Project A: An Optical Ray Tracer

Tik Tsoi

I. Introduction

In this project, our aims were to design a 3-D optical ray tracer in Python with object-oriented programming and to conduct numerical investigation in various optical systems. The performance of the ray tracer was rigorously tested and optimized by setting up simple test cases. Further investigation was carried out to examine the imaging performance of simple lenses.

In our simulation, the propagation of rays obeys basic laws of geometrical optics. The ray propagates in the direction of its wave vector until it incidents with an interface. As a result, optical phenomena take place such as reflection or refraction of rays.

II. Description of Project Work

A. Initial algorithm development

With the object-oriented approach, rays and optical elements were treated as objects in our system. To model a ray, the ray object was represented by a wave vector which specifies its direction of propagation and a list of vertices which documents its path. To create an optical element, a number of parameters were required including refractive indices and the geometrical properties of its shape. OpticalElements serves as the base class for all optical elements, which stores the more fundamental properties for an optical element. The methods for calculating the intercept with the surface and performing refraction vary slightly for each element, so methods are written specifically under each subclass.

SphericalRefraction was the first subclass of OpticalElements, it was used to model a single spherical refracting surface. To locate the intersection point of the ray with the surface, the following equation [1]

$$l = -\vec{r} \cdot \hat{k} \pm \sqrt{\left(\vec{r} \cdot \hat{k}\right)^2 - (|\overrightarrow{r}| - R^2)}$$

where l is the distance from the position of the ray to the intersection point, $\vec{r}$ is the vector from the centre of the sphere to the position of the ray, $\hat{k}$ is the unit vector in the direction of the ray, and R is the radius of the sphere,

was applied to compute the intercepts of the ray with the spherical surface. Since only one intercept was desired, the algorithm was designed to return the minimum value of the two intercepts for a surface with positive curvature and vice versa the maximum value for a surface of negative curvature. Otherwise if no valid intercept was returned the ray would be marked as terminated.

Since Snell's law requires the refracted ray to remain in the same plane as the incident ray and surface normal, a more general expression of the Snell's law (in vector form) would be better suited to deal with wave vectors in higher dimensions:

$$n_r'\vec{r}\,' \times \vec{n} = n_r\vec{r} \times \vec{n}$$

where n_r' and n_r are the refractive indices of the media either side of the surface, $\vec{r}\,'$ and $\vec{r}$ are the incident and refracted ray vectors, and $\vec{n}$ is the normal vector of the surface.

Our objective was to use this equation to calculate the refracted wave vectors and set it as the new wave vector for the ray. The ray continues until it encounters the output plane.

B. Testing

Simple test cases were run to validate the methods in the SphericalRefraction class. To ensure the refraction methods is working, an incident ray travelling in the y-z plane was propagated in the system, since the refracted ray would still on the same plane, we could ignore the third dimension and apply basic Snell's law to the problem. The calculated incident and refracted angles proved that the method was functioning correctly.

Further testing was carried out with a system comprising a single spherical refracting surface and an output plane. A few example rays were traced and their paths were rendered using matplotlib. At this stage, a Simulation class had been devised to contain all the rays on the plane and optical elements in the system. Rays were propagated in sequence to the order of optical elements in the system and according to each of their properties. The position of the paraxial focus was found by propagating a parallel ray close to the optical axis. The spherical refracting surface only works well in the paraxial limit. If non-paraxial rays were added to the system, spherical aberration would occur and lead to blurriness in the image. The consequences of aberrations are studied at a greater depth in the later section.

To explore the geometrical aspects of the model, a spherical mirror was created. Instead of having refraction, when rays are propagated reflection occurs. From Fig 3, all rays are refracted at the same angle, we could deduce that the surface must be non-flat and symmetrical.

C. Raytracing investigations

In task 12 we were asked to generate a uniform collimated beam for a larger diameter, a separate class UniformCollimatedBeam was responsible. To approximate a uniform beam, points are generated in rings at a fixed interval that depends on the density, with each consecutive ring away from the centre doubling the number of points.

III. KEY RESULTS

From Fig 5, the points at the paraxial plane are subjected to spherical aberration. Rays that strike the surface off-centre are refracted to a greater extent than those close to the centre. Theoretically if all the points were perfectly focused, the spot diagram should only contain a single spot. Further investigation was carried out to examine the error by calculating the RMS spot radius. This was found to be 9.88 x 10^{-4} mm which corresponds to an angular separation of 1.32 x 10^{-5} rad from the midpoint of the refracting surface. The non-paraxial rays contributed heavily to the error, the greater their distance from the optical axis, the larger the error. The diffraction limit was also computed and it has a value of 1.53 x 10^{-7} rad. This is a factor of two out which implies that the size of geometrical focus is very insignificant and the image can still be resolved with confidence.

A plano-convex singlet lens was modelled with the convex face facing the input and followed by the convex face facing away the input. The paraxial focus for the first case was shorter, it gave a value of 245 mm whereas for the later case it was 250 mm. Consequently, the geometrical spot size for the first case was 5.50 x 10^{-3} mm and the later case $5.61\,x\,10^{-3}$ mm. This has a huge significance to the orientation of the plano-convex lens to produce the best quality image. When the curved face is facing the input, the sharpest possible focus is achieved. This will produce minimum aberration as supported by the smaller value of the RMS error.

IV. IMPROVEMENTS

The use of parent class could be explored further. Both the SphericalRefraction and Plane class have the intercept and refraction method. For future reference, subclasses that share the same method should be attempted to be written in the parent class to improve efficiency in the coding.

V. CONCLUSION

In this project, our primary aim to design a 3-D optical ray tracer using principles of object-oriented programming was successful. Classes were arranged and located to their corresponding modules in great clarity. Rays were propagated in accordance to the principles of geometrical optics. The methods of each class were tested and validated with confidence.

The imaging performance of simple lenses were investigated by plotting the spot diagram and calculating the geometrical spot radius (same as the RMS error). Smaller RMS error implies less aberration and higher imaging quality.

REFERENCES

[1] **R Kingham (2020*)*** ***Project A: An Optical Ray Tracer*** Imperial College London [Accessed 4 Nov. 2020].

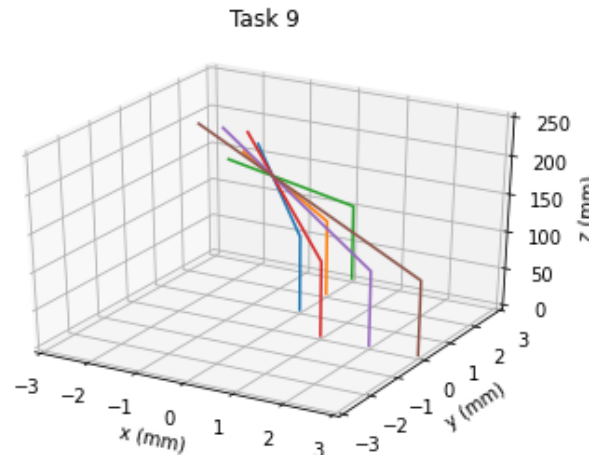

Fig 1 Rays from different points on the input plane

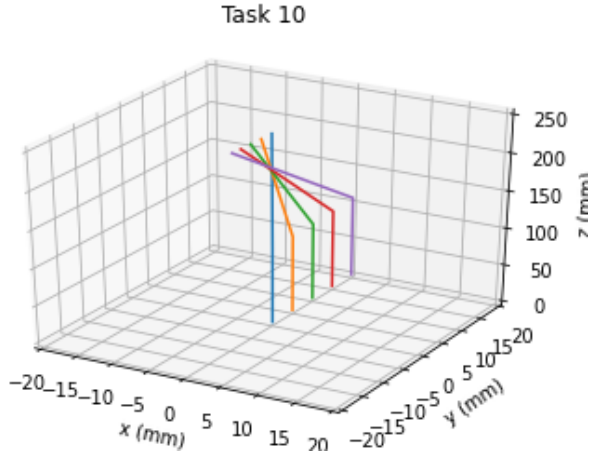

Fig 2 Rays parallel to the optical axis

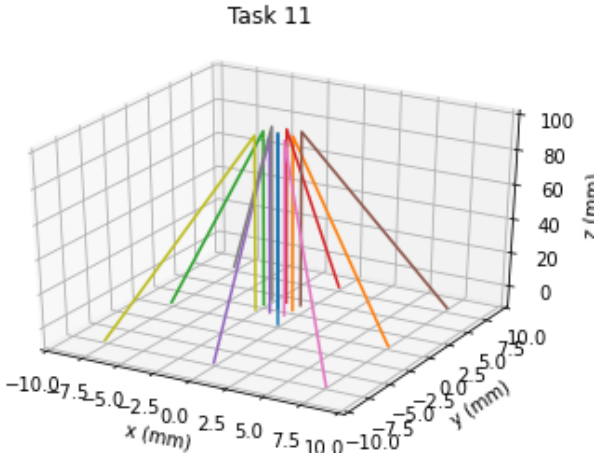

Fig 3 Rays reflecting off the spherical surface

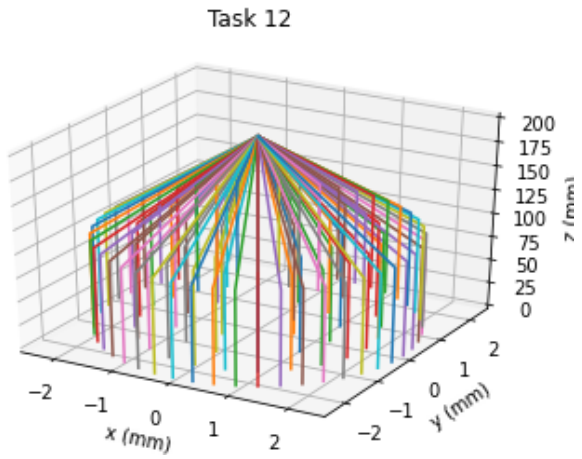

Fig 4 A uniform collimated beam parallel to the optical axis

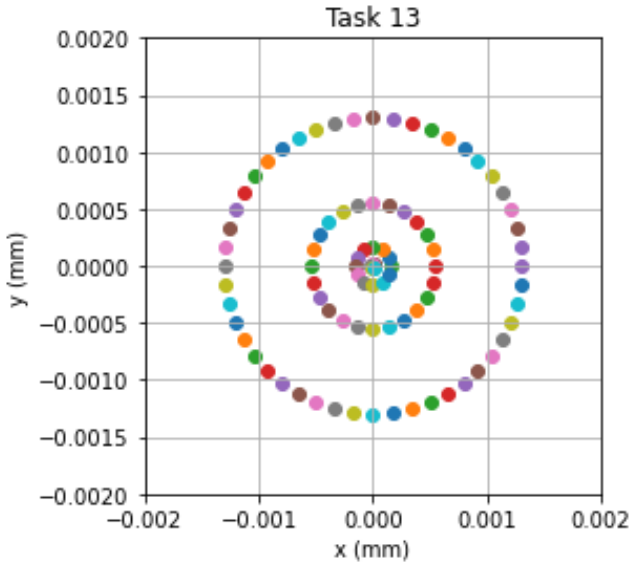

Fig 5 Spot diagram at the paraxial plane

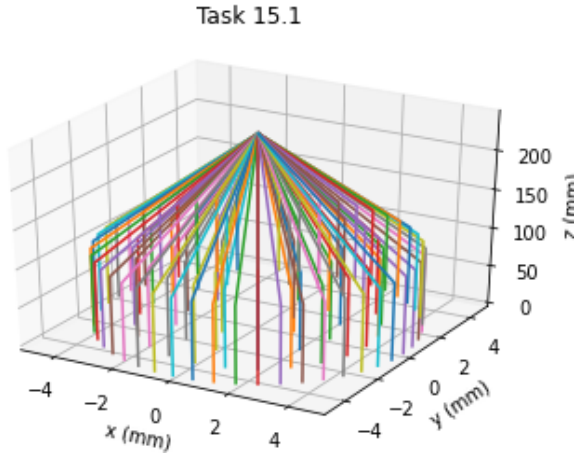

Fig 6 A uniform collimated beam travelling towards the convex face of a plano-convex lens

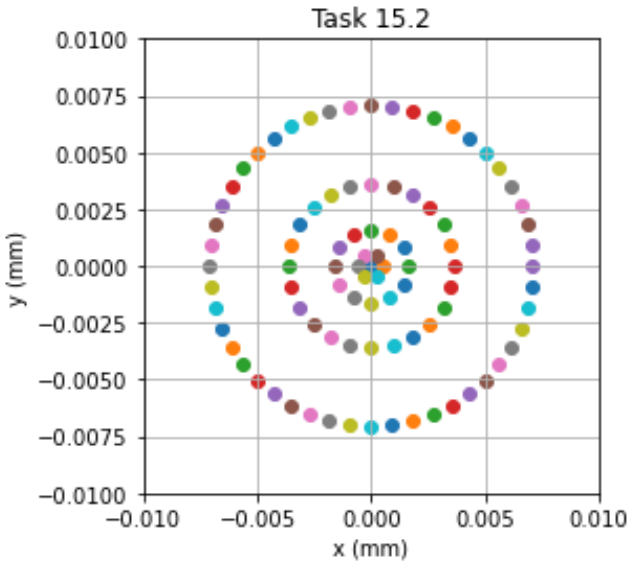

Fig 7 Spot diagram at the paraxial at the paraxial plane for a uniform collimated beam travelling towards the convex face of a plano-convex lens

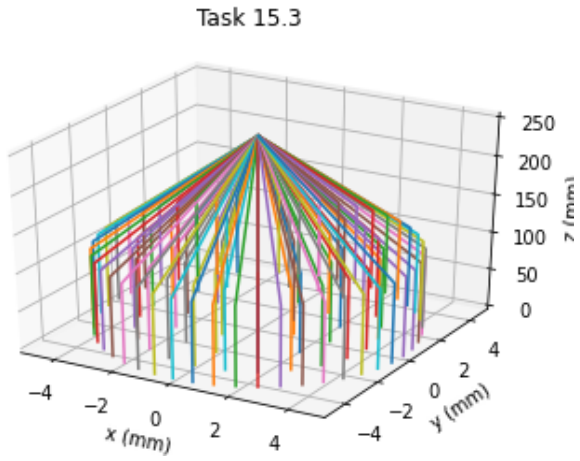

Fig 8 *A uniform collimated beam travelling towards the plane face of a plano-convex lens*

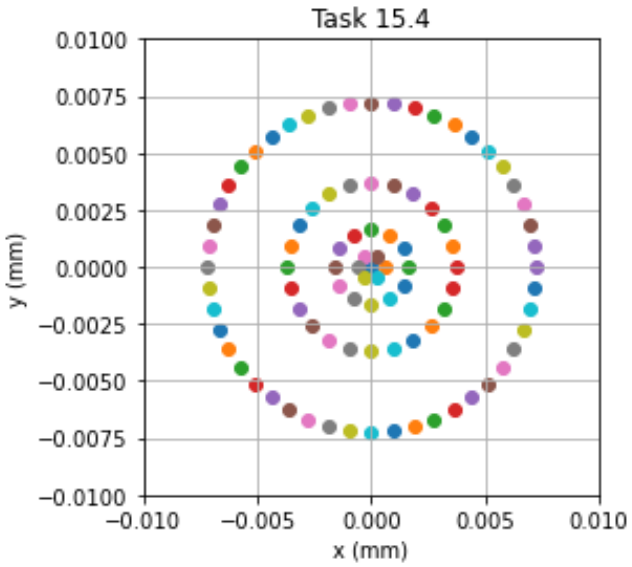

Fig 7 Spot diagram at the paraxial at the paraxial plane for a uniform collimated beam travelling towards the convex face of a plano-convex lens