the rectangle is

$$N_{\text{rec}} = N_{\text{ray}} \times \frac{A_{\text{rec}}}{A_{\text{rec}} + A_{\text{circ}}}, \quad (3.2)$$

where $N_{\text{rec}}$ is rounded to the nearest whole number. Since the sum of the rays in the rectangle and the circle must equal the total number of rays, the number of rays generated in the circle can be expressed as

$$N_{\text{circ}} = N_{\text{ray}} - N_{\text{rec}}. \quad (3.3)$$
Figure 3.2 Aperture divided into two shapes with rays distributed by relative area.

Next, $N_{rec}$ rays are randomly distributed in the rectangle and $N_{circ}$ rays are randomly distributed in the circle. For the rectangle, the coordinates of a random point $p(x, y)$ are

$$p_x = \left( R_x - \frac{1}{2} \right) L_{rec} \tag{3.4}$$

and

$$p_y = \left( R_y - \frac{1}{2} \right) W_{rec}, \tag{3.5}$$

where $R_x$ and $R_y$ are random numbers (different in each instance) between 0 and 1. For the circle [9],

$$p_x = r \cos \theta \tag{3.6}$$

and

$$p_y = r \sin \theta, \tag{3.7}$$

where

$$r = \frac{W_{rec}}{2} \sqrt{R_r} \tag{3.8}$$

and

$$\theta = 2\pi R_\theta. \tag{3.9}$$

Once rays are randomly distributed in the two parts of the entrance aperture, they can be recombined to make the slot. This is done by splitting the circle and adding half of it to either
end of the rectangle. If $p_x$ for a ray point in the circle is greater than zero, $\frac{L_{rec}}{2}$ is added to it. If it is less than zero, $\frac{L_{rec}}{2}$ is subtracted from it. At this point, the rays are randomly distributed in a slot shape.

To be sure no pattern could be detected in the distribution of the rays, the aperture was populated using several different values of $N_{ray}$. Figure 3.3 (a) - (e) shows the aperture populated with $10^2, 10^3, 10^4, 10^5,$ and $10^6$ rays, respectively. These figures exhibit no discontinuities between the rounded end and the middle section. Rather, they appear to have been populated with rays as a single surface.

![Figure 3.3](image)

Figure 3.3 Rays generated in aperture using (a) $10^2$ rays (b) $10^3$ rays (c) $10^4$ rays (d) $10^5$ rays (e) $10^6$ rays.
Once the ray points are generated, the ray directions must be determined. For the model, the rays are assumed to enter the aperture as a collimated beam. The direction of the collimated beam can be on-axis, (perpendicular to the aperture and along the positive \( z \)-axis) or rotated around the \( y \)-axis. The beam should not be rotated away from the \( z \)-axis by more than 16 deg. The baffle has a view angle of 32 deg (16 deg on either side of the \( z \)-axis).

3.3 The Baffle

The next component to define is the baffle. To incorporate the baffle into the ray-trace, its shape was approximated using a triangular 3-D mesh. First a CAD model of the baffle was created, then the CAD model was converted into a 3-D STL mesh file. Finally, the mesh was read into the Matlab code as a collection of triangular plane surfaces. The following sections treat the details of this process.

3.3.1 CAD Model Development

The baffle CAD model for this project was created using a program called FreeCAD [12]. Any CAD modelling software could be used for this part of the process. Images provided by Quartus Engineering, Inc [8]. were used to create an approximate CAD facsimile of a preliminary baffle design. Figure 3.4 is an image of the CAD model, and Figs. 3.5 and 3.6 are cut-away images of the same model.
Figure 3.4 CAD model of the DEMETER baffle.

Figure 3.5 Side elevation cut-away view of the baffle.

Figure 3.6 Plan cut-away view of the baffle.
3.3.2 Model Meshing

Once a CAD model had been created, it needed to be converted to a mesh that could be read into Matlab. A common type of meshing in 3-D modelling is STL meshing. STL meshes are used in 3-D printing and other modelling applications. In STL meshing, 3-D sculptures, or CAD models, are divided into triangular planar elements that approximate the shape of the original model [13]. A finer triangular mesh results in a better approximation. These meshes can be easily sliced and converted to G-code commands for a 3-D printer.

STL meshes are ubiquitous, and so an excellent choice of format for importing CAD models into a ray-tracing environment. Any 3-D file can be converted into an STL file which means that if a ray-trace can be conducted in an arbitrary STL file, a ray-trace can be conducted in virtually any 3-D model provided without having to change the ray-trace code. This feature will be extremely useful for tracing rays through the baffle system. If any modifications need to be made to the 3-D model, even radical changes, the ray-trace code will not need to be modified before rays can be traced through the new baffle design. Therefore, the ray-trace code developed here is completely general. The STL format is compatible with the standard ray-tracing algorithm, as shown below.

FreeCAD contains a toolbox dedicated to creating STL meshes from CAD models. In the toolbox, the user can adjust some parameters to control the fineness and pattern of the mesh, and the software then automatically meshes the model [14]. A disadvantage is that the user does not have complete control over the mesh pattern. The user may give basic constraints but the baffle system is too complex to permit control over the meshing pattern of individual surfaces. Other CAD software may allow for more control over meshing. Several constraints must be considered when meshing the baffle:
• Quality of Approximation
• Number of triangular mesh elements
• Relative size and distribution of elements

These constraints are now considered separately.

3.3.2.1 Quality of Approximation

When meshing the baffle, it is desirable to approximate the surface as closely as possible. The better the surfaces are approximated, the more accurately the model represents the true system behavior. It is impossible to exactly match every curved surface with planar elements but suitable approximations can be attained. Figure 3.7 shows a meshing scheme that matches the curves of the baffle slots adequately for visual inspection.
Figure 3.7 Several views of a baffle triangular mesh example. Panels (a) and (b) show the same view without and with the mesh. Panel (c) is an isometric view of the baffle mesh. This mesh consists of 84,774 elements.
3.3.2.2 Number of Triangular Mesh Elements

If approximation was the only consideration, the mesh in Fig. 3.7 would be more than adequate. However, computational limits must also be considered. The more elements used to mesh the surface, the longer it will take to execute a ray-trace through the 3-D model. The mesh in Fig. 3.7 has 84,774 elements. For each ray traced, each of those elements needs to be visited for possible ray intersection, both for the original ray path and after each reflection. Even without reflection, a ray-trace with one million rays would require visiting each of those elements once for each ray, which would involve visiting 84 billion surfaces before it is finished. A trade-off between element number and surface approximation is a trade-off between speed and accuracy.

Figure 3.8 shows another meshing scheme. This mesh still approximates the surface well but does not have nearly as many elements. This mesh has only 2,304 elements. In other words, less than 3% of the number of elements in the previous meshing scheme.
Figure 3.8 Several views of a baffle triangular mesh example. Panels (a) and (b) show the same view without and with the mesh. Panel (c) is an isometric view of the baffle mesh. This mesh consists of 2,304 elements.
Tracing rays through the mesh in Fig. 3.8 would be much faster than tracing the rays through the mesh in Fig. 3.7. Even with the reduced number of elements, accuracy would not be compromised. However, since radiation distribution factors for the aperture may be required in anticipation of an eventual conduction analysis, the mesh in Fig. 3.8 is inappropriate.

**3.3.2.3 Relative Size and Distribution of Elements**

To increase the utility of the distribution factors, all of the elements should be approximately the same size. Also, they should be distributed fairly evenly over the surfaces. The triangular elements in the meshes in Figs. 3.7 and 3.8 both vary greatly in distribution and size. The purpose in calculating radiation distribution factors is to evaluate the sensitivity of different regions on the model to the aperture radiance. This purpose is better met using equal-sized, uniformly distributed elements.

Figure 3.9 shows the mesh from Fig. 3.8 with a single element highlighted in red. Evaluating the distribution factor for this element would not be very meaningful. The radiation distribution factor for this element would indicate the sensitivity of a thin region extending from the baffle slot all the way to the inner wall. However, the sensitivity of the baffle face must in reality vary along the length of this element. This aspect of the sensitivity would not be captured by this meshing scheme.
Figure 3.9 Meshing scheme from with a single element highlighted in red.

Additionally, this meshing scheme involves elements much larger than the highlighted element. A larger element has more surface area to absorb rays and would be more sensitive to radiation from the aperture than the highlighted element. Because of their varied sizes, comparing the distribution factors between the two elements would be uninformative, especially in the context of a conduction analysis.

3.3.2.4 Selected Mesh

A balance between surface approximation, number of elements, element size, and element distribution is needed for a practical and informative ray-trace. Figure 3.10 shows the meshing scheme selected for the ray-trace reported here. As seen in the figure, the baffle openings are blockier than the openings in the two previous meshes. Some geometrical accuracy was compromised to reduce the number of elements while assuring a well distributed mesh.
Figure 3.10 Selected baffle mesh. Panels (a) and (b) show the same view without and with the mesh. Panels (c) and (d) are different isometric views of the baffle mesh. Panel (e) is a cut-away view showing the meshing on the inside of the baffle. This mesh consists 10,590 elements.

The mesh in Fig. 3.10 has 10,590 elements. While containing more elements than the mesh in Fig. 3.8, it contains considerably fewer elements than the mesh in Fig. 3.7. Also, the elements in the mesh in Fig. 3.10 are comparative in size to each other and distributed such that a single element will not experience an excessive variation in sensitivity across its surface. While
not the ideal mesh, this mesh is judged sufficient for achieving the project objectives. In the future, it may be possible to create a superior mesh with fewer but still well distributed elements. Also, an accelerated ray-trace algorithm could possibly mitigate the need for simpler meshes.

Once the model has been meshed, it needs to be exported and saved as a .stl ASCII file. This is important for reading the mesh into the ray-trace environment.

### 3.3.3 Reading the Model

In order to use a plane surface in a ray-trace, its bounding vertices and the direction of its outward-facing normal are needed. This information is needed for every element in the mesh. Additionally, to distinguish the mesh elements, each element needs to be assigned a unique index number. A .stl ASCII file is the perfect format for exporting this information in a way that can be easily read into a Matlab code.

If the user exports the mesh as a .stl ASCII file, the mesh can be opened and viewed as a 3-D model in 3-D modelling software or as a list of parameters in a plain text editor. An STL mesh with \( N \) elements as seen in a plain text editor would read as in Table 3.1 [13].
Table 3.1: .stl ASCII file syntax.

\begin{verbatim}
solid mesh name
  facet normal nx1 ny1 nz1
  outer loop
    vertex px1 py1 pz1
    vertex px2 py2 pz2
    vertex px3 py3 pz3
  endloop
.
.
endfacet
  facet normal nxN nyN nzN
  outer loop
    vertex pxN pyN pzN
    vertex pxN pyN pzN
    vertex pxN pyN pzN
  endloop
endfacet
endsolid Mesh
\end{verbatim}

In an actual file, the italicized values would be replaced with the actual values for the specific mesh. The .stl ASCII file reveals the element number, the outward normal vector, and the three vertices of every triangular element in the mesh. This is all the information needed to trace rays through the model.

The format given in Table 3.1 lends itself very easily to being read into a Matlab matrix. In Matlab an \( N \times 12 \) element array is created to store the information from the .stl file, where \( N \) is the number of elements in the model. Table 3.2 illustrates the format of this matrix. The first three columns are the \( x-, y-, \) and \( z-\)components of the outward normal. The next three columns are the coordinates of the first vertex of each element. The next three columns after that are dedicated to the second vertex, and the last three columns correspond to the third vertex. The indices of the elements become the \( N \) indices of the row vectors in the array.
Table 3.2: Mesh matrix format.

<table>
<thead>
<tr>
<th>Element Number</th>
<th>Outward Normal Vector</th>
<th>First Vertex</th>
<th>Second Vertex</th>
<th>Third Vertex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( n_x1 ) ( n_y1 ) ( n_z1 )</td>
<td>( p_{xa1} ) ( p_{ya1} ) ( p_{za1} )</td>
<td>( p_{xb1} ) ( p_{yb1} ) ( p_{zb1} )</td>
<td>( p_{xc1} ) ( p_{yc1} ) ( p_{zc1} )</td>
</tr>
<tr>
<td>2</td>
<td>( n_x2 ) ( n_y2 ) ( n_z2 )</td>
<td>( p_{xa2} ) ( p_{ya2} ) ( p_{za2} )</td>
<td>( p_{xb2} ) ( p_{yb2} ) ( p_{zb2} )</td>
<td>( p_{xc2} ) ( p_{yc2} ) ( p_{zc2} )</td>
</tr>
<tr>
<td>3</td>
<td>( n_x3 ) ( n_y3 ) ( n_z3 )</td>
<td>( p_{xa3} ) ( p_{ya3} ) ( p_{za3} )</td>
<td>( p_{xb3} ) ( p_{yb3} ) ( p_{zb3} )</td>
<td>( p_{xc3} ) ( p_{yc3} ) ( p_{zc3} )</td>
</tr>
<tr>
<td>N</td>
<td>( n_{xN} ) ( n_{yN} ) ( n_{zN} )</td>
<td>( p_{xaN} ) ( p_{yaN} ) ( p_{zbN} )</td>
<td>( p_{xbN} ) ( p_{ybN} ) ( p_{zbN} )</td>
<td>( p_{xcN} ) ( p_{ycN} ) ( p_{zcN} )</td>
</tr>
</tbody>
</table>

Once the mesh is stored in a matrix, Matlab graphics functions can be used to view the mesh. Figures 3.11 and 3.12 are Matlab-generated images of the same baffle mesh shown in Fig. 3.10.

![Figure 3.11 Baffle mesh in Matlab.](image-url)
3.3.4 Ray-Tracing with a Triangular Mesh

One of the great advantages of using an STL mesh in a ray-trace is that all surfaces encountered during a ray-trace are planar. This greatly simplifies finding intersection points. The main problem for triangle meshes is identifying the triangle that an intersection point falls within. This is the most computationally expensive step in the process.

Multiple algorithms are available for determining whether or not a point lies within a given co-planar triangle. A simple inside-outside test was selected for the ray-trace [15]. When a given triangular element is visited, three vectors are created around the triangle based on the three sides. It should be emphasized that a ray-trace code organized as described here is completely general; that is, once Table 3.2 has been imported to the code, the resulting ray-trace
logic is independent of the actual geometry. Figure 3.13 shows a triangular element defined by the three vectors whose sum is zero, with a ray intersection represented by a red marker.

![Figure 3.13 Ray intersection point inside a triangular element defined by three vectors whose sum is zero.](image)

In Fig. 3.13, if the outward normal to the triangle is directed out of the page, the vectors point counter-clockwise around the perimeter of the triangle. The red point in Fig. 3.13 will be inside the triangle only if it lies on the left-hand side of all three vectors. Thus, if a ray intersection is on the left-hand side of all three vectors of a given element, it may be concluded that the intersection is within the element. The ray is then either absorbed or reflected by that element.

### 3.4 The Freeform Mirror

To incorporate the freeform mirror into the ray-trace, a surface equation for the mirror was needed. A common way of expressing optical surfaces mathematically is as a surface sagitta. The term sagitta, or sag, is rooted in the manufacturing of optical surfaces. It refers to the amount of material that must be removed to create an optical surface. Consider a solid block of material such as the one shown in Fig. 3.14. The coordinate system origin is centered on the top face of
the block. The z-axis points downward into the block. For a given x, y-coordinate point, the sag at that point would be the depth, or distance along the z-axis, from that point to the optical surface.

![Diagram of a block with sag profile](image)

**Figure 3.14** Block of material with sag profile shown.

The sag for an optical surface can be expressed as,

\[ z = g(x, y), \]  

(3.10)

where \( g \) is a function of \( x \), and \( y \) defined for all points on the optical surface. By subtracting \( z \) from both sides, this expression easily becomes a surface equation like Eq. (3.1); that is,

\[ f(x, y, z) = g(x, y) - z = 0. \]  

(3.11)

A sag expression for the mirror surface is now needed. Sag expressions for traditional optical surfaces such as spheres and paraboloids are simple and straightforward. The mirror in this system, however, is a freeform surface. Before the sag expression used for the freeform mirror is explained, some background to freeform surfaces is required.

### 3.4.1 Freeform Surfaces

What is a freeform surface? While more in-depth definitions for freeform surfaces exist, for this discussion it is sufficient to say that freeform surfaces are surfaces without rotational
symmetry [16]. In other words, surfaces beyond basic shapes like spheres, ellipsoids, hyperboloids, and other conic surfaces.

Such surfaces can be advantageous in optics. Freeform surfaces can be used to reduce aberrations inherent to traditional optical systems. While the concept of freeform surfaces is not new, their contribution in optics is relatively new. Because of their irregular shapes, manufacturing difficulties have made them traditionally impractical. However, improved manufacturing techniques and technology have now enabled the use of freeform optics. Breaking away from traditionally shaped optics has led to new and advantageous designs. Mirrors and lenses made from freeform surfaces have been used to reduce the size of optical systems and increase performance [16].

Due to their complex shapes, freeform surfaces cannot be simply expressed mathematically. Usually the sag for a freeform surface is expressed as a base conic surface followed by summations of polynomials. These polynomials “sculpt” the shape of the optical surface away from its base shape to a more optically advantageous shape.

Many different types of freeform surface sags have been defined, each used for different applications and better suited for different families of shapes [16, 17]. To be consistent with the method used by Quartus Engineering, Inc. when designing the DEMETER mirror, a form of freeform sag called Zernike sag was used. In the following sections, the Zernike sag expression used to create the model of the mirror is derived.

### 3.4.2 Zernike Sag

The standard Zernike sag is expressed in cylindrical coordinates by the equation

$$z = \frac{cr^2}{1+\sqrt{1-(1+k)c^2r^2}} + \sum_{i=1}^{8} \alpha_i r^{2i} + \sum_{j=1}^{N} A_j Z_j(\rho, \phi).$$  \hspace{1cm} (3.12)
In order for it to be used in the ray-trace, parameters specific to the DEMETER mirror need to be substituted into it. Also, Eq. (3.12) needs to be converted from cylindrical coordinates to Cartesian coordinates.

Equation (3.12) is the summation of three distinct parts: the conic term, the even aspheric terms, and the Zernike terms. Each of these three parts is considered separately in subsequent sections. The conic term, the even aspheric terms and the Zernike terms are referred to here, respectively, as \( z_1 \), \( z_2 \), and \( z_3 \). The meaning of each term and its particular variables and constants are explained. Each term is manipulated to the form used in the ray-trace code. Ultimately, the three parts are recombined to create a new expression, equivalent to Eq. (3.12), in Cartesian coordinates, and specific to the DEMETER mirror.

### 3.4.2.1 Conic Term

The conic term is the foundation for the shape of the surface. It is a function of radial position, \( r \); i.e.,

\[
Z_1 = \frac{cr^2}{1+\sqrt{1-(1+k)c^2r^2}}. \tag{3.13}
\]

By defining the two constants \( c \) and \( k \), the baseline conic shape upon which the mirror is based is defined. For this particular mirror, the baseline conic is a sphere. For a sphere, \( k = 0 \) and \( c = 1/R \), where \( R \) is the radius of the sphere. With these substitutions, Eq. (3.13) becomes,

\[
Z_1 = \frac{r^2}{R+\sqrt{R^2-r^2}}. \tag{3.14}
\]

Equation (3.14) can be converted to Cartesian coordinates by recalling that \( r = \sqrt{x^2 + y^2} \). Making this substitution results in,

\[
Z_1 = \frac{x^2+y^2}{R+\sqrt{R^2-x^2-y^2}}, \tag{3.15}
\]

or, after simplifying,
\[ z_1 = R - \sqrt{R^2 - x^2 - y^2}. \quad (3.16) \]

The half of the sphere being considered now needs to be determined. The current expression defines a hemisphere with its vertex at the system origin and concave upward (i.e., opening towards positive z). The mirror is based on a hemisphere with its vertex at the system origin but concave downward. To make this change, the right-hand side of the expression is multiplied by negative unity, obtaining

\[ z_1 = -R + \sqrt{R^2 - x^2 - y^2}. \quad (3.17) \]

Figure 3.15 is a cut-away view of half of a mirror created from just the \( z_1 \) term, bounded by the edges of the actual edges of the true mirror.

![Cut-away view of the spherical mirror trimmed to DEMETER size.](image)

Figure 3.15 Cut-away view of the spherical mirror trimmed to DEMETER size.

Now come the even aspheric terms.

### 3.4.2.2 Even Aspheric Terms

The even aspheric terms are of the form,

\[ z_2 = \sum_{i=1}^{\infty} \alpha_i r^{2i}, \quad (3.18) \]

or,
\[ z_2 = \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}, \]

(3.19)

where \( \alpha_i \) is a set of user-defined constants which adjust the shape of the mirror. For the actual mirror, none of these coefficients have an order of magnitude greater than \( 10^{-6} \), and most of them are much smaller than \( 10^{-6} \). The last several even aspheric terms have the largest exponents of \( r \) in the entire sag expression. Because of this, increases in \( r \) radically increase the contribution of the higher-order even aspheric terms. The DEMETER mirror required minimal adjustments from the even aspheric terms.

The conversion to Cartesian coordinates is straightforward; that is,

\[ z_2 = \alpha_1(x^2 + y^2) + \alpha_2(x^2 + y^2)^2 + \cdots + \alpha_7(x^2 + y^2)^7 + \alpha_8(x^2 + y^2)^8, \]

(3.20)

or,

\[ z_2 = \sum_{i=1}^{8} \alpha_i(x^2 + y^2)^i. \]

(3.21)

Like the conic term, the even aspheric terms are each a function of \( r \) only. They do not vary with rotations about the \( z \)-axis. At this point, if \( z_1 \) and \( z_2 \) were summed to create a mirror, it would be shaped as in Fig. 3.16. This surface is an aspheric surface but not a freeform surface. It is still axisymmetric and thus, not freeform. The freeform element of the mirror comes from the Zernike terms.
3.4.2.3 Zernike Terms

Zernike polynomials are named after Fritz Zernike, who derived them in 1934 [18]. They are useful in that the terms correspond well to common optical aberrations [18]. They are also continuously orthogonal over the unit disk on which they are defined [19]. They are used in analyzing optical aberrations in atmospheric studies [20] and the human eye [21]. They are also used in wave-front sensors [22]. They can also be used (as in this project) to represent freeform optical surfaces.

Zernike polynomials are functions of both radial position and azimuthal angle. This makes them well suited for representing unique asymmetric shapes. As seen in Eq. (3.1), the basic form of the sum of the Zernike polynomials is,

\[ z_3 = \sum_{j=1}^{N} A_j Z_j(\rho, \phi), \tag{3.22} \]

where \( A_j \) is a set of \( N \) user-defined coefficients. The set of functions \( Z_j(\rho, \phi) \) represents the actual Zernike polynomials. These functions are described in greater detail subsequently. The variable \( \rho \) is the normalized radial position (remember, the polynomials are defined over a unit disk). It is equal to radial position \( r \), divided by a user-defined normalization radius, \( R_{\text{norm}} \). The
user selects a normalization radius sufficiently large to exceed the boundaries of the optical surface they are designing. This is to ensure that the Zernike terms behave as expected over the entire optical surface.

Several indexing schemes and variations are available for describing Zernike polynomials. Some are very similar, which can lead to confusion when trying to define a surface. The user should take great care to verify the variation they are using.

For the DEMETER mirror, the indexing scheme used is a variation defined by Noll in 1975 [20]. This version of the Zernike polynomials was used because it is the standard Zernike polynomial set in the commercial optical design software Zemax [23], the software Quartus Engineering, Inc. used to design the DEMETER mirror.

Noll’s Zernike polynomials are governed by three indices: Noll’s ordering index, $j$, and two additional indices $m$ and $n$ which harken back to Zernike’s original indexing pattern. The index $m$ indicates the polynomial azimuthal frequency, while the index $n$ indicates the radial degree (highest power of $\rho$ in the polynomial). For a given polynomial term, $n - m = \text{even}$ and $m \leq n$. The indices $m$ and $n$ are both zero-based indices while $j$ is unity-based. When $m$ and $n$ are zero, $j$ is unity and increases with $m$ and $n$. For a given $n$, lower values of $m$ correspond to lower values of $j$. Except when $m = 0$, two $j$ values per $m$ and $n$ pair are used. To help illustrate the relationship between the three indices, Table 3.3 shows the Zernike polynomials up to $Z_{43}$ in the context of $m$ and $n$ [20].
Table 3.3: First 43 Zernike terms organized by \( n \) (rows) and \( m \) (columns).

<table>
<thead>
<tr>
<th>( n )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( Z_1 )</td>
<td>( Z_2, Z_3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( Z_4 )</td>
<td>( Z_5, Z_6 )</td>
<td>( Z_7, Z_8 )</td>
<td>( Z_9, Z_{10} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>( Z_{11} )</td>
<td>( Z_{12, 13} )</td>
<td>( Z_{14, 15} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( Z_{16, 17} )</td>
<td>( Z_{18, 19} )</td>
<td>( Z_{20, Z_{21}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>( Z_{22} )</td>
<td>( Z_{23, 24} )</td>
<td>( Z_{25, Z_{26}} )</td>
<td>( Z_{27, Z_{28}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>( Z_{29, Z_{30}} )</td>
<td>( Z_{31, Z_{32}} )</td>
<td>( Z_{33, Z_{34}} )</td>
<td>( Z_{35, Z_{36}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( Z_{37} )</td>
<td>( Z_{38, 39} )</td>
<td>( Z_{40, Z_{41}} )</td>
<td>( Z_{42, Z_{43}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The expressions for \( Z_j \) can now be expanded.

When \( j \) is even and \( m \neq 0 \),

\[
Z_j = \sqrt{n + 1} R_n^m(\rho) \sqrt{2} \cos(m\phi).
\]  
(3.23)

When \( j \) is odd and \( m \neq 0 \),

\[
Z_j = \sqrt{n + 1} R_n^m(\rho) \sqrt{2} \sin(m\phi).
\]  
(3.24)

When \( m = 0 \),

\[
Z_j = \sqrt{n + 1} R_n^0(\rho).
\]  
(3.25)

In Eqs. (3.23) - (3.25), the function \( R_n^m(\rho) \) is defined,

\[
R_n^m(\rho) = \sum_{s=0}^{\lfloor(n-m)/2\rfloor} \frac{(-1)^s (n-s)!}{s! \lfloor(n+m)/2\rfloor! \lfloor(n-m)/2\rfloor!} \rho^{n-2s}.
\]  
(3.26)

If these definitions are taken in the context of Table 3.3, all the Zernike term pairs that share \( m,n \)-coordinates are the same polynomial function of \( \rho \) multiplied by a function of \( \phi \). All odd numbered terms are multiplied by \( \sqrt{2} \sin(m\phi) \) and all even terms are multiplied by \( \sqrt{2} \cos(m\phi) \). This is the point of Noll’s indexing system. The even numbered terms have an even azimuthal component and the odd numbered terms have an odd azimuthal component. The
reason the first column does not seem to follow this rule is that the azimuthal frequency is zero. In other words, no azimuthal variation exists in the first column; thus, no function of \( \phi \) and no pair. Each Zernike term can now be written out as functions of \( \rho \) and \( \phi \).

Table 3.4 lists the first ten terms. For the ray-trace code, the first 37 terms were used. For a list of the first 37 Zernike terms in cylindrical coordinates, refer to Appendix A. Only the first 37 terms were used because that is the number of terms used by Quartus Engineering. Additional terms are usually not needed.

Table 3.4: First ten Zernike terms in cylindrical coordinates.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( Z_i(\rho, \phi) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>( 2\rho \cos \phi )</td>
</tr>
<tr>
<td>3</td>
<td>( 2\rho \sin \phi )</td>
</tr>
<tr>
<td>4</td>
<td>( \sqrt{3}(2\rho^2 - 1) )</td>
</tr>
<tr>
<td>5</td>
<td>( \sqrt{6}\rho^2 \sin 2\phi )</td>
</tr>
<tr>
<td>6</td>
<td>( \sqrt{6}\rho^2 \cos 2\phi )</td>
</tr>
<tr>
<td>7</td>
<td>( 2\sqrt{2}(3\rho^3 - 2\rho) \sin \phi )</td>
</tr>
<tr>
<td>8</td>
<td>( 2\sqrt{2}(3\rho^3 - 2\rho) \cos \phi )</td>
</tr>
<tr>
<td>9</td>
<td>( 2\sqrt{2}\rho^3 \sin 3\phi )</td>
</tr>
<tr>
<td>10</td>
<td>( 2\sqrt{2}\rho^3 \cos 3\phi )</td>
</tr>
</tbody>
</table>

Converting the 37 Zernike polynomials from cylindrical coordinates to Cartesian coordinates is significantly more challenging than converting the conic and even aspheric terms. For this step, a program in Matlab was written to automatically convert the polynomials.
The program reads in the 37 cylindrical polynomials as symbolic functions of $\rho$ and $\phi$. It then substitutes $\sqrt{x^2 + y^2/R_{norm}}$ for $\rho$ and $\tan^{-1}\left(\frac{y}{x}\right)$ for $\phi$. The program then simplifies the expressions.

It should be noted that, after converting the 37 polynomials to Cartesian coordinates, one more change is required to ensure that they are correct. All of the Zernike terms involving a sine function had to be multiplied by negative one. This is due to an ambiguity in the application of the sine function when solving symbolically in Matlab. This problem can come up when transferring Zernike sag data from one entity to another; for example, from an optical component designer to a manufacturer. Care should be taken to verify a freeform surface shape before application.

Table 3.5 shows the first ten Zernike terms in Cartesian coordinates. The full list of the 37 terms used is reproduced in Appendix B.
Table 3.5: First ten Zernike terms in Cartesian coordinates.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$Z_i(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>$\frac{2x}{R_{\text{norm}}}$</td>
</tr>
<tr>
<td>3</td>
<td>$-\frac{2y}{R_{\text{norm}}}$</td>
</tr>
<tr>
<td>4</td>
<td>$\frac{2\sqrt{3}(x^2 + y^2)}{R_{\text{norm}}^2} - \sqrt{3}$</td>
</tr>
<tr>
<td>5</td>
<td>$-\frac{2\sqrt{6}xy}{R_{\text{norm}}^2}$</td>
</tr>
<tr>
<td>6</td>
<td>$\frac{\sqrt{6}(x^2 - y^2)}{R_{\text{norm}}^2}$</td>
</tr>
<tr>
<td>7</td>
<td>$-\frac{2\sqrt{2}y(-2R_{\text{norm}}^2 + 3x^2 + 3y^2)}{R_{\text{norm}}^3}$</td>
</tr>
<tr>
<td>8</td>
<td>$\frac{2\sqrt{2}x(-2R_{\text{norm}}^2 + 3x^2 + 3y^2)}{R_{\text{norm}}^3}$</td>
</tr>
<tr>
<td>9</td>
<td>$-\frac{2\sqrt{2}y(3x^2 - y^2)}{R_{\text{norm}}^3}$</td>
</tr>
<tr>
<td>10</td>
<td>$\frac{2\sqrt{2}x(x^2 - 3y^2)}{R_{\text{norm}}^3}$</td>
</tr>
</tbody>
</table>

With these changes, all of Eq. (3.12) can now be written in Cartesian coordinates:

$$z = -R + \sqrt{R^2 - x^2 - y^2} + \sum_{i=1}^{8} \alpha_i(x^2 + y^2)^i + \sum_{j=1}^{N} A_j Z_j(x, y).$$  \hspace{1cm} (3.27)

Before showing the completed mirror, a final alteration to the sag equation is noted. In the mirror, the effect of the Zernike terms has been decentered. This means that the effects of the Zernike terms do not center around the vertex of the base spherical mirror. The Zernike terms are
offset from the origin by a distance along the \( y \)-axis, \( \Delta y \). This offset is incorporated into the expression thusly,

\[
z = -R + \sqrt{R^2 - x^2 - y^2} + \sum_{i=1}^{8} \alpha_i (x^2 + y^2)^i + \sum_{j=1}^{N} A_j Z_j (x, y + \Delta y).
\]  

(3.28)

Figure 3.17 shows the final mirror. While the actual mirror is bounded by a curved profile as in Fig. 3.17 to simplify the ray-trace, the mirror is instead bounded by a larger rectangular boundary as in Fig. 2.3.

![Figure 3.17 Cut-away view of the freeform mirror used in the DEMETER instrument.](image)

The mirror in Fig. 3.17 was compared with sag data provided by Quartus Engineering to confirm that the shape of the mirror was correct. Towards the edges of the mirror, there was a slight discrepancy between the mirror in Fig. 3.17 and the data provided by Quartus Engineering. This was most likely caused by a difference in precision between Matlab and Zemax. To confirm that the mirror behaved sufficiently like the mirror designed by Quartus Engineering, some preliminary ray-traces were conducted using only the mirror and the focal plane. The results of these ray-traces are discussed in Section 3.6 after the description of the focal-plane.
3.5 Focal-plane

The surface function for the focal-plane in the ray-trace model is an infinite plane. In reality, of course, the focal-plane will be bounded. However, for the ray-trace model, it was desirable to have all rays that reach the mirror also reach the focal-plane. Thus, the plane is boundless. All regular rays will contact the focal-plane within the bounds of the actual focal-plane detector. The stray rays will be distributed across the plane.

The focal-plane is defined by tracing a single ray, called the chief ray, from the center of the aperture, along the positive $z$-axis, to the mirror. This ray is reflected from the mirror and is traced a fixed distance. This distance is the effective focal length from the mirror to the focal-plane. The coordinates of the chief ray intersection point are considered the origin of the focal-plane local coordinate system. The focal-plane is defined using this point and the focal-plane unit normal vector. Often in such systems the focal-plane normal is parallel to the chief ray. In the current case, however, the focal-plane normal is defined independent of the chief ray. The focal-plane unit normal vector is perpendicular to the $x$-axis and is rotated about the $x$-axis a fixed angle $\theta_{fp}$ away from the $z$-axis.

Figure 3.18 shows the ray-trace 3-D model. A portion of the focal-plane is highlighted in yellow. This rectangular section encompasses all regular rays that reach the focal-plane. The path of the chief ray is shown in red.
In the model, any ray that reaches the focal-plane is absorbed. When the ray-trace is completed, the Matlab code will generate a PSF plot of the rays incident on the focal-plane. Regular rays are represented in one color while stray rays are represented in another. This helps the user evaluate the effectiveness of different surface finishes for the baffle. Stray rays contaminate the detector measurements; therefore, the goal is to have as few stray rays reach the focal-plane as possible.

With the focal plane defined, the preliminary tests used to confirm the mirror prescription can be discussed.

**3.6 Preliminary Focal-Plane PSF Comparison**

As mentioned in Section 3.4, preliminary ray-traces were conducted with only the freeform mirror and focal plane. Rays were traced in collimated beams from a circular pupil to the mirror and then to the focal plane. The point-spread functions for these ray-traces were
compared with results obtained by Quartus Engineering [24]. Figures 3.19 - 3.21 show the PSF plot comparisons for beam angles of 0, 8, and 16 deg, respectively.

Figure 3.19 Comparison of PSF plots for a beam angle of 0 deg. Panel (a) is the PSF obtained by Quartus Engineering and has units of micrometers [24]. Panel (b) is the PSF from the ray-trace code and has units of millimeters.
Figure 3.20 Comparison of PSF plots for a beam angle of 8 deg. Panel (a) is the PSF obtained by Quartus Engineering and has units of micrometers [24]. Panel (b) is the PSF from the ray-trace code and has units of millimeters.
Figure 3.21 Comparison of PSF plots for a beam angle of 16 deg. Panel (a) is the PSF obtained by Quartus Engineering and has units of micrometers [24]. Panel (b) is the PSF from the ray-trace code and has units of millimeters.

The PSF plots in Fig. 3.19 are very similar to each other in that they share somewhat the same basic shape. The PSF plots in Fig. 3.20 still share somewhat the same basic shape. The shapes of the two plots in Fig. 3.21, however, are less similar to each other. This will not be a
significant problem as long as the PSF plots for each beam angle are located in the same place and the root-mean-square (RMS) spot radii for each beam angle are about equal.

3.6.1 PSF Position Comparison

The positions of the PSF plots can be compared using the coordinates of the point where a ray traced from the center of the aperture is absorbed on the focal-plane relative to where the chief ray is absorbed. In the PSF plots provided by Quartus Engineering, the \( x, y \)-coordinates of this point appear beneath the image. The PSF plots in Fig. 3.19 are both centered at the coordinate system origin. The PSF in Fig. 3.20 (a) is centered at \( x = 5.447 \text{ mm} \) and \( y = -0.242 \text{ mm} \). The PSF in panel (b) is center at \( x = 5.319 \text{ mm} \) and \( y = -0.271 \text{ mm} \). The PSF in Fig. 3.21 (a) is centered at \( x = 10.750 \text{ mm} \) and \( y = -1.034 \text{ mm} \). The PSF in panel (b) is center at \( x = 10.413 \text{ mm} \) and \( y = -1.128 \text{ mm} \). It may be concluded that the positions for both the current and the Quartus PSF plots are sufficiently near each other.

3.6.2 Root-Mean-Square (RMS) Spot Radii Comparison

The root-mean-square (RMS) spot radius is a measure of the size of the collection of points (referred to as the spot) on the focal plane [25]. It can be calculated using the equation [25]

\[
R_{\text{spot}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} R_i^2},
\]

where \( N \) is the number of points in the PSF and

\[
R_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}.
\]

The variables \( x_i \) and \( y_i \) represent the coordinates of individual points in the PSF, and
\[ x_0 = N^{-1} \sum_{i=1}^{N} x_i, \quad (3.31) \]

and

\[ y_0 = N^{-1} \sum_{i=1}^{N} y_i. \quad (3.32) \]

The spot radius for the image in Fig. 3.19 (a) is 0.378 mm and the spot radius for Fig. 3.19 (b) is 0.377 mm. These values are very close. This suggests that the mirror is well approximated near the center. The spot radius for the image in Fig. 3.20 (a) is 0.330 mm and the spot radius for the image in Fig. 3.20 (b) is 0.387 mm. The spot radius for the image in Fig. 3.21 (a) is 0.300 mm and the spot radius for the image in Fig. 3.21 (b) is 0.429 mm. This indicates a slight variation between the Quartus prescription and the one used in this investigation near the edge of the mirror.

### 3.6.3 Comparison Conclusions

From these results, it may be concluded that the freeform surface equation used in the ray-trace well approximates the mirror designed by Quartus Engineering. The freeform surface equation better represents the mirror towards the center, but still represents the mirror sufficiently towards the edge. While the spot radius for a beam angle of 16 deg is slightly larger in the ray-trace model, the PSF is still in relatively the same position as the PSF obtained by Quartus Engineering.
Chapter 4: Ray-Trace Code Input and Output

4.1 Ray-Trace Code Logic Flow

Figure 4.1 is a logic flow chart of the ray-trace process showing the input and output of the code. The full ray-trace code is included in Appendix C. It will be necessary to give a background of the ray-trace code input and output to better understand the results.

![Ray-trace logic flow chart](image)

Figure 4.1 Ray-trace logic flow chart.

4.2 Ray-Trace Input

The user defines the absorptivity and specularity for the surfaces of the baffle. Both of these are values between 0 and 1. The user also defines the number of rays to be traced. The user
selects the entrance angle of the collimated beam of rays. The beam angle should be an angle measured in degrees between 0 deg and 16 deg. The user also has some control over the ray-trace graphical output. These settings are explained when the output they control is discussed.

The user can also select which mesh is imported into the ray-trace code to represent the baffle. In order to be imported, the .stl file for the mesh must be contained in the same computer directory as the ray-trace code.

4.3 Ray-Trace Output

Once the ray-trace is complete, the Matlab code has several outputs, both visual and numeric.

4.3.1 3-D Ray-Trace Model

The first output the code generates is a 3-D model of the system shown in Fig. 4.2. This always includes a transparent model of the baffle, a model of the mirror, and a model of the focal-plane with points representing the rays absorbed by the focal-plane. The user can also choose to include additional visual components in the 3-D model.
Figure 4.2 Ray-trace 3-D model with 1,000 rays. No additional graphical components are activated.

For Fig. 4.2 and the other example plots in this chapter, the absorptivity and specularity ratio settings are arbitrary. These figures merely serve to illustrate the format of the numerical model graphical outputs. Absorptivity and specularity ratio settings are addressed in greater detail in the following chapter in the context of actual test results.

4.3.2 Ray Paths

If the user chooses, the 3-D model will also include the optical paths of all the rays. Figure 4.3 (a) - (c) shows ray path models for a beam angle of 0 deg with 10, 100, and 1,000 rays traced, respectively. Ray paths inside the baffle and to the mirror are represented by red lines. Points of reflection or absorption are represented by blue points. Rays reflected from the mirror
to the focal-plane are represented by blue lines. Regular rays absorbed by the focal-plane are represented on the focal-plane as blue points. Stray rays absorbed by the focal-plane are represented on the focal-plane as red points.

Note that if the user chooses to view the optical paths, the 3-D model of the system will be oriented in the mirror local coordinate system and not the global coordinate system, as in Fig. 4.2. The program does not have the capability of translating the optical paths from the mirror local coordinate system to the global coordinate system.
Figure 4.3 Ray-traces for a beam angle of 0 deg with (a) 10 rays, (b) 100 rays, and (c) 1,000 rays.

Ray paths are visible. The figure is in the mirror local coordinate system.

The user should be warned that adding ray path visualization can significantly increase the run time of the ray-trace. Depending on the number of rays traced, showing the optical paths also increases the size of the 3-D figure file and will make it balky to manipulate. Three-
Dimensional plots in the Matlab programming environment can be revolved, magnified, and sliced to view different features of the plot. After 10,000 rays, the 3-D ray path plot freezes frequently and is not easy to rotate and manipulate. Without the optical paths, the user will still see the points representing the regular rays and stray rays absorbed on the focal-plane.

4.3.3 Power Flux Highlights

The user can also choose to highlight the elements of the baffle where rays are absorbed, as illustrated in Fig. 4.4. The elements where rays are absorbed are opaque. They are also color coded according to power flux. The power flux for each element is calculated by dividing the number of rays absorbed in an element by the area of the element. The units used are rays per square millimeter. Elements with fewer absorbed rays per square millimeter are darker. Elements with more rays per square millimeter are brighter. Elements where no rays are absorbed are not highlighted and remain transparent.

Figure 4.4 Cut-away view of a ray-trace for a beam angle of 0 deg with 1,000,000 rays. Power flux highlights are visible.
The power flux highlights setting is independent of the optical paths setting. The user can choose to show the flux highlights with or without the optical paths, as illustrated in Fig. 4.5. However, showing both is somewhat redundant and makes the image too crowded and more difficult to interpret. Showing power flux highlights adds little to the run-time of the ray-trace code. Also, it does not add to the complexity of the 3-D model. Ray-trace models with 1,000 rays or 1,000,000 rays with power flux highlights are equally easy to manipulate.

Figure 4.5 Ray-trace for a beam angle of 0 deg with 1,000 rays. Both power flux highlights and ray paths are visible.

The power flux highlights can be misleading in ray-traces with relatively small numbers of rays. Since the power flux of an element is a ratio of absorbed rays to element area, a large number of rays is needed to get an accurate comparison between different elements. The fewer
the rays, the more significant the area of the element becomes in the calculation. If an element is very small but it absorbs one ray in a ray-trace with only 100 rays, the element will probably appear to have the highest power flux. With more rays, this problem decreases. Again, this is why a well-distributed mesh is so essential.

The 3-D model is saved as both a Matlab figure file (.fig file) and a .jpg image file. The Matlab figure can be rotated and viewed from all angles and cutaway to inspect the interior of the model. This can be useful when visualizing ray paths. The image file shows a single view of the ray-trace captured from an angle that shows all components of the model.

**4.3.4 Focal-Plane Point-Spread Function**

The second visual output the code generates is a plot of the focal-plane point-spread function (PSF). This is basically a scatter plot of the rays absorbed on the focal-plane oriented in the focal-plane local coordinate system. Therefore, the figure is a 2-D $x$-$y$ plot. The data in this figure are also represented in the global coordinate system in the 3-D figure, but the figure is easier to interpret by isolating it. Figure 4.6 shows an example PSF plot.

![Figure 4.6 Focal-plane point-spread function (PSF) plot for a beam angle of 0 deg with 100,000 rays.](image)

Figure 4.6 Focal-plane point-spread function (PSF) plot for a beam angle of 0 deg with 100,000 rays.
As in the 3-D figure, blue points represent the regular rays, and red points represent the stray rays. This output is also saved as a Matlab .fig file and an image file.

4.3.5 Numeric Output

The ray-trace code also has several numeric outputs. These data are saved as sheets in a single Excel file. The most important dataset saved to the Excel file is the distribution factor files, but the Excel file also includes the coordinates of the regular and stray rays, the number of reflections of each ray, and the run-time of the ray-trace.

4.3.5.1 Distribution Factors

The first sheet of the Excel file contains the distribution factors. Remember that these are the distribution factors from the aperture to all surfaces in the model. These are shown as a list in the first column on the sheet. The baffle element indices are listed first. After these several thousand values, an additional 12 values are given representing the distribution factors for the walls of the imaginary enclosure surrounding the model. These will always be very small or zero. In a ray-trace, if any of these 12 distribution factors are non-zero, they should be taken into consideration when evaluating the number of stray rays on the focal-plane. The last two values represent the distribution factors for the focal-plane and the aperture, in that order.

It is useful to assure that everything is running correctly by adding up all the values in this list. Since the distribution factors are calculating by dividing the number of rays absorbed by each element by the total number of rays traced, the sum of all the distribution factors must be exactly unity.

4.3.5.2 Regular Ray and Stray Ray Coordinates

The next two sheets contain the coordinates of the absorbed regular and stray rays, respectively. The absence of an Excel spreadsheet for the stray rays indicates the absence of stray
rays reaching the focal-plane. In both sheets, the first column represents the \(x\)-coordinates of the absorbed rays, \(y\)-coordinates appear in the second column, and the \(z\)-coordinates appear in the third. Since these points are in the focal-plane local coordinate system, all the \(z\) values should be calculated zeros.

4.3.5.3 Ray Reflections

The next sheet in the Excel file is a list of the number of times each ray was reflected. This is included for diagnostic purposes. The number of times rays reflect; once, twice, three times, etc., should make sense in the context of the number of rays traced and the absorptivity of the baffle. If the user traces 100 rays with a baffle absorptivity of 0.9, and 20 regular rays reach the mirror, they should expect approximately 10% of the remaining 80 rays (eight rays) to be reflected at least once. Of those eight rays, approximately one could be reflected twice but it is highly unlikely a ray would be reflected three times. By inspecting the list of ray reflections, the user can assure that the ray-trace is behaving as expected.

4.3.5.4 Ray-Trace Run-Time

The final sheet contains a single value: the ray-trace code run-time in seconds. This is useful for determining the amount of time needed for a particular run. The run-time is roughly proportional to the number of rays traced. The user can run a ray-trace for a particular test with fewer rays first to estimate how long a more meaningful test will take. For example, if the user finds a certain ray-trace takes two minutes to trace one thousand rays, they can estimate it will take over a day to complete the same ray-trace with one million rays.

Now that the input and output of the ray-trace code has been explained, several ray-trace results can be discussed.
Chapter 5: Results

5.1 Test Results

Several example tests were conducted using the ray-trace model described in Chapter 3. Each test was selected to highlight different features of the code and test different aspects of the DEMETER instrument behavior.

While it is anticipated that more tests will be conducted in the future using this model, these tests act as a demonstration of the capabilities and potential of this model. They showcase the different visualization features and their interpretation to make conclusions about the actual instrument. Possible future tests using this model are considered in Chapter 6.

5.1.2 Surface Coating Comparison

The example tests in this chapter explore the difference between two surface coatings: a highly absorptive, highly specular surface coating and a highly absorptive, highly diffuse surface coating. The highly specular surface coating has an absorptivity, $\alpha$, of 0.9 and a specularity ratio, $R$, of 0.9. Specularity ratio is defined as

$$R \equiv \frac{\rho^s}{\rho^s + \rho^d}$$

(5.1)
where \( \rho^s \) is the specular component of the reflectivity and \( \rho^d \) is the diffuse component [9]. This means approximately 90% of the radiation incident to the baffle surfaces is absorbed and of the 10% not absorbed, approximately 90% is reflected specularly and 10% is reflected diffusely.

For the highly diffuse surface coating \( \alpha = 0.9 \) still, but \( R = 0 \). For this surface coating, still only 10% of the radiation is reflected, but all of that 10% is reflected diffusely. These properties were selected because they are comparable to the properties of the popular aerospace coatings Aeroglaze® Z302 and Z306 paints [26] [27]. Aeroglaze® Z302 and Z306 paints are both commonly used for coating telescope baffles [28]. They are both highly absorptive black surface coatings, but Z302 is highly specular and Z306 is highly diffuse. In the subsequent discussion, the two surface coatings used in the ray-trace are not referred to as Z302 and Z306 since they are only representative of Z302 and Z306. In reality, as with all surfaces, the specularity ratios and absorptivity of Z302 and Z306 are functions of radiation wavelength, polarization, and angle of incidence [9, 28]. In the ray-trace tests, the two surface coatings are referred to as the specular surface coating and the diffuse surface coating.

5.1.3 Test Sets

The first test traces 10,000 rays for each trial and is used to visualize ray paths at several beam angles for the specular and diffuse surface coatings. The next test uses the same beam angle values as the previous test but traces 100,000 rays for each trial. The purpose of this test was to visualize power flux and evaluate the point-spread functions of each trial. The next test traces 1,000,000 rays for each trial but only had two trials for the on-axis beams for the two surface coatings. The following sections present the results of these tests, including the graphical output generated in each trial.
5.2 Test 1: Ray Path Visualization

The first test set was conducted using 10,000 rays. This is relatively few rays for a ray trace but sufficient for this test. In this test, rays were traced at beam angles of 0, 8, and 16 deg. Each trial was repeated using the specular surface coating properties and the diffuse surface coating properties.

The purpose of this test was ray path visualization. In order to do this, the ray paths were highlighted as in Fig. 4.3. Ray-traces of the same beam angle but with differing surface coatings are compared side by side.

5.2.1 Test 1 Results

Figure 5.1 (a) and (b) show the two ray-traces for a beam angle of 0 deg.
Figure 5.1 Ray paths with 10,000 rays and a beam angle of 0 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 
In Fig. 5.1 (and in the other ray-trace figures for this test), red lines indicate the paths of the rays inside the baffle. The points where rays are absorbed or reflected are marked with blue points. The ray segments between the mirror and focal-plane are shown as blue line segments. The points where regular rays are absorbed on the focal-plane are indicated with blue points, and the points where stray rays are absorbed on the focal-plane are indicated with red points.

Comparing panels (a) and (b) in Fig. 5.1 reveals the differences between the specular surface coating and the diffuse surface coating. In the highly specular case (a), it appears that most of the reflected rays exit back through the aperture. A circular gap in the midst of the rays exiting the baffle indicates the path centered on the circular pupil in the middle baffle vane. In panel (a) several stray rays reach the mirror and focal-plane. In the diffuse trial, fewer rays exit back through the aperture but instead are absorbed by the baffle walls. In panel (b) no stray rays reach the mirror and focal-plane. As should be expected, the paths of the regular rays in both panels (a) and (b) appear identical.

Figure 5.2 (a) and (b) show the results of the trials with a beam angle of 8 deg.
Figure 5.2 Ray paths with 10,000 rays and a beam angle of 8 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 

68
These images tell a similar story to that of Fig. 5.1. More rays exit the baffle through the aperture in the more specular baffle in panel (a), and fewer stray rays reach mirror and focal-plane in the diffuse baffle in panel (b). Because of the oblique beam angle, the distribution of absorbed rays is different from the first trial. More rays exit the left side of the entrance aperture in panel (a) and more rays are scattered and absorbed in the left side of the baffle in panel (b).

Figure 5.3 reinforces the results shown in Figs. 5.1 and 5.2.
Figure 5.3 Ray paths with 10,000 rays and a beam angle of 16 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 
With the greater entrance angle, the majority of the rays exiting the baffle in panel (a) are confined to a narrow strip along the left edge. Consistently, fewer stray rays reach the focal-plane in the diffuse model.

5.2.2 Test 1 Conclusions

From these trials it can be observed that the ray-trace code can effectively show the paths of rays in the DEMETER instrument. This visualization can give insight into the effect of changes in surface finish. It appears from these trials that fewer stray rays exit the baffle at the mirror end in the diffuse model. This result indicates a more diffuse surface coating like Z306 would be better for preventing scattering instrument noise in measurements due to stray rays. However, with a diffuse surface coating, more rays will be absorbed inside the system instead of reflected back out the aperture. This could pose a concern if this increase in absorbed radiation raises the temperature of the baffle surfaces significantly.

It should also be remembered that none of these tests claim to give an estimate of the amount of stray radiation that actually reaches the detector. Recall that the actual instrument may have several more baffle vanes and perhaps even a precision aperture array between the mirror and the focal-plane to further eliminate stray rays. Also, the detector in the focal-plane is not infinite. In the real DEMETER instrument, fewer stray rays will be detected than are shown here. However, these test trials do show that the extra baffle vanes and apertures will not need to work as hard to block stray rays if the baffle is coated with a highly absorptive and diffuse surface coating such as Z306.

5.3 Test 2: Power Flux Visualization

In this test set, the same surface coating settings are used as before but now 100,000 rays are traced in each trial. Instead of highlighting the ray paths, in this test set the absorbed power
flux for each element is highlighted. Also, the point-spread function for each trial is collected. As in the first test set, trials of the same beam angle but differing specularity ratios were compared to each other.

5.3.1 Test 2 Results

Figure 5.4 (a) and (b) show the 3-D images of the ray-traces with a beam angle of 0 deg for both surface finishes. In Fig. 5.4, the opaque surfaces make the interior of the baffle difficult to observe. This effect is remediated in Fig. 5.5 (a) and (b), which are cut-away images of Fig. 5.4.
Figure 5.4 Power flux highlights with 100,000 rays and a beam angle of 0 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 
Figure 5.5 Cut-away views of Fig. 5.4. The beam angle is 0 deg.
In these images, the opaque triangular elements in the baffle indicate absorbed rays. Lighter colored opaque elements indicate elements that absorbed more rays per unit area. The number of rays absorbed per unit area in a given element is directly proportional to the power absorbed per unit area, or power flux.

In both panels (a) and (b) of Figs. 5.4 and 5.5, lighter colored elements are clustered around the edges of the slots in each baffle vane. This is intuitively pleasing. With a highly absorptive surface coating, most of the rays will be absorbed where they first make contact. For most rays, this will be the front face of a baffle vane not directly blocked by the baffle vane before it. Figure 5.6, panels (a) and (b) show the same 3-D plots as in Fig. 5.4 but rotated to show the entrance aperture. From this direction most of the elements are lighter colored. This is because the view is now aligned with the beam of rays entering the aperture.
Figure 5.6 Front elevation view of panels (a) and (b) of Fig. 5.4. The beam angle is 0 deg. A color bar has been added to indicate power flux. The units for the color bar are rays per square millimeter.

The two plots in Fig. 5.6 have color bars indicating the power flux for the elements in units of rays absorbed per square millimeter. An observer may be confused to see the color bar
ranging from dark blue to yellow but only see blue and green elements in the plot. While superior to the other meshes mentioned in Chapter 3, this mesh still has some disproportionately small elements. The smallest elements that absorb rays have the highest power flux because of their small surface areas. These elements are so small that they cannot be discerned at this scale; therefore, no yellow triangles are visible in Fig. 5.6. This problem decreases when more rays are traced, as will be seen in the next test set.

Returning to Fig. 5.4, as in the previous test set it can be observed that more stray rays reach the focal-plane in the ray-trace with a highly specular surface coating than in the ray-trace with a diffuse surface coating. Also, as before, it can be seen that more rays are absorbed in the baffle in the diffuse case. In Fig. 5.4, panel (a) the elements of the baffle are sparsely filled in while in panel (b) the front half of the baffle is almost completely opaque. This indicates more radiation is absorbed in the baffles with a specularity ratio of \( R = 0 \). It also indicates more radiation reaching the baffle walls surrounding the baffle vanes in the diffuse case.

Both panels (a) and (b) of Fig. 5.5 show that most of the rays absorbed in the baffle are absorbed before the circular pupil. This is clearer in Fig. 5.7, which is another cut-away side view of the 3-D plot in Fig. 5.4 panel (b).
Figure 5.7 Side elevation cut-away view of Fig. 5.4 (b). The beam angle is 0 deg.

This demonstrates the high quality of the baffle design. Independent of the surface coating, the geometry of the baffle is such that stray rays do not easily pass the pupil.

Figures 5.8 - 5.11 show the plots for the trials with beam angles of 8 deg and 16 deg. Similar observations can be made about these trials as were made for the trials with a beam angle of 0 deg.
Figure 5.8 Power flux highlights with 100,000 rays and a beam angle of 8 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 

79
Figure 5.9 Cut-away views of Fig. 5.8. The beam angle is 8 deg.
Figure 5.10 Power flux highlights with 100,000 rays and a beam angle of 16 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 
Figure 5.11 Cut-away views of Fig. 5.10. The beam angle is 16 deg.
5.3.1.1 Point-Spread Functions (PSF)

In addition to the 3-D power flux plots, the point-spread functions for these tests were also collected. Figures 5.12 - 5.14 show the point-spread functions corresponding to the plots in Figs. 5.4, 5.8, and 5.10.
Figure 5.12 Point-spread function plots for a ray-trace with 100,000 rays and a beam angle of 0 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 

84
Figure 5.13 Point-spread function plots for a ray-trace with 100,000 rays and a beam angle of 8 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 
Figure 5.14 Point-spread function plots for a ray-trace with 100,000 rays and a beam angle of 16 deg and $\alpha = 0.9$. For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$.

Comparing the two coatings for each angle shows the regular rays create the same PSF independent of the surface coating. This validates the consistency of the model. As always, more
stray rays reach the focal-plane in the more specular case than in the diffuse case. The fact that
this reinforces the results of the test set with 10,000 rays indicates that this phenomenon is not
dependent on the number of rays traced but is indeed a direct result of the specularity ratio of the
baffle surfaces. Additionally, as more rays are traced, the stray rays in the specular cases are
beginning to form distinct patterns. This is discussed more in connection with the results of the
next test.

To quantify the difference between the two surface coatings, Table 5.1 shows the number
of regular rays absorbed on the focal-plane and the number of stray rays absorbed on the focal-
plane for each trial. The table also represents the stray rays in each trial as a percentage of the
total number of rays on the focal-plane.

Table 5.1: Percentage of stray rays in each trial, $\alpha = 0.9$.

<table>
<thead>
<tr>
<th>Beam Angle</th>
<th>Specularity Ratio, $R$</th>
<th>Regular Rays</th>
<th>Stray Rays</th>
<th>Percent Stray</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.9</td>
<td>18137</td>
<td>111</td>
<td>0.61%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>18132</td>
<td>7</td>
<td>0.04%</td>
</tr>
<tr>
<td>8</td>
<td>0.9</td>
<td>18281</td>
<td>101</td>
<td>0.55%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>18256</td>
<td>4</td>
<td>0.02%</td>
</tr>
<tr>
<td>16</td>
<td>0.9</td>
<td>18343</td>
<td>51</td>
<td>0.28%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>18266</td>
<td>7</td>
<td>0.04%</td>
</tr>
</tbody>
</table>

Included in the stray ray quantities for each trial are the stray rays that exited the baffle at
the mirror end but missed the mirror and were absorbed by the imaginary enclosure. From Table
5.1, it can be seen that the percentage of stray rays is significantly higher in the specular cases. In
the trials with beam angles of 0 deg and 8 deg, the difference is greater by a factor of ten.

5.3.2 Test 2 Conclusions

The second test set supports the conclusions of the first test set. With the specular surface
coating, more stray rays escaped the baffles at the mirror end. With the diffuse surface coating,
fewer stray rays escaped the baffles at the mirror end and more were absorbed inside the baffle. With both surface coatings, most of the radiation absorbed in the baffles occurred before the circular pupil. This indicates that the baffle geometry is effective in beam forming.

5.4 Test 3: Power Flux Visualization with 1,000,000 rays

For the final test, 1,000,000 rays were traced using both surface coatings with a beam angle of 0 deg. As in the second test, the radiative heat flux to elements in the baffle were highlighted. This test was used to verify the results of the second test. The results of this test should be comparable to the results of two trials in the second test with a beam angle of 0 deg.

5.4.1 Test 3 Results

Figure 5.15 panels (a) and (b) show the 3-D images of results of the two ray-traces. Figure 5.16 (a) and (b) are cut-away images of Fig. 5.15.
Figure 5.15 Power flux highlights with 1,000,000 rays and a beam angle of 0 deg and $\alpha = 0.9$.

For the baffle in panel (a) $R = 0.9$ and in panel (b) $R = 0$. 
Figure 5.16 Cut-away views of Fig. 5.15. The beam angle is 0 deg.
More surfaces absorbed rays in this test than in the previous test; however, the surfaces in the diffuse case are still more populated than in the specular case. In contrast to the model in the second test, more shades are visible in the elements that absorbed radiation. As predicted, with more rays the model better represents the number of rays absorbed per unit area. Figure 5.17 shows the 3-D plots from Fig. 5.15 but viewed from the entrance aperture. A color bar has also been added to each figure to indicate the number of rays absorbed per square millimeter.
Figure 5.17 Front elevation view of panels (a) and (b) of Fig. 5.15. The beam angle is 0 deg. A color bar has been added to indicate power flux.

This is an improvement on Fig. 5.6. The distribution and magnitude of the heat flux in Fig. 5.17 (a) and (b) appear very similar to each other. Figure 5.17 shows the elements that absorbed the most rays per area surrounding the slots of each baffle vane. Figure 5.18 shows
another isometric view of the specular ray trace. In this image, the contrast between the parts of baffle vanes exposed to the incoming beam and the parts guarded from direct exposure is stark.

Figure 5.18 An isometric view of the 3-D model in Fig. 5.15 panel (a). The beam angle is 0 deg. A color bar has been added to indicate power flux.

For each baffle vane, the collection of brighter-colored elements roughly matches the shape of the slot in the vane before it. Figure 5.15 still shows most of the radiation absorbed in the baffles being absorbed before the circular aperture. It can be concluded from these ray-trace images that most of the heat flux in the baffle occurs before the pupil, and the highest heat flux for both the specular case and the diffuse case is on the baffle vanes around the slots of each vane; i.e., on the area directly exposed to the beam entering the aperture.

Figure 5.19 shows the point-spread functions for both surface coatings. As in the previous two tests, more stray rays appear on the focal-plane in the specular case. However, with more rays being traced, the number of stray rays has increased in the diffuse model as well.
While the stray rays in panels (a) and (b) cover roughly the same area, the distribution is very different between the two plots. The stray rays in Fig. 5.19 panel (a) form a distinct shape
with two symmetric lobes, intersecting in the center at the cluster of regular rays. This same pattern can be seen in the PSF plots in the previous test as well. The distribution of stray rays in panel (b) appears to be completely random. This difference between these two PSF plots highlights the difference in predictability of specular reflections and diffuse reflections. In specular reflection, once the direction of the incident ray is known, the reflected ray direction is determined. In the diffuse case, however, the reflected ray direction is random.

Table 5.2 shows the comparison of percent stray rays collected in both trials of this test. This table has the same form as Table 5.1.

Table 5.2: Percentage of stray rays in both trials, $\alpha = 0.9$.

<table>
<thead>
<tr>
<th>Beam Angle</th>
<th>Specularity Ratio, $R$</th>
<th>Regular Rays</th>
<th>Stray Rays</th>
<th>Percent Stray</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.9</td>
<td>181341</td>
<td>1045</td>
<td>0.57%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>182183</td>
<td>54</td>
<td>0.03%</td>
</tr>
</tbody>
</table>

Consistent with Table 5.1, Table 5.2 shows an order-of-magnitude difference between the percentage of stray rays in the diffuse case and the specular case.

### 5.4.2 Test 3 Conclusions

The third test set supports the conclusions of the first and second test sets. With the specular surface coating, more stray rays escape the baffles at the mirror end. With the diffuse surface coating, an order of magnitude fewer stray rays escape the baffles at the mirror end and more rays are absorbed inside the baffle. With both surface coatings, most of the radiation absorbed in the baffles occurs before the circular pupil. This indicates that the baffle geometry is effective in beam forming. For both the diffuse and the specular cases, the highest heat flux was concentrated on the faces of the baffle vanes exposed directly to the beam entering the aperture. This means that a subsequent conduction analysis will predict temperature excursions near the
slot edges as the Earth scene being viewed changes. The stray rays collected on the focal-plane in the specular case create a distinct symmetric pattern while the stray rays in the diffuse case are randomly distributed.
Chapter 6: Summary, Conclusions, and Suggested Future Effort

6.1 Summary of Effort

To summarize this thesis, the DEMETER instrument is a candidate satellite-embarked ERB instrument which, if selected, would replace the current-generation of instruments for monitoring radiation reflected and emitted from the Earth. Measurements from this instrument will be useful in analyzing the evolution of the planetary climate over time. The Virginia Tech TRG has been collaborating with NASA LaRC and Quartus Engineering to develop a ray-trace model of the DEMETER instrument.

The ray-trace model uses the MCRT method to trace rays through a 3-D model of the DEMETER instrument. This model includes a bounded planar surface representing the entrance aperture, an STL mesh representing the baffle, a 3-D sag expression representing the freeform mirror, and an infinite plane representing the focal plane array.

The ray-trace code allows the user to change the surface properties of the baffles, the number of rays traced, and some output settings. Also, because the geometry is imported from a
CAD file, it can be easily modified. The ray-trace code generates several outputs, useful for evaluating the results of a particular test.

6.2 Conclusions and Results

A set of example tests is documented to illustrate the capabilities of the ray-trace model. These tests show its capabilities including ray-path visualization, surface element power flux visualization, PSF generation, and regular and stray ray tracking. The results of these tests are discussed.

From the tests it is concluded that a diffuse surface coating such as Aeroglaze® Z306 paint would allow less stray radiation to leave the baffle at the mirror end of the instrument than a specular surface coating such as Aeroglaze® Z302. However, with a diffuse surface coating like Z306, more rays will be absorbed inside the baffle. This is a potential concern if it raises the temperature of the baffle walls significantly. The geometry of the baffles effectively forms the radiation beam from the entrance aperture and prevents most of the stray rays from getting past the circular pupil.

6.3 Future Effort

The TRG will continue to develop this ray-trace code and run more tests driven by the evolution of DEMETER. This section lists some improvements to the code that will be made in the near future. A brief outline of several future tests is given below.

6.3.1 Code Improvements

Now that the ray-trace code is functional, several augmentations and improvements are envisioned. The major changes are as follows:

- Add capability for diffuse emission from the aperture.
- Optimize the ray-trace code.
• Modify the baffle mesh.
• Modify baffle and aperture models.

6.3.1.1 Add Capability for Diffuse Emission from the Aperture

Of the changes listed, this change is the most straightforward. Currently, the model only supports collimated beams entering the aperture. This is useful for evaluating PSFs at discrete angles, but the actual detector will receive diffuse radiation (i.e., radiation from every angle simultaneously). To make this change would only require several additional lines of code following the method outlined in Mahan [9].

6.3.1.2 Optimize the Ray-Trace Code

The major limiting factor in completing tests with this ray-trace code is run-time. Tests with one million rays take 1-2 days to complete. One potential area of improvement is the algorithm used for searching the baffle mesh for intersection points. The method currently used is simple but may not be the fastest. Finding intersection points occupies most of the code run-time. A small improvement in the algorithm could decrease the run-time considerably. Once the code is running faster, more complicated meshes could be imported without increases in run-time.

6.3.1.3 Improve the Baffle Mesh

As mentioned in Chapters 3 and 5, the mesh is not yet ideal. A mesh that approximates the geometry well with relatively few yet well distributed elements has not yet been identified. More extensive tests with mesh design in FreeCAD and other CAD software could yield improved meshing schemes. Problems associated with meshing a 3-D model are not novel. Further research into the realm of finite element analysis could reveal better methods for meshing and handling meshes.
6.3.1.4 Modify Baffle and Aperture Models

The baffle modeled here is based on a preliminary design for the DEMETER baffle. Once the actual baffle design is frozen, the CAD file can be directly incorporated into the ray-trace code without having to remodel it in FreeCAD. Once the actual model is acquired, the baffles between the mirror and focal-plane would be incorporated in the ray-trace model. Also, the contemplated precision apertures in front of the detector array could be incorporated once a design is finalized. The focal-plane array would no longer be modeled as an infinite plane but instead would be bounded by the actual dimensions of the detector and discretized into individual detector pixels.

6.3.2 Future Tests

The listed changes would make the ray-trace code a more powerful tool and enable the user to run more meaningful tests. The following is a list of some tests that could be completed in the future using the ray-trace code:

- Trace rays diffusely from the aperture.
- Perform a parametric study of different focal plane positions and orientations.
- Trace rays from every mesh element and calculate element-to-element radiation distribution factors.
- Perform a reverse ray-trace from detector pixels into the instrument.
- Incorporate conduction heat transfer in the model to facilitate studies of uncorrected background emission equivalent radiance (UBEER).

6.3.2.1 Trace Rays Diffusely from the Aperture

Once the model is updated to emit rays diffusely from the aperture, ray-traces would be run to evaluate the performance of the system illuminated with diffuse radiation. The PSF plots
would be useful in evaluating the results. The program could be changed to color code the markers in the PSF according to the entrance angle of the rays they represent. Additionally, the code could be modified to create a probability density function (PDF) plot of the distribution of rays collected across the focal-plane. This would consist of dividing the focal-plane along the local $y$-axis into bins of equal width and summing the number of rays collected in each bin. The results would then be reported as a bar chart.

6.3.2.2 Carry Out a Parametric Study of Different Focal Plane Positions and Orientations

Without any modifications, the current ray-trace code could be used to evaluate the influence of different positions and tilt angles for the focal-plane. This would consist of adjusting the length of the chief ray segment between the mirror and the focal-plane, adjusting the tilt angle of the focal-plane, performing a ray-trace, and comparing the PSF to ray-traces with other effective focal-length and tilt angle settings. The goal would be to find the settings that yield the field-angle-weighted smallest mean spot size for the regular rays.

6.3.2.3 Trace Rays from Every Mesh Element and Calculate Distribution Factors

Distribution factors for every element in the baffle mesh would be required by a complete conduction heat transfer model of the baffle. To collect distribution factors for every surface element, rays would need to be traced from every surface element to every surface element, taking advantage of reciprocity where appropriate. This is conceptually simple but time intensive. This process could be commenced once the baffle design has been settled.

6.3.2.4 Perform a Reverse Ray-Trace from Detector Pixels into the Instrument

A common practice in ray-trace modeling is reverse ray-tracing. In a reverse ray-trace, rays are traced from the final destination to the radiation source [29]. In the DEMETER ray-trace model, this would mean tracing rays from a pixel of the focal-plane detector array back to the
entrance aperture. This is a way to evaluate other aspects of the model that would otherwise not be highlighted with the same number of rays traced from the aperture. This test would also serve to calculate the distribution factors for the focal-plane pixels and facilitate uncorrected background emission equivalent radiance (UBEER) studies.

6.3.2.5 Incorporate Conduction Heat Transfer in the Model to Facilitate UBEER Studies

As has been mentioned previously in this thesis, the ultimate motivation for computing radiation distribution factors is the need for a transient conduction heat transfer model. This will be essential for giving an accurate estimate of the temperatures of the different surfaces in the instrument. Conduction heat transfer affects the radiation heat transfer in the system. If sufficient energy is absorbed, emission from surfaces inside the instrument can become important. This contribution of background emission can be accounted for by conducting UBEER studies using the combined conduction and radiation model.
Works Cited


Appendix A: First 37 Zernike Polynomials in Cylindrical Coordinates

\[ Z_1 = 1 \]
\[ Z_2 = 2\rho \cos \phi \]
\[ Z_3 = 2\rho \sin \phi \]
\[ Z_4 = \sqrt{3}(2\rho^2 - 1) \]
\[ Z_5 = \sqrt{6}(\rho^2 \sin 2\phi) \]
\[ Z_6 = \sqrt{6}(\rho^2 \cos 2\phi) \]
\[ Z_7 = 2\sqrt{2}(3\rho^3 - 2\rho) \sin \phi \]
\[ Z_8 = 2\sqrt{2}(3\rho^3 - 2\rho) \cos \phi \]
\[ Z_9 = 2\sqrt{2}\rho^3 \sin 3\phi \]
\[ Z_{10} = 2\sqrt{2}\rho^3 \cos 3\phi \]
\[ Z_{11} = \sqrt{5}(6\rho^4 - 6\rho^2 + 1) \]
\[ Z_{12} = \sqrt{10}(4\rho^4 - 3\rho^2) \cos 2\phi \]
\[ Z_{13} = \sqrt{10}(4\rho^4 - 3\rho^2) \sin 2\phi \]
\[ Z_{14} = \sqrt{10}\rho^4 \cos 4\phi \]
\[ Z_{15} = \sqrt{10}\rho^4 \sin 4\phi \]
\[ Z_{16} = 2\sqrt{3}(10\rho^5 - 12\rho^3 + 3\rho) \cos \phi \]
\[ Z_{17} = 2\sqrt{3}(10\rho^5 - 12\rho^3 + 3\rho) \sin \phi \]
\[ Z_{18} = 2\sqrt{3}(5\rho^5 - 4\rho^3) \cos 3\phi \]
\[Z_{19} = 2\sqrt{3}(5\rho^5 - 4\rho^3) \sin 3\phi\]

\[Z_{20} = 2\sqrt{3}\rho^5 \cos 5\phi\]

\[Z_{21} = 2\sqrt{3}\rho^5 \sin 5\phi\]

\[Z_{22} = \sqrt{7}(20\rho^6 - 30\rho^4 + 12\rho^2 - 1)\]

\[Z_{23} = \sqrt{14}(15\rho^6 - 20\rho^4 + 6\rho^2) \sin 2\phi\]

\[Z_{24} = \sqrt{14}(15\rho^6 - 20\rho^4 + 6\rho^2) \cos 2\phi\]

\[Z_{25} = \sqrt{14}(6\rho^6 - 5\rho^4) \sin 4\phi\]

\[Z_{26} = \sqrt{14}(6\rho^6 - 5\rho^4) \cos 4\phi\]

\[Z_{27} = \sqrt{14}\rho^6 \sin 6\phi\]

\[Z_{28} = \sqrt{14}\rho^6 \cos 6\phi\]

\[Z_{29} = 4(35\rho^7 - 60\rho^5 + 30\rho^3 - 4\rho) \sin \phi\]

\[Z_{30} = 4(35\rho^7 - 60\rho^5 + 30\rho^3 - 4\rho) \cos \phi\]

\[Z_{31} = 4(21\rho^7 - 30\rho^5 + 10\rho^3) \sin 3\phi\]

\[Z_{32} = 4(21\rho^7 - 30\rho^5 + 10\rho^3) \cos 3\phi\]

\[Z_{33} = 4(7\rho^7 - 6\rho^5) \sin 5\phi\]

\[Z_{34} = 4(7\rho^7 - 6\rho^5) \cos 5\phi\]

\[Z_{35} = 4\rho^7 \sin 7\phi\]

\[Z_{36} = 4\rho^7 \cos 7\phi\]

\[Z_{37} = \sqrt{9}(70\rho^8 - 140\rho^6 + 90\rho^4 - 20\rho^2 + 1)\]
Appendix B: First 37 Zernike Polynomials in Cylindrical Coordinates

\[ Z_1 = 1 \]
\[ Z_2 = \frac{2x}{R_{norm}} \]
\[ Z_3 = -\frac{2y}{R_{norm}} \]
\[ Z_4 = \frac{2 \sqrt{3}(x^2 + y^2)}{R_{norm}^2} - \sqrt{3} \]
\[ Z_5 = -\frac{2 \sqrt{6}xy}{R_{norm}^2} \]
\[ Z_6 = \frac{\sqrt{6}(x^2 - y^2)}{R_{norm}^2} \]
\[ Z_7 = -\frac{2 \sqrt{2}y(-2 R_{norm}^2 + 3 x^2 + 3 y^2)}{R_{norm}^3} \]
\[ Z_8 = \frac{2 \sqrt{2}x(-2 R_{norm}^2 + 3 x^2 + 3 y^2)}{R_{norm}^3} \]
\[ Z_9 = -\frac{2 \sqrt{2}y(3 x^2 - y^2)}{R_{norm}^4} \]
\[ Z_{10} = \frac{2 \sqrt{2}x(x^2 - 3 y^2)}{R_{norm}^4} \]
\[ Z_{11} = \sqrt{5} + \frac{6 \sqrt{5}(x^2 + y^2)(-R_{norm}^4 + x^2 + y^2)}{R_{norm}^4} \]
\[ Z_{12} = \frac{\sqrt{10}(x^2 - y^2)(-3 R_{norm}^4 + 4 x^2 + 4 y^2)}{R_{norm}^4} \]
\[
Z_{13} = -\frac{2\sqrt{10}xy (-3 R_{\text{norm}}^2 + 4x^2 + 4y^2)}{R_{\text{norm}}^4}
\]

\[
Z_{14} = \frac{\sqrt{10}(x^4 - 6x^2y^2 + y^4)}{R_{\text{norm}}^4}
\]

\[
Z_{15} = -\frac{4\sqrt{10}xy (x^2 - y^2)}{R_{\text{norm}}^4}
\]

\[
Z_{16} = \frac{6\sqrt{3}R_{\text{norm}}^4x + 20\sqrt{3}x(x^2 + y^2)^2 - 24\sqrt{3}R_{\text{norm}}^2x(x^2 + y^2)}{R_{\text{norm}}^5}
\]

\[
Z_{17} = -\frac{6\sqrt{3}R_{\text{norm}}^4y + 20\sqrt{3}y(x^2 + y^2)^2 - 24\sqrt{3}R_{\text{norm}}^2y(x^2 + y^2)}{R_{\text{norm}}^5}
\]

\[
Z_{18} = \frac{2\sqrt{3}x(x^2 - 3y^2)(-4R_{\text{norm}}^2 + 5x^2 + 5y^2)}{R_{\text{norm}}^5}
\]

\[
Z_{19} = -\frac{2\sqrt{3}y(3x^2 - y^2)(-4R_{\text{norm}}^2 + 5x^2 + 5y^2)}{R_{\text{norm}}^5}
\]

\[
Z_{20} = \frac{2\sqrt{3}x(x^4 - 10x^2y^2 + 5y^4)}{R_{\text{norm}}^5}
\]

\[
Z_{21} = -\frac{2\sqrt{3}y(5x^4 - 10x^2y^2 + y^4)}{R_{\text{norm}}^5}
\]

\[
Z_{22} = \frac{20\sqrt{7}(x^2 + y^2)^3 - 30\sqrt{7}(x^2 + y^2)^2}{R_{\text{norm}}^6} - \sqrt{7} + \frac{12\sqrt{7}(x^2 + y^2)}{R_{\text{norm}}^2}
\]

\[
Z_{23} = -\frac{12\sqrt{14}xy}{R_{\text{norm}}^2} + \frac{40\sqrt{14}xy(x^2 + y^2)}{R_{\text{norm}}^4} - \frac{30\sqrt{14}xy(x^2 + y^2)^2}{R_{\text{norm}}^6}
\]

\[
Z_{24} = \frac{\sqrt{14}(x^2 - y^2)(6R_{\text{norm}}^4 - 20R_{\text{norm}}^2x^2 - 20R_{\text{norm}}^2y^2 + 15x^4 + 30x^2y^2 + 15y^4)}{R_{\text{norm}}^6}
\]

\[
Z_{25} = -\frac{4\sqrt{14}xy(x^2 - y^2)(-5R_{\text{norm}}^2 + 6x^2 + 6y^2)}{R_{\text{norm}}^6}
\]

\[
Z_{26} = \frac{\sqrt{14}(-5R_{\text{norm}}^2 + 6x^2 + 6y^2)(x^4 - 6x^2y^2 + y^4)}{R_{\text{norm}}^6}
\]
\[ Z_{27} = -\frac{2 \sqrt{14} x y (x^2 - 3 y^2)(3 x^2 - y^2)}{R_{\text{norm}}^6} \]
\[ Z_{28} = \frac{2 \sqrt{14} x^2 (x^2 - 3 y^2)^2 - \sqrt{14} (x^2 + y^2)^3}{R_{\text{norm}}^6} \]
\[ Z_{29} = \frac{140 y (x^2 + y^2)^3 - 16 R_{\text{norm}}^6 y - 240 R_{\text{norm}}^2 y (x^2 + y^2)^2 + 120 R_{\text{norm}}^4 y (x^2 + y^2)}{R_{\text{norm}}^7} \]
\[ Z_{30} = \frac{140 x (x^2 + y^2)^3 - 16 R_{\text{norm}}^6 x - 240 R_{\text{norm}}^2 x (x^2 + y^2)^2 + 120 R_{\text{norm}}^4 x (x^2 + y^2)}{R_{\text{norm}}^7} \]
\[ Z_{31} = -\frac{4 y (3 x^2 - y^2)(10 R_{\text{norm}}^4 - 30 R_{\text{norm}}^2 x^2 - 30 R_{\text{norm}}^2 y^2 + 21 x^4 + 42 x^2 y^2 + 21 y^4)}{R_{\text{norm}}^7} \]
\[ Z_{32} = \frac{4 x (x^2 - 3 y^2)(10 R_{\text{norm}}^4 - 30 R_{\text{norm}}^2 x^2 - 30 R_{\text{norm}}^2 y^2 + 21 x^4 + 42 x^2 y^2 + 21 y^4)}{R_{\text{norm}}^7} \]
\[ Z_{33} = -\frac{4 y (-6 R_{\text{norm}}^2 + 7 x^2 + 7 y^2)(5 x^4 - 10 x^2 y^2 + y^4)}{R_{\text{norm}}^7} \]
\[ Z_{34} = \frac{4 x (-6 R_{\text{norm}}^2 + 7 x^2 + 7 y^2)(x^4 - 10 x^2 y^2 + 5 y^4)}{R_{\text{norm}}^7} \]
\[ Z_{35} = \frac{4 y (56 x^4 y^2 - 7 (x^2 + y^2)^3 + 8 y^6)}{R_{\text{norm}}^7} \]
\[ Z_{36} = \frac{4 x (56 x^2 y^4 - 7 (x^2 + y^2)^3 + 8 x^6)}{R_{\text{norm}}^7} \]
\[ Z_{37} = \frac{270 (x^2 + y^2)^2}{R_{\text{norm}}^4} - \frac{60 (x^2 + y^2)}{R_{\text{norm}}^2} - \frac{420 (x^2 + y^2)^3}{R_{\text{norm}}^6} + \frac{210 (x^2 + y^2)^4}{R_{\text{norm}}^8} + 3 \]
Appendix C: Full Ray-Trace Code

%This program traces rays from a slot shaped aperture through a
%baffle, to a mirror and then to a focal plane. The distribution factors
%from the aperture to all surfaces in the system are calculated.

clear;
clc;
tic %start timing program

Ray_Num = 1000000; %Number of Rays
Col_Ang = 0; %Angle of columnated beam

absorb = 0.9; %absorptivity of surfaces
spec = 0.9; %specularity of surfaces

OpPath = false; %Enable/Disable Optical Path View
FluxHL = true; %Enable/Disable Flux Highlights

%V2: this program translates and rotates the baffles and aperture instead of
%the mirror but puts everything back for the graphics. In other words, the
%ray trace is done in a local coordinate system but everything is
%transferred back to the global coordinate system after the ray trace is
%completed. If the user would like to see the optical paths of all rays,
%they will have to keep the graphics in the local coordinate system.

%V3: Now the user doesn't have to worry about commenting out and
%uncommenting blocks of the code to hide or view the optical paths of the
%rays. Set OpPath to true to see optical paths and set it to false to hide
%them. Set FluxHL to True to see the elements in the baffles that absorb rays
%highlighted according to flux.
%Additionally, the program now outputs the point spread function in the
%local coordinates of the focal plane. The plot is centered about the point
%pFM where the chief ray is absorbed on the focal plane. Blue points
%represent optical rays, red points represent stray rays.

%V5: The two matlab figure outputs of this program are automatically saved
%along with jpeg image files of both figures. All numeric output of the
%program is saved to an excel spread sheet. The files generated from this
%program are named according to the number of rays traced, the specularity
%ratio and absorptivity of the baffles surfaces, the angle of the
%collimated beam, and the time the code started running.

Ang = XXXXXX; %tilt of mirror
tM = XXXXXXXXX;%focul length
Ap2Pup = XXXXXX; %distance from entrance aperture to pupil
Pup2Mir = XXXXXXXX; %distance from pupil to mirror vertex
MirYOff = XXXXXXXXXXX; %y-displacement of mirror vertex

%mirror limits
x_min = -40;
x_max = 40;
y_min = -45;
y_max = 5;

%%%%%%%%%%%%%%%%%%%%%%%%%%%Import Baffle Mesh%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%additional mesh names
%note: to use one of these meshes in the ray trace, it must be saved to the
%same file folder as the ray-trace program
'Baffle_V3_M9.stl'
%fileID = fopen('Baffle_V3_M9.stl','r'); %read this mesh
NameExtract = fgets(fileID);
MeshName = NameExtract(7:length(NameExtract)-2);
ReadTemp1 = [' facet normal ' '%f' '%f' '%f' '
outer loop
vertex' '%f' '%f'
vertex' '%f' '%f' '%f'
endloop
endfacet'];
ReadTemplate=horzcat(ReadTemp1,ReadTemp2);
MeshMat = fscanf(fileID,ReadTemplate,[12,inf]);
fclose(fileID);

%%%%%%%%%%%%%%%%%%%%%%%%%%Add Enclosure%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%this block of code adds a triangular mesh to the model that represents the
%imaginary enclosure surrounding the model.
Box_Mat = 
[B0   -1    0  -BZ  BZ   -BZ  BZ  BZ  -BZ  BZ  BZ  -BZ; 
 1    0    0  -BZ  -BZ  -BZ  BZ  BZ  BZ  BZ  BZ  -BZ;
 1 0  -1  -BZ  BZ  -BZ  -BZ  BZ  BZ  BZ  BZ  -BZ;
-1 0  -1  -BZ  BZ  BZ  -BZ  BZ  BZ  BZ  BZ  -BZ;
-1 0    0  -BZ  BZ  BZ  BZ  BZ  BZ  BZ  BZ  -BZ;
 0    0  -1  -BZ  BZ  BZ  BZ  BZ  BZ  BZ  BZ  -BZ;
 0 0  -1  -BZ  BZ  BZ  -BZ  BZ  BZ  BZ  BZ  -BZ;
 0 0    1  -BZ  BZ  -BZ  -BZ  BZ  -BZ  -BZ  -BZ;
 0 0    1  -BZ  BZ  BZ  -BZ  -BZ  -BZ  -BZ  -BZ;
 0 1    0  -BZ  BZ  -BZ  -BZ  -BZ  -BZ  -BZ  -BZ;
 0 1    0  -BZ  BZ  -BZ  -BZ  -BZ  -BZ  -BZ  -BZ;
];
MeshMat = [MeshMat transpose(Box_Mat)];
ElNum = length(MeshMat); %number of elements in mesh

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Aperture Rays%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This program generates rays in a slot aperture in its own local coordinate
%system. The rays between the round ends and center block are balanced by
%area ratios
L_slot = 65; %slot length in mm ( +---65 mm----+ )
W_slot = 30; %slot width (aka diameter of round ends)

slot_lim1 = [((L_slot+W_slot)/2 W_slot/2 0);
slot_lim2 = -slot_lim1;

A_box = L_slot*W_slot;  %Area of rectangular middle section

A_circ = pi*W_slot^2/4;  %Area of BOTH round ends together

N_box = round(Ray_Num*A_box/(A_box+A_circ));  %Number of rays in rectangle part

N_circ = Ray_Num - N_box;  %number of rays in round ends

pa = zeros(Ray_Num,3);  %coordinates of each ray point in aperture

%populate rectangle
for i=1:N_box
   pa(i,1) = (rand()-0.5)*L_slot;
   pa(i,2) = (rand()-0.5)*W_slot;
end

%populate round ends
for i=N_box+1:Ray_Num
   rnr = W_slot/2*sqrt(rand());  %random radius
   phinr = 2*pi*rand();  %random angle

   pa(i,1) = rnr*cos(phinr);  %random x
   pa(i,2) = rnr*sin(phinr);  %random y

   if pa(i,1)<0
      pa(i,1) = pa(i,1)-L_slot/2;
   else
      pa(i,1) = pa(i,1)+L_slot/2;
   end
end

%vectors for ray absorption
power = zeros(ElNum,1);
aPower = 0;
fpPower = 0;
vM = [0 0 1];
nA = [0 0 1];  %aperture normal

fpTilt = XXXXXXXXXXXX;
f = [0 cosd(90+fpTilt) cosd(-fpTilt)];  %focal plane normal

%---------------------------------------------------------------
%in this block, all relevant components of the model are rotated and %translated into the local coordinate system of the mirror in order to run %a ray-trace
%---------------------------------------------------------------

%Aperture
pal = transpose(pa);
pal = Glob2Loc(pa);

%focal plane

nf = transpose(Glob2Loc(transpose(nf)));

% %Baffle
MeshMat(1:3,:) = Glob2Loc(MeshMat(1:3,:));
MeshMat(4:6,:) = Glob2Loc(MeshMat(4:6,:));
MeshMat(7:9,:) = Glob2Loc(MeshMat(7:9,:));
MeshMat(10:12,:) = Glob2Loc(MeshMat(10:12,:));

%rays
vM = transpose(Glob2Loc(transpose(vM)));
nA = transpose(Glob2Loc(transpose(nA)));
slot_lim1 = transpose(Glob2Loc(transpose(slot_lim1)));
slot_lim2 = transpose(Glob2Loc(transpose(slot_lim2)));

va = ones(Ray_Num,1)*transpose(Glob2Loc([sind(Col_Ang); 0; cosd(Col_Ang);]));

%translate
Offset = [0; MirYOff; (Ap2Pup+Pup2Mir)];
Offset = Glob2Loc(Offset);

Origin = -[Offset(1) Offset(2) Offset(3)];

%baffles
MeshMat(5:3:11,:) = MeshMat(5:3:11,:) - Offset(2);
MeshMat(6:3:12,:) = MeshMat(6:3:12,:) - Offset(3);

%aperture
pal(2,:) = pal(2,:) - Offset(2);
pal(3,:) = pal(3,:) - Offset(3);

slot_lim1 = slot_lim1-Offset;
slot_lim2 = slot_lim2-Offset;

pa = transpose(pal);
MeshMat = transpose(MeshMat);

syms xv yv R r_norm %sphere(xv,yv,R)
R_sph = XXXXXX;

r_n = XXXXXX;

y_offset = XXXXXXXX;

asph = xv*ones(8,1);

%aspheric coefficients
alpha = zeros(1,8);
% insert alpha values here

for i=1:8
    asph(i) = (xv^2+yv^2)^i;
end

% Zernike Coefficients
A=zeros(1,37);
% insert Zernike coefficients here

% Zernike Polynomials
zrk = xv*ones(37,1);

zrk(1) = 1;
zrk(2) = (2*xv)/r_norm;
zrk(3) = -(2*yv)/r_norm;
zrk(4) = (2*3^(1/2)*(xv^2 + yv^2))/r_norm^2 - 3^(1/2);
zrk(5) = -(2*6^(1/2)*(xv*yv))/r_norm^2;
zrk(6) = (6*^2 + yv^2))/r_norm^2;
% i=1:20
% zrk = 5^(1/2) + 6*5^(1/2)*(xv^2 + yv^2)*(xv^2 - r_norm^2 + yv^2))/r_norm^4;
% zrk(21) = -(2*6^(1/2)*(xv^2 + yv^2))/(xv^2 + yv^2))/r_norm^5;
% zrk(22) = (2*6^(1/2)*(xv^2 + yv^2))/r_norm^5;
% zrk(23) = -(2*6^(1/2)*(xv^2 + yv^2))/r_norm^6;
% zrk(24) = (2*6^(1/2)*(xv^2 + yv^2))/r_norm^6;
% zrk(25) = -(2*6^(1/2)*(xv^2 + yv^2))/r_norm^6;
% zrk(26) = (2*6^(1/2)*(xv^2 + yv^2))/r_norm^7;
% zrk(27) = -(2*6^(1/2)*(xv^2 + yv^2))/r_norm^7;
\[ zrk(31) = -(4yv^2(3xv^2 - yv^2)(42xv^2yv^2 - 30r_{norm}^2yv^2 - 30r_{norm}^2xv^2 + 10r_{norm}^4 + 21xv^4 + 21yv^4))/r_{norm}^7; \]
\[ zrk(32) = (4xv(7xv^2 - 6r_{norm}^2 + 7yv^2)(5xv^4 - 10xv^2yv^2 + yv^4))/r_{norm}^7; \]
\[ zrk(33) = -(4yv(7xv^2 - 6r_{norm}^2 + 7yv^2)(5xv^4 - 10xv^2yv^2 + yv^4))/r_{norm}^7; \]
\[ zrk(34) = (4xv(7xv^2 - 6r_{norm}^2 + 7yv^2)(5xv^4 - 10xv^2yv^2 + yv^4))/r_{norm}^7; \]
\[ zrk(35) = (4yv((56xv^4yv^2 - 7(xv^2 + yv^2)^3 + 8yv^6))/r_{norm}^7; \]
\[ zrk(36) = (4xv((56xv^2yv^4 - 7(xv^2 + yv^2)^3 + 8xv^6))/r_{norm}^7; \]
\[ zrk(37) = (270(xv^2 + yv^2)^2)/r_{norm}^4 - (60xv^2 + 60yv^2)/r_{norm}^2 - (420(xv^2 + yv^2)^3)/r_{norm}^6 + (210(xv^2 + yv^2)^4)/r_{norm}^8 + 3; \]

\[ \text{syms } sag(xv,yv,R,r_{norm}) \text{ sag2}(xv,yv) \text{ zernike}(xv,yv,r_{norm}); \]
\[ \text{zernike}(xv, yv, r_{norm}) = A\cdot zrk; \]
\[ \% \text{general sagitta} \]
\[ \text{sg}(xv,yv,R,r_{norm}) = -(R-\sqrt{R^2-xv^2-yv^2})+\alpha \cdot \text{asph}+\text{zernike}(xv,\ y\_offset+yv,r_{norm}); \]
\[ \% \text{specific sagitta with } R=R_{sph} \text{, and } r_{norm} = r_n \]
\[ \text{sg}(xv,yv) = \text{sg}(xv,yv,R_{sph},r_n); \]
\[ \% \text{get x and y gradients of mirror sag} \]
\[ \text{sgx} = \text{diff}(\text{sg},xv); \]
\[ \text{sgy} = \text{diff}(\text{sg},yv); \]
\[ \text{sg} = \text{vpa}(\text{sg}); \]
\[ \text{sgx} = \text{vpa}(\text{sgx}); \]
\[ \text{sgy} = \text{vpa}(\text{sgy}); \]

\[ \text{syms } \text{Surf1}(P1, P2, P3, tv, V1, V2, V3) \]
\[ \text{Surf1}(P1, P2, P3, tv, V1, V2, V3) = \text{sg}(V1*\text{tv}+P1,V2*\text{tv}+P2)-(V3*\text{tv}+P3); \]

\[ \% \text{trace a chief ray to define the focal plane} \]
\[ \text{tt} = \text{Surf1}(\text{Origin}(1),\text{Origin}(2),\text{Origin}(3),\text{tv},\text{vm}(1),\text{vm}(2),\text{vm}(3)); \]
\[ \text{t} = \text{vpasolve}(\text{tt}\text{,tv},150); \]
\[ \text{pmM}\text{=} \text{Origin}+\text{vmM}\text{t}; \]
\[ \text{nM} = \{-\text{sgx}(\text{pmM}(1),\text{pmM}(2))\ \text{sgy}(\text{pmM}(1),\text{pmM}(2))\ -1\}; \]
\[ \text{nM}(1) = 0; \]
\[ \text{nM} = \text{vmM}/\sqrt{(\text{nM}(1)^2+\text{nM}(2)^2+\text{nM}(3)^2)}; \]
\[ \text{vrM} = \text{vmM}^2\cdot\text{dot}(\text{vmM},\text{nM})\cdot\text{nM}; \% \text{reflected direction of chief ray} \]
\[ \text{pfM} = \text{pmM}+\text{vrM}\text{tM}; \% \text{point chief ray intersects focal plane} \]
\[ \% \text{scatter3(pfM(1),pfM(2),pfM(3),'o','k');} \]
pf = zeros(Ray_Num,3);
%optical ray tracker
Optical = ones(Ray_Num,1);

%ray reflection tracker
refl = zeros(Ray_Num,1);
%combine ray path images
if(OpPath)
    hold on
end
%scatter3(Origin(1),Origin(2),Origin(3),''','m')

%%%%%%%%%%%%%%%%%%%%%%%%%%Ray Trace%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:Ray_Num %trace every ray till it terminates
    p0 = pa(i,:);
    v0 = va(i,:);
    while 1
        el_old = 0; %variable for element where ray hits
        t_old = inf; %variable for ray length
        for j = 1:ElNum %check every element
            dir_check = dot(MeshMat(j,1:3),v0); %Check direction of ray
            %and surface are compatible
            if (dir_check < 0)
                t_new = dot(MeshMat(j,1:3),(MeshMat(j,4:6)-p0))/(dir_check);
                %NOTE: may need to say >= 0. In complex shapes may be ruling
                out
                %right answer
                if (t_new > 0)
                    p1 = p0+v0*t_new; %find point of intersection
                    %vectors around triangular element
                    edge0 = MeshMat(j,7:9)-MeshMat(j,4:6);
                    edge1 = MeshMat(j,10:12)-MeshMat(j,7:9);
                    edge2 = MeshMat(j,4:6)-MeshMat(j,10:12);
                    pVec0 = p1-MeshMat(j,4:6);
                    pVec1 = p1-MeshMat(j,7:9);
                    pVec2 = p1-MeshMat(j,10:12);
                    TriTest0 = dot(MeshMat(j,1:3),cross(edge0,pVec0));
                    TriTest1 = dot(MeshMat(j,1:3),cross(edge1,pVec1));
                    TriTest2 = dot(MeshMat(j,1:3),cross(edge2,pVec2));


if(TriTest0>0 && TriTest1>0 && TriTest2>0)

% if the current element is closer than the old one.

Update

% to new element
if (t_new <= t_old)
  el_old = j;
  t_old = t_new;
  p_old = p1;
  n_old = MeshMat(j,1:3);
end
end
end
end
end
end
end
end

% check aperture
APcheck = dot(nA,v0);
if APcheck < 0 % if facing the right way
  t_new = dot(nA,Origin-p0)/APcheck;
  if t_new<=t_old % if the length is shorter
    pAP = p0+v0*t_new;
    if pAP(1)>= slot_lim1(1) && pAP(1) <= slot_lim2(1) % if within
      x range
    % if abs(pAP(1))<L_slot/2 % if in rectangular portion
      if pAP(2)>=slot_lim1(2) && pAP(2) <= slot_lim2(2) % if
        within y range
          el_old = ElNum+2;
          t_old = t_new;
          p_old = pAP;
        end
      end
    end
  end
end
end
end
end

% check the mirror
tmir = Surf1(p0(1),p0(2),p0(3),tv,v0(1),v0(2),v0(3));
t_new = vpasolve(tmir,tv,160);
if(~isempty(t_new))
  if t_new>=0 && t_new <= t_old
    pm = p0+v0*t_new;
    if pm(1) <= x_max && pm(1) >= x_min && pm(2) <= y_max && pm(2) >= y_min
      nm = [sag2x(pm(1),pm(2)) sag2y(pm(1),pm(2)) -1];
      nm = nm/sqrt(nm(1)^2+nm(2)^2+nm(3)^2);
      dir_check = dot(nm,v0); % Check direction of ray and
      surface are compatible
      if (dir_check < 0)
        vm = v0-2*dot(v0,nm)*nm;
        vm = vm/sqrt(vm(1)^2+vm(2)^2+vm(3)^2);
        el_old = ElNum+1;
        t_old = t_new;
        p_old = pm;
      end
    end
  end
end
end
end
end
end
end
end
end
end
n_old = nm;
t2 = dot(nf, pfM - pm)/dot(nf, vm);
pf(i,:) = pm + vm*t2;
% uncomment for ray paths (note: must uncomment 2
% lines below as well
if (OpPath)
    plot3([p_old(1) pf(i,1)],[p_old(2) pf(i,2)],[p_old(3) pf(i,3)],'b');
end
end
end
end
% also uncomment for ray paths
if (OpPath)
    plot3([p0(1) p_old(1)],[p0(2) p_old(2)],[p0(3) p_old(3)],'r');
    scatter3(p_old(1), p_old(2), p_old(3), '.', 'b');
end
if el_old == ElNum+1 % Hit Mirror and focal plane
    % p0 = p_old;
    % v0 = vm;
    fpPower = fpPower+1;
    break;
elseif el_old == ElNum+2 % Exited Aperture
    aPower = aPower+1;
    break;
elseif el_old > ElNum-12
    power(el_old) = power(el_old)+1;
    break;
elseif (rand > absorb) % Hit anywhere else and reflected
    refl(i) = refl(i)+1;
    Optical(i) = 0;
p0 = p_old;
    if (rand < spec) % specular
        v0 = v0-2*dot(v0, n_old)*n_old;
    else % Diffuse
        ther = acos(2*rand-1);
        phir = 2*pi*rand;
        tr = sqrt(2*(1+n_old(1)*sin(ther)*cos(phir)+n_old(2)*sin(ther)*sin(phir)+n_old(3)*cos(ther)));
        v0 = [n_old(1)+sin(ther)*cos(phir)
        n_old(2)+sin(ther)*sin(phir) n_old(3)+cos(ther)]/tr;
    end
    else % Absorbed
        power(el_old) = power(el_old)+1;
        break;
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%% Distribution Factors
D_Factor = [power; fpPower; aPower;]/Ray_Num;

118
Flux = zeros(ElNum,1);

for i=1:ElNum
    Sa = sqrt((MeshMat(i,4)-MeshMat(i,7))^2+(MeshMat(i,5)-MeshMat(i,8))^2+(MeshMat(i,6)-MeshMat(i,9))^2);
    Sb = sqrt((MeshMat(i,7)-MeshMat(i,10))^2+(MeshMat(i,8)-MeshMat(i,11))^2+(MeshMat(i,9)-MeshMat(i,12))^2);
    Sc = sqrt((MeshMat(i,4)-MeshMat(i,10))^2+(MeshMat(i,5)-MeshMat(i,11))^2+(MeshMat(i,6)-MeshMat(i,12))^2);
    TA = 1/4*sqrt((Sa+Sb-Sb)*(Sa-Sb+Sc)*(-Sa+Sb+Sc)*(Sa+Sb+Sc));
    Flux(i) = power(i)/TA;
end

mFP = 1;
nFP = 1;
kFP=1;

OpRay = zeros(Ray_Num,3);
NonOpRay = zeros(Ray_Num,3);
pf1 = zeros(Ray_Num,4);

for i=1:Ray_Num
    if pf(i,2) ~= 0 && pf(i,3) ~= 0
        pf1(kFP,1:3) = pf(i,:);
        pf1(kFP,4) = Optical(i);
        kFP=kFP+1;
        if Optical(i) == 1
            OpRay(mFP,:) = pf(i,:);
            mFP = mFP+1;
        else
            NonOpRay(nFP,:) = pf(i,:);
            nFP = nFP+1;
        end
    end
end

pf=pf1(1:kFP-1,:);
OpRay = OpRay(1:mFP-1,:);

if(nFP>1)
    NonOpRay = NonOpRay(1:nFP-1,:);
end
if (~OpPath)
    pa = transpose(pa);
    pa(2,:) = pa(2,:) + Offset(2);
    pa(3,:) = pa(3,:) + Offset(3);
    pa = transpose(Loc2Glob(pa));
    slot_lim1 = slot_lim1 + Offset;
    slot_lim2 = slot_lim2 + Offset;
    slot_lim1 = transpose(Loc2Glob(transpose(slot_lim1)));
    slot_lim2 = transpose(Loc2Glob(transpose(slot_lim2)));

    pfM(2) = pfM(2) + Offset(2);
    pfM(3) = pfM(3) + Offset(3);
    pfM = transpose(Loc2Glob(transpose(pfM)));
    pf = transpose(pf);
    pf(2,:) = pf(2,:) + Offset(2);
    pf(3,:) = pf(3,:) + Offset(3);
    pf(1:3,:) = Loc2Glob(pf(1:3,:));
    pf = transpose(pf);
    OpRay = transpose(OpRay);
    OpRay(2,:) = OpRay(2,:) + Offset(2);
    OpRay(3,:) = OpRay(3,:) + Offset(3);
    OpRay = transpose(Loc2Glob(OpRay));
    nf = transpose(Loc2Glob(transpose(nf)));
    vrM = transpose(Loc2Glob(transpose(vrM)));

    if nFP>0
        NonOpRay = transpose(NonOpRay);
        NonOpRay(2,:) = NonOpRay(2,:) + Offset(2);
        NonOpRay(3,:) = NonOpRay(3,:) + Offset(3);
        NonOpRay = transpose(Loc2Glob(NonOpRay));
    end

    MeshMat = transpose(MeshMat);
    MeshMat(5:3:11,:) = MeshMat(5:3:11,:) + Offset(2);
    MeshMat(6:3:12,:) = MeshMat(6:3:12,:) + Offset(3);

    MeshMat(1:3,:) = Loc2Glob(MeshMat(1:3,:));
    MeshMat(4:6,:) = Loc2Glob(MeshMat(4:6,:));
    MeshMat(7:9,:) = Loc2Glob(MeshMat(7:9,:));
    MeshMat(10:12,:) = Loc2Glob(MeshMat(10:12,:));

    MeshMat = transpose(MeshMat);
    vM = transpose(Loc2Glob(transpose(vM)));
    nA = transpose(Loc2Glob(transpose(nA)));
Offset = Loc2Glob(Offset);

Origin = [0 0 0];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%Graphics%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%Baffle Graphics%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Xg = transpose(MeshMat(:,4:3:10));
Yg = transpose(MeshMat(:,5:3:11));
Zg = transpose(MeshMat(:,6:3:12));

MeshSurf = patch(Xg(1:ElNum-12),Yg(1:ElNum-12),Zg(1:ElNum-12),'b','EdgeColor','none','FaceAlpha',0.1);%,'FaceAlpha',0.7);
view(3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%Flux Highlights%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (FluxHL)
Xf = zeros(3,ElNum);
Yf = zeros(3,ElNum);
Zf = zeros(3,ElNum);
Ff = zeros(ElNum,1);
kf = 1;
for i=1:ElNum-12
if Flux(i)>0
Xf(:,kf) = Xg(:,i);
Yf(:,kf) = Yg(:,i);
Zf(:,kf) = Zg(:,i);
Ff(kf) = Flux(i);
kf=kf+1;
end
end
Xf = Xf(:,1:kf-1);
Yf = Yf(:,1:kf-1);
Zf = Zf(:,1:kf-1);
Ff = Ff(1:kf-1);
MeshSurf2 = patch(Xf,Yf,Zf,Ff,'EdgeColor','none');
view(3);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Graphics General%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Fix scaling of figure
ax = gca;
ax.DataAspectRatio = [1 1 1];

%label axes
xlabel("x-axis (mm)");
ylabel("y-axis (mm)");
zlabel("z-axis (mm)");
%fix view
%control angles for view
if ~OpPath
  vang1 = -125.9248;
  vang2 = -39.0693;
  rollang = -65;
else
  vang1 = 87.5971;
  vang2 = -41.3156;
  rollang = -115;
end
view([vang1 vang2]);
camroll(rollang);

hold on

% Aperture Graphics
% scatter3(pa(:,1),pa(:,2),pa(:,3),'r','.'); % uncomment to see ray genesis

% Mirror Graphics
x_step = 2;
y_step = 2;

[mX,mY] = meshgrid(x_min:x_step:x_max,y_min:y_step:y_max);
[I,J] = size(mX);

m2 = double(sag2(mX,mY));

if(~OpPath)
  for i=1:I
    for j=1:J
      pvec2 = Loc2Glob([mX(i,j); mY(i,j); mZ(i,j)]);
      mX(i,j) = pvec2(1);
      mY(i,j) = pvec2(2);
      mZ(i,j) = pvec2(3);
    end
  end
  mZ = mZ+Offset(3);
  mY = mY+Offset(2);
end

S3 = surf(mX,mY,mZ,'FaceColor','g','EdgeColor','none');%,'FaceAlpha',0.3);

% Focal Plane
syms sf(x,y)
pc = pfM;
sf(x,y) = pc(3)-(nf(1)*(x-pc(1))+nf(2)*(y-pc(2)))/nf(3);
dx = 30;
dy = 6;

X2 = [pc(1)+dx; pc(1)+dx; pc(1)-dx; pc(1)-dx];
Y2 = [pc(2)+dy; pc(2)-dy; pc(2)-dy; pc(2)+dy];
Z2 = zeros(4,1);

for i=1:4
    Z2(i) = sf(X2(i),Y2(i));
end

SFP = fill3(X2,Y2,Z2,'y','FaceAlpha',0.6,'EdgeColor','none');

%optical rays
OPFP = scatter3(OpRay(:,1),OpRay(:,2),OpRay(:,3),'.','b');
if nFP>1
    NONOPFP = scatter3(NonOpRay(:,1),NonOpRay(:,2),NonOpRay(:,3),'.','r');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Figure Name
 RayTag = 'R'+string(Ray_Num); %part of file name indicating Ray number
AbsorbTag ='_AB'+string(absorb*10); %part of file name indicating absorptivity
SpecTag ='_SP'+string(spec); %part of file name indicating specularity
BeamAngTag ='_BA'+string(Col_Ang); %part of file name indicating beam angle
GraphTag ='_N'; %part of file name indicating graphic type: ray paths, no ray
paths and heat flux highlights

%define graphics tag
if OpPath
    GraphTag ='_P';
end
if FluxHL
    GraphTag = GraphTag+'_F';
end

DateTag = '_D'+string(datestr(now,'mm_dd_yy')); %part of file name indicating
date
TimeTag = '_T'+string(datestr(now,'HH_MM_SS')); %part of file name indicating
time

FigRootName =RayTag+AbsorbTag+SpecTag+BeamAngTag+GraphTag; %basic name for
all graphics in this test
FigTime = DateTag+TimeTag; %time graphics were generated
Fig1Name = FigRootName+FigTime;
Fig2Name = FigRootName+'_PSF'+FigTime;
savefig(Fig1Name+'.fig');
exportgraphics(ax,Fig1Name+'.jpg','Resolution',500);

%%%%%%%%%%%%%%%%%%%point spread function%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Xp = [1 0 0];
Zp = nf;
Yp = [nf(1) nf(3) -nf(2)];
Xo = [1 0 0];
Yo = [0 1 0];
Zo = [0 0 1];
fpAng = acosd(dot(Zp,Zo));
Tr = ([1 0 0; 0 cosd(fpAng) cosd(fpAng-90); 0 cosd(90+fpAng) cosd(fpAng)];); %Local to Global
OpRayL = double(transpose(Tr*transpose(OpRay-pfM)));
if OpPath
    rotMat = transpose([-1 0 0; 0 -1 0; 0 0 1]); %rotation about z-axis (180 rotation)
    OpRayL = transpose(rotMat*transpose(OpRayL));
end

figure()
newPoint = pfM+nf;
scatter(OpRayL(:,1),OpRayL(:,2),'.','b');
hold on
if nFP>1
    NonOpRayL = double(transpose(Tr*transpose(NonOpRay-pfM)));
    if OpPath
        NonOpRayL = transpose(rotMat*transpose(NonOpRayL));
    end
    scatter(NonOpRayL(:,1),NonOpRayL(:,2),'.','r');
end
hold on
%plot3([pfM(1) newPoint(1)],[pfM(2) newPoint(2)],[pfM(3) newPoint(3)],'r');
%Fix scaling of figure
ax = gca;
ax.DataAspectRatio = [1 1 1];

%label axes
xlabel("x-axis (mm)");
ylabel("y-axis (mm)");
zlabel("z-axis (mm)");

savefig(Fig2Name+'.fig');
exportgraphics(ax,Fig2Name+'.jpg','Resolution',500);

mattimer = toc; %stop timing program

ExcelName = Fig1Name+'.xlsx';
writematrix(D_Factor,ExcelName,'Sheet','D_Factor');
writematrix(OpRayL,ExcelName,'Sheet','Optical Rays');
if nFP>1
writematrix(NonOpRayL,ExcelName,'Sheet','Non-Optical Rays');
end
writematrix(refl,ExcelName,'Sheet','Ray Reflections');

writematrix(mattimer,ExcelName,'Sheet','Timer');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [BMat] = Glob2Loc(AMat)
rotMat = [-1 0 0; 0 -1 0; 0 0 1]; %rotation about z-axis (180 rotation)
Ang = 40.94857093; %Tilt of mirror about x-axis
tMat = transpose([1 0 0; 0 cosd(Ang) cosd(Ang-90); 0 cosd(90+Ang) cosd(Ang);]); %Local to Global
BMat = rotMat*AMat;
BMat = tMat*BMat;
end

%rotate xyz sets from local coordinate system to global coordinate system
function [BMat] = Loc2Glob(AMat)
rotMat = transpose([-1 0 0; 0 -1 0; 0 0 1]); %rotation about z-axis (180 rotation)
Ang = 40.94857093; %Tilt of mirror about x-axis
tMat = transpose([1 0 0; 0 cosd(Ang) cosd(Ang-90); 0 cosd(90+Ang) cosd(Ang);]); %Local to Global
BMat = rotMat*AMat;
BMat = tMat*BMat;
end