LAB

THREE

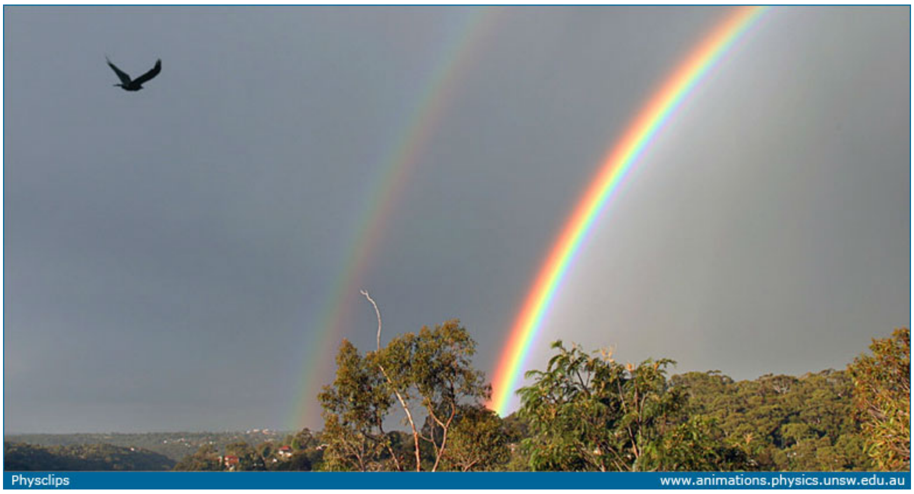
Numerical Modelling

When sunlight is scattered by raindrops, why is it that colorful arcs appear in certain regions of the sky? Answering this subtle question has required all the resources of mathematical physics.

H. M. Nussenzveig, Scientific American, 1977

3.1 Introduction

Rainbows are a well known atmospheric phenomena caused by the dispersion of sunlight by water droplets. Different wavelengths of light are refracted at different angles resulting in distinct bands of colour. The formation of a rainbow can be understood using concepts from geometric optics.



In this lab we will use the MATLAB skills we've learnt so far to construct a *numerical model* of a rainbow that demonstrates why a rainbow appears visually as it does. The exercise is split into several sections. In each section you will write and test code a piece of code which relies on work from the previous section. At the end you will have a complete working model. Writing and testing the code in pieces makes it easier to find bugs and mimics the development cycle for large numerical codes used in research.

Before you start coding it is critical you understand what you're trying to code! One way of doing this is to try and explain the solution in plain English to your partner. Another way is to write comments in your code outlining the steps, and then write code for each part.

3.2 Lab objectives

By the end of this lab sessions you should be able to:

1. Solve physics problems using MATLAB.
2. Develop a MATLAB script built from multiple components.
3. Visualize simple datasets using `plot()`.

If you still don't know how to do any of these things once you have completed the lab, please ask your tutor for help. You should also use these lists of objectives when you are revising for the end of semester exam.

3.3 Activity: Creating a rainbow — Part I

3.3.1 Refraction in a raindrop

Refraction is the change in direction of a ray of light when it traverses the boundary of two media with different refractive indices. For light travelling from a medium with refractive index n_1 to a medium with refractive index n_2, the relationship between the angle of incidence θ_1 and angle of refraction θ_2 is given by Snell's law:

$$n_1 \sin\theta_1 = n_2 \sin\theta_2. \tag{3.1}$$

A dispersive medium is one where the refractive index is a function of wavelength. Water is dispersive and it is this property that gives rise to rainbows, as different wavelengths of light are refracted at different angles resulting in the angular separation of different colours.

In order to construct a rainbow it is necessary to consider the wavelength dependence of the refractive index of water. It is difficult to derive a formula for $n(\lambda)$ for water from first principles. Instead, it is easier to use an approximate model constructed from empirical data. A good approximation in the visible range (400 – 700 nm) is

$$n(\lambda) = c_0 + \frac{c_1}{\lambda} + \frac{c_2}{\lambda^2} + \frac{c_3}{\lambda^3}, \tag{3.2}$$

where $c_0 = 1.3128$, $c_1 = 15.7622$ nm, $c_2 = -4382$ nm^2, $c_3 = 1.1455 \times 10^6$ nm^3, and λ is the wavelength of light in *nanometres*.

Question 1
Write a script that calculates $n(\lambda)$ over the range $425 - 650$ nm using 256 points. Produce a plot of $n(\lambda)$ vs λ and sketch it in the box below. You can attack this problem in three parts:

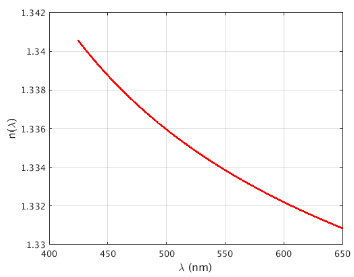
1. Write code to generate a vector of wavelength values over the specified range;
2. Write code to calculate the value of $n(\lambda)$ using Equation 3.2;
3. Plot $n(\lambda)$ vs λ with appropriate labels.

Each of these parts requires a few lines of MATLAB, and you can test each part as you write it.

Hint: The `linspace` function allows you to construct vectors with a fixed number of points over a given range. For example, the code below generates a vector of 256 uniformly spaced points spanning the interval [1, 10].

```
>> xmin = 1.;
>> xmax = 10.;
>> npoints = 256;
>> x = linspace(xmin, xmax, npoints);
```

Note: When you are writing 1.1455×10^6 in MATLAB, you should write `1.1455e6` NOT `1.1455*10^6`. The latter works but it is clumsy.



3.3.2 Path of rays in a raindrop

The next stage of our model is calculating the path of a single ray of light as it passes through a raindrop.

The left panel of Figure 3.1 shows the path of a single ray of monochromatic light inside a raindrop. The ray is parallel to the dotted line when it enters the drop but after undergoing two refractions and a reflection it exits at an angle of deflection γ. The angle γ depends on the angle of incident of the ray ϕ, and the wavelength of light λ.

The right panel of Figure 3.1 shows how the path varies depending on wavelength. A single ray of light is a superposition of different wavelengths and all are initially parallel to the x axis of the figure. It can be seen that upon exiting the raindrop the different colours are travelling along different paths, i.e. γ differs for each.

Using this information we can calculate $\gamma(\phi, \lambda)$ for a range of different wavelengths and entry points.

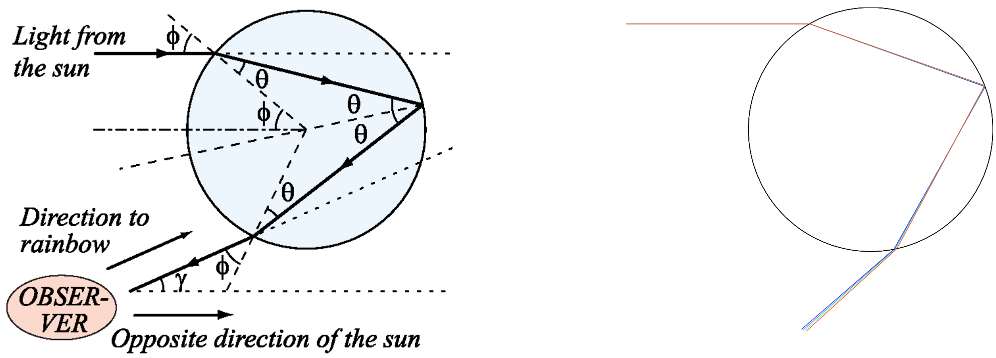


Figure 3.1: **Left:** The path of a ray through a drop of water. The angles shown are the angle of incidence ϕ, the angle of refraction θ and the total angle of deflection γ. **Right:** This figure shows how the path changes with wavelength of light (represented by different colours).

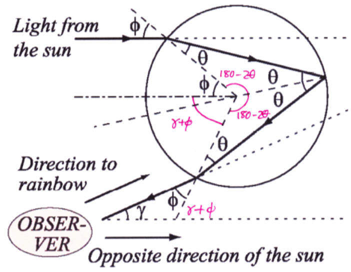
Question 2
Using Snell's law derive an expression for the angle of refraction θ in terms of ϕ and $n(\lambda)$. You can assume that the refractive index of air is 1.0.

$n_1 = 1$ **(air)**
$n_2 = n(\lambda)$ **(water, wavelength dependent)**
$\theta_1 = \phi$ **(angle of incidence)**
$\theta_2 = \theta$ **(angle of refraction)**
Now substitute and rearrange

$$\theta = \sin^{-1}\left(\frac{n_1 \sin\phi}{n_2}\right)$$

Question 3
Using Figure 3.1 as a guide, derive an expression for the angle of deflection, γ, in terms of θ and ϕ.

Consider the angle the light deflects (clockwise) at each point:
1st turning angle of the light ray: $\phi - \theta$
2nd turning angle of the light ray: $\pi - 2\theta$
3rd turning angle of the light ray: $\phi - \theta$
Alternatively, add angles at centre:
$\phi + (\phi + \gamma) + (180 - 2\theta) + (180 - 2\theta) = 360$
$(\phi - \theta) + (\pi - 2\theta) + (\phi - \theta) = \pi - \gamma$
$\gamma(\phi, \lambda) = 4\theta - 2\phi$



Question 4
Write a MATLAB script that computes the angle of deflection, γ, for a single ray given a wavelength λ and an initial angle of incidence ϕ. The computation requires three steps:

1. Compute $n(\lambda)$ from Equation (3.2)
2. Compute θ using your formula from Question 2
3. Compute γ using your formula from Question 3

Use your program to calculate γ assuming $\lambda = 650\,\text{nm}$, $\phi = 30^\circ$ and write the answer in the box. Remember to use the trigonometric functions that take input in degrees not radians (i.e. `sind` rather than `sin`).

n = 1.3308 **theta =** 22.07°
gamma = 28.27°
Tutor note: check for degrees/radians confusion (if they use the radians version of the trig functions the answer will be -63.34).

Question 5
The next step is to calculate γ for multiple monochromatic rays that are initially parallel and uniformly spaced vertically (as in Figure 3.2). If y is the perpendicular distance between the starting point of the ray and the centre of the drop, we can represent this as a vector of points $y = \{y_1, y_2, \ldots, y_N\}$ in the range $[0, R_{\text{drop}}]$ which are the starting points of our rays.

The radius of the drop R_{drop} doesn't matter for the purpose of calculating angles, for convenience we can use $R_{\text{drop}} = 1$. Each ray strikes the drop with a different angle of incidence ϕ. From Figure 3.1 you can see that the angle of incidence is related to the distance y by $y = R_{\text{drop}} \sin\phi$.

Make a copy of your script from Question 4 and extend it to compute the angles of incidence (ϕ) and deflection (γ) for multiple monochromatic rays. To do this you will need to:

1. Construct a vector `y`, of 256 points in the range [0, 1] representing the starting positions of the rays (you might want to use `linspace`);
2. Use this vector to compute the angle of incidence ϕ for each ray;
3. Calculate the corresponding angle of deflection γ for each point.

Plot γ versus y/R_{drop} for $\lambda = 650\,\text{nm}$. Sketch your plot below.

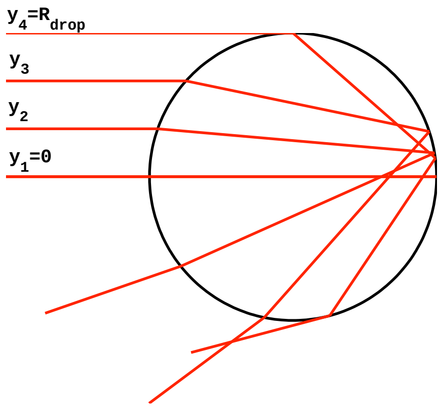
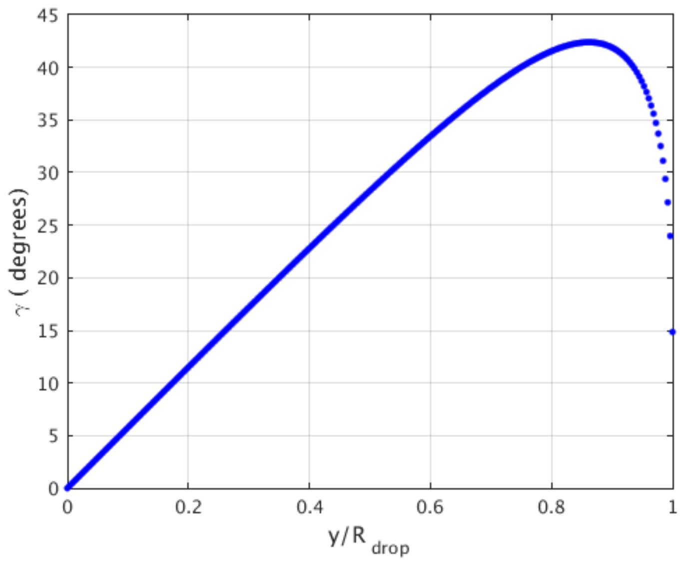


Figure 3.2: Multiple monochromatic rays entering the drop from vertically spaced starting points. In this example there are four rays. The first ray has an elevation of zero ($y_1 = 0$), and the final ray has an elevation equal to the radius of the drop ($y_4 = R_{\text{drop}}$).



3.3.3 Creating a function

In the second half of this lab we want to calculate these angles for light at a range of different wavelengths. Rather than modify the code each time, if we save it as a function we can reuse it by passing in different parameters.

Your function needs to take in the wavelength of incident light, λ, and the radius of the raindrop, R_{drop} and also a vector of values for the initial positions, y. It should return a vector of values for the angles of deflection, γ.

If we translate these requirements to MATLAB, we end up with a function definition something like this:

```
function gamma = calculate_deflection(y, lambda, rdrop)
  % Your function code here
end
```

The last line of your function should calculate `gamma` and hence return the vector to the main program. Don't include your plotting inside the function as that will change depending on what problem we are solving.

You can test your function by passing the same input values as we used in Question 5:

```
>> clear all;
>> lambda = 650.0;
>> rdrop = 1.0;
>> y = linspace(0, rdrop, 256);
>> gamma = calculate_deflection(y, lambda, rdrop);
>> plot(y/rdrop, gamma, 'b.');
```

It's a good idea to use `clear all` before you test your function, so you can ensure that it is working independently, and not accidentally using variables that are already defined in your workspace.

Make sure you have this function working before you continue with the lab.

Checkpoint 1:

3.4 Activity: Creating a rainbow — Part II

Now we have code for computing γ for a single wavelength, we can to extend the model to polychromatic light. To do this we need to loop over the range of wavelengths we're interested in.

Question 6

Write a program to plot γ versus $y/\mathrm{R_{drop}}$ for each of 32 wavelengths in the range 425 – 650 nm. To do this, you should:

1. Create a vector of 32 wavelength values between 450 and 650 nm;
2. Loop through the vector and call your `calculate_deflection` function;
3. Plot γ versus $y/\mathrm{R_{drop}}$ for each wavelength, on the same plot.

Use `hold on;` to force the plots to appear on the same figure.

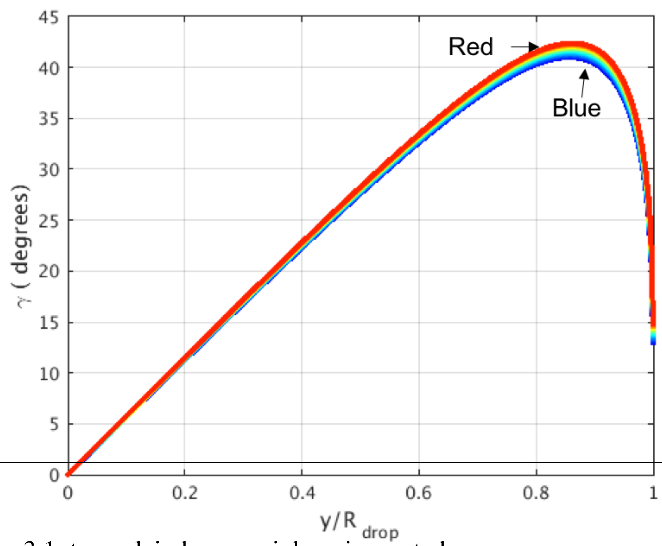
To make the plot more informative, we have provided you with a MATLAB function `rainbow_ct.m` (download this from the unit website) that converts a wavelength in nanometres to an RGB colour value. This way the colours of the curves will reflect the actual colour of that wavelength of light. The code snippet below demonstrates the use of the function.

```
>> lambda = 650;
>> rgb = rainbow_ct(lambda);
>> plot(y, gamma, 'Color', rgb);
```

Note that the input must always be a scalar, passing a vector will results in an error.

Sketch your 'rainbow' below.

Make sure students to label their sketches red and blue



Question 7

Use your plot, and Figure 3.1, to explain how a rainbow is created.

There are two effects evident in Figure 1.2. Firstly, rays with a large angle of incidence tend to experience the largest deflection. This can be inferred from γ vs y/R_{drop} plot which obtains a maximum around $\phi \approx 60°$ (correponding to $\gamma \approx 43°$). Secondly, for a given ϕ, red rays are deflected more than blue. This can be seen in the γ vs y/R_{drop} plot as the red curve sits above the blue.

3.5 The intensity of scattered light (PHYS2911 and PHYS2921)

So far we have determined that different wavelengths of light initially travelling in same direction will be scattered at different angles after transversing the raindrop. To complete the model of the rainbow we need to calculate the intensity of the light exiting the drop $I(\gamma)$, as a function of γ.

The intensity of light is determined by the number of rays which exit for a particular γ value. We can construct a histogram of γ values (which provides the number of rays in an interval $\Delta\gamma$). MATLAB has a built in function **`hist`** which computes the data required to plot a histogram. The code snippet below illustrates the use of this function.

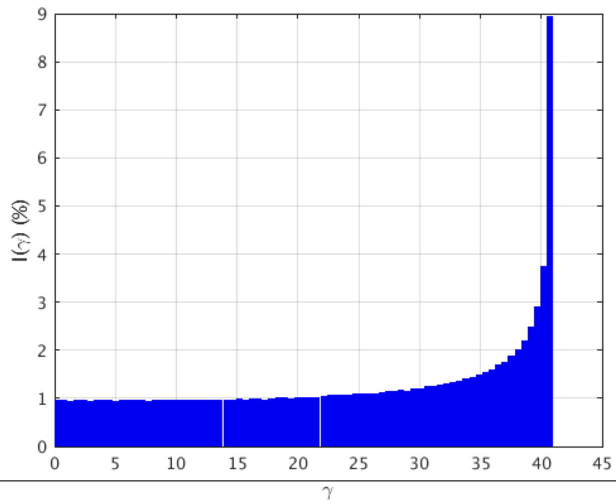
```
% Create a vector of N random numbers
N = 1e3;
y = rand(N, 1);

% Create a histogram using the hist function
% hist(data,number of bins)
[counts bin_loc] = hist(y,sqrt(N));

% Plot histogram using bars
bar(bin_loc,counts);
```

The first argument of **`hist`** is the data; the second is the number of bins. It is convenient to set the number of bins to the square root of the number of points. The function **`hist`** returns two vectors. The first (called **`counts`** in the example) is the number of counts in each bin, the second (called **`bin_loc`** in the example) is the location of the start of each bin. The function **`bar`** plots the histogram as a bar graph.

Question 8 (PHYS2911 and PHYS2921)
With $\lambda = 450$ nm plot a histogram showing how γ is distributed for 6000 rays uniformly spaced vertically. Sketch your plot below.



Question 9 (PHYS2911 and PHYS2921)
Compute $I(\gamma)$ due to 32 different wavelengths. To do this put your code from Question 8 inside a `for` loop which loops over the different values of λ. Use 10^5 rays to ensure a smooth histogram.

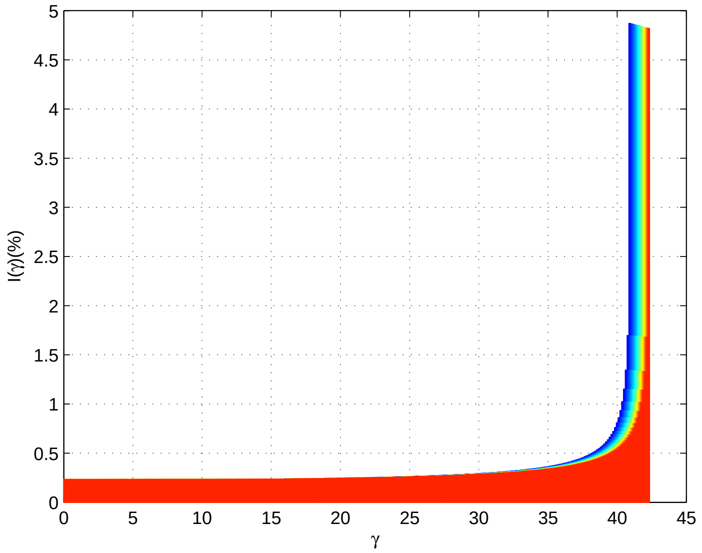
To colour code your plot you need to set the `'FaceColor'` and `'EdgeColor'` options when calling `bar` as demonstrated below.

```
bar(bins,I,'FaceColor',rgb,'EdgeColor',rgb);
```

Where `rgb` is generated using `rainbow_ct.m`.

Sketch your plot below.

If the colours do not look right, try zooming in



Question 10 (PHYS2911 and PHYS2921)
Describe how the intensity of different wavelengths varies with angle. Measure the angular size of the bow.

Each wavelength obtains an intensity maximum at a different value of γ. This results in the formation of the distinct bands of colour characteristic of rainbows. The bow is $\approx 1.8°$ in size.

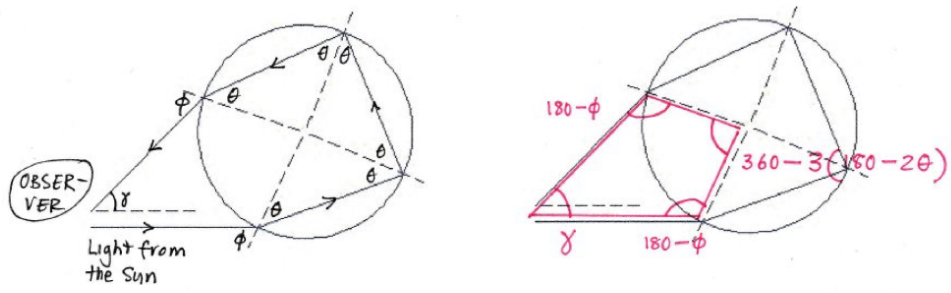
Checkpoint 2:

3.6 Challenge: Double rainbows

In Figure 3.1 the ray undergoes a single reflection inside the raindrop. The rainbow formed as a results is called a first-order rainbow. A second-order rainbow results when the ray undergoes a second reflection before exiting the drop. A double rainbow occurs when both the first-order and second-order rainbows are visible.

In Figure 3.1 the ray undergoes a single reflection inside the raindrop. The rainbow formed as a results is called a first-order rainbow. A second-order rainbow results when the ray undergoes a second reflection before exiting the drop. A double rainbow occurs when both the first-order and second-order rainbows are visible.

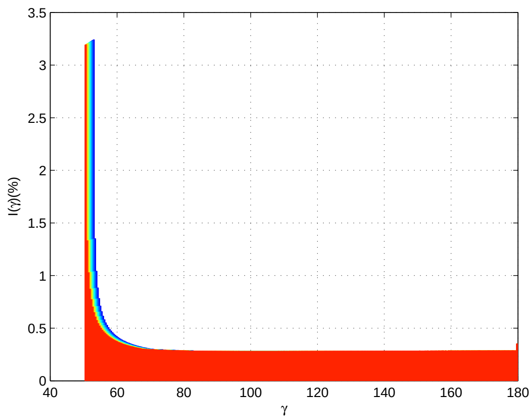
Question 11
Sketch the light path for a second-order rainbow. (Feel free to use Google if you need help. . .)



Question 12
Compute γ in terms of θ and ϕ for the second-order rainbow.

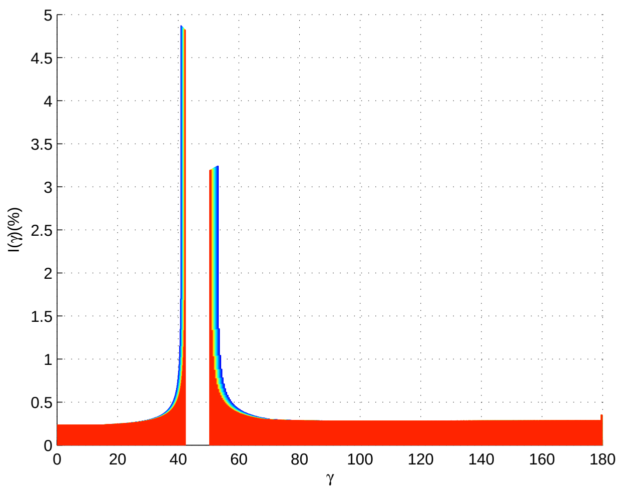
$\gamma + 2(180 - \phi) + 360 - 3(180 - 2\theta) = 360$
$\gamma = 180 + 2\phi - 6\theta$
Please, refer to figures in the Q11 answer box

Question 13
Adapt your code from Question 9 to compute the intensity distribution for the second-order rainbow. Draw your plot in the box. What do you notice about the order of the colours? *Hint:* Construct y in the range $[-R_{\text{drop}}, 0]$.



Note that the colours are reversed compared to the first order rainbow.

Question 14
Plot the intensity of the first and second order rainbows on the same plot. Compare the widths of the two bows.



Second order rainbow is thicker that the first order, but is fainter in general. The intensities of the rainbows are not realistic. We have ignored the losses due to transmission/reflection at each refraction/reflection. These effects require use of the Fresnel equations at each interface.

3.7 Extra information and references

The activities in this lab are based on Amundsen et al. 2009, *American Journal of Physics*, **77**, 795.
`http://scitation.aip.org/content/aapt/journal/ajp/77/9/10.1119/1.3152991`
The schematic in Figure 3.1 is also taken from this reference.

A great introduction to the science of rainbows is given in the Scientific American article:
`http://www.phys.uwosh.edu/rioux/genphysii/pdf/rainbows.pdf`

For a full quantitative treatment of rainbows, you might be interested in this paper:
`http://www.cems.uvm.edu/~tlakoba/AppliedUGMath/rainbow_glory_review.pdf`

Equation 3.2 is taken from Quan and Fry 1995 *Applied Optics*, **34**, 3477.
`http://www.opticsinfobase.org/ao/fulltext.cfm?uri=ao-34-18-3477&id=45728`

Note that some of these articles may only be available through the University library subscription (in other words you will have to download them on campus using your University account).