Measuring and modelling the growth of silver nanorods using time-resolved UV/visible extinction spectroscopy

A thesis presented in partial fulfilment of the
requirements for the degree of

Master of Science
in
Chemistry

at Massey University, Manawatū,
New Zealand.

Robert McEwen
2026

Abstract

Metal nanoparticles are known to have plasmon resonances that are dependant on the size and shape of the particle. As these resonances are easy to observe in situ during the growth of these particles, UV-Visible extinction spectroscopy provides an excellent way to monitor the growth and aid with the understanding of the underlying growth mechanisms.

Silver nanorods present an ideal candidate for this, as the longitudinal plasmon peak is known to have a linear relationship with the aspect ratio of the nanorods. This presents an interesting challenge, as any associated growth models would need to explain the growth in term of the aspect ratio rather than the absolute size of the nanorod.

The discrete-dipole approximation (DDA) was used to determine the relationship between the plasmon resonance position and the aspect ratio, which was found to be $\lambda_{ext} = (113 \pm 10) \times A\ + (277 \pm 20)$. To achieve this, a correction for the error associated with the DDA calculations was required for the capsule geometry that was used for the nanorods. This correction was found to be the same correction the has been reported for spherical particles.

A growth model was developed that explains the growth of nanorods in term of the evolution of the aspect ratio. Nanorods were prepared using various reaction condition and the plasmon peaks were observed during the growth.

Transmission electron microscopy (TEM) images however, revealed the presence of trigonal platelets and other geometries in the nanorod preparations. Efforts were made to remove these other geometries from the nanorod solution so that the growth model could be applied to the experimental results

Acknowledgements

I am extremely grateful to my supervisor, Professor Mark Waterland, for providing guidance and feedback throughout this project. Your enthusiasm and encouragement was extremely helpful whenever I had any doubts about my research and I would never have been able to complete this without your help.

I would like to thank every one that I have shared a lab or office with during my time at Massey University — especially the two Sams. You have certainly made this time more enjoyable and you have provided many valuable discussions along the way.

Finally, I would like to thank my friends and family for all of their support over the years.

Contents

List of Figures

List of Tables

Acronyms

CCG	Complex-conjugate gradient method
CTAB	Cetyltrimethylammonium bromide
DDA	Discrete-dipole approximation
FCC	Face-centred cubic
FFT	Fast Fourier transform
LCTEM	Liquid cell transmission electron microscopy
ODE	Ordinary differential equation
PVP	Polyvinylpyrrolidone
SERRS	Surface-enhanced resonance Raman scattering
SERS	Surface-enhanced Raman scattering
TDDFT	Time-dependant density functional theory
TEM	Transmission electron microscopy

List of Symbols

A	Aspect ratio of the rod-like nanoparticle geometries
A'	Aspect ratio of the prolate spheroids used in the Gan's theory calculations
χ	Convergence parameter used for the correction of the DDAcalculations
d	Inter-dipole spacing for the DDA calculations
ε	Complex dielectric constant of the nanoparticle
ε_1	Real component of the dielectric constant of the nanoparticle
ε_2	Imaginary component of the dielectric constant of the nanoparticle
ε_m	Dielectric constant of the medium surrounding the nanoparticle
m_{Ag}	Complex refractive index of silver
n_{Ag}	Real component of the refractive index of silver
k_{Ag}	Imaginary component of the refractive index of silver
n_m	Refractive index of the medium surrounding the nanoparticle in the computational calculations
γ	Parameter characterising the various nanorod preparations by the ratio of silver atoms in the growth solution to that in the seed solution
N	The total number of dipoles in a DDA target geometry
N_S	The total number of surface dipoles in a DDA target geometry

η	Parameter of a pseudo-Voigt profile which describes the weight between the Lorentzian and Gaussian functions. A Value of 0 means the pseudo-Voigt is purely Lorentzian while a value of 1 mean it's purely Gaussian
a_{eff}	Effective radius of the target geometries used in the dda calculations which is the radius of a sphere with the same volume as the target

Chapter 1

Introduction

Plasmon resonances of noble metal nanoparticles have been unknowingly used for thousands of years, with examples of coloured Roman glass dating back to the late bronze age ($\sim$1000 BC) [1], with the most famous example being the Lycurgus Cup [2].

Despite these early applications, it wasn't until 1857 when Michael Faraday discovered that this colour comes from the presence of metal nanoparticles [3][4]. He was studying solutions of gold nanoparticles and he correctly identified that the colour of these solutions was dependent on the size of the nanoparticles contain within.

In 1908, Gustav Mie developed a theory based on Maxwell's equations to explain the colour of colloidal gold solutions [5]. This theory, now known as Mie Theory or Lorenz-Mie theory, showed that the colour of the solution was dependant on particle size and the optical properties of the metal as well as that of the medium they were contained in [6].

These optical properties are now known to be due to the collective oscillations of the electrons at the surface of the nanoparticles which are excited by electromagnetic radiation. This is known as surface plasmon resonance. The surface plasmon resonances of silver and gold are of special interest as their plasmon resonances are within the visible spectrum.

One application of the plasmon resonances of noble metal nanoparticles is in surface-enhanced Raman scattering (SERS). The enhanced electromagnetic field near the surface of the nanoparticles due to the surface plasmon resonance, can be rescattered by any adsorbed molecules. As the intensity of the inelastic Raman scattering is proportional to the fourth power of the electromagnetic field strength, the surface plasmon resonance also enhances the Raman scattering of the adsorbed molecule [7]. Enhancement factors of up to 10^{11} can be achieved which can lead to detected Raman scattering from single molecules [8][9].

It has been shown that this surface plasmon resonance can be altered by changing the size and shape of the nanoparticles [10][11][12]. The preparation of many shapes have been reported for gold and silver nanoparticles, such as: rods [13][14], trigonal platelets [15] and cubes [16].

SERS also presents an excellent application for the tunable plasmon resonances. If the plasmon resonance of the nanoparticle along with the laser frequency are tuned to match the electronic resonance of the adsorbed molecule, then further enhancement of the Raman scattering is observed [17]. This is known as surface-enhanced resonance Raman scattering (SERRS).

Another application which exploits this tunability of the plasmon resonances is found in gold nanorods which are used in the photothermal cancer therapy [18][19]. The size and shape of the nanorods are used to tune the plasmon resonance to the transparent window of the skin (800–1,200 nm) as well. Then either passive or target delivery of the nanorod is used to deliver the nanorods to the tumor site where excitation of the plasmon resonance of the rods, by near-infrared laser light, causes heating of the nanorod which then induces cell death of the surrounding tumor.

To ensure total control of the position of the plasmon resonances, methodologies must exist which can reproducibly create nanoparticles with a specific size and shape.

One method for producing anisotropic silver nanoparticles with various shapes is known as the polyol method. In this method, silver nitrate is reduced by a polyol in the presence of polyvinylpyrrolidone (PVP); with the PVP acting as a capping agent to increase the stability of the resulting nanoparticles. A number of polyols can be used, such as: ethylene glycol, propylene glycol or butylene glycol to name a few [20].

The silver nitrate is reduced into silver atoms which agglomerate to form crystal clusters. In the initial stages, the structure of these clusters can fluctuate and defects can be introduced into the crystal structure. Twin defects result in thermodynamically favoured crystals as these crystals are bound by the close-packed $\{111\}$ facets, which has the lowest energy [21][22].

Once these clusters reach a certain size, the thermal energy is no longer sufficient to introduce further crystal defects and the crystal structure of the resulting nanoparticle is fixed.

The geometry of single crystals can still, however, be influenced by the use of various capping molecules. These capping molecules preferentially bind to specific crystal facets and stabilise any geometries which are bound by these facets. Figure 1.1 shows a few of these possible geometries for a single face-centred cubic (FCC) crystal. PVP has shown to preferentially bind to the $\{100\}$ facet of the nanoparticle crystal [21] while trisodium citrate, with its 3-fold symmetry preferentially binds to the close-packed $\{111\}$ facets [23].

Another method which is used for the production of twinned silver [24] and

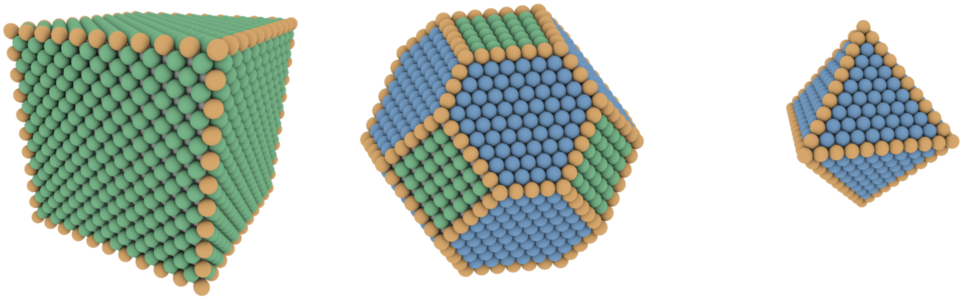


Figure 1.1: Single crystal of silver showing the various geometries that can exist depending of which facets are stabilised by capping molecules. The {111} facet is show in blue and the {100} in green.

gold [25] nanorods was first reported by Jana et al. This is a seed-mediated method in which penta-twinned seed particles, stabilised by trisodium citrate, are first produced with sodium borohydride acting as the reducing agent. Then the nanorods are grown from these seed particles, where a weak reducing agent is used to promote the growth from the existing seeds rather than forming new particles. The resulting nanorods are stabilised from cetyltrimethylammonium bromide (CTAB) which preferential binds to the {100} facets found on the sides of the nanorods and forms a bilayer around the rods [26]. The CTAB has been shown to lower the surface energy of the {100} facets as well as inhibit growth on these facets resulting in directed growth at the ends of the rod [27][28]. The crystal structure of these rods is shown in figure 1.2.

The CTAB is also said to stabilise strain in the nanorods which is due to a mismatch between the penta-twinned crystal and the single crystal parts that it is composed of. Figure 1.3 shows an origami representation of this, where the tetrahedral single crystal sections does not completely make up

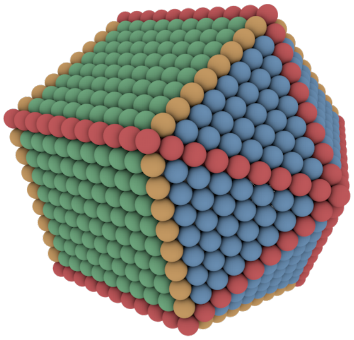


Figure 1.2: Crystal structure of a penta-twinned nanorod. The {111} facet is show in blue and the {100} in green. The five twin planes are shown in red.

the penta-twinned structure of the seed particles.

To gain a greater understanding of these growth mechanism, the size and shape of the nanoparticles needs to be characterised during the growth. Typically transmission electron microscopy (TEM) is used to characterise the size and shape of nanoparticles. However, up until recently, this could not be used in situ while the nanoparticle is growing; the growth of the nanoparticles would first need to be halted before the TEM images could be recorded. Recent work has shown that liquid cell transmission electron microscopy (LCTEM) can be used to study the growth of nanoparticles in situ [28]. LCTEM however, is still new and not everyone has access to use it.

As the surface plasmon resonance of the anisotropic nanoparticles is dependant on its size and shape, observing these resonances using UV-Visible extinction spectroscopy provides an excellent way to monitor the growth of

Figure 1.3: Origami representation of the penta-twinned seed particles made up from 5 tetrahedra. The five twin-planes can also be seen where the faces of each tetrahedra meet

these particles.

It has been shown that there is a linear relationship between the aspect ratio of gold/silver nanorods and the position of their associated longitudinal plasmon peak [29]. This provides an excellent way of determining the aspect ratio in situ during the growth of the nanorods However, this also presents a unique problem in that any growth models would need to describe the change in aspect ratio over time rather than the absolute size of the nanorod.

1.1 Project Aim

The aim of this masters project is to monitor the growth of silver nanorods in real time by examining the associated plasmon resonances found in the time-resolved UV-Visible extinction spectra.

The first goal will be to determine the relationship between the aspect ratio of silver nanorods and the associated plasmon resonances using the discrete-dipole approximation (DDA). This relationship will then be used to calculate the evolution of the aspect ratio from the longitudinal plasmon resonance recorded during the growth of the nanorods.

Finally, growth models will be used to explain the evolution of the aspect ratio of these nanorods grown under various conditions.

1.2 Definition of modes of attenuation of light from nanoparticle solutions

UV-Visible spectroscopy is a technique used in analytical chemistry to determine the quantity of an analyte present in a sample by measuring the attenuation of light that passes through a sample as a function of wavelength. There are two processes that contribute to this attenuation: absorption and scattering.

Typically when this technique is used, the scattering component is negligible so this technique is known simply as UV-Vis absorption spectroscopy. When this technique is used for nanoparticles, the scattering component is no longer negligible — especially for larger particles. Despite this, many of the sources cited in this thesis stick to the convention of using the name "UV-Vis absorption spectroscopy".

In Astronomy, The combined effect of the scattering and absorption is known as extinction. This definition is used in this thesis as the DDA software used for this research, DDSCAT, was developed to study interstellar dust grains and therefore uses the definition of extinction. When referring the work of other groups, "UV-Vis extinction spectroscopy" will be used, even when the referenced sources does not use this name.

1.3 Naming convention for the plasmon peaks found in the UV-Visible extinction spectra of silver nanorods

Chapter 4 will summarise the extinction spectra for silver nanorods of various sizes. These spectra all show two distinct peaks which are found to be associated with the longitudinal and transverse plasmon resonances of the nanorods. These peaks will therefore be called the longitudinal and transverse plasmon peaks. The transverse plasmon peak is always found at a lower wavelength than the longitudinal one.

In chapter 5, silver nanorods are prepared and studied. Throughout the chapter, UV-Visible extinction spectra will be recorded for these various nanorod preparations. As will be seen, two prominent plasmon peaks are present in all of these spectra, much like what is found in the computational results. To make it easier when discussing these spectra and the associated plasmon peaks, the terms ‘longitudinal peak’ and ’traverse peak’ will be used, even though it is not certain that these plasmon peaks are only associated with nanorod geometries.

1.4 Thesis Outline

Chapter 2 will outline the methods used to obtain all of the data presented in this thesis. Firstly, the development of the python module used for the peak fitting performed in this thesis is discussed. Then the computational methods used to calculate the extinction spectra of silver nanospheres and nanorods are outlined — The methods used include: Mie theory, DDA and Gan's Theory. Finally, the experimental methods used for the preparation and characterisation of silver nanorods are described.

A growth model which will be used to describe the evolution of the aspect ratio of the nanorod preparations will be outlined in chapter 3.

All of the computation calculations for silver nanoparticles will be presented in chapter 4. The first part is concerned with the correction that is required to be applied to the DDA calculation for the nanorods. The second part of this chapter will present the computed extinction spectra calculated by various methods for silver spheres and nanorods.

The experimental results obtained from the preparation of silver nanorods are presented in chapter 5. TEM and UV-Visible extinction spectroscopy results are reported to characterise the various plasmon peaks of the nanorod preparations. Finally, time-resolved UV-Visible extinction spectra, recorded during the growth of nanorods under various conditions, is reported.

Chapter 6 will discuss any problems that were found, either in the computation calculations or in the preparation of the nanorods. It will also be mentioned how the computation results, along with the presented growth model, will be combined to explain the evolution of the plasmon peaks observed in the time-resolved UV-Visible extinction spectra reocorded during the growth of the nanoparticles.

Finally, chapter 7 will present any conclusions as well as future work.

Chapter 2

Materials and methods

The first part of this chapter sets out the need for the data-fit Python module which was used to fit all the spectra in this thesis. A brief theory on plasmon peak fitting is mentioned and then instructions are provided to explain how the `data_fit` module is used.

The second part of this chapter goes over the computational calculations that were performed. Mie theory, DDA, and Gan's theory were used to calculated the extinction spectra of various silver nanoparticles. The exact parameters of each calculation are provided as well as instruction on how these calculation were analysed.

The last section of this chapter goes over the experimental methods that were used to prepare and characterise silver nanorods.

2.1 Spectral peak fitting using the data_processing python package

One of the objectives of this thesis was to determine the relationship between the geometry of silver nanoparticles and the parameters of their associated plasmon peaks.

One way to characterise these plasmon peaks is to use a local-maxima peak detection to find the peak positions and amplitudes. Obviously, this will lead to many peaks being detected in the experimental results as all the spikes in the noise will also be detected as peaks. This can be easily fixed, however, by requiring a minimum amplitude change for each local maximum detected.

The results of such a peak detect function applied to the time-resolved UV-Visible extinction spectra collected during the growth of a silver nanorods are shown in figure 2.1 [1].

The first thing to notice from this figure is the fact that the peak detect function fails to detect the second plasmon peak during the initial stages of the nanorod growth process. This occurs when there is only a small separation between the two plasmon peaks and the overlap of these peaks results in a superposition with only a single local maximum. The position of the local maximum from the superposition of the two peaks lies somewhere in between the positions of the individual peaks. This explains the slight shift to longer wavelengths of the detected position of the first peak during the initial stages of the growth.

As the reaction progresses the separation between the peaks increases and at some point the superposition of the overlapping peaks results in two sep-

---

[1] The peak detect function used here comes from a python package developed by Sixten Bergman [30]. The function is derived from a Matlab script developed by Eli Billauer [31]

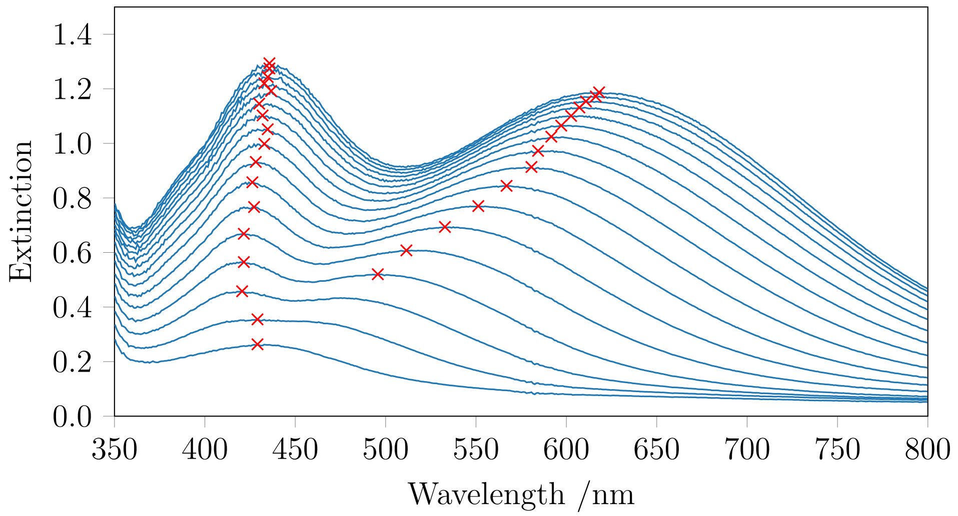


Figure 2.1: Detected plasmon peaks found using a local-maxima peak detect function for UV-Visible extinction spectra collected during the growth of silver nanorods.

arate local maxima. The peak detect function will still fail to detect the second peak, however, until difference in amplitude between the second peak maximum and the trough located between between the two plasmon peaks is larger than the minimum amplitude change requirement.

Figure 2.2 shows the evolution of the detected peak position of the two plasmon peaks found using the peak detect function.

The detected peak positions also appears to jump around, this can be seen in figure 2.2, which shows the evolution of the peak positions of the detected peaks. This occurs because of the noise of the data near the top of the plasmon peaks. A random spike from the noise in the data points near the top could make the extinction larger than the top of the peak and therefore, the spike in the noise would be detected as the peak. This becomes more prominent when the width of the plasmon peak increases in the later stages of the reaction as the top of the peak is flatter which allows these spikes to be detected as the peak over a larger range.. The noise in the spectra also increases in the later stages of growth, further adding to this problem. This occurs as the extinction increases as the nanoparticles grow larger which limits the amount of light that reaches the detector. In the later stages of the reaction, the detected peak position can jump as much as 20 nm.

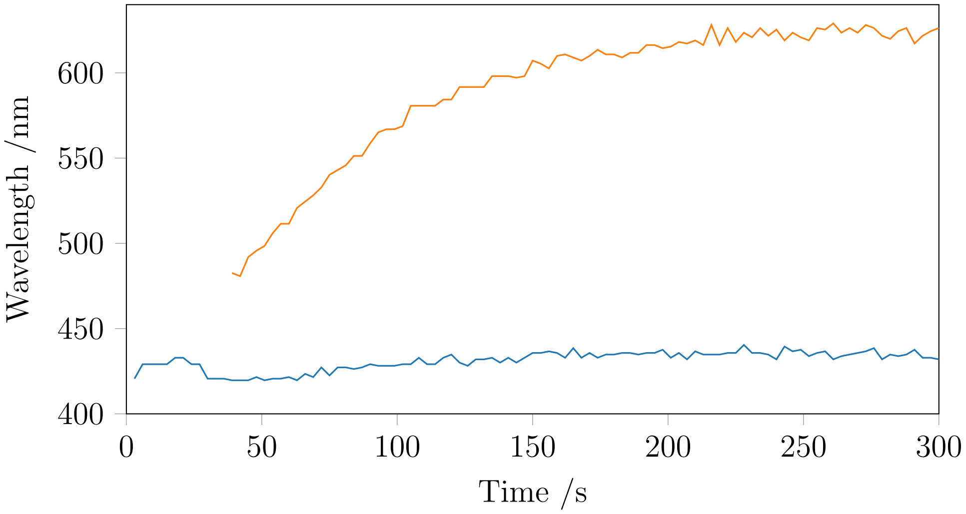


Figure 2.2: Evolution of the peak positions of the two peaks detected using a local maxima peak detect function for UV-Visible extinction spectra collected during the growth of silver nanorods.

Another method for characterising the plasmon peaks, which can accommodate the noise in experimental data, is peak fitting. Gaussian, Lorentzian, or Voigt functions can be fit to the plasmon peaks and a non-linear least squares regression can be used to find the optimised parameters of these peaks. Because the optimisation of these functions takes into account all the data points of the plasmon peak, the noise in the individual data points no longer has an affect.

Fitting these peak functions to the data has another advantage over a simple peak detection, in that the fitted functions also provide information about the width of the plasmon peaks. The width of these peaks is related to the polydispesity of the nanoparticles.

The shape of the the observed peaks can be the result of many different dispersive processes. Processes which involve exponentially decaying lifetimes, result in Lorentzian profiles — the Fourier transform of an exponential decay is a Lorentzian function. The equation of a Lorentzian function is:

$$f_{\mathrm{L}}\left(\nu_0, A, w\right) = \frac{A}{1 + \left(\frac{\nu - \nu_0}{w}\right)^2} \tag{2.1}$$

Where ν_0 is the position, A is the amplitude, and w is the width of the Lorentzian peak.

Gaussian profiles arise from processes involving a random distribution of exponentially decaying lifetimes. The equation of a Gaussian function is:

$$f_{\mathrm{G}}\left(\nu_0, A, w\right) = A e^{-\left(\frac{(\nu - \nu_0)}{w}\right)^2} \tag{2.2}$$

For situations where there are many dispersive processes occurring, the peak can have a Voigt profile — which is the convolution of a Gaussian and Lorentzian function. Because this convolution is computationally intensive, a pseudo Voigt function can be used instead. The pseudo Voigt function is the superposition of a Gaussian and a Lorentz function. The equation of the pseudo-Voigt function is:

$$f_{\mathrm{V}}\left(\nu_0, A, w\right) = \eta f_{\mathrm{L}} + \left(1-\eta\right) f_{\mathrm{G}} \tag{2.3}$$

The factor η determines the weighting between the Lorentzian and Gaussian functions.

These functions are all represented in the frequency-domain, while the UV-Visible extinction spectrum is obtained in the wavenumber-domain. This is because the grating in the UV-Visible spectrometer separates the light based on its wavelength, effectively taking the Fourier transform and transforming from the wavelength-domain into the wavenumber-domain.

Even though the extinction spectrum is in the wavenumber-domain, the UV-Visible spectrometer records the extinction in terms of wavelength. It is done this way because the light is dispersed linearly with respect to wavelength.

In the wavenumber-domain, the plasmon peaks are symmetric but the data points are not spaced linearly, and in the wavelength representation, the data points are spaced evenly but the plasmon peaks are asymmetric.

Before the peak functions can be fit to the extinction spectra, the frequency-domain functions need to be transformed to the wavelength-representation of the wavenumber-domain functions. This is done by replacing the frequency,ν, with $\frac{c}{\lambda}$, where c is the speed of light.

The wavelength-representation of the Lorentzian and Gaussian functions are shown in equations 2.4 - 2.5.

$$f_{\mathrm{L}}\left(\lambda_0, A, w\right) = \frac{A}{1+\left(\frac{\frac{1}{\lambda}-\frac{1}{\lambda_0}}{w}\right)^2} \tag{2.4}$$

$$f_{\mathrm{G}}\left(\lambda_0, A, w\right) = Ae^{-\left(\frac{\left(\frac{1}{\lambda}-\frac{1}{\lambda_0}\right)}{w}\right)^2} \tag{2.5}$$

$$f_{\mathrm{V}}\left(\lambda_0, A, w\right) = \eta f_{\mathrm{L}} + \left(1-\eta\right) f_{\mathrm{G}} \tag{2.6}$$

Where the speed of light, c, relating the frequency and the wavenumber is incorporated into the width parameter.

As this peak fitting was required to be performed on a large number of spectra, a Python module was created to automate this process. This module is called `data_fit` and is contained in a package called `data_processing`. The `data_fit` module was primary used for peak fitting but can also be used to fit any combination of mathematical functions to 2-dimensional data.

> Note: The mathematical functions used for the peak fitting should not be confused with the python functions found in the `data_fit` module or other python scripts. All of these python functions are explicitly referred to as ‘python functions’ in this thesis.

The following section provides an overview of how the `data_fit` module is used for peak fitting.

2.1.1 Using the data_fit module

To use the `data_fit` module, the `data_processing` package first needs to be imported. The location of the `data_processing` package either needs to be specified in the python script or added to the system PATH variable. The following block of code shows how to specify the location of this package as well as how to import and instantiate the `data_fit` module.

```
sys.path.insert(0, 'path/to/data-processing/folder')
from data_processing.data_fit import data_fit

Fit = data_fit()
```

This `Fit` object has a number of useful methods but the only two required to set up the function fitting are: `set_data` and `add_function`.

When using the `set_data` method, the data is required to be in a 2-dimensional numpy array where the first index refers to each data point and the second index refers to either the x or y value for that data point. This is the natural format for data that is read from a file as each line in the file contains all the information for a single data point.

```
data = numpy.array([
        [x1,y1],
        [x2,y2],
        ...
        ])
Fit.set_data(data)
```

If the x and y data is defined in the python script itself, the resulting numpy array will need to be transposed before it can be used in the `set_data` method.

```
x = [x1,x2,...] # x data
y = [y1,y2,...] # y data

data = numpy.array([x,y])
Fit.set_data(data.transpose())
```

The `add_functions` method is used to add the fit functions. There are many different fit functions that can be used and the `add_functions` method can be called multiple times to add additional functions to the fit. The total fit with be a superposition of all of the fit functions.

Only the functions used for the spectral peak fitting will be mentioned here, however, the remaining fit functions used will be discussed in the relevant section in the methods chapter. The functions used for the spectral peak fitting are the Lorentzian, Gaussian and pseudo-Voigt functions. The source code for all of the fit functions can be found in appendices C.3 - C.6.

These functions are expected to match the plasmon peaks of the nanoparticles in the wavenumber-domain, however, all of the spectra in this thesis were in terms of the wavelength. As discussed in the previous section, these function need to be transformed into the wavelength representation, which is shown in equations 2.4 - 2.6.

These functions are referred to as the wavelength-domain Lorentzian, Gaussian, and pseudo-Voigt, which are named, `'lorentzian_wavelength_domain'`, `'gaussian_wavelength_domain'`, `'pseudo_voigt_wavelength_domain'` in the `data_fit` module.

The following block of code shows how a wavelength-domain Lorentzian function is added to the `Fit` object:

```
position = ... # peak position in nm
amplituide = ... # peak amplitude
width = ... # peak width

bounds = ((position_min, position_max),
        (amplitude_min, amplitude_max),
        (width_min, width_max))

Fit.add_function('lorentzian_wavelength_domain',position,
        amplitude,width,bounds=bounds)
```

The 3 fit parameters are the initial values used in the optimisation and in most cases it does not matter what these are set to, although, they do need to be within the relevant bounds. All of the peak functions have the same 3 parameters and are added to the `Fit` object in the exact same way; the only difference being the name of the function which is passed as the first argument. `'gaussian_wavelength_domain'` and `'pseudo_voigt_wavelength_domain'` are used for the Gaussian and pseudo-Voigt function respectively.

The wavelength-domain pseudo-Voigt does have an additional parameter, η, however, this does not need to be specified as it is set to 0.5 by default and the bounds are (0,1).

Once the data has been set and all of the functions have been added to the `Fit` object, the optimisation can be started by calling the `optimise` method of the `Fit` object.

```
Fit.optimise(maxfev=1000000, ftol=3e-16,
            gtol=3e-16, xtol=3e-16, sigma=[...])
```

The `optimise` method calls the `curve_fit` python function found in the `scipy.optimize` module.

Before this `curve_fit` function is called, the parameters of all of the fit functions are compiled into a 1-dimensional list. This is done using the `Fit.get_optimise_params` method and the resulting list is passed to the `curve_fit` function using the `p0` keyword argument. The `Fit.get_optimise_params` method also compiles the bounds for all the function parameters which is then passed to `curve_fit` using the `bounds` keyword.

The `curve_fit` function also expects 3 positional arguments. The first being a python function that builds the total fit out of the individual fit functions; the `Fit.build_fit` method is used for this purpose. The remaining two positional argument are the lists containing the x and y data.

During each iteration of the optimisation, the `curve_fit` function will pass the list of updated parameters to the `Fit.build_fit` method. `Fit.build_fit` will then unpack these parameters and pass them onto their associated fit function. The `build_function` method is then called for each of the fit functions and the returned fits are added together to give the total fit. This total fit is then returned back to `curve_fit` where the cost function is calculated and if the required tolerances are not satisfied, the process is repeated for the next iteration of the optimisation.

Any additional keyword arguments that are passed to the `Fit.optimise` method are passed on to `curve_fit`. These additional keyword arguments were used to set the maximum number of evaluations, the tolerances, and the uncertainty in the y values of the data.

The `maxfev` argument sets the maximum number of evaluations that are performed before the optimisation fails. This was set to 1,000,000, which is larger than required but was set this large to ensure that the optimisation did not fail. The value of this argument has no effect on the results obtained from the optimisation.

The `ftol`, `gtol` and `xtol` arguments were used to set the tolerance for termination conditions used in the least squares optimisation. The termination conditions for these arguments are based on the change of the cost function, the change of the independent variables, or the norm of the gradient respectively. Exact definitions of these tolerances can be found in the documentation for the `scipy.optimise.least_squares` module [32].

All 3 of the arguments were set to 3×10^{-16} which is close to the machine epsilon of the computer used for the calculations. The machine epsilon is the smallest possible increment that can be represented between two floating point numbers. The machine epsilon of the machine used for all of these calculations was approximately 2.22×10^{-16}.

The final argument used was `sigma`. This argument takes a list containing the relative uncertainties in the y-values for each data point and was used to change the weighting of each data point in the least-squares optimisation. If this argument is not set, the weighting for each data point will be the same. This argument was rarely used so it can be assumed that it was not set unless explicitly stated.

After the optimisation has finished, the optimised parameters for each fit function are retrieved using the `Fit.get_functions` method. This method returns a list where each element in the list is another list containing the optimised parameters for each fit function. An example of this is shown below for a `Fit` object with a wavelength-domain Lorentzian and pseudo-Voigt functions.

```
[
```

```
    ['Lorentzian_wavelength_domain', position, amplitude,
    width],
    ['pseudo_voigt_wavelength_domain', position, amplitude,
    width, eta],
]
```

The order that the functions appears in this list is the same order in which they were added to the `Fit` object using the `Fit.add_function` method.

The confidence intervals for each optimised parameter are obtained in a similar way, using the `Fit.get_function_cis(confidence_level)` method. The `confidence_level` argument is a value between 0 and 100 and corresponds to the required confidence level.

This method returns a list with the same format as the `Fit.get_functions` method, however, the optimised parameters are replaced with the confidence intervals of each parameter.

```
[
    ['Lorentzian_wavelength_domain', position_ci,
    amplitude_ci, width_ci],
    ['pseudo_voigt_wavelength_domain', position_ci,
    amplitude_ci, width_ci, eta_ci],
]
```

The confidence interval for each parameter is calculated using the following equation:

$$CI = t\sqrt{\frac{ssr \times std}{dof}} \tag{2.7}$$

Where t is the t-value for a given confidence level calculated using the `scipy.stats.t` method. ssr is the sum of the square residuals. dof is the number of degrees of freedom, which is found by taking the number of data points minus the total number of fit function parameters. Finally, std is the standard deviation of each parameter, which is obtained by taking the square root of the diagonal elements of the covariance matrix returned by the `curve_fit` method.

For data sets consisting of a large number time-based spectra, the optimisation function can take a long time to compute. The data fit object is not required to be instantiated for every iteration, only the data needs to be updated. This will maintain the fit functions from the previous iteration, and the initial values of the fit function parameters will be set to the optimised values from the previous iteration — the optimised parameters are close to that of adjacent time steps. Reusing the data fit object along with the corresponding fit functions reduces the computation time by factor of approximately two.

An optimised fit produced by the `data_fit` module is shown in figure 2.3. The resulting fit (blue line) shows an excellent agreement to the data, using only three pseudo-Voigt functions.

In this thesis, the `data_fit` module is used to fit the time-based spectra recorded during the growth of the nanoparticles. An example of this is shown in figure 2.4. As can be seen, the position of the longitudinal peak (orange line) is shifted to longer wavelength when compared to the position detected using the peak detect function. Like the peak detect function, the data fit fails to detect the position of the longitudinal peak during the initial stages of the reaction, however, it does not detect the position until after it has already been detected by the peak detect function. This is because, when the plasmon peaks are overlapping, the wavelength-domain pseudo-

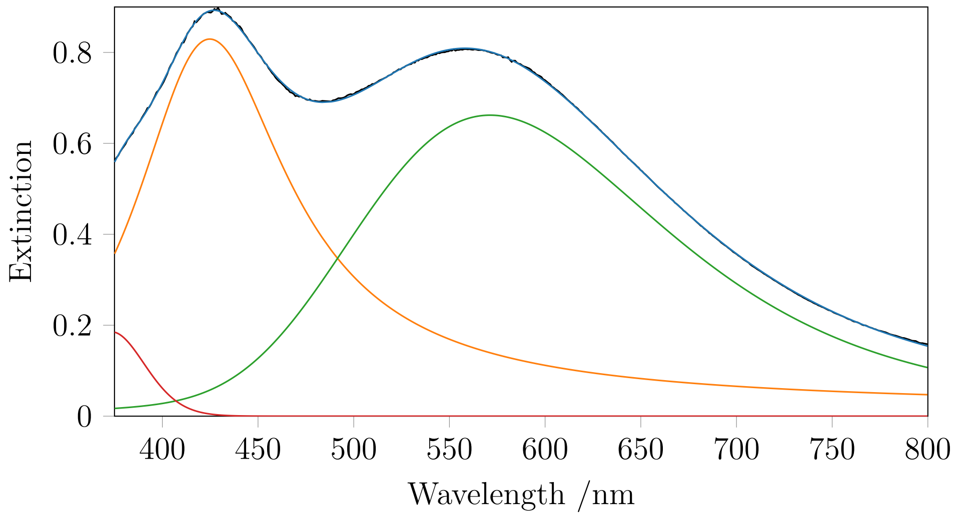


Figure 2.3: Optimised frequency domain pseudo-Voigt functions fitted to the UV-Visible absorption spectrum of a nanorod solution using the data fit python package.

Voigt function provides a sufficient fit using only a single peak, leaving the other function to move to longer wavelength to cover the absorption bands in this region.

Because the this method is using the optimised parameters from the previous iteration, the direction of the iterations is important. The natural choice is to have the iterations follow the direction of the reaction — from beginning to the end. However, if the iterations are reversed, the resulting optimisation successfully finds a fit for the initial stages of the reaction where it had previously failed to identify the position of the longitudinal plasmon peak. This can be seen in figure 2.5. When the iterations reach the time steps associated with the initial stages of the reaction, the data fit has already detected the position of both of the plasmon peaks from the previous time steps. The optimise method is able to find a fit where it had previously failed

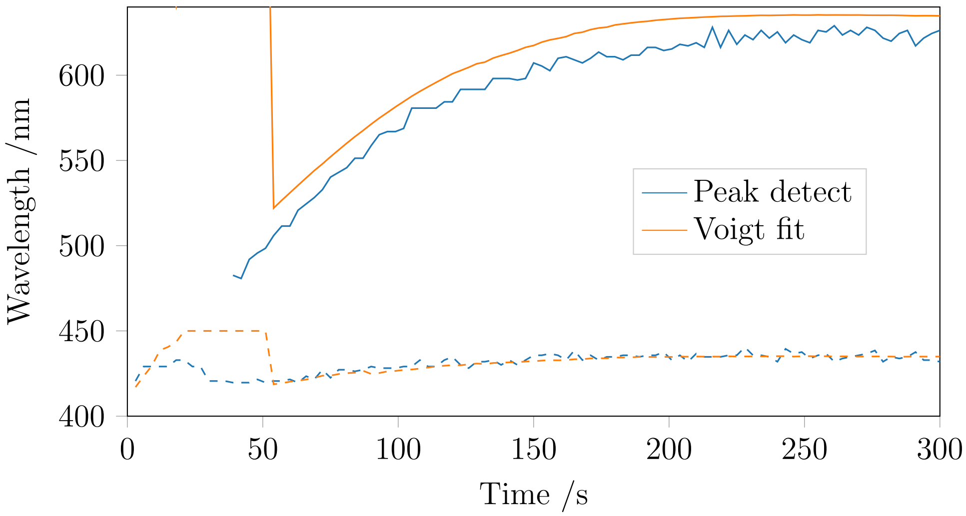


Figure 2.4: Evolution of the longitudinal and transverse plasmon peaks determined using an optimised pseudo-Voigt fit. The longitudinal peak position is shown as a solid line, while the transverse peak position is shown as a dashed line.

because the initial parameters already provide a close fit to the data. The evolution of the fitted peaks is shown in figure 2.6

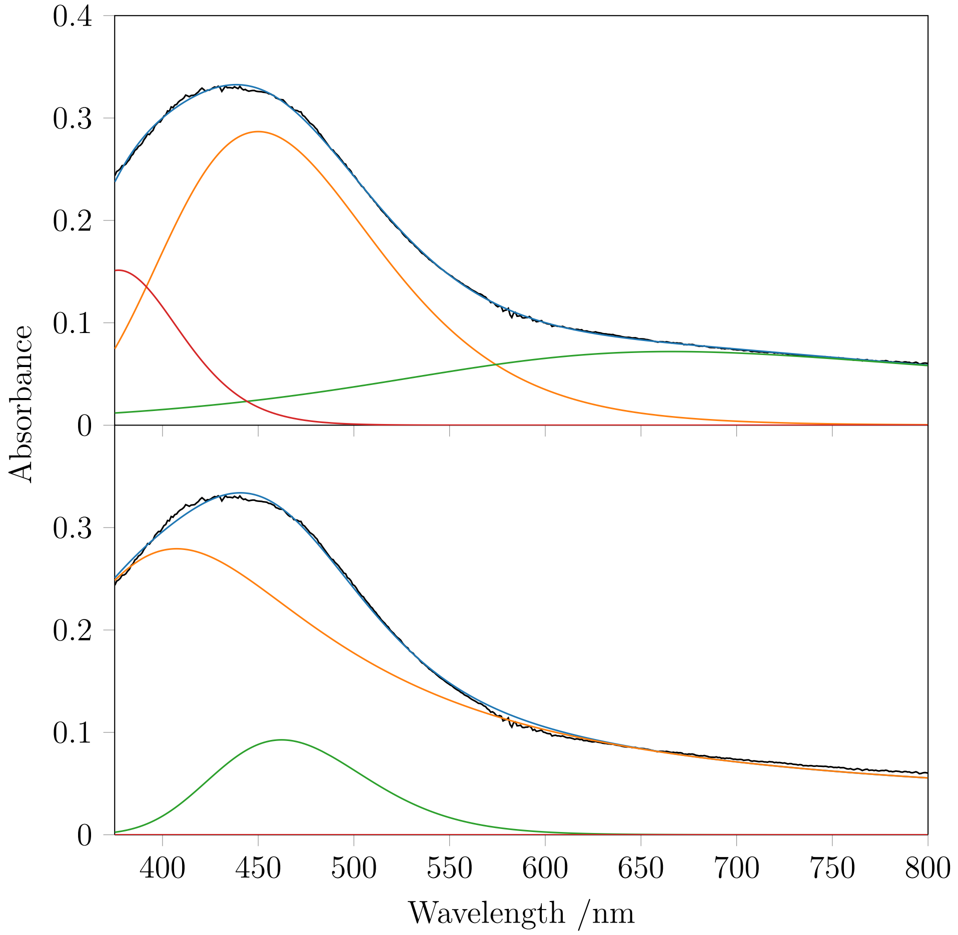


Figure 2.5: Optimised wavelength-domain pseudo-Voigt functions fitted to the UV-Visible extinction spectrum of the initial stages of nanorod growth. The top plot shows the optimised fit obtained with arbitrary initial fit parameters while the bottom plot uses the optimised fit parameters from the next time-step as the initial values. The peak on the far left shifts further left to fit the extinction outside of what is shown in the plot

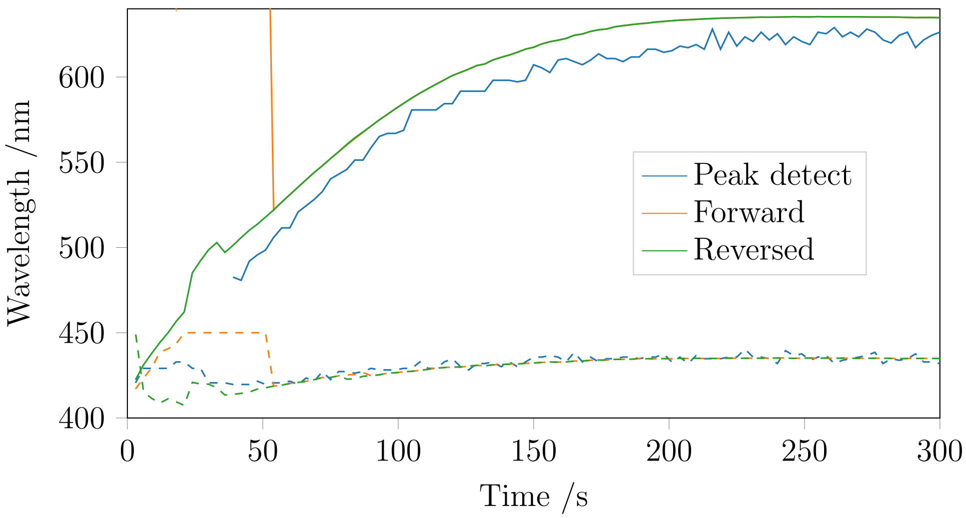


Figure 2.6: Evolution of the longitudinal and transverse plasmon peaks determined using optimised wavelength-domain pseudo-Voigt fits with different orders of iteration. The longitudinal peak position is shown as a solid line, while the transverse peak position is shown as a dashed line.

2.2 Computational methods for calculating the attenuation modes of silver nanoparticles

2.2.1 Optical properties of silver

All of the computational calculations in this thesis used the optical properties of silver which were found and reported by Johnson et al. [33]. The optical properties were measured from evaporated films of silver and are typically used when modelling nanostructures. The csv data file was obtained from `https://refractiveindex.info` [34].

The real and imaginary components of the refractive index are provided for a range of wavelengths from 200 - 1000 nm. Interpolated values for these parameters were calculated for wavelengths within this range which were not coincident with any of the given data points. Figure 2.7 shows the real and imaginary components of the refractive index provided by Johnson et al. along with the interpolated values.

Interpolation may introduce an error in the region between 280 and 340 nm where there is a significant change in the values of the data points. This is not expected to be a problem as all of the plasmon peaks found using the computational calculations were above 340 nm.

The complex dielectric constant was used in the Gan's theory calculations. This was calculated from the refractive index using equation 2.8

$$\varepsilon = n^2 - k^2 + i(2nk) \tag{2.8}$$

where n and k and the real and imaginary components of the refractive index.

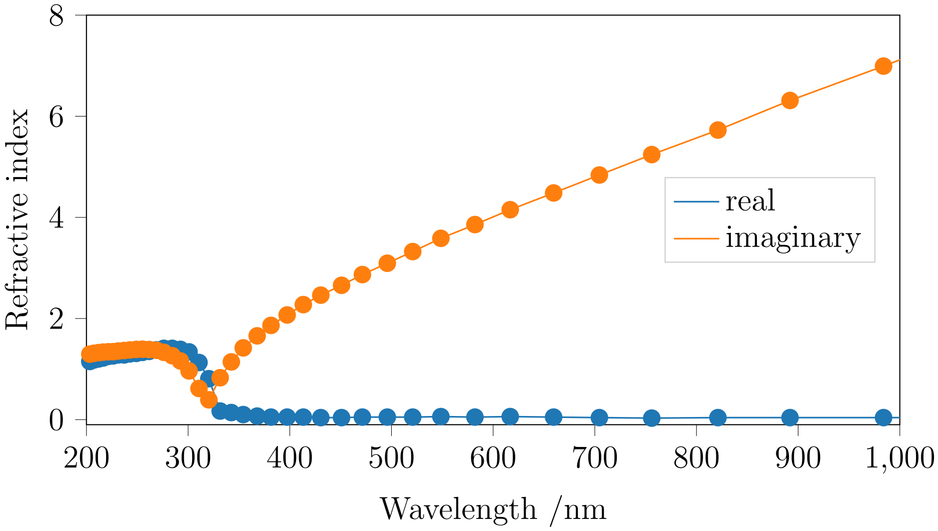


Figure 2.7: Real and imaginary components of the refractive index of silver obtained from Johnson et al. [33]. The solid line represents the interpolated values used in the computational calculations.

2.2.2 Mie theory for spherical nanoparticles

Bohren and Huffman developed the formulation for calculation the extinction, scattering and absorption efficiencies of spherical particles using Mie scattering theory [35].

The efficiencies are calculated using equations 2.9 - 2.11 [35][36].

$$C_{\text{ext}} = \frac{2}{x^2} \sum_{n=1}^{\infty} (2n+1)\, Re(a_n + b_n)) \tag{2.9}$$

$$C_{\text{scat}} = \frac{2}{x^2} \sum_{n=1}^{\infty} (2n+1) \left(|a_n|^2 + |b_n|^2 \right) \tag{2.10}$$

$$Q_{\text{abs}} = Q_{\text{ext}} - Q_{\text{scat}} \tag{2.11}$$

Where x is the size parameter, $x = ka$, where $k = \frac{2\pi}{\lambda}$, a is the radius of the sphere and λ is the wavelength.

Bohren and Huffman [35] recommend stopping the infinite sums when:

$$n_{\text{max}} = x + 4x^{1/3} + 2 \tag{2.12}$$

a_n and b_n are the Mie coefficients. Assuming the magnetic permeability of the sphere is 1, these equations become:

$$a_n = \frac{m^2 j_n(mx)[xj_n(x)]' - j_n(x)[mxj_n(mx)]'}{m^2 j_n(mx)[xh_n^{(1)}(x)]' - h_n^{(1)}(x)[mxj_n(mx)]'} \tag{2.13}$$

$$b_n = \frac{j_n(mx)[xj_n(x)]' - j_n(x)[mxj_n(mx)]'}{j_n(mx)[xh_n^{(1)}(x)]' - h_n^{(1)}(x)[mj_n(mx)]'} \tag{2.14}$$

Where m is the refractive index of the sphere relative to that of the surounding medium. j_n and $h_n^{(1)}$ and the spherical Bessel functions of order n.

The `PyMieScatt` Python package was used to calculate the attenuation spectra for spherical silver nanoparticles using Mie theory [37][38]. This package was written by Benjamin Sumlin and was installed using pip.

The PyMieScatt package uses a Rayleigh-regime approximation if the ratio of the diameter to the wavelength is less than $\frac{0.05}{\pi}$. For the calculations reported in this thesis, this approximation was used for spheres smaller than 6 nm. The result calculated while using this approximation was found to disagree with the results calculated when the approximation was not used. Because of this, the package was altered so that this approximation was never used for any of the Mie theory calculations. This was done by deleting lines 22 and 23 from the Mie.py file in the `PyMieScatt` package. These two lines are shown below:

```
elif x<=0.05:
    return RayleighMieQ(m, wavelength, diameter, nMedium,
    asDict)
```

The following block of python shows how the `PyMieScatt` package was used to calculated the extinction, scattering and absorptions efficiencies for a single wavelength and particle diameter.

```
import PyMieScatt as ps

diameter  = # Particle diameter in nm
wavelength = # Wathelength in nm
n_m = 1.33 # refractive index of water
m_Ag = n_Ag - 1.0j*k_Ag # Complex refractive index of
silver at the corresponding wavelength

qext, qscat, qabs, *_ =
        ps.MieQ(m_Ag,wavelength,diameter,nMedium=n_m)
```

	Function 1	Function 2
Position /nm	370	400
- Bounds /nm	365 - 380	300 - 500
Amplitude	0.000001	10
- Bounds	0 - 1000	0 - 1000
Width ($\times 10^{-6}$)	10	10
- Bounds ($\times 10^{-6}$)	1 - 40	10 - 600

Table 2.1: Parameters for the two wavelength-domain pseudo-Voigt functions used for the spectral peak fitting of the Mie theory calculations

`MieQ()` returns additional results such as the asymmetry parameter, however, none of these other parameters were used.

`MieQ()` was used to calculate the attenuation efficiencies for spheres with a diameter ranging from 5 to 60 nm with an increment of 5 nm. The wavelength was varied from 340 to 600 nm with an increment of 1 nm.

To determine the relationship between the size of the nanoparticles and the plasmon peaks, two wavelength-domain pseudo Voigt functions were fit to the spectra obtained from `PyMieScatt`. This was achieved using the python package described in section 2.1.1.

The initial parameters and parameter bounds of these two `'voigt_pseudo_frequency_domain'` fit functions are summarised in table 2.1.

2.2.3 Discrete-dipole approximation for spherical nanoparticles and nanorods

In practice, the Mie scattering theory can only be used to calculate the attenuation modes for particles which have a spherical geometry [10]. The DDA is a technique that was developed to find the scattering and absorption from particles with arbitrary geometry. The DDA was initially developed by Purcell and Pennypacker in 1973 to describe the scattering from interstellar dust grains [39].

In the DDA the scattering target is represented by by a finite array of polarisable dipoles. The scattering and absorption of the target is calculated by considering the polarisation that is induced at each of the lattice points.

The dipole moment that is generated at each point is proportional to the magnitude of the local electric field. The constant of proportionality is called the polarisability, α, and is related to the optical properties of the material that is associated with each lattice point.

$$P_i = \alpha_i E_{\mathrm{loc},i} \tag{2.15}$$

The local electric field at each lattice point is the sum of the external electric field and the scattered field from every other dipole.

$$E_{\mathrm{loc},i} = E_{\mathrm{ext}} + \sum_{j \neq i} A_{ij} P_j \tag{2.16}$$

where A is a matrix which describes the interaction between each dipole.

The induced polarisation at point i then becomes:

$$P_i = \alpha(E_{\text{ext}} + \sum_{j \neq i} A_{ij} P_j) \tag{2.17}$$

This results in a system of linear equations which can be solved to find the unknown polarisations.

DDScat is a Fortran program which solves the DDA for an arbitrary geometry. DDScat was developed by Draine to calculate the scattering and absorption from interstellar graphite grains [40]. Rather than solving equation 2.17 directly, DDScat implements the complex-conjugate gradient method (CCG) to iteratively find solutions for the dipole moments. When the number of dipoles is large, the CCG is typically faster than finding the direct solution [41]. To increase the speed of the CCG method further, fast Fourier transform (FFT) can be used to simplify the matrix-vector multiplications. As a consequence of this, DDScat uses a cubic lattice as the FFT technique requires that the lattice be periodic.

The main source of error from the DDA come from the fact that curved surfaces must be described in terms of cubic dipoles. This error is expected to decrease as the dimensions of the dipoles decrease, and disappear entirely when infinitely small dipoles are used.

It had been shown that for spherical targets this error scales with the inverse cube root of the number of dipoles, $\mathrm{N}^{-\frac{1}{3}}$ [40]. The relationship between the result for a finite and infinite number of dipoles is shown in equation 2.18. This scaling of the error matches the ratio of the number of surface dipoles to the total number of dipoles.

$$Q_N = Q_\infty - AN^{-\frac{1}{3}} \tag{2.18}$$

Figure 2.8 shows the extinction efficiencies calculated for spherical targets containing a various number of dipoles along with equation 2.18 optimised by non-linear regression. The data points show good agreement to equation 2.18, however, there is still some variation from this line.

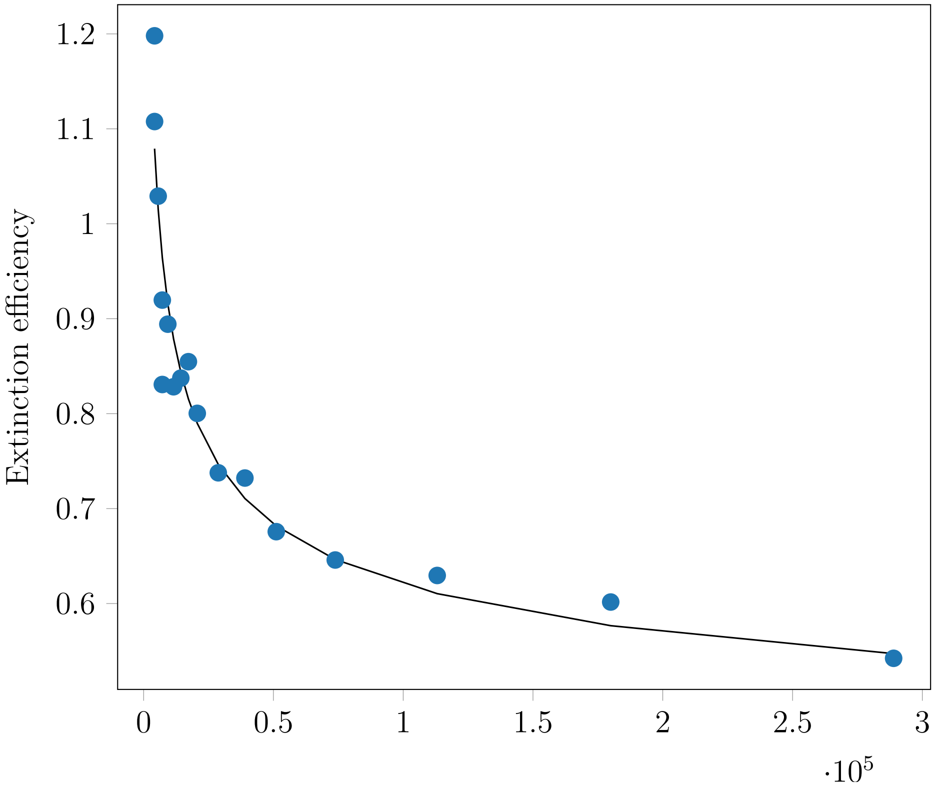


Figure 2.8: Extinction efficiencies calculated for targets containing a various number of dipoles.

To correct this error, Draine suggests performing the DDA calculations for targets with two different dipole densities and then extrapolating these results to find the solution when $N \rightarrow \infty$ [40]. Equation 2.19 shows the relationship between Q_∞ and Q_N obtained from the two different dipole densities. However, significant variation in the extrapolated values of Q_∞ is expected if there are only two data points used for the extrapolation. Because of this variation, DDA calculations were performed at 3 different inter-dipole spacings and a non-linear regression was used to fit equation 2.18. Non-linear regression also provides a standard deviation of the fit parameters which can be used to calculate confidence intervals for Q_∞.

$$\begin{aligned} A = \frac{Q_\infty - Q_1}{N_1^{-\frac{1}{3}}} &= \frac{Q_\infty - Q_2}{N_2^{-\frac{1}{3}}} \\ Q_\infty &= \frac{Q_1 - Q_2\left(\frac{N_1}{N_2}\right)^{-\frac{1}{3}}}{1 - \left(\frac{N_1}{N_2}\right)^{-\frac{1}{3}}} \\ Q_\infty &= \frac{Q_2 - Q_1\left(\frac{N_1}{N_2}\right)^{\frac{1}{3}}}{1 - \left(\frac{N_1}{N_2}\right)^{\frac{1}{3}}} \end{aligned} \tag{2.19}$$

2.2.4 Using DDSCAT for the DDA calculations

DDSCAT 7.3.2 was used to calculate the attenuation modes of the silver nanoparticles using the DDA [42]. DDSCAT is written in Fortran and was compiled from source using ifort. Before compilation, the Makefile was edited to enable single-precision arithmetic, OpenMp, Intel Math Kernel Library (MKL), and Message Passing Interface (MPI). The machine used for the DDA calculations was running Debian.

mpirun was used to run DDSCAT and was called from within python scripts using the subprocess module.

```
import subprocess
subprocess.Popen(['mpirun','-np','7','ddscat','ddscat.par'],
cwd='path/to/.par/file').wait()
```

The `-np` argument was used to set the number of processes used for the calculations and was limited by the amount of memory of the machine, which was 16 Gb. The command-line process-manager htop was used to monitor the memory used by each process and, if required, `-np` was reduced so that the machine did not run out of memory. The maximum number of processes used for a single calculation was 8, which is the number of threads of the CPU.

All the information for the various DDA calculations was provided to DDSCAT in the form of a file with a '.par' extension; this file will be referred to as the '.par' file.

There were many parameters that were required to be defined in the '.par' file, however, only a few of these were altered between the various DDA calculations that were performed. The important parameters are discussed here.

The first parameter that was changed specifies the target geometry. DDSCAT has a number of built-in target geometries that can be used. The 'ELLIPSOID' geometry was used for the spherical targets; when all three of the associated size parameters were set to the same value, the resulting geometry was spherical. For the nanorods, the 'CYLNDRCAP' geometry was used. This geometry is known as a capsule and consists of a cylinder with two hemispheric caps at each end, as shown in figure 2.9.

The size and shape of these geometries is given to DDSCAT in the form

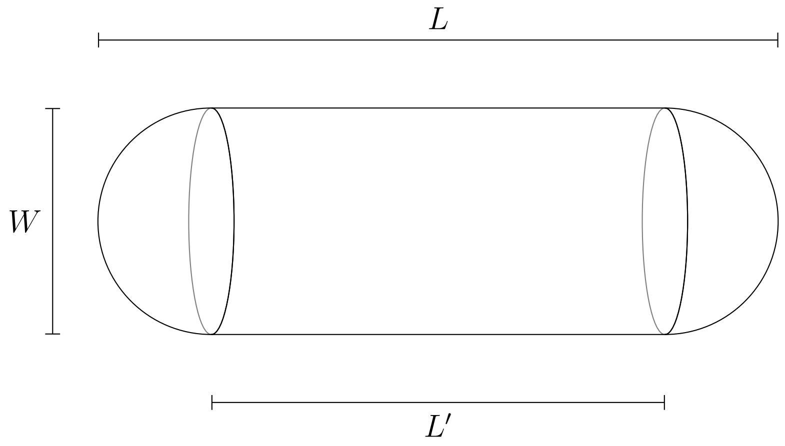


Figure 2.9: The "CYLNDRCAP" target geometry used in the DDA calculations for the nanorods. The parameter used to specify the 'length' of the rod, L', only specifies the length of the cylinder section of the rod and does not include the additional length from the end caps.

of unit-less parameters which describe the shape and size of the geometry relative to the inter-dipole spacing, d. The physical dimensions of the target are then defined by an additional parameter, called the 'effective radius', a_{eff}, which is the radius of a sphere which has the same volume as the target geometry. For spherical targets, the 'effective radius' is just equal to the radius of the target itself, but for the capsule targets this is not the case.

> Note: The 'length' specified when creating the 'CYLNDRCAP' geometry is not the true length of the capsule; it is only the length of the cylinder section and does not include the hemispheric end caps. A prime symbol is used to distinguish between this length parameter, L', and the true length, L — this is shown in figure 2.9. The true length of the rod, L, can be calculated from the specified size parameters using the following equation: $L = L' + W$

To determine the size parameters for a given geometry, a value for the inter-dipole spacing was first chosen. Then the size parameters were calculated by dividing either the diameter for the spheres or the 'length' and width for the capsules, by the inter-dipole spacing. For example, the size parameter for a sphere with diameter of 10 nm and a dipole spacing of 1 nm, would be 10.

The effective radius is then calculated. For the spherical targets, this is simply the radius of the target itself, but for the nanorods, a_{eff} is calculated using equation 2.20.

$$a_{eff} = \sqrt[3]{\frac{3}{16}L'W^2 + \frac{1}{8}W^3} \tag{2.20}$$

The refractive index of the silver targets was provided in a file containing the tab delimited real and imaginary components of the refractive index at various wavelengths; this file is discussed in section 2.2.1. The refractive index of water, 1.33, was used for the medium surrounding the nanoparticle targets and remained constant for all DDA calculations.

Although a range of wavelengths can be specified in the .par file, each DDSCAT calculation was performed at only a single wavelength and the results were later compiled using another python script. This was done so that if the calculations failed or the computer shutdown for any reason, the calculations could be restarted without having to repeat the calculations for wavelengths which already had a solution. The wavelength is specified in micron.

The final parameter that was defined in the .par file is the target rotation, as the attenuation modes of the capsule targets depend on the orientation of the target. To remedy this, the DDA calculations were performed for a range of target rotations and the results are averaged; this is all handled by

DDSCAT. The symmetry of the capsule geometry was used to reduce the range of target orientations. The required range is 90 degrees of rotation about the axis perpendicular to the long axis of the capsule geometry. The two extremes of this rotation has the capsule either pointing directly at the incoming light or side on to it. Five angles evenly spaced across this range were used for the orientations of the capsule target in the DDA calculations, the resulting attenuation modes were then averaged.

A python script was used to generate the .par files for the various geometries and another script was used to compile the resulting data after the DDA calculations were complete. Examples of .par files for the sphere and capsule geometries are shown in appendix A.

2.2.5 Convergence of the DDA calculations

DDSCAT was used to study the convergence of the DDA calculations for spherical and rod-like targets. This was achieved by calculating the extinction efficiency at a single wavelength for targets with a range of inter-dipole spacings. All of the DDA calculations were performed for a wavelength of 420 nm.

The '.par' files for each calculations were set up as discussed in section 2.2.4.

For the spherical geometry, the diameter of the targets was varied from 5 - 60 nm with a 5 nm step size. The target geometry was set to 'ELLIPSOID' and all 3 of the size parameters were set to the same value which was equal to the target diameter divided by the inter-dipole spacing.

The inter-dipole spacings were varied from 0.1 to 1 nm with a step size of 0.05 nm. However, for the larger particles, the calculations failed at the smaller dipole spacings. The final inter-dipole spacing used for all of the spherical particles are summarised in table 2.2.

Target diameter /nm	Minimum d /nm	Maximum d /nm	d increment / nm
5	0.10	1	0.05
10	0.10		
15	0.20		
20	0.25		
25	0.30		
30	0.35		
35	0.40		
40	0.45		
45	0.50		
50	0.55		
55	0.60		
60	0.65		

Table 2.2: Inter-dipole spacing ranges used to determine the DDA convergence for spherical targets

The data-fit package (see section 2.1.1) was used to fit equation 2.21 to the calculated extinctions efficiencies.

$$f = Q_\infty - aN^{\frac{1}{\chi}} \tag{2.21}$$

Where N in the total number of dipoles in the target geometry, χ is the correction factor and Q_∞ is the extinction efficiency the DDA calculations would converge on for an infinitely small dipole spacing.

The fit function for equation 2.21 is called 'dda_correction_general' in the `data_fit` module. The bounds for the 3 parameters were left as the defaults.

The y-data of the `Fit` object was set to the calculated extinctions efficiencies and the x-data was set to the total number of dipoles in the target, which was returned by the DDA calculations.

When calling the optimise method of the `Fit` object, the sigma argument was set to $1/N$ for all of the data points.

The optimised fit parameter χ was used to confirm the required correction that was applied to the DDA calculation for the spherical targets.

This was repeated for the rod-like geometries to determine the required correction for these geometries. The rod sizes and inter-dipole spacings that were used for these calculations are summarised in table 2.3. Initially, the calculations for all of the targets were performed using an inter-diploe spacing range of 0.1 - 1 nm, however, some of these calculations failed. For some of the targets, additional calculations were performed; these calculations are summarised in table 2.4.

Target width /nm	Target 'length' /nm	Minimum d /nm	Maximum d /nm	d increment / nm
5	5	0.1	1	0.1
5	10	0.1	1	
10	5	0.1	1	
10	10	0.1	1	
10	15	0.1	0.5	
15	5	0.1	1	
15	10	0.1	1	
15	15	0.2	0.6	

Table 2.3: Inter-dipole spacing ranges used to determine the DDA convergence for rod-like targets

Target width /nm	Target 'length' /nm	Additional d values /nm
10	5	0.12, 0.15
10	10	0.12, 0.15
10	15	0.15, 0.25
15	15	0.15, 0.75

Table 2.4: Additional inter-dipole spacings used to determine the DDA convergence for rod-like targets

2.2.6 Determining DDA correction factor by counting the surface-dipoles

In another attempt to determine the correction factor required for the DDA calculations of the rod-like particles, Python scripts were used to study how the proportion of surface dipoles in the target geometry varied as the inter-dipole spacing was decreased.

In the Python script, a 3-dimension grid of dipoles was envisioned, with the script then looping over each of these to determines whether the coordinates of each dipole was contained within the target geometry or not. If the dipoles were also within one inter-dipole spacing from the surface of the target, these dipoles were considered to be 'surface dipoles'.

The below block of pseudo Python code describes how this dipole counting was achieved for the spherical targets.

```
x, y, z = # Coordinates of the dipole

# Radius of the target divides by the inter-dipole
spacing
radius_d = radius/d

# Checks if the dipole is contained within the spherical
target
if np.sqrt(np.power(x,2)+np.power(y,2)+np.power(z,2))
    <= radius_d:

    # Increments the total number of dipoles
    total_N += 1
```

```
    # Check if the dipole is less than d away from the
    target surface
    if np.sqrt(np.power(x,2)+np.power(y,2)+np.power(z,2))
        >= (radius_d-1):

        #  Increment the total number of surface dipoles
        surface_N += 1
```

The code for the capsule geometry was similar, however, the cylinder region of the geometry and the end caps were examined separately. This was primarily done because it was the simplest way to determine which dipoles were contained within the target geometry, but it had the added advantage of allowing the surface dipoles of the cap and the cylinder regions to be counted separately. The total number of surface dipoles was then calculated by adding up the surface dipoles from the two regions.

The pseudo Python code is show below to demonstrate how the surface dipoles are determined for the capsule geometries.

```
x, y, z = # Coordinates of the dipole

# Width and length of the target divides by the
inter-dipole spacing
width_d = width/d
length_d = length/d

# Checks if the dipole is contained within the cylinder
region of the capsule geometry
```

```
if abs(y) <= (length_d-width_d)/2 and
    np.sqrt(np.power(x,2)+np.power(z,2))
    <= width_d/2:
    # Increments the total number of dipoles
    total_N += 1

    # Check if the dipole is less than d away from the
    target surface
    if
    np.sqrt(np.power(x,2)+np.power(z,2))>=((width_d/2)-1):

        # Increments the total number of surface dipoles
        in the cylinder region
        cylinder_N += 1

# Checks if the dipole is contained within the
hemispherical end caps of the capsule geometry
elif np.sqrt(np.power(x,2)+np.power(z,2)
    + np.power(abs(y)-(length_d-width_d)/2,2))
    <= width_d/2:

    # Increments the total number of dipoles
    total_N += 1

    # Check if the dipole is less than d away from the
    target surface
    if np.sqrt(np.power(x,2)+np.power(z,2) +
        np.power(abs(y)-(length_d-width_d)/2,2))
        >= ((width_d/2)-1):
```

```
        # Increments the total number of surface dipoles
        in the end cap region
        end_cap_N += 1

# Calculates the total number of surface dipoles
surface_N = cylinder_N + end_cap_N
```

Looping over all of the dipoles with Python is slow, therefore, a Rust program was create to calculate this part. This Rust program was called within a Python script and the variables were passed between Python and rust using STDIN and STDOUT. The Python script called the Rust program using `subprocess.Popen()` as shown below.

```
result = subprocess.Popen(["path/to/rust/program",
    str(int(d*100)), str(length), str(width),
    str(xMax),str(yMax),str(zMax)],
    text=True,stdout=subprocess.PIPE)
result.wait()
out, err = result.communicate()

rust_result = [int(x) for x in out.split(',')]
```

The inter-dipole spacing was multiplied by 100 before being passed to Rust so that it could be passed as an integer rather than a floating point number.

The two Python scripts for the spheres and rods, along with the Rust source code, are shown in appendix B. The Python script for the spheres calls the same Rust program used for the Rods, where the length of the rod was set equal to the width, resulting in a spherical geometry.

After the script has looped through all of the dipoles, the inter-dipole spacing, d, was altered so that the total volume of all the dipoles considered to be part of the target matched the expected volume of the target geometry. This was done to follow the method that DDSCAT uses to create its target geometries. As a result of this, the size of the target was changed slightly. This affects the smaller targets the most, with the diameter of the 5 nm sphere changing by 5% for the calculation with am initial d of 1 nm. This change was insignificant for the majority of the calculations however, with the diameter only changing by 0.001% for the 5 nm sphere calculated using an initial d of 0.01 nm.

The following block of code shows how the inter-dipole spacing was scaled:

```
import numpy as np

# If using the spherical targets
target_vol = 3/4 * np.pi * radius**3

# If using the capsule targets
target_vol = 3/4 * np.pi * (width/2)**3
    + (np.pi*(width/2)**2 * length)

# Expected volume of a single cubic dipole
dipole_vol = total_vol/total_N

# Scale the inter-dipole spacing so that dipole volume
matches what is expected
d = np.cbrt(dipole_vol)
```

This dipole-counting was performed for spherical targets with diameters ranging from 5 to 30 nm with a 1 nm increment. For the targets with a diameter of 17 nm or larger, the inter-dipole spacing ranged from 0.02 to 1 nm with a increment size of 0.01 nm, while the inter-dipole spacing range for the smaller targets started at 0.01 nm with the same increment size.

When performed on the capsule geometries, the width of the capsule target ranged from 5 to 15 nm and the length ranged from (width+1) to (width+30) nm with an increment size of 1 nm for each. The inter-dipole spacing for all of the geometries ranged from 0.01 to 1 nm with an increment size of 0.01 nm.

After the surface dipole counting, the `data_fit` module was used to fit the 'dda_correction_general' function to the data. The y-data was set to the proportion of surface dipoles to total dipoles, $\frac{N_S}{N}$, while the x-data was set to the total number of dipoles, N. The sigma argument of the `Fit.optimise()` method was set to $\frac{1}{N}$ for all of the data points.

For the capsule geometry, the function fitting was performed two more times where the y-data was set to the proportion of surface dipoles in the cylinder section to the total number of dipoles, as then the same for the proportion of surface dipoles in the end caps.

2.2.7 DDA calculations for silver spheres and nanorods

DDSCAT was used to calculated the attenuation spectra for spherical nanoparticles, which was then used to compare the results obtained using DDA and Mie theory.

For the spherical nanoparticles, the DDA calculations were performed using

Diameter /nm	Inter-dipole spacings /nm
5	0.05, 0.1, 0.2
10	0.1, 0.2, 0.4
15	0.2, 0.3, 0.6
20	0.2, 0.3, 0.8
30	0.3, 0.4, 0.5
40	0.4, 0.5, 1.0
50	0.4, 0.5, 1.0

Table 2.5: Inter-dipole spacings for the DDA calculations for spherical nanoparticles

the ‘ELLIPSOID’ geometry with diameters ranging from 5 to 50 nm and wavelengths ranging from 300 to 500 nm, in increments of 2 nm. These calculations were repeated twice more for smaller inter-dipole distances. The exact values used for each of the spherical targets are shown in table 2.5.

For the nanorods, the DDA calculations were performed with the ‘CYLNDRCAP’ target geometry, with widths of 10, 20, and 30 nm and various lengths.

The calculations for the nanorods were significantly slower than that for the spheres and as a result, the wavelengths had to be manually chosen to coincide with the positions of the plasmon peaks. To determine the wavelengths to use for a given target, the DDA calculations were first performed with a relatively large inter-dipole spacing, d, for a range of wavelengths around where the plasmon peaks were expected to be — roughly between 300 - 800 with a step size of 20 nm, depending on the aspect ratio of the target. The resulting spectra were observed and the wavelengths that were coincident with the plasmon peaks were noted and used in the calculations with smaller inter-dipole spacings. The wavelengths were chosen such that there were at least six data points used for each plasmon peak and that the wavelengths were spread out over the entire width of the peak; if this condition was not met, additional wavelengths were added in between the existing ones. The fact the plasmon peaks were expected to shift to shorter wavelengths when the inter-dipole spacing was reduced was also kept in mind while selecting the wavelengths to use for the DDA calculations.

The maximum inter-dipole spacing for each of the nanorod targets was chosen such that it was less than $\frac{1}{10}th$ the width of the nanorod target. The maximum inter-dipole size used for any of the targets was 1 nm.

The `data_fit` module was used to apply the DDA correction as shown in equation 2.18. This correction was applied to each set of 3 efficiencies calculated for each wavelength which was then repeated for all of the different geometries as well as for the 3 attenuation modes.

The fit function used was 'dda_correction'. All of the fit parameters were left unbounded and their initial values were not changed as they are hard-coded in the function definition.

The x-data was set to the wavelength and the y-data was set to the efficiencies of each attenuation mode. The sigma values of all of the data points was set to $\frac{1}{N}$.

After the corrected spectra for all of the target geometries was found, the `data_fit` module was used to fit the plasmon peaks found in each spectra.

For the spherical target geometries, 2 wavelength-domain pseudo-Voigt functions were used. The function parameters and associated bounds for these two functions are summarised in table 2.7.

Width /nm	'Length' /nm	inter-dipole spacings /nm
10	5	0.2, 0.3, 0.4
10	10	0.2, 0.3, 0.4
10	15	0.2, 0.3, 0.4
10	20	0.3, 0.4, 1.0
20	15	0.2, 0.8, 1.0
20	20	0.2, 0.8, 1.0
20	30	0.2, 0.8, 1.0
30	5	0.5, 0.8, 1.0
30	10	0.5, 0.8, 1.0
30	15	0.5, 0.8, 1.0
30	20	0.5, 0.8, 1.0
30	30	0.5, 0.8, 1.0

Table 2.6: Inter-dipoles spacings used for the DDA calculations for the silver nanorods

	Function 1	Function 2
Position /nm	370	400
- Bounds /nm	365 - 375	300 - 500
Amplitude	1	10
- Bounds	0 - 100	0 - 100
Width ($\times 10^{-6}$)	10	10
- Bounds ($\times 10^{-6}$)	1 - 50	10 - 600

Table 2.7: Parameters for the two wavelength-domain pseudo-Voigt functions used for the spectral peak fitting of the DDA calculations for silver spheres

For the rod-like target geometries, 3 wavelength-domain Lorentzian functions were used for the spectral peak fitting. The function parameters and associated bounds for these two functions are summarised in table 2.8.

The `data_fit` module was finally used to add a linear fit to the position of the longitudinal plasmon peak against the aspect ratio of the nanorod. This fit function is called 'linear' and was left unbounded.

2.2.8 Gan's theory calculations for prolate spheroids

Gan's theory was used to calculate the extinction spectra of prolate silver spheroids with various aspect ratios. Spectral peak fitting was then used to find the parameters of the plasmon peaks in the extinction spectra which were compared to that found using the DDA. Finally, The Gan's calculations were repeated using different values for the dielectric constant of the surrounding medium, ε_m, ranging from 1 to 2. These calculations along with the relationship between the spheroid and capsule geometries were used to determine how the DDA calculations would be affected by a change in the

	Function 1	Function 2	Function 3
Position /nm	360	370	450
- Bounds /nm	360 - 380	365 - 400	400 - 800
Amplitude	1	1	10
- Bounds	0 - 100	0 - 100	0 - 100
Width ($\times 10^{-6}$)	10	10	10
- Bounds ($\times 10^{-6}$)	1 - 50	1 - 50	10 - 600

Table 2.8: Parameters for the three wavelength-domain Lorentzian functions used for the spectral peak fitting of the DDA calculations for silver nanorods

dielectric constant, ε_m.

The extinction cross sections, for the spheroids with aspect ratio, A, were calculated using equations 2.22 [43].

$$\sigma_{ext} = \frac{2\pi\varepsilon_m^{3/2}}{3\lambda} \sum_{j=a,b,c} \frac{\frac{1}{P_j^2}\varepsilon_2}{\left(\varepsilon_1 + \frac{1-P_j}{P_j}\varepsilon_m\right)^2 + \varepsilon_2^2} \tag{2.22a}$$

$$P_a = \frac{1-e^2}{e^2}\left[\frac{1}{2e}\ln\left(\frac{1+e}{1-e}\right) - 1\right] \tag{2.22b}$$

$$P_b = P_c = \frac{1-P_a}{2} \tag{2.22c}$$

$$e = \sqrt{1 - \left(\frac{1}{A}\right)^2} \tag{2.22d}$$

The indices, j, of the the sum in equation 2.22a correspond to the 3 axes of the spheroid; a being the long axis while b and c are the short axes. The P_j components are, therefore, the contribution to the extinction cross section coming from the three dimensions of the spheroid. ε_1 and ε_2 are the real and imaginary components of the of the dielectric constant of silver respectively, and ε_m is the dielectric constant of the medium surrounding the spheroid.

The Gan's calculations were performed for prolate spheroids with aspect ratios ranging from 1 to 4 with an increment of 0.01. The wavelength was varied from 340 to 1000 nm with an increment of 0.001 nm.

Two wavelength-domain pseudo-Voigt functions were fit to the plasmon peaks found in the extinction spectra. The initial parameters and associated bounds are shown in table 2.9. The optimised parameters of these peaks were compared to that found using the DDA calculations.

The `data_fit` module was then used to fit a linear function to the position of the longitudinal plasmon peak against the aspect ratio of the nanorod. This fit function is called 'linear' and was left unbounded. The comparison between these linear fit parameters and what was found using DDA revealed the relationship between the aspect ratios of the prolate spheroids and the capsule geometries.

The DDA and the Gan's calculations use two different geometries and are, therefore, not expected to agree. They do, however, both have a linear relationship between the aspect ratio and the longitudinal plasmon peak position and are expected to agree when the aspect ratios are both equal to 1, as the spheroid and capsule geometries converge on a sphere. The relationship of the aspect ratios of these two geometries is expected to be linear as both geometries show a linear relationship of the plasmon position with aspect ratio. This relationship is shown in equation 2.23.

$$A_{DDA} = \phi(A_{Gan's} - 1) + 1 \tag{2.23}$$

	Function 1	Function 2
Position /nm	370	400
- Bounds /nm	330 - 400	390 - 1000
Amplitude	0.01	10
- Bounds	0 - 100	0 - 100
Width ($\times 10^{-6}$)	10	10
- Bounds ($\times 10^{-6}$)	1 - 60	10 - 600

Table 2.9: Parameters for the two wavelength-domain pseudo-Voigt functions used for the spectral peak fitting of the Gan's theory calculations for silver prolate spheroids

The Gan's calculations were then repeated with different refractive indices of the medium surrounding the prolate spheroids, n_m. The refractive was varied from 1 to 2 with a step size of 0.01 and all the other parameters in the calculations remained the same.

2.3 Preparation of silver nanorods

CTAB capped silver nanorods were prepared using two step seed-mediated method. In the first step citrate capped silver seed particles were prepared by using sodium borohydride to reduce silver nitrate in a solution containing trisodium citrate. The next step in the preparation was to initiate regrowth of the silver seed particles, in a CTAB solution, by reducing additional silver nitrate with a weaker reducing agent (ascorbic acid), which only allows for reduction at the surface of the seed particles.

> Note: The geometry of the resulting silver nanorods can be controlled by varying the ratio of the silver atoms contained in the growth solution to that contained in the silver seeds. This ratio, γ, is the defining factor of the various nanorod preparations and is used in this thesis to distinguish between them all. Provided there are no new nanorods formed in the growth step, a larger γ value will correspond to nanorods with a larger size.

2.3.1 Preparation of silver seed particles

A 2.5 mM solution of silver nitrate and trisodium citrate was prepared in a 25 mL volumetric flask by dissolving 11.0 mg of silver nitrate and 18.0 mg of trisodium citrate dihydrate in milli-Q water (>18.2 M·cm). 2.5 mL of this solution was added to a second 25 mL volumetric flask, which was then topped up with Milli-Q water. This resulted in a final concentration of 0.25 mM for both silver nitrate and trisodium citrate. The 0.25 mM solution was then placed in a 50 mL Falcon tube along with a magnetic stir bar.

A 10 mM sodium borohydride solution was prepared in a 10 mL volumetric flask by dissolving 3.8 mg of sodium borohydride in Milli-Q water (>18.2 M·cm). This solution was placed in an ice bath to slow the decomposition of the sodium borohydride.

The Falcon tube containing the 0.25 mM silver nitrate/trisodium citrate solution was placed in a beaker on top of a magnetic stirrer, which was turn up to high. 750 µL of the ice cold sodium borohydride solution was added to the Falcon tube, and the solution was left to stir for 30 seconds. The growth of silver nanoparticles was evident by the yellow/brown colour of the solution, which developed over a couple of minutes.

The resulting seed solution was left for at least 2 hours before it was used for

γ	80 mM CTAB /mL	10 mM $AgNO_3$ /µL	100 mM Ascorbic acid /µL	Seed solution /µL
0.5	5	125	250	1000
1				500
2				250
4				125
8				62.5
16				31.25

Table 2.10: Volumes of solutions used in the preparation of silver nanorods

the nanorod preparation. This was to allow any excess $NaBH_4$ to decompose.

2.3.2 Regrowth of silver seed particles

A 1 M sodium hydroxide solution was prepared by dissolving 200 µg of sodium hydroxide pellets in a 5 mL volumetric flask topped up with Milli-Q water. An 80 mM CTAB solution was prepared by dissolving 2.9157 g of CTAB in a 100 mL volumetric flask topped up with Milli-Q water. Both of these solution were stored and used for future nanorod preparations.

A 10 mM silver nitrate solution was prepared by dissolving 8.5 µg of silver nitrate in a 5 mL volumetric flask topped up with Milli-Q water. A 100 mM Ascorbic acid solution was prepared by dissolving 88.1 µg of ascorbic acid in a 5 mL volumetric flask topped up with Milli-Q water.

Six nanorod solutions were prepared in 15mL Falcon tubes containing the CTAB, silver nitrate, and ascorbic acid solution along with varying volumes of the seed solution. The volumes of each solution, along with the defining ratio for each preparation, γ, are shown in table 2.10.

To initiate the reaction 50 µL of the 1 M sodium hydroxide solution was added to the Falcon tube. The cap was quickly screwed on and the tube was inverted a few times to ensure the sodium hydroxide was thoroughly mixed in.

The regrowth of the silver seed particles was evident from the colour change of the solution. The solution changed from the initial pale yellow colour and went through a series of colour changes over the course of a few minutes. The solution colour followed the series: pale yellow → dark yellow → orange → red → blue → green → pale green. The colour change stopped once all of the silver ions in the solution were consumed. The colour of each preparation stopped at a different point in this series and was dependant on the volume of the seed solution used.

2.3.3 Centrifugation of the nanorod preparations

Centrifugation was used to remove the excess CTAB along with any particles that did not grow into nanorods.

All of the nanorod preparations were weighed on a 4 figure balance and Milli-Q water was used to balance all the preparations.

The two nanorod preparations with γ values of 8 and 16 were spun at 3000 rpm for 10 minutes while the other four preparations were spun at 5000 rpm for 30 minutes. The temperature of the centrifuge was set to 25°C so that the CTAB did not crystallise.

After centrifugation, the supernatant was carefully removed from each solution using a glass pipette. Milli-Q water was added to the pellet until the Falcon tube weighed the same as it did before the supernatant was removed. The Falcon tube were then placed in a sonicator for 1 minute at a setting of 30% to redisperse the pellet.

The geometry of the resulting nanorods were characterised using TEM and the profile of the plasmon peaks were determined using UV-Visible extinction spectroscopy.

A Shimadzu UV-1800 UV-Visible Spectrophotometer was used to collect the UV-Visible extinction spectra of the nanorod preparations, quartz cuvettes were used.

The TEM images were obtained using an FEI Tecnai G2 biotwin TEM with tomography unit. The nanorod preparations first went through 2 rounds of centrifugation as described above and then a drop of each preparation was placed on top of a carbon coated Formvar TEM grid. The preparations were left for 1 minute after which filter paper was touched against the side of the droplet to soak up the additional liquid

2.3.4 Time-resolved UV-Visible extinction spectroscopy during nanorod growth

A Scinco S-3100 photodiode array UV-Visible spectrometer was used to observe the evolution of the plasmon peaks during the growth of the silver nanorods. The spectrometer also had a temperature controller attached. Quartz cuvettes were used.

The procedure used was the same as the bulk nanorod preparation, with all of the volumes reduced so that the reaction could be carried out in a cuvette. The volumes of each solution used are shown in table 2.11.

The Quartz cuvette containing the growth solution was place in the diode-array UV-Visible spectrometer and 30 µL of the 1 M sodium hydroxide solution was added to initialise the reaction. The solution was immediately stirred using a glass pipette and the spectrometer was started. An extinction spectrum was collected every 3 seconds and collection continued until

γ	80 mM CTAB /ml	10mM $AgNO_3$ /µL	100mM Ascorbic acid /µL	Seed solution /µL
0.5	3	75	150	600
1				300
2				150
4				75
8				37.5
16				18.75

Table 2.11: Volumes of precursor solutions used in the preparation of silver nanorods grown in the diode array UV-Visible absorption spectrometer.

the reaction was complete (3 - 5 minutes). The extinction spectrum of the solution of CTAB was recorded and used as the reference.

The diode array UV-Visible absorption spectrometer records the transmittance of light through the sample. Instead of using the spectrometer software to convert this to extinction, the transmittance from the reference sample was measured separately and was used to calculate the extinction using the following equation:

$$\text{Extinction} = \log_{10} \frac{I_o}{I} \tag{2.24}$$

Where I_o is the recorded spectrum of the reference sample and I is that from the nanorod sample.

2.4 Photochemical transformation of the silver seed particles

Aqueous solutions of trisodium citrate, PVP, L-argenine, silver nitrate, and sodium borohydride were prepared using Milli-Q water. The final concentration, weight of each chemical, and the volume of water used are shown in table 2.12. After the silver nitrate solution was prepared, aluminium foil was used to cover the volumetric flask as silver nitrate is very sensitive to light. The sodium borohydride solution is prepared just before it is to be used in the preparation of the silver nanoparticles as it decomposes in water.

	Concetration mM	Weight /µg	Water mL
Trisodium citrate	50	73.5	5
PVP	50	27.5	5
L-arginine	5	8.7	10
Silver nitrate	5	8.5	10
Sodium borohydride	100	18.9	5

Table 2.12: Solutions used to prepare silver nanoparticles for photochemical transformation.

The growth solution was then prepared by adding 0.5 ml of the trisodium citrate, 0.015 ml of the PVP, 0.05 ml of the L-arginine, and 0.2 ml of the silver nitrate solutions to 7 ml of Milli-Q Water in a Falcon tube along with a magnetic stir bar. This solution was stirred and the silver nitrate was reduced by adding 0.08 ml of the sodium borohydride solution.

After stirring for a few minutes, 1 ml of the nanoparticle solution was placed in a 4 ml quartz cuvette along with a micro magnetic stir bar. This cuvette was placed on top of a magnetic stir plate in front of an Oriel Q Series 75 W Xenon arc lamp, which was then irradiated with white light while the solution was stirred. A lens on the lamp was focused so that all of the light passed through the cuvette and a mirror was placed behind to reflect the light back through the solution — the path of the light could clearly be seen in the cuvette. The mirror was placed at an angle so that the reflected light irradiated a larger proportion of the solution in the cuvette. This can be seen in figure 2.10.

The actual setup used is shown in figure 2.11. Inside the black tube coming out the side of the lamp is a convex lens, which is used to focus the light on the cuvette.

At various points in this process, the UV-Visible extinction spectrum was recorded for the nanoparticle solution to determine the changes that occur in the plasmon peaks. The final extinction spectrum was recorded after 250 minutes of irradiation by the lamp.

Finally, TEM images of the resulting nanoparticles were obtained.

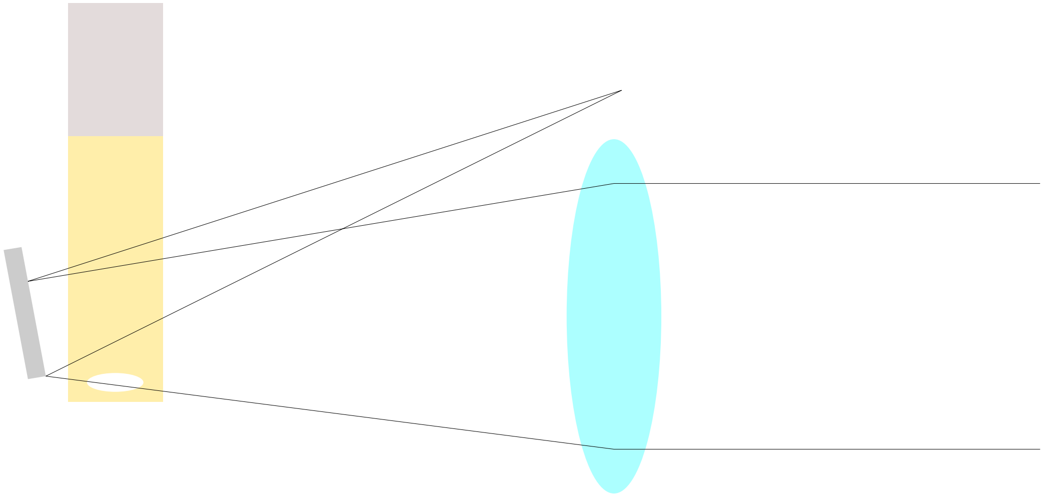


Figure 2.10: Diagram of the setup used for the photochemical transformation of silver nanoparticles

Figure 2.11: Experimental setup for the photochemical transformation of silver nanoparticles

Chapter 3

Modelling the growth kinetics of silver nanorods

This chapter follows the derivation of a growth model that describes the change in aspect ratio during the growth of nanorods. This model will then be used, along with the DDA results, to explain the evolution of the plasmon peaks observed during nanorod growth.

The following assumption was made to simplify the model of the growth of the nanorods:

- Only rod geometries are present in the nanoparticle solution
- The geometry of the nanorods are hemispherical-capped cylinders
- The seed particles are spherical
- No new seed particles form during the reaction
- The particle geometries are monodisperse

3.1 Hemispherical-capped cylinder growth model

The hemispherical-capped cylinder geometry used to model the growth of nanorods is shown in figure 3.1 — this is also known as a capsule geometry. The length of the rod is defined as L and the width as W.

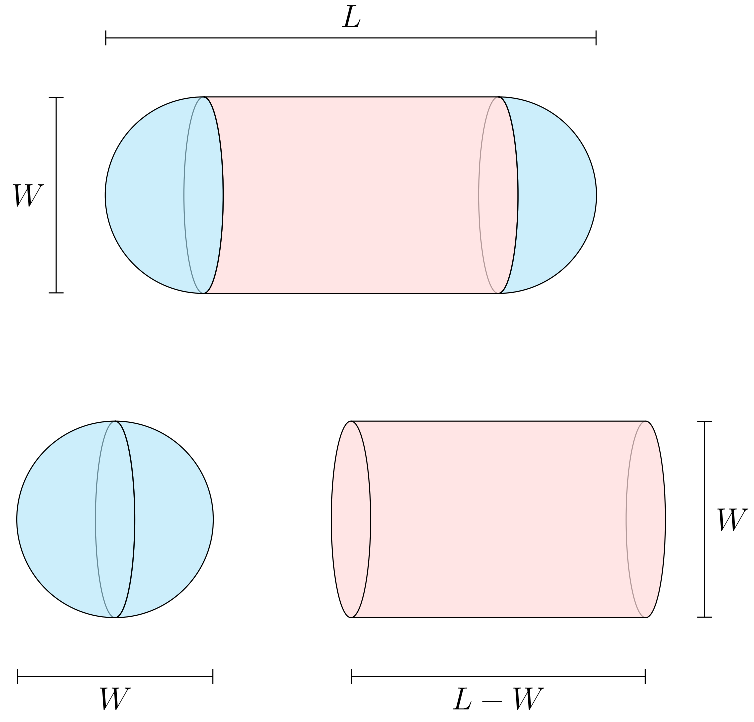


Figure 3.1: Geometry of the hemispherical-capped cylinder used for the modelling of nanorod growth. The two end caps are shown to form a sphere with diameter of W. The cylinder section is shown to have a length of $L - W$ and a diameter of W.

The first step in building a model for the growth kinetics of the capsule nanoparticles is to determine the equation for the volume of the capsule geometry. As can be seen in figure 3.1, the capsule geometry can be broken down into a sphere — formed by the two hemispherical end caps — and the middle cylinder section. The volume of the capsule geometry is therefore given by the sum of the volume of this sphere and cylinder.

The sphere formed by the two end caps has a diameter equal to the width of the capsule geometry, W. The total volume of the end caps is therefore given by the volume of this sphere:

$$\begin{aligned} V_{sphere} &= \frac{4}{3}\pi\left(\frac{diameter}{2}\right)^3 \\ &= \frac{4}{3}\pi\left(\frac{W}{2}\right)^3 \\ &= \frac{4}{3}\pi\frac{W^3}{8} \\ &= \frac{\pi}{6}W^3 \end{aligned} \tag{3.1}$$

The cylinder section of the capsule geometry has a length equal to the total length of the capsule, L, minus the length from the two end caps, $2 \times \frac{W}{2}$ — this can be seen in figure 3.1. The length of the cylinder section is therefore $L - W$ and the diameter is W. The volume of the cylinder section then becomes:

$$\begin{aligned} V_{cylinder} &= length \times \pi\left(\frac{diameter}{2}\right)^2 \\ &= (L - W) \times \pi\left(\frac{W}{2}\right)^2 \\ &= (L - W) \times \frac{\pi}{4}W^2 \\ &= \frac{\pi}{4}(L - W)\,W^2 \end{aligned} \tag{3.2}$$

The volume of a capsule geometry is therefore given by sum of equation 3.1

and equation 3.2:

$$V = \frac{\pi}{4}\left(L - W\right)W^2 + \frac{\pi}{6}W^3 \tag{3.3}$$

As the longitudinal plasmon peak position is known to have a linear relationship with the aspect ratio of silver nanorods, it is useful to define the aspect ratio of the capsule geometry and substitute this into the growth model in place of the length. The aspect ratio of capsule geometry is defined as:

$$A = \frac{L}{W} \tag{3.4}$$

Substituting the relationship between the aspect ratio and the length, $L = AW$, into equation 3.3 gives the equation for the volume of the capsule geometry in term of the aspect ratio:

$$\begin{aligned} V &= \frac{\pi}{4}\left(AW - W\right)W^2 + \frac{\pi}{6}W^3 \\ &= \frac{\pi}{4}\left(A - 1\right)W^3 + \frac{\pi}{6}W^3 \end{aligned} \tag{3.5}$$

As the nanorods grow, silver ions from the growth solution adsorb onto the surface of the rods resulting in an increase in the volume of the rods. As a result of this, the number of silver ions in solution will decrease as the growth of the nanorods proceeds. The relationship between the change in volume of the rods, dV_{rod}, and the change in the number of silver ions in solution, dN_{ions}, is given by:

$$dN_{ions} = -\frac{N_{rods}}{V_{atom}} dV_{rod} \tag{3.6}$$

where dV_{rod} is the change in volume of a single rod, N_{rods} is the number of rods in solution, and V_{atom} is the volume that a single atom occupies in the nanorod crystal.

This equation can also be expressed in terms of the concentration of silver ions and rods in the growth solution:

$$dC_{ions} = -\frac{C_{rods}}{V_{atom}} dV_{rod} \tag{3.7}$$

This equation can be integrated with respect to time to give an equation relating the volume of the nanorods and the concentration of silver ions in solution during the growth phase:

$$C_{ions}(t) = -\frac{C_{rods}}{V_{atom}} V_{rod}(t) + R \tag{3.8}$$

The volume of the nanorod geometry, equation 3.5, can be substituted in this equation to give:

$$C_{ions}(t) = -\frac{C_{rods}}{V_{atom}} \left(\frac{\pi}{4} \left(A(t) - 1 \right) W(t)^3 + \frac{\pi}{6} W(t)^3 \right) + R \tag{3.9}$$

The constant R can then be determined from the initial conditions of the growth solution (when $t = 0$).

The only nanoparticles present in growth solution at $t = 0$ are the spherical seed particles; therefore, the initial aspect ratio, $A(0)$, is equal to 1 and the initial width, $W(0)$, is W_0. $C_{ions}(0)$ is equal to the concentration of silver ions added to the growth solution, C_0. Substitution these initial conditions into equation 3.9 and rearranging gives the following equation for R:

$$\begin{aligned} R &= C_0 + \frac{C_{rods}}{V_{atom}}\left(\frac{\pi}{4}\left(1-1\right)W_0^3 + \frac{\pi}{6}W_0^3\right) \\ &= C_0 + \frac{C_{rods}}{V_{atom}}\left(\frac{\pi}{6}W_0^3\right) \end{aligned} \tag{3.10}$$

Equation 3.9 then becomes:

$$\begin{aligned} C_{ions}(t) &= C_0 + \frac{C_{rods}}{V_{atom}}\left(\frac{\pi}{6}W_0^3\right) - \frac{C_{rods}}{V_{atom}}\left(\frac{\pi}{4}\left(A(t)-1\right)W(t)^3 + \frac{\pi}{6}W(t)^3\right) \\ &= C_0 + \frac{C_{rods}}{V_{atom}}\left(\frac{\pi}{6}W_0^3\right) - \frac{C_{rods}}{V_{atom}}\left(\frac{\pi}{6}W(t)^3\right) \\ &\quad - \frac{C_{rods}}{V_{atom}}\left(\frac{\pi}{4}\left(A(t)-1\right)W(t)^3\right) \end{aligned} \tag{3.11}$$

As the rods only grow from the seed particles, and it is assumed that no additional seed particles form during the reaction, the concentration of the rods is equal to the concentration of the seed particles added to the nanorod preparation, $C_{rods} = C_{seeds}$. Although the seed concentration is not known, it can be related to the parameters used for the nanorod growth step.

$$C_{seeds} = \frac{V_{atom}}{V_{seed}}C_0' \tag{3.12}$$

where V_{seed} is the volume of a single spherical seed particle and C_0' is the concentration of silver atoms from the seed solution which is added in the nanorod growth step.

The ratio of the two concentrations, C_0 and C_0', is known, and is a defining factor of the various nanorod preparations:

$$\gamma = \frac{C_0}{C_0'} \tag{3.13}$$

so,

$$C_{seeds} = \frac{V_{atom}}{V_{seed}} \frac{C_0}{\gamma} \tag{3.14}$$

The volume of the seed in the hemispherical-capped cylindrical rod growth model is equal to the volume of a sphere with a diameter W_0:

$$V_{seed} = \frac{\pi}{6} W_0^3 \tag{3.15}$$

therefore,

$$C_{rods} = C_{seeds} = \frac{6}{\pi} \frac{V_{atom}}{W_0^3} \frac{C_0}{\gamma} \tag{3.16}$$

Substituting this into equation 3.11 gives:

$$
\begin{aligned}
C_{ions}(t) &= C_0 + \frac{V_{atom}}{V_{atom}W_0^3}\frac{C_0}{\gamma}\left(W_0^3\right) - \frac{6}{\pi}\frac{V_{atom}}{V_{atom}W_0^3}\frac{C_0}{\gamma}\left(\frac{\pi}{6}W(t)^3\right) \\
&\quad - \frac{6}{\pi}\frac{V_{atom}}{V_{atom}W_0^3}\frac{C_0}{\gamma}\left(\frac{\pi}{4}\left(A(t)-1\right)W(t)^3\right) \\
&= C_0 + \frac{C_0}{\gamma} - \frac{C_0}{\gamma}\left(\frac{W(t)}{W_0}\right)^3 - \frac{3}{2}\frac{C_0}{\gamma}\left(A(t)-1\right)\left(\frac{W(t)}{W_0}\right)^3
\end{aligned}
\tag{3.17}
$$

This is the general growth model describing how the size of the nanorods change as the concentration of silver ions in solution decreases

3.1.1 Growth along a single axis

If the nanorods are assumed to only grow in length during the growth reaction then the width of the rods will remain constant and $W(t) = W_0$.

This simplifies the general growth model, equation 3.17; which then becomes:

$$
C_{ions}(t) = C_0 - \frac{3}{2}\frac{C_0}{\gamma}\left(A(t)-1\right) \tag{3.18}
$$

Under the assumption that the growth rate is proportional to the concentration of silver ions in solution, the change in length of the nanorod can be expressed as:

$$
\frac{dL}{dt} = kC_{ions}(t) \tag{3.19}
$$

where k is the rate constant.

This rate equation can be expressed in terms of the change in the aspect ratio of the nanorods by using the definition of the aspect ratio (equation 3.4):

$$\frac{dA}{dt} = \frac{dA}{dL}\frac{dL}{dt} = \frac{1}{W_0}kC(t) \tag{3.20}$$

Substituting the growth model for nanorods growing along only the longitudinal axis, equation 3.18, into this rate equation gives:

$$\begin{aligned}\frac{dA}{dt} &= \frac{k}{W_0}\left(C_0 + \frac{3}{2}\frac{C_0}{\gamma}\left(1 - A(t)\right)\right) \\ &= \frac{k}{W_0}\left(C_0 + \frac{3}{2}\frac{C_0}{\gamma}\right) - \frac{3}{2}\frac{C_0}{\gamma}\frac{k}{W_0}A(t) \\ &= \frac{k}{W_0}C_0 + a - aA(t)\end{aligned} \tag{3.21}$$

where

$$a = \frac{3}{2}\frac{C_0}{\gamma}\frac{k}{W_0} \tag{3.22}$$

This equation is a first order linear ordinary differential equation (ODE), which can be solved using an integrating factor:

$$\frac{dA}{dt} + aA(t) = \frac{k}{W_0}C_0 + a \tag{3.23}$$

$$e^{at}\frac{dA}{dt} + e^{at}aA(t) = e^{at}\left(\frac{k}{W_0}C_0 + a\right) \tag{3.24}$$

$$\int \left(e^{at}\frac{dA}{dt} + e^{at}aA(t) \right) dt = \int \left(e^{at}\left(\frac{k}{W_0}C_0 + a \right) \right) dt \tag{3.25}$$

$$e^{at}A(t) = \frac{e^{at}\left(\frac{k}{W_0}C_0 + a \right)}{a} + R \tag{3.26}$$

$$\begin{aligned} A(t) &= \frac{k}{aW_0}C_0 + 1 + Re^{-at} \\ &= b + 1 + Re^{-at} \end{aligned} \tag{3.27}$$

where

$$b = \frac{k}{aW_0}C_0 \tag{3.28}$$

R can then be determined using the initial conditions ($t = 0$, $A(0) = 1$):

$$R = -b \tag{3.29}$$

Substituting R into equation 3.27 gives the final equation describing the change in aspect ratio during the growth of the nanorods:

$$A(t) = b + 1 - be^{-at} \tag{3.30}$$

An example of the change in the aspect ratio during this growth step is shown in figure 3.2. For this example, the following values were used for the parameters in the model: $a = 0.2$ and $b = 1$.

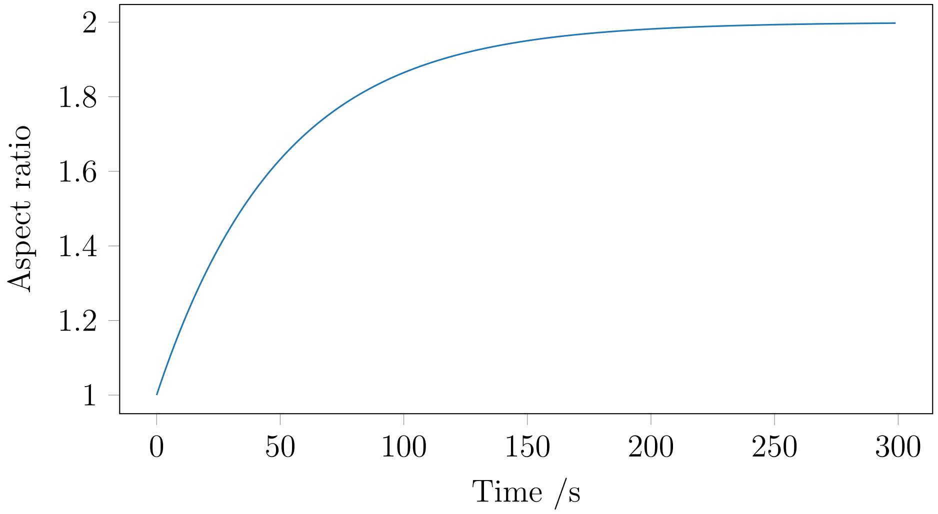


Figure 3.2: One dimensional cylindrical nanorod growth model, equation 3.30, with parameters $a = 0.2$ and $b = 1$.

Chapter 4

Computational attenuation modes of silver nanostructures

Chapter 4 summarises the computational calculations that were performed in an attempt to understand the relationship between the plasmon peak frequencies and the geometries of silver nanoparticle and nanorods. This was achieved using the DDA to calculate the attenuation spectra of spheres and hemispheric-capped cylindrical rods of various sizes.

The first section of this chapter deals with the convergence of the DDA calculations. The largest contribution to the error associated with the DDA comes from the attempt to approximate the curved surfaces of the target geometry with cubic dipoles. As the inter-dipole spacing converges on zero, this error is expected to disappear as the infinitely small cubic dipoles would provide a perfect representation of the desired target geometry. If the convergence relationship between the dipole size and the error is known, a correction can be applied to eliminate this error. For spherical targets, the convergence relationship of the attenuation efficiencies, as well as the associated correction, is known; however, this is not the case for the capsule geometry. In an attempt to determine this relationship and the associated correction, the attenuation efficiencies are calculated for a range of rod sizes and inter-dipole spacings and then the convergence of these efficiencies is examined.

The next section summarises the computational attenuation efficiencies obtained for sphere and rods geometries, and peak fitting was used to determine the relationship between the particle geometry and the plasmon peaks.

For spherical particles, both DDA and Mie theory results are shown. These results are primarily used to check the agreement of the DDA calculations, as well as the applied correction, with Mie Theory. This cannot be shown for the rods as there is no exact solution to Maxwell's equations for rod geometries.

DDA calculations are then used to calculate the attenuation efficiencies for with rod geometries. Wavelength-domain pseudo-Voigt functions are then fitted to the calculated spectra to determine the relationship between the plasmon peaks and the rod geometries. This is primarily focused on the longitudinal plasmon peak position and its relationship with the aspect ratio of the rods.

Finally, the effect of the surrounding refractive index on the plasmon peaks is studied using Gan's theory, after the relationship between DDA and Gan's theory is established. Gan's theory is used for this as the calculations are magnitudes faster than that of the DDA.

4.1 Convergence of the DDA results

Figure 4.1 shows the extinction efficiency calculated using various dipole sizes for spherical silver nanoparticles with diameters of 15 nm and 60 nm.

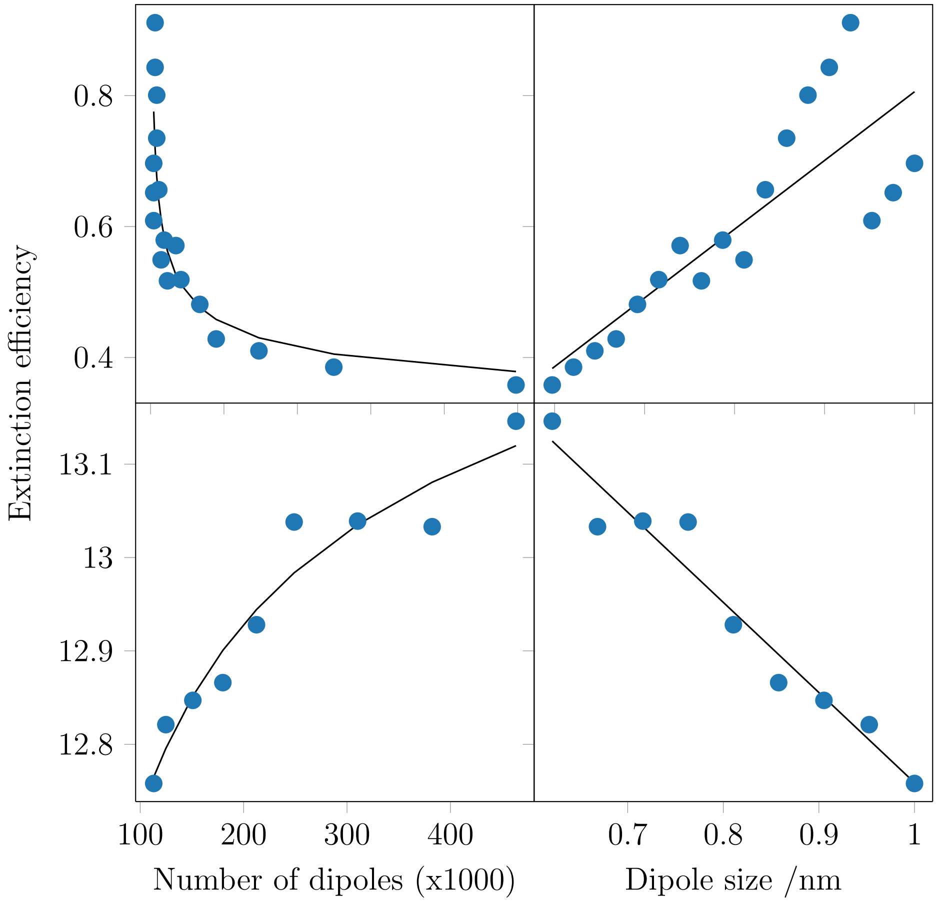


Figure 4.1: Extinction efficiencies at 420 nm wavelength for silver spheres calculated using DDScat for a range of dipole sizes. The top two plots show results for a 15 nm diameter sphere while the bottom two show the result for a 60 nm sphere

The extinction efficiency can be seen to converge as the number of dipoles increases; this is equivalent to decreasing the inter-dipole spacings. The 15 nm spheres shows a decaying converge while the 60 nm shows an increasing one. This is due to the fact that the plasmon peak for each one is on either side of the 420 nm wavelength that is studied. As the inter-dipole spacing decreases, both peaks shift to shorter wavelength, so for the 15 nm peak it will be shifting away from 420 nm and therefore the extinction will be decreasing, while the opposite will be true for the 60 nm sphere.

The relationship between the extinction efficiency and the inter-dipole spacing is linear, which suggests that the convergence follows a $N^{-\frac{1}{3}}$ relationship with the number of dipoles.

Equation 4.1 was fitted to the convergence data for all of the spheres to determine the parameter χ; an example of this is shown in figure 4.1.

$$y = a - b \cdot N^{-\frac{1}{\chi}} \tag{4.1}$$

The optimised fit parameter, χ, for all of the sphere sizes in shown in table 4.1.

As can be seen, for most of the spheres, the optimisation failed to find a suitable fit. This is evident from most of the fit parameters, χ, being set to the maximum allowed by the bounds, which was 6.

It is unclear why these fits failed as the optimised fits look good when plotted against the data. The only fits that should be expected to fail is that for the 45, 50 and 55 nm spheres; this is evident from the large confidence intervals. The reason for this is due to the fact that the convergence shape swaps from a decaying one to an increasing one over this range. The convergence for the

Width $/nm$	χ
5	$6.0 \pm nan$
10	5.57 ± 0.03
15	6.0 ± 0.8
20	6.0 ± 0.3
25	6 ± 1
30	2.3 ± 0.3
35	1.0 ± 0.3
40	6 ± 6
45	6 ± 75
50	6 ± 500
55	6 ± 26
60	4.5 ± 0.9

Table 4.1: Optimised convergence parameter χ along with 95% confidence intervals found from DDA calculations for silver spheres of various sizes.

Width /nm	Length /nm	χ
5	10	3 ± 1
5	15	5 ± 5
10	15	3.4 ± 0.5
10	20	2.8 ± 0.1
10	25	5.4 ± 0.6
15	20	2.7 ± 0.2
15	25	2.82 ± 0.04
15	30	2.88 ± 0.02

Table 4.2: Optimised convergence parameter χ along with 95% confidence intervals found from DDA calculations for nanorods of various sizes.

50 nm sphere is a flat line.

Figure 4.2 shows the extinction efficiency calculated using various dipole sizes for a silver nanorod with a width of 10 nm and a length of 20 nm. Like what was seen with the spheres, the extinction efficiencies converges as the number of dipole increases. This convergence in significantly more noticeable than was seen for the spheres, as the rod calculations were performed using a smaller dipole size and a larger number of dipoles.

Equation 4.1 was fit to this data for all of the nanorods. The optimised parameters, χ, found for all the nanorod sizes are shown in table 4.2. These values of χ are all around 3 apart from the 5 × 15 nm and 10 × 25 nm rods. The average of these parameters, excluding the two outliers mentioned, is:

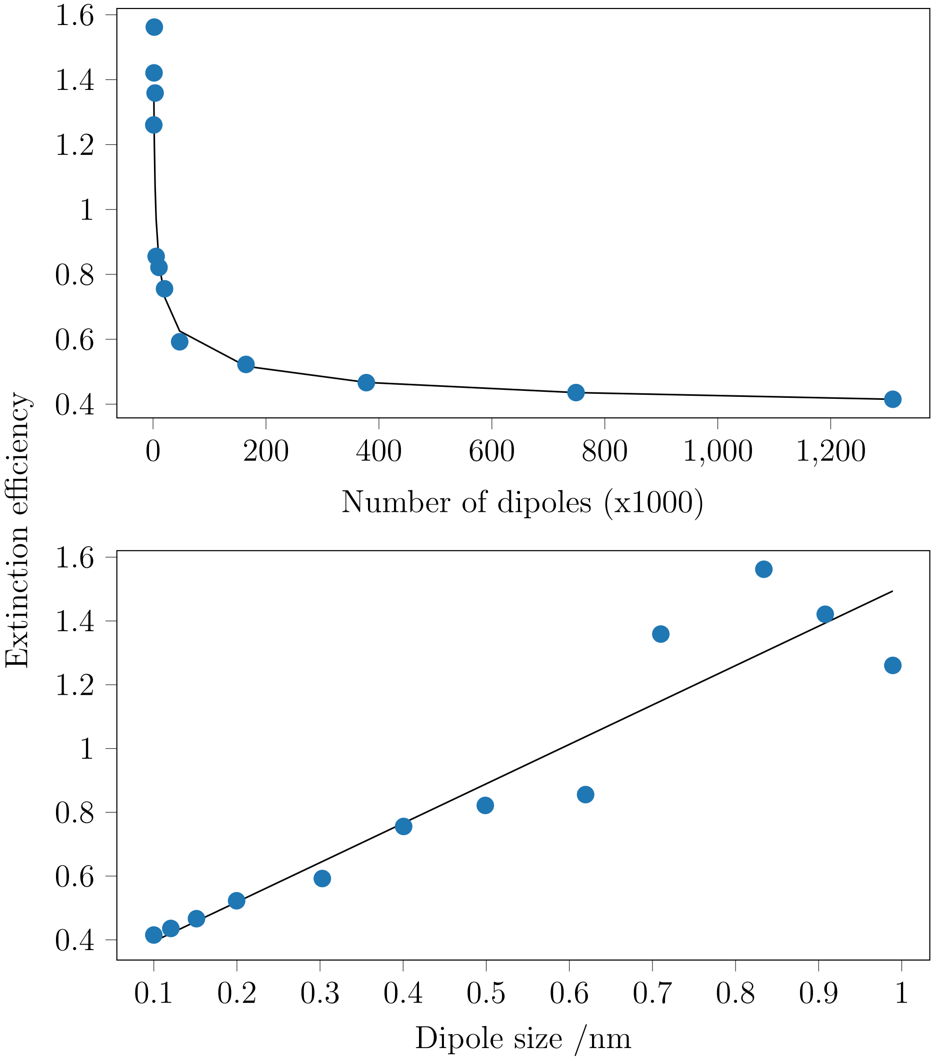


Figure 4.2: Extinction efficiencies calculated at a wavelength of 420 nm for a silver nanorod with a width of 10 nm and a length of 20 nm calculated using DDScat for a range of dipole sizes.

$$\chi_{rod} = 2.9 \pm 0.2 \tag{4.2}$$

4.1.1 Surface-dipole counting to estimate the convergence of the DDA results

In an attempt to determine the correction that is required for the DDA calculations of the nanorods, a Python/Rust script was used to build a model of the nanorods using cubic dipoles and then the number of dipoles at the surface were counted and compared to the total number of dipoles in the model. A dipole was considered to be part of the surface if it was less than $\frac{d}{2}$ away from the surface. This dipole counting was repeated with various inter-dipole distances, d.

This surface dipole counting was first performed on spherical targets as the DDA correction for this geometry is known and can therefore be used to validate the surface dipole counting method as a means to determine the required correction for the rods.

The surface-dipole counting was performed on spheres with diameters ranging from 5-30 nm with the smallest inter-dipole spacing being 0.01 nm. For the larger particles, the minimum inter-dipole spacing was limited to 0.02 nm as the dipole count for these spheres was above 2 billion and the Rust program was slowing down considerably.

The results of the dipole counting were incredibly stochastic at larger inter-dipole spacings; even at 0.5 nm, the optimised fit parameter was inaccurate for the smaller spheres. Nevertheless, the fit parameter converged on the expected value for all of the spheres as the inter-dipole spacing was reduced.

The relationship between the proportion of surface dipoles and the total number of dipoles is the same as seen in the DDA calculations for the smaller spheres, figure 4.1.

Equation 4.1 was fit to this data in order to determine the convergence parameter, χ. There was no obvious deviation of the data points from the optimised fit for any of the spheres.

At larger inter-dipole spacings, the variability of the data points is noticeably larger. Because of this, the optimisation is weighted towards the data points with a lower inter-dipole spacing (higher total dipole count). This is achieved by setting the standard deviation of each data point, called 'stigma' in the optimise function of the python script, to $\frac{1}{N}$; where N is the total number of dipoles for the corresponding data point.

The optimised values of χ are shown in figure 4.3.

At first glance, there still appears to be considerable deviation in the values of χ, however, this is significantly better than what was seen when the convergence fitting with performed using fewer dipole spacings. Even when the minimum dipole spacing was 0.05 nm, there was an obvious slope in the fit parameter plot. Adding extra data points, calculated at smaller dipole spacings, caused this plot to level out and provide a consistent value for χ. Adding the weighting to the optimise function also helped level out the fit parameter plot.

The variation in the optimised values for χ decrease as the diameter increases, this is because the total number of dipoles increases with an increase in diameter. A slight increase in the variability can be seen for diameters slightly larger than 15 nm; this is because the minimum inter-dipole spacing increases to 0.02 nm from 0.01 nm for the smaller spheres. Although, there is a significant deviations in χ, the 95% confidence interval for this parameter is smaller than the width of the line in figure 4.3.

The average value of χ for all the sphere diameters is:

$$\chi_{sphere} = 3.03 \pm 0.01 \tag{4.3}$$

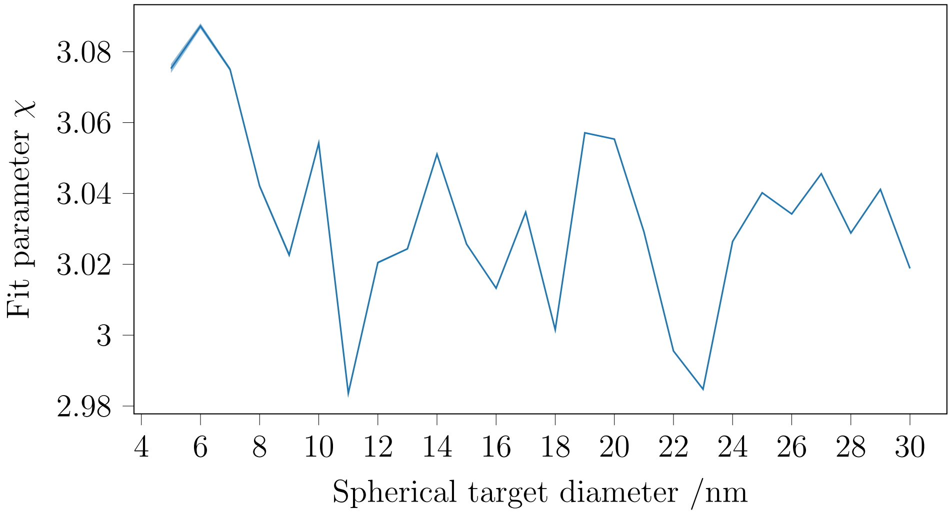


Figure 4.3: Optimised values of χ from equation 4.1 found for spheres with varying diameters for the surface-dipole counting method

To determine the DDA correction for nanorods, the surface-dipole counting was repeated for hemispheric-capped cylindrical rods, with widths ranging from 5 to 15 nm and lengths ranging from from (width) to (width+30) nm. The smallest inter-dipole spacing used was 0.01 nm for the rods with widths between 5 - 10 nm and 0.02 nm was used for the remaining rods. The maximum number of dipoles used was over 2 billion, just like what was used for the spheres.

Equation 4.1 was fit to the results. The optimised fit parameters, χ, are shown in figures 4.4 and 4.5 for the rods, either grouped by width or length.

The values of χ are shown to diverge as the aspect ratio of the rods increase and converge when the aspect ratio is 1. This convergence is expected as

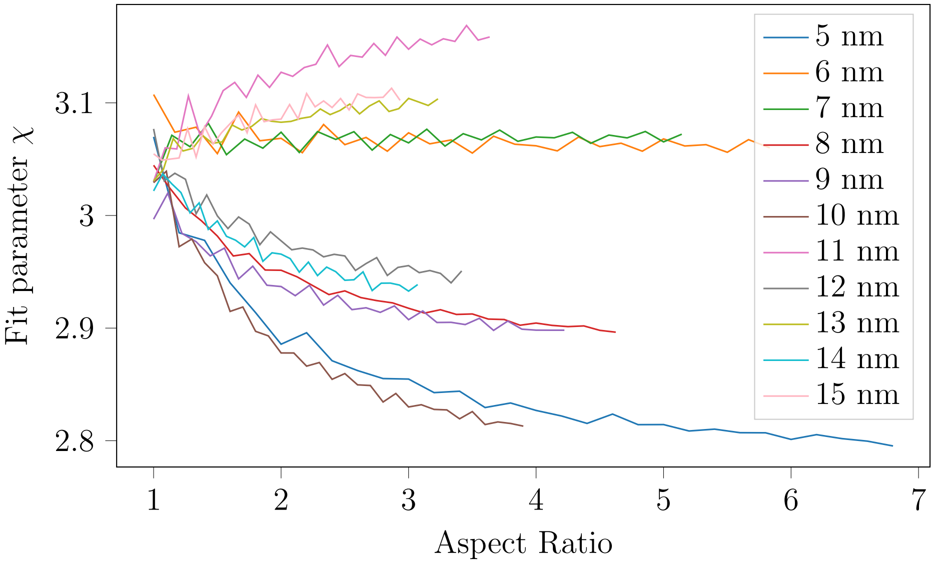


Figure 4.4: Optimised values of χ from equation 4.1 found for hemispheric-capped cylinders with varying diameters for the surface-dipole counting method. The various colours relate to rods which have the same width.

the geometry is a sphere for an aspect ratio of 1, and it has already been shown that χ is 3 for all the sphere diameters. In fact, the results should be exactly the same as for the spheres as the same Rust program is used for both geometries, with the length just being set to the same value as the width for the spheres.

The divergence that is seen as the aspect ratio increases is unexpected though. This is the opposite trend as what was seen for the spheres, where the divergence was largest for the smaller particle sizes and disappeared as the calculations were performed with smaller dipole spacings. However, this is still believed to be an artefact from the surface-counting rather than some relationship between χ and the aspect ratio, as the divergence changes dramatically as the minimum dipole spacing decreases.

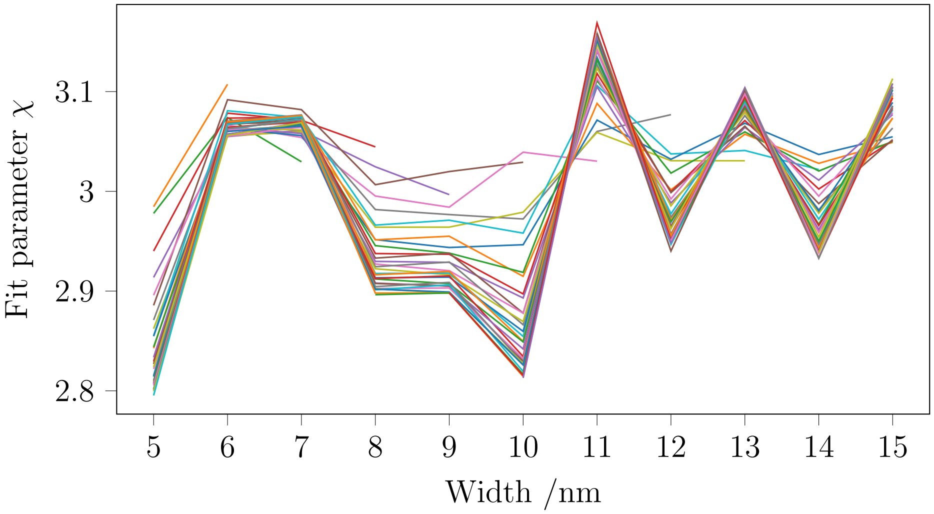


Figure 4.5: Optimised values of χ from equation 4.1 found for hemispheric-capped cylinders with varying diameters for the surface-dipole counting method. The data is grouped by length of the rods

Interestingly, when the data points are grouped by the length of the rods, all the lines follow the same trend and for a given rod width, there is little variation as the length changes. This can be seen in figure 4.5.

The fit parameter χ was averaged for all of the rods and was found to be:

$$\chi_{rod} = 3.00 \pm 0.01 \tag{4.4}$$

This surface-dipole counting procedure was repeated for the nanorods, however, instead of counting all of the surface dipoles together, the surface dipoles of the end caps and the cylinder section were counted separately; The fit parameter, χ, was then calculated for each of these regions.

The optimised fit parameter, χ, for the various nanorod sizes are shown in figure 4.6.

The top plot show the χ values found using only the surface dipoles from the end caps. This plot shows that there is no relationship between the aspect ratio and χ. This plot is very similar to that found for the spheres.

The average value of χ from this plot, along with the 95% confidence interval, is:

$$\chi_{rod} = 3.044 \pm 0.003 \tag{4.5}$$

The bottom plot of figure 4.6 shows the values of χ found using only the surface dipoles from the cylinder section of the rod geometry. As can be seen from this plot, χ settles on a single value as the aspect ratio increases for a given width. When the aspect ratio is near 1 there is an obvious increase in the variance of χ; this is due to the fact that there are very few surface dipoles from the cylinder section when the aspect ratio is small.

The average fit parameter for the cylinder plots, along with the 95% confidence interval, are shown in equation 4.5 and 4.6 respectively.

$$\chi_{rod} = 2.97 \pm 0.02 \tag{4.6}$$

The values of χ found when counting all of the surface dipoles is seen to be a combination of these two plots. When the aspect ratio is near 1, the majority of the surface dipoles come from the end cap, and so the the values of χ shown in figure 4.4 converge on what is seen in the top plot of figure 4.6. As the aspect ratio increases, the number of surface dipoles from the cylinder section becomes more significant and so the values of χ converge on

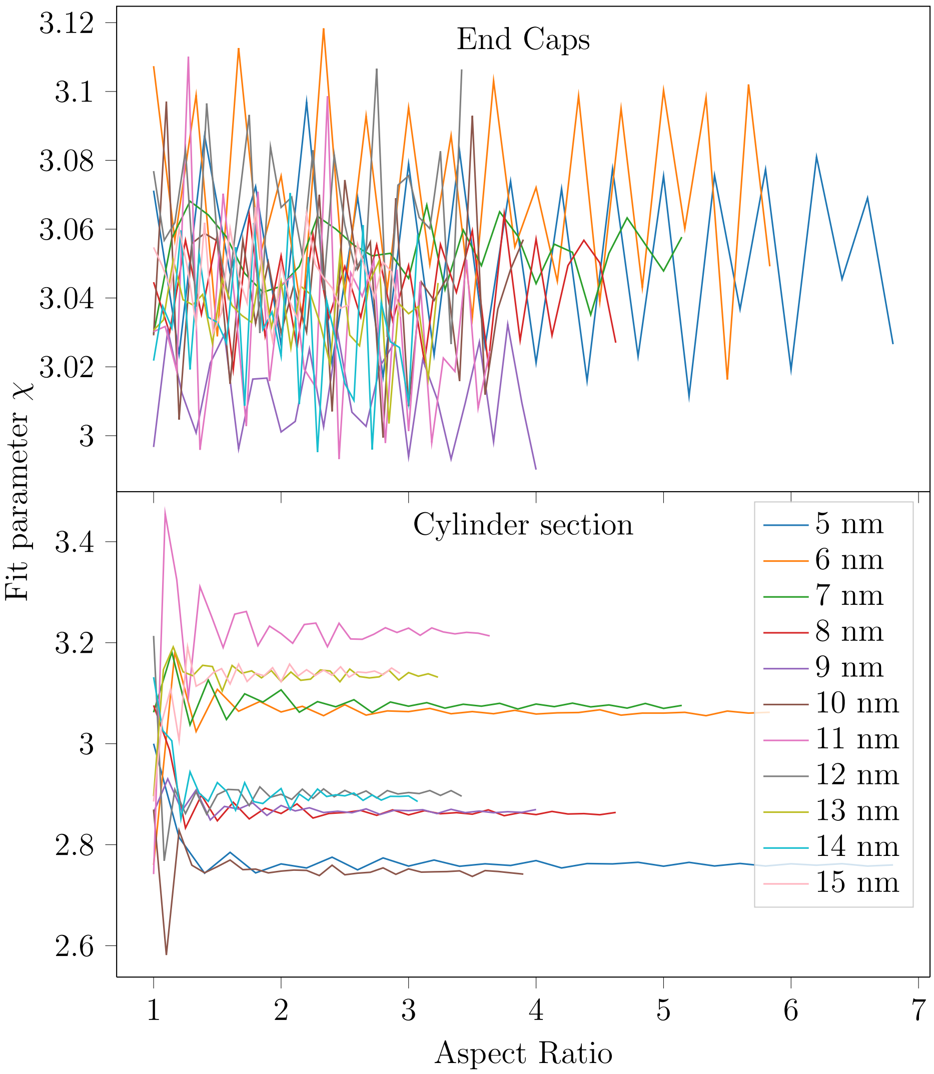


Figure 4.6: Optimised values of χ from equation 4.1 found for the surface-dipoles of the cylinder section of hemispheric-capped cylinders with varying diameters. The data is grouped by width of the rods

what is seen in the bottom plot of figure 4.6.

4.2 Computational attenuation spectra for spherical particles

4.2.1 Mie theory results

The attenuation spectra of spherical silver nanoparticles, calculated using PyMieScatt, is shown in figure 4.7. For small spheres, the absorption spectrum consists of only a single absorption peak, which is associated with the dipole mode of the plasmon resonance. This peak increases in amplitude and shifts to longer wavelength as the size of the sphere increases. As the size of the particle continues to increase, a second absorption peak appears around 370 nm; this peak is associated with the quadrupole mode of the plasmon resonance. After the appearance of the second peak, the amplitude of the original peak starts to decreases. An isosbestic point is seen between the two peaks indicating that the quadrupole peak is growing at the expense of the dipole peak.

Only one peak is present in the spectrum of scattering efficiency, however, there is an isosbestic point in the spectrum which may indicate that a quadrupole peak is present but not as prominent as the one seen in the absorption spectra. This peak increases in amplitude as the particle diameter increase and reaches a maximum efficiency for spheres with a diameter of $45nm$. The position of the scattering peak shifts to longer wavelength as the particle diameter increases, and the peak position is coincident with the dominant absorption peak.

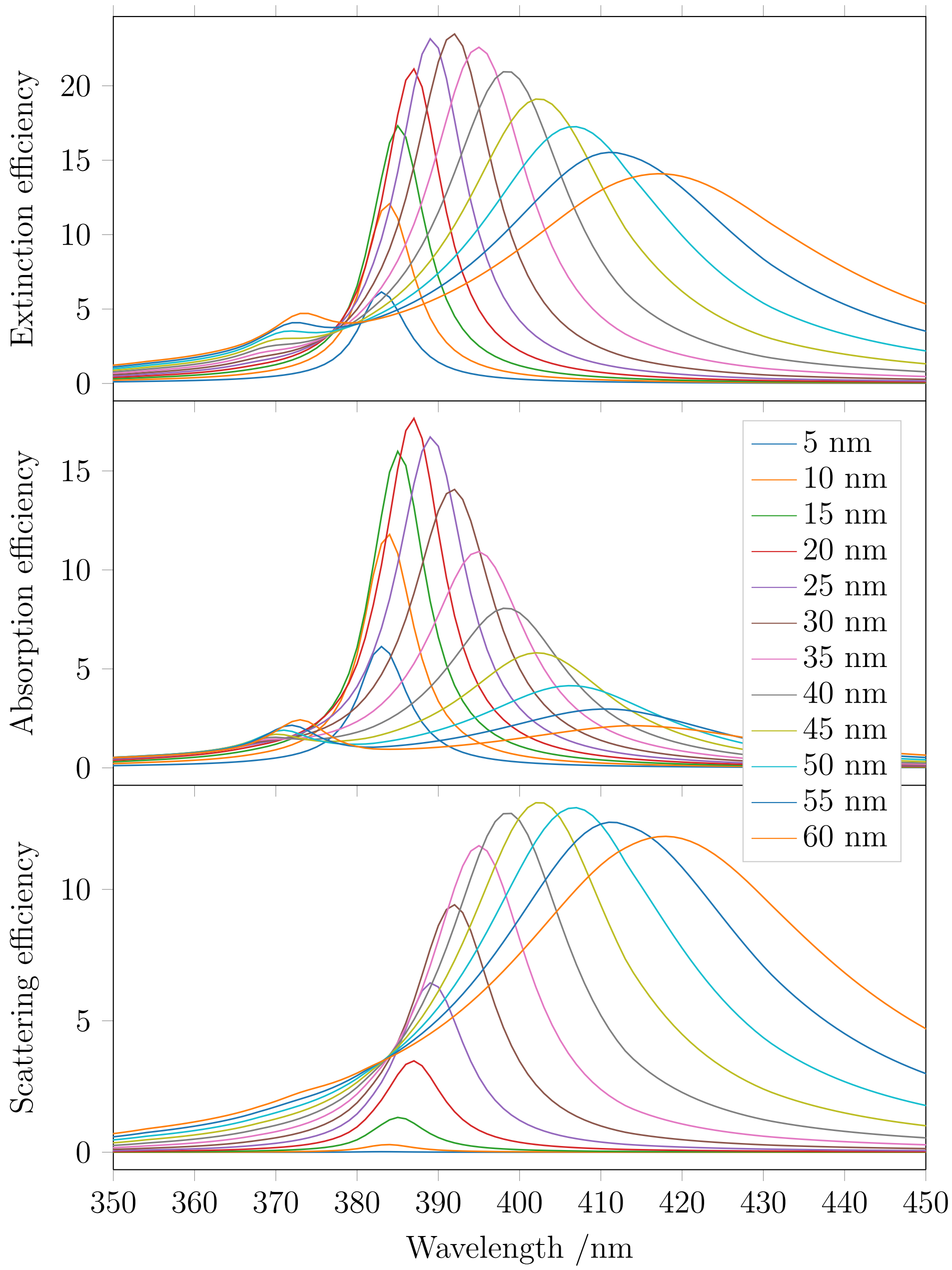


Figure 4.7: Extinction, absorption, and scattering efficiencies, calculated using PyMieScatt, of silver spheres with various diameters

Two wavelength-domain pseudo-Voigt functions were fitted to the spectra for all the attenuation modes to determine the parameters of each peak. An example of one of these fits is shown in figure 4.8 and the resulting peak parameters for the extinction peaks are shown in figure 4.9.

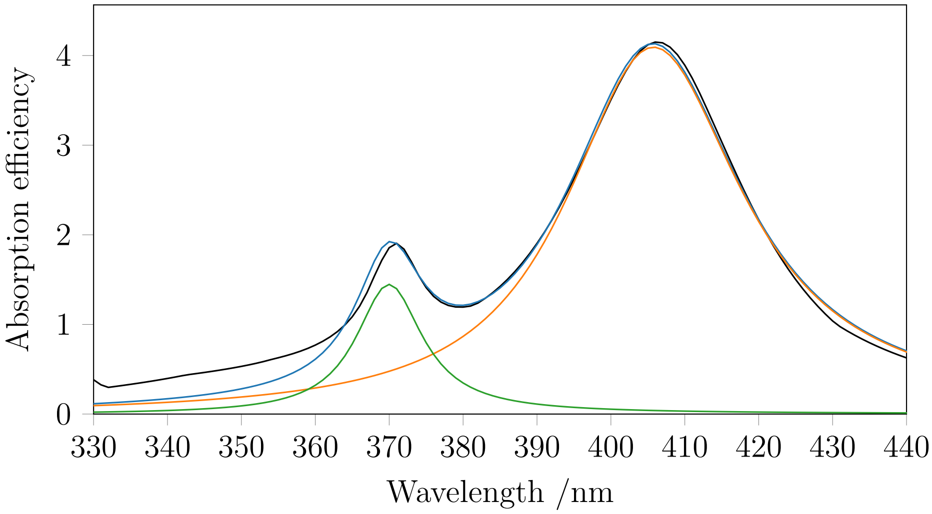


Figure 4.8: Pseudo-Voigt functions fitted to the plasmon peaks in the absorption efficiency spectrum for a silver sphere with a diameter of 50 nm.

The position of the primary peak appears to be proportional to the square of the particle diameter, and is the same for all of the attenuation modes.

The width of the primary peak has the same relationship with the particle diameter as the peak position does. The width of the secondary peak is constant because it is limited by the upper bound of the optimisation function. If the bounds are increased, the width will increase to cover the absorption at lower wavelengths which is not related to the plasmon resonance. This can be seen in figure 4.8.

The η parameter has values around one for all attenuation modes and particle diameters. This suggests that all of the peaks have a Lorentzian profile.

The η value for the secondary peak is exactly one in all of the diameters. This is also an artefact due to the absorption at lower wavelengths which is not related to the plasmon resonance. This occurs because the extended tails of the Lorentzian function cover more of the absorption at lower wavelengths than a Gaussian peak would. Despite this, the secondary peak is still believed to be a Lorentzian.

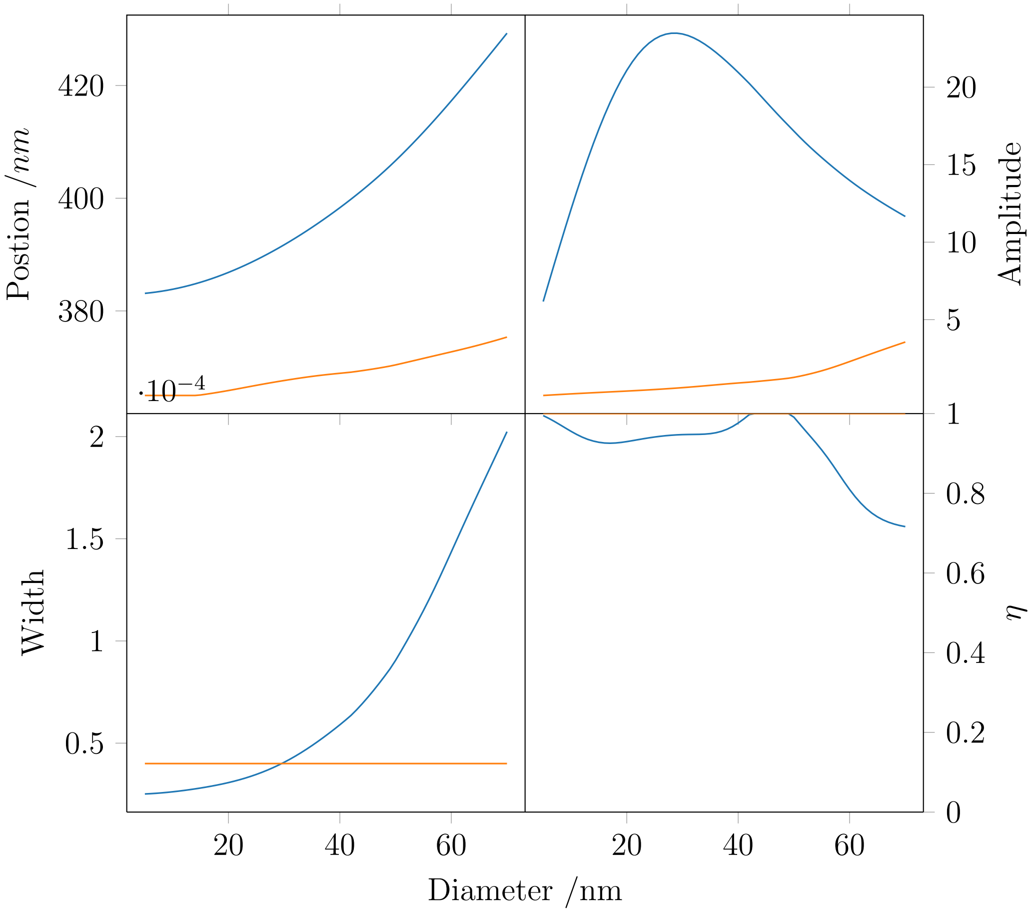


Figure 4.9: Parameters for the wavelength-domain pseudo-Voigt peaks fitted to the extinction efficiency spectra of silver spheres of various diameters calculated using PyMieScatt.

4.2.2 Discrete-dipole approximation results

Figures 4.10 and 4.11 show the extinction, absorption, and scattering spectra of 20 and 50 nm silver spheres calculated using the DDA. The corrected spectra are shown along with the spectra obtained using Mie theory (See section 4.2.1).

All spectra show a primary plasmon with a secondary quadrupole plasmon peak visible in the absorption spectrum for the 50 nm sphere. As the inter-dipole spacing is decreased, the primary plasmon peak shifts to shorter wavelengths; this peak shifts even further after the correction is applied. Importantly, the plasmon peak position of the primary peak matches that of the Mie spectrum after the correction is applied.

The amplitude of the absorption peak for the 20 nm sphere is lower than that of the Mie peak, however the 95% confidence intervals of the corrected spectrum do overlap with the Mie peaks. As the particle size increased, the amplitude of the absorption peak increased relative to that of the Mie peak. For particles with a diameter larger than 30 nm, the amplitude of the absorption peak for the DDA was larger than that of the Mie peak. At the same time the confidence intervals decreased and eventually no longer overlapped with the Mie peaks. However, the extinction peaks do agree as the DDA underestimates the scattering peak while overestimating the absorption peak.

For smaller particle diameters, the scattering peak in the DDA is significantly lower in amplitude compared to the corresponding Mie peak. This does not result in a disagreement in the extinction spectra, however, as the absorption peak dominates the attenuation for these diameters. Somewhere between the 30 and 40 nm spheres, the scattering becomes the dominate attenuation mode. For diameters larger than 30 nm, the scattering peak for the DDA does agree with the Mie peak.

A quadrupole plasmon peak appears in the spectra for the spheres with a diameter greater than 20 *nm*; this agrees with what was observed in the Mie spectra.

The corrected extinction spectra for all of the sphere diameters are shown in figure 4.12.

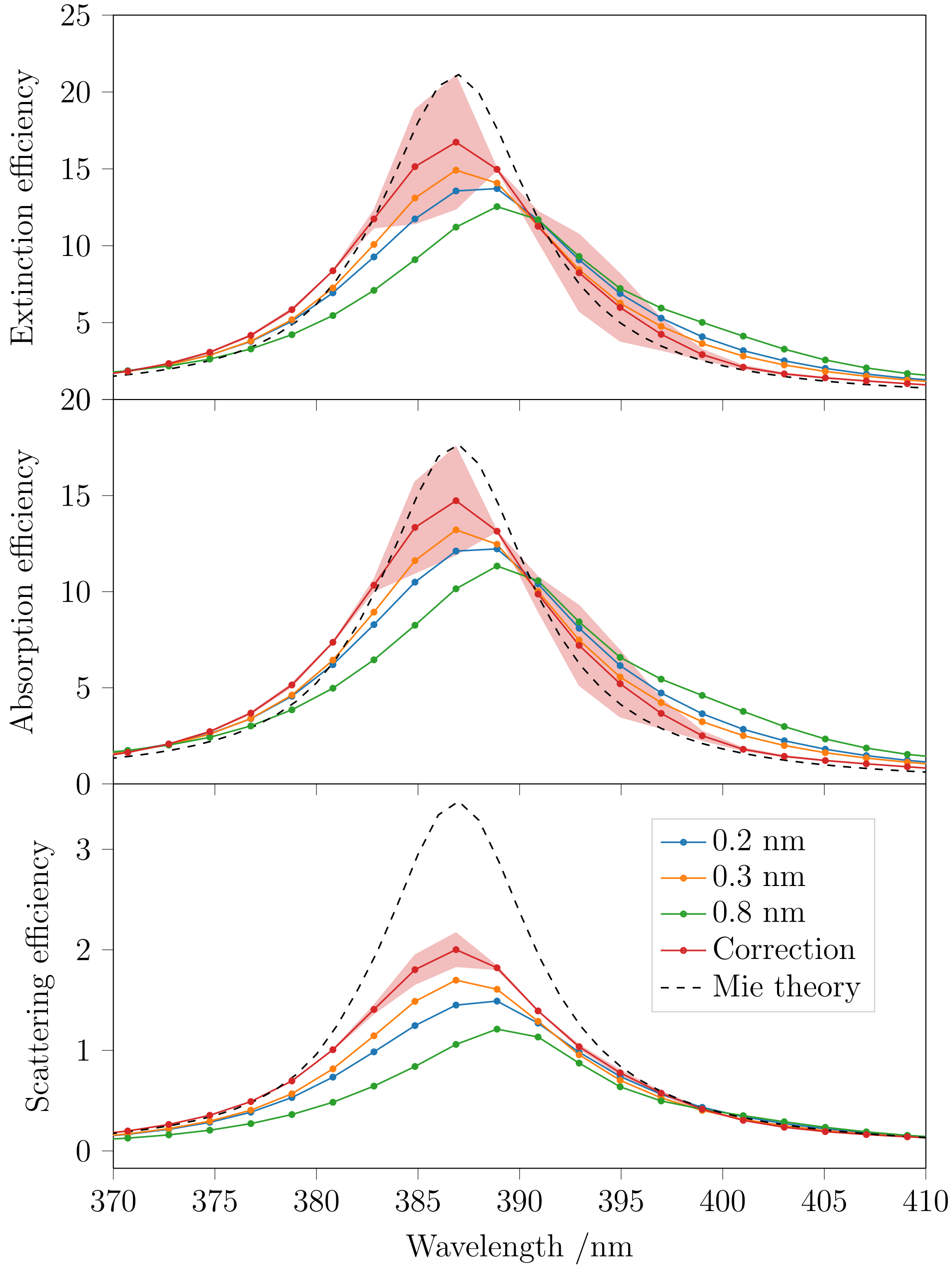


Figure 4.10: Extinction, absorption, and scattering efficiencies for a 20 nm silver sphere calculated using DDScat for a range of inter-dipole spacing, *d*, along with the corrected spectrum. The Mie theory results are shown with the dashed line.

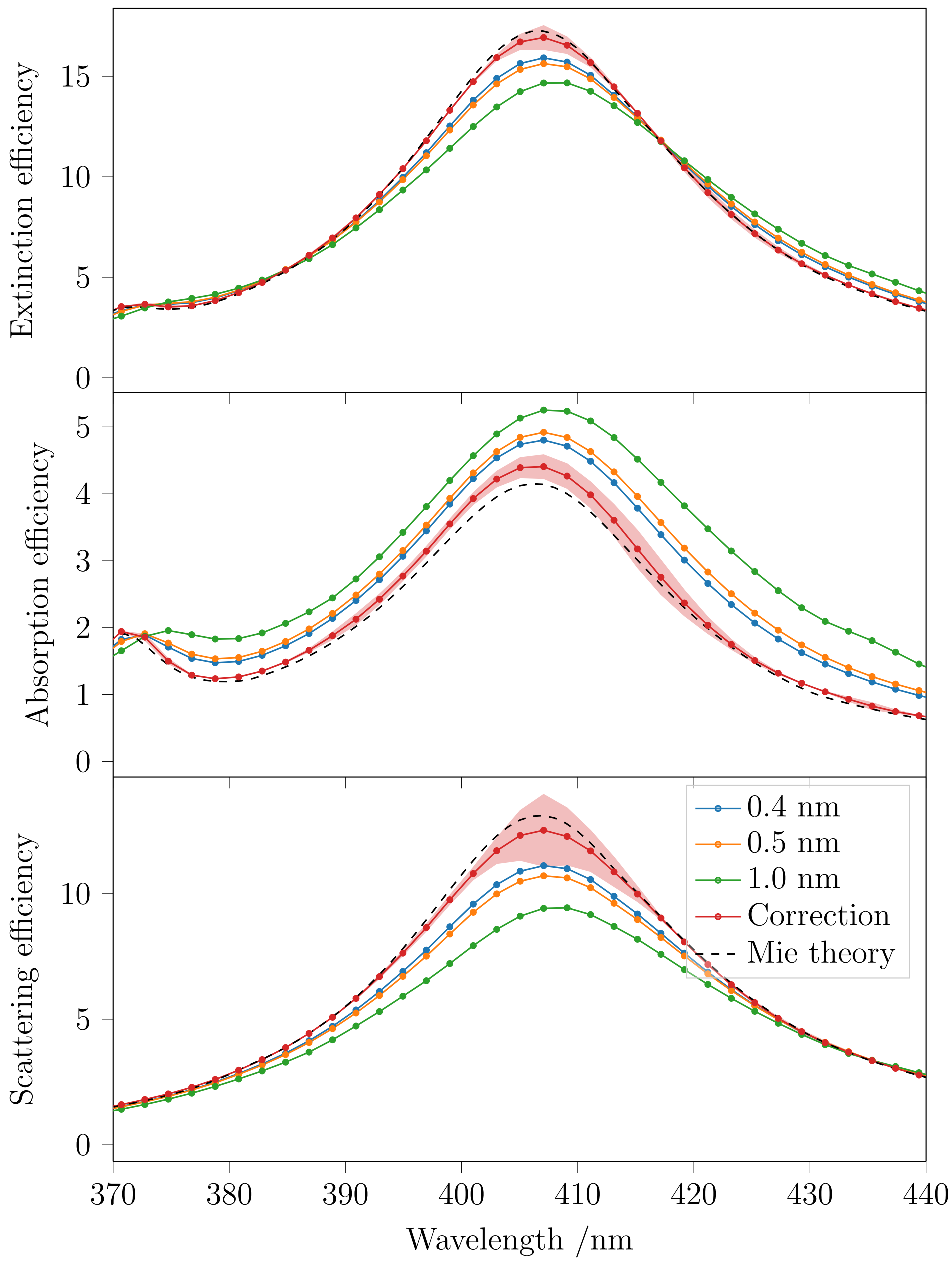


Figure 4.11: Extinction, absorption, and scattering efficiencies for a 50 nm silver sphere calculated using DDScat for a range of dipole densities along with the corrected spectrum. The Mie theory results are shown with the dashed line.

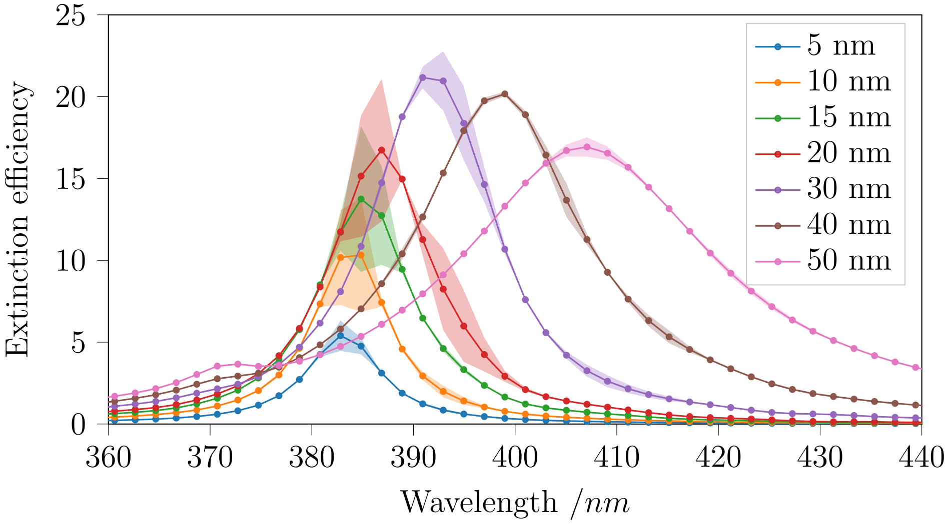


Figure 4.12: Corrected extinction spectra, along with 95% confidence bands, for silver spheres calculated for a range of diameters using the DDA

Spectral peak fitting of the DDA results

Wavelength-domain Lorentzian functions were fitted to the corrected extinction, absorption, and scattering spectra which were obtained from the DDScat calculations for spherical silver particles.

Figure 4.13 shows the position of the various attenuation peaks shifting to longer wavelengths as the size of the nanoparticle increase. There is little variation in the peak position between the attenuation modes, with a slight deviation appearing for larger particle sizes. A similar trend is seen for the fitted peak widths. There is, however, a vast difference in the peak amplitudes for these modes, with absorption being the dominant process for smaller particles with scattering taking over as the particle size increases. This can be seen in figure 4.14.

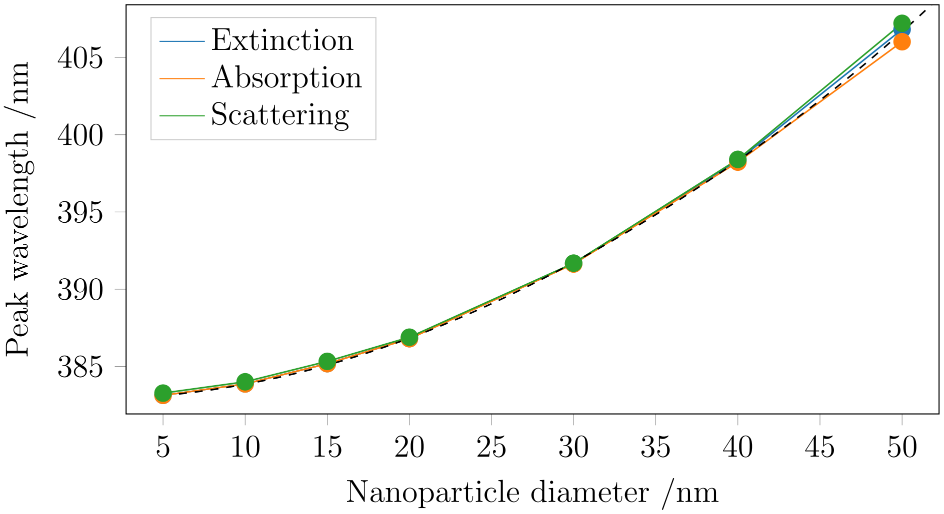


Figure 4.13: Peak position of the primary plasmon peak fitted to the DDScat calculations for silver spheres of various sizes for all three attenuation modes. The dashed line shows the peak positions obtained via the Mie theory calculations

The peak positions show a strong agreement with the Mie theory results.

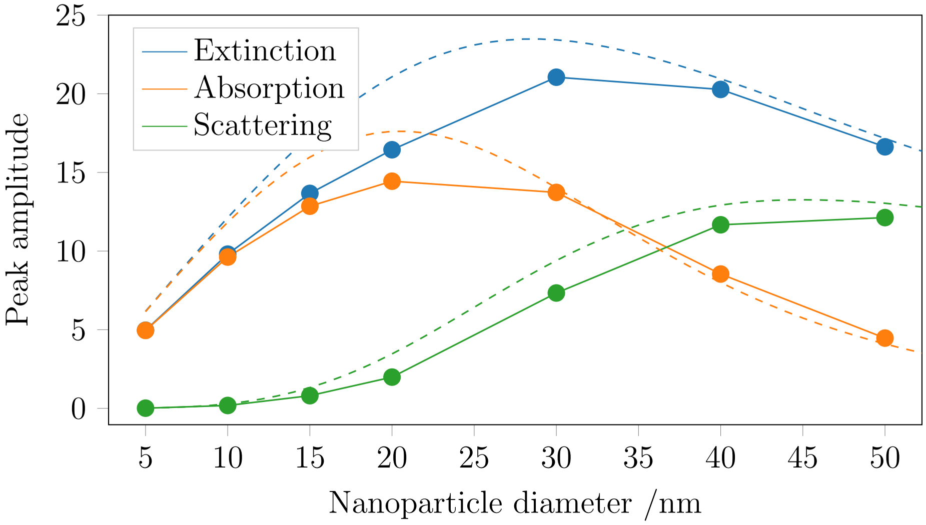


Figure 4.14: Peak amplitude of the wavelength-domain pseudo-Voigt function fitted to the primary peak in the DDScat calculations for silver spheres of various sizes. The dashed lines show the peak amplitude for the amplitudes of the attenuations obtained via the Mie theory calculations.

The average deviation of the longitudinal plasmon peak position from the Mie theory results was less than 0.06 nm. The largest deviation was for the 50 nm sphere which was still only 0.15 nm.

There is a slight disagreement with the peak amplitudes and widths. With the DDA calculations underestimating the peak amplitudes and overestimating the widths. Interestingly, the peak areas do agree with the Mie theory results despite the disagreement seen for the amplitudes and widths. This can be seen in figure 4.15.

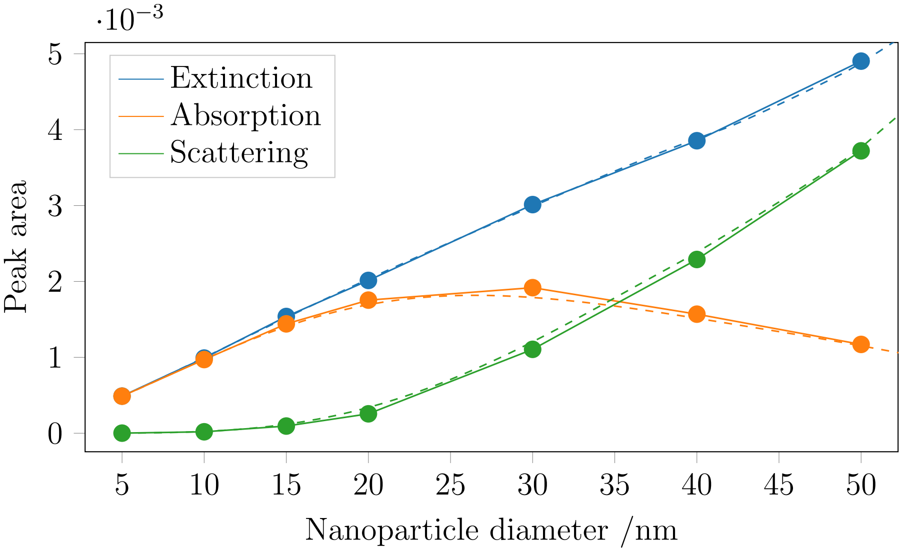


Figure 4.15: Peak area of the primary plasmon peak found in the DDScat calculations for silver spheres of various sizes. The dashed lines show the peak areas for the amplitudes of the attenuations obtained via the Mie theory calculations.

4.3 Computational attenuation spectra for rod-like particles

4.3.1 Discrete-dipole approximation results

Figure 4.16 shows the corrected extinction spectra of hemispheric-capped cylindrical silver nanorods for a range of length and a width of 30 nm, which were calculated using DDScat.

Two main attenuation peaks are apparent for the silver nanorods. One of these peaks shows a strong dependence on the length of the nanorods and shifts to longer wavelengths as the length of the rod increases. The position of this peak also show a dependence on the width of the nanorod, with wider rods resulting in the peak position shifting the shorter wavelength.

When the spectrum is calculated for polarised light, the peak at longer wavelength only appears when the polarisation of the light is parallel to the long axis of the nanorod. This suggests that this peak is associated with the longitudinal plasmon mode of the nanorod. The two peaks are referred to as the transverse and longitudinal plasmon peaks.

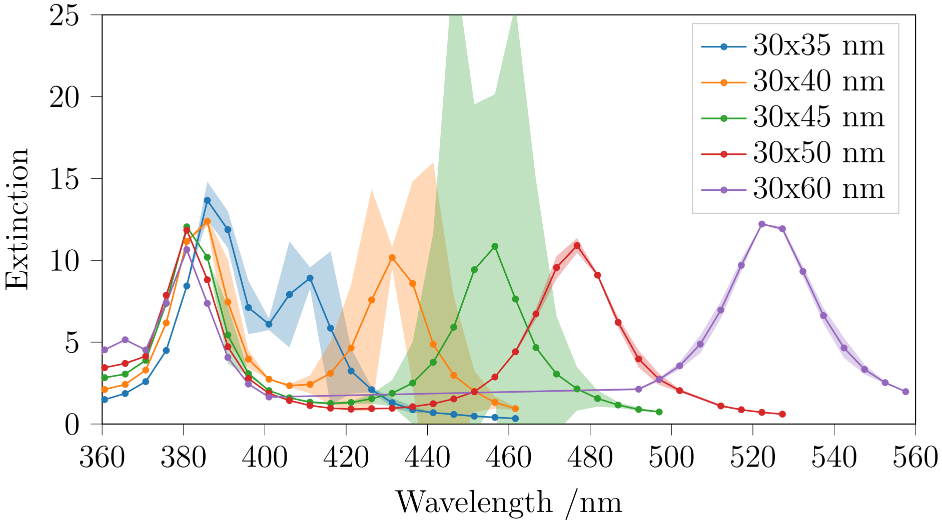


Figure 4.16: Corrected extinction spectra calculated using DDSCAT for nanorods with a range of lengths and a width of 30 nm

Spectral peak fitting of the nanorod DDA results

Wavelength-domain pseudo Voigt functions were fit to the attenuation efficiencies which were obtained from DDScat calculations. The fit parameter η was found to be close to 1 for all rod sizes, suggesting the peaks have a Lorentzian profile. To reduce the number of fit parameters, wavelength-domain Lorentzian functions were then fitted to the spectra.

As shown in figure 4.17, the position of the longitudinal plasmon peak is the same for all attenuation modes and is proportional to the aspect ratio of the nanorod. A linear regression was performed to find the relationship between the peak positions and aspect ratio. The relationship for all three attenuation modes was found to be:

$$\left.\begin{aligned}\lambda_{ext} &= (113 \pm 10) \times A \ + (277 \pm 20) \\ \lambda_{abs} &= (113 \pm 10) \times A \ + (277 \pm 21) \\ \lambda_{scat} &= (113 \pm 10) \times A \ + (277 \pm 21)\end{aligned}\right. \qquad [\text{nm}] \tag{4.7}$$

where λ is the peak position of the corresponding attenuation mode, and A is the aspect ratio of the nanorod. 95% confidence intervals are included for the fit parameters.

When the data points are grouped by the width of the associated rod, a difference in the linear fit parameters are observed. However, because of the limited number of data points, it is impossible to determine this relationship as the confidence intervals all overlap. The width dependence of these parameters is shown in figure 4.18

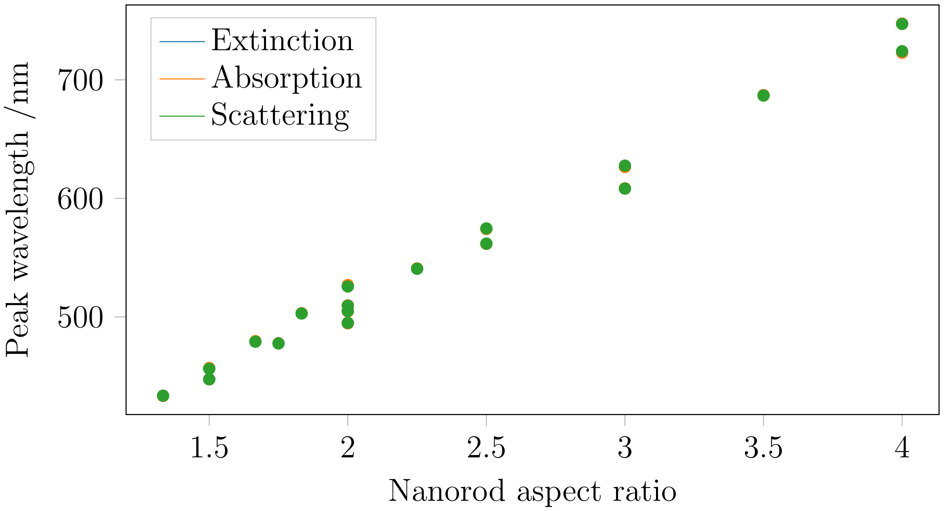


Figure 4.17: Position of the longitudinal plasmon peak, for all attenuation modes, of silver nanorods with various aspect ratios

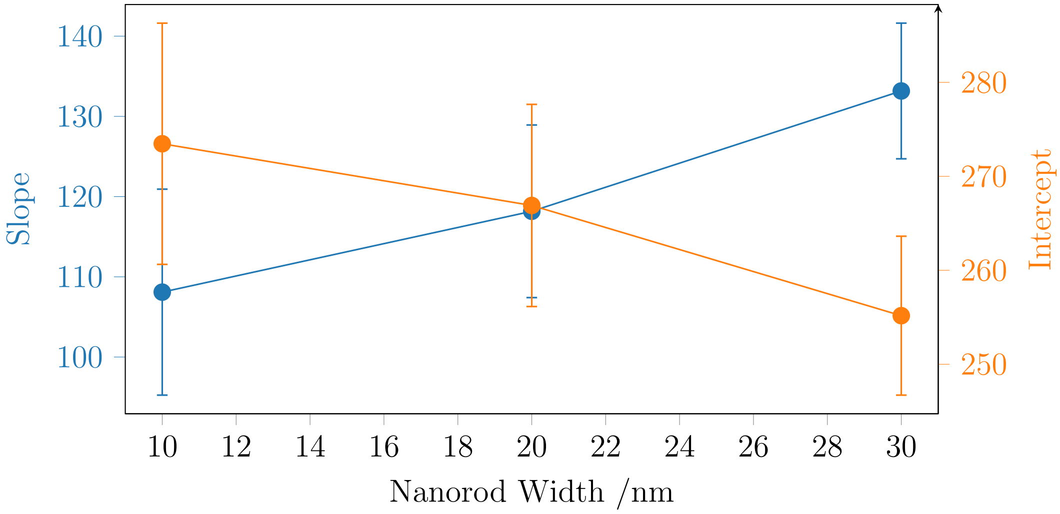


Figure 4.18: Linear fit parameters, with 95% confidence intervals, relating the position of the longitudinal plasmon peak to the aspect ratio of nanorods with various widths

4.3.2 Gan's Theory results

Figure 4.19 show the extinction spectra calculated using Gan's theory for a range of aspect ratios. Like the DDA calculations, there is a transverse and longitudinal plasmon peak and these two peaks show the same response to an increase in aspect ratio. with the longitudinal plasmon peak shifting to longer wavelengths and the transverse peak shifting to shorter wavelengths.

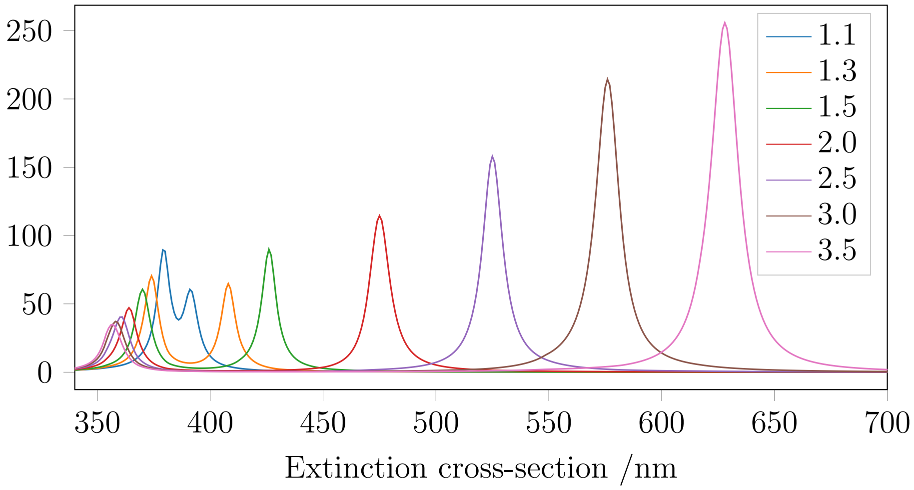


Figure 4.19: Extinction cross-section spectra calculated using Gan's theory for a range of particle aspect ratios

Wavelength-domain pseudo-Voigt function are fit to these extinction spectra to characterise the plasmon peaks. Like what was observed with the DDA results, the longitudinal plasmon peak displays a linear relationship with the aspect ratio. A linear fit was applied to this plot and the relationship was found to be:

$$\lambda = (103.1 \pm 0.3)A \; + (267.3 \pm 0.8) \qquad [\mathrm{nm}] \qquad (4.8)$$

Although the confidence intervals overlap between this equation and the one found using the DDA(equation 4.7), the two clearly do not agree when viewed together on a plot; this can be seen in figure 4.20. This disagreement is likely due to the different geometries used in the Gan's and DDA calculations.

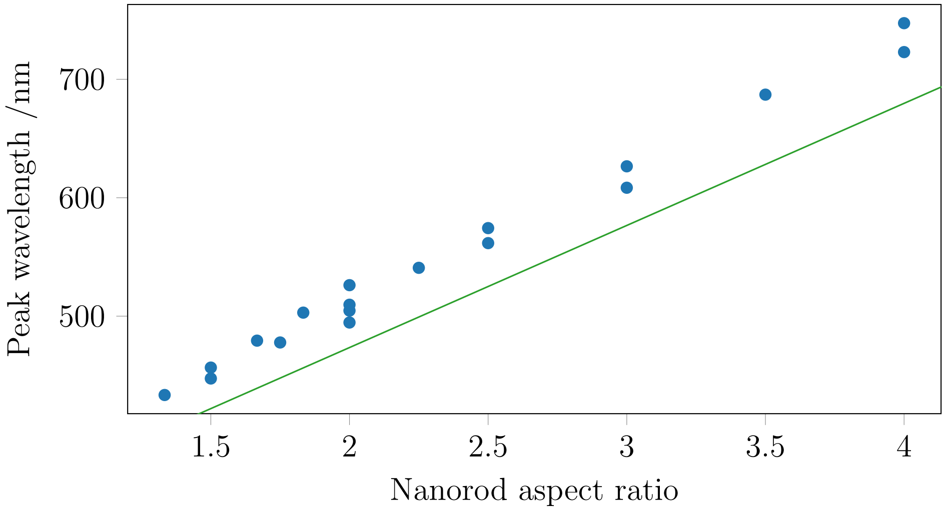


Figure 4.20: Liner fit of plasmon peak position found using Gan's theory (green line) along with the plasmon peak positions found using DDSCAT (blue dots) for particles with various aspect ratios

Because the relationship of the peak position is linear for both the DDA and Gan's theory calculations, the aspect ratios of the two geometries are also expected to have a linear relationship.

The two calculations are also expected to agree for an aspect ratio of 1, as both geometries converge on a sphere. To find the relationship between these two geometries, the aspect ratios were first shifted by negative one to make the intercepts equal. Then the slopes from the two equations, 2.17 and 2.22, were used to calculate the scale factor before finally shifting the aspect ratios back by one.

The relationship between the two aspect ratios was found to be:

$$A_D = \frac{m_D}{m_G}\left(A_G - 1\right) + 1 \qquad (4.9)$$

where A is the aspect ratio and m is the slope of the associated results.

The scale factor that describes the relationship between the aspect ratios of these two geometries was found to be:

$$\frac{m_D}{m_G} = 0.91 \qquad (4.10)$$

Equation 4.9 was used to calculate the effective DDA aspect ratio for the prolate spheroids. The relationship between the peak position and the effective DDA aspect ratio, A', was found to be:

$$\lambda = (113.1 \pm 0.3) \times A' + (257.3 \pm 0.8) \qquad [\mathrm{nm}] \quad (4.11)$$

Figure 4.21 show the extinction cross section spectra calculated for a range of refractive indices. As the refractive index of the surrounding medium increases, both of the plasmon peaks are red shifted. The amplitude of the two plasmon peaks also depends on the refractive index of the solution. There is a minimum for amplitude of the longitudinal plasmon peak when the refractive index is approximately 1.6. While longitudinal plasmon peak is at a minimum for this refractive index, the amplitude of the transverse peak is at a maximum.

The relationship between the longitudinal peak position and the aspect ratio, A', remains linear for all refractive indices examined. Linear regression was used to determine the slope and intercept of this relationship for each

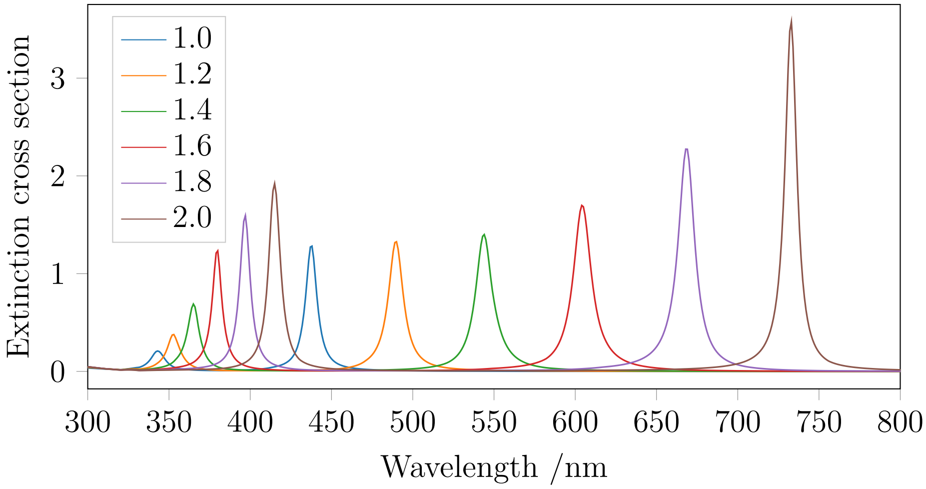


Figure 4.21: Extinction cross section spectra calculated using Gan's theory for a prolate spheroid with an aspect ratio of 2.5 for a range of solution refractive indices

spectrum. Figure 4.22 shows the relationship between these parameters and the refractive index of the surrounding medium.

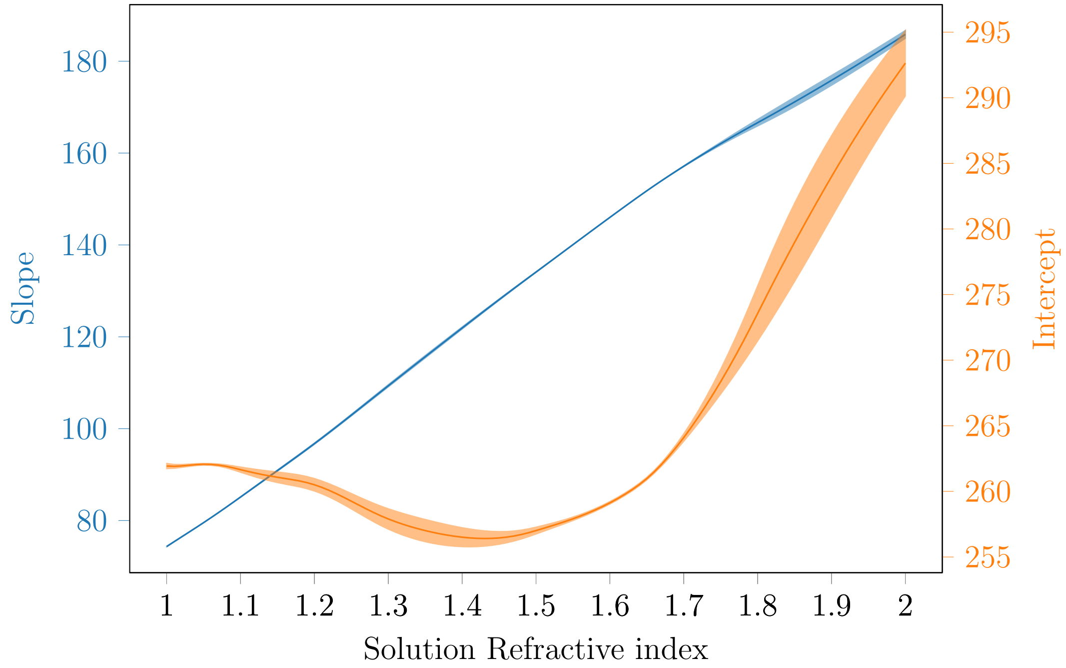


Figure 4.22: Linear fit parameters describing the relationship between the longitudinal plasmon peak position and the aspect ratio of silver nanoparticles calculated using Gan's theory. To make the results agree with the DDA, the aspect ratios have been corrected to compensate for the different geometries used in the two methods.

Chapter 5

Preparation and characterisation of silver nanorods

This chapter summarises the experimental results that were obtained during the preparation of the silver nanorod solutions.

The first part of this chapter deal with characterising the nanorod preparations as well as the seed solutions. UV-Visible extinction spectroscopy was used to characterise the plasmon peaks found in the nanorod preparations and was used, along with centrifugation, to determine whether or not the nanoparticles were aggregating. TEM images of the nanorod preparations were obtained to determine the shape and size of the nanoparticles found in the various preparation.

The last part of this chapter deals with the time-resolved UV-Visible extinction spectra which was collected during the growth of the nanorods.

5.1 Preparation of silver nanorods

5.1.1 Preparation of silver seeds

The growth of the silver seed particle was evident from the immediate colour change of the solution after the addition of the sodium borohydride. The colour of the solution was bright yellow colour which darkened slightly over the 2 hours that the solution was left before it could be used in the nanorod preparation.

In most of the seed preparations, a single plasmon peak was observed in the UV-Visible extinction spectrum at around $380nm$. Occasionally, a shoulder was observed on the longer wavelength side of this primary plasmon peak, which is possibly an indication of unexpected particle geometries present. This can be seen for the green line in figure 5.1.

The amplitude of the primary plasmon peak varied between all the seed preparations along with a slight variation in the position. These variations seen in the extinction spectra of the silver seed particles were not expected, as the exact same methodology was used for each preparation; the only difference being that they were prepared on different days.

The silver seed particles were too small to be observed with TEM.

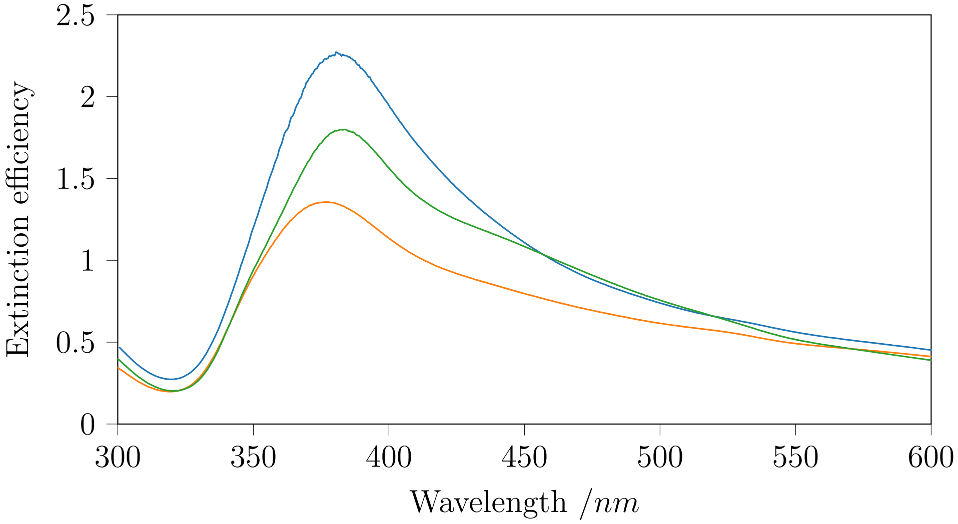


Figure 5.1: UV-Visible extinction spectra of silver seeds solutions used in the preparation of nanorods.

5.1.2 Growth of silver nanorods

Like the preparation of the silver seed particles, the regrowth of the seed particles is evident from the colour change in the solution. The colour change in the regrowth step is more drastic than was seen for the seed preparation, suggesting that the particles formed are no longer spherical.

The colour of the solution goes through a series of changes starting at the initial yellow colour of the seed solution before changing to orange, then red, then blue before finally ending up at a light green. The colour change starts immediately after the addition of the sodium hydroxide, and the final colour of the nanorod solution was dependant on the volume of the silver seed solution which was used in the preparation, with larger volumes resulting in the colour progress halting earlier in the colour sequence; this corresponds to a smaller γ value. Figure 5.2 shows the resulting solutions for nanorods prepared with various volumes of the silver seed solution.

5.1.3 Removing excess CTAB from the nanorod preparations using centrifugation

After the nanorods were centrifuged after they were prepared and the resulting pellet was redispersed in Milli-Q water to remove any excess CTAB that would otherwise crystallise overtime. The centrifugation step would also remove any smaller particles that did not grow into nanorods.

The UV-Visible extinction spectrum was recorded for the redispersed pellet, as well as the supernatant, to study the effect the centrifugation has on the plasmon peaks. A comparison of these spectra for one of the nanorod preparations can be seen in figure 5.3. These spectra show a obvious shift in the longitudinal plasmon peak resulting from the centrifugation; these shifts were not as obvious for the other nanorod preparations which had lower

γ values. To determine whether there was any change occurring with the plasmon peaks for these samples, wavelength-domain pseudo-Voigt functions were fit to these spectra.

For the preparations with low γ values, the plasmon peaks don't show any significant shift in their position after centrifugation. However, the preparations with γ values of 8 or higher do show an obvious red shift for the longitudinal peak for the pellet and a slight blue shift in the supernatant. This is shown in figure 5.4 . This shift is especially dramatic for the preparation with a γ value of 16 as there is no obvious longitudinal peak present at all before centrifugation and it only appears in the pellet afterwards.

The amplitude of the plasmon peaks in both the supernatant and pellet are reduced compared to the original extinction spectra. This is expected as both solutions contain a lower concentration of nanoparticles compared to the original, however, the sum of the supernatant and pellet spectra was also lower than the original spectrum. This was partly due to the shifts of the plasmon peaks but also could be due to errors in the concentrations of the pellet solutions which were redispersed in Milli-Q water after centrifugation. The amplitudes of both the transverse and longitudinal peaks are shown in figure 5.5. The most interesting observation from these plots is that, for a given centrifuge speed, the peak amplitudes in the pellet solutions increase as the γ value increases., meanwhile the plasmon peaks in the supernatant decrease. This suggests that the nanoparticles in the preparations with larger γ values are less stable and larger in size. This is consistent with the idea that there are no additional nanoparticles forming in the ascorbic acid regrowth step and the added silver atoms are used in the growth of the existing seed particles.

After centrifugation, the pellet was still a liquid and care had to be taken when removing the supernatant. Because of this, all of the CTAB could not be removed without losing part of the pellet as well. An additional centrifugation step could be used to remove more of the CTAB however the nanoparticles in the resulting pellet were no longer stable and would eventually precipitate out of solution. The longitudinal plasmon peaks were also blue-shifted and for the preparations with small γ values, merged completely with the transverse plasmon peak. This can be seen in figure 5.6. This is thought to be from the aggregation of the nanorods after all of the CTAB was removed. Because of this, a second centrifugation step was only used when preparing samples for TEM, where any remaining CTAB would ruin the TEM images.

Sonication was used to redisperse the pellets in Milli-Q water after they were spun down in the centrifuge. UV-Visible extinction spectra were collected before and after sonication to check the effect sonication might have on the nanorods. After 5 minutes of sonication at 30%, all of the plasmon peaks showed a slight decrease in amplitude. After 3 hours of sonication, the amplitude of the plasmon peaks did not appear to decrease any further, however, the longitudinal plasmon peaks were blue-shifted, with the preparations with larger γ values showing a greater shift. Because of this effect on the plasmon peaks, the sonication time was limited to 1 minute at 30% whenever sonication was required.

/ 2/

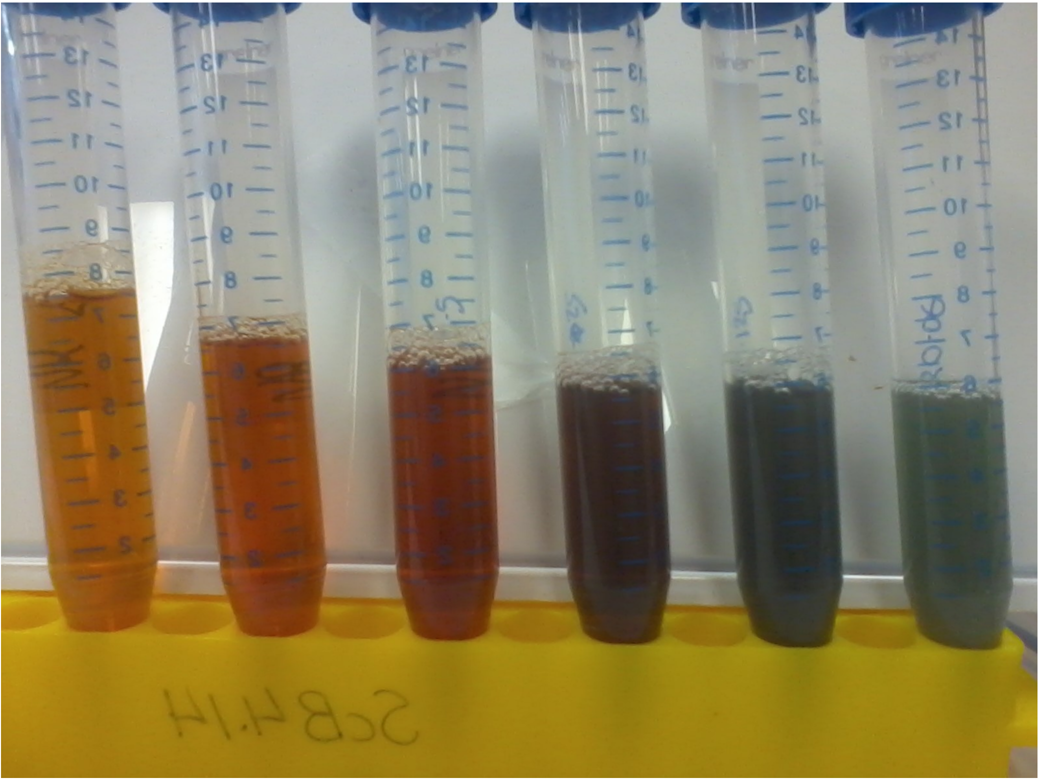
2 

Figure 5.2: Nanorod solutions prepared with various volumes of the citrate capped silver seed solution. From left to right, the volume of seed solution used is: 2,1,0.5,0.25,0.125,0.6 mls. This image has been horizontally flipped so that the position of the samples match the position of the corresponding plasmon peak in the UV-Visible extinction spectrum.

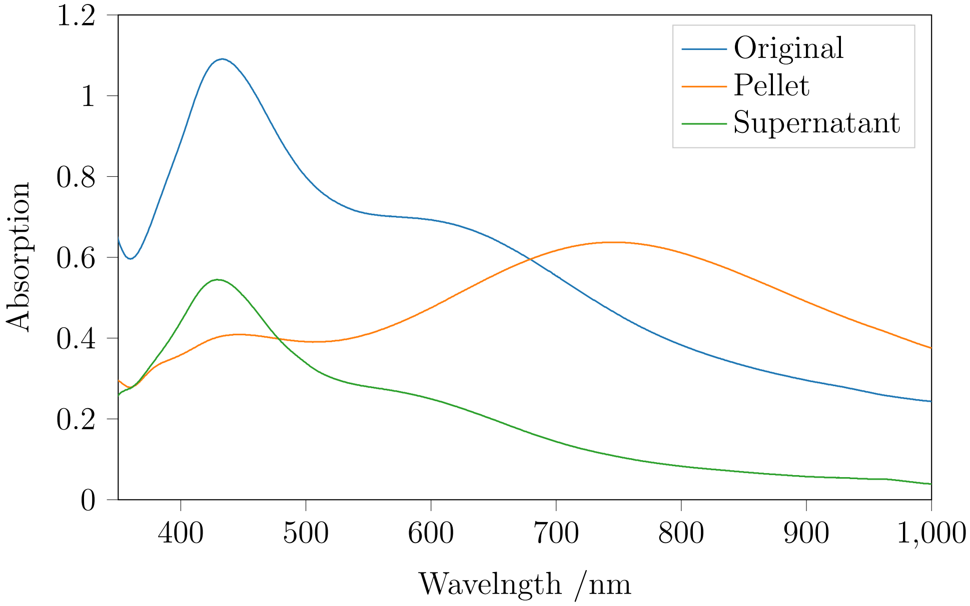


Figure 5.3: UV-Visible extinction spectra of silver nanorods, prepared with a γ value of 8, along with the redispersed pellet and supernatant from the same sample after centrifugation.

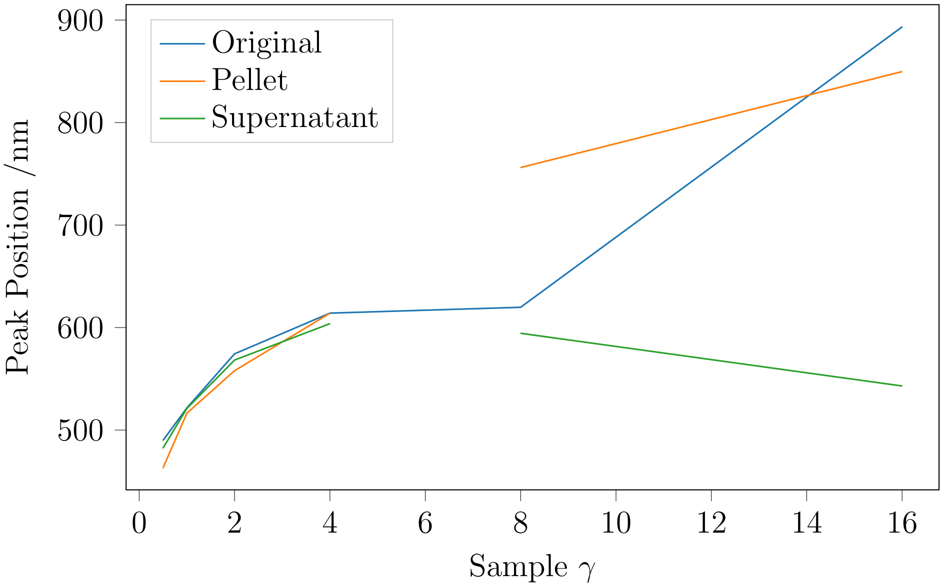


Figure 5.4: Position of the longitudinal plasmon peak for nanorod preparations with various γ values after centrifugation. There is a split in the plots for the pellet and supernatant as a slower centrifuge speed was used for the preparations with a γ value greater than 8. The peak position for the original preparation with a γ value of 16 is an artefact as there was no longitudinal plasmon peak present in the extinction spectrum.

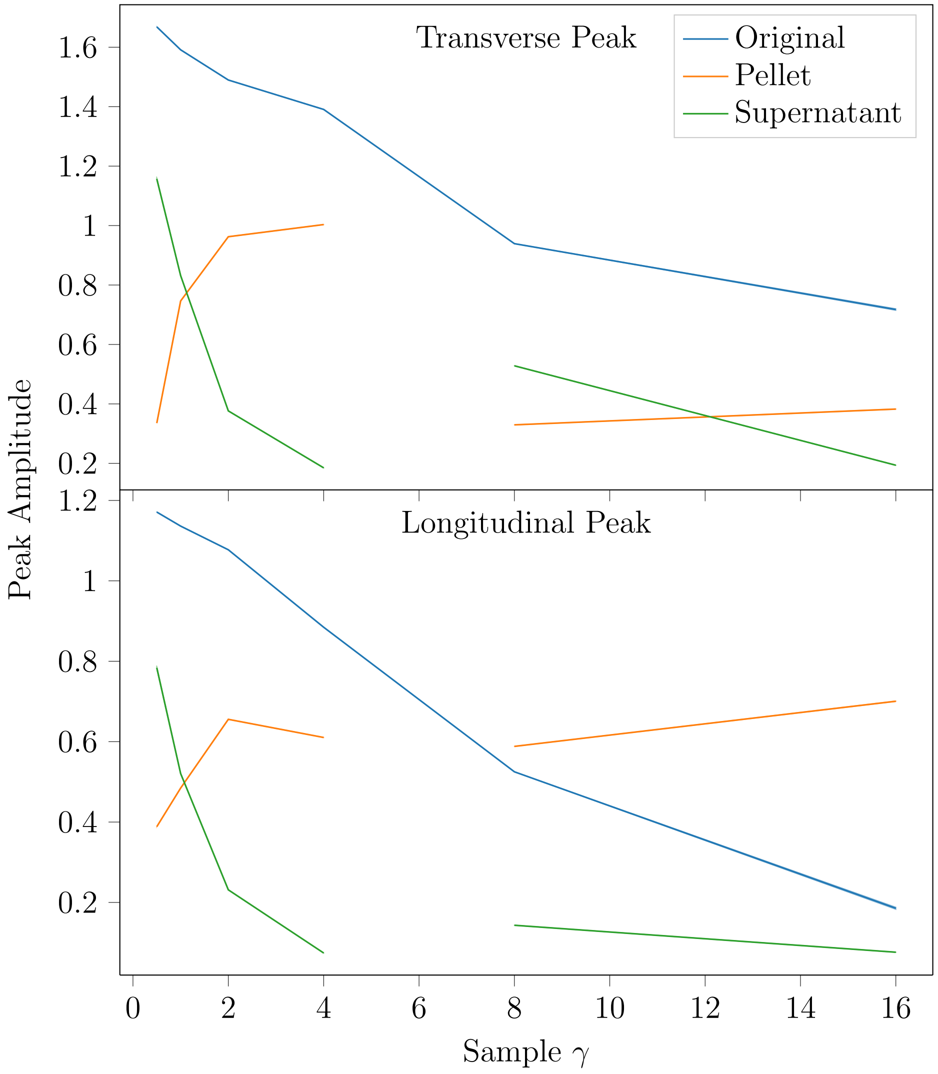


Figure 5.5: Peak amplitude of wavelength-domain pseudo-Voigt functions fitted to the plasmon peaks of nanorod solution before and after centrifugation. The top plot shows the amplitude for the transverse plasmon peaks and the longitudinal peak is shown in the bottom plot. There is a split in the plots for the pellet and supernatant as a slower centrifuge speed was used for the preparations with a γ value greater than 8.

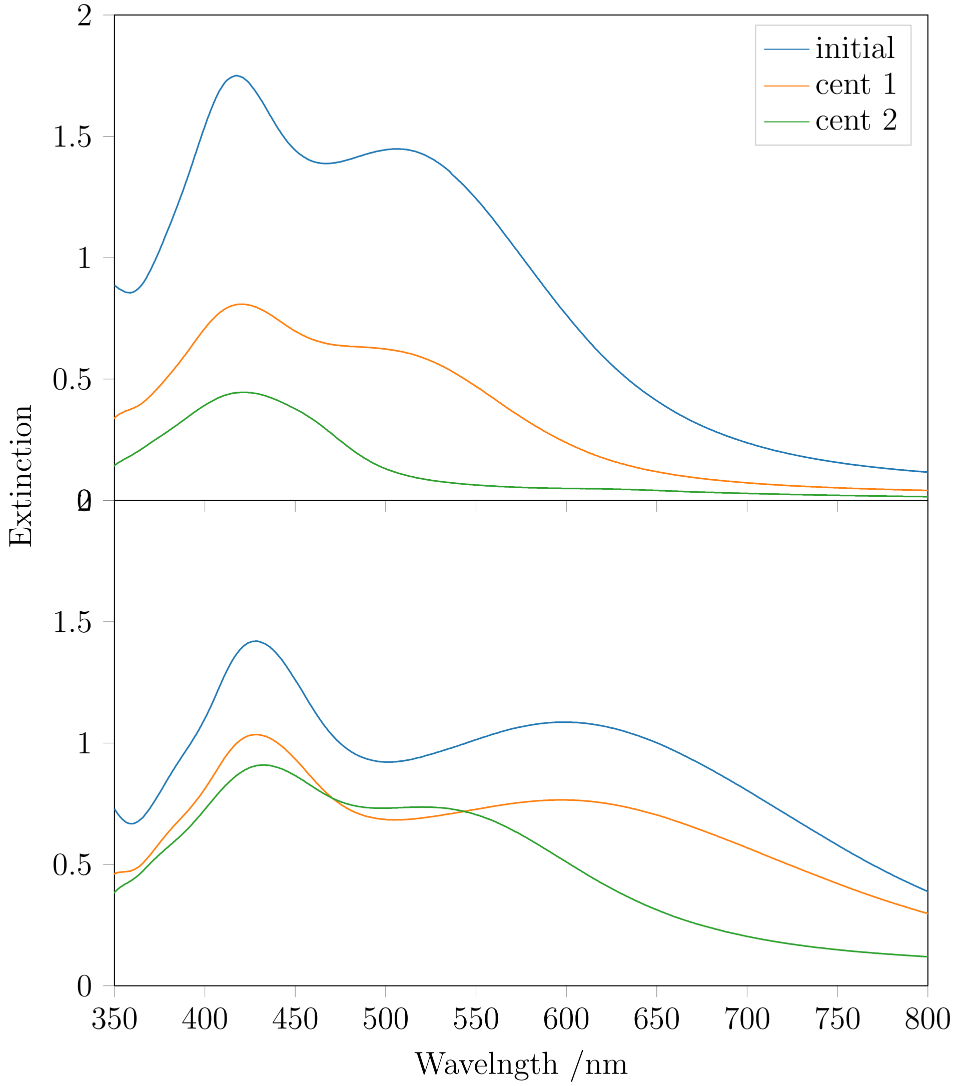


Figure 5.6: UV-Visible extinction spectra of nanorods after 2 centrifugation steps. The plot on the top is for a nanorod preparation with a γ value of 1 amd the bottom plot is for that with a γ value of 4.

5.2 UV-Visible extinction spectra of silver nanorod preparations

UV-Visible extinctions spectroscopy was used to examine the plasmon peaks of the nanorod preparations. The resulting extinction spectra for the various preparations after centrifugation are shown in figure 5.7. Two plasmon peaks can be seen in all of the extinction spectra, which is consistent with the computational results for the nanorods. The transverse plasmon peak, which is found at shorter wavelengths compared to the other peak, is located in a similar region as the plasmon peak of the seed solution, and shifts to a slightly longer wavelength when the volume of the seed solution is reduced (larger γ values). The longitudinal peak has a more pronounced red-shift associated with reduced volume of seed solution and is the main contributor to the colour change of the nanorod preparations.

The amplitude of these two peaks decreases as the volume of the seed solution is reduced (increased γ).

At lower volumes of the seed solution, the amplitude of the longitudinal peak decreases further and the width of the peak increases, resulting in broadband absorption out into the infrared — This is clearly observed during growth and the colour of the solution almost disappears entirely leaving only a faint green colour. It is believed that this reduction is due to aggregation and has already been discussed in the previous section. The return of these plasmon peaks after centrifugation can be seen in the bottom plot of figure 5.7. This increase is not due to a higher concentration of nanoparticles in the pellet solution, as the final weight of the solution was the same as before centrifugation.

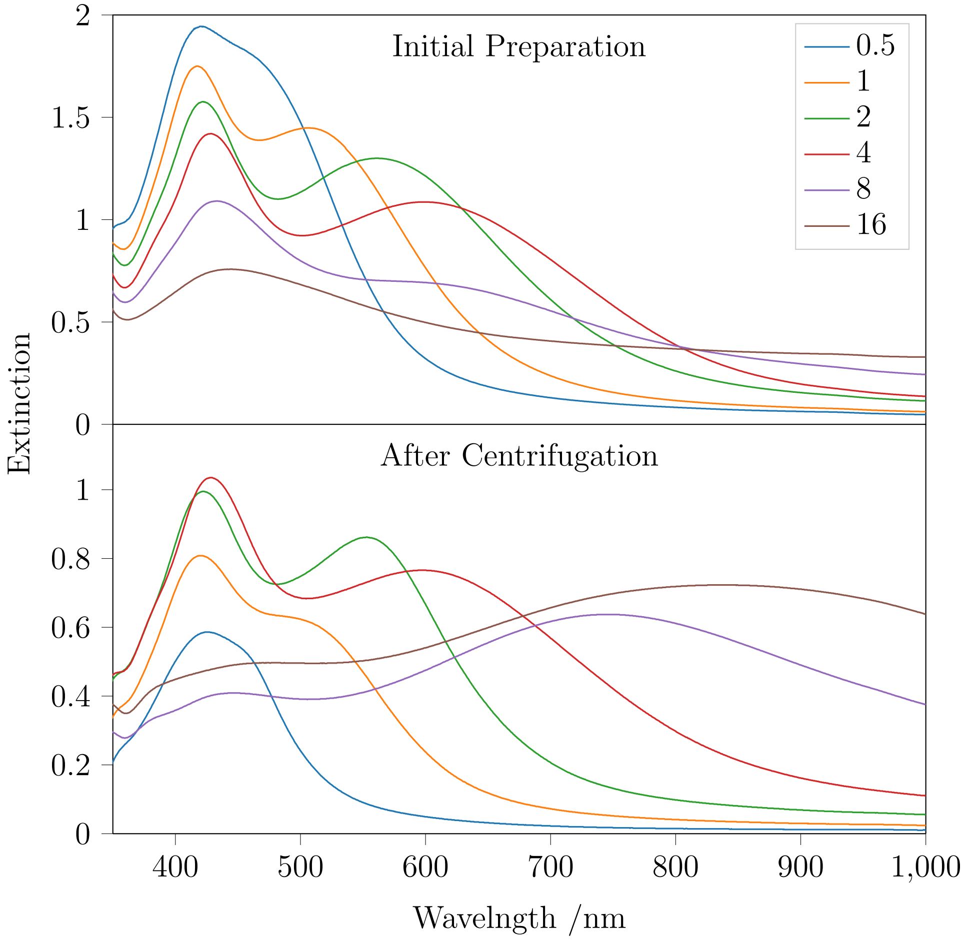


Figure 5.7: UV-Visible extinction spectra of silver nanorod preparations with various γ values. The top plot is from the initial preparations while the bottom plot is from the redispersed pellets after centrifugation.

5.2.1 Deconvolution of the plasmon peaks found in the extinction spectra of the nanorod preparations

Wavelength-domain pseudo Voigt functions were fitted to the extinction spectra of the nanorod preparations to get a closer look at the differences between the plasmon peaks of each preparation. Two of these functions were used to fit the transverse and longitudinal plasmon peaks and an additional function was used to fit the shoulder that appears on the side of the transverse peak. An example of one of these peak deconvolutions is shown in figure 5.8 and the optimised parameters of the psuedo Voigt functions for all of the preparations are shown in figure 5.9.

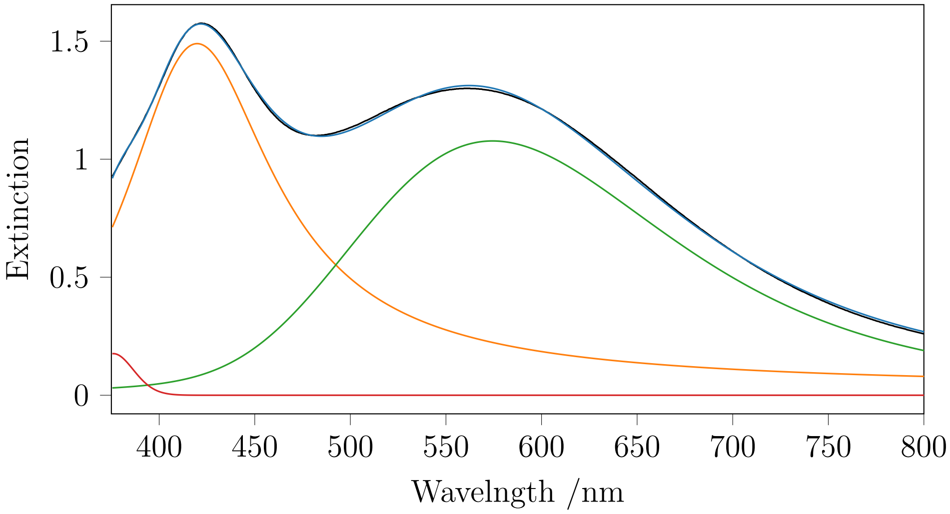


Figure 5.8: Wavelength-domain pseudo-Voigt functions fitted to the plasmon peaks in the UV-Visible extinction spectrum of one of the nanorod preparations (γ value of 2). The extinction spectrum is shown in blue and the transverse and longitudinal plasmon peaks are shown in yellow and green respectively. The convolution of all the fitted peaks in shown in black.

The shifts in the plasmon peak positions and reduction in amplitude, which have already been mentioned, can be seen in the top two plots.

The width of both of the peaks increase when the volume of seed solution is reduced. Except for large volumes of the seed solution, the width of the longer-wavelength peak is larger than that of the shorter-wavelength peak. This discrepancy for smaller values of γ is assumed to be an artefact from the peak fitting function. Where there is overlap between the two peaks, one of the fitted peaks can take on an increased width to compensate for a smaller width of the other peak.

The η values are show in the bottom right plot of figure 5.7. Most of the transverse plasmon peaks have a η value close to 1, which means the peaks are almost entirely Lorentzian. This is consistent with what was seen from the DDA calculations, however, the η value of the longitudinal peaks is closer to 0, meaning the peak is mostly Gaussian. Although there are many possible reasons why this peak has a Gaussian profile, the most likely reason is that there is a range of nanorod lengths in the preparations. The transverse peak isn't Gaussian because the range of widths in the nanorod preparations is narrower and the response of the traverse plasmon peak position to a change in width is smaller than that of the longitudinal peak.

The η value of the longitudinal plasmon peak increases to 1 at higher γ values. This is because the plasmon peak starts to disappear as the nanorods start to aggregate. This may occur because the aggregation results in a narrower distribution of the nanorod length due to the larger ones aggregating first, however, it is more likely due to a

For the lowest γ value, the η values are close to each other and have swapped places, with the longitudinal peak having the largest η value. This is assumed to be another artefact from the peak fit function, this time due to the large overlap between two plasmon peaks. This is also the expected reason for the widths swapping over as well.

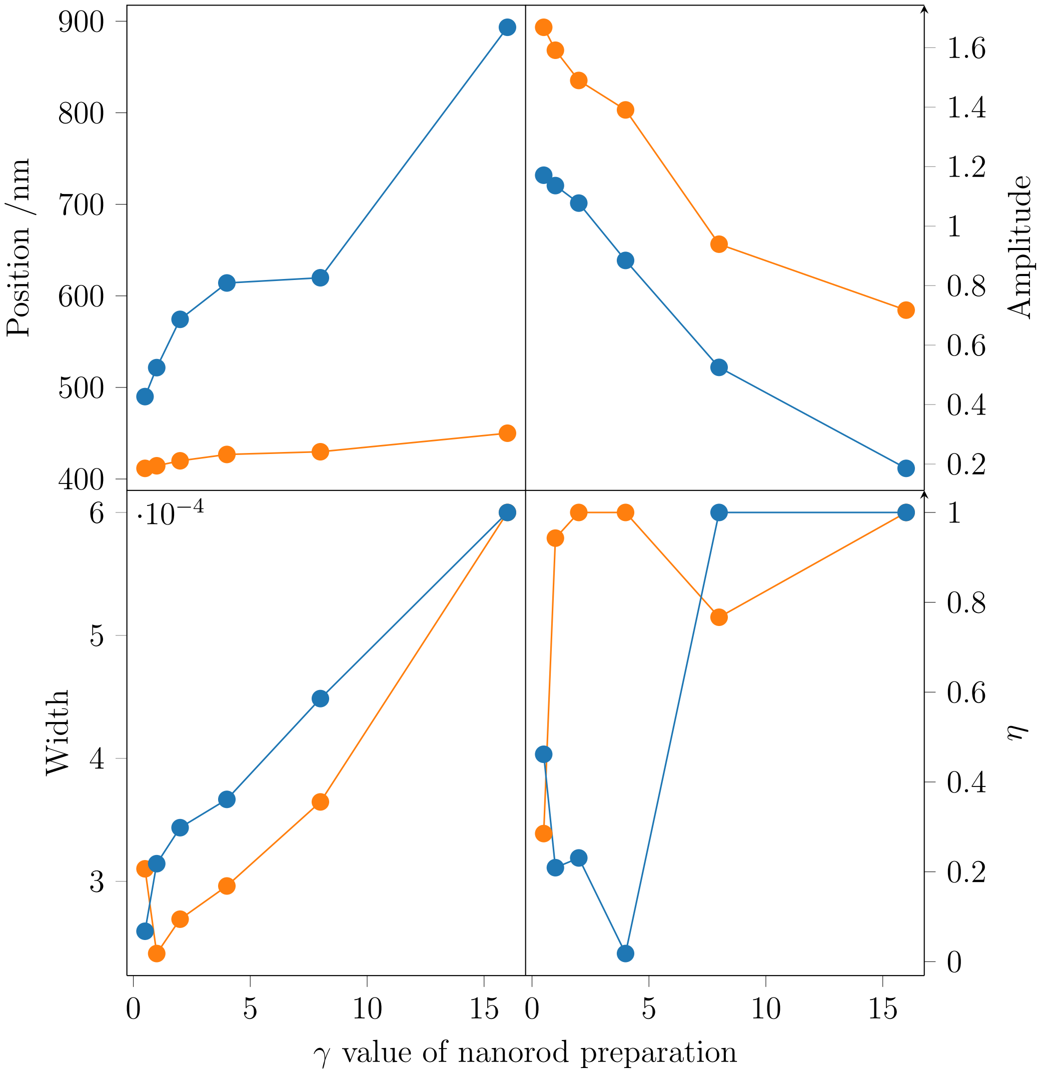


Figure 5.9: Optimised wavelength-domain pseudo-Voigt parameters for the transverse and longitudinal plasmon peaks of silver nanorods prepared using various volumes of seed solution. The transverse plasmon peak is represented in blue longitudinal peak represented in yellow.

5.3 TEM images of the silver nanorod preparations

TEM images were recorded for the various nanorod preparations.

The number of particles found in the TEM images was very low and a range of particle geometries were observed. This can be seen in figure 5.10.

The preparations with larger γ values did contain larger nanorod as expected, however, there was a large variance in the lengths of the nanorods present for any given preparation. Because of the low number of particles present, a comprehensive study of the size distribution could not be performed.

Despite the range of lengths seen, the width of the nanorods were roughly constant for all the preparations at around 30 nm.

Many of the nanorods had a dark line running down the centre and the end caps did not appear to be symmetric. This may be indicative of the penta-twinned crystal structure which is expected for the nanorods.

The dominant geometry found in the majority of the preparations was the trigonal platelets.

After a second centrifugation step for the preparations with γ values of 8 and 16, the trigonal platlets had more or less been entirely removed from the TEM images. This can be seen in figures 5.11 - 5.12. A few nanorods can still be seen in the supernatant after this centrifugation, however, the majority of the particles were the unwanted geometries. This can be seen in figure 5.13. The excess CTAB can also be seen coating these particles.

A few of the nanorods were abnormally long at more than 400 nm. Some of these rods appeared to be fused together as can be seen in figure 5.14.

Figure 5.10: TEM image showing the various particle geometries found in a nanorod preparation with a γ value of 4

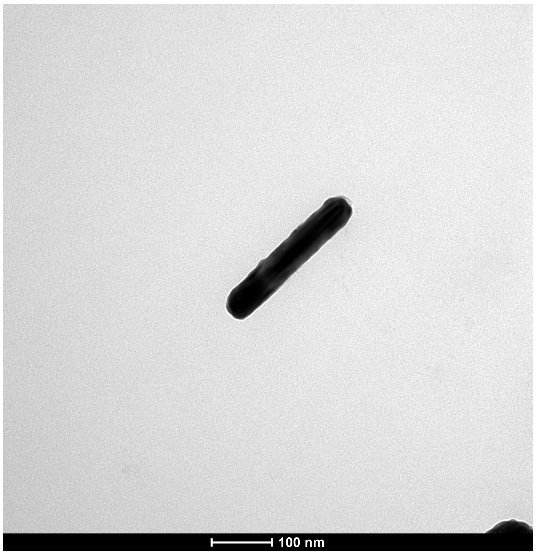


Figure 5.11: TEM image showing the nanorods found in the preparation with a γ value of 16 after centrifugation

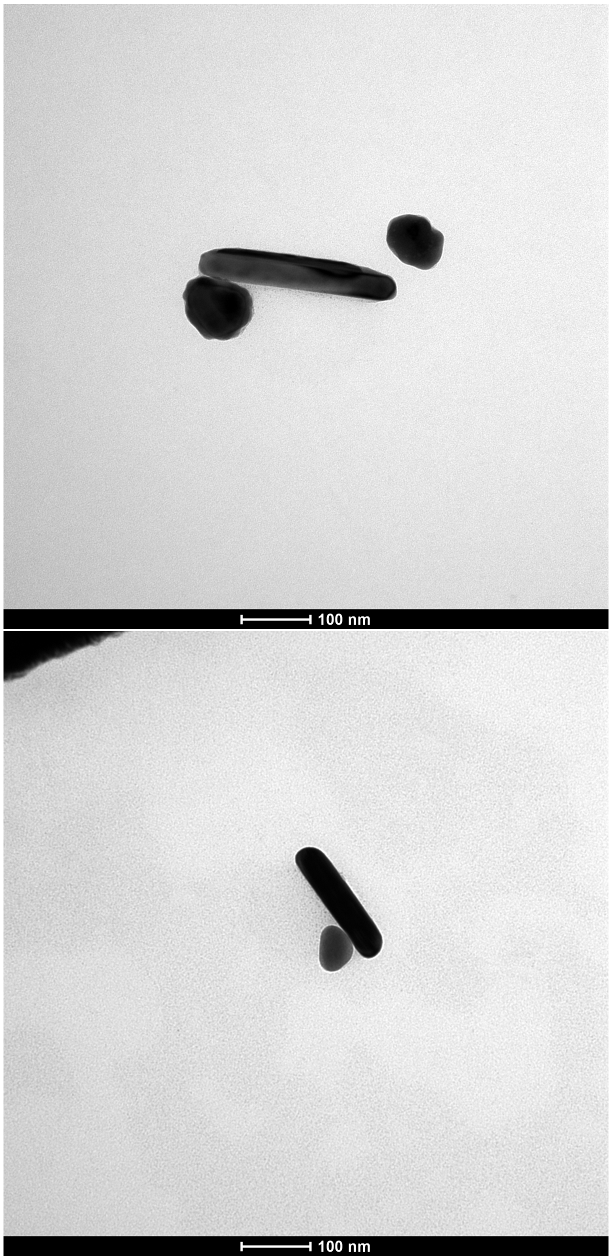


Figure 5.12: TEM image showing the nanorods found in the preparation with a γ value of 16 after centrifugation

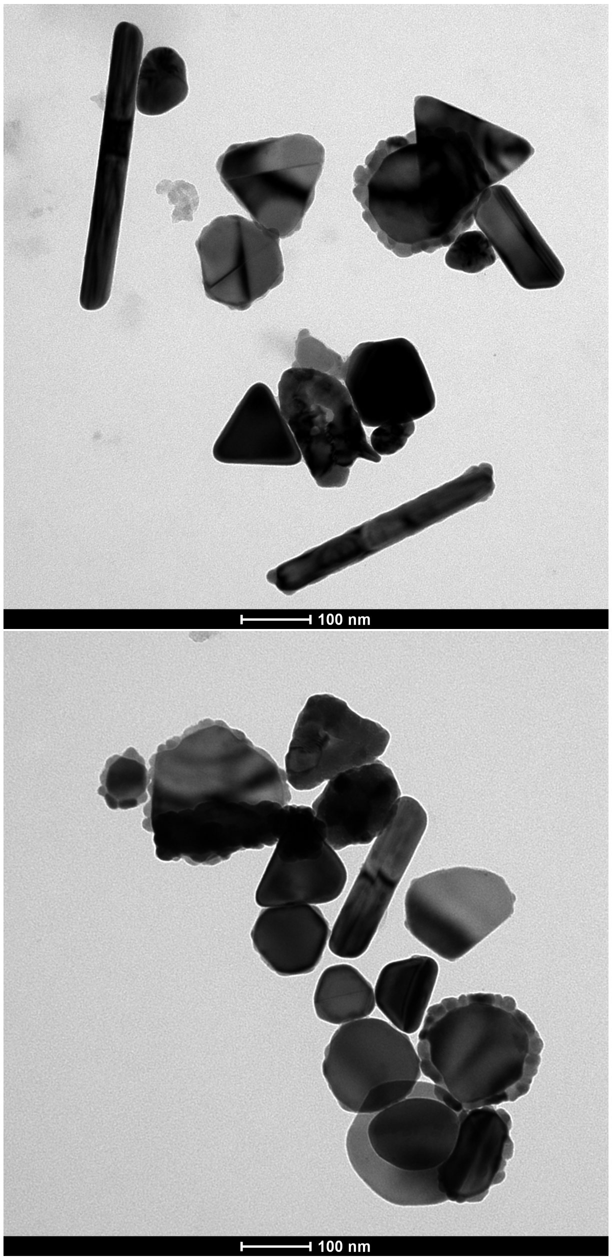


Figure 5.13: TEM images showing the various particle geometries found in the supernatant of a nanorod preparation with a γ value of 16 after centrifugation

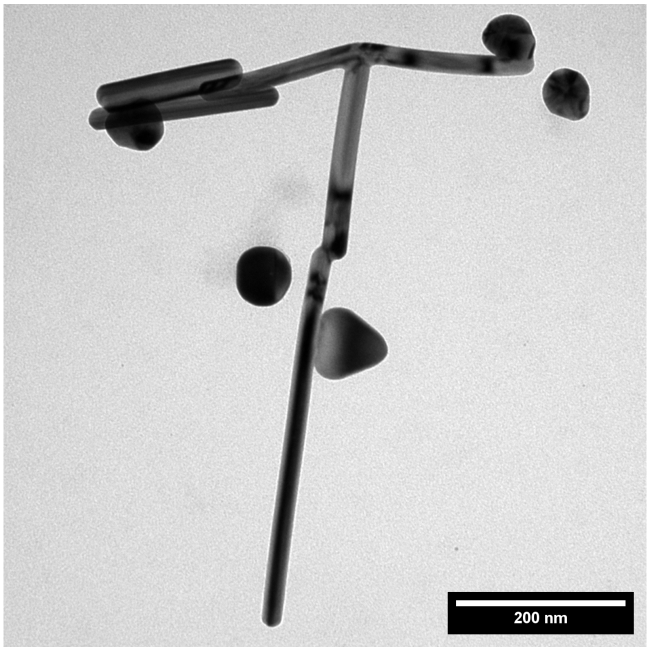


Figure 5.14: TEM image showing abnormally long nanorods in a preparation with a γ value of 16

5.4 Altering the geometries of the silver seed particles using photochemical transformation

After exposure to the lamp, the initial pale yellow colour of the solution began to darken and then change colour. The UV-Visible extinction spectra of the nanoparticles were recorded after various exposure times to determine the effect on the plasmon resonances of the nanoparticles. These spectra are show in figure 5.15.

The UV-Visible extinction spectra shows that there was initially only a single plasmon peak and after exposure to the lamp, two additional peaks appear while the initial one slowly disappears. It has been reported that these two new peaks are from the decahedral nanoparticles and trigonal platelets, with the longer wavelength peak belonging to the platelets [44].

An isosbestic point can also be seen around 460 nm which typically arises when two species are in equilibrium. This supports the idea that certain nanoparticle geometries are growing while others are dissolving.

Three wavelength-domain pseudo Voigt functions were fitted to each spectra to get a closer look at what is happening to the plasmon peaks. The evolution of the peak amplitudes are shown in figure 5.16. In the first 90 minutes of exposure to the lamp, the initial plasmon peak (blue line) decreased in amplitude while that of the decahedral nanoparticles (yellow line) and trigonal platelets (green line) increased. After 90 minutes, the plasmon peak of the decahedral nanoparticles continued to increase while both of the other two peaks decreases. This is consistent with what was reported in the initial paper [44] where they found exposure to the lamp initially promoted transformation into the trigonal platelets and further exposure resulted in

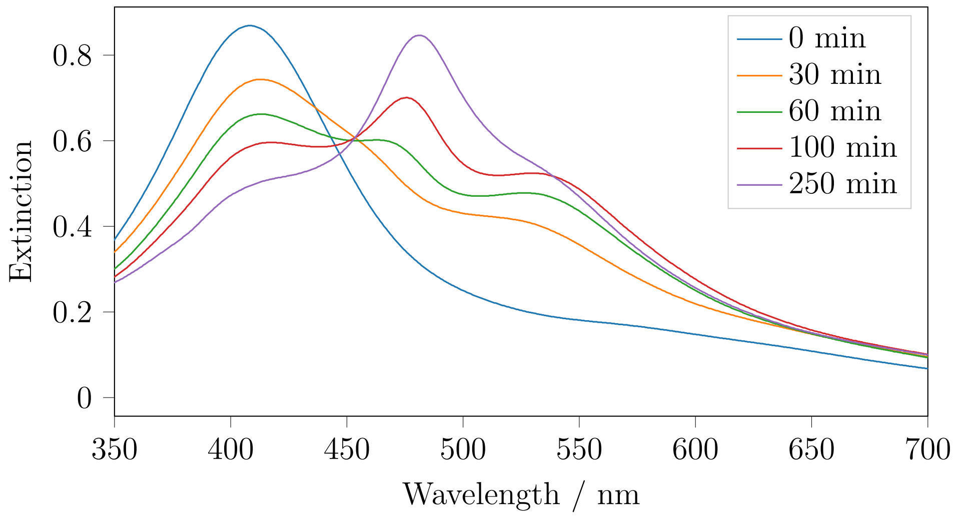


Figure 5.15: UV-Visible extinction spectra of silver nanoparticles during exposure to light from a xenon arc lamp.

the growth of the decahedral particles.

There is a spike in the peak amplitude for the decahedral nanoparticles (yellow line) at the 30 minutes mark This is expected to be an artefact of the peak fitting, as there is also a decrease in amplitude for the initial plasmon peak position and there is significant overlap between these peaks.

The position of the plasmon peaks of the decahedral nanoparticles and the trigonal platelets both shift to longer wavelengths, which is consistent with these particles growing larger in size. After the amplitude of the trigonal platelets peak starts to decrease, the position also stops shifting to longer wavelengths.

TEM images of the nanoparticles after 250 minutes of exposure are shown in figure 5.17. It is immediately obvious that the particles had grown larger in size which meant the geometries of each particle could now be observed

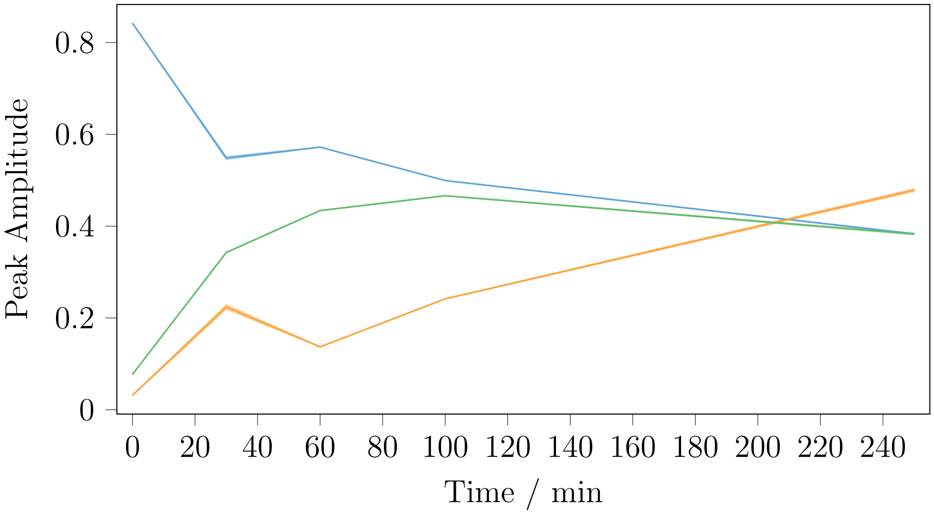


Figure 5.16: Evolution of the plasmon peak amplitudes of different particle geometries during exposure to light from a xenon arc lamp. the blue line shows the change for the plasmon peak that was initially present before exposure and the yellow and green lines show the change for decahedral nanoparticles and trigonal platelets respectively.

with TEM. The top image shows a particle with the pentagonal bipyramid decahedral geometry that is required in the preparation of the nanorods. These bipyramids all had a width ranging from 20 - 55 nm. There were also smaller nanoparticles with a width around 12 nm but it was unclear if these had a bipyramid geometry. The concentration of the nanoparticles was low which meant it was not possible to get an accurate idea of the size distribution of any of the particle geometries.

As expected, the trigonal platelet geometries were also present in the TEM images, which can be seen in the bottom image of figure 5.17. A large proportion of these platelets had truncated corners, with a few being perfect hexagonal platelets. This was not observed in any of the platelets found in the TEM images from the nanorod preparation and could be more evidence that the platelets were in the process of dissolving in favour of the growth of the decahedral nanoparticles. The size distribution of the trigonal platelets was similar to that of the decahedral particles.

Another geometry that was observed is the diamond shaped particles which can be seen in the bottom image of figure 5.17. Initially this was assumed to be two trigonal platelets stuck together, however, the particles appear darker in the image when compared to any nearby platelets suggesting that they are thicker than the platelets.

The remaining nanoparticles either had a roughly circular shape or they looked like grains of rice. These particles were much smaller than the previously discussed geometries. The larger of the circular particles had more irregular geometries and they did not appear any darker than the surrounding smaller circular particles which suggests that they might be platelets. These particle may be the remnants of the trigonal platelets that had been significantly dissolved.

The particles that looked like grains of rice are of interest as this geometry was also found in the nanorod TEM images. These were roughly 10 nm long and there appeared to be a shadow running down the length of the particle, however, as this geometry was rare and had a small size, the actual geometry could not be determined.

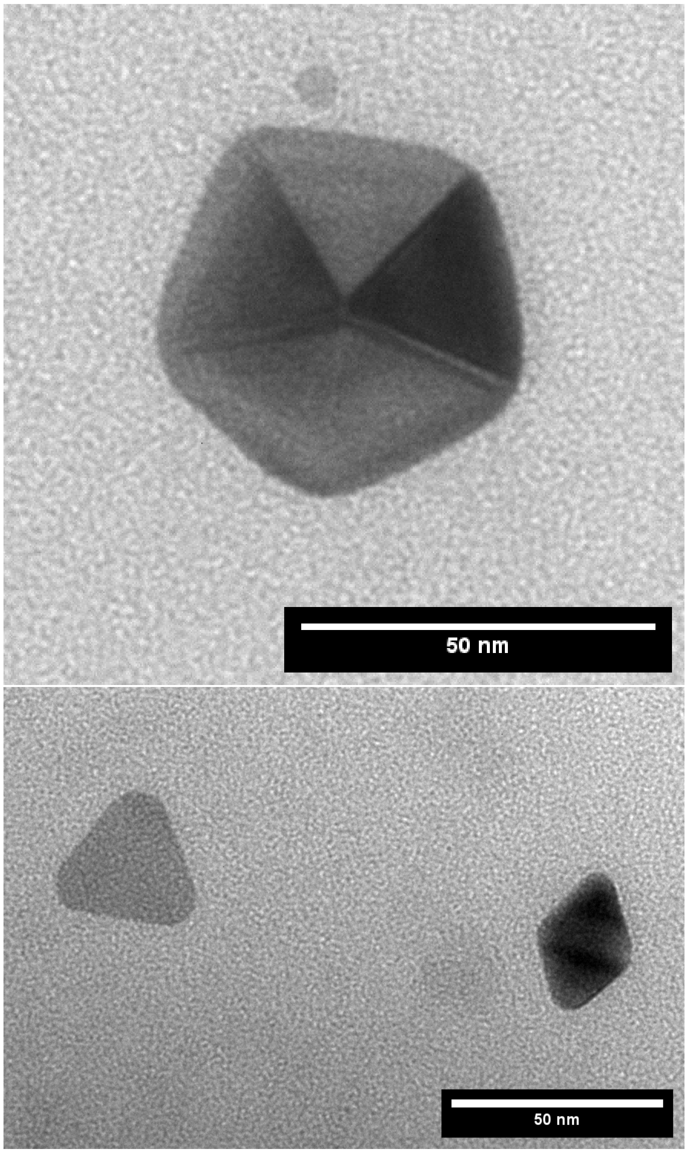


Figure 5.17: TEM images of silver nanoparticles after 250 minutes of exposure to light from the Xenon arc lamp. The top image shows a decahedral bipyramid nanoparticle while the bottom shows a trigonal platelet along with an unknown diamond shaped particle

5.5 Time-resolved UV-Visible extinction spectroscopy during nanorod growth

Wavelength-domain pseudo Voigt functions were fit to the time-resolved extinction spectra recorded during the grow of nano rods.

Figure 5.18 shows the evolution of the longitudinal plasmon peak position during the growth of nanorods for preparation with various γ values.

All of the preparation are shown to reach a steady state once the growth has finished. However the preparation with γ value of 16 appears to start shifting again after 200 seconds. This is not because of additional particle growth but rather it's an artefact from the peak fitting because the longitudinal peak disappears entirely which allows the fitted peak shifts to longer wavelengths to cover the baseline extinction. This can be seen in figure 5.19. The amplitude starts to decrease after around 200 seconds, this is consistent with what is seen in figure 5.18. The η value of the longitudinal peak also increases to 1 after 200 seconds, which is likely due to the tails of the Lorentzian function fitting the baseline extinction. This is expected to be due to aggregation which is consistent with what was observed from the centrifugation results.

The evolution of the longitudinal plasmon peak positions for nanorods grown at various temperatures is shown in figure 5.20. The longitudinal peak position reaches the maximum earlier for the nanorods grown at a higher temperature. This is expected, as the growth of the nanorod should be faster at higher temperatures.

The maximum peak position is at a lower wavelength for the nanorods grown at a higher temperature. This may suggest the higher temperature also promotes the growth in transverse direction which would result in an increase in the width and a reduction in the final aspect ratio of the nanorods.

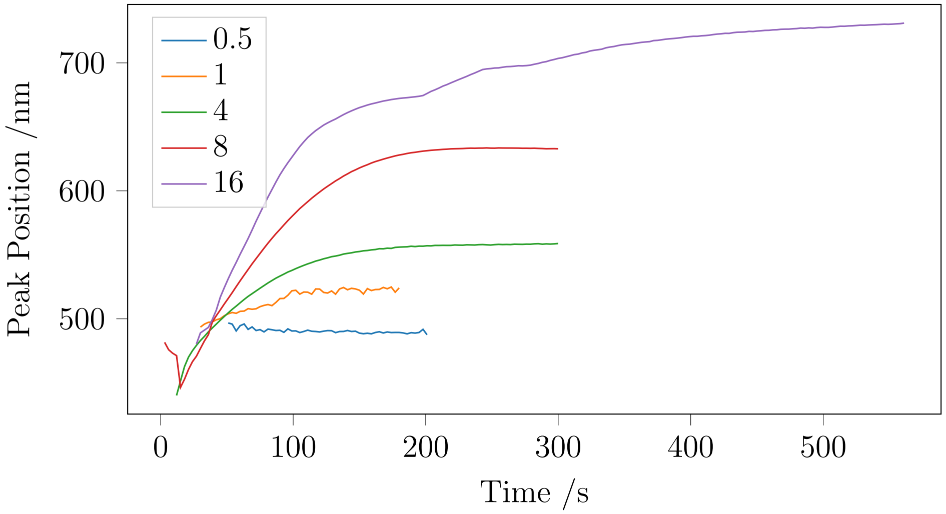


Figure 5.18: Position of the longitudinal plasmon peak during the growth of silver nanorods grown with various values of γ.

Interestingly, the higher temperatures seems to decrease the stability of the nanorods as the plasmon peak position starts to blue shift over time. There is also a reduction in the amplitude of the longitudinal peak, similar to what was seen for the larger rods which aggregated as time went on, seen in figure 5.19.

However, this may not be due to aggregation, as there was not blue shift observed in figure 5.18. Therefore, this may be some ripening process that is promoted with an increase of temperature; it may be that the widths continue to grow at the expense of the length. TEM would be required to verify this.

Figure 5.21 shows the longitudinal plasmon peak during the growth of silver nanorods grown with various volumes of silver nitrate in the growth solution. The yellow line is for the preparation with a γ value of 4. The other lines are from preparations with γ values of: 3.2 for the blue line, 4.8 for the green,

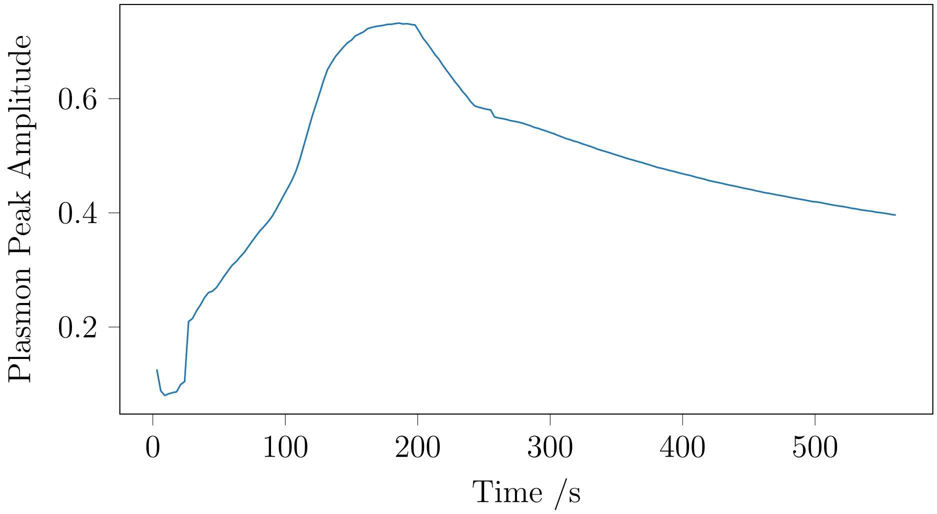


Figure 5.19: Amplitude of the longitudinal plasmon peak during the growth of silver nanorods, with a γ value of 16, showing the aggregation of the nanoparticles

and 8 for the red line. It must be noted that these preparations are not equivalent to the preparations with the same γ values presented elsewhere in this thesis. In the case of the red line, the γ value is 8, however, the concentration of particles is going to be double that of the preparation with γ value of 8 discussed in other parts of this thesis.

The red line shows a blue-shift in the plasmon peak position. This could be a sign of aggregation due to a combination of the larger particle size as well as a higher concentration of particles. However, a blue-shift was not seen in figure 5.18 where aggregation was seen to occur, so TEM would need to verify this.

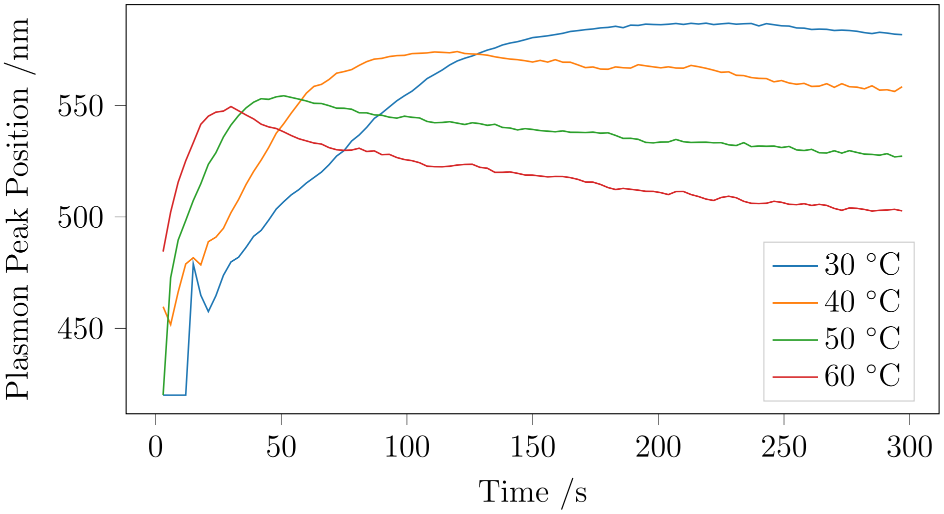


Figure 5.20: Position of the longitudinal plasmon peak during the growth of silver nanorods, with a γ value of 4, grown at various temperatures.

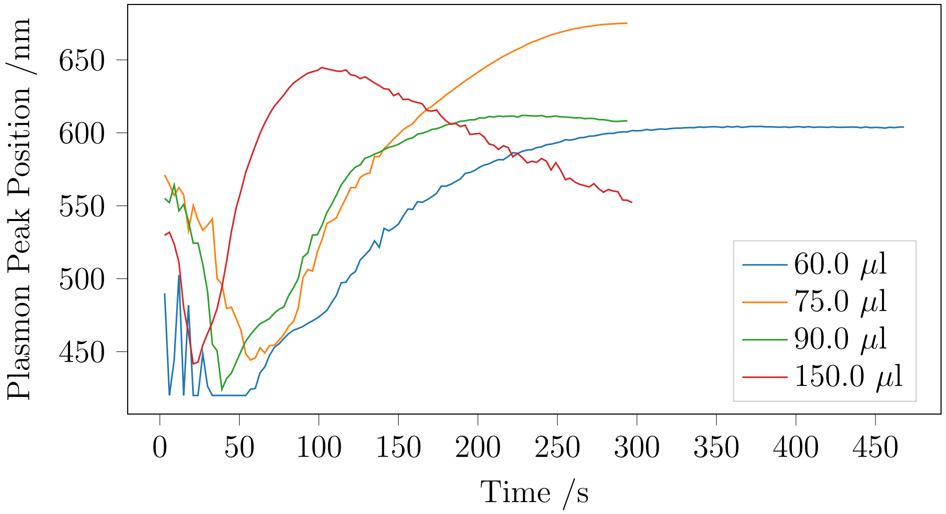


Figure 5.21: Position of the longitudinal plasmon peak during the growth of silver nanorods, with a γ value of 4, grown with various volumes of silver nitrate in the growth solution.

Chapter 6

Discussion

6.1 Spectral peak fitting using the data_fit Python module

The `data_fit` module that was created for performing the peak fitting in this Masters project proved to be invaluable and saved many months of work that would have otherwise been spent performing the peak fitting manually.

The `data_fit` module not only proved useful for the work performed in this thesis, but was also used by other masters and PhD students for the data analysis in their own projects.

One of the features that made this module so useful was the modularity and the ability to add new fit functions without altering any of the existing code or files in the module. Initially this module was created to only perform the peak fitting of the spectral data but this was very quickly expanded to include fit functions that were used in the other sections of this thesis. Some of these other applications include: determining and applying the corrections for the DDA calculations and fitting the growth models to the experimental extinction spectra of the nanorods preparations. The `data_fit` module was so useful that all of data reported in this thesis was either created using the `data_fit` module or was analysed using it.

One major improvement that was discovered when developing this module was to reuse the same `Fit` object and its associated fit functions for multiple spectra. This effectively allowed the optimised fit parameters for one spectrum to be used as the initial parameters for the next. As all the various spectra were similar, this led to a significant improvement in the ability for the optimisation to find the correct solution as well as a reduction in the time taken to do so.

A good example of reusing the `Fit` object is seen in the time-resolved extinction spectra captured during the growth of the silver nanorods. In this example the optimised fit from one time step was used as the initial fit for the next time step. This is important for the time-resolved spectra as the signal to noise ratio was much lower than the non-time-resolved spectra which made it harder for the optimisation to fit the correct fit. This was partly due to the fact that the diode-array spectrometer which was used to capture these spectra was not fully enclosed which let stray light into the spectrometer. Another reason, is that the diode-array spectrometer is only a single-beam spectrometer, compared to the other spectrometer which is double-beam. The double-beam spectrometer corrects any fluctuations in beam intensity while the single-beam spectrometer does not. Reusing the `Fit` object for the time-resolved spectra roughly halved the time it took for the optimisation to find a solution.

Another improvement was discovered by reversing the order in which the time-resolved spectra were analysed. Essentially finding a fit for the spectrum associated with the final time step first and then working backwards to the initial time step. This had an immense improvement on finding correct fits for the plasmon peaks during the initial stages of growth. During these stages, there is significant overlap between the plasmon peaks which allowed a single fit function to provide the fit for both peaks while the other fit function would provide a fit for the baseline noise. This is seen in figure 2.5.

When the order of iteration was reversed, the optimisation would easily find the correct fit for the peaks at the later stages of growth as there was significant separation between the peaks. Because the time step was small, there was only a slight change between subsequent spectra. So once the initial correct fit was found, the correct fit for all subsequent spectra was found easily, including the spectra from the initial stages of growth where there was significant overlap of the plasmon peaks.

This could have also been achieved by by tightening the bounds and setting the initial parameters closer to what is expected for the optimised fits. However, this would have needed to be done manually for each spectrum, drastically increasing the time required for the analysis.

One possible future improvement for the `data_fit` module would be to include relative bounds for the fit parameters.

One problem that occurred in some of the optimised fits was when the fit function assigned to the longitudinal peak swapped position with that for the transverse peak. This occurred when there was significant overlap between the two peaks.

Obviously the bounds of each fit function could be tightened to forbid this swapping over. However, these restrictive bounds would not be suitable for fitting any subsequent spectra where there was sufficient separation between the plasmon peaks, meaning the `Fit` object could not be reused without manually setting the bounds for each spectrum.

Being able to set relative bounds would remove the need to examine each spectrum to manually determine the required bounds and would still allow adequate freedom in the bounds so the `Fit` object could be used for subsequent spectra.

Currently it is not possible to update the bounds during each iteration of the `curve_fit` optimisation. A new implementation of `curve_fit` would need to be created to allow for this.

An easier way to effectively add relative bounds while still using the existing `curve_fit` function would be to purposefully return a bad fit if the fit functions had swapped positions. This could be easily achieved when `curve_fit` call the `Fit.build_fit` method. Care would need to be taken, however, to ensure that the purposefully bad fit did not inadvertently satisfy any of the tolerance conditions of the `curve_fit` function.

Another problem was observed when the plasmon peak amplitude was relatively low compared to the background in the extinction spectrum. This occurred when the nanoparticle solution was either in the initial stages of growth or it had been diluted. In these conditions it was not uncommon for the fitted longitudinal function to reduce in amplitude and increase in width in an attempt to provide a fit for the background. This problem could be addressed by either performing a background correction or by setting the sigma argument of the `curve_fit` function to provide a larger weight to the data point with higher extinction and essentially ignoring the low background. However, setting the sigma values in this way would not be a suitable solution if there is significant overlap with the plasmon peaks. This is because the weight given to one side of the plasmon peaks would be larger than for the other side as the extinction on one side of the peak is higher due to the overlap with the other peak. This could potentially cause a shift in the optimised peak positions.

Another future improvement of the `data_fit` module would be to update the way the uncertainty of the optimised parameters is expressed. Currently this is expressed with the use of confidence intervals, however, these confidence intervals assume there is not covariance between the parameters and is calculated using only the diagonal elements of the covariance matrix. Obviously, the covariance between the parameters is going to become significant as the peaks overlap. This problem is seen for these overlapping peaks where the confidence intervals are incredibly small even though the uncertainty in the optimised parameters is expected to be large. When plotted, most of these confidence intervals are smaller than the width of the plot line. It is unclear how the uncertainty of these parameters should be expressed while taking in to account the off-diagonal elements of the covariance matrix.

6.2 Computational calculations for silver nanorods

6.2.1 Validity of the discrete-dipole approximation calculations for nanorods

The DDA, as the name implies, is only an approximation. Before the DDA results can confidently be used to describe the nanorod preparations, the validity of this approximation must be verified.

The first condition that must be meet before the DDA results can be considered valid, is to ensure that the dipole size is small relative to the wavelength of light. The authors of the DDScat software state that the dipole size must satisfy the following condition:

$$d < \frac{\lambda}{2\pi|m|}, \tag{6.1}$$

where d is the dipole size, $|m|$ is the complex refractive index, and λ is the wavelength [45]. For silver nanoparticles, this corresponds to a maximum dipole size of 20 nm for the wavelength range examined (300 - 800 nm). This condition is definitely satisfied, as the maximum dipole size used was 1 nm.

The next thing that needs to be verified is if the target geometry can be approximated with the cubic dipoles.

It is well known that the largest components of error for the DDA comes from this approximation of the geometry [46]. One of these errors comes from the discretization of the electric field inside the target due to it being constant over the volume of each dipole. This type of error is known as the discretization error. The other type of error is known as the shape error and arises from having to approximate the curved surfaces of the target geometry with the cubic dipoles.

Both of these errors are expected to disappear as $d \to 0$ and the discretization error has been shown to disappear faster than the shape error [46].

Draine has shown that, for spherical particles, these errors are proportional to $N^{-\frac{1}{3}}$, where N is the total number of dipoles [40]. To correct this error, it was suggested to perform the DDA calculations at two different dipole sizes, and then use this relationship to extrapolate the convergence out to find the calculated value for an infinitely small dipole. No such correction has been reported for the nanorod geometry.

In attempt to determine the required correction of the capsule geometry, the DDA calculations were calculated at increasing small inter-dipole spacings in order to study the convergence. When this was done for the spheres and equation 4.1 was fit to the data, the expected result was not obtain; which is that the converges with a $N^{-\frac{1}{3}}$ relationship. However, it is believed that this error arises from the data fitting and not from a disagreement with this relationship.

When the convergence is studied for the capsule geometry, a must better result is obtained. some of the fits failed, however, for the rest that didn't, the average convergence parameter, $N^{-\frac{1}{\chi}}$, was found to be $chi = 2.9 \pm 0.2$; which agrees with the known convergence for the spheres.

It is believed that some of the fits failed because the inter-dipole spacings used were not small enough, even though the maximum number of dipoles used was over 1.2 million.

The DDA calculations need to be repeated at smaller dipole sizes to confirm this relationship. However, as the DDA calculations are slow, alternative methods were devised to determine this convergence.

The discretization error comes from all of the dipoles in the target, however, the shape error only comes from the dipoles at the surface of the target geometry. Because of this, the shape error has been approximated by studying the proportion of dipoles that are at the surface of the geometry as the inter-dipole spacing is decreased.

For spheres, the number of surface dipoles has been estimated by calculating the volume of a shell at the surface of a sphere. Then ratio of this shell volume to the total volume is the same as the ratio of the number of surface dipoles to the total number of dipoles for a sphere.

Sosa et al. estimated the proportion of surface atoms in spherical particles by considering the proportion of the volume of a sphere which is contained in a spherical shell at the surface of the sphere which has a thickness of $\frac{d}{2}$

[47]. This relationship was found to be:

$$\begin{aligned}\frac{N_S}{N} &= 1 - \frac{1}{N}\left(N^{1/3} - 1\right)^3 \\ &= -\frac{3}{N^{2/3}} + \frac{3}{N^{1/3}} + \frac{1}{N}\end{aligned} \tag{6.2}$$

The second term in equation 6.2 is the dominant term, as the other two disappear quickly with increasing N. This is where the correction factor of $\frac{1}{3}$ comes from.

Kooij et al. applied the same approach to estimate the proportion of surface dipoles in a cylinder [48]. This relationship was found to be:

$$\begin{aligned}\frac{N_S}{N} &= \pi(2 \cdot A + 1)\left(\frac{1}{2\pi A}\right)^{2/3}\left(\frac{1}{N}\right)^{1/3} \\ &+ \pi(2 \cdot A + 1)\left(\frac{1}{2\pi A}\right)^{1/3}\left(\frac{1}{N}\right)^{2/3} + \frac{\pi}{2}\left(\frac{1}{N}\right)\end{aligned} \tag{6.3}$$

This suggests that the required correction for the nanorods may be dependant on the aspect ratio of the rods. Unfortunately the solution is not as simple when this technique is applied to the cylindrical rods with hemispheric end caps that are used to describe the nanorods in the DDA calculations.

In order to estimate this shape error and determine the required correction for the nanorod DDA calculations, Python/Rust scripts were created to count the proportion of surface dipoles at increasingly small inter-dipole separations. Unlike the studies that were just mentioned, the dipole-counting scripts considered any dipole less than d away from the surface to be considered a surface dipole. When $\frac{d}{2}$ was used, most of the dipole that would obviously be considered part of the surface were not detected. This value of d was determined by viewing a 3D plot of all the surface dipoles and changing the shell thickness until the detected dipoles gave a good representation of the expected surface dipoles.

The results from the surface dipole counting method for the spheres is incredibly stochastic and incredibly small inter-dipole distances had to be used before all the fit parameter χ converged on the expected value of 3. This required over 2 billion dipoles for the larger spheres; much larger than what was used when performing the DDA calculations, which was 1.2 million for the 50 nm spheres. This may explain why the attempt to find the convergence of the DDA calculations in the previous section failed.

The data points from the dipole counting showed an obvious decrease in stochasticity as the number of dipoles increased; for this reason, the optimisation of the correction function was weighted in favour of the data points corresponding to larger dipole counts. This was achieved by setting the `sigma` values for each of the data points to $\frac{1}{N}$ in the optimisation. This assisted in the convergence of the correction parameter χ without the need perform calculations at smaller inter-dipole spacing, saving considerable amount of time. This is expected to be very helpful for the DDA calculations which take significantly longer to perform and therefore cannot be performed at the smaller inter-dipole distances that were used in the surface-dipole counting method.

When this surface dipole counting is performed for spheres, the calculated correction factor, $\frac{1}{\chi}$, gives excellent agreement with that reported by Draine, with χ being equal to 3.03 ± 0.01.

When this is repeated for rod geometries, It appears that there is some relationship between χ and the aspect ratio of the nanorod. However this relationship changes drastically when the counting is repeated with smaller inter-dipole spacings. This observation paired with the fact that there is no obvious trend in this relationship when nanorods are grouped by their width, suggests that the observed relationship is merely an artefact of the dipole counting. This can be seen in figure 4.4.

When the data points were grouped by the length of the rods, the plot of the χ values looked more like that of the spheres, where there was no obvious relationship with the particle size. This is seen in figure 4.5. Interestingly, even though there was considerable variation in all the data points, there was only a small variation for the data points associated with a fixed width.

The average value of χ for the rods is 3.00 ± 0.01, suggesting the correction for the rods is the same as that for the spheres.

This method assumes that all of the surface dipoles have the same contribution to the shape error. This may not be the case, however, as the curvature of the capsule geometry is not the same over its entire surface. For the cylinder section of the geometry, the curvature is only in one direction as opposed to the end-caps where it is in two directions. Because of this, it may be expected that the surface dipoles of the end-caps would contribute more to the shape error.

When this surface-dipole counting was repeated, counting the surface dipoles of the end caps only, the average value of χ was found to be 3.044 ± 0.003. The plot of all the χ values looks similar to that of the spheres and it did not appear like there was a relationship with the aspect ratio.

When this surface-dipole counting was repeated for the surface dipoles of cylinder section of the rod, the average value of χ was found to be 2.97 ± 0.02. These fit parameters can be seen in the bottom plot of figure 4.6. Although, there appears to be some relationship between this fit parameter and the width of the nanorod, this is expected to be an artefact from the dipole-counting.

The reason why the χ is constant for these widths, is because the ratio of surface dipoles to total dipoles, $\frac{N_S}{N}$, is constant for an increase in aspect ratio. The noise seen at low aspect ratios is due to the low number of surface dipoles from the cylinder section; at an aspect ratio of 1, there are none of these surface dipole at all.

If the surface dipoles from the end caps does have a larger contribution to the shape error than that of the cylinder section dipoles, the DDA correction for the rods is likely to be the same as for the spheres as there is no relationship between the correction parameter and the aspect ratio; although it is not possible to confirm this with looking at the DDA calculations.

As it was not possible to conclusively determine if the aspect ratio of the rods had any affect on the DDA correction, the correction that was used was the same as for the spheres, where the correction factor was $\frac{1}{3}$.

Although Draine et al. suggests performing the DDA calculations for 2 inter-dipole spacings and then applying the correction, in this thesis it was calculated for 3 values and then the `data_fit` module was used to fit the 'dda_correction' fit function. This was done with 3 data point as the variability in data is high and only using 2 data points is expected to produce a significant error in the corrected values. Using even more data points would get a more accurate corrected value, however, the DDA calculations are slow and there was no extra time to perform these.

For spherical particles, Mie theory can be used to provide an exact solution to Maxwell's equations which can then be used as a reference to verify the DDA correction. This cannot be done for the nanorods as the Mie calculations do not extend to the hemispherical-capped cylinder geometry that is used to describe the nanorods. Importantly, after the correction is applied to the DDA calculation for spheres, the position of the plasmon peaks show an excellent agreement with what was found using Mie theory.

The final part that needs to be verified for the DDA calculation of nanorods, is the target geometry used.

The nanorods are known to have a pentagonal-twinned crystal structure, see figure 1.2, however a hemispheric-capped cylindrical geometry was used for the DDA calculations.

Guidez et al. used time-dependant density functional theory (TDDFT) to calculate the extinction spectra of pentagonal nanorods with various sizes [49]. Four longitudinal plasmon peaks were observed whereas there was only a single peak for the DDA calculations using the cylindrical nanorods. The longitudinal plasmon peaks were red shifted as the length of the rod was increased and the relationship was also found to be linear.

When viewing the plasmon peaks in the UV-Visible extinction spectra of the nanorod preparations, only two peaks are observed. However, due to multiple geometries and sizes of particles present in the preparations, it is expected that any fine structure in the extinction spectra would be lost.

The TEM images show evidence for the twin planes in the rods. However, the end caps were rounded and no sharp crystal facets were observed. Because of this, the hemispheric-capped cylindrical geometry is expected to provide an excellent approximation for the true geometry of the nanorods.

6.2.2 Relationship between the DDA and Gan's theory calculations

The Gan's theory calculations are approximately 10^7 times faster than the DDA calculations. The Gan's theory calculations can therefore be used to determine the relationship between the linear fit parameters and the refractive index of the solution, which would be impossible to perform using the DDA.

When looking at the Gan's theory calculations, it was found that they did not agree with that found using DDA; even though a linear relationship was still observed between the longitudinal plasmon peak position and the aspect ratio.

It was proposed that the aspect ratios of the prolate spheroid and the capsule geometry, which are used for the two calculations, are not equivalent.

A correction factor was determined based on the fact that the two geometries are expected to give the same result for an aspect ratio of 1, as both of the geometries are spheres.

However, when the intercepts of the two linear fits (equations 4.7 and 4.8) are examined, they are found to intercept when the aspect ratio is -1, as shown below:

$$\begin{aligned} 113 \times A + 277 &= 103.1 \times A + 267.3 \\ (113 - 103.1)A &= 277 - 267.3 \\ A &= \frac{9.7}{-9.9} = -0.98 \end{aligned} \tag{6.4}$$

It is not clear why the DDA and Gan's theory calculations give equivalent results when the aspect ratio is -1. The only definitive way to determine the relationship between the DDA and the Gan's results, is to perform DDA calculations using a prolate spheroid in order to conclusively determine if there is some relationship between the aspect ratios of the two geometries.

Despite not knowing the relationship between the two geometries, the proposed correction was used and the relationship between the longitudinal plasmon peak position and the refractive index of the surrounding medium is shown in figure 4.22.

6.2.3 Relationship between the aspect ratio and longitudinal plasmon peak position

The relationship between the aspect ratio of the nanorods and the longitudinal plasmon peak position of the nanorods was determine using the DDA calculations. This relationship for the peak found in the extinction spectrum was found to be:

$$\lambda_{ext} = (113 \pm 10) \times A \ + (277 \pm 20) \tag{6.5}$$

One example was found in literature where this relationship was also published [29]. This relationship did not agree with what was found in this thesis. The published relationship that was determined using DDA is:

$$\lambda_{ext} = (130.9 \pm 0.2) \times A \ + (285) \tag{6.6}$$

And it was found that this agreed with what was found experimentally, which was:

$$\lambda_{ext} = (132 \pm 6) \times A \ + (289) \tag{6.7}$$

One difference between these reported results and what is found in this thesis, is that a correction was not applied to the DDA calculations found in the literature. It has been shown that the correction resulted in a blue-shift of the plasmon peaks which may explain the difference found here. However, as the parameters used in the DDA calculation were not reported, the extent of this shift cannot be determined.

The disagreement with the experiment results could be explained by the presence of the capping molecules, which was PVP in this case. The capping molecules change the effective refractive index for the surrounding medium, which is not taken into consideration in the result reported in this thesis.

6.3 Preparation of silver nanorods

The plasmon peak position was around 380 nm for all solution prepared, however there was some deviation in both the position and amplitude for the plasmon peak. The silver nitrate solution is a possible candidate for the unexpected changes in the plasmon peaks, as it is known to oxidise to silver oxide. This reaction is catalysed by light. Because of this, the prepared silver nitrate solution was covered in tin-foil and kept in a dark cupboard until it was required.

The most likely reason for this non-reproducibility is due to the decomposition of the sodium borohydride solution. The concentration of this solution could also be incorrect due to the sodium borohydride absorbing moisture from the air when it is being weighed.

This variation in the seed solutions, lead to slight difference in the nanorods that were prepared using different seeds; which meant the nanorod preparations were not reproducible. Because of this, comparisons between the nanorod solutions were only made for nanorods prepared using the same seed solution.

6.3.1 Centrifugation of the nanorod solutions

Centrifugation was required to remove excess CTAB from the nanorod preparations as well as any smaller particles that failed to grow into nanorods. The excess CTAB would crystallise over time if it wasn't removed and was required to be removed entirely before TEM images could be obtained for the nanorod solutions.

After centrifugation, the majority of the nano particles were concentrated in the pellet at the bottom of the falcon tubes. This pellet was still a liquid and care was required when removing the supernatant. Not all of the supernatant could be removed without also removing part of the pellet and as a result multiple centrifugation steps were required to remove all of the CTAB. Once all of the CTAB was removed from the nanorod solutions, The nanorods were no longer stable and would precipitate out of the solution overtime.

The nanorod preparations with a γ values of 8 and 16, were expected to contain larger particles and therefore a slower centrifuge speed was required to separate them from the CTAB. This is the method that Jana et al. used [24]. For the other preparations, the centrifuge speed was set to the maximum.

For the preparations with γ values of 8 and 16, there was a clear difference in the longitudinal plasmon peaks positions after centrifugation. This is shown in figure 5.4. The plasmon peak of the pellet solution had a longer wavelength than in the supernatant solution, this suggests there is a separation based on geometry of the nanoparticles. However, the sum of the extinction spectra of of the pellet and supernatant solutions does not match that of the original solution, as seen in figure 5.3. Although such a change would be observed if there was a change in the geometry of the nanoparticles, this is expected to be a result of aggregation. One observation supporting this theory of aggregation is the fact that this was only seen in the preparations with large γ values which are expected to contain larger particles. If the centrifugation was causing a change in the particle geometries, it would be expected that this effect would be larger for the preparations with low γ, where the centrifugation speed was much faster. Another observation supporting the theory of aggregation can be seen in the time-resolved extinction spectra obtained during the growth of the nanorods. For the case of the preparation with larger γ values, the longitudinal plasmon peak position can be seen to shift to longer wavelengths as the growth progresses. However, at some point this peak starts to decrease in amplitude and shifts back to shorter wavelengths, which is what would be observed if the particles are aggregating.

This theory of aggregation is confirmed when looking at the extinction spectra of the nanorod preparations after a second centrifugation step. For the preparations with the large γ values, the longitudinal plasmon peak that shows up in the spectrum of the pellet solution after the first centrifugation step is only found in that of the supernatant after the second centrifugation, as seen in figure 5.6. Due to aggregation in the original preparation, the larger particle aggregates were separated from the solution and are found in the pellet during the first centrifugation step, however, in the second centrifugation step where the particles were redispersed and no aggregates were present, these particles are not separated from the solution and are left in the supernatant.

For the other preparations which were spun at the maximum speed, there was minimal change in the longitudinal plasmon peak position between the pellet,supernatant and the original solution. This suggests that there was no aggregation during the growth of these preparations, which is also consistent with what was observed in the time-resolved extinction spectra.

The amplitudes for both of the plasmon peaks decrease in the extinction spectrum of the supernatant as γ increases. At the same time, the amplitudes increase for the pellet solutions, suggesting the nanoparticles in the preparations with higher γ values are easier to separate out from the initial solution. This is consistent with the idea the the preparations with a higher γ values contain larger particles. This also agrees with the idea that the nanoparticles are grow primarily from the existing seed particles and new particles are not formed.

6.3.2 Trigonal platelets found in the transmission electron microscopy images

The TEM images of the nanorod preparations showed the presence of unwanted trigonal platelets and spherical particles in the nanorod preparations, and for some, the trigonal platelets appeared to be the dominant geometry.

Even though the TEM images show that these other geometries to be dominant, it is not clear that this is the case for the bulk solution of the preparation.

Very few particles were found in the TEM sample — much fewer than was expected. Therefore, the small sample size of the particles found in the TEM images cannot be used to reliably characterise the geometries found in the bulk solution.

One reason for this could be because of the way the TEM samples are prepared. The nanorods are prepared in an almost saturated CTAB solution. If this CTAB isn't removed before preparing the TEM samples, the CTAB will crystallise in the preparation and ruin the TEM images. Two methods were used to remove the CTAB: centrifugation and dialysis.

When all of the CTAB was removed with two centrifugation steps, the final pellet solutions appeared colourless and the amplitudes of the plasmon peaks had reduced considerably with the peaks also shifting to shorter wavelengths. Jana et al. reported that silver nanorods with aspect ratios less than 2.5 were unstable and in the absence of CTAB and reverted back to spheres. This might be what was observed, as the longitudinal plasmon peak for the preparation with γ value of 0.5, disappeared entirely.

When dialysis was used, it was clear that most of the nanoparticles had fallen out of suspension as the solution no longer had a visible colour and the dialysis tubing was stained a dark grey colour.

Centrifugation and dialysis might explain why the trigonal platelets are the dominant geometry present in the TEM images however it does not explain why the trigonal platelets are present at all. The CTAB used to stabilise the particles in the nanorod preparation should encourage the growth of the nanorods over other geometries as it preferentially binds to the (100) surface found on sides of the nanorods over the (111) surface of the trigonal platelets.

The crystal structure of the trigonal platelets also doesn't match that of the nanorods. The trigonal platelets have a single laminar twin plane in the crystal structure while the nanorods, and the seeds that they grow from, have 5-fold cyclic twin planes. The nanorods and trigonal platelets are therefore unlikely to grow from the same seed particles.

The geometries of the seed particles cannot be determined as the size of these particles is below the resolution limit of the TEM.

6.3.3 Trimeric clusters in the silver nitrate solutions

Xiong et al. reported the presence of trimeric clusters in silver nitrate powders and their solutions. These trimeric cluster then served as the nuclei for the formation of these trigonal platelets [50].

It was also found that when the production of Ag atoms was slow, the concentration of silver atoms in the solution was too low to allow the nucleation of new particles and therefore particle growth was limited to the trimeric clusters already present in solution.

Sodium borohydride is a strong reducing agent and the reduction is fast enough as to not promote the growth from these trimeric cluster in this way.

However, the formation of nanoparticles was visibly slowed if the sodium borohydride was cooled in ice water before use. This was done to slow the decomposition of the sodium borohydride but may have inadvertently slowed the reduction rate enough to promote the growth of these trigonal platelets. In light of this, the sodium borohydride was no longer chilled before use.

Like the pentagonal bipyramid geometry, the trigonal platelets are bound by the 111 plane and therefore would also be stabilised by the citrate molecules.

Xiong et al. also reported a photoluminescence emission band at 475 nm for the silver nitrate solution. This band slowly disappeared over time and mass spectrometry showed that the number of trigonal platelets also diminished over time, suggesting this photoluminescence band was due to the trimeric clusters.

When the photoluminescence spectrum was recorded for a silver nitrate solution which was used for the nanorod preparations, no photoluminescence band was observed and there was no change over a period of 2 hours.

6.3.4 Photochemical transformation of the seed particles

The TEM images for the nanoparticles after exposure to the lamp shows that the trigonal platelets and decahedral particles had grown considerably. However, the initial particles were too small to be examined in the TEM, therefore, it couldn't be confirmed that the geometries were transformed.

The UV-Visible extinction spectra, however, did show evidence for this transformation. With the two new plasmon peaks appearing which belong to the trigonal platelets and decahedral particles. This same evolution of the plasmon peak has been reported in literature and the two new peaks were found to be from the trigonal platelets and the decahedral particles [44][51][52].

Wang et al. reported that this transformation was dependant on the temperature of the solution. It was found that at 20 °C the transformation favoured only the decahedral particles, and the trigonal platelets started being favoured as the temperature was increased. At 40 °C, the trigonal platelets were favoured exclusively.

Although the temperature of the nanoparticle solution was not actively controlled, the air-conditioning in the lab was set to 19 °C. However, heating from the lamp may have raised the temperature of the solution.

Although the trigonal platelets were still present in the TEM images, it is expected that these particles would disappear upon further illumination with the lamp. This is because the plasmon peak associated with the platelets was still decreasing when the illumination was stopped and the presence of truncated platelets in the TEM images suggests they were still dissolving.

The resulting nanoparticle solution from the photochemical transformation was not suitable for the preparation of nanorods as the decahedral particles were too large to act as the seed particles for the rods. It has been reported that the wavelength of the lamp can be used to control the size of these particles, with the use of blue light resulting in the smallest particles [51][44]. This could not be done for this project as a suitable blue light source or the appropriate filter were not available.

Although the photochemical transformation proves to be an excellent candidate for removing the platelet and other geometries from the nanorod seed solution, the process was very time consuming and only a small volume of the seed solution could be transformed.

6.4 Modelling the growth of silver nanorods

The significant presence of trigonal-platelets in the nanorod preparations meant the the growth model derived in chapter 3 could not be used to explain the evolution of the longitudinal plasmon peak during the growth of the nanorod preparation; which was recording using time-resolved UV-Visible extinction spectroscopy.

The peak fitting and the time-resolved UV-Visible extinction spectra recorded during the growth of the nanorods provide the perfect situation for fitting this growth model. However, as the plasmon peaks are likely due to the trigonal platelets found in the solution rather than the rods, the DDA results describing the relationship between this plasmon peak position and the aspect ratio of the nanorods cannot be used

Fit functions for the `data_fit` module were created. For the 1-dimensional growth mode, the fit function is called
`'growth_kinetics_cylindrical_rod_1D'`.

There is also a model that has the refractive index of the surrounding medium as a fit parameter. This used the relationship between the longitudinal plasmon peak position and the refractive index that was obtained using the Gan's theory calculations. This fit function is called
`'growth_kinetics_cylindrical_rod_1D_nmed'`.

Both of thes fit function can be found in appendix C.6.

Chapter 7

Conclusions and future work

DDA results show that for silver spheres a correction is needed to be applied before the computed extinction spectra agrees with what is found using Mie theory. Such a correction has not been reported for nanorods. A script that counts the surface-dipoles was developed in an attempt to determine the convergence of the shape error in the DDA calculations. The results from this surface-dipole counting suggest that the correction that is required for nanorods is the same as is used for the spheres. However, further study of the convergence of the DDA calculations needs to be performed to verify that this correction is not dependant on the aspect ratio of the nanorods.

After applying this correction, the parameters of the plasmon peaks for the nanorods were determined using DDA and the linear fit parameters describing the relationship between the longitudinal plasmon peak position and the aspect ratio of the rods was determined.

After determining the relationship between the aspect ratios of a prolate spheroid and the capsule geometry used for the DDA calculations, Gan's theory was used to determine the effect that the refractive index of the surrounding medium has on the position of the longitudinal plasmon peak. However, it is not clear that the relationship between the geometries used in the DDA and Gan's theory calculations is correct. DDA calculations will need to be preformed on prolate spheroids to confirm this.

The time-resolved UV-Visible extinction spectra was recorded during the growth of the nanorod preparations and the parameters of the plasmon peaks were determined under various growth conditions.

TEM images of these nanorod preparations showed that the dominant geometry was not in fact nanorods but rather trigonal platelets. Because of this, the relationship between the plasmon peaks and aspect ratio of the rods could not be applied to the plasmon peaks measured during the growth of the nanorod preparations.

Photochemical transformation was looked into as a possible solution for stopping the growth of these platelets by altering the geometries of the particles found in the seed solution. TEM images along with UV-Visible extinction spectra showed that this method was successful in transforming the seed particles into the pentagonal bipyramids that are required for nanorod growth. However, more study of this method is required as the size of these particles need to be limited before they can be used in the preparation of nanorods.

Bibliography

[1] Ivana Angelini et al. “Chemical analyses of Bronze Age glasses from Frattesina di Rovigo, Northern Italy”. en. In: *Journal of Archaeological Science* 31.8 (Aug. 2004), pp. 1175–1184. ISSN: 0305-4403. DOI: 10.1016/j.jas.2004.02.015.

[2] Ian Freestone et al. “The Lycurgus Cup — A Roman nanotechnology”. en. In: *Gold Bulletin* 40.4 (Dec. 2007), pp. 270–277. ISSN: 2190-7579. DOI: 10.1007/BF03215599.

[3] Michael Faraday. “The Bakerian Lecture. —Experimental relations of gold (and other metals) to light”. In: *Philosophical Transactions of the Royal Society of London* 147 (Dec. 1857). Publisher: Royal Society, pp. 145–181. DOI: 10.1098/rstl.1857.0011.

[4] Peter P. Edwards and John Meurig Thomas. “Gold in a Metallic Divided State—From Faraday to Present-Day Nanoscience”. In: *Angewandte Chemie International Edition* 46.29 (2007). _eprint: https://onlinelibrary.wiley.com/d
pp. 5480–5486. ISSN: 1521-3773. DOI: 10.1002/anie.200700428.

[5] Gustav Mie. “Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen”. en. In: *Annalen der Physik* 330.3 (1908). _eprint: https://onlinelibrary.wiley.c
pp. 377–445. ISSN: 1521-3889. DOI: 10.1002/andp.19083300302.

[6] Thomas Wriedt. “Mie Theory: A Review”. en. In: *The Mie Theory: Basics and Applications*. Ed. by Wolfram Hergert and Thomas Wriedt. Springer Series in Optical Sciences. Berlin, Heidelberg: Springer, 2012, pp. 53–71. ISBN: 978-3-642-28738-1. DOI: `10.1007/978-3-642-28738-1_2`.

[7] Tamitake Itoh et al. “Surface-enhanced resonance Raman scattering and background light emission coupled with plasmon of single Ag nanoaggregates”. In: *The Journal of Chemical Physics* 124.13 (Apr. 2006). Publisher: American Institute of Physics, p. 134708. ISSN: 0021-9606. DOI: `10.1063/1.2177662`.

[8] Jiang et al. “Single Molecule Raman Spectroscopy at the Junctions of Large Ag Nanocrystals”. en. In: *The Journal of Physical Chemistry B* 107.37 (Sept. 2003), pp. 9964–9972. ISSN: 1520-6106, 1520-5207. DOI: `10.1021/jp034632u`.

[9] Eric C. Le Ru and Pablo G. Etchegoin. “Single-Molecule Surface-Enhanced Raman Spectroscopy”. In: *Annual Review of Physical Chemistry* 63.1 (2012). _eprint: https://doi.org/10.1146/annurev-physchem-032511-143757, pp. 65–87. DOI: `10.1146/annurev-physchem-032511-143757`.

[10] Cecilia Noguez. “Surface Plasmons on Metal Nanoparticles: The Influence of Shape and Physical Environment”. In: *The Journal of Physical Chemistry C* 111.10 (Mar. 2007). Publisher: American Chemical Society, pp. 3806–3819. ISSN: 1932-7447. DOI: `10.1021/jp066539m`.

[11] M. A. Garcia. “Surface plasmons in metallic nanoparticles: fundamentals and applications”. en. In: *Journal of Physics D: Applied Physics* 44.28 (June 2011), p. 283001. ISSN: 0022-3727. DOI: `10.1088/0022-3727/44/28/283001`.

[12] Luis M. Liz-Marzán*. *Tailoring Surface Plasmons through the Morphology and Assembly of Metal Nanoparticles.* en. review-article. Aug. 2005. DOI: `10.1021/la0513353`.

[13] Olivia Nicoletti et al. “Surface plasmon modes of a single silver nanorod: an electron energy loss study”. EN. In: *Optics Express* 19.16 (Aug. 2011), pp. 15371–15379. ISSN: 1094-4087. DOI: `10.1364/OE.19.015371`.

[14] Huei-Huei Chang and Catherine J. Murphy. “Mini Gold Nanorods with Tunable Plasmonic Peaks beyond 1000 nm”. In: *Chemistry of Materials* 30.4 (Feb. 2018), pp. 1427–1435. ISSN: 0897-4756. DOI: `10.1021/acs.chemmater.7b05310`.

[15] Hajar Zareyi and Majid Vaezzadeh. “Geometry Effect on Plasmon Frequency in Triangular Nanoprism”. en. In: *Plasmonics* (Jan. 2018), pp. 1–7. ISSN: 1557-1955, 1557-1963. DOI: `10.1007/s11468-017-0689-8`.

[16] Seog-Jin Jeon et al. “Synthesis of Monodisperse Single Crystalline Ag Microcubes via Seed-Mediated Growth”. In: *Crystal Growth & Design* 17.1 (Jan. 2017), pp. 284–289. ISSN: 1528-7483. DOI: `10.1021/acs.cgd.6b01523`.

[17] Anne Myers Kelley. “A molecular spectroscopic view of surface plasmon enhanced resonance Raman scattering”. eng. In: *The Journal of Chemical Physics* 128.22 (June 2008), p. 224702. ISSN: 1089-7690. DOI: `10.1063/1.2931540`.

[18] Akash Kumar, Nabojit Das, and Raja Gopal Rayavarapu. “Role of Tunable Gold Nanostructures in Cancer Nanotheranostics: Implications on Synthesis, Toxicity, Clinical Applications and Their Associated Opportunities and Challenges”. en. In: *Journal of Nanotheranostics* 4.1 (Mar. 2023). Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, pp. 1–34. ISSN: 2624-845X. DOI: `10.3390/jnt4010001`.

[19] Jeremy B. Vines et al. "Gold Nanoparticles for Photothermal Cancer Therapy". In: *Frontiers in Chemistry* 7 (2019). ISSN: 2296-2646.

[20] Bolla G. Rao, Deboshree Mukherjee, and Benjaram M. Reddy. "Chapter 1 - Novel approaches for preparation of nanoparticles". en. In: *Nanostructures for Novel Therapy*. Ed. by Denisa Ficai and Alexandru Mihai Grumezescu. Micro and Nano Technologies. Elsevier, Jan. 2017, pp. 1–36. ISBN: 978-0-323-46142-9. DOI: `10.1016/B978-0-323-46142-9.00001-3`.

[21] Benjamin J. Wiley et al. "Maneuvering the surface plasmon resonance of silver nanostructures through shape-controlled synthesis". eng. In: *The Journal of Physical Chemistry. B* 110.32 (Aug. 2006), pp. 15666–15675. ISSN: 1520-6106. DOI: `10.1021/jp0608628`.

[22] P. M. Ajayan and L. D. Marks. "Quasimelting and phases of small particles". In: *Physical Review Letters* 60.7 (Feb. 1988), pp. 585–587. DOI: `10.1103/PhysRevLett.60.585`.

[23] José M. Gisbert-González et al. "Why Citrate Shapes Tetrahedral and Octahedral Colloidal Platinum Nanoparticles in Water". In: *The Journal of Physical Chemistry C* 122.33 (Aug. 2018). Publisher: American Chemical Society, pp. 19004–19014. ISSN: 1932-7447. DOI: `10.1021/acs.jpcc.8b05195`.

[24] Nikhil R. Jana, Latha Gearheart, and Catherine J. Murphy. "Wet chemical synthesis of silver nanorods and nanowires of controllable aspect ratio". In: *Chemical Communications* 7 (2001), pp. 617–618. ISSN: 13597345, 1364548X. DOI: `10.1039/b100521i`.

[25] Nikhil R. Jana, Latha Gearheart, and Catherine J. Murphy. "Wet Chemical Synthesis of High Aspect Ratio Cylindrical Gold Nanorods". In: *The Journal of Physical Chemistry B* 105.19 (May 2001). Pub-

lisher: American Chemical Society, pp. 4065–4067. ISSN: 1520-6106. DOI: `10.1021/jp0107964`.

[26] Babak Nikoobakht and Mostafa A. El-Sayed. "Evidence for Bilayer Assembly of Cationic Surfactants on the Surface of Gold Nanorods". In: *Langmuir* 17.20 (Oct. 2001), pp. 6368–6374. ISSN: 0743-7463. DOI: `10.1021/la010530o`.

[27] Danielle K. Smith and Brian A. Korgel. "The Importance of the CTAB Surfactant on the Colloidal Seed-Mediated Synthesis of Gold Nanorods". In: *Langmuir* 24.3 (Feb. 2008), pp. 644–649. ISSN: 0743-7463. DOI: `10.1021/la703625a`.

[28] Biao Jin et al. "Understanding Anisotropic Growth of Au Penta-Twinned Nanorods by Liquid Cell Transmission Electron Microscopy". In: *The Journal of Physical Chemistry Letters* 10.7 (Apr. 2019). Publisher: American Chemical Society, pp. 1443–1449. DOI: `10.1021/acs.jpclett.9b00164`.

[29] Mahmoud A. Mahmoud and Mostafa A. El-Sayed. "Different Plasmon Sensing Behavior of Silver and Gold Nanorods". In: *The Journal of Physical Chemistry Letters* 4.9 (May 2013). Publisher: American Chemical Society, pp. 1541–1545. DOI: `10.1021/jz4005015`.

[30] Sixten Bergman. *analytic_wfm.py*. Sept. 2017.

[31] Eli Billauer. *peakdet: Peak detection using MATLAB (non-derivative local extremum, maximum, minimum).*

[32] *scipy.optimize.leastsquares - SciPy v 1.10.0 Manual.* `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html`.

[33] P. B. Johnson and R. W. Christy. “Optical Constants of the Noble Metals”. In: *Physical Review B* 6.12 (Dec. 1972), pp. 4370–4379. DOI: `10.1103/PhysRevB.6.4370`.

[34] *Refractive index of Ag (Silver) - Johnson.* `https://refractiveindex.info/?shelf=main&book=Ag&page=Johnson`.

[35] Craig F. Bohren and Donald R. Huffman. *Absorption and Scattering of Light by Small Particles.* en. 1st ed. Wiley, Apr. 1998. ISBN: 978-0-471-29340-8 978-3-527-61815-6. DOI: `10.1002/9783527618156`.

[36] Christian Mätzler. *MATLAB Function for Mie Scattering and Absorption.* Tech. rep. University of Bern: Institute of Applied Physics, June 2002.

[37] Benjamin Sumlin. *PyMieScatt/index.rst at master · bsumlin/PyMieScatt.* en.

[38] Benjamin Sumlin. *Online user’s guide for the Python Mie Scattering package (PyMieScatt).* en.

[39] Edward M. Purcell and Carlton R. Pennypacker. “Scattering and Absorption of Light by Nonspherical Dielectric Grains”. In: *The Astrophysical Journal* 186 (Dec. 1973). ADS Bibcode: 1973ApJ...186..705P, pp. 705–714. ISSN: 0004-637X. DOI: `10.1086/152538`.

[40] B. T. Draine. “GRAPHITE REVISITED”. In: *The Astrophysical Journal* 831.1 (Oct. 2016), p. 109. ISSN: 1538-4357. DOI: `10.3847/0004-637X/831/1/109`.

[41] Bruce T. Draine and Piotr J. Flatau. “Discrete-Dipole Approximation For Scattering Calculations”. EN. In: *JOSA A* 11.4 (Apr. 1994), pp. 1491–1499. ISSN: 1520-8532. DOI: `10.1364/JOSAA.11.001491`.

[42] *Downloads - DDSCAT.*

[43] Daan P. Sprünken et al. “Influence of the Local Environment on Determining Aspect-Ratio Distributions of Gold Nanorods in Solution Using Gans Theory”. In: *The Journal of Physical Chemistry C* 111.39 (Oct. 2007), pp. 14299–14306. ISSN: 1932-7447, 1932-7455. DOI: `10.1021/jp074500r`.

[44] Brendan Pietrobon and Vladimir Kitaev. “Photochemical Synthesis of Monodisperse Size-Controlled Silver Decahedral Nanoparticles and Their Remarkable Optical Properties”. en. In: *Chemistry of Materials* 20.16 (Aug. 2008), pp. 5186–5190. ISSN: 0897-4756, 1520-5002. DOI: `10.1021/cm800926u`.

[45] B. T. Draine and P. J. Flatau. *User Guide for the Discrete Dipole Approximation Code DDSCAT 7.3*. Number: arXiv:1305.6497 arXiv:1305.6497 [astro-ph, physics:cond-mat, physics:physics]. May 2013. DOI: `10.48550/arXiv.1305.6497`.

[46] M. A. Yurkin and A. G. Hoekstra. “The discrete dipole approximation: An overview and recent developments”. en. In: *Journal of Quantitative Spectroscopy and Radiative Transfer*. IX Conference on Electromagnetic and Light Scattering by Non-Spherical Particles 106.1 (July 2007), pp. 558–589. ISSN: 0022-4073. DOI: `10.1016/j.jqsrt.2007.01.034`.

[47] Iván O. Sosa, Cecila Noguez, and Rubén G. Barrera. “Optical Properties of Metal Nanoparticles with Arbitrary Shapes”. In: *The Journal of Physical Chemistry B* 107.26 (July 2003). Publisher: American Chemical Society, pp. 6269–6275. ISSN: 1520-6106. DOI: `10.1021/jp0274076`.

[48] E. Stefan Kooij and Bene Poelsema. “Shape and size effects in the optical properties of metallic nanorods”. In: *Phys. Chem. Chem. Phys.* 8.28 (July 12, 2006), pp. 3349–3357. ISSN: 1463-9084. DOI: `10.1039/B518389H`.

[49] Emilie B. Guidez and Christine M. Aikens. "Diameter Dependence of the Excitation Spectra of Silver and Gold Nanorods". In: *The Journal of Physical Chemistry C* 117.23 (June 13, 2013), pp. 12325–12336. ISSN: 1932-7447, 1932-7455. DOI: `10.1021/jp4023103`.

[50] Yujie Xiong et al. "Trimeric clusters of silver in aqueous AgNO3 solutions and their role as nuclei in forming triangular nanoplates of silver". eng. In: *Angewandte Chemie (International Ed. in English)* 46.26 (2007), pp. 4917–4921. ISSN: 1433-7851. DOI: `10.1002/anie.200700942`.

[51] P. E. Cardoso-Avila et al. "Photochemical transformation of silver nanoparticles by combining blue and green irradiation". en. In: *Journal of Nanoparticle Research* 17.3 (Mar. 2015), p. 160. ISSN: 1572-896X. DOI: `10.1007/s11051-015-2920-x`.

[52] Haitao Wang et al. "Transformation from Silver Nanoprisms to Nanodecahedra in a Temperature-Controlled Photomediated Synthesis". In: *The Journal of Physical Chemistry C* 116.45 (Nov. 2012), pp. 24268–24273. ISSN: 1932-7447. DOI: `10.1021/jp304941b`.

Appendices

Appendix A

DDSCAT configuration files

A.1 Example .par file for spherical silver target

The following is the .par file for a silver sphere with a diameter of 10 nm, a dipole size of 0.2 nm and the wavelength is 400 nm.

```
===Generated by ScatPy (Sat Feb  3 10:15:15 2018)===
**** Preliminaries ****
NOTORQ
PBCGS2
GPFAFT
GKDLDR
NOTBIN
**** Initial Memory Allocation ****
100 100 100
**** Target Geometry and Composition ****
ELLIPSOID
```

```
50 50 50
1
'/home/robert/Documents/Analysis/ddscat/mp/Ag_christy'
**** Additional Nearfield calculation? ****
0
0. 0. 0. 0. 0. 0.
**** Error Tolerance ****
1e-05
**** maximum number of iterations allowed ****
10000
**** Interaction cutoff parameter for PBC calculations ****
0.01
**** Angular resolution for calculation of <cos>, etc. ****
0.5
**** Vacuum wavelengths (micron) ****
0.4 0.4 1 LIN
**** Refractive index of ambient medium
1.33
**** Effective Radii (micron) ****
0.005 0.005 1 LIN
**** Define Incident Polarizations ****
(0.000000, 0.000000)  (1.000000, 0.000000)  (0.000000, 0.000000)
2
**** Specify which output files to write ****
1
**** Prescribe Target Rotations ****
0.000000  0.000000  1
0.000000  0.000000  1
```

```
0.000000  0.000000  1
**** Specify first IWAV, IRAD, IORI (normally 0 0 0) ****
0, 0, 0
**** Select Elements of S_ij Matrix to Print ****
6
11 12 21 22 31 41
**** Specify Scattered Directions ****
LFRAME
2
0.000000  0.000000  180.000000  5
90.000000  0.000000  180.000000  5
```

A.2 Example .par file for a silver capsule target

The following is the .par file for a silver nanorod with a width of 10 nm and a cylinder length of 10 nm. The dipole size is 0.2 nm and the wavelength is 400 nm.

```
===Generated by ScatPy (Sat Feb  3 10:15:15 2018)===
**** Preliminaries ****
NOTORQ
PBCGS2
GPFAFT
GKDLDR
NOTBIN
**** Initial Memory Allocation ****
100 100 100
**** Target Geometry and Composition ****
```

```
CYLNDRCAP
50 50 0
1
'/home/robert/Documents/Analysis/ddscat/mp/Ag_christy'
**** Additional Nearfield calculation? ****
0
0. 0. 0. 0. 0. 0.
**** Error Tolerance ****
1e-05
**** maximum number of iterations allowed ****
10000
**** Interaction cutoff parameter for PBC calculations ****
0.01
**** Angular resolution for calculation of <cos>, etc. ****
0.5
**** Vacuum wavelengths (micron) ****
0.4 0.4 1 LIN
**** Refractive index of ambient medium
1.33
**** Effective Radii (micron) ****
0.006786 0.006786 1 LIN
**** Define Incident Polarizations ****
(0.000000, 0.000000)  (1.000000, 0.000000)  (0.000000,
0.000000)
2
**** Specify which output files to write ****
1
**** Prescribe Target Rotations ****
0.000000  0.000000  1
```

```
0.000000  90.000000  5
0.000000  0.000000  1
**** Specify first IWAV, IRAD, IORI (normally 0 0 0) ****
0, 0, 0
**** Select Elements of S_ij Matrix to Print ****
6
11 12 21 22 31 41
**** Specify Scattered Directions ****
LFRAME
2
0.000000  0.000000  180.000000  5
90.000000  0.000000  180.000000  5
```

Appendix B

Surface-dipole counting code

B.1 Python script for surface-dipole counting

```python
import csv
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
import os
import pickle
import subprocess
import math

from tikzplotlib import save as tikz_save

def load_fit_cache(name):
    if os.path.exists(f'cache/{name}'):
        print(f'Loaded from cache: cache/{name}')
        with open(f'cache/{name}', 'rb') as handle:
```

```
            b = pickle.load(handle)
        return b
    else:
        return False

def save_fit_cache(name,result):
    with open(f'cache/{name}', 'wb') as handle:
        pickle.dump(result, handle,
        protocol=pickle.HIGHEST_PROTOCOL)

use_cache = True

lPrimes = range(1,30,1)
lPrimes = [30]
widths = range(5,15,1)
widths=[5]

ds = range(10,101,1)
ds = [x/100 for x in ds]

for width in widths:
    lengths = [width+x for x in lPrimes]
    for length in lengths:
        results = []
        vol = 3/4 * math.pi * (width/2)**3 +
        (math.pi*(width/2)**2 * length)

        for d in ds:
            width_d = width/d
```

```python
        length_d = length/d

        surfaceN = 0
        surface_cap = 0
        surface_mid = 0
        totalN = 0

        name = f'rod_{width}_{length}_{d}.cache'
        if cache := load_fit_cache(name) and use_cache:
            surfaceN, totalN = load_fit_cache(name)
        else:
            numCells = int(1.10 * width/d)
            numCellsLength = int(1.10 * (length+width)/d)

            xMax = numCells
            yMax = numCellsLength
            zMax = numCells

            centre = (xMax/2,yMax/2,zMax/2)

            result = subprocess.Popen(["path/to/rust/excecutable", str(int(d*100)), str(length), str(width), str(xMax),str(yMax),str(zMax)],text=True,stdout=subprocess.PIPE)
            result.wait()
            out, err = result.communicate()

            rust_result = [int(x) for x in out.split(',')]

```

```
                save_fit_cache(name,(surfaceN, totalN, surface_mid, surface_cap))


            totalVol = totalN*d**3
            d = np.cbrt(vol/totalN)

            results.append([d, surfaceN, totalN, surfaceN/totalN])


        results = np.array(results)


        with open('data/rod_results_{}x{}nm.csv'.format(width,length),'w', newline='') as f:
            writer=csv.writer(f)
            for row in results:
                writer.writerow(row)

```

B.2 Rust source code

```
use std::env;

fn main() -> std::io::Result<()> {

    let args: Vec<String> = env::args().collect();
```

```

    let d: f32 = args[1].parse::<i32>().unwrap() as f32 /100_f32;

    let length: f32 = args[2].parse::<i32>().unwrap() as f32 / d;
    let width: f32 = args[3].parse::<i32>().unwrap() as f32 / d;

    let x_max: i32 = args[4].parse::<i32>().unwrap();
    let y_max: i32 = args[5].parse::<i32>().unwrap();
    let z_max: i32 = args[6].parse::<i32>().unwrap();

    let centre_x: f32 = x_max as f32 /2_f32;
    let centre_y: f32 = y_max as f32 /2_f32;
    let centre_z: f32 = z_max as f32 /2_f32;

    let mut _n_total = 0;
    let mut _n_surface = 0;
    let mut _n_cap = 0;
    let mut _n_mid = 0;

    for i in 0..x_max+1 {
        for j in 0..y_max+1 {
            for k in 0..z_max+1 {
                let x = i as f32 - centre_x;
                let y = j as f32 - centre_y;
                let z = k as f32 - centre_z;

```

```
                if y.abs() < (length-width)/2_f32 && (x.powi(2)+z.powi(2)).sqrt() <= width/2_f32 {
                    _n_total += 1;
                    if (x.powi(2)+z.powi(2)).sqrt() >= ((width/2_f32) - 1_f32) {
                        _n_surface += 1;
                        _n_mid += 1;
                    }
                }
                else if (x.powi(2)+z.powi(2)+(
                    y.abs()-(length-width)/2_f32
                ).powi(2)).sqrt() <= width/2_f32 {
                    _n_total += 1;
                    if (x.powi(2)+z.powi(2)+(
                        y.abs()-(length-width)/2_f32
                    ).powi(2)).sqrt() >= ((width/2_f32) - 1_f32) {
                        _n_surface += 1;
                        _n_cap += 1;
                    }
                }
            }
        }
    }
    let result = vec![_n_surface,_n_total,_n_mid,_n_cap];
    print!("{},{},{},{}",_n_surface,_n_total,_n_mid,_n_cap);
    Ok(())
}

```

Appendix C

Data fit python package

C.1 Data fit class

```
import csv
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from data_processing.datafit.Function import FitFunctionFactory
import scipy.stats as stats

class data_fit():

    def __init__(self):
        self.__data_x = None
        self.__data_y = None
        self.__functions = []
        self.__optimise_params = []
        self.__optimise_bounds = []
```

```

    def set_data(self, data):
        data = np.array(data)
        self.__data_x = data[:,0]
        self.__data_y = data[:,1]
        return self

    def check_data_set(self):
        if self.__data_x is None:
            raise RuntimeError("__data_x is not set")
        if self.__data_y is None:
            raise RuntimeError("__data_y is not set")


    def add_function(self, function, *args, **kwargs):
        self.__functions.append(FitFunctionFactory(function,
        *args, **kwargs))

    def get_optimise_params(self):
        optimise_params = []
        optimise_bounds_min = []
        optimise_bounds_max = []
        for function in self.__functions:
            if function.optimise:
                optimise_params +=
                function.get_params().values()
                optimise_bounds_min += function.get_bounds()[0]
                optimise_bounds_max += function.get_bounds()[1]
        self.__optimise_params = optimise_params
```

```
        self.__optimise_bounds = [optimise_bounds_min,
        optimise_bounds_max]
        return tuple(optimise_params)

    def update_function_params(self,params):
        for function in self.__functions:
            function_params, params =
            np.split(params,[function.get_num_params()])
            function.set_params(function_params)

    def set_function_params_std(self,paramsStd):
        for function in self.__functions:
            function_paramsStd, paramsStd =
            np.split(paramsStd,[function.get_num_params()])
            function.set_params_std(function_paramsStd.tolist())

    def optimise(self,**kwargs):
        self.check_data_set()
        self.get_optimise_params()
        popt, pcov = curve_fit(self.build_fit , self.__data_x,
        self.__data_y, p0=self.get_optimise_params(),
        bounds=self.__optimise_bounds, **kwargs)
        # popt, pcov = curve_fit(self.build_fit ,
        self.__data_x, self.__data_y, p0=[1,1,1])
        self.update_function_params(popt)

        # chisq = (self.get_residual()**2).sum()
        # dof = np.shape(self.__data_x)[0] - len(popt)
        # factor = chisq/dof
```

```python
        # pcov = [x/factor for x in pcov]

        self.set_function_params_std(np.sqrt(np.diag(pcov)))

    def build_fit(self, x, *params):
        self.update_function_params(params)
        y = np.zeros_like(x)
        for function in self.__functions:
            y += function.build_function(x)
        return y

    def get_residual(self):
        return self.__data_y - self.build_fit(self.__data_x,
        *self.get_optimise_params())


    def plot_data(self,show=True, **kwargs):
        self.check_data_set()
        plt.plot(self.__data_x,self.__data_y,'k',**kwargs)
        if show:
            plt.show()

    def plot_functions(self,show=True, **kwargs):
        self.check_data_set()
        for function in self.__functions:
            plt.plot(self.__data_x,function.build_function(self.__data_x),**kwargs
        if show:
            plt.show()

```

```
    def plot_fit(self, show=True, **kwargs):
        self.check_data_set()
        plt.plot(self.__data_x,self.build_fit(self.__data_x,
        *self.get_optimise_params()), **kwargs)
        if show:
            plt.show()

    def plot_residual(self,show=True):
        self.check_data_set()
        plt.plot(self.__data_x,self.get_residual())
        if show:
            plt.show()

    def plot_residual_dist(self,normalDist=False, show=True,
    *args, **kwargs):
        self.check_data_set()
        residuals = self.get_residual()
        plt.hist(residuals,*args,**kwargs)
        if normalDist:
            x = np.linspace(min(residuals), max(residuals),
            100)
            pdf = stats.norm.pdf(x, np.mean(residuals),
            np.std(residuals))
            plt.plot(x,pdf,'-')
        if show:
            plt.show()

    def save_plot(self,fileLocation, cla=True, **kwargs):
        plt.savefig('{}'.format(fileLocation), **kwargs)
```

```
        if cla:
            plt.cla()

    def get_functions(self,returnObjects=False):
        if returnObjects:
            return self.__functions
        functions = []
        for function in self.__functions:
            functions.append([type(function).__name__] +
            list(function.get_params().values()))
        return functions

    def get_function_std(self):
        functionsStd = []
        for function in self.__functions:
            functionsStd.append([type(function).__name__] +
            function.get_params_std())
        return functionsStd

    def get_ci(self,std, ci, dof, ssr):
        SE = np.sqrt( ssr / dof) * std
        return abs(stats.t.ppf((100-ci)/200, dof)) * SE

    def get_ci_list(self,std, ci, dof, ssr):
        return [self.get_ci(x, ci, dof, ssr) for x in std]

    def get_function_cis(self,ci):
        std = self.get_function_std()[0][1:]
        stats = self.get_stats()
```

```
        ssr = stats[3][1]
        dof = stats[2][1]

        functionsCI = []
        for function in self.__functions:
            std = function.get_params_std()
            functionsCI.append([type(function).__name__,
            *self.get_ci_list(std,ci,dof,ssr)])
        return functionsCI

    def save_functions(self,fileLocation):
        with open('{}'.format(fileLocation), 'w', newline='')
        as file:
            writer = csv.writer(file, delimiter=',',
                                    quotechar='|',
                                    quoting=csv.QUOTE_MINIMAL)
            for function in self.__functions:
                writer.writerow([type(function).__name__] +
                list(function.get_params().values()) +
                function.get_params_std())
        return self

    def save_functions_ci(self,fileLocation,ci):
        std = self.get_function_std()[0][1:]
        stats = self.get_stats()
        ssr = stats[3][1]
        dof = stats[2][1]

```

```
        with open('{}'.format(fileLocation), 'w', newline='')
        as file:
            writer = csv.writer(file, delimiter=',',
                                    quotechar='|',
                                    quoting=csv.QUOTE_MINIMAL)
            for function in self.__functions:
                writer.writerow([type(function).__name__] +
                list(function.get_params().values()) +
                self.get_ci_list(std,ci,dof,ssr))
        return self

    def get_stats(self):
        nData = len(self.__data_x)
        nParams = len(self.__optimise_params)
        dof = nData - nParams
        sumSqrRes = np.sum(np.power(self.get_residual(), 2))

        return [['num data points', nData], ['num parameters',
        nParams], ['DOF', dof], ['sum squared residuals',
        sumSqrRes]]

    def save_stats(self,fileLocation):
        stats = self.get_stats()

        with open('{}'.format(fileLocation), 'w', newline='')
        as file:
            writer = csv.writer(file, delimiter=',',
                                    quotechar='|',
                                    quoting=csv.QUOTE_MINIMAL)
```

```
            writer.writerows(stats)
        return self
```

C.2 FitFunction and factory class

```
import importlib
from abc import ABC, abstractmethod

class FitFunctionFactory(object):
    def __new__(self,function,*args,**kwargs):
        FitFunction = None
        try:
            module =
            importlib.import_module('.functions.{}'.format(function),
            'data_processing.datafit')
            FitFunction = getattr(module,
            function)(*args,**kwargs)

        except ImportError as error:
            print('Error importing data fit function: ' +
            function + '\nError: ' + repr(error))

        return FitFunction

class FitFunction(ABC):

    optimise = True
    _params = []
    _bounds = []
```

```

    @property
    def _paramNames(self): raise NotImplementedError

    @property
    def _defaultBounds(self): raise NotImplementedError

    def __init__(self):
        self.set_bounds(self._defaultBounds)
        self.set_params_std([0]*self.get_num_params())

    def get_num_params(self):
        return len(self._paramNames)

    def set_params(self,params):
        self._params = dict(zip(self._paramNames, params))
        return self

    def get_params(self):
        return self._params

    def set_params_std(self, std):
        self._paramsStd = std
        return self

    def get_params_std(self):
        return self._paramsStd

    def set_bounds(self,bounds):
```

```
        bounds_min = []
        bounds_max = []
        for b in bounds:
            bounds_min += [b[0]]
            bounds_max += [b[1]]
        self._bounds = [bounds_min,bounds_max]
        return self

    def get_bounds(self):
        return self._bounds

    def set_optimise(self,optimise):
        self.optimise = optimise
        return self

    @abstractmethod
    def build_function(self): raise NotImplementedError
```

C.3 Spectral peak fitting functions

C.3.1 Wavelength-domain Lorentzian

```
import numpy as np
from ..Function import FitFunction

class lorentzian_wavelength(FitFunction):
    _paramNames = ('position','amplitude','width')
    _defaultBounds = ((0,float('inf')),
                      (0,float('inf')),
```

```
                    (0,float('inf')))

    def __init__(self, position, amplitude, width,
    bounds=None):
        FitFunction.__init__(self)
        self.set_params([position, amplitude, width])
        if bounds:
            self.set_bounds(bounds)

    def build_function(self,x):
        x = np.array(x)
        y = self._params['amplitude'] / (1 + np.power((((1/x) -
        (1/self._params['position']))/
        self._params['width']),2))
        return y
```

C.3.2 Wavelength-domain Gaussian

```
import numpy as np
from ..Function import FitFunction

class gaussian_wavelength(FitFunction):
    _paramNames = ('position','amplitude','width')
    _defaultBounds = ((0,float('inf')),
                    (0,float('inf')),
                    (0,float('inf')))

    def __init__(self, position, amplitude, width,
    bounds=None):
```

```
        FitFunction.__init__(self)
        self.set_params([position, amplitude, width])
        if bounds:
            self.set_bounds(bounds)


    def build_function(self,x):
        x = np.array(x)
        y = self._params['amplitude'] * np.exp( -(((1/x) -
        (1/self._params['position']))/self._params['width'])**2)
        return y
```

C.3.3 Wavelength-domain Pseudo Voigt

```
import numpy as np
from ..Function import FitFunction

class voigt_pseudo_wavelength(FitFunction):
    _paramNames = ('position','amplitude','width','eta')
    _defaultBounds = ((0,float('inf')),
                    (0,float('inf')),
                    (0,float('inf')),
                    (0,1))

    def __init__(self, position, amplitude, width, bounds):
        FitFunction.__init__(self)
        self.set_params([position, amplitude, width, 0.5])
        if bounds:
            bounds.append([0,1])
```

```
            self.set_bounds(bounds)

    def build_function(self,x):
        x = np.array(x)
        y = self._params['amplitude'] * ( self._params['eta'] *
        (1 / (1 + np.power((((1/x) -
        (1/self._params['position']))/
        self._params['width']),2))) + (1 - self._params['eta'])
        * (np.exp( -(((1/x) -
        (1/self._params['position']))/self._params['width'])**2))
        )

        return y
```

C.4 Linear fit function

```
import numpy as np
from ..Function import FitFunction

class linear(FitFunction):
    _paramNames = ('m','c')
    _defaultBounds = ((-float('inf'),float('inf')),
                    (-float('inf'),float('inf')))

    def __init__(self,  bounds=None):
        FitFunction.__init__(self)
        self.set_params([1, 1, ])
        if bounds:
            self.set_bounds(bounds)
```

```


    def build_function(self,x):
        x = np.array(x)
        y = self._params['c'] + self._params['m']*x
        return y
```

C.5 DDA correction Functions

C.5.1 Function used to apply the DDA correction

```
import numpy as np
from ..Function import FitFunction

class dda_correction(FitFunction):
    _paramNames = ('Qinf','A')
    _defaultBounds = ((-float('inf'),float('inf')),
                    (-float('inf'),float('inf')))

    def __init__(self,  bounds=None):
        FitFunction.__init__(self)
        self.set_params([1, 1, ])
        if bounds:
            self.set_bounds(bounds)


    def build_function(self,x):
        x = np.array(x)
        y = self._params['Qinf'] - self._params['A']*x**(-1/3)
```

```
        return y
```

C.5.2 Function used to find the DDA correction parameter χ

```
import numpy as np
from ..Function import FitFunction

class dda_correction_general(FitFunction):
    _paramNames = ('Qinf','A','factor')
    _defaultBounds = ((-float('inf'),float('inf')),
                      (-float('inf'),float('inf')),
                      (1,6))

    def __init__(self,  bounds=None):
        FitFunction.__init__(self)
        self.set_params([1, 1, 3])
        if bounds:
            self.set_bounds(bounds)


    def build_function(self,x):
        x = np.array(x)
        y = self._params['Qinf'] -
        self._params['A']*x**(-1/self._params['factor'])
        return y
```

C.6 Growth Model Functions

C.6.1 1-dimensional nanorod growth kinetics

```
import numpy as np
from ..Function import FitFunction

class growth_kinetics_cylindrical_rod_1D(FitFunction):
    _paramNames = ('width','rateConstant')
    _defaultBounds = ((0,100),
                      (0,10),)

    def __init__(self, C0, r, bounds=None):
        FitFunction.__init__(self)
        self.set_params([10, 1])
        if bounds:
            self.set_bounds(bounds)

        # r = 1/r

        C0 = C0/1000

        self.__b = 1/ (3/2 * r * C0)

    def build_function(self,t):
        t = np.array(t)

        y =  self.__b + 1 - (self.__b)* np.exp(-t * (
        self._params['rateConstant'])/(self.__b*self._params['width']))

```

```
        return y
```

C.6.2 1-dimensional nanorod growth kinetics with n_m fit

```
import numpy as np
from ..Function import FitFunction
from scipy.interpolate import interp1d


class growth_kinetics_cylindrical_rod_1D_nmed(FitFunction):
    _paramNames = ('width','rateConstant','nmed')
    _defaultBounds = ((0,10),
                    (0,1),
                    (1,2),)

    def __init__(self, C0, r, bounds=None):
        FitFunction.__init__(self)
        # self.set_params([10, 100000, -4, 0])
        self.set_params([10, 0.1,1.8])
        if bounds:
            self.set_bounds(bounds)

        # r = 1/r

        C0 = C0/1000

        self.__b = 1/ (3/2 * r * C0)
        print('b')
```

```
        print(self.__b)

        self.load_conversion_parameters()

    def load_conversion_parameters(self):
        with
        open('/home/robert/Documents/Masters/Gans/data/fit_params_nmed_transformed
        as f:
            next(f)
            data = [x.strip('\n').split(',') for x in list(f)]

        data = np.array(data).astype('float')
        self.__slopeSpline = interp1d(data[:,0],data[:,1])
        self.__interceptSpline = interp1d(data[:,0],data[:,2])

    def get_conversion_slope(self):
        return self.__slopeSpline(self._params['nmed'])

    def get_conversion_intercept(self):
        return self.__interceptSpline(self._params['nmed'])

    def calc_conversion(self,R):
        return self.get_conversion_slope()*R +
        self.get_conversion_intercept()


    def build_function(self,t):
        t = np.array(t)

```

```
        R =  self.__b + 1 - (self.__b)* np.exp(-t * (self._params['rateConstant'])/(self.__b*self._params['width']))

        y = self.calc_conversion(R)
        return y
```