Explanation of Computations of Expected Scattering Intensities

Link to code: https://github.com/eloisezeng/QLI_Lab_Project/tree/main/sim_img_demo

Goal: Find the expected scattering intensities of a laser beam propagating at θLº with respect to the imaging plane of a stationary camera. The laser beam is always propagating to the right because this is most convenient when using the protractors for our experiment.

More precise goal: Find a function, $I(\theta, R, \theta_L)$, where:

θ (aka "Camera's Angle"): angle between focal point in camera lens and a ray of scattered light

R: the normal distance between the focal point and the laser beam i.e. the distance is the length of the segment that is perpendicular to the imaging plane and connects the focal point and the laser beam

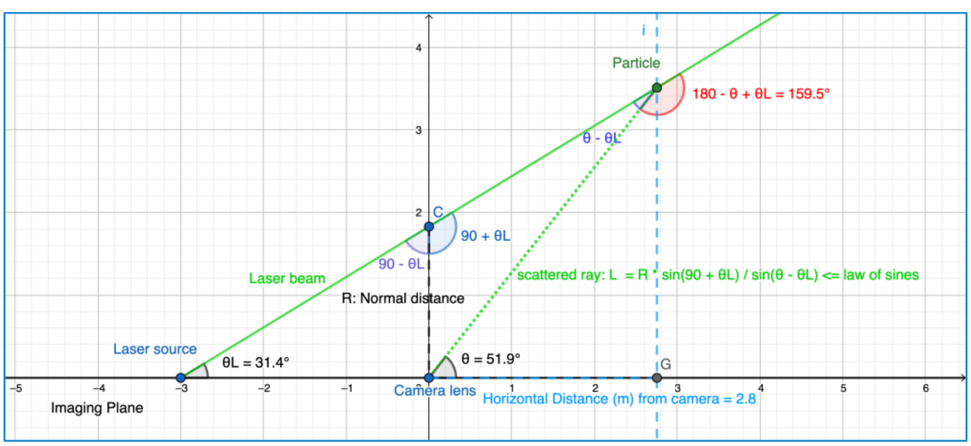
θ_L (aka theta_L): angle of laser beam with respect to the imaging plane

If you graph this function, angle θ is on the x-axis. On the y axis, it is the intensity of the ray of scattered light propagating at angle θ relative to the camera's sensor.

Mieplot provides a function, $I_{Mieplot}$, that is displayed in a graph. The graph shows the intensity of a ray of scattered light propagating from a particle at a **certain angle** and **distance** away from the camera. At 0º, the intensity of the ray is highest because the ray is pointed directly at the camera. At 90º, the intensity of the ray is lowest. At 180º the intensity of the ray is higher than at 90º.

The only varying parameters we'll feed to the $I_{Mieplot}$ function are a particle's **angle (aka "**Angle Theta**")** and **distance (L)** from the camera.

Set-up: θL ∈ [0, 90]



Note that $I(\theta, R, \theta_L) = I_{Mieplot}$("Angle Theta"=180 - θ + θL, L=R * sin(90 + θL) / sin(θ - θL))

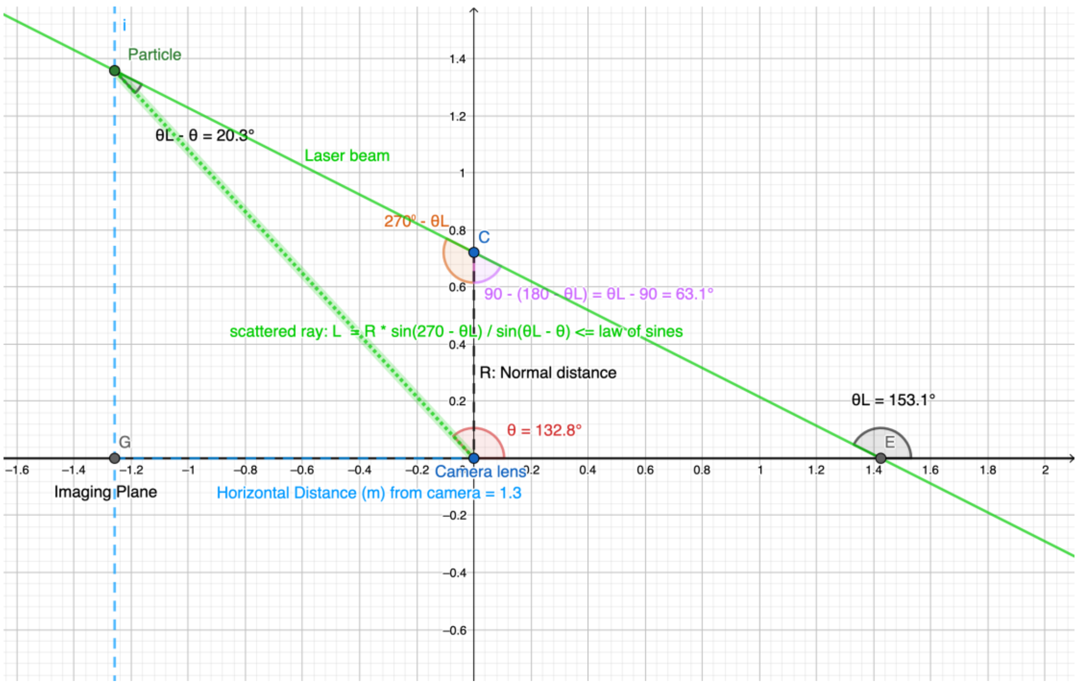
Since we don't want to generate dozens of Mieplot graphs with different L's, we'll compute how a single Mieplot function will change if we varied L.

We can take advantage of the fact that the light intensity is proportional to $1/L^2$ where L is the distance between the light source and the camera which is measuring the light intensity.

Therefore, $I_{Mieplot}$ (“Angle Theta” =180 - θ + θL, L=R * sin(90 + θL) / sin(θ - θL)) = $I_{Mieplot}$ (“Angle Theta” =180 - θ + θL, 1) / L ^ 2 = $I_{Mieplot}$ (“Angle Theta” = 180 - θ + θL, 1) * (sin(θ - θL) / R * sin(90 + θL)) ^ 2

This is true even when θ = 0º.

Set-up: θL ∈ [90, 180]

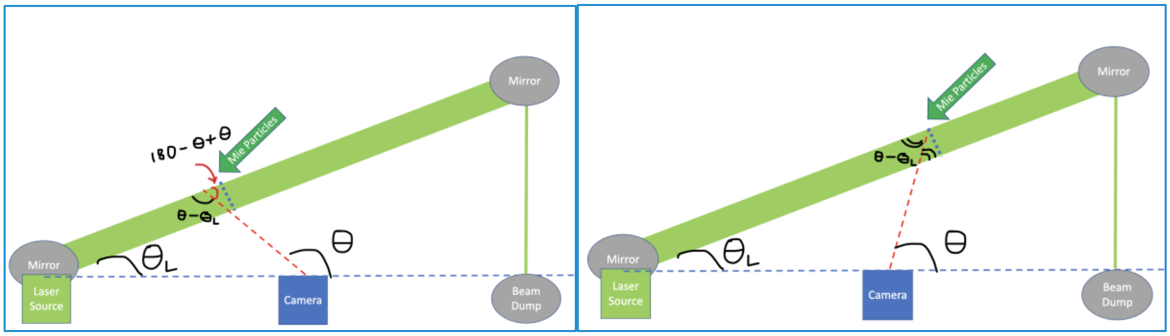


Note that I(θ, R, $θ_L$) = $I_{Mieplot}$ (“Angle Theta” =θL - θ, L=R * sin(270- θL) / sin(θL - θ))

Since we don’t want to generate dozens of Mieplot graphs with different L’s, we’ll compute how a single Mieplot function will change if we varied L.

Here’s how: $I_{Mieplot}$ (“Angle Theta”=θL - θ, L=R * sin(270- θL) / sin(θL - θ)) = $I_{Mieplot}$ (“Angle Theta”=θL - θ, 1) / L ^ 2 = $I_{Mieplot}$ (“Angle Theta”=θL - θ, 1) * (sin(θL - θ) / R * sin(270 - θL)) ^ 2

Next, we’ll account for the diameter (2.1mm) of the laser. There will be more scattering of light when the laser hits more particles. Look at the diagram below. When the camera views the scattered light from angle θ, it views laser light that scattered off of Mie particles along path x. As the camera’s angle θ varies in the diagram below, the length of path x varies, and thus the intensity of scattered light from angle θ.

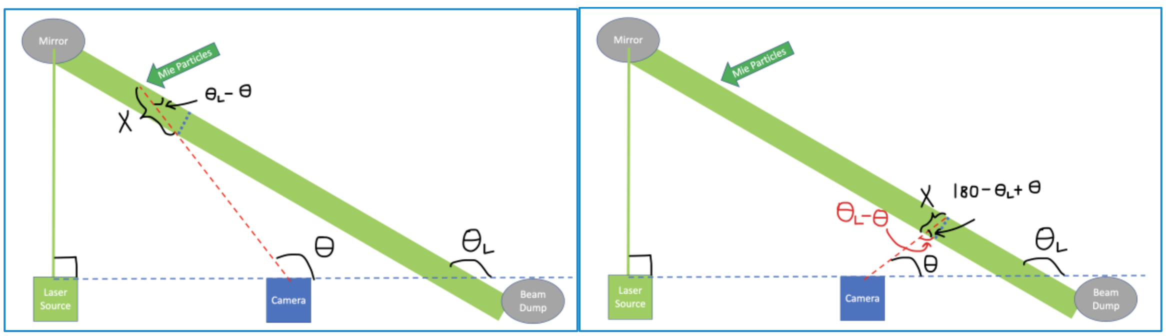


For 0º < θL < 90º:

sin(θ - θL) = sin(180 - (θ - θL)) = diameter of laser / x

x = diameter of laser / sin(θ - θL)

$I_{Mieplot}$ (“Angle Theta” = 180 - θ + θL, 1) * (sin(θ - θL) / R * sin(90 + θL)) ^ 2 * x / (diameter of laser) = $I_{Mieplot}$ (“Angle Theta” = 180 - θ + θL, 1) * (sin(θ - θL) / R * sin(90 + θL)) ^ 2 / sin(θ - θL)



For 90º < θL < 180º:

sin(θL - θ) = sin(180 - (θL - θ)) = diameter of laser / x

x = diameter of laser / sin(θL - θ)

$I_{Mieplot}$ (“Angle Theta”=θL - θ, 1) * (sin(θL - θ) / R * sin(270 - θL)) ^ 2 * x / (diameter of laser) = $I_{Mieplot}$ (“Angle Theta”=θL - θ, 1) * (sin(θL - θ) / R * sin(270 - θL)) ^ 2 / sin(θL - θ)

Fixed Parameters for the Mieplot function

Wavelength: 0.5179 µm

Polarization: Both

Angle range: [0, 180º]

Power of Laser: 15 mW, doesn’t say if laser is polarized or not over here. I assumed the laser is unpolarized: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1489

Size of Mie particle (ranges from 10 µm to 100 µm): I inputted 10 µm

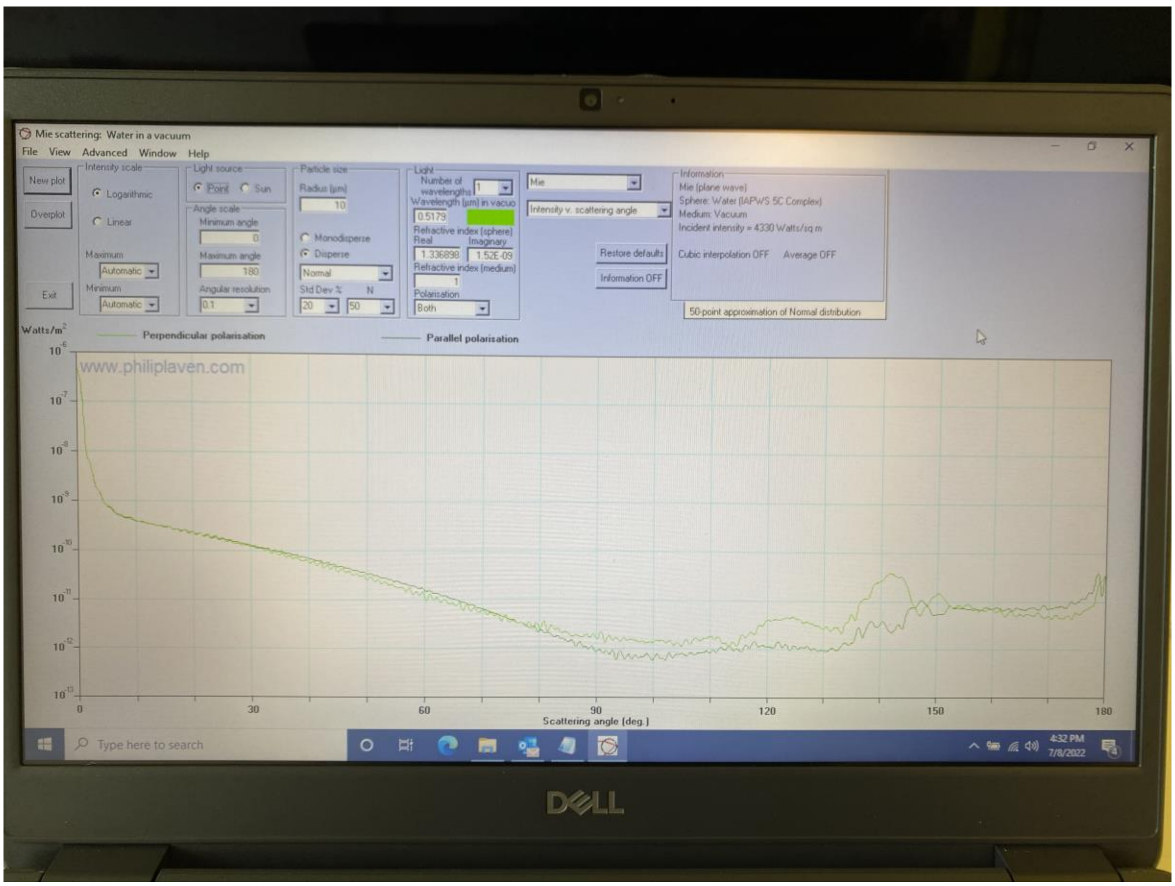
Unpolarized Incident Intensity (Watts/m^2) = 15 mW / ((2.1 mm / 2) ^2 * pi) = 15 * 10^-3 W / ((1.05 m * 10^-3) ^ 2 * pi) = 15 * 10^-3 / (1.05^2 * 10^-6 * pi) = 15 * 10 ^ 3 / (1.05^2 * pi) = 4330.7 = 4.33 * 10^3 W/m^2

(Polarized Incident Intensity = 2.165 * 10 ^ 3 W / m^2) <= unused

Distance of Measurement from scattering sphere (m): 1. I input a distance of 1m from which we can extrapolate data. (Mieplot allows you to input the distance of measurement from the scattering point of light if you click on the drop-down menu “Intensity v. scattering angle”)

Disperse: Normal distribution, Standard deviation of particle size = 20% <= the maximum, the number of different types of particles is 50. These parameters were chosen since the atmosphere has particles of various sizes and it makes the graph look smoother. Before, the graph had lots of bumpy waves and that’s due to diffraction of scattered light that we’d not likely see in our experiment.

Picture of Mieplot screen



Factors that are unaccounted for include:

- Attenuation of laser beam as it propagates
- Diameter of laser

The data from the Mieplot graph is stored in mieplot_disperse_data.xlsx.

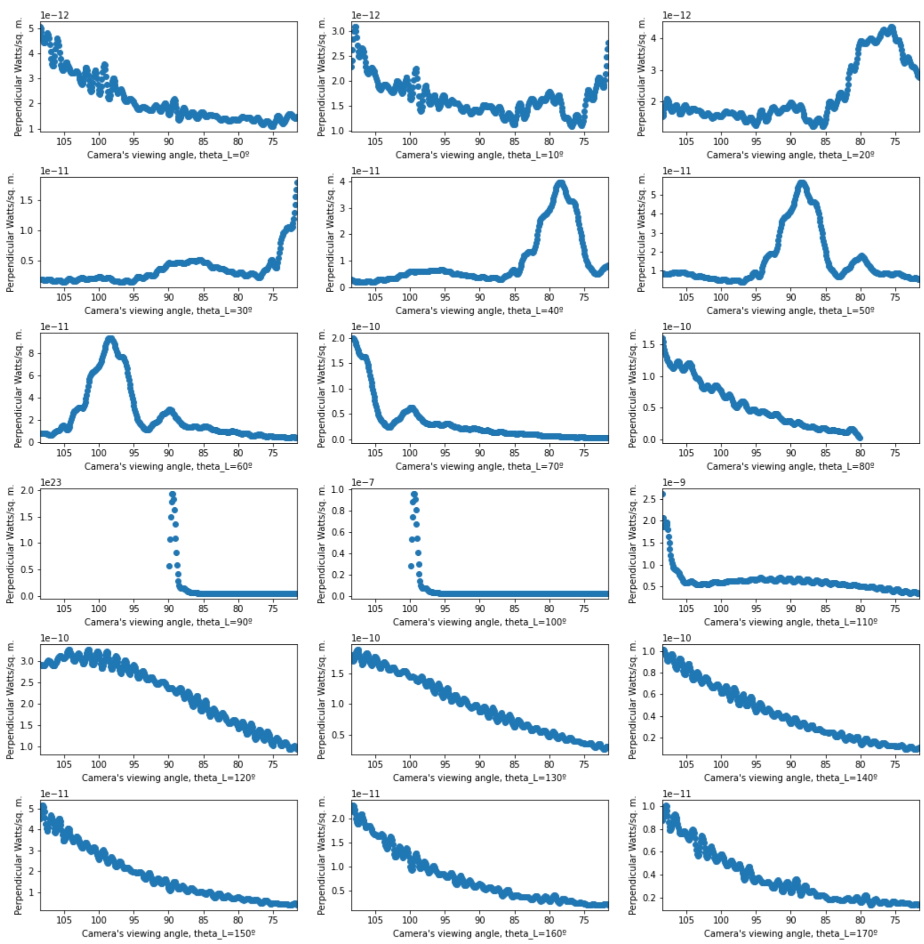
These are the ranges of the parameters.

$\theta \in (90^\circ - 18.5^\circ, 90 + 18.5^\circ) = (71.5^\circ, 108.5^\circ)$

$R \in (0, 1.83\text{m})$

Once $I(\theta, R, \theta_L)$ is programmed, we can see plots like these!

Note that graphs theta_L = 20º and theta_L = 160º aren't exactly the reverse of each other because the laser beam is traveling to the right in both cases. The graphs would be exact opposites if their laser beams were traveling in opposite directions. The y-axis scales differ.

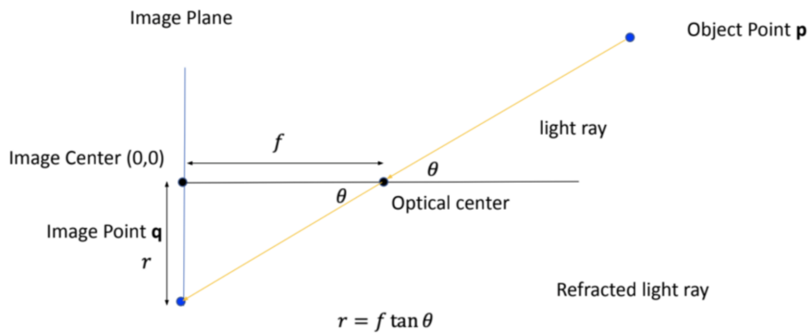


The next goal is to the $I(\theta, R, \theta L)$ func to write $I(h, R, \theta_L)$ where h is the horizontal pixel position of where the scattered light ray hits the camera sensor. For now, (0, 0) coordinates are in the center of the camera sensor.

We can compute r, the horizontal sensor position, as follows.

Imagine the image below showing the top view

Mapping 2D Image to 3D Object (Rectilinear)



If 90 < “Camera’s θ” < 180º, r = f * tan(θ) = 16mm * tan(“Camera’s θ” - 90º)

If 0 < “Camera’s θ” < 90º, r = -f * tan(θ) = -16mm * tan(90 - “Camera’s θ”) = 16mm * tan(“Camera’s θ” - 90º)

Thus, r = 16mm * tan(“Camera’s θ” - 90º)

To find h, we can convert the units of r from mm to pixels.

Quantalux Camera

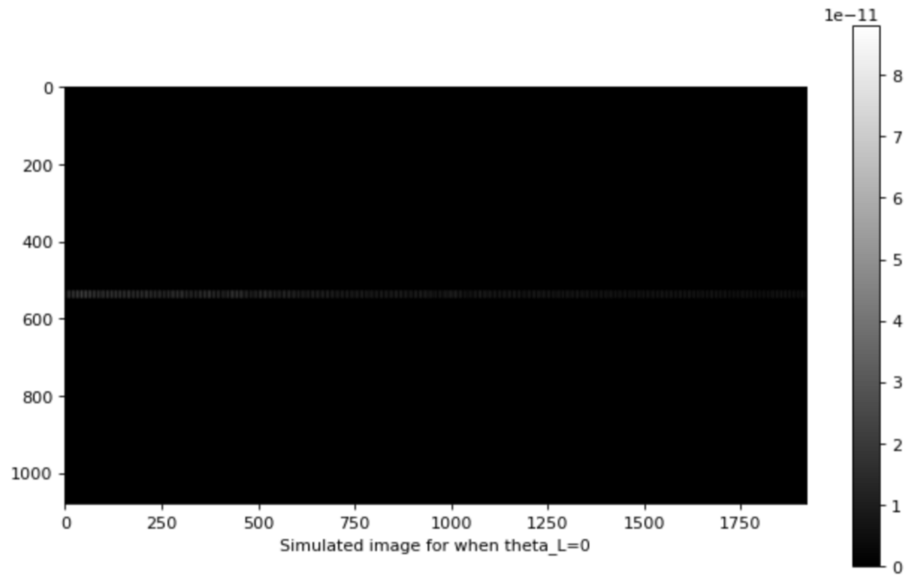
- Sensor width: 9.677mm
- Sensor height: 5.443mm <= unused
- Resolution: 1920 (width) x 1080 (height) Pixels

h = -1920 / sensor_width * r

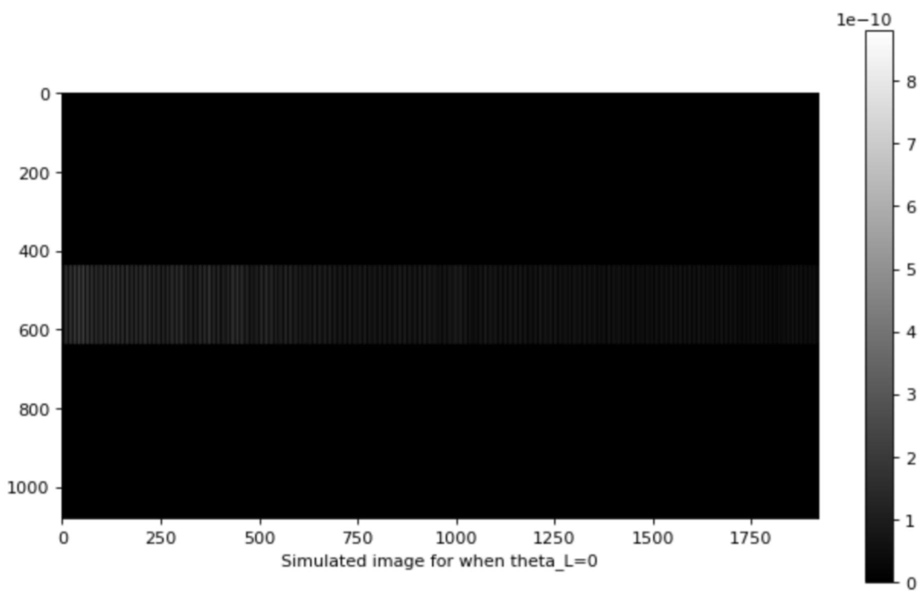
Now we’ll shift the coordinate system such that the top left corner of the image is (0 pixels, 0 pixels).

h = round(h + resolution_h / 2)

Now we can produce graphs like these! By the way, we can change the width of the laser beam by editing the width_of_laser_in_pixels parameter in the show_expected_scattering function.



Simulated image for when theta_L=0



Simulated image for when theta_L=0