

Characterisation and calibration of Optical tweezers

av

René Christian Castberg

MASTER THESIS

for the degree

Master in Condensed Matter Physics

(Master of Science)



Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo

June 2008

Faculty of Mathematics and Natural Sciences
University of Oslo

Contents

1	Introduction	2
2	Theory of Optical tweezers	2
2.1	Initial development of the Optical tweezer	2
2.2	Single Beam Gradient Trap	3
2.2.1	Mie theory of trapping	3
2.2.2	Rayleigh Scattering	6
2.3	Brownian Motion	8
2.4	Langevin Equation	9
2.4.1	Using the Langevin equation of motion to obtain an expression for the power spectrum for a trapped particle.	10
2.5	Obtaining the power spectra via the Wiener-Khintchine and equipartition	11
2.6	Obtaining force constant from the power spectrum	13
2.6.1	Determination with prior knowledge of viscosity	13
2.6.2	Determination by using only the power spectra	14
2.7	Review of Paper: “Calibration of optical tweezers with posi- tional detection in the back focal plane”	14
2.7.1	Introduction	14
2.7.2	Equation of Motion	14
2.7.3	Fourier transform of response functions	18
2.7.4	Power spectrum of the response functions	20
2.7.5	Calibration	25
2.7.6	Using the experimentally determined data	26
3	Dynamic Light Scattering	27
3.1	Theory, Dynamic light scattering	27
3.1.1	Diffusion	27
3.1.2	Dynamic Light Scattering	28
3.2	Experimental setup	31
3.2.1	Particle size and settling time	32
3.2.2	Effect of salt on viscosity and refractive index	33
3.3	Results	34
4	Equipment review	37
4.1	Introduction	37
4.2	Laser	37
4.2.1	Laser Controller	39
4.2.2	Powering up the laser	39

4.2.3	Laser beam stability	40
4.2.4	Laser power and noise	41
4.3	Optics	43
4.3.1	Optical Losses	45
4.4	Microscope	45
4.5	Quadrant Photo diode (QPD)	47
4.6	Data Acquisition Card	49
4.7	Optical Alignment	49
4.7.1	Course alignment	49
4.7.2	Fine alignment	51
4.7.3	Walking the beam	52
4.8	Piezo Electric Stage	52
5	Experimental Procedure	53
5.1	Data capture — “optimising” sampling	53
5.2	Data Processing	54
5.3	Cell preparation	56
5.4	Verifying program code	58
5.4.1	Frequency accuracy	58
5.4.2	Amplitude accuracy	58
5.5	Power spectrum analysis	63
5.6	Power spectrum for a trapped $1.54\mu\text{m}$ particle	63
5.7	Using sinusoidal stage movement for calibration	65
5.8	Measuring trap parameters by measuring particle position in a flow	67
5.8.1	Theory for calculating experimental parameters	67
5.8.2	Experimental method	68
5.9	Experimental data for sinusoidal stage movement	73
6	Characterisation of noise	75
6.1	DAQ Card	75
6.1.1	Using a different DAQ	75
6.1.2	DAQ linearity	76
6.1.3	Aliasing due to sampling frequency and filters	76
6.1.4	Frequency matching with piezo stage	77
6.2	Quadrant photo diode	78
6.2.1	QPD Linearity	78
6.2.2	Power supply	78
6.2.3	Signal level	79
6.3	Laser problems	82
6.3.1	Signal variations	82

6.3.2	Low frequency noise	82
6.4	Problems associated with the sample cell	83
6.4.1	Changes in salt concentration	83
6.5	External sources of noise	84
6.5.1	Vibration	84
7	Flow cell Design	86
7.1	Initial flow cell design	86
7.2	Flow cell revisions	86
7.2.1	Characterisation of Flow cell	87
7.3	Flow cell use	89
8	Discussion of project results	90
A	Computer Code	93
A.1	Automatic laser blocker	93
A.2	Code for taking compressed data and making a power spectrum	96
A.3	Code used to verify Fast Fourier transform coefficients	106
A.4	Labview code	108
A.4.1	Code for viewing live data	108
A.4.2	Code for acquiring experimental data set	108
A.4.3	Code for initialising DAQ	109
A.5	Matlab code	112
A.5.1	Code for finding physical properties of system	112
A.5.2	Code for importing Labview HWS data	114
B	Micelle preparation	114
B.1	Preparation of Entangled Worm like Micells	114
B.2	Calculations:	115
B.2.1	Molecular Masses:	116
B.2.2	Preparation of the Worm like micells:	116
	Bibliography	119

1 Introduction

The initial goal of this masters project was to use the optical tweezer as a tool to probe the rheological properties of sponge like micells. As the optical tweezer setup was new the plan was to verify that the calibration worked, and then start using the tweezers. Unfortunately the calibration procedure was not as straight forward as expected and a substantial amount of time was spent learning the basics of the theory behind the trapping process. During the attempts to calibrate the system each component was checked, this resulted in numerous different experiments, swapping the laser and adding extra sensors like a photo diode. The end result is that we still haven't been able to calibrate it properly although given more time I am sure that we could get the "flow method" to work.

2 Theory of Optical tweezers

2.1 Initial development of the Optical tweezer

During the early part of the 17th century J. Kepler suggested that the reason a comet's tail points away from the sun was due to the radiation pressure of light. It wasn't until 1873 when J. C. Maxwell made an order of magnitude prediction of the forces involved in radiation pressure. Attempts at showing the radiation pressure prior to these calculations had been off by 4–5 orders of magnitude. This is rather unsurprising considering that the forces being measured are of a couple of pico-newtons, preventing real world applications until the discovery of the laser in the 1960's. The introduction of the laser provided a coherent and spectrally pure ray of light which could be focused to a spot approximately the size of a single wavelength. Focusing the beam allowed for extremely high intensities, on the order of 10,000 times the intensity at the surface of the sun. By using order of magnitude calculations A. Ashkin[2] realised that the radiation pressure from a laser would be sufficient to accelerate a $0.5\mu\text{m}$ dielectric particle at one million g^1 , where g represents the acceleration due to gravity. This acceleration is possible even though only 10% of the light is reflected by the particle.

Initial experiments showed that the radiation pressure in the beam was sufficient to overcome the force of gravity and cause a particle to move with a

¹This was by: $F_{rad} = 2qP/c$, where q is the reflectivity (0.1), P is the laser power (1W) giving $F_{rad} = 6.6 \cdot 10^{-10} \text{N}$.

Using the mass of a neutrally buoyant $1\mu\text{m}$ particle ($5.7 \cdot 10^{-19} \text{ kg}$) the acceleration is found as $1.15 \cdot 10^6 \text{ m/s}$

constant velocity. Once in motion it would travel until it reached the far side of the cell where it remained until the beam was interrupted. The particle would start wandering due to Brownian motion, but the moment the laser was switched on the particle would snap back to the centre of the beam, as described in Figure 1.

A similar technique is possible using two opposing lasers. This setup is called an “Optical Bottle” as the lasers are focused to a spot slightly off from the centre of the cell as can be seen in Figure 2

Any particles in the centre of the cell would be trapped by the radiation pressure from the two lasers, although any alignment errors quickly causes problems as the particle would not be in a stable trap and forced off to one side.

2.2 Single Beam Gradient Trap

Both the previous methods for trapping particles relied on either multiple lasers or gravity to keep the particle trapped. For this reason the single beam gradient trap was developed. This simple design uses a strongly focused laser beam to create a particle trap which is stable in both axial and radial directions.

Trapping theory falls into two distinct categories which depends on the particle diameter (d) and trapping wavelength (λ), Mie theory ($d \gg \lambda$) and Raleigh theory ($d \ll \lambda$). The particles used in this project are of the same order as the wavelength ($d \approx \lambda$), this implies that neither theory accurately explains the forces on particles, Mie is usually associated with particles around 10λ while for Raleigh theory particles are of the order 0.1λ . Theoretical calculations for $d \approx \lambda$ fall in neither category and thus require a more complete electromagnetic theory[4, 5].

2.2.1 Mie theory of trapping

When the particle to be trapped satisfies $d \gg \lambda$ we are in the Mie regime and the forces in the optical trap can be calculated using ray optics. The trapping forces will fall into two parts; the axial and lateral trapping forces. Light entering the particle will undergo refraction, this causes a change in the momentum which in turn corresponds to a force in the opposite direction. This is easily understood by looking at Newtons second and third law.

- 2 : $\mathbf{F} = \frac{d\mathbf{p}}{dt}$
- 3 : “Every action has an equal and opposite reaction”

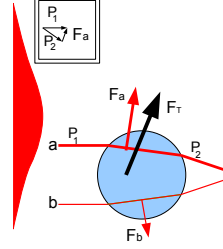


Figure 1: Radiation force on a dielectric sphere, displaced from the central axis. Two parallel beams a and b with differing intensities enter the particle causing a total force of $F_T = F_a + F_b$. Each force is the sum of the radiation pressure due to refractions when light enters or leaves the particle.

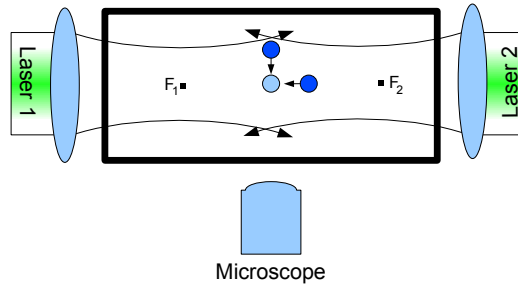


Figure 2: "Optical bottle" with two counter-propagating laser beams, showing the restoring force for the two dark blue particles towards the equilibrium position, light blue particle. F_1 and F_2 are the focal points for the two beams

where p is the momentum and t is the time. As the momentum of light is given by $p = \frac{U}{nc}$, where U is the energy, n the index of refraction and c is the speed of light, we can quickly see that moving into a medium with a different refraction index will change the momentum, and hence apply a force. For a particle in a suspension, the direction of the force, depends on the surrounding medium; if it has a lower index of refraction the force on the bead will be in the direction of the gradient, otherwise it will be away from the gradient. Hence a particle like silica ($n=1.37$) with a higher index of refraction than water ($n=1.33$) will move towards the centre of the Gaussian beam, while an air bubble ($n=1.00$) will move out of the beam[2].

Lateral Trapping Lateral trapping is the trapping that occurs perpendicular to the direction of propagation of the laser beam. Figure 3 shows the forces involved in axial trapping. The forces are due to the two rays F_a and F_b , where F_b represents the more intense ray. The rays are in the directions of propagation and hence have a component pushing the bead away from the laser, this is represented by $F_{Scatter}$. $F_{Gradient}$ represents the force due to the larger gradient in the F_b direction, resulting in a movement towards the centre of the beam (negative y direction in Figure 3).

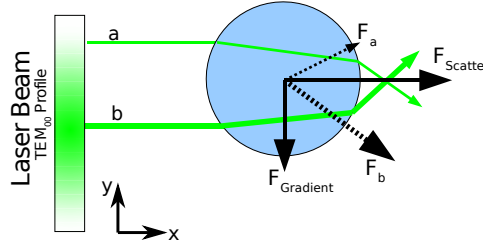


Figure 3: Lateral trapping of a particle, illustrated by two incident rays of different intensities striking the particle symmetrically about its centre.

Axial Trapping Axial trapping is the trapping that occurs in the direction of propagation of a focused beam. Figure 4 shows the forces involved in axial trapping, these forces arise due to:

- Scattering due to gradient of laser comes from two components F_{g1} and F_{g2} , resulting in a force $F_{Gradient}=F_{g1} + F_{g2}$ in the negative x direction.
- Scattering of reflected light. As the light reflects from the particle, it

will result in an equal and opposite force due to the change in momentum. $F_{Scatter} = F_{R1} + F_{R2}$

For transparent dielectric particles the end result is that the reflected light applies a smaller force than the scattered light, $F_{Scatter} < F_{Gradient}$, thus pulling the particle in towards the centre of the trap (negative x). Due to the reflected light, it is clear that force required to pull the particle away from the trap in the positive x direction will be less than pulling it away from the trap in the negative x direction, for the same reason the particle will be offset in the direction of propagation (positive x) direction.

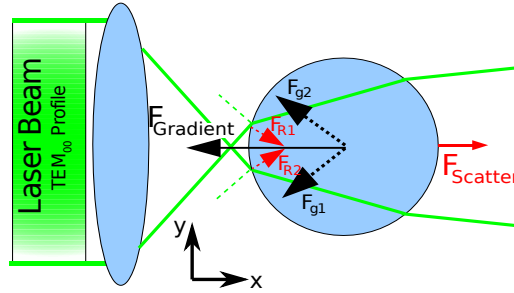


Figure 4: Axial trapping showing two highly focused rays striking a particle symmetrically about its centre. The particle is shown below the focus point.

2.2.2 Rayleigh Scattering

Rayleigh scattering involves particles with diameters smaller than the wavelength ($d \ll \lambda$) and we can treat the particle as an induced dipole. Once this assumption has been made we can treat the particle with standard electromagnetic theory. The following section uses the derivations found at [28, 13].

Gradient force The gradient force is due to the Lorenz force acting on the dipole. For a point charge (q), the force (F) is given by:

$$\mathbf{F} = q(\mathbf{E} + \frac{d\mathbf{x}}{dt} \times \mathbf{B}), \quad (1)$$

where E is the electric field, q the electric charge, dx/dt the velocity of the particle and B the magnetic field. Inserting for a dipole of two opposite charges ($q = q_1 = q_2$), separated by distance $(x_1 - x_2)$ gives

$$\mathbf{F} = q(\mathbf{E}_1 - \mathbf{E}_2 + \frac{d(\mathbf{x}_1 - \mathbf{x}_2)}{dt} \times \mathbf{B}), \quad (2)$$

rewriting \mathbf{E}_2 in terms of the averaged field, \mathbf{E} , and \mathbf{E}_1 gives,

$$\mathbf{F} = q(\mathbf{E}_1 - \mathbf{E}_1 + ((x_1 - x_2) \cdot \nabla)\mathbf{E} + \frac{d(\mathbf{x}_1 - \mathbf{x}_2)}{dt} \times \mathbf{B}), \quad (3)$$

$$\mathbf{F} = q(((x_1 - x_2) \cdot \nabla)\mathbf{E} + \frac{d(\mathbf{x}_1 - \mathbf{x}_2)}{dt} \times \mathbf{B}), \quad (4)$$

and multiplying through with the charge,

$$\mathbf{F} = (\mathbf{p} \cdot \nabla)\mathbf{E} + \frac{d\mathbf{p}}{dt} \times \mathbf{B}. \quad (5)$$

Using $\mathbf{p} = \alpha\mathbf{E}$, where α is the polarizability gives

$$\mathbf{F} = \alpha(\mathbf{E} \cdot \nabla)\mathbf{E} + \frac{d\mathbf{E}}{dt} \times \mathbf{B}, \quad (6)$$

and then substituting the following vector analysis identity,

$$\nabla \mathbf{E}^2 = 2(\mathbf{E} \cdot \nabla)\mathbf{E} + 2\mathbf{E} \times (\nabla \times \mathbf{E}), \quad (7)$$

gives

$$\mathbf{F} = \alpha\left(\frac{1}{2}\nabla \mathbf{E}^2 - \mathbf{E} \times \left(-\frac{d\mathbf{B}}{dt}\right) + \frac{d\mathbf{E}}{dt} \times \mathbf{B}\right). \quad (8)$$

Where we used Maxwell's equation $\nabla \times \mathbf{E} = \frac{-d\mathbf{B}}{dt}$. Thus the last two terms can be combined,

$$\mathbf{F} = \alpha\left(\frac{1}{2}\nabla \mathbf{E}^2 + \frac{d}{dt}(\mathbf{E} \times \mathbf{B})\right). \quad (9)$$

The last term is dependent on the variation in intensity of the laser, which in our case remains constant. Therefore this term can be dropped,

$$\mathbf{F} = \frac{1}{2}\alpha\nabla \mathbf{E}^2. \quad (10)$$

The resulting force on the particle is in the direction of the gradient.

Scattering force Taking a standard spherical particle, the Rayleigh cross-sectional area is:

$$\sigma_s = \frac{8\pi}{3} \left(\frac{2\pi n_{med}}{\lambda_0}\right)^4 r^6 \left(\frac{m^2 - 1}{m^2 + 2}\right) \cdot [1] \quad (11)$$

Where r is the radius, λ_0 is the wavelength in vacuum and n_{med} is the refractive index of the surrounding medium, $m = n_{particle}/n_{med}$ is the ratio of

refractive index of the particle to that of the surroundings. We know that the momentum is given by $p = U/c$, and that pressure is given by $P = \frac{1}{A} \frac{dp}{dt}$, putting these together gives:

$$P = \frac{1}{A} \frac{dp}{dt} = \frac{1}{A} \frac{d}{dt} \left(\frac{U}{c} \right) = \frac{1}{c} \frac{dU/dt}{A}, \quad (12)$$

where $\frac{dU/dt}{A}$ is the rate at which energy is arriving at a surface per unit time, which is also the time averaged Poynting vector ($\langle S \rangle$). Thus the radiation pressure, which is a force per unit area is given by:

$$P_{rad} = \frac{F}{\sigma} = \frac{\langle S \rangle}{c}. \quad (13)$$

Rewriting this to take into account different mediums, and hence different refractive indexes gives,

$$F_{scat} = \frac{m\sigma \langle S \rangle}{c}, \quad (14)$$

where m is the refractive index of the particle divided by the refractive index of the medium.

Thus for a focused beam we have two forces F_{scat} and F_{grad} , one pushing the particle in the direction of propagation and the other pulling the particle towards the centre of the trap.

2.3 Brownian Motion

One of the central theories applicable to optical tweezers is that of Brownian motion. Initially Brownian motion was discovered by observing the movement of small particles in water. With what seems like no external force the particles would wander around as if they were colliding with other objects. The term Brownian motion is a credit to the botanist Robert Brown who is traditionally acknowledged with the discovery of this process.

As can clearly be seen from Figure 5 of 2d Brownian motion it would be impossible to make deterministic predictions of particle positions and we therefore have to take a probabilistic route.

To create a mathematical model of Brownian motion the Wiener process (W_t) was developed, this model is a continuous time stochastic process with the following properties:

- $W(t = 0) = 0$, Initially the particle is at zero and defined for all times greater than zero.
- $W(t)$ is a continuous function of time.

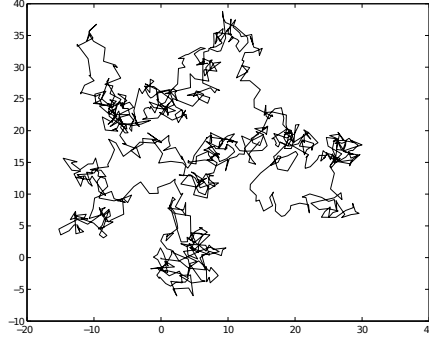


Figure 5: Example simulation of 2D Brownian motion, particle started out at $\{0,0\}$ and took 1000 steps. Each step direction is determined by a random number, and another random number determines the length of the step.

- $W(t)$ has independent increments with a normal distribution, and hence an expectance of value μ and variance of σ^2

Using the Wiener process it is possible to prove[25] the Einstein relation relating the mobility (μ) to the diffusion constant, $D = \mu k_B T$. For the derivation of the power spectrum we will be using the random properties of Brownian motion by linking it to the auto correlation function by means of the equation of motion.

2.4 Langevin Equation

The equation of motion for a free particle in a solution can be written as:

$$m\ddot{\mathbf{x}} = m \frac{d^2 \mathbf{x}}{dt^2} = \mathbf{F}. \quad (15)$$

Where m is the mass, \mathbf{x} is the position, t is the time and \mathbf{F} is the force. By splitting up the force into a frictional term and a random force we can rewrite this as:

$$m\ddot{\mathbf{x}} = -m\gamma\dot{\mathbf{x}} + \mathbf{R}(t), \quad (16)$$

where γ is the drag coefficient, and \mathbf{R} is a random force. The stochastic differential equation above is also known as a Langevin equation.

Now if the particle is placed in a force field with potential V , the equation becomes

$$m\ddot{\mathbf{x}} = -\nabla \cdot V - m\gamma\dot{\mathbf{x}} + \mathbf{R}(t), \quad (17)$$

By approximation of the potential and random force in the expression above it is possible to solve these equations for trapped particles in solution.

2.4.1 Using the Langevin equation of motion to obtain an expression for the power spectrum for a trapped particle.

For a more physical description, see the alternative method in the next section. Particles trapped in an optical trap will undergo a Brownian motion but with a restoring force pulling them back to the centre of the trap. The theory presented here will show how we can use the power spectrum to determine physical properties of the system. The theory for this section has been derived in [14]

The theory is presented for the one dimensional case but should work well for both 2 and 3 dimensional cases. From the Einstein-Ornstein-Uhlenbeck theory of Brownian motion we get a Langevin equation describing the motion of a bead trapped in a potential, in this case we will assume a harmonic potential,

$$m\ddot{\mathbf{x}}(t) + \gamma_0\dot{\mathbf{x}}(t) + \kappa\mathbf{x}(t) = (2k_B T \gamma_0)^{1/2} \boldsymbol{\eta}(t), \quad (18)$$

where $\mathbf{x}(t)$ is the x displacement of the Brownian particle, γ_0 is the viscous drag, κ is the harmonic force constant, and $(2k_B T \gamma_0)^{1/2} \boldsymbol{\eta}$ is the random force with a Gaussian distribution. Since $\boldsymbol{\eta}$ has a Gaussian distribution we know the following:

$$\langle \boldsymbol{\eta}(t) \rangle = 0 \quad \text{and} \quad \langle \boldsymbol{\eta}(t) \boldsymbol{\eta}(t') \rangle = \delta(t - t'). [14] \quad (19)$$

Since the trapping will mostly occur at low Reynolds numbers we can drop the inertial term as long as the experiments last longer than the characteristic timescales².

Eq. (18) is thus reduced to:

$$\gamma_0\dot{\mathbf{x}}(t) + \kappa\mathbf{x}(t) = (2k_B T \gamma_0)^{1/2} \boldsymbol{\eta}(t), \quad (20)$$

Additionally we can simplify this equation by introducing the corner frequency $f_c = \kappa/(2\pi\gamma_0)$ and the Einstein relation $D = k_B T / \gamma_0$, where D is the diffusion coefficient, k_B the Boltzmann constant and T the temperature. Hence we get:

$$\dot{\mathbf{x}}(t) + 2\pi f_c \mathbf{x}(t) = (2D)^{1/2} \boldsymbol{\eta}(t). \quad (21)$$

²This can be seen by looking at Eq. (41) in section 2.5, where the auto correlation function has the expression $e^{-|\tau|\kappa/\gamma}$ where we can define a characteristic time, $\tau_0 = \kappa/\gamma$ which is of order 1ms for a typical trap.

We define the Fourier transform as:

$$\hat{x}(f) = \int_{-\infty}^{\infty} dt x(t) e^{-i2\pi ft} \quad (22)$$

$$x(t) = \int_{-\infty}^{\infty} df \hat{x}(f) e^{i2\pi ft} \quad (23)$$

Taking the Fourier transform of Eq. (21) with $f_k = k/T_{msr}$, where T_{msr} is the measurement time, results in,

$$\tilde{\mathbf{x}}_k = \frac{(2D)^{1/2} \tilde{\boldsymbol{\eta}}_k}{2\pi(f_c - if_k)}. \quad (24)$$

From Eq. (19) we can define a similar set of equations for the Fourier transformed $\tilde{\boldsymbol{\eta}}$

$$\langle \tilde{\boldsymbol{\eta}}_k \rangle = 0 \quad \text{and} \quad \langle \tilde{\boldsymbol{\eta}}_k^* \tilde{\boldsymbol{\eta}}_l \rangle = T_{msr} \delta_{k,l}. \quad (25)$$

The power spectra S_k is defined as

$$S_k \equiv |\tilde{\mathbf{x}}_k|^2 / T_{msr}, \quad (26)$$

inserting the expression for $\tilde{\mathbf{x}}(f)$ gives

$$S_k = \frac{D |\tilde{\boldsymbol{\eta}}(f)|^2}{T_{msr} 2\pi^2 (f_c^2 + f_k^2)}. \quad (27)$$

Taking the average and going to the continuous f leads to

$$S_k = \frac{DT_{msr} \delta_{k,l}}{T_{msr} 2\pi^2 (f_c^2 + f_k^2)}, \quad (28)$$

$$= \frac{D \delta_{k,l}}{2\pi^2 (f_c^2 + f_k^2)}, \quad (29)$$

hence

$$\langle S(f) \rangle = \frac{D}{2\pi^2 (f_c^2 + f^2)}. \quad (30)$$

2.5 Obtaining the power spectra via the Wiener-Khintchine and equipartition

The theory presented in this section is based on the derivation at [6]. For a particle trapped in a harmonic potential we can write the equation of motion as:

$$m\ddot{\mathbf{x}}(t) + \gamma\dot{\mathbf{x}}(t) + \kappa\mathbf{x}(t) = \boldsymbol{\eta}(t), \quad (31)$$

where m is the mass, \mathbf{x} the displacement from equilibrium, γ the drag coefficient, κ the force constant and $\boldsymbol{\eta}$ the random force due to Brownian motion. For the large time scales, such as those used in this project the damping of the inertial term isn't significant and thus we drop this term. This gives an equation that is easier to solve:

$$\dot{\mathbf{x}}(t) + \frac{\kappa \mathbf{x}(t)}{\gamma} = \frac{1}{\gamma} \boldsymbol{\eta}(t). \quad (32)$$

As $\boldsymbol{\eta}$ will be a random driving force, we can solve Eq. (32) using an integrating factor,

$$Y(t) = \frac{1}{I_F} \int Q(t) I_F dt, \quad \text{where} \quad I_F = e^{\int P(t) dt}. \quad (33)$$

Choosing $P(t)$ as κ/γ gives an integrating factor $I_F = e^{t\kappa/\gamma}$ and $Q(t) = \frac{1}{\gamma} \boldsymbol{\eta}(t)$, then substituting this into Eq. (33) gives

$$\mathbf{x}(t) = e^{-t\kappa/\gamma} \int_{-\infty}^t \frac{1}{\gamma} \boldsymbol{\eta}(t') e^{t'\kappa/\gamma} dt' \quad (34)$$

The correlation function from t to $t + \tau$ can then be written:

$$\mathbf{x}(0)\mathbf{x}(\tau) = \frac{e^{-\tau\kappa/\gamma}}{\gamma^2} \int_{-\infty}^0 \boldsymbol{\eta}(t'') e^{t''\kappa/\gamma} dt'' \int_{-\infty}^{\tau} \boldsymbol{\eta}(t') e^{t'\kappa/\gamma} dt' \quad (35)$$

The auto correlation function can then be found by taking the ensemble average

$$\langle \mathbf{x}(0)\mathbf{x}(\tau) \rangle = \frac{e^{-\tau\kappa/\gamma}}{\gamma^2} \int_{-\infty}^0 dt'' \int_{-\infty}^{\tau} \langle \boldsymbol{\eta}(t') \boldsymbol{\eta}(t'') \rangle e^{(t'+t'')\kappa/\gamma} dt' \quad (36)$$

as $\boldsymbol{\eta}$'s are the only stochastic terms. We assume that the random term $\boldsymbol{\eta}$ has a Gaussian distribution we can therefore write $\langle \boldsymbol{\eta}(t') \boldsymbol{\eta}(t'') \rangle = r \delta(t' - t'')$, where r is an amplitude and δ is a Dirac delta function.

$$\langle \mathbf{x}(0)\mathbf{x}(\tau) \rangle = \frac{r e^{-\tau\kappa/\gamma}}{\gamma^2} \int_{-\infty}^0 dt'' \int_{-\infty}^{\tau} \delta(t' - t'') e^{(t'+t'')\kappa/\gamma} dt' \quad (37)$$

$$= \frac{r e^{-\tau\kappa/\gamma}}{\gamma^2} \int_{-\infty}^0 e^{(2t'')\kappa/\gamma} dt'' \quad (38)$$

$$= \frac{r e^{-\tau\kappa/\gamma}}{\gamma^2} \frac{\gamma}{2\kappa} \quad (39)$$

$$= \frac{r}{2\kappa\gamma} e^{-\tau\kappa/\gamma} \quad (40)$$

from equipartition we know $\langle x^2 \rangle = k_B T / \kappa$, setting $\tau = 0$ and substituting this into Eq. (40) we see that $r = 2k_B T \gamma$, giving the auto correlation function as:

$$\phi(t) = \langle x(t)x(t+\tau) \rangle = \frac{k_B T}{\kappa} e^{-|\tau|\kappa/\gamma} \quad (41)$$

The Wiener-Khintchine[25] theorem states that the Fourier transform of an auto correlation function is the power spectral density of a stationary random process, and defined as follows:

$$S(f) = \int_{-\infty}^{\infty} \phi(t) e^{-2i\pi f t} dt \quad (42)$$

$$\phi(t) = \int_{-\infty}^{\infty} S(f) e^{2i\pi f t} df \quad (43)$$

Substituting our auto correlation function into the Wiener-Khintchine theorem we get:

$$S(f) = \frac{k_B T}{\kappa} \frac{1}{2\pi} \int_{-\infty}^{\infty} d\tau e^{-|\tau|\kappa/\gamma} e^{-i\omega\tau} \quad (44)$$

$$= \frac{k_B T}{\kappa} \frac{2\kappa/\gamma}{(\kappa/\gamma)^2 + \omega^2} \quad (45)$$

$$= \frac{2D}{(\kappa/\gamma)^2 + \omega^2} \quad (46)$$

$$= \frac{D}{2\pi^2(f_c^2 + f^2)}, \quad (47)$$

where we have used definitions for $f_c = \kappa/(2\pi\gamma)$.

2.6 Obtaining force constant from the power spectrum

When a particle is trapped whether, it has an externally driven force on it or not, as long as fluid particles collide with the trapped particle we will see fluctuations in the light intensity. Recording these with a QPD we can obtain a power spectrum with a Lorentzian form. Fitting this curve with Eq. (30) will provide us with two variables, the corner frequency (f_c) and diffusion coefficient (D_V). Using these values it is possible to determine the trap stiffness and drag constant for the system.

2.6.1 Determination with prior knowledge of viscosity

Once we have made a fit of Eq. (30) we should have a value for the corner frequency, f_c . In our original definition we set the corner frequency to $f_c =$

$\kappa/(2\pi\gamma)$, thus by rearranging this expression we obtain an expression for the force constant κ .

$$\kappa = 2f_c\pi\gamma. \quad (48)$$

2.6.2 Determination by using only the power spectra

Alternatively we can find the force constant using only the power spectrum. Rearranging Eq. (30) and putting in the definition for diffusion constant (Eq. (146)) we can find an expression for γ ,

$$\gamma = \frac{k_B T}{2S(f)\pi^2(f_c^2 + f^2)}. \quad (49)$$

Using Eq. (49) we can solve for γ by setting the frequency to zero. The resulting $S(0)$ is the height of the plateau.

$$\gamma = \frac{k_B T}{2S_0\pi^2 f_c^2}. \quad (50)$$

Adding this expression to Eq. (48) we arrive at,

$$\kappa = \frac{k_B T}{\pi S_0 f_c}. \quad (51)$$

This method wasn't used much as it would only give the value of κ in terms of volts as we had no method for converting the voltage to distance.

2.7 Review of Paper: “Calibration of optical tweezers with positional detection in the back focal plane”

2.7.1 Introduction

The following section is based on the paper titled, “Calibration of optical tweezers with positional detection in the back focal plane”[27], which is a method to calibrate the optical tweezer and at the same time get a measurement of the viscosity. This gives an indication of the accuracy of the calibration. I will only be treating sections I-IV, as the hydrodynamical corrections are a refinement of these principles.

2.7.2 Equation of Motion

Ignoring hydrodynamical effects we can write a Langevin equation of motion for a spherical bead in a trap as:

$$\gamma [\dot{x}(t) - v_{drive}(t)] + \kappa x(t) = F_T(t). \quad (52)$$

This equation contains a contribution of three parts, $\gamma [\dot{x}(t) - v_{drive}(t)]$ represents the drag force, $\kappa x(t)$ represents the trapping force, and finally $F_T(t)$ represents the random thermal force.

Having assumed that the equation of motion (EOM) Eq. (52) is linearly independent for both the thermal ($F_T(t)$) and driving force (γv_{drive}), we can write the solution of the equation of motion,

$$x(t) = x_T(t) + x_{response}(t). \quad (53)$$

The thermal response $x_T(t)$ and the driving force response are found by solving Eq. (52) when v_{drive} and F_T are respectively set to zero.

Thermal fluctuations Setting the driving force equal to zero

$$\gamma \dot{x}_T(t) + \kappa x_T(t) = F_T(t) \quad (54)$$

$$\dot{x}_T(t) + \frac{\kappa}{\gamma} x_T(t) = \frac{F_T(t)}{\gamma} = \sqrt{2D} \xi(t), \quad (55)$$

where we assumed the statistical properties of white noise to apply to the thermal force,

$$F_T(t) = \sqrt{2\gamma k_B T} \xi = \gamma \sqrt{2D} \xi. \quad (56)$$

Using κ/γ we find the integrating factor $e^{\frac{\kappa t}{\gamma}}$ and integrate the right hand side

$$e^{\frac{\kappa}{\gamma} t} \left[\dot{x}_T(t) + \frac{\kappa}{\gamma} x_T(t) \right] = \frac{d}{dt} \left[x_T(t) e^{\frac{\kappa t}{\gamma}} \right] = \gamma \sqrt{2D} \xi(t) e^{\frac{\kappa t}{\gamma}} \quad (57)$$

$$x_T(t) = \frac{1}{e^{\frac{\kappa}{\gamma} t}} \sqrt{2D} \int_{-\infty}^t e^{\frac{\kappa t'}{\gamma}} \xi(t') dt'. \quad (58)$$

We define the corner frequency f_c as

$$f_c = \frac{\kappa}{2\pi\gamma} \quad (59)$$

$$\begin{aligned} x_T(t) &= e^{-2\pi f_c t} \sqrt{2D} \int_{-\infty}^t e^{2\pi f_c t'} \xi(t') dt' \\ x_T(t) &= \sqrt{2D} \int_{-\infty}^t e^{-2\pi f_c (t-t')} \xi(t') dt' \end{aligned} \quad (60)$$

Driving motion We start by writing an expression for the stage position with respect to time. This is represented by Eq. (61), differentiating this with respect to time we can also obtain the velocity of the stage, which enters in the equation of motion.

$$x_{drive}(t) = A \sin(2\pi f_{drive}t) \quad (61)$$

$$\dot{x}_{drive}(t) = 2\pi f_{drive}A \cos(2\pi f_{drive}t) \quad (62)$$

In a similar manner to section 2.7.2 above, the thermal force can be set to zero and a solution can be found for the equation of motion. Substituting v_{drive} for Eq. (62) and $\frac{\kappa}{2\pi\gamma}$ with f_c , $2\pi f_{drive}$ with ω_d and finally $2\pi f_c$ with ω_c we obtain

$$\begin{aligned} \gamma [\dot{x}(t) - v_{drive}(t)] + \kappa x(t) &= F_T(t) = 0 \\ \dot{x}(t) - v_{drive}(t) + \frac{\kappa}{\gamma}x(t) &= 0 \\ \dot{x}(t) - 2\pi f_{drive}A \cos(2\pi f_{drive}t) + \omega_c x(t) &= 0 \\ \dot{x}(t) - A\omega_d \cos(\omega_d t) + \omega_c x(t) &= 0. \end{aligned} \quad (63)$$

Taking the Laplace transform of Eq. (63)

$$sX(s) - \frac{A\omega_d s}{s^2 + \omega_d^2} + \omega_c X(s) = 0$$

where $X(s) = \mathcal{L}(x(t))$, and solving for $X(s)$ we obtain:

$$X(s) = \frac{A\omega_d s}{(s^2 + \omega_d^2)(s + \omega_c)}. \quad (64)$$

In order to perform an inverse Laplace transform we choose to split the right hand side of Eq. (64) into two partial fractions. We make the ansatz

$$\frac{P}{s + \omega_c} + \frac{Qs + R}{s^2 + \omega_d^2} = \frac{A\omega_d s}{(s + \omega_c)(s^2 + \omega_d^2)},$$

where P, Q and R are to be found. Rearranging terms,

$$Ps^2 + P\omega_d^2 + Qs^2 + Rs + Qs\omega_c + R\omega_c = A\omega_d s,$$

and grouping coefficients for each power of s :

$$s^2 : P + Q = 0 \quad \Rightarrow \quad Q = -P \quad (65)$$

$$s^1 : Q\omega_c + R = A\omega_d \quad (66)$$

$$s^0 : P\omega_d^2 + R\omega_c = 0 \quad \Rightarrow \quad R = \frac{-P\omega_d^2}{\omega_c} = \frac{Q\omega_d^2}{\omega_c}. \quad (67)$$

Substituting R in Eq. (66) with Eq. (67)

$$Q\omega_c + Q\frac{\omega_d^2}{\omega_c} = A\omega_d \quad (68)$$

$$Q\omega_c^2 + Q\omega_d^2 = A\omega_d\omega_c \quad (69)$$

$$Q(\omega_c^2 + \omega_d^2) = A\omega_d\omega_c. \quad (70)$$

Rewriting Eq. (70) with the help of Eq. (65) gives

$$Q = \frac{A\omega_d\omega_c}{\omega_c^2 + \omega_d^2}, \quad L = \frac{A\omega_d\omega_d^2}{\omega_c^2 + \omega_d^2}, \text{ and } P = \frac{-A\omega_d\omega_c}{\omega_c^2 + \omega_d^2}.$$

Defining $\alpha = \frac{f_c}{f_{drive}} = \frac{\omega_c}{\omega_d}$ we can simplify this to:

$$Q = \frac{A\alpha}{1 + \alpha^2}, \quad P = \frac{-A\alpha}{1 + \alpha^2}, \quad L = \frac{A\omega_d}{\alpha^2 + 1}. \quad (71)$$

This allows us to write the transformed Laplace equation (Eq. (64)) as

$$X(s) = \frac{-A\alpha}{(\alpha^2 + 1)(s + \alpha\omega_d)} + \frac{A(s\alpha + \omega_d)}{(\alpha^2 + 1)(s^2 + \omega_d^2)}. \quad (72)$$

Looking up these terms in tables allows us to quickly rewrite Eq. (72) as

$$x_{response}(t) = \frac{-\alpha A e^{-\alpha\omega_d t}}{\alpha^2 + 1} + \frac{\alpha A \cos(\omega_d t) + A \sin(\omega_d t)}{\alpha^2 + 1}. \quad (73)$$

We are interested in finding the behaviour of this function once the transient behaviour has passed, therefore taking t to be large the exponential term falls away and we can write this as

$$x_{response}(t) = \frac{\alpha A \cos(\omega_d t) + A \sin(\omega_d t)}{\alpha^2 + 1}. \quad (74)$$

In order to simplify Eq. (74) we will need a couple of identities:

$$\sin(\omega_d t + \phi) = \sin(\omega_d t) \cos(\phi) + \cos(\omega_d t) \sin(\phi) \quad (75)$$

$$\cot(\phi) = \frac{\cos(\phi)}{\sin(\phi)} \quad (76)$$

$$\tan(\theta - \frac{\pi}{2}) = -\cot(\theta) \quad (77)$$

Starting with Eq. (74) and choosing³ $\cos(\phi) = \frac{1}{\sqrt{1+\alpha^2}}$ and $\sin(\phi) = \frac{\alpha}{\sqrt{1+\alpha^2}}$ one obtains

$$x_{response}(t) = \frac{A \sin(\phi) \cos(\omega_d t) + A \cos(\phi) \sin(\omega_d t)}{\sqrt{\alpha^2 + 1}}. \quad (78)$$

Using Eq. (76) and the formula's for sin and cos selected above we can easily show that $\cot(\phi) = \frac{1}{\alpha}$ and along with Eq. (77) this gives

$$\begin{aligned}\cot(\phi) &= \frac{\cos(\phi)}{\sin(\phi)} = \frac{1}{\sqrt{1+\alpha^2}} \frac{\sqrt{1+\alpha^2}}{\alpha} = \frac{1}{\alpha} \\ \frac{1}{\alpha} &= \cot(\phi) = -\tan(\phi - \frac{\pi}{2}),\end{aligned}$$

taking arctan and substituting for α

$$\begin{aligned}\arctan\left(\frac{1}{\alpha}\right) &= -\phi + \frac{\pi}{2} \\ \arctan\left(\frac{f_{drive}}{f_c}\right) - \frac{\pi}{2} &= -\phi.\end{aligned}$$

We then define the phase difference $\frac{\phi}{\omega_d}$ to be a lag time t_{lag} and end up with the expression in Eq. (79)

$$t_{lag} = \frac{-\phi}{\omega_d} = \frac{\arctan\left(\frac{f_{drive}}{f_c}\right) - \frac{\pi}{2}}{2\pi f_{drive}}. \quad (79)$$

Taking Eq. (78) and making the substitution with Eq. (75) and inserting Eq. (79) we arrive at

$$x_{response}(t) = \frac{A \sin(\omega_d t + \phi)}{\sqrt{1+\alpha^2}} = \frac{A \sin(\omega_d t - \omega_d t_{lag})}{\sqrt{1+\alpha^2}}. \quad (80)$$

And finally using Eq. (61) and substituting for α we arrive at the desired expression

$$x_{response}(t) = \frac{x_{drive}(t - t_{lag})}{\sqrt{1 + (f_c/f_{drive})^2}}. \quad (81)$$

2.7.3 Fourier transform of response functions

We could fit Eq. (81) to an experimentally attained waveform, but this has shown not to be very accurate and we therefore wish to take the power spectrum of Eq. (60) and Eq. (81). This can in turn be compared with the experimental power spectrum.

³We are able to make this assignment as we still meet the requirement that $\cos^2 \phi + \sin^2 \phi = 1$

Thermal response Taking the Fourier transform (defined as Eq. (23)) of x_T (Eq. (60))

$$\mathcal{F}(x_T(t)) = \mathcal{F} \left(\sqrt{2D} \int_{-\infty}^t dt' e^{-2\pi f_c(t-t')} \xi(t') \right) \quad (82)$$

$$\hat{x}_T(f) = \sqrt{2D} \int_{-\infty}^{\infty} dt \int_{-\infty}^t dt' e^{-2\pi f_c(t-t')} \xi(t') e^{-i2\pi ft}. \quad (83)$$

As the first integration over t' is from minus infinity to t , while the second integration goes from minus infinity to plus infinity, covering the shaded area in Figure 6. This means that we can change the integration order and simplify the integral in Eq. (83) resulting in

$$\hat{x}_T(f) = \sqrt{2D} \int_{-\infty}^{\infty} dt' \int_{t'}^{\infty} dt e^{-2\pi f_c(t-t')} e^{-i2\pi ft} \xi(t'), \quad (84)$$

reordering the exponential and integrating

$$\hat{x}_T(f) = \sqrt{2D} \int_{-\infty}^{\infty} dt' \int_{t'}^{\infty} dt e^{-2\pi(f_c+if)t} e^{2\pi f_c t'} \xi(t') \quad (85)$$

$$= \sqrt{2D} \int_{-\infty}^{\infty} dt' \left[\frac{-1}{2\pi(f_c+if)} e^{-2\pi(f_c+if)t} \right]_{t'}^{\infty} e^{2\pi f_c t'} \xi(t') \quad (86)$$

$$= \sqrt{2D} \int_{-\infty}^{\infty} dt' \frac{-1}{2\pi(f_c+if)} \left[e^{-2\pi(f_c+if)t'} - e^{-\infty} \right] e^{2\pi f_c t'} \xi(t') \quad (87)$$

$$= \sqrt{2D} \int_{-\infty}^{\infty} dt' \frac{1}{2\pi(f_c+if)} e^{2\pi i f t'} \xi(t') \quad (88)$$

$$= \frac{\sqrt{2D}}{2\pi(f_c+if)} \hat{\xi}(f). \quad (89)$$

Drive response Taking the Fourier transform of $x_{response}$ Eq. (80) one gets

$$\mathcal{F}(x_{response}(t)) = \mathcal{F} \left(\frac{x_{drive}(t - t_{lag})}{\sqrt{1 + (f_c/f_{drive})^2}} \right). \quad (90)$$

Substituting Eq. (61) for x_{drive} and writing the sine as a sum of two exponential gives

$$\hat{x}_{res}(f) = \frac{A}{\sqrt{1 + (f_c/f_{drive})^2}} \int_{-\infty}^{\infty} dt e^{-i2\pi ft} \sin(2\pi f_d t - 2\pi f_d t_{lag}) \quad (91)$$

$$= \frac{A}{2i\sqrt{1 + (f_c/f_{drive})^2}} \int_{-\infty}^{\infty} dt e^{-i2\pi ft} \left(e^{i2\pi f_d(t-t_{lag})} - e^{-i2\pi f_d(t-t_{lag})} \right). \quad (92)$$

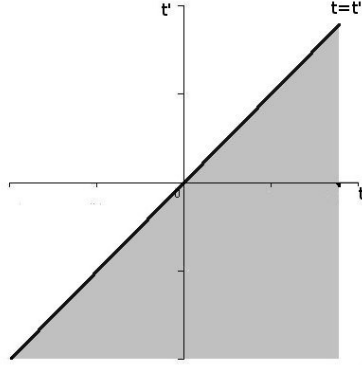


Figure 6: Plot showing the integration domain

Rearranging the exponentials and writing only the integrals gives

$$Int = \int_{-\infty}^{\infty} dt \left(e^{-i2\pi(f-f_{drive})t} e^{-i2\pi f_{drive}t_{lag}} - e^{-i2\pi(f+f_{drive})t} e^{i2\pi f_{drive}t_{lag}} \right). \quad (93)$$

We can immediately see that we have a Fourier representation of a delta function[26, p106] around $(f \pm f_{drive})$, allowing us to rewrite the integral as

$$Int = [\delta(f - f_{drive}) e^{-i2\pi f_{drive}t_{lag}} - \delta(f + f_{drive}) e^{i2\pi f_{drive}t_{lag}}]. \quad (94)$$

As each delta function selectively selects negative and positive frequencies at $f = \pm f_{drive}$ this can be further simplified by pulling out the exponential term

$$\hat{x}_{resp}(f) = \frac{A e^{i2\pi f t_{lag}}}{2i\sqrt{1 + (f_c/f_{drive})^2}} [\delta(f + f_{drive}) - \delta(f - f_{drive})]. \quad (95)$$

Giving the final form of Fourier transformed response functions as

$$\hat{x}(f) = \frac{\sqrt{2D}}{2\pi(f_c - if)} \hat{\xi}(f) + \frac{A e^{i2\pi f t_{lag}}}{2i\sqrt{1 + (f_c/f_{drive})^2}} [\delta(f + f_{drive}) - \delta(f - f_{drive})] \quad (96)$$

2.7.4 Power spectrum of the response functions

We now have the Fourier transform of the response, the definition[24, p498] that we will use for the power spectrum is:

$$P(f) = \frac{2 \langle |\hat{x}(f)|^2 \rangle}{t_{msr}} \quad (97)$$

We should note that taking the square of Eq. (96) will result in a cross terms containing $\langle \hat{\xi} \rangle$ this has the statistical properties of white noise and therefore will be equal to zero. This simplifies matters and allows us to use Eq. (97) on Eq. (89) and Eq. (95) independently.

Thermal Power Spectrum Starting with Eq. (89)

$$\begin{aligned} P_T(f) &= \frac{2 \langle |\frac{\sqrt{2D}}{2\pi(f_c+if)} \hat{\xi}(f)|^2 \rangle}{t_{msr}} \\ &= \frac{2 \langle \frac{2D}{4\pi^2(f_c-if)(f_c+if)} \hat{\xi}(f) \hat{\xi}^*(f) \rangle}{t_{msr}} \\ &= \frac{D \langle \hat{\xi}(f) \hat{\xi}^*(f) \rangle}{\pi^2(f_c^2 + f^2) t_{msr}} \end{aligned} \quad (98)$$

$$= \frac{D}{\pi^2(f_c^2 + f^2)}, \quad (99)$$

where we have used the fact that $\langle \hat{\xi}_k(f) \hat{\xi}_l(f) \rangle = t_{msr} \delta_{kl}$ and in going from Eq. (98) to Eq. (99) k and l are equal thus giving us the first part of total power spectrum.

Drive response Power spectrum Unfortunately squaring Eq. (95) gives a double delta function, which is not defined, therefore we have to do a discrete Fourier transform of Eq. (81) for a finite time

$$\hat{x}_{resp}(f_k) = \int_{-\frac{t_{msr}}{2}}^{\frac{t_{msr}}{2}} dt e^{-i2\pi f_k t} x(t) = \int_{-\frac{t_{msr}}{2}}^{\frac{t_{msr}}{2}} dt e^{-i2\pi f_k t} \frac{x_{drive}(t - t_{lag})}{\sqrt{1 + (f_c/f_{drive})^2}}.$$

Using Eq. (61),

$$\begin{aligned} \hat{x}_{resp}(f_k) &= \frac{A}{\sqrt{1+(f_c/f_{drive})^2}} \int_{-\frac{t_{msr}}{2}}^{\frac{t_{msr}}{2}} dt e^{-i2\pi f_k t} \sin(2\pi f_{drive}(t - t_{lag})) \\ &= \frac{A}{2i\sqrt{1+(f_c/f_{drive})^2}} \int_{-\frac{t_{msr}}{2}}^{\frac{t_{msr}}{2}} dt e^{-i2\pi f_k t} \\ &\quad [e^{-i2\pi f_{drive}(t-t_{lag})} - e^{i2\pi f_{drive}(t-t_{lag})}]. \end{aligned} \quad (100)$$

Rearranging the exponentials gives

$$\hat{x}_{resp}(f_k) = \frac{A}{2i\sqrt{1+(f_c/f_{drive})^2}} \int_{-\frac{t_{msr}}{2}}^{\frac{t_{msr}}{2}} dt \left[e^{-i2\pi f_{drive} t_{lag}} e^{-i2\pi(f_k+f_{drive})t} - e^{i2\pi f_{drive} t_{lag}} e^{-i2\pi(f_k-f_{drive})t} \right] \quad (101)$$

$$= \frac{A}{2i\sqrt{1+(f_c/f_{drive})^2}} \left[\frac{e^{-i2\pi f_{drive} t_{lag}} e^{-i2\pi(f_k-f_{drive})t}}{i2\pi(f_k+f_{drive})} - \frac{e^{i2\pi f_{drive} t_{lag}} e^{-i2\pi(f_k+f_{drive})t}}{i2\pi(f_k-f_{drive})} \right]_{-\frac{t_{msr}}{2}}^{\frac{t_{msr}}{2}} \quad (102)$$

$$= \frac{A}{2i\sqrt{1+(f_c/f_{drive})^2}} \left[\frac{e^{-i2\pi f_{drive} t_{lag}} \left[e^{i2\pi(f_k-f_{drive})\frac{t_{msr}}{2}} - e^{-i2\pi(f_k+f_{drive})\frac{t_{msr}}{2}} \right]}{i2\pi(f_k+f_{drive})} - \frac{e^{i2\pi f_{drive} t_{lag}} \left[e^{-i2\pi(f_k-f_{drive})\frac{t_{msr}}{2}} - e^{-i2\pi(f_k+f_{drive})\frac{t_{msr}}{2}} \right]}{i2\pi(f_k-f_{drive})} \right]. \quad (103)$$

Using the exponential definition for sine we can rewrite this as

$$\hat{x}_{resp}(f_k) = \frac{A}{2i\sqrt{1+(f_c/f_{drive})^2}} \left[\frac{e^{-i2\pi f_{drive} t_{lag}} \sin(2\pi(f_k-f_{drive})\frac{t_{msr}}{2})}{\pi(f_k+f_{drive})} - \frac{e^{i2\pi f_{drive} t_{lag}} \sin(2\pi(f_k+f_{drive})\frac{t_{msr}}{2})}{\pi(f_k-f_{drive})} \right]. \quad (104)$$

At this point we can return to the definition of the power spectrum, Eq. (97) and insert $\hat{x}_{resp}(f)$

$$P_{resp}(f_k) = \frac{2 \langle |\hat{x}_{resp}(f)|^2 \rangle}{t_{msr}} \quad (105)$$

$$P_{resp}(f_k) = \frac{A^2}{2(1+(f_c/f_{drive})^2)t_{msr}} \left[\frac{\sin^2(2\pi(f_k-f_{drive})\frac{t_{msr}}{2})}{\pi^2(f_k-f_{drive})^2} - \frac{\sin(2\pi(f_k+f_{drive})\frac{t_{msr}}{2}) \sin(2\pi(f_k-f_{drive})\frac{t_{msr}}{2}) [e^{i4\pi f_{drive} t_{lag}} + e^{-i4\pi f_{drive} t_{lag}}]}{\pi^2(f_k+f_{drive})(f_k-f_{drive})} + \frac{\sin^2(2\pi(f_k+f_{drive})\frac{t_{msr}}{2})}{\pi^2(f_k+f_{drive})^2} \right]. \quad (106)$$

The cross term will typically be several orders of magnitude smaller than the other two terms and is therefore ignored

$$P_{resp}(f_k) = \frac{A^2}{2(1 + (f_c/f_{drive})^2)t_{msr}} \left[\frac{\sin^2(\pi(f_k + f_{drive})t_{msr})}{\pi^2(f_k + f_{drive})^2} + \frac{\sin^2(\pi(f_k - f_{drive})t_{msr})}{\pi^2(f_k - f_{drive})^2} \right]. \quad (107)$$

We can define part of the equation above as a term r_k

$$r_k = \frac{1}{\sqrt{2}} \frac{\sin(\pi(f_k \pm f_{drive})t_{msr})}{(f_k \pm f_{drive})t_{msr}}. \quad (108)$$

This almost contains a Dirac delta function definition,

$$\delta(f \pm f_{drive}) = \frac{1}{\pi} \lim_{t_{msr} \rightarrow \infty} \left[\frac{\sin(\pi t_{msr}(f_k \pm f_{drive}))}{f_k \pm f_{drive}} \right] = \frac{\sqrt{2} t_{msr}}{\pi} r_k. \quad (109)$$

Since r_k comes close to a Dirac delta function for normal sampling times, we can simplify Eq. (107) as we are only interested in the positive half of the power spectral density (PSD) and Dirac delta functions filter out either the positive or negative frequencies. Therefore we drop the negative half $\delta(f_k + f_{drive})$.

$$P_{resp}(f_k) = \frac{A^2}{2\pi^2(1 + (f_c/f_{drive})^2)} \frac{\sin^2(\pi(f_k - f_{drive})t_{msr})}{(f_k - f_{drive})^2 t_{msr}} \quad (110)$$

$$= \frac{A^2 r_k^2 t_{msr}}{\pi^2(1 + (f_c/f_{drive})^2)} \quad (111)$$

$$= \frac{A^2 r_k^2 t_{msr} f_{drive}^2}{\pi^2(f_{drive}^2 + f_c^2)}. \quad (112)$$

If we choose the measurement time t_{msr} to be an integer stage period we can set $f_k = f_{drive}$, which allows us to write

$$r_k = \frac{\pi}{\sqrt{2}} \delta_{f_k f_{drive}}. \quad (113)$$

Inserting this into Eq. (112) give

$$P_{resp}(f_k) = \frac{A^2 f_{drive}^2}{2(f_k^2 + f_c^2)} t_{msr} \delta_{f_k f_{drive}}. \quad (114)$$

From this equation it is clear that the spike will increase linearly in size as the measurement time is increased, if we let $t_{msr} \rightarrow \infty$ the spike will become infinitely sharp and therefore we have,

$$t_{msr} \delta_{f_k f_{drive}} \rightarrow \delta(f - f_{drive}), \quad t \rightarrow \infty \quad (115)$$

Using this equation in the continuous limit,

$$P_{resp}(f) = \frac{A^2 f_{drive}^2}{2(f^2 + f_c^2)} \delta(f - f_{drive}). \quad (116)$$

Taking Eq. (99) and Eq. (116) gives us the power spectrum.

$$P(f) = \frac{D}{\pi^2(f_c^2 + f^2)} + \frac{A^2 f_{drive}^2}{2(f^2 + f_c^2)} \delta(f - f_{drive}). \quad (117)$$

A typical spectra for a trapped particle is shown in Figure 7.

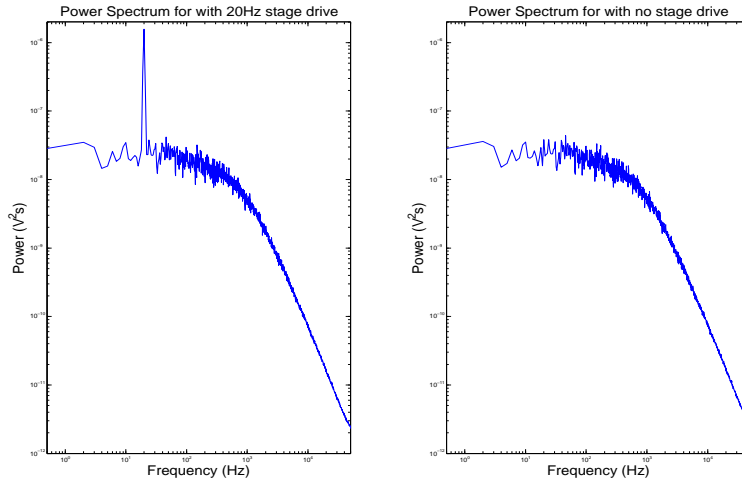


Figure 7: Power spectra for a trapped particle with and without stage movement

Alternative method for obtaining drive response spectrum Initially we follow a similar procedure as in section 2.7.4, but when we arrive at Eq. (107) we can directly use a definition for a delta function[9, p224],

$$\delta(f \pm f_{drive}) = \frac{1}{\pi^2} \lim_{t_{msr} \rightarrow \infty} \frac{\sin^2(\pi t_{msr}(f \pm f_{drive}))}{t_{msr}(f \pm f_{drive})^2}, \quad (118)$$

giving

$$P(f) = \frac{A^2 f_{drive}^2}{2(f^2 + f_c^2)} [\delta(f - f_{drive}) + \delta(f + f_{drive})]. \quad (119)$$

Here again we can disregard the negative frequencies, thus ignoring $\delta(f + f_{drive})$, this brings us straight to Eq. (116) where we follow on from there. Although it must be noted that this method does not highlight the procedure for obtaining the experimental power as clearly.

2.7.5 Calibration

In order to calibrate the tweezer we need to assume that the Quadrant photo diode voltage is linear with position,

$$x(t) = \beta x^{voltage}(t). \quad (120)$$

This allows us to write the PSD for the response as

$$P_{resp}(f) = \beta^2 P_{resp}^{voltage}(f). \quad (121)$$

Theoretical power We can determine β from the experimental peak and we know the drive amplitude and frequency from the experimental parameters. By rearranging Eq. (121) and integrating over f to obtain the power we can obtain an expression for β

$$\beta = \sqrt{W_{th}/W_{ex}}. \quad (122)$$

Here W_{th} and W_{ex} are the expressions for the power of the theoretical and experimental peaks in the spectral density. For the theoretical movement, we know that the spectral density will have the shape as in Eq. (116), therefore we can find the power by integrating this over the measured frequency range:

$$W_{th} = \int_0^{F_{Nyq}} P_{resp}(f) df = \frac{A^2}{2(1 + f_c^2/f_{drive}^2)}. \quad (123)$$

Experimental power Following on from Eq. (114) we saw that if f_{drive} is an integer multiple of f_k the spike will consist of a single datum, when this happens the spike should represent only the power at the drive frequency, hence we can write

$$W_{ex} = [P^{voltage}(f_{drive}) - P_T^{voltage}(f_{drive})] \Delta f. \quad (124)$$

Where we have subtracted the thermal background, and multiplied by the width of the spike (Δf in this case, as we have a single datum), although it is worth to note that in many cases it might not be necessary to subtract the thermal background as it will often be several orders of magnitude smaller.

2.7.6 Using the experimentally determined data

From the fit we obtain in Eq. (99) obtain values for D_{volt} and f_c while the spike in Eq. (116) gives a value for β . D_{volt} is quickly converted to its standard units by multiplying by β^2 ,

$$D = \beta^2 D^{volt}. \quad (125)$$

Using the Einstein relation and the definition of f_c (Eq. (59))

$$f_c = \frac{\kappa}{2\pi\gamma_0} \Rightarrow \gamma_0 = \frac{\kappa}{2\pi f_c}$$

$$D = \frac{k_B T}{\gamma_0}.$$

Putting these two together along with Eq. (125)

$$D = \frac{k_B T}{\frac{\kappa}{2\pi f_c}} = \frac{k_B T 2\pi f_c}{\kappa}$$

$$\kappa = \frac{2\pi k_B T f_c}{D}.$$
(126)

Gives us an expression for κ_{ex}

$$\boxed{\kappa_{ex} = \frac{2\pi f_c k_B T}{\beta^2 D^{volt}}}.$$
(127)

By taking the Einstein relation we can also find a value for the γ and hence the viscosity (η) which is independent of any tabulated values and hence obtain a check on the experimental results,

$$D = \frac{k_B T}{\gamma_0} \quad (128)$$

$$\gamma_0 = \frac{k_B T}{D}. \quad (129)$$

Which gives us an expression for γ_{ex}

$$\boxed{\gamma_{ex} = \frac{k_B T}{\beta^2 D^{volt}}}.$$
(130)

Using the stokes Einstein relation,

$$\gamma = 6\pi\eta r \quad (131)$$

$$(132)$$

we arrive at an expression for the viscosity,

$$\boxed{\eta = \frac{\gamma}{6\pi r}}. \quad (133)$$

3 Dynamic Light Scattering

3.1 Theory, Dynamic light scattering

3.1.1 Diffusion

As we already know small enough particles undergo Brownian motion, moving randomly due to the thermal forces applied to them by the surrounding fluid. We will make use of this by applying Dynamic light scattering to measure the fluid viscosity and particle size by use of the Einstein and Stokes relations. The theory was taken from lectures on experimental physics, [18]

Calculating the diffusion coefficient We assume that particles are in a potential field (U) where the density is given by a Boltzmann distribution:

$$\rho_{eq} = \rho_0 e^{-\frac{U}{k_B T}}. \quad (134)$$

In a steady state the flux of the particles due to diffusion and forces resisting diffusion will be equal,

$$J_D + J_F = 0. \quad (135)$$

Where J_D , the flux due to diffusion is given by,

$$J_D = -\mathcal{D} \nabla \rho_{eq}, \quad (136)$$

and J_F , the flux due to forces on the particles as

$$J_F = v \rho_{eq}. \quad (137)$$

Inserting the expression for the potential we arrive at:

$$J_D = -\mathcal{D} \rho_0 e^{-\frac{U}{k_B T}} \left(-\nabla \frac{U}{k_B T} \right) = -\frac{\rho_{eq}}{k_B T} (-\nabla U), \quad (138)$$

$$J_F = v \rho_0 e^{-\frac{U}{k_B T}} = v \rho_{eq}. \quad (139)$$

Setting these equal and solving for the diffusion constant.

$$J_D + J_F = -\frac{\mathcal{D} \rho_{eq}}{k_B T} (-\nabla U) + v \rho_{eq} = 0 \quad (140)$$

$$\mathcal{D} = \frac{v k_B T}{-\nabla U} \quad (141)$$

We know that we can write $F = -\nabla U$ and that $\mu = v/F$,

$$\frac{v}{F} = \mu, \quad (142)$$

$$v = -\mu \nabla U, \quad (143)$$

allowing us to write an expression for the potential

$$\mu = \frac{\mathcal{D}}{k_B T}. \quad (144)$$

Using Eq. (142) and the stokes relation,

$$F = 6\pi\eta r. \quad (145)$$

we arrive at an expression for the diffusion coefficient,

$$\mathcal{D} = \frac{k_B T}{6\pi\eta R}. \quad (146)$$

3.1.2 Dynamic Light Scattering

To perform the experiment we will be using the auto correlation function. We know that a planar wave can be written as:

$$\mathbf{E} = \mathbf{E}_0 \mathbb{R}(e^{i(\mathbf{k} \cdot \mathbf{r} - \omega t)}). \quad (147)$$

We also know that we can write the intensity as the energy flux $\langle s \rangle$, and can therefore easily prove that

$$\langle s \rangle = \frac{\langle \mathbf{E}^2 \rangle}{\mu c} \propto \mathbf{E}_0^2 = \mathbf{E} \mathbf{E}^*. \quad (148)$$

When a single particle scatters light we will see something similar to Figure 8. Where the wave number $K_i = n \frac{2\pi}{\lambda}$ and with the refraction index n and wavelength λ . From this it is possible to determine that:

$$\mathbf{E}_s \propto \frac{\mathbf{E}_i(t_0)}{|\mathbf{R} - \mathbf{r}|}. \quad (149)$$

$$t_0 = t - \frac{|\mathbf{R} - \mathbf{r}|}{c}. \quad (150)$$

Where Eq. (150) shows the retarded time.

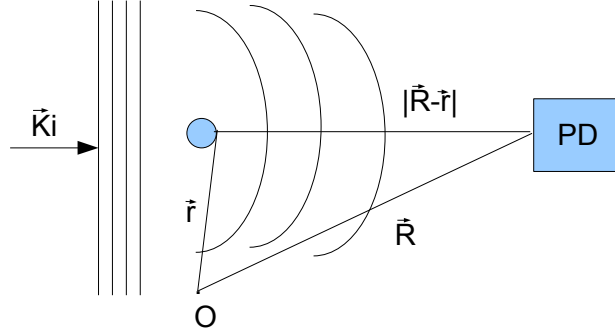


Figure 8: Scattering of an incoming wave, where O a point of reference

$$\mathbf{E}_s \propto \frac{\mathbf{E}_0}{|\mathbf{R} - \mathbf{r}|} \mathbb{R}(e^{i(\mathbf{k}_i \cdot \mathbf{r}(t_0) - \omega t_0)}). \quad (151)$$

Substituting in $|\mathbf{R} - \mathbf{r}| = R - \frac{R}{r} \cdot r$ and using the retarded time gives an expression for the scattered field:

$$\mathbf{E}_s \propto \frac{\mathbf{E}_0}{|\mathbf{R} - \mathbf{r}|} e^{iKR} e^{i(\mathbf{q} \cdot \mathbf{r}(t_0) - \omega t)}. \quad (152)$$

Where \mathbf{q} is the scattering vector $\mathbf{K}_i - \mathbf{K}_s$ which can be written as

$$\mathbf{q} = \frac{4\pi n}{\lambda} \sin\left(\frac{\theta}{2}\right), \quad (153)$$

n is the refractive index of the solution and λ is the wave length of the incident light. Assuming that $R \gg r$ and hence $\frac{r}{R} \ll 1$,

$$\mathbf{E}_s = \frac{\mathbf{E}_0}{R} e^{iKR} e^{i(\mathbf{q} \cdot \mathbf{r}(t_0) - \omega t)}. \quad (154)$$

Using this information we can now determine the auto correlation function:

$$G(\tau) = \frac{\langle I(t)I(t+\tau) \rangle}{\langle I(t) \rangle}. \quad (155)$$

We know from Eq. (148) that $I(t) \propto EE^*$ which means that we can set up an equation for $I(t)I(t+\tau)$ knowing that E_0 will be the same for all the particles

$$\langle I(t)I(t+\tau) \rangle \propto h(\tau) = \left\langle \sum_{ij} e^{i\mathbf{q}(\mathbf{r}_i(t_0) - \mathbf{r}_j(t_0))} \sum_{lm} e^{i\mathbf{q}(\mathbf{r}_l(t_0+\tau) - \mathbf{r}_m(t_0+\tau))} \right\rangle,$$

$$\text{since } \mathbf{r}_x(t_0 + \tau) = \mathbf{r}_x(t_0) + \Delta \mathbf{r}_x(\tau) = \mathbf{r}_x + \Delta \mathbf{r}_x, \quad (156)$$

$$h(\tau) = \left\langle \sum_{ijlm} e^{i\mathbf{q}(\mathbf{r}_i - \mathbf{r}_j + \mathbf{r}_l - \mathbf{r}_m)} e^{i\mathbf{q}(\Delta \mathbf{r}_l - \Delta \mathbf{r}_m)} \right\rangle. \quad (157)$$

Since the particles are statistically independent in position their contributions will only count when $\mathbf{r}_i - \mathbf{r}_j + \mathbf{r}_l - \mathbf{r}_m = 0$, when one of the terms i,j,l,m is different from the rest the overall contribution will be 0. Counting up the contributions from all these terms, there are N^2 terms for i=j, l=m and $e^{i\mathbf{q}(\Delta\mathbf{r}_l - \Delta\mathbf{r}_m)}$ gives 1 as l=m.

$$h(\tau)_{i=j, l=m} = N^2. \quad (158)$$

There are $(N^2 - N)$ terms for which $i = m, j = l$ and $m \neq l$,

$$h(\tau)_{i=m, j=l, m \neq l} = (N^2 - N) \langle e^{i\mathbf{q}(\Delta\mathbf{r}_l)} \rangle \langle e^{i\mathbf{q}(\Delta\mathbf{r}_m)} \rangle. \quad (159)$$

Assuming that the particles are completely independent allows us to write Eq. (157) as:

$$h(\tau) = (N^2 - N) \langle e^{i\mathbf{q}(\Delta\mathbf{r}_l)} \rangle \langle e^{i\mathbf{q}(\Delta\mathbf{r}_m)} \rangle + N^2. \quad (160)$$

Assuming an ergodic system we can rewrite the time average as an ensemble average, where $P(\Delta\mathbf{r}, \tau)$ is the probability that a particle had diffused a length $\Delta\mathbf{r}$ in time τ :

$$\langle e^{i\mathbf{q}\Delta\mathbf{r}} \rangle = \int P(\Delta\mathbf{r}, \tau) e^{i\mathbf{q}\Delta\mathbf{r}} d^3\Delta\mathbf{r}, \quad (161)$$

$$\langle e^{i\mathbf{q}\Delta\mathbf{r}} \rangle = \int \frac{1}{(4\pi\mathcal{D}\tau)^{\frac{3}{2}}} e^{-\frac{\Delta\mathbf{r}^2}{4\mathcal{D}\tau}} e^{i\mathbf{q}\Delta\mathbf{r}} d^3\Delta\mathbf{r}, \quad (162)$$

$$\langle e^{i\mathbf{q}\Delta\mathbf{r}} \rangle = e^{-\mathcal{D}\mathbf{q}^2\tau}. \quad (163)$$

Integrating this gives Eq. (163) Since these are two completely independent particles we can rewrite $h(t)$ as:

$$h(\tau) = N^2 + (N^2 - N)e^{-2\mathcal{D}\mathbf{q}^2\tau}. \quad (164)$$

Using the fact that $\langle I(t) \rangle = N$ we can find $G(\tau)$, and with the assumption that N is large ($N^2 \gg N$) this can be simplified to Eq. (167)

$$G(\tau) = \frac{\langle I(t)I(t+\tau) \rangle}{\langle I(t) \rangle} = \frac{N^2 + (N^2 - N)e^{-2\mathcal{D}\mathbf{q}^2\tau}}{N^2}, \quad (165)$$

$$G(\tau) = 1 + \frac{(N^2 - N)}{N^2} e^{-2\mathcal{D}\mathbf{q}^2\tau}, \quad (166)$$

$$G(\tau) = 1 + e^{-2\mathcal{D}\mathbf{q}^2\tau}. \quad (167)$$

Since the correlation function is measured and we know the scattering vector q from Eq. (153) we can determine \mathcal{D} . This can be done by noticing that the decay time is $1/2\mathcal{D}Q^2$ something that can easily be measured from a plot. And hence using Eq. (146) we can determine the viscosity of the fluid or the particle size.

3.2 Experimental setup

In Figure 9 we can see the experimental setup measurement of the correlation function. Briefly it consists of:

- The laser is a Helium-Neon laser with a relatively stable signal without too much interference.
- Half-wave plate, The maximum interaction between the particle and laser will occur parallel with the polarisation of the laser. As the laser is vertically polarised the maximum scattering was vertical. Placing the half-wave plate in place rotated the interaction towards the sensors.
- Two lenses for focusing purposes (focal length $\approx 10\text{cm}$).
- A sample cell containing silica beads, the concentration of beads was increased until the beam was visible and the count rate was just high enough, ensuring that multiple scattering was not an issue.
- Thermistor, a thermistor is placed in contact with the cell in order to obtain an accurate temperature reading for each run.
- A single slit and aperture (turned down to its smallest opening) are placed in the beam path to ensure that the projection from the sample allows only a small spot of light through. Allowing too much light to pass allows widely spaced particles to contribute to the correlation function, lowering the accuracy of this technique.
- A red filter is placed in front of the photo diode preventing most external sources of the stray light from entering.
- As the photo diode is extremely sensitive it requires shielding from as many external sources of light as possible.
- The photo diode is connected to an auto correlator which is read and analysed by a computer.

Measurements of 60 seconds were made for a total of 15 data sets for a suspension of particles with and without salt (NaCl) added. The salt was added in order to determine if adding a small amount of positively charged ions changed the shielding length and consequently reduce the particle size.

The data for the experimental setup is shown in Table 1.

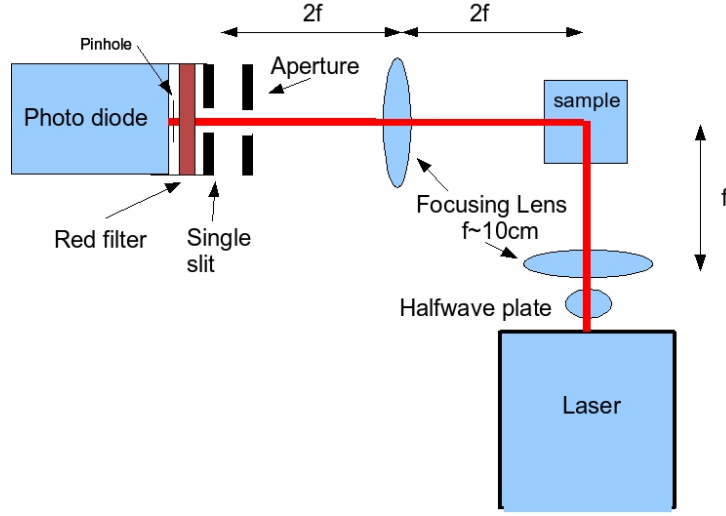


Figure 9: Experimental setup

Data	Value
Laser Wavelength (λ)	632.8 nm
Refractive index of suspension (n)	1.33
Particle Diameter ($2r$)	$1.54 \mu\text{m}$
Slit size	$0.5 \mu\text{m}$
Viscosity Water at 20°C (η)	$1.001 \cdot 10^{-3} \text{ Pa s}$
Density Water (ρ_w)	$1 \cdot 10^3 \text{ kg/m}^3$
Density Silica (ρ_p)	$1.96 \cdot 10^3 \text{ kg/m}^3$

Table 1: Experimental parameters

3.2.1 Particle size and settling time

As this experiment was usually performed with a particle size of $0.05 \mu\text{m}$ a quick calculation ensured that the $1.54 \mu\text{m}$ particles fall slowly enough for this to be a viable technique. Balancing the force of gravity with the bouncy of the particles it is possible to determine the terminal velocities of the particles, and hence calculate the time it takes to fall a distance of one wavelength.

$$\mathbf{F}_g = (\rho_p - \rho_w)V_p\mathbf{g}, \quad (168)$$

$$\mathbf{F}_{\text{Drag}} = -\gamma\mathbf{u}, \quad (169)$$

$$\gamma = 6\pi\eta r. \quad (170)$$

Where F_g is the force due to gravity, $\rho_{p,w}$ is the density of the particle and water respectively, g the acceleration due to gravity, γ is the drag coefficient, u the velocity, η is the viscosity of the water, V_p the volume of the particle and finally r is the radius of the particle. Replacing the volume with an equation for the volume of a sphere and calculating the total force (F_T), which during free fall should be zero gives:

$$\mathbf{F}_T = \mathbf{F}_g + \mathbf{F}_{Drag} = (\rho_p - \rho_w)V_p\mathbf{g} - 6\pi\eta r\mathbf{u}, \quad (171)$$

$$(\rho_p - \rho_w)V_p\mathbf{g} = 6\pi\eta r\mathbf{u}, \quad (172)$$

$$\mathbf{u} = \frac{(\rho_p - \rho_w)\frac{4}{3}\pi r^3\mathbf{g}}{6\pi\eta r}. \quad (173)$$

$$(174)$$

Setting the velocity (\mathbf{u}) to distance (λ)/time (τ),

$$\frac{\lambda}{\tau} = \frac{(\rho_p - \rho_w)\frac{4}{3}\pi r^3 g}{6\pi\eta r}, \quad (175)$$

$$\tau = \frac{9\lambda\eta}{2(\rho_p - \rho_w)r^2 g}. \quad (176)$$

Using the values tabulated in Table 1, where λ is assumed to be the laser wavelength, gives a settling time of 0.5s for a 1.54μ m bead. This number is much larger than the time between collisions and does not hinder the use of this method.

3.2.2 Effect of salt on viscosity and refractive index

Adding salt to the suspension of particles in water will have an effect on the viscosity of water. For the Dynamic light scattering experiment a mass concentration of 0.25% was used, assuming a linear response⁴ of viscosity to a change in mass concentration we can interpolate the data in Table 3.2.2 to obtain the viscosity of the salt solution.

$$\eta = \eta_A + \frac{(m - m_a)(\eta_b - \eta_a)}{(m_b - m_a)}, \quad (177)$$

$$= 1.001 \cdot 10^{-3} + \frac{(0.25 - 0)(1.011 \cdot 10^{-3} - 1.001 \cdot 10^{-3})}{0.5 - 0}, \quad (178)$$

$$= 1.0065 \cdot 10^{-3} \text{Pa s}. \quad (179)$$

Similarly adding salt will change the refractive index of the solution, again we assume a linear response⁵ to change in mass concentration. Interpolating

⁴This is a reasonable assumption of the viscosity, and can be clearly seen when data is plotted for small mass concentrations[17]

⁵This can be clearly seen when data is plotted[17]

for a mass concentration of 0.25%,

$$\eta = \eta_A + \frac{(m - m_a)(\eta_b - \eta_a)}{(m_b - m_a)}, \quad (180)$$

$$= 1.3330 \cdot 10^{-3} + \frac{(0.25 - 0)(1.3339 \cdot 10^{-3} - 1.3330 \cdot 10^{-3})}{0.5 - 0}, \quad (181)$$

$$= 1.3336 \cdot 10^{-3} \text{Pa s}. \quad (182)$$

This gives a very small correction to the particles size and viscosity, the calculations used for results will take this change into account.

Mass Concentration % by mass NaCl	Viscosity 10^{-3}Pa s	Refractive index
0	1.002	1.3330
0.5	1.011	1.3339

Table 2: Viscosity for Sodium Chloride solution.[17]

3.3 Results

In total 15 data sets were measured for the particles with and without NaCl (0.04M solution), these were averaged and fitted with the function $1 + A e^{-B\tau}$ where B is $2\mathcal{D}q^2$.

The fitting parameters A and B are listed in Table 3.3. Using these in conjunction with $B = 2\mathcal{D}q^2$. We can find a value for \mathcal{D} and hence obtain a value for either the viscosity or particle size.

$$\text{Viscosity; } \eta = \frac{k_B T}{6\pi r \mathcal{D}} \quad (183)$$

$$\text{Radius; } r = \frac{k_B T}{6\pi \eta \mathcal{D}} \quad (184)$$

These equations require either the particle radius or fluid viscosity and hence do not allow both to be obtained simultaneously. To allow an easy comparison the values for both have been listed in Table 3.3 using tabulated values from Table 1 for the unknowns.

In order to get an idea for the stability of the measurements the two data sets were split up into two halves, the results are tabulated in the Table 3.3. These clearly show that there is a large separation between the Salted and Unsalted data sets of at least $0.04 \mu\text{m}$.

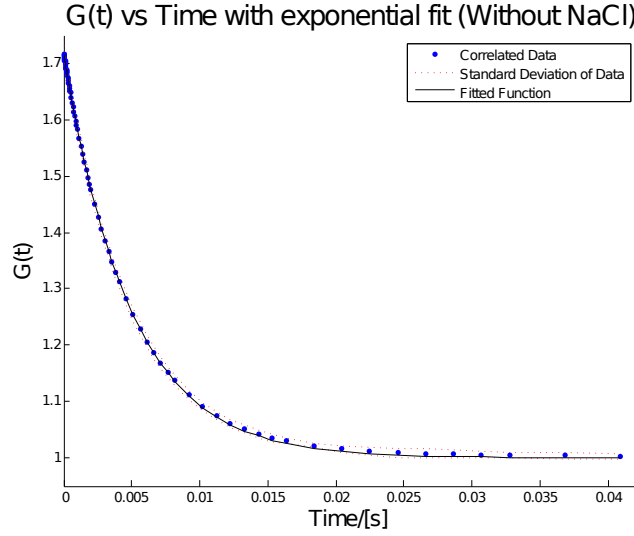


Figure 10: Correlation function for a sample without added Salt

	Fit Parameters		Particle Diameter	Viscosity	Temp
	A	B	μm	Pa s	K
Unsalted	0.7190(5)	204.22(4)	$1.614(3) \cdot 10^{-6}$	$9.38(2) \cdot 10^{-4}$	297.93
Salted	0.6826(7)	212.39(7)	$1.545(5) \cdot 10^{-6}$	$9.02(3) \cdot 10^{-4}$	297.73

Table 3: Experimental results for Dynamic light scattering with and without salt

These results are consistent with the manufacturers specifications of $1.54 \mu m \pm 10\%$ standard deviation. Both the Unsalted and Salted samples fall within 10% of the expected results. Although it should be noted that the angular alignment of this experiment is complicated to perform accurately. Errors up to a couple of degrees are very likely, resulting in a minimum uncertainty 4–5%. The diffusion coefficients for Unsalted and salted were $2.92 \cdot 10^{-13} m^2/s$ and $3.04 \cdot 10^{-13} m^2/s$ respectively, these are very close to the calculated value of $3.01 \cdot 10^{-13}$.

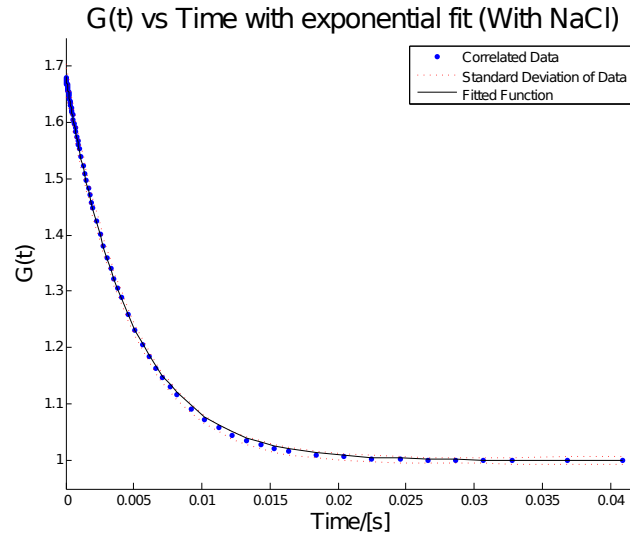


Figure 11: Correlation function for a sample with added Salt[0.04M]

	Particle Diameter	First 7	Last 7
	μm	μm	μm
Unsalted	$1.614(3) \cdot 10^{-6}$	$1.618(4) \cdot 10^{-6}$	$1.614(3) \cdot 10^{-6}$
Salted	$1.545(5) \cdot 10^{-6}$	$1.523(6) \cdot 10^{-6}$	$1.573(5) \cdot 10^{-6}$

Table 4: Experimental results showing stability of measurements

4 Equipment review

4.1 Introduction

In the following section I will describe the equipment used for trapping a single particle in the optical tweezer lab. I will also go into detail on alignment procedure, a requirement for getting good data.

4.2 Laser

The laser used for this setup is a water cooled LEXEL 95-3 argon ion laser, capable of delivering up to 3 W in a multi line setup. Although for our purposes it will be limited to a maximum of 1070 mW at the 514.5 nm line in a TEM_{00} mode (see Figure 12). An external controller is used to control the emitted power of the laser and ensures that the water cooling is running before the laser is switched on. The laser itself has very few adjustments, they consist of a height adjust at each of the four corners and an adjustment for the rear mirror. Adjusting the height allows us to control the way that the beam travels along the table. Adjusting the mirror allows us to choose the wavelength that the laser operates at and adjusts the beams position with respect to the output aperture. This argon ion laser is capable of operating

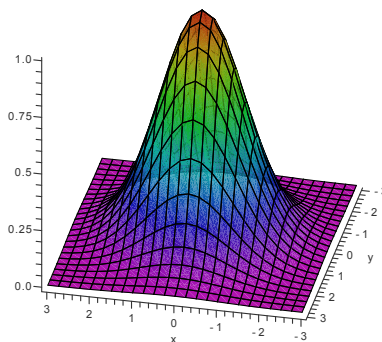


Figure 12: TEM_{00} mode, X and Y axis show spatial position, Z shows light intensity (where 1 is the maximum)

at the wavelengths listed in Table 5. The laser is capable of delivering most of the power at 514.5 nm, and this is also the wavelength that the optics have been optimised for. To choose the wavelength a dial on the rear of the laser is adjusted until the desired wavelength is visible and the expected power is measured. Mirror adjustment works by reflecting a chosen wavelength back through the prism along the optical path as shown in Figure 13.

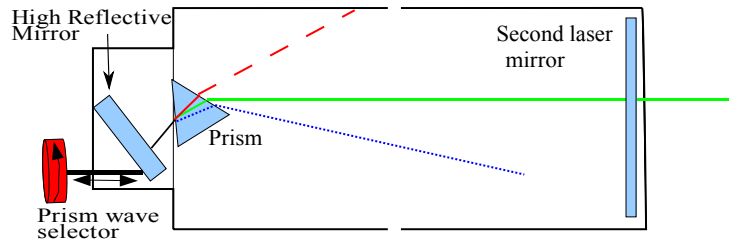


Figure 13: Prism Wavelength selector

Wavelength nm	Power mW
1092.3	100
528.7	250
514.5	1200
501.7	200
496.5	400
488.0	700
476.5	400

Table 5: Tabulated specifications for laser power levels[16]

4.2.1 Laser Controller

The Controller provides the laser with a stable power supply and a source of cooling. The power supply operates from 208 V and requires a 3 phase electrical service capable of supplying a minimum of 35 amps, the unit itself is protected using a 30 amp fuse. The water supply should have a flow rate of 7.6 litres per minute at a pressure of 1.7 atmospheres. The laser will operate in two modes:

- **Current control:** In current control the plasma-tube current is kept at a constant level which results in a constant power. Unfortunately it does not compensate for changes in the internal optics, resulting in fluctuations in output power. Stability is generally attained after 60 minutes (see Section 4.2.3).
- **Light control:** In light control a small portion of the laser light is split from the main beam and measured. A feedback regulator circuit compares the output intensity to a value set by the operator, and maintains a constant intensity by varying the current in response to the measured error. See Figure 14 for a diagram of the setup. When operated at the maximal current a drift can occur as the laser will not be able to compensate for changes in the optics.

4.2.2 Powering up the laser

Powering up the laser requires a number of steps to be taken:

- Cooling water has to be switched on, the laser will not power up unless a steady stream of water at a minimum rate of 7.6 l/min at a pressure of 1.7 atmospheres. If the laser is already in operation it will automatically shut down to prevent damage from overheating.
- The main power switch is switched on, this should light the “line” and “fuse” indicators.
- Key switch should be switch on, this should light the “INTLK” indicator, assuming that all covers are closed, water flow and temperature are adequate, and regulator temperature are within bounds. Should any of these items not pass the test their respective indicator will not light.
- The method of power control is chosen, this can be either current or light control

- The power on button is pressed, this will cause the power supply to activate and will start warming the cathode in the laser, this will also light the laser radiation emission indicator.
- Once the cathode has obtained operating temperature (this takes approximately 30 seconds) a ready indicator should light.
- At this point the laser is ready to lase and will do so once the laser start button is pressed.
- The power control should then be adjusted using the respective dial.

4.2.3 Laser beam stability

In order to test the stability of the laser a series of output intensity measurements were made. The measurements were made using a Thorlabs PM100 with a 50 mW (S120A) laser power sensor. The sensor contains correction factors which are internally applied in the PM100 to deliver a calibrated result to the display and via a serial connection. All stability measurements were made after the beam splitter and before the beam expander. The laser was set to a power of approximately 350mW which was reduced to 28mW using the half wave plate and beam splitter in order to remain below the 50mW limitation of the laser power meter.

Short term intensity stability Current and light control both aim to provide a stable beam with minimal fluctuations, to verify this a series of measurements over a 3 minute period were made, this was short enough so that no significant amount of heating would be measured in the power sensor but should provide a good representation of the power fluctuations. Figure 15 shows that the light control is better able to correct the plasma-tube current so that a stable beam is observed. Light control had a peak-to-peak stability

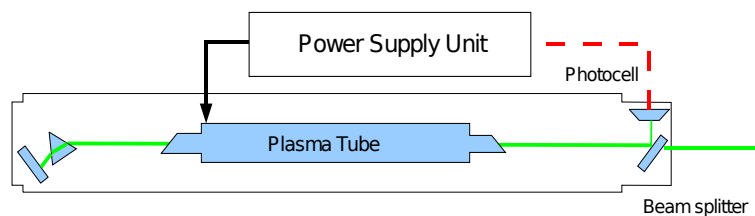


Figure 14: Laser internal diagram for light control

of less than 0.05 mW, while the current control had a peak-to-peak stability of approximately 0.15 mW.

Long term intensity Stability In order to characterise the long term stability of the laser a set of power measurements were made over a period of a several hours. The heating of the sensor due to the laser meant that we had a continual drift during initial measurements. To overcome this a programmable beam blocker was set up to allow the beam to pass for ten seconds for every minute. This was done using a microchip PIC12F629 micro controller which was programmed to control a hobby servo motor using pulse width modulation. The micro controller sent a full left pulse for 10 seconds and then a full right pulse for the next 50 seconds, allowing the beam to pass for ten seconds in every minute. For full details on the controller see appendix A.1. The setup is shown in Figure 16.

Figure 17 shows the measured power as a function of time for light and current control. Both plots clearly show that the laser requires some time to warm up and stabilise. The manual recommends waiting at least 30 minutes when using current control, but from the figure it is clear that it might be advisable to wait at least 1 hour in order to avoid the rapid change in intensity seen in both modes. Light control does not have as large fluctuations but takes longer to stabilise, requiring close to two hours before it levels off. After one hour the change in intensity varied only by 0.25 mW for light control, while it is just over 0.5 mW for current control in the following 4 hours.

Beam stability In general the beam is sufficiently stable so as not to effect the measurements by any substantial amount, this is especially true because the quadrant photo diode measures variations in the spacial power and should not be effected as much by the temporal fluctuations. Additionally any fluctuations seen in the X and Y channels are normalised by dividing by the total intensity Z.

4.2.4 Laser power and noise

During attempts at measuring movement of a particle in the trap it was noticed that there was a slight bump in the positional power spectrum, the source was identified as coming from the laser, furthermore it was noted that the amount of noise was dependent on the operating power of the laser and independent of the power in the trap. In order to verify this a series of measurements were made by varying the laser power and keeping the trapping beam at constant intensity (by using the half wave plate and beam splitter). A constant power of approximately 21.3 mW was kept throughout

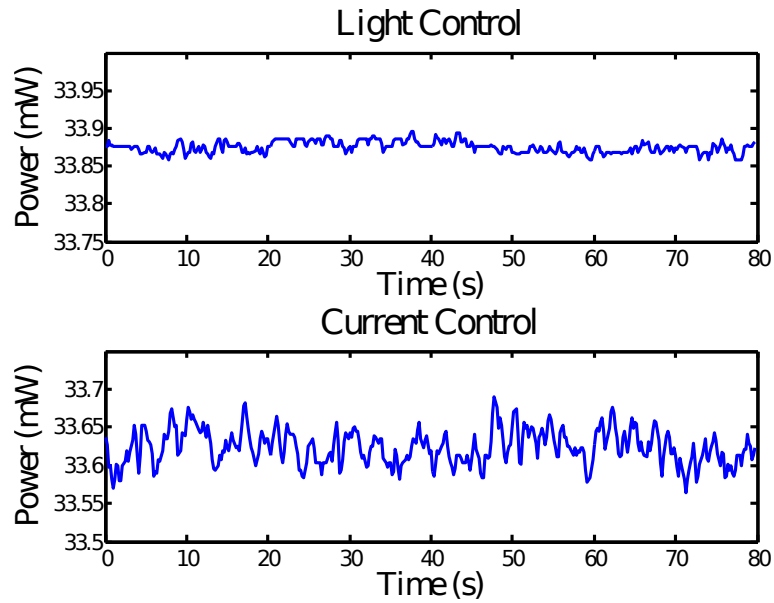


Figure 15: Measured laser intensity for current and light Control modes, plotted on same scale with a reduced time period for clarity

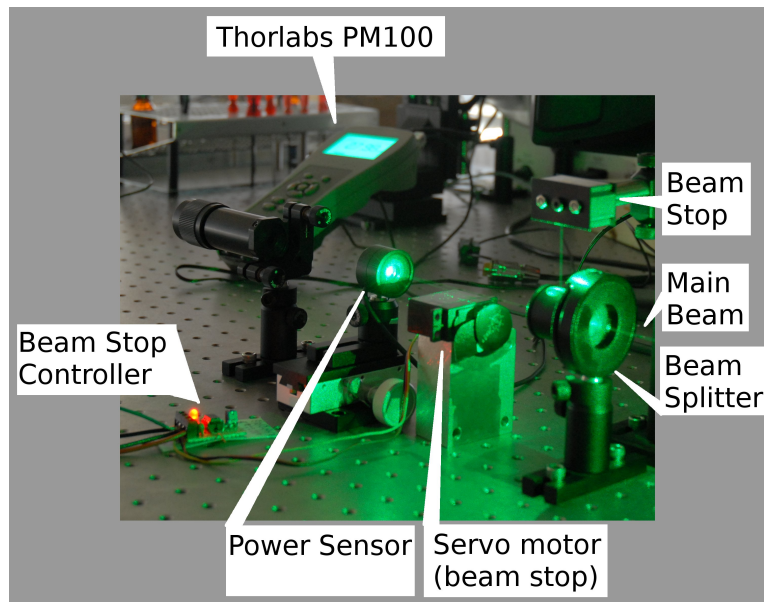


Figure 16: Setup used to block laser beam in order to avoid heating the laser power sensor. The servo motor rotates a disc up to block the laser beam at pre-programmed intervals

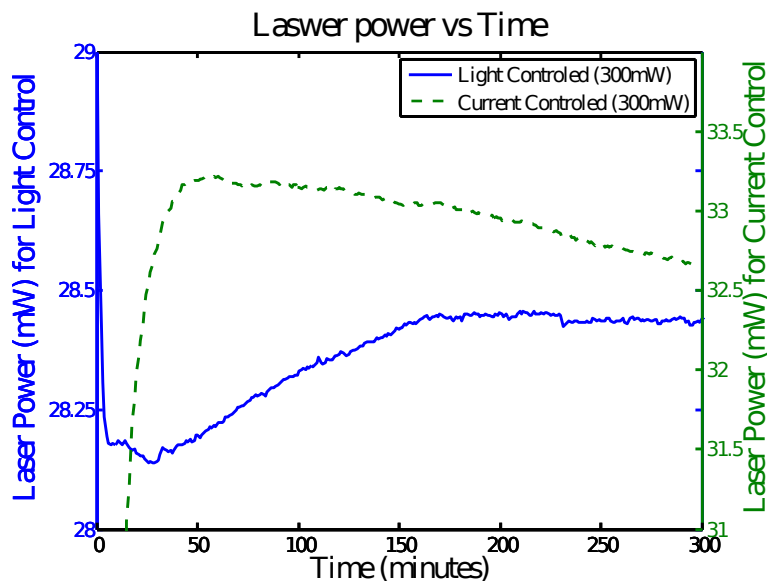


Figure 17: Measured laser power plotted against time for light and current control

the experiment with the laser intensity varying from 28 mW to 1 W, the results of which can be seen in 18. From these results it is clear that if possible a power of around 200mW should be chosen to have as low a noise floor as possible.

4.3 Optics

The optical path between the laser and the microscope are described in Figure 19.

- **Halfwave Plate** The Laser light first passes through the half wave plate (Thorlabs WPH05M-514), this allows the rotation of the polarisation of the Laser light as the light emitted is vertically polarised.
- **Beam splitter** Together with the half wave plate, the polarising beam splitter (Thorlabs GL10-A) allows the power of the beam to be varied. Any light which is polarised parallel to the prism will pass through while the rest is reflected vertically, eventually hitting the beam stopper. It should be noted that the polarisation should be selected so as to maximise reflection in the dichroic mirror in the microscope, in our case vertically polarised.
- **Beam expander** The beam expander is a Linos 7x expander. Its main

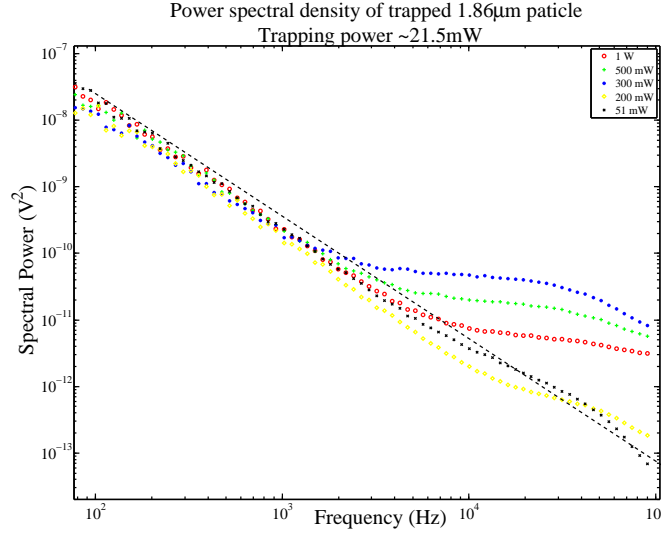


Figure 18: Power spectrum for the laser beam at selected intensities (for clarity)

purpose is to enlarge the cross-section of the laser beam (about 1.3 mm) to overfill the back aperture of the microscope (about 7 mm) such that we maximise the light gradient in the axial direction of the beam for maximal trapping. This is done so as to obtain the largest convergence angle (ϕ) possible, $\tan \phi = r_{\max} / l$, where r_{\max} is the rear aperture of the lens and l is the distance from the lens to the focus, for a more detailed overview see[3]. It is important to get the correct size as when the diameter of the incoming beam is approximately the same as the objective results in the strongest trap. Should the beam overfill the objective the trap beam will have a short trapping depth and because all the light is not transmitted the trapping intensity will be lower.

- **Mirrors** These mirrors (Thorlabs RS99/M) are placed in a periscope setup to allow steering of the beam and allow the beam to enter the microscope through the epi-fluorescence port at the correct height.
- **Telescope Lenses** The first of these two lenses (Thorlabs LA1509–9, 100mm focal length) is mounted on a stage, this allows an accurate adjustment of the trap position within the sample cell. Together with the second lens they can accurately steer the beam so as to get a parallel beam hitting the microscope objective.

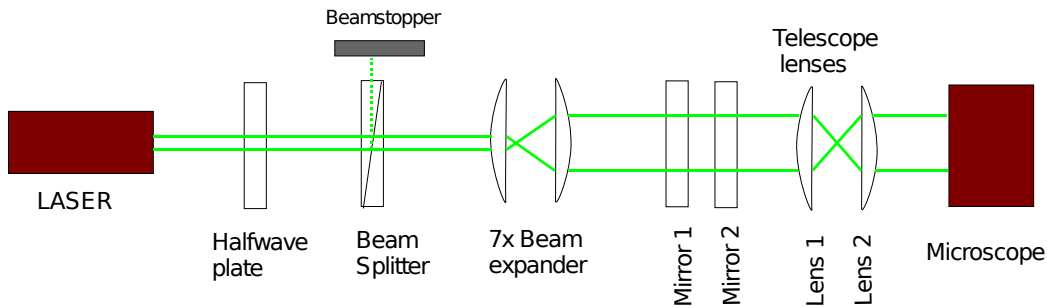


Figure 19: Optical path between Laser and microscope

4.3.1 Optical Losses

To characterise the optical losses, measurements were made by placing the PM100 laser power meter between each stage in the optics. The measurements have been presented in Figure 20. The results indicate that there is a loss of approximately 17% between the beam splitter and to the point where it enters the objective. This is something that will have to be taken into account when a desired trapping intensity is required.

4.4 Microscope

The microscope used for the optical tweezers is an inverted compound microscope, Leica DM IRB. A simplified diagram of the microscope internals is shown in Figure 21. The parallel laser light entering the microscope hits a dichroic mirror and is reflected into the objective lens. The focused beam passes through the slide containing the particles to be trapped and then through the condenser lens. Once through the condenser lens it will reflect off a second dichroic mirror, and then hits a quadrant photo diode. Light from the microscope is used to illuminate the sample and passes through both the dichroic mirrors and can be observed by either a person or a camera.

- **Filter Cube** The filter cube (Chroma Technology corp, Z514RDC) blocks a selective band of light from approximately 500 nm to 520 nm. This ensures that no laser light can damage the eyes of anybody viewing the trap.
- **Objective** The objective is a Leica 63x magnification 1.2 Numerical Aperture water immersion objective, part number 11506213
- **Condenser** The condenser is a Leica 1.4 Numerical Aperture oil immersion lens

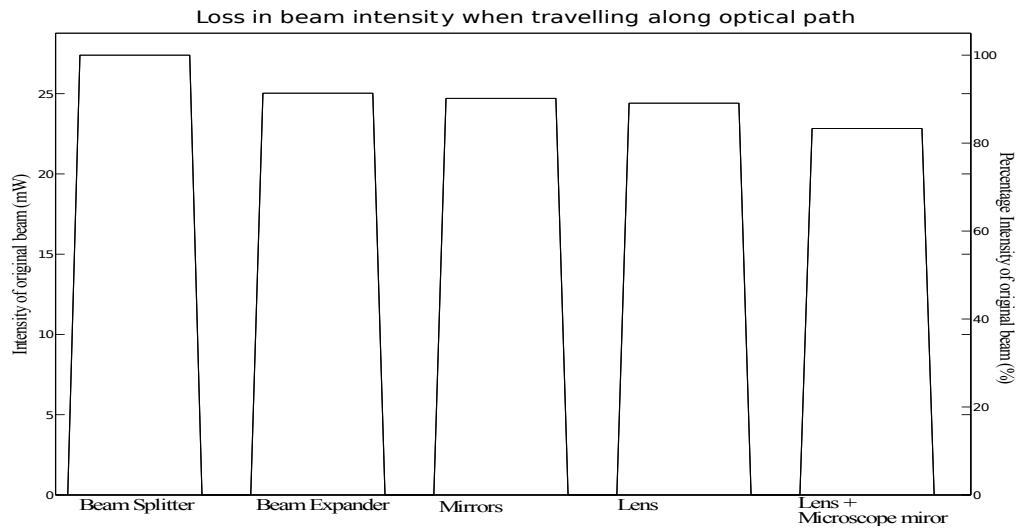


Figure 20: Loss in intensity as beam travels from the beam splitter (measured as 100%) passing through optics until it reaches the microscope objective

Microscope

Model: Leica DM IRB
 Tube length (L): 200mm

Objective

Leica part number: 11506213
 Magnification (m): 63x
 Numerical Aperture: 1.2

Condenser

Leica part number: 11551004
 Numerical Aperture: 1.4

General

Lateral Resolution: 261.5nm[21]
 Max Working distance: 220 μ m[15]
 Focal length L/m: 3.17mm
 F Number ($f^\#$): 0.554
 Diffraction limited Spot diameter:⁶ 0.174 μ m

Table 6: Microscope specifications

[21]

4.5 Quadrant Photo diode (QPD)

The QPD is a set of four photo diodes (Figure 22) made from a single silicon chip, this is to ensure uniform sensitivity and similar performance between the diodes. The photo diode module (Pacific Silicon, QP50–6SD2) contains an integrated circuit which performs difference and summation of the 4 diode voltages. The voltage on the photo diodes are proportional to the incident intensity and are used to provide three signals V_x, V_y and V_z which are independently measured by the DAQ.

$$V_x = (V_A + V_B) - (V_C + V_D) \quad (185)$$

$$V_y = (V_A + V_C) - (V_B + V_D) \quad (186)$$

$$V_z = V_A + V_B + V_C + V_D \quad (187)$$

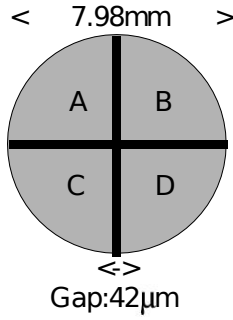


Figure 22: QPD

The QPD will be used to sense the position of the particles as the fluctuations in the laser beam intensity causes changes in the measured voltage. The photo diode module has two modes of operation:

- **Zero bias mode;** no external potential is applied to the amplifier. In this mode the photo diode operates in a photovoltaic mode, causing a current to flow across the device when light is incident on it.
- **Reverse bias mode;** by applying a voltage (V_{pad1}) of up to +15V to the amplifier via pad 1. This will apply a bias voltage to the photo diode of 0.91 times V_{pad1} . In contrast to zero bias mode a current is applied to the photo diode causing a widening in the depletion layer and hence a strengthened photocurrent. This in turn leads to a higher sensitivity for light and a lower capacitance improving the

response time. Unfortunately this comes at the cost of more noise compared to zero bias mode.

In our setup the QPD will be used with a reverse bias of $\pm 0V$. The technical details of the photo diode are giving in Table 7[22]

⁶The spot diameter is given by $1.22f\#\frac{\lambda}{2}$

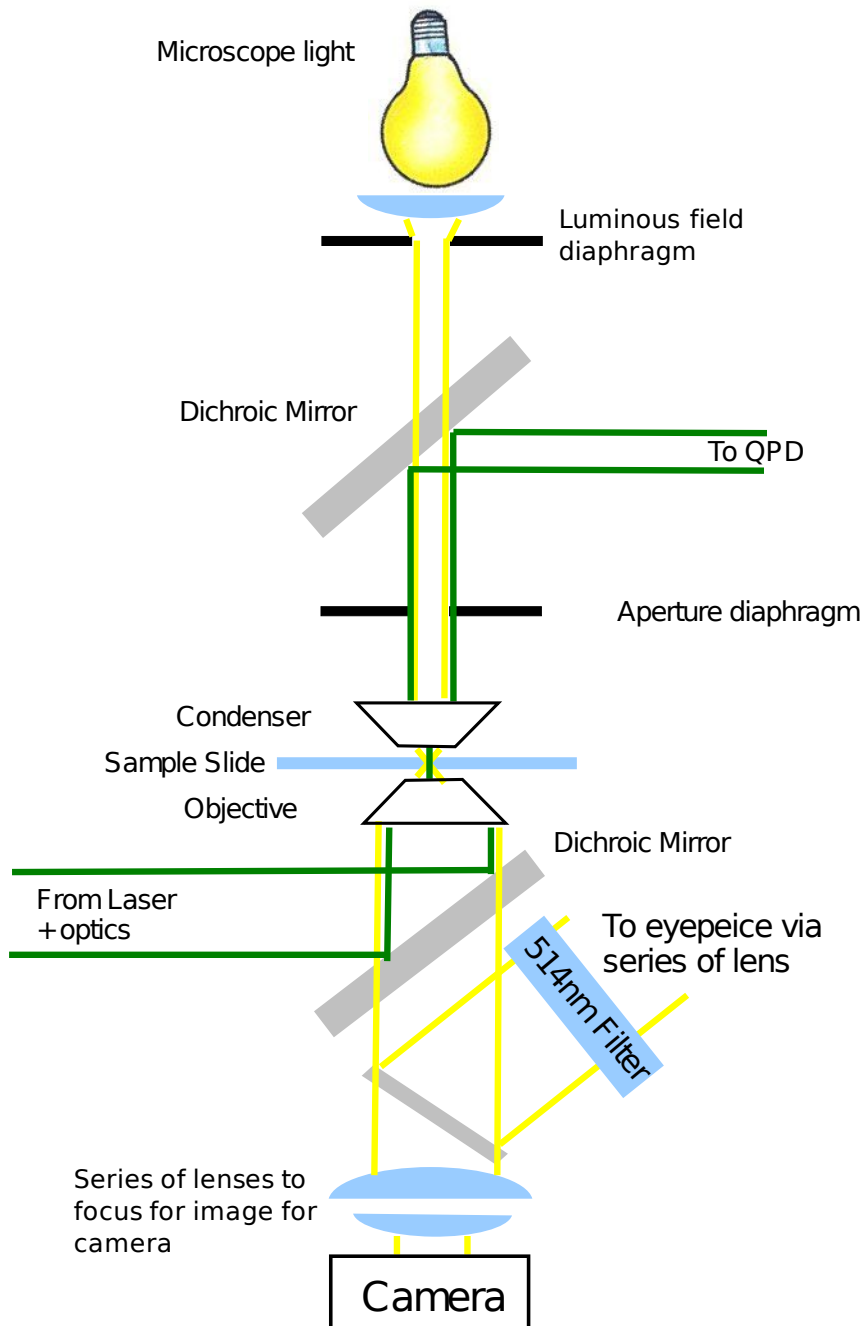


Figure 21: Microscope internals

Power Supply	$\pm 4.5V$ to $\pm 18V$
Maximum slew rate	$10V/\mu S$
Theoretical noise	$15nV/\sqrt{Hz}$
Bandwidth	257kHz

Table 7: Specifications for QPD

4.6 Data Acquisition Card

A National Instruments PCI-6120 DAQ card is used to capture data for these experiments. The technical specifications can be seen in Table 8. This DAQ was set up so that only the 3 channels from the QPD are used as inputs while the remaining channels have been terminated to ensure that there is no interference. When a second device (laser power meter) was connected we saw a shift in the signals, indicating that the signals were not properly isolated. Labview is used to acquire the data from the card allowing complete control of the settings for data capture.

4.7 Optical Alignment

The alignment of the optical tweezer should be done in a number of stages, starting with course alignment and moving to progressively finer alignment[8]:

4.7.1 Course alignment

1. **Table** The optical tweezer is mounted on a passive vibration isolation table (Melles Griot 070T1063). It is important that the table is flat

Number of analog input channels	4
Input resolution	16 bits
Maximum sampling rate	1 MS/s
Voltage Input range	$\pm 0.2V$ to $\pm 42V$
Range sensitivity	$6.1\mu V$ to $20.5mV$
Range Accuracy	$344\mu V$ to $87.47mV$
Anti Aliasing Filter Frequency	5 pole Bessel at 100kHz
Sample Buffer	64MS

Table 8: Technical parameters for DAQ[19]

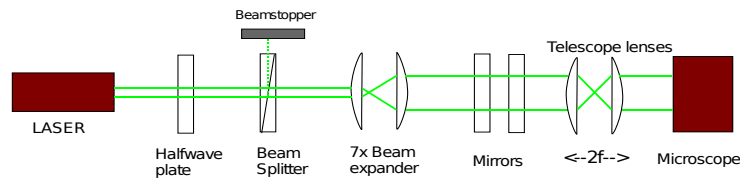


Figure 23: Optics that are adjusted during alignment

and level. This is done to make sure the beam is perpendicular to the optics and a horizontal surface for the microscope. In our case this required adjusting the air pressure of the feet, this is done by checking the surface against a highly accurate spirit level.

2. **Laser mirror** As discussed in section 4.2, the wavelength of light is determined by the angle of the laser's rear mirror. The mirror is adjusted so as to give the brightest spot, this corresponds to the wavelength (514.5 nm) that we will be working with.
3. **Laser Alignment** The laser is situated upon 4 feet that allow vertical adjustment, these are rotated to allow the beam to have a constant height over the whole length of the table. The laser is powered up at its lowest intensity setting and a reference point is marked on a screen, this will serve as a centre point for further alignment.
4. **Microscope** The microscope is placed in the desired position so that the beam can be steered into the epi-fluorescence hole. The objective is removed to make initial alignment easier.
5. **Half-wave plate and beam splitter** First the half-wave plate is placed in the beam, and centred in such a way that the beam remains on the mark. Additionally we try to make sure that the reflected beam hits the side of the laser's aperture so as not to cause any interference. The beam splitter is then added in the same way as the half-wave plate.
6. **Beam expander** The beam expander is placed in the beam path, here it is important to ensure that the beam enters and exits the centre of this device. The beam should be as large as possible but still converging slightly to ensure effective trapping. The projection of the beam is examined on a screen and ensure that the centre is still on the reference point and that no aberrations have been introduced.

7. **Mirror 1** This mirror reflects the beam vertically, and the only alignment that need doing is ensuring that it is vertical.
8. **Mirror 2** The second mirror is placed above the first and the beam is directed into the epi-fluorescence port. This mirror together with mirror 1 is adjusted until the beam travels vertically out of the microscope and passes through the centre of the objective hole.
9. **Lens 2** The second lens is placed in the beam path, it is aligned such that the beam continues to pass through the microscope vertically and through the centre of the objective hole. It should also be verified that the beam should come into focus somewhere behind the objective hole.
10. **Microscope Objective** The objective is re-attached to the microscope and a slide is placed over the lens, the two mirrors are adjusted so that the beam is centred.
11. **Lens 1** The first lens is inserted into the optical path at a distance (z position) of $2f$ ($2 \times$ focal length of the lens) from lens 2 and centred vertically and horizontally on the beam (x and y position), it is adjusted such that the beam hits the centre of the back aperture on the objective lens. Leaving the mirrors untouched the x and y position are adjusted until the beam emerges vertically and centred from the microscope.

4.7.2 Fine alignment

1. **Lens 1 adjustment** With the objective and condenser lens in place the beam emerging from the condenser is observed. This should be a uniform disc of light, if this is not the case adjust the x and y position of Lens 1 until it is.
2. **Centring** The microscope is focused on a glass slide, you should be able to see a bright spot, indicating the point that the beam reflects off the glass slide. The x and y position of lens 1 is adjusted so as to centre the beam on the glass slide. The z position is used to minimise the diameter of the spot on the slide.
3. **Focus** The microscope is focused on the lower water/glass interface of a sample cell containing silica beads, and the x and y position of lens 1 is adjusted such that the beam is centred in the field of view.
4. **QPD** The dichroic mirror is adjusted so that the beam is reflected to the centre of the QPD, if the beam is too large or too small a lens

should be placed in the path so as to fill the QPD. If the beam is less than 1 mm in diameter it will show a pronounced effect when on the borders of each quadrant, something that we wish to avoid.

4.7.3 Walking the beam

Once the optical path has been aligned we need to adjust the beam so that it is perfectly coaxial with the optical axis of the objective. The microscope is focused to the lower glass-water interface. On either side of the interface the beam spot will expand, if this spot isn't uniform it should be corrected using a procedure that is called "Walking the beam". The adjustment is made using Lens 1 and mirror 2.

Initially the beam is focused slightly above the cover slide, such that a set of concentric rings are visible. The spot is first moved away from the centre of field using either the mirror or the lens, and then brought back to the centre using the lens or the mirror (depending on what was used first). For example, the lens is adjusted so that the spot is moved to the upper right, the mirror is then adjusted to bring the spot to the centre of the field of view. If everything was done in the correct direction the spot should be more symmetric. If the wrong order was used the adjustments would have worsened the uniformity of the spot, when this happens adjustment should be done in the opposite direction. This same procedure is repeated for both the x and y axis. You should now have a symmetric spot, and can verify this by moving the particle above and below the water/air interface. You are now ready to trap a particle.

4.8 Piezo Electric Stage

The microscope stage has been fitted with a piezo electric sample stage and controller (PI-PS17.3CD and a PI E-710.p3D). Together these devices allow for precision control of any samples placed under the microscope. The controller accepts data from software running on the computer (in our case Nanocapture) and acts on this data so as to adjust the piezo stage to the required position. The stage has been fitted with three parallel capacitive sensors ensuring linearity and allows correction of errors by adjusting each of the 3 piezo actuators. Nanocapture has a tool to generate simple waveforms, "generate waveform", this allows for simple adjustments of both the amplitude and frequency of the stage. "Generate waveform" takes 3 parameters:

Amplitude The distance travelled from the zero point.

Offset Offset from the piezo's zero to our initial starting point.

Frequency Frequency of the movement.

Apply to X, Y or Z Which axis to this movement should be applied to, in our case this was generally the X.

Once the desired waveform has been entered, pressing “generate” will create the waveform, choosing “Run Continuously” will move the piezo stage in the desired motion.

5 Experimental Procedure

Using a trapped particle to investigate materials requires that we know the properties of the trap, one of the most important properties is the trap stiffness (κ). The trap stiffness is essentially a spring constant, giving a relationship between the displacement and force applied to the system, such that once we know the trap stiffness it is possible to calculate the forces involved. In theory this is simple but the Quadrant photo diode (QPD) measures a voltage in all the spacial dimensions (two horizontal X and Y and vertical Z). Each of the three voltage signals are linear with position, which for a basic calibration will not be an issue. If on the other hand we wish to avoid assuming the viscosity of water, the calibration method quickly becomes quite more complex.

5.1 Data capture — “optimising” sampling

When capturing the data, it is important to try and optimise the DAQ parameters, so that data can be processed easily.

Frequency A frequency which is an integer power of 2 should be chosen, this ensures that fast Fourier transform method can be used. Although in practise this results in faster data processing.

Active axes	3 (X, Y, Z)
Travel (μm)	100,100,20
Closed loop Resolution (nm)	1,1,0.1
Open Loop Resolution (nm)	0.3,0.3,0.1
Closed Loop Linearity (%)	0.03
Full-Range repeatability (nm)	$\pm 5, \pm 5, \pm 1$
Resonant Frequency, 500g load (Hz)	250

Table 9: Technical parameters for P-517.3CD sample stage[23]

Measurement time If calibration procedure involves moving the piezo stage with a sinusoidal motion with frequency f_d , we should ensure that $t_{msr}f_d = \text{integer}$. If this is not done there will be a signal leakage, resulting in a underestimation of the power at that frequency.

Drive frequency and amplitude This is the frequency that the piezo stage is driven at when we use a drive frequency for calibration. The drive frequency and amplitude should be chosen so that they do not excite any resonance in the system.

Figure 24 shows the effect of time and sampling rate on the peak in a power spectrum.

5.2 Data Processing

Before any analysis of raw data can be done, it needs to be processed correctly. During the project a C++ program was made to perform these tasks, see appendix A.2 for the program code. The following tasks are performed on the data set:

Data centring The data is read into memory and the X and Y data is averaged, this will provide us with the baseline for each data set. This average is then subtracted from the data, centring it around zero. This is necessary as it is very difficult to align the QPD such that the signal varies exactly around zero.

Data Normalisation This data is then normalised by dividing by the Z signal to remove noise from the laser or signal acquisition system.

Windowing As the signal never returns to exactly the same values as the starting point we need to use a windowing function to prevent signal leakage when doing a Fast Fourier transform. The windowing function takes the data and lets it slowly decay to zero at the ends of the data set. There are many different functions to choose from, each with there own characteristics. A couple of different windows were chosen and the resulting viscosity was compared to tabulated values. The best windows were the triangular, rectangular and Hann[12]. A triangular window resulted in a viscosity value which was not far off the tabulated value, but resulted in a number of side lobes. This improved by using a rectangular window, but resulted in a peak that wasn't as sharp. Choosing a Hann window gave a sharp peak and maximised the value of viscosity and the force constant. A detailed reference on windowing can be found in a paper by F. J. Harris[12]

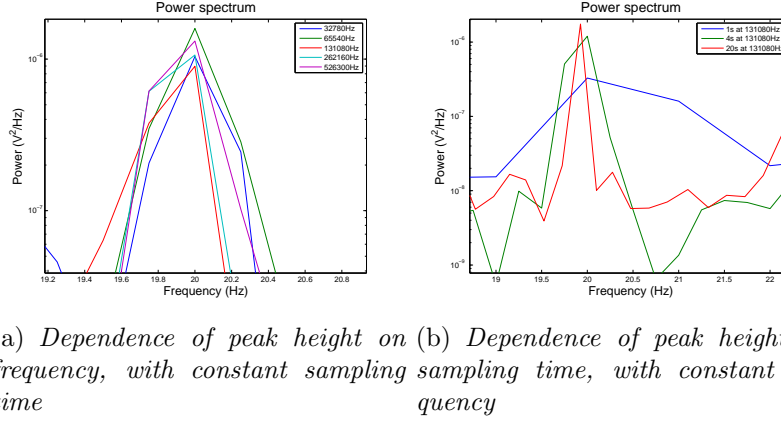


Figure 24: Dependence of peak height on acquisition parameters, it should be noted that the height is a function of the sampling time, see section 2.7.4

Converting data to power spectrum A fast Fourier transform is taken of the normalised data and multiplied by the conjugate of itself. The resulting power spectrum is then ready for analysis.

By taking the Fourier transform, the resolution and lowest frequency of the processed data is determined by the length of the data set, more specifically $\delta f = 1/T_{msr}$ where δf is the frequency resolution and T_{msr} is the measurement time. Similarly the lowest frequency is given by $f_{low} = 1/T_{msr}$. This means that a choice has to be made on the sampling rate and length of the raw data, ideally we would want to make infinitely long measurements and infinitely large sampling rate, thus getting a high resolution in the raw data and power spectrum. The high sampling rate is necessary in order to obtain accurate raw data and have a higher Nyquist frequency (see section 6.1.3). Due to the limitations of computational power and time at our disposal, most of the measurements were made with a sampling rate of 128kHz for a period of 1–4 seconds.

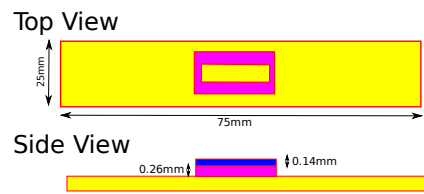
Binning These experiments often produced large amounts of data, even after the power spectrum has been made. To make further analysis easier, logarithmic binning is applied to the power spectrum. This procedure takes data blocks which get exponentially larger with larger frequency and averages them to a single data point. This has no effect on the shape or height of the power spectrum and will only result in data which is easier to work with. It should be noted, if we are driving the piezo stage to obtain a spike in the power spectrum the height of the raw spike should be recorded as averaging

with the plateau will lower the power by a considerable amount, resulting in an underestimation of both the force constant and viscosity.

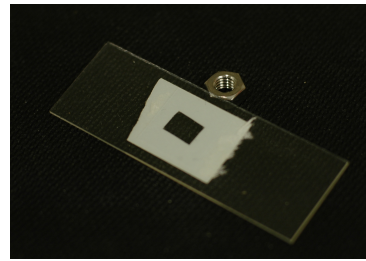
5.3 Cell preparation

For all the experiments discussed in this section, a sample cell needs to be made to hold the suspension of silica beads in water. The basic design involved a microscope slide with a small piece of tape with a well cut out of it and covered with a cover slip (Figure 25).

Once the well has been cut in the tape and mounted on the microscope slide, the suspension of silica beads in water is placed in the well and covered with the cover slip. It is important to ensure that there is no trapped air in the slide as this could allow the fluid to move around. This is achieved by lowering the cover slip onto a sample cell which is slightly overfilled, thereby expelling the excess and completely filling the cell. An alternative method is to create a cell with two strips of tape, one on either side and sealing the cell with vacuum grease.



(a) Schematic view



(b) Photo of sample cell

Figure 25: Sample cell used for trapping of particles. Yellow represents the microscope slide, purple represents the tape and blue represents the cover slip. For clarity the top view shows the cell without the cover slip.

5.4 Verifying program code

5.4.1 Frequency accuracy

To verify the program code (section A.2) a signal generator was attached to the DAQ inputs and a 20Hz sine signal was measured (Figure 26). Additionally a couple of waveforms were generated using Matlab (Figures 27 and 28). Comparing the code to the built-in functions in Matlab and Labview, resulting in almost identical power spectrum with only a small shift in height. This is probably due to the different FFT implementations.

5.4.2 Amplitude accuracy

There was some concern on whether the power estimate for a Fourier transform is accurate. Starting with a Gaussian function $y(t) = e^{-\lambda t^2}$, taking the Fourier transform results in another Gaussian with the form:

$$y(f) = \sqrt{\frac{\pi}{\lambda}} e^{-\frac{(\pi f)^2}{\lambda}}. \quad (188)$$

Therefore if we generate a data set for the initial Gaussian function and take a fast Fourier transform on this data set we should obtain the expression in Eq. (188). A data set was made (Figure 29) using the code listed in section A.3 for a sampling rate of 2048Hz and for 400 seconds. From Figure 29(b) it is clear that the prefactor is not correct. This can easily be explained by noting that the relation between the discrete Fourier transform and the continuous transform sampled at a rate F_S is given by Eq. (189).

$$H(f_n) \approx \frac{H_n}{F_S} \cdot [24] \quad (189)$$

As the raw data set only represents half of a Gaussian function, we will need to fit the function shown in Eq. (190)

$$y(f) = \frac{1}{2} \sqrt{\frac{\pi}{\lambda}} e^{-x^2/b} \quad (190)$$

Figure 30 shows that the amplitude information has been conserved through the Fourier transform. This implies that the FFT algorithm uses same definition (Eq. (23)) as the form used to derive the power spectral density. Similar the second fitting parameter b is scaled due to the sampling of data, so far i have been unable to find a reference for this scaling but by changing the sampling rate and time i came to Eq. (191)

$$b = \frac{t^2 \lambda}{F_s^2 \pi^2}. \quad (191)$$

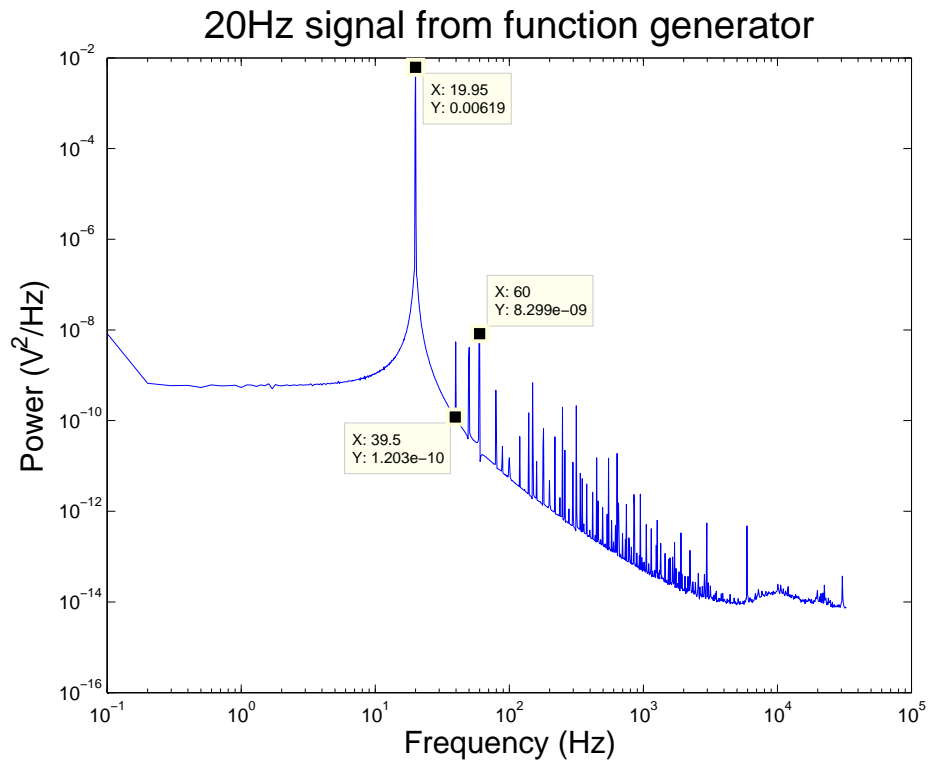


Figure 26: Plot showing a 20Hz sine wave acquired from a signal generator through the DAQ. The signal is clearly visible at 19.95Hz along with higher harmonics and (39.5 and 60Hz) and some other undefined noise (e.g. 50Hz mains). The deviation from 20Hz is most likely due to the accuracy of the signal generator.

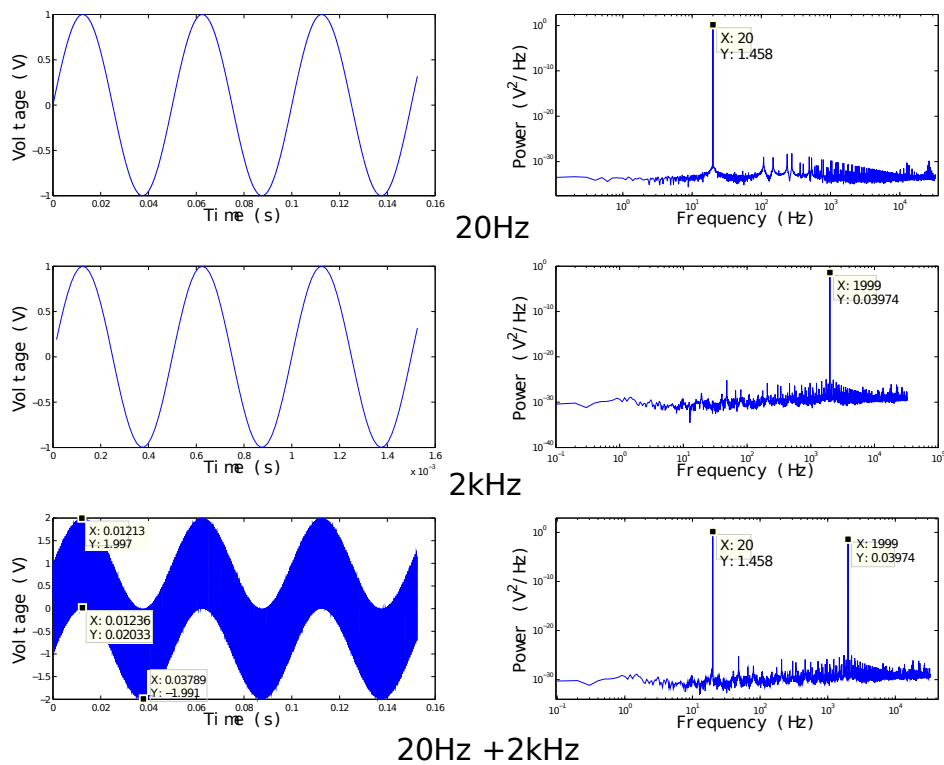


Figure 27: Plots showing Matlab generated waveforms which were converted to a power spectrum. Figures represent sine wave at 20Hz, Sine Wave at 2kHz and a Sine wave at 20Hz with a 2kHz added on top. We can clearly see that the power spectrum is accurate, picking out the correct frequencies.

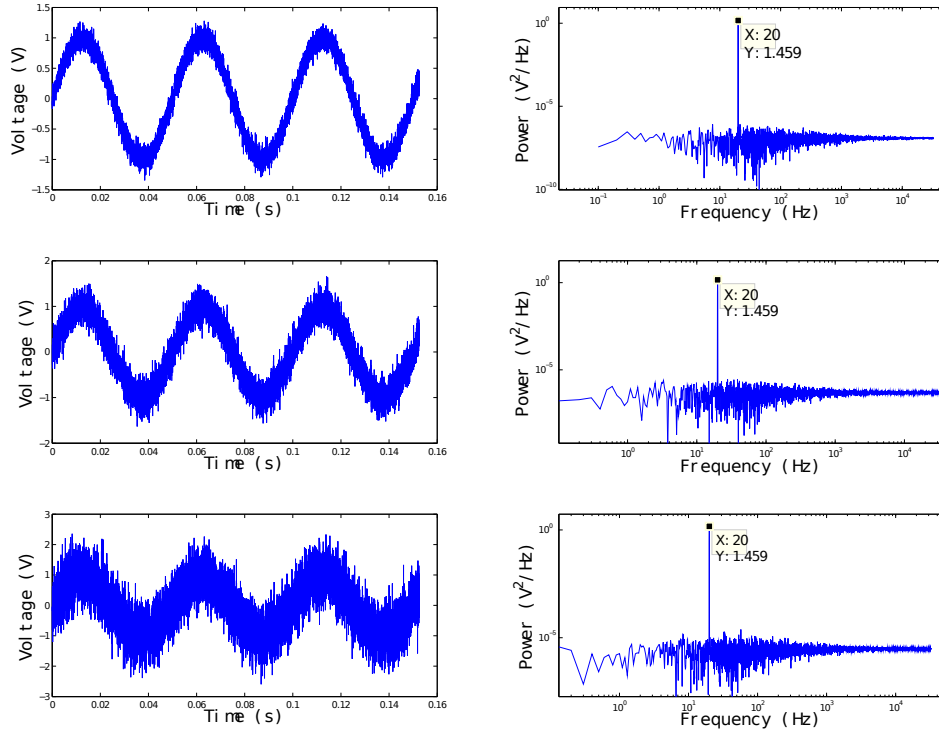


Figure 28: Plots showing Matlab generated waveforms which were converted to a power spectrum. Figures represent, Sine waves at 20Hz with a varying amount of Gaussian distributed white noise applied to them. The amount of noise doubles between each of the plots. The signal at 20Hz is still clearly visible and retains the same power as in Figure 27

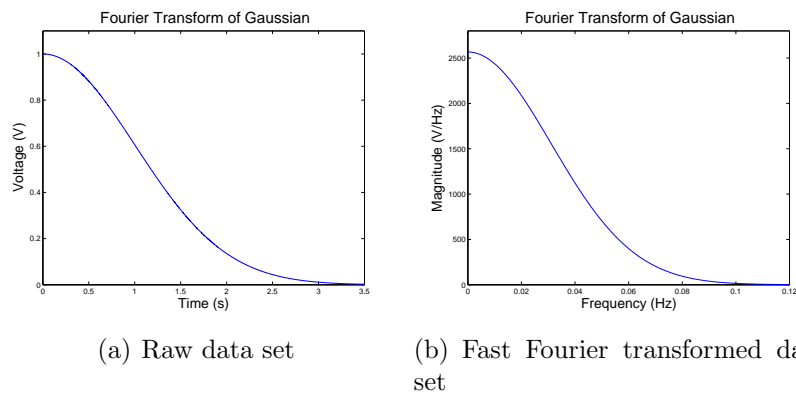


Figure 29: Raw and Fast Fourier transformed Gaussian signal. Parameters, $\lambda = 0.5$, Sampling frequency $F_S = 2048\text{Hz}$ and Sampling time $t = 400\text{s}$.

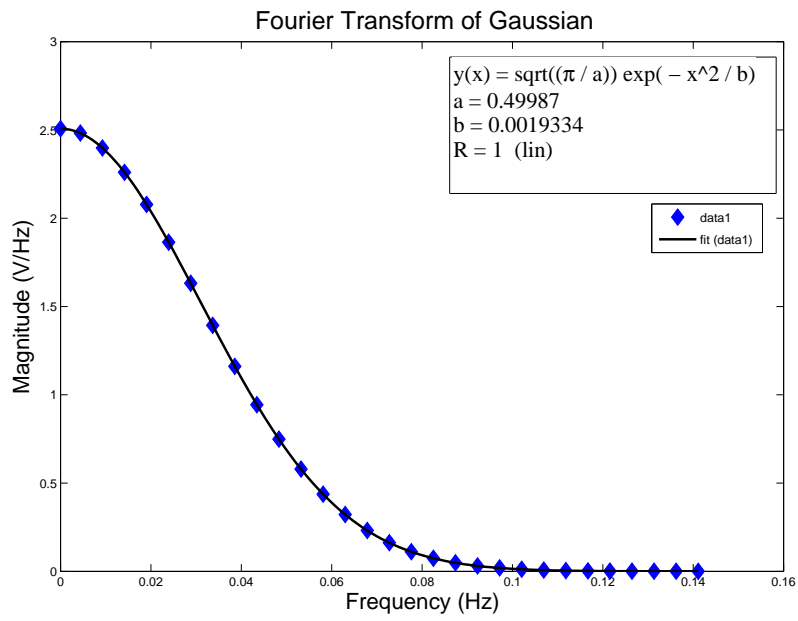


Figure 30: Fast Fourier transform of Gaussian function, normalised according to Eq. (189)

Initial Gaussian parameters, $t = 400$, $F_s = 2048$ and $\lambda = 0.5$. Fitting a Gaussian with Eq. (190) gives, $\lambda = 0.5$ and $b = 0.0194$.

Using this to check the parameter in Figure 30, gives a value for λ of 0.5, which is consistent with the input value.

The final form of the fitting function becomes:

$$y(f) = \frac{1}{2} \sqrt{\frac{\pi}{\lambda}} e^{\frac{-F_s^2 \pi^2 x^2}{t^2 \lambda}} \quad (192)$$

From these results we can verify that the power spectral density generated using the code listed in appendix A.2 and A.3 accurately converts the raw data. This test also verified that the FFT code transforms the raw data using the same Fourier transform (Eq. (23)) used to calculate the many equations in this project.

5.5 Power spectrum analysis

One of the most widely used methods of calibrating optical tweezers is done by using the power spectrum that arises due to the Brownian motion of the particle in the trap. The theory for this is given in section 2.4. A typical power spectrum will have a Lorentzian shape, and therefore has a plateau at lower frequencies due to the confinement of the particle. At higher frequencies the power spectrum will have a slope of -2 , which is a characteristic of free diffusion as the particle will not “feel” the trap. The split between the two regions in the power spectra will occur at a characteristic frequency called the corner frequency. It is only possible to calibrate the trap from the power spectrum resulting from thermal motion if the bead is of known size and the surrounding fluid viscosity is known.

5.6 Power spectrum for a trapped $1.54\mu\text{m}$ particle

The experimental data presented here was taken from two 4 second measurements, which were independently Fourier transformed and averaged. The fit was obtained using Matlab’s least squares fitting algorithm.

Figure 31 shows a fit of Eq. (193) to the data.

$$S(f) = \frac{D}{(2\pi^2)(f_c^2 + f^2)} \quad (193)$$

By checking the literature[14, 11] the force constant found ($3.82 \cdot 10^{-2}$ pN/nm) with this data set is of comparable size and a plausible result.

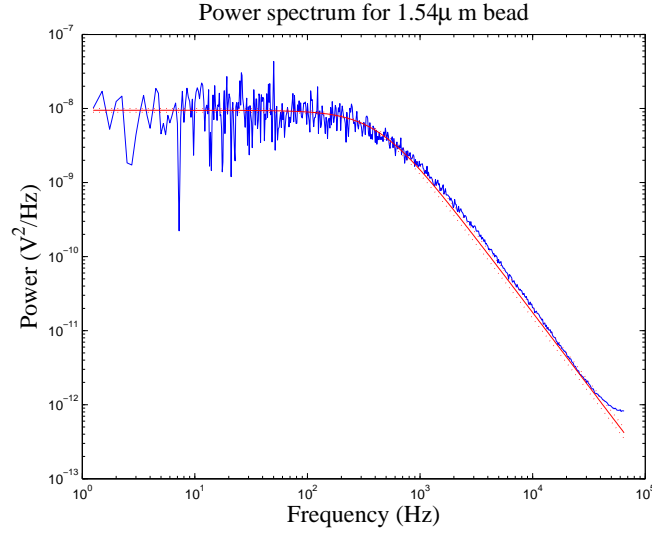


Figure 31: Lorentzian fit (red) to an experimentally obtained data set (blue)

Measured and tabulated data

Temperature	21.57°C
Sampling Frequency	131080 Hz
Total Sampling Time	8s
Laser power	100.0mW
Particle Size	1.54μm
Tabulated water viscosity	9.699 Pa s

Calculated Data

Corner Frequency	431.82Hz
Diffusion constant in volts (D_V)	$1.74 \cdot 10^{-2} \text{ V}^2/\text{s}$
Power at f=0 (S_0)	$9.45 \cdot 10^{-9} \text{ V}^2/\text{Hz}$
Force constant (κ)	$3.82 \cdot 10^{-2} \text{ pN/nm}$
Voltage Conversion (β)	$1.57 \cdot 10^{-6} \text{ m/V}$
Diffusion constant (D)	$2.88 \cdot 10^{-13} \text{ m}^2/\text{s}$

Table 10: Experimental data and Fit parameters for trapped particle without an external drive.

5.7 Using sinusoidal stage movement for calibration

Ideally we wanted to be able to calibrate the optical tweezer without knowing the viscosity of the fluid that the bead was trapped in. One of the quickest experimental methods of doing this is outlined in Section 2.7.1. This method involves moving the glass slide in a sinusoidal motion and recording the spectra. Once the spectra has been obtained it is fit with the standard equation for the power spectrum, Eq. (30). We can then use Eq. (124) to determine the experimental power from the height of the peak at our driving frequency (f_d). Using Eq. (122) the Voltage to distance conversion factor (β) is found. From the standard fit as outlined in section 5.5 we have a value for the corner frequency and diffusion coefficient. Then it is just a matter of plugging the values into Eq. (127), Eq. (130) and Eq. (133) to find the values for the force constant (κ), the drag coefficient (γ) and the viscosity of surrounding fluid (η).

The results obtained using this method varied from just a couple of percent to close to 100% for the viscosity of water. The data set shown above is one of the better data sets, with an almost perfect (96%) result. Some of the more typical data sets are in section 5.9.

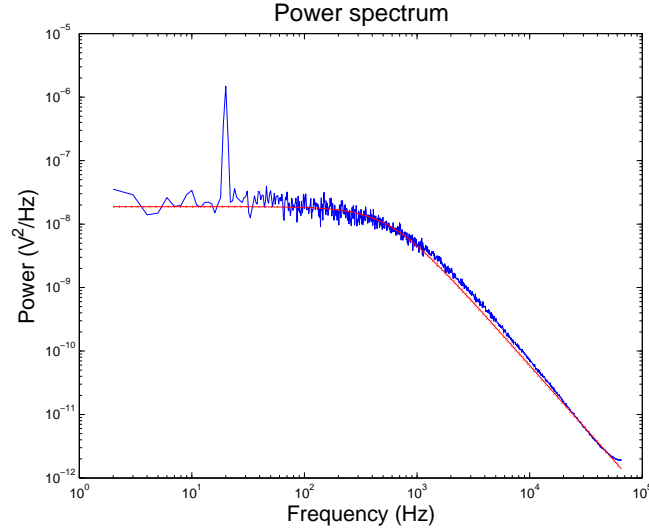


Figure 32: Power spectrum for sinusoidal stage movement of a $1.54\mu\text{m}$ particle trapped in water. The peak is at 20Hz, the drive frequency for this experiment

Tabulated data

Particle Size	$1.54\mu\text{m}$
Temperature	294.5K
Viscosity of Water	$9.70 \cdot 10^{-4} \text{ Pa s}$

Experimental data

Total Sampling time	20s
Drive frequency	20Hz
Drive amplitude	208nm
Peak Height	$7.24 \cdot 10^{-6} \text{ V}^2/\text{Hz}$
Corner frequency (f_c)	564 Hz
Diffusion Coefficient (D)	$3.08 \cdot 10^{-13} \text{ m}^2/\text{s}$
Power at $f=0$ (S_0)	$3.76 \cdot 10^{-8} \text{ V}^2/\text{Hz}$
Voltage conversion factor (β)	$1.60 \cdot 10^{-6} \text{ m/V}$
Force constant (κ)	$4.82 \cdot 10^{-2} \text{ pN/nm}$
Viscosity (η) (% of tabulated)	$9.39 \cdot 10^{-4} \text{ Pa s}$ (96.8%)

Table 11: Table showing experimental data and results for sinusoidal stage movement, from an average of 20 one second data sets.

5.8 Measuring trap parameters by measuring particle position in a flow

As the sinusoidal stage movement did not give reliable results, an attempt was made by moving the stage at a steady velocity while keeping the particle fixed. The voltage for the particle deviation from the centre of the cell was measured.

5.8.1 Theory for calculating experimental parameters

Using the Einstein relation, relating the mobility to the diffusion constant, the definition for the corner frequency, Hookes law and stokes drag we can find a set of three equations and solve for each of the three unknowns.

Finding an expression for distance voltage conversion factor

Using Hookes law,

$$\mathbf{F} = -\kappa \mathbf{x}, \quad (194)$$

where κ is the force constant and \mathbf{x} is the displacement. And an equation for stokes drag

$$\mathbf{F} = -\gamma \mathbf{u} = 6\pi\eta r \mathbf{u}, \quad (195)$$

where γ is the drag coefficient, \mathbf{u} is the velocity, η the viscosity and r is the radius of the particle. Combining these equations and expressing them in terms of γ ,

$$\mathbf{F} = -\kappa \mathbf{x} = -\gamma \mathbf{u} \quad (196)$$

$$\gamma = \frac{\kappa \mathbf{x}}{\mathbf{u}} \quad (197)$$

$$\gamma = \frac{\kappa \beta V}{\mathbf{u}}. \quad (198)$$

Where in Eq. (198) we used an expression for the position (\mathbf{x}) in terms of voltage (V). $\mathbf{x} = \beta V$. We also have an expression for the corner frequency,

$$f_c = \frac{\kappa}{2\pi\gamma}. \quad (199)$$

Substituting in the expression for the drag coefficient gives:

$$f_c = \frac{\kappa}{2\pi} \frac{\mathbf{u}}{\kappa \beta V}. \quad (200)$$

Rearranging to get an expression

$$V = \mathbf{u} \frac{1}{f_c 2\pi \beta}. \quad (201)$$

Where the voltage, velocity and corner frequency are terms that we are able to find and can obtain a value for β .

Finding the viscosity The Einstein relation is given by

$$D = \beta^2 D_V = \frac{k_B T}{\gamma}. \quad (202)$$

Where D is the diffusion constant, β is the conversion factor found in the previous section, $k_B T$ the Boltzmann constant time. We have already found a value for the diffusion constant (in volts, D_V) and from the previous section have a value for the conversion factor, and can therefore obtain an expression for the viscosity.

$$D = \beta^2 D_V = \frac{k_B T}{6\pi\eta r} \quad (203)$$

$$\eta = \frac{k_B T}{6\pi r \beta^2 D_V} \quad (204)$$

Finding the force constant As we now have an expression for the conversion factor (β) and the viscosity we can also determine the force constant (κ).

$$f_c = \frac{\kappa}{2\pi\gamma} \quad (205)$$

$$\kappa = \frac{2\pi\gamma}{f_c} \quad (206)$$

5.8.2 Experimental method

Obtaining the data is done in two stages, the first stage involves capturing a regular power spectrum as was done in section 5.5 while the second stage involves moving the piezo stage with particular velocities and noting the position (voltage). This then provides all the information necessary to calibrate the tweezers.

Data set for standard calibration Using a $1.54\mu\text{m}$ trapped in a 100mW laser beam, a power spectrum was recorded. This was analysed in the usual way and results presented in Figure 33 and Table 12.

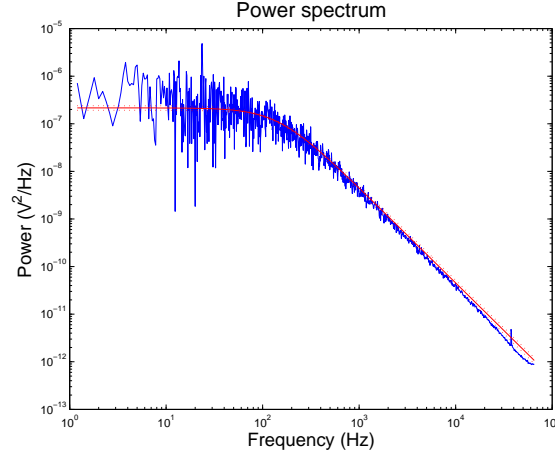


Figure 33: Power spectrum for Flow calibration of a $1.54\mu\text{m}$ particle trapped by a 100mW laser beam.

Corner frequency (f_c)	98.4Hz
Diffusion coefficient (D_V)	$6.17 \cdot 10^{-2} \text{ V}^2/\text{Hz}$
Temperature (K)	296.6K
Measurement time	5s
Sampling rate	128kHz

Table 12: Experimental results for Lorentzian fit.

Data set for flow The same particle is then used to measure the position in volts with different flow rate. The piezo stage is set up to make one triangle wave moving $50\mu\text{m}$ each way. The frequencies were adjusted such that the water surrounding each particle will have increasingly large velocities relative to the particle. For each frequency we are presented with a plot similar to that shown in Figure 34.

Measurements were made for a number of different frequencies, the average shift for each was calculated and presented in Table 13

Figures 35 and 36 show a plot of the voltage for different stage velocities. From these figures there is clearly a problem with the last data set with a velocity of 1.125m/s as both of these are not on the linear curve. This is believed to be because the particle is not trapped properly as the flow of the water is almost enough to dislodge it.

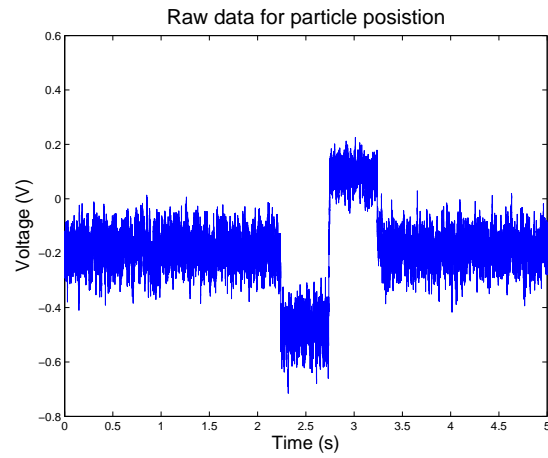


Figure 34: Raw data of a trapped $1.54\mu\text{m}$ particle with water flowing past.

Analysis of data Using the equations listed in section 5.8.1 we find the results listed in Table 14. Unfortunately the conversion factor seems to be an order of magnitude larger than what it should be, this has severe repercussions for the calculation of the remaining results. Due to lack of time I was unable to repeat these experiments and they will have to be verified at a later date.

Frequency (Hz)	Velocity (m/s)	Negative position (V)	Positive position (V)
0.50	$5.00 \cdot 10^{-5}$	-0.1340	0.1340
0.75	$7.50 \cdot 10^{-5}$	-0.2080	0.2010
1.00	$1.00 \cdot 10^{-4}$	-0.2903	0.2710
1.25	$1.25 \cdot 10^{-4}$	-0.3490	0.3370
1.50	$1.50 \cdot 10^{-4}$	-0.4380	0.4040
2.00	$2.00 \cdot 10^{-4}$	-0.5950	0.6120
2.25	$2.25 \cdot 10^{-4}$	-0.2400	0.3370

Table 13: Experimental results for different flow rates. The negative displacement is the displacement of the particle in the backward direction of the piezo stage and the positive is the displacement in the forward direction. The total piezo stage amplitude was $50\mu\text{m}$ thus a total movement of $100\mu\text{m}$ for each of the frequencies.

Parameter	Negative flow	Positive flow	Units
Slope of fit	-3083.80	3126.6	Vs/m
Conversion factor (β)	$5.25 \cdot 10^{-7}$	$5.17 \cdot 10^{-7}$	m/V
Conversion factor from PSD and tabulated values	$2.55 \cdot 10^{-6}$		m/V
Viscosity (η)	$1.7 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$	Pa s
Tabulated viscosity	$9.6 \cdot 10^{-4}$		Pa s
Force constant (κ)	$1.57 \cdot 10^{-4}$	$1.570 \cdot 10^{-4}$	pN/nm
Diffusion Coefficient (D)	$1.7 \cdot 10^{-14}$	$1.65 \cdot 10^{-14}$	m^2/s

Table 14: Experimental results for flow method. Tabulated value for β were obtained by fitting the Lorentzian and using the tabulated values for water viscosity

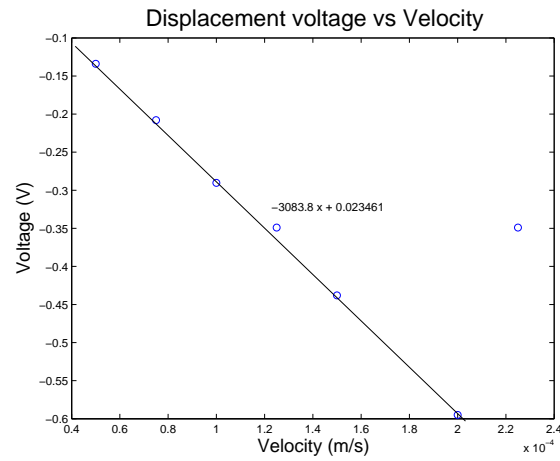


Figure 35: Linear fit to voltage vs position for flow into the negative direction

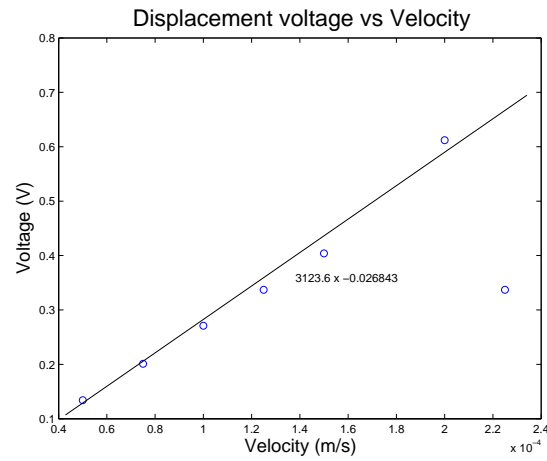


Figure 36: Linear fit to voltage vs position for flow into the positive direction.

5.9 Experimental data for sinusoidal stage movement

The following three figures (Figure 37–39) are representative of all the experiments performed using the optical tweezers.

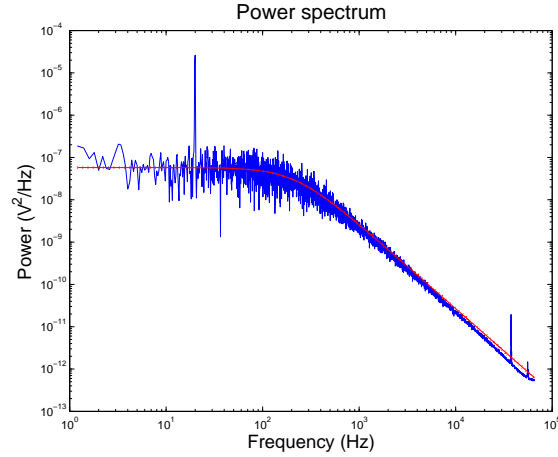


Figure 37: Data set for a $1.54\mu\text{m}$ particle, trapped with 20mW laser sampled at 128kHz for 5 seconds. Resulting trap force (κ) is $1.02\cdot 10^{-2}\text{pm/nm}$, and a viscosity (η) of $5.19\cdot 10^{-4}\text{Pa s}$ (54% of tabulated).

From these results we can clearly see that there is a problem with the spectra obtained as the viscosity for water was at 50% of the tabulated values.

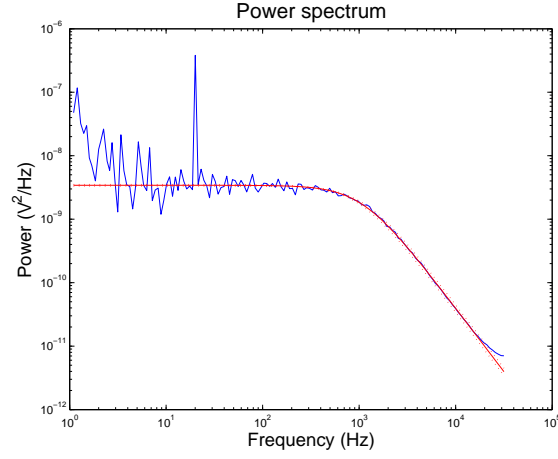


Figure 38: Data set for a $1.54\mu m$ particle, trapped with 100mW laser sampled at 64kHz for 10 seconds. Resulting trap force (κ) is $5.01 \cdot 10^{-2} pm/nm$, and a viscosity (η) of $5.09 \cdot 10^{-4} Pa s$ (53% of tabulated).

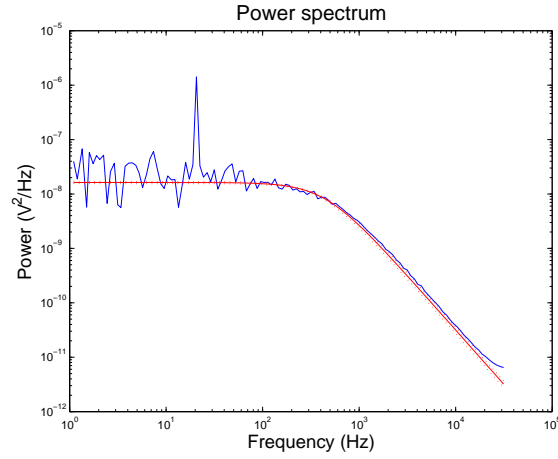


Figure 39: Data set for a $1.54\mu m$ particle, trapped with 100mW laser sampled at 64kHz for 10 seconds. Resulting trap force (κ) is $2.18 \cdot 10^{-2} pm/nm$, and a viscosity (η) of $5.03 \cdot 10^{-4} Pa s$ (52% of tabulated).

6 Characterisation of noise

As there seemed to be a large problem with the amount of noise in the power spectrum an attempt was made to characterise these. The noise varied from large areas raising the plateau to sharp peaks at well defined frequencies.

6.1 DAQ Card

As all data is measured through the DAQ, initial suspicion was on this instrument. There were a number of possible sources of problems like a voltage drop at certain frequencies somewhere between the QPD and DAQ, noise from the surroundings or the DAQ card itself. To investigate this, a number of experiments were performed.

6.1.1 Using a different DAQ

To rule out problems with the DAQ and eliminate some of the ground sources, a National instruments “DAQCard-6024E” was connected to a battery operated laptop. This shortened the length of cable that the signal has to travel through, from 6m to approximately 1m. As the laptop could be placed right next to the QPD, a minimum length of cable was required to make measurements. A power spectrum for a particle was made using both desktop and laptop DAQs, the resulting power spectrum is shown in Figure 40 It is clear

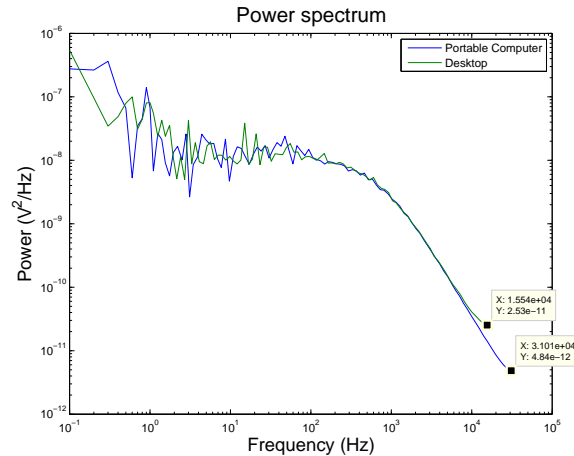


Figure 40: Laptop and Desktop computer captured data of a $1.54\mu\text{m}$ bead.

from these plots that there is no significant difference between the two cards

with the exception of the final frequency which is somewhat lower due to lower sampling rate. The desktop shows a small amount of additional noise at low frequencies, but this is not relevant as at frequencies of around 1 Hz surrounding conditions play an important role are hard to control.

6.1.2 DAQ linearity

To ensure that the DAQ measured voltages linearly a signal generator was connected to the DAQ inputs and a triangular wave was measured for approximately 1.6 V peak to peak and 0.08V peak to peak, see Figure 41. This

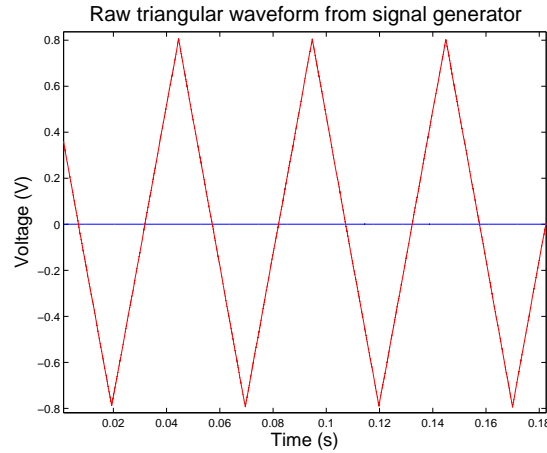


Figure 41: Triangle wave from signal generator, 1.6V peak to peak

resulted in a triangle waveform with perfectly straight lines, and concluded that the DAQ does in fact make linear measurements.

6.1.3 Aliasing due to sampling frequency and filters

In all measurements where data is sampled we are limited by the Nyquist frequency, in order to measure a frequency accurately we should have a sampling rate twice as high. For the power spectral density analysis this simply implies that we need to choose a frequency which is large enough to be able to accurately fit the Lorentzian. Unfortunately close to the Nyquist frequency the data will level off slightly. The reason for this is that a high frequency signal (f) can create a false image at $f_{false} = nF_{sample} - f$, where n is an integer, F_{sample} is the sampling frequency and f is the new lower false frequency. This effect is similar to seeing a wheel go backwards in old movies.

To completely avoid aliasing the sampling rate would have to be at least 10 times higher than the frequency you wish to measure.[20]

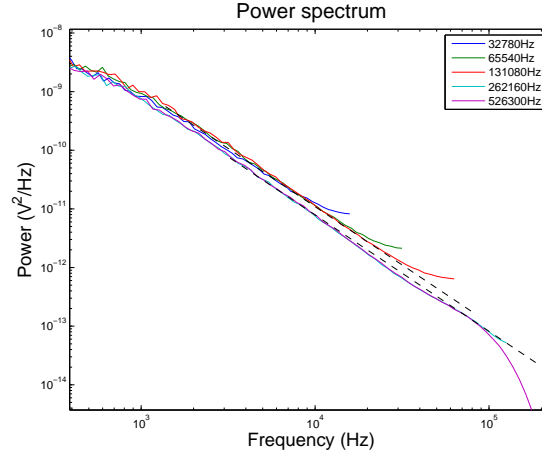


Figure 42: Lower part of a power spectrum for a $1.54\mu\text{m}$ bead trapped by a 100mW laser.

These two effects can be clearly seen in Figure 42. For the higher frequencies we can clearly see the upward curve deviating from the ideal curve (black dashed lines). In addition we can see the 100kHz filter in the DAQ kick in by observing the sharp drop in the 256kHz and 512kHz curves. Neither the sampling frequency or aliasing had any effect on the outcome of measurements for the force constant and viscosity.

6.1.4 Frequency matching with piezo stage

As the internal clocks on the piezo stage and computer ran at different speeds, there was a chance that the leakage in power was due to a mismatch between the sampling frequency and stage frequency. A number of measurements were made using different stage frequencies. These data are presented in Figure 43.

The largest power registered was at 16.0256Hz which is the closest to the drive frequency of 16Hz (where the sampling rate is an integer multiple of the drive frequency). Taking the three neighbouring frequencies on each curve, and summing them up can give an estimate for the total power under the assumption that we have a leakage in the power spectrum. Summing the peak at 16.1290Hz and comparing to the peak at 16.0256Hz indicates a slight increase in power ($3.01 \cdot 10^{-6} \text{ V}^2/\text{Hz}$ versus $2.63 \cdot 10^{-6} \text{ V}^2/\text{Hz}$). Further

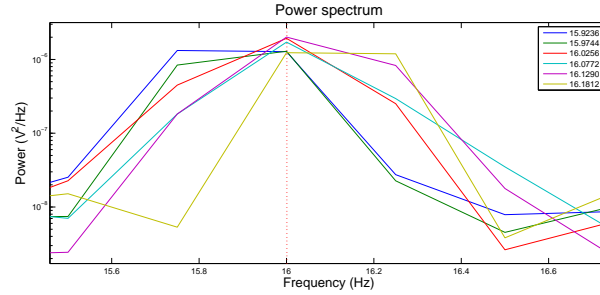


Figure 43: Blow-up of spike in power spectra for a $1.54\mu\text{m}$ bead trapped by a 100mW laser. The drive frequency for the piezo stage was 16Hz (marked with red dotted line) and sampling frequency was at 128kHz

investigation at 20Hz drive frequency showed that this was not consistent throughout the remaining data sets.

6.2 Quadrant photo diode

6.2.1 QPD Linearity

In order to access the linearity of the QPD a silica bead was fixed onto the surface of the cover slip. This was done by using an excess of salt in the suspension. The laser power was lowered and the piezo stage was programmed to make a sinusoidal wave.

Figure 44 shows the recorded sine wave, although it does not return to the same voltage after each cycle. This could be due to the fact that the particle is not properly fixed and is able to move at the turning point.

6.2.2 Power supply

As the QPD seemed to behave correctly with respect to linearity, measurements were made using a battery as the supply voltage instead of the standard mains power supply.

Both signals in Figure 45 are relatively similar in shape, the mains signal has a slightly higher noise floor, as would be expected. There was a very slight improvement on the low frequency but overall no significant difference. Figure 46 shows a similar experiment where “no signal” and a LED light was used to create a 16Hz . This shows us that the noise from the dark QPD only has a real effect at the lower and upper frequencies that were measured. It should be noted that light coming from the laser is going to be more intense and should show a larger difference in magnitude.

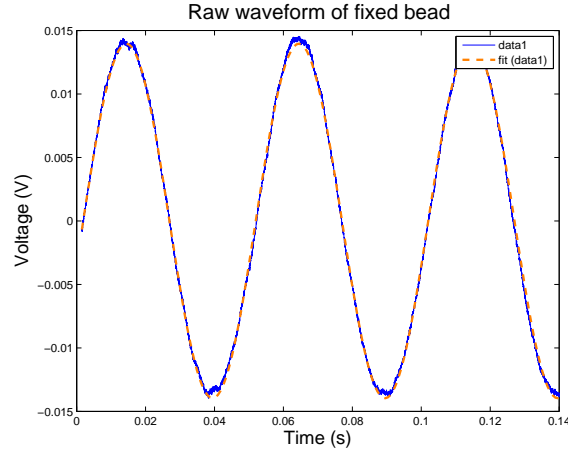


Figure 44: $1.54\mu\text{m}$ bead fixed to cover slip, laser of 2.5mW shining through the particle. The particle was moved a distance of 408nm peak to peak at 16Hz .

6.2.3 Signal level

Through experience from previous experiments with National Instruments DAQ cards there was some concern on the ability of the cards to measure low voltages. In order to verify if this was the case a signal amplifier was acquired and placed between the QPD and the DAQ. The power of the peaks in Figure 47 remain unchanged although the corner frequency has increased resulting in a better estimate for the force constant and viscosity. Unfortunately this was not a repeatable experiment and the resulting viscosity's varied from 95% to 50% of the tabulated values.

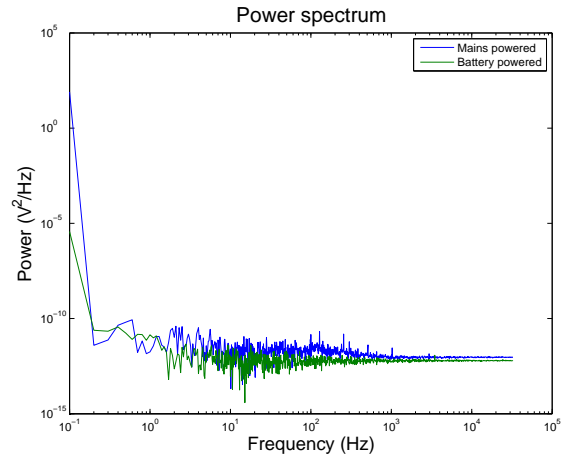


Figure 45: Power spectrum for dark signal on covered QPD.

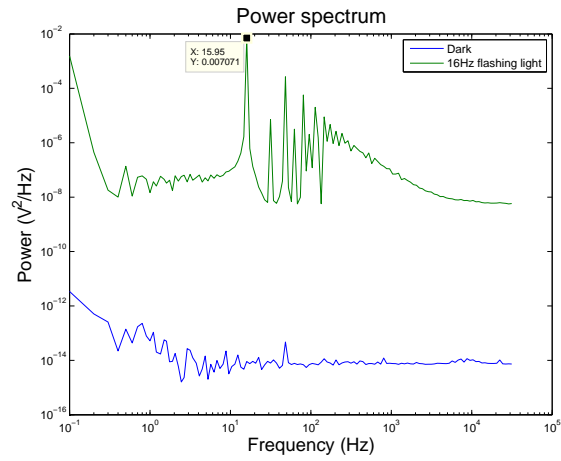


Figure 46: Power spectrum for a dark signal and a 16Hz flashing LED

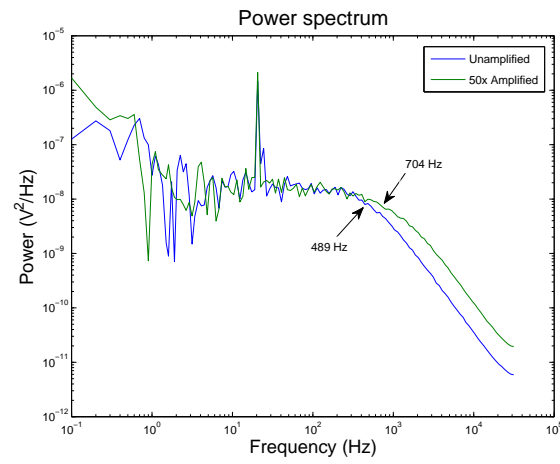


Figure 47: Power spectrum for two different trapped $1.54\mu\text{m}$ beads, trapped using a 100mW laser. The two frequencies indicate the corner frequency f_c

6.3 Laser problems

6.3.1 Signal variations

The Z channel on the QPD not only measures variations in height of the particle but also measures any fluctuations in the laser beam intensity. In order to be able to distinguish between the two, a photo diode was purchased and attached to the fourth channel on the DAQ. A glass cover slip was used to reflect a very small percentage of the beam onto the diode. This signal was recorded in parallel for the X, Y and Z axis's. Dividing the data sets for X, Y and Z by this new signal would give a normalised signal which should be less dependent on laser intensity fluctuations. Figure 48 shows the power

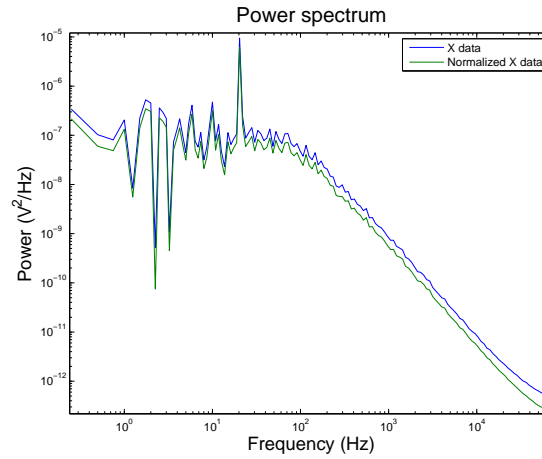


Figure 48: A 1.54 μ m particle trapped using a 100mW beam, with the piezo state driving a sine wave at 20Hz. The normalised signal is shifted to a slightly lower power but has an identical shape to the unnormalised

spectrum for a 1.54 μ bead trapped by a 100mW laser beam. The plot shows two signals, the X signal and the signal normalised using the other photo diode. Both signals have also been normalised using the Z axis of the photo diode. The difference is very small, amounting only to a couple of percent, which could easily be due to small variations of other external parameters.

6.3.2 Low frequency noise

As there was a substantial amount of low frequency noise the laser was swapped out with a red laser we had available. This laser is substantially weaker, around 20mW, and is just enough power to trap a particle. The

noise was reduced by an insignificant amount and didn't show any marked improvements in viscosity calculations.

6.4 Problems associated with the sample cell

6.4.1 Changes in salt concentration

In section 3.2 we saw that adding a low concentration of a counter ion like sodium chloride can result in an improvement in the estimate for viscosity. It is believed this is due to the sodium ions increasing the shielding with the result that there are fewer water molecules attached to each bead, effectively decreasing its radius. A salt concentration similar to section 3.2 was added to a sample cell and measurements were made and compared with an unsalted solution. In Figure 49 it is clear that there is no significant difference in

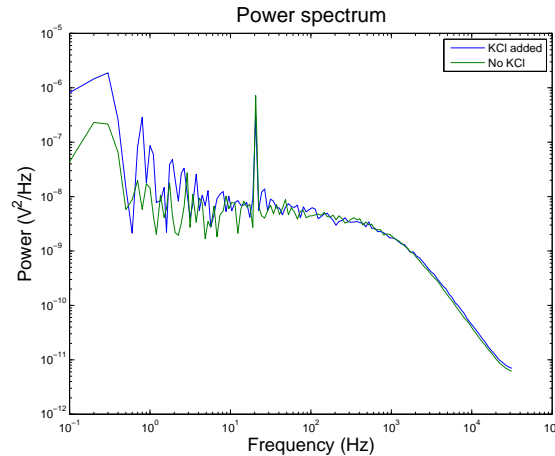


Figure 49: Two power spectra of a $1.54\mu\text{m}$ bead trapped by a 100mW laser driven using the piezo stage at 20Hz . The spectra are essentially identical with the exception of low frequency noise, probably due to external sources.

the spectra. Although there is a slight increase in low frequency noise which doesn't affect the data fit. The two spectra only differ by a couple of percent, which is well within the experimental error.

6.5 External sources of noise

6.5.1 Vibration

When a particle is trapped it is essentially floating in water, this implies that any vibrations will be transmitted to the bead, resulting in low frequency noise. Figure 50 shows an example of this. During the project it quickly

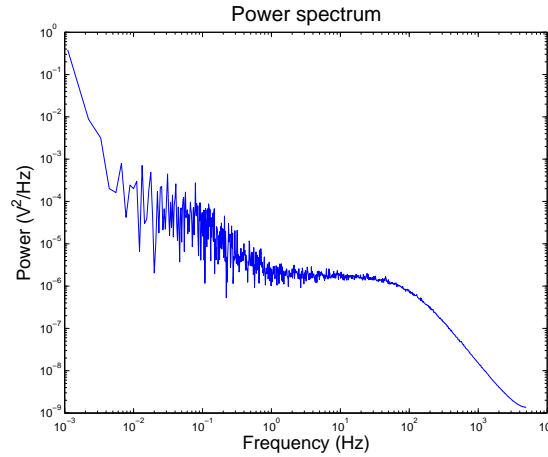


Figure 50: Example of low frequency noise, the power spectrum shows a $1.54\mu\text{m}$ bead trapped by a 100mW laser. Measurement time was 900s at 10kHz . This clearly shows that frequencies lower than 5Hz have no significance in the measurements as the Lorentzian is usually fit from around $10\text{--}100\text{Hz}$.

became apparent that measurements would contain a large amount of low frequency (around 10Hz and less) noise when vehicles drove outside the lab. To gauge the effect of this a number of experiments were performed.

Figures 51 and 52 show the effect of vibrations in the building and from sources on the optical vibration damping table.

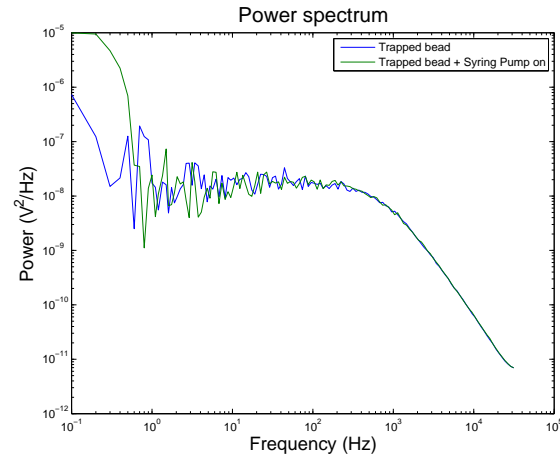


Figure 51: A $1.54\mu\text{m}$ particle trapped using a 100mW laser. The vibrational noise was generated by a syringe pump that was placed on the table was switched on.

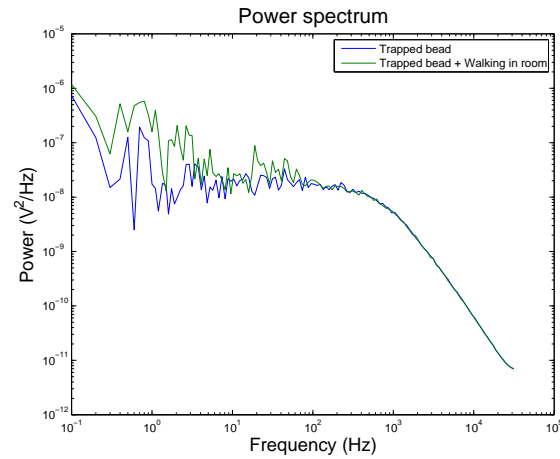


Figure 52: A $1.54\mu\text{m}$ particle trapped using a 100mW laser while walking around the room.

7 Flow cell Design

Initial plans for the project consisted of measuring the rheological properties of sponge like micells. The micelles will orientate themselves depending on the flow history. If we transport the fluid through a narrow channel the micells will arrange themselves so that they are parallel with the direction of flow. This would allow us to study the deformation and flow of the micells when a pressure was applied using the optical tweezers. For this reason we required a flow cell where the micelles could be aligned, which should therefore fulfil the following criteria.

Volume: The volume had to be small enough so that there was not too much wasted material.

Transparent: As light would be passing through the flow cell, transparency was essential, preferably with as low absorption as possible.

Thickness: The microscope objective posed a limit on the thickness of the glass, in our case we are limited to 0.14–0.18mm.

Control: As the micell solution is pushed through the flow cell they align themselves with the flow, upon stopping the flow, the micells should ideally come to a complete standstill without reorienting themselves.

7.1 Initial flow cell design

The initial idea (Figure 53) was to use standard microscopy glassware as this would be clear and thin enough not to cause any problems with the microscope. The microscope slide and cover slip were glued together using a layer of double sided tape with a channel cut into it. Two tubes designed for gas chromatography were used to deliver the micell solution into the flow cell by way of a nipple.

7.2 Flow cell revisions

This initial design (Figure 54(a) and Figure 56(a)) worked but suffered from some problems, the nipples would loosen, double sided tape would loosen and we had some elasticity which caused the fluid to flow even after the syringe pump was stopped.

A new design (Figure 54(b)) involved exchanging the nipples by widening the hole and placing the tubing directly into the glass microscope slide. The surface around the tube was roughened and a dab of silicone glue placed

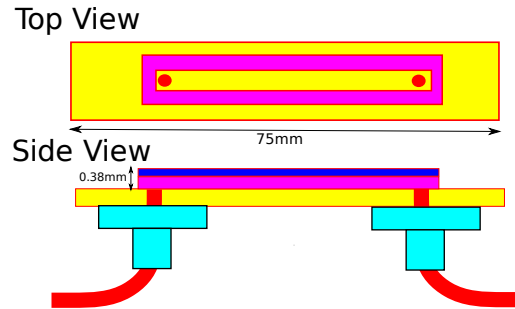


Figure 53: Initial Flow cell design, yellow shows the microscope slide, blue shows cover slip, red shows tubing, purple shows the tape and finally light blue shows a plastic nipple. The cover slip is not shown in the top view for clarity.

around the tubing, see Figure 54(d). This held up very well, even under relatively high flow rates (thus pressure) without any leaks.

The second issue encountered was that the tape was unable to hold the glass plates together when a larger flow rate was applied. Initially different tape was used and this alleviated the problem for a short while, but in the process of making measurements on the channel size a leak formed. An attempt was made using microscope slides and Ultraviolet activated glass glue as in Figure 55 and Figure 56(b) but the glue was unable to completely seal the channel. For this reason as well as the complexity this design was dropped.

For the final design a hole was drilled at both ends of a glass slide, and tubing was placed into each of the holes then sealed off with a drop of silicone glue. Taking a piece of Parafilm®M (thickness $127\mu m$) a small channel was cut into the centre such that when placed onto the glass slide a channel between the two holes was formed (Figure 56(c)). Heating the flow cell with a hot plate for a couple of seconds caused the Parafilm to melt and seal the channel.

7.2.1 Characterisation of Flow cell

In order to determine the volume of the final flow cell two different methods used. These consisted of using measurements of the channel geometry and using a syringe pump.

Volume by channel size Measurements of the length and width were made using a vernier calliper. The height was measured using a micrometre,

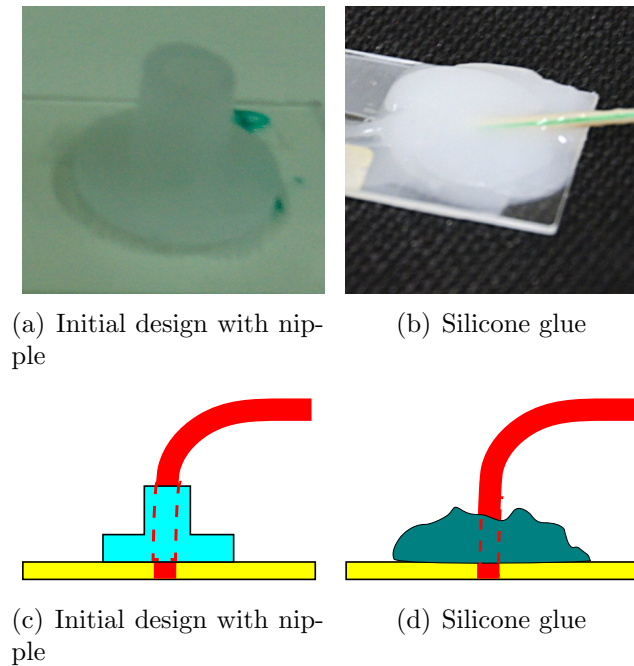


Figure 54: Two different designs for connecting tubing to the flow cell

by comparing the height of a microscope slide and cover slip with and without a layer of parafilm

The volume measurements aren't very accurate due to the large variance in width. For this reason a direct volumetric measurement of the channel was made.

Volume using syringe pump. Using the syringe pump, water was run through the setup until it first entered the Flow cell. Once inside the Flow cell the pump was set to a very slow speed and the volume required to move the front from one side to the other was measured. This resulted in a

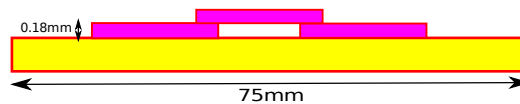


Figure 55: Side view showing design using ultraviolet activated glue. 3 glass cover slips (purple) placed on top of each other to create a channel through the centre together with the microscope slide (yellow).

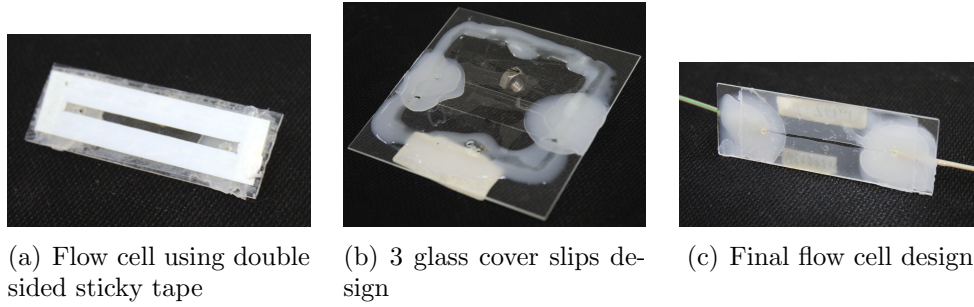


Figure 56: Different Flow cell designs

Length:	$44.6mm \pm 0.2mm$
Width:	$1.1mm \pm 0.5mm$
Height:	$0.130mm \pm 0.05mm$
Volume:	$6.378\mu l \pm 1.5\mu l$

Table 15: Flow cell properties by means of length measurements

volume measurement of $7.61\mu l$. The uncertainty in the volume is very low as a $250\mu l$ Hamilton μl syringe was used, hence flow rates of less than $0.01\mu l$ per minute were possible.

7.3 Flow cell use

A clean flow cell is mounted in the microscope in the usual way, with the syringe pump mounted within reach of the tubing. A syringe filled with a suspension containing silica beads is placed into the syringe pump and attached to the tubing. The other tube is placed level with the slide so there will be little to no flow due to gravity. The syringe pump is programmed with the syringe specifications and flow rate. Initially the flow rate can be quite large as we are only filling the tubing and there will be no pressure on the flow cell. Once inside the cell the flow rate was decreased. After it has reached the other side the flow can be stopped and should result in almost no continued flow.

8 Discussion of project results

This project initially started out with a goal of measuring the rheological properties of sponge like micelles, unfortunately due to problems calibrating the optical tweezers this goal was never achieved. The standard technique of calibrating optical tweezers using the Lorentzian curve worked quite well, this is rather unsurprising as we are not given any independent variables with which we can compare the accuracy of this calibration. This method would be sufficient in situations where relative measurements are made. For this reason a great deal of time and effort was spent on trying to get the sinusoidally driven calibration to work.

One of the greatest problems we encountered with this method was consistency, most of the time the experimental results would lie around 30–50% with occasional readings at close to 90%. There are a couple of likely roots of the problem.

- **Leakage:** As the data is obtained in a finite time with a finite sampling rate there will be some spectral leakage, causing neighbouring frequencies to gain some amplitude and the drive frequency to lose some power. In all the measurements made the signal was always spread over 3 frequencies. One of the methods used to try and correct this was to sum all three frequencies, although this could be seen as an incorrect way of making measurements. If we assume we have an infinitely sharp peak centred around our drive frequency, any leakage would “leak” to the neighbouring frequencies. On the other hand if the peak is spread out over a couple of frequencies, taking each of the neighbouring peaks might represent another part of the curve, in this case it would not be correct to sum the peaks in the spectrum.
- **Fourier transform:** As detailed in section 5.4 and A.3 there may have been an issue with the coefficients in the Fourier transform. Although the FFT algorithm used seems to give the correct response when fed with a Gaussian function. Unfortunately due to limited time I have been unable to do a similar test by starting with a known data set.
- **Low frequency noise:** One of the major issues during the project was that of low frequency noise. The low frequency noise was at times so large that it was necessary to tweak the fitting parameters for the Matlab fitting function in order to get a fit. In the final stages of the project I came across one possible solution, which would be to increase the laser power. This should create a stronger trap and hopefully prevent the particle moving too much. Additionally increasing the power

would increase the plateau, pushing the corner frequency to higher values. Increasing the power will also lead to the inertial term becoming less important and should increase the adherence to the equation that have been derived. The effect of the beam stability should be accessed to ensure that it does not introduce more noise.

- **Unidentified peaks:** A large number of the data sets have a large peak at a frequency that I was unable to identify, it occurred at around 24–26Hz, attempts to find the source of this noise, detailed in section 5.9, failed to identify any particular source that could be responsible.
- **Alignment:** Errors in alignment can lead to a noisy signal and weak trap. This occurred on one occasion, but a simple realignment solved this problem.

As final attempt to calibrate the optical tweezer was made using the flow method (detailed in section 5.8.1). This looked quite promising, with a constant slope in the velocity-voltage curve. Unfortunately due to an error that I have been unable to find, the calibration results are at least an order of magnitude out. Had there been a bit more time, it would have been a good idea to try this calibration method using the green laser, which was unavailable at the time this calibration technique was tested.

I did notice that if there were any alignment errors, there would be a considerable amount of low frequency noise. These alignment errors are generally a simple fix but take a considerable amount of time to correct. considerable amount.

One method to eliminate some of the issues encountered in determining β the conversion factor would be to make voltage-displacement measurements using a video system. This could give an indication to where the calibration issues could reside.

Comparing the diffusion coefficients of the dynamic light scattering and the various calibration methods shows that we are very close to the tabulated values (Table 16).

It is clear that the flow method is a long way off, but considering the simplicity of the method and that almost no time was available to use this method I strongly believe that given more time this would prove to be a reliable method.

Tabulated	$3.01 \cdot 10^{-13}$
Dynamic light scattering	$2.98 \cdot 10^{-13}$
Basic Lorentzian fit	$2.88 \cdot 10^{-13}$
With external drive (best)	$3.08 \cdot 10^{-13}$
With external drive (mean)	$5.52 \cdot 10^{-13}$
Flow method	$1.67 \cdot 10^{-13}$

Table 16: Diffusion coefficients for the various experiments performed in this project. All values is in m^2/s .

A Computer Code

All the following code is available either from the original PDF or through the site listed in the references[10].

A.1 Automatic laser blocker

The code presented below compiles in BoostC and is compatible with either the microchip PIC-12F675 or PIC-12F629. Schematics can be seen in Figure 57. The timing choices are adjusted using jumper switch S1.

```
#include <system.h>

//Target PIC16F84 configuration word
#pragma DATA _CONFIG, _CPD_OFF & _PWRTE_OFF & _WDT_OFF & _CP_OFF &
        _BODEN_OFF & _MCLRE_OFF & _PWRTE_ON & _INTRC_OSC_NOCLKOUT
//Set clock frequency
#pragma CLOCK_FREQ      4000000

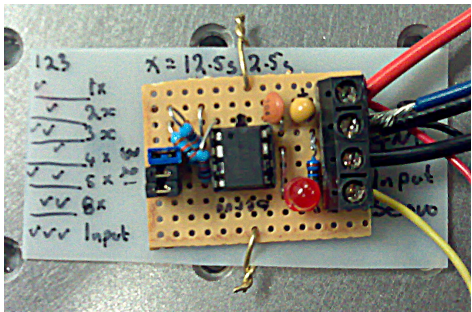
void main( void ) {
    //Initialize microchip and set ports to zero
    int loops=0,delay1=0,delay2=0,p,mult=1;
    trisio=111010b;
    gpio=0x00;
    ansel = 0x00;
    cmcon = 0x07;

    //Do a power up flash with LED to shall all ok
    for(p=0;p<2;p++){
        gpio.2=1;
        delay_ms(500);
        gpio.2=0;
        delay_ms(500);}
    delay_s(4);

    //Check Jumper settings, and store settings
    if( gpio.3 && !gpio.4 && !gpio.5)
        mult=1;
    else if(!gpio.3 && gpio.4 && !gpio.5)
        mult=2;
    else if(gpio.3 && gpio.4 && !gpio.5)
        mult=3;
    else if(!gpio.3 && !gpio.4 && gpio.5)
        mult=4;
    else if(gpio.3 && !gpio.4 && gpio.5)
        mult=5;
    else if(!gpio.3 && gpio.4 && gpio.5)
        mult=8;
    else if(gpio.3 && gpio.4 && gpio.5)
        mult=0;
    else
        mult=99;

    //Blink LED To indicate setting
    for(p=0;p<mult;p++){
        gpio.2=1;
        delay_ms(1500);
        gpio.2=0;
        delay_ms(1000);}
    //Manual controll
    if(mult==0)
        for(p=0;p<10;p++){
            gpio.2=1;
            delay_ms(300);
            gpio.2=0;
            delay_ms(100); }
    delay1=730*mult;
    delay2=155*mult;

    //Repetativly lift and lower beam block
    while(1){
        if(mult!=0){
            while(loops<delay1){
```

(b) Image of finished laser blocker

Figure 57: Schematics for PWM laser blocker, servo sens is connected to pin 2 of JP1, S1 is used to select timing

S1	S2	S3	Blocked	Open
X			12.5s	2.5s
	X		25s	5s
X	X		37.5s	7.5s
		X	50s	10s
X		X	62.5	12.5s
	X	X	100s	20s
X	X	X	I/O	I/O

Table 17: Timing choices for beam blocker. The columns represent the jumpers. I/O represents a computer controlled sensing on port GP1, where 1 blocks and 0 is open.

```
        delay_100us(190);
        gpio.0=1;
        delay_100us(10);
        gpio.0=0;
        loops++;}
    loops=0;
    while(loops<delay2){
        delay_100us(180);
        gpio.0=1;
        gpio.2=1;
        delay_100us(20);
        gpio.0=0;
        loops++; }
    gpio.2=0;
    loops=0; }
else{
    while(1){
        if(gpio.1){
            delay_100us(190);
            gpio.0=1;
            gpio.2=1;
            delay_100us(10);
            gpio.0=0; }
        else{
            delay_100us(180);
            gpio.0=1;
            gpio.2=0;
            delay_100us(20);
            gpio.0=0; }
        }
    }
}
```

A.2 Code for taking compressed data and making a power spectrum

The following code was used to make power spectra as on larger data sets Matlab was unable to cope with 1Gb of ram. The program uses the "Fastest Fourier Transform in the West", and HDF5 libraries to take a Fourier transform and import the data.

```

/*****
 *
 * Take the FFT of a data set and export as a power spectrum
 *
 * to compile:
 *   g++ pspectra_Vx.cpp -lhdf5_hl -lhdf5 -Wall -pedantic \
 *   -lfftw3 readinput.cpp *
 *
 * v5.2 Changed formula for binning
 * v5.1 Added labview v3 fileformat support
 * v5.0 Attempt to add welch windowign
 *   added flat top window
 * v4.10 Removed longdouble
 * v4.9 Introduced longdouble
 * v4.8 Added max checking and output
 * v4.8 Added max checking and output
 * v4.7 Clarified help section a bit
 * v4.6 Fix bug with freq calculation, add log binnig
 * v4.5 Attempt at adding hdf5 output, added a couple of
 *   window functions and set Hamming default
 *   Run through valgrind
 * v4.4 Added sine window
 * V4.3 Modified code to accept a double time, added no window
 * V4.2 Added colour and shift of data, default no normalize
 * V4.1 Changed default to not normalize or bin
 * V4 Checked working with variance = sum(S(f)*df)
 *   Matlab var(test_raw_windowed), sum(test_psd(2:end,:)*0.1)*
 * V3 Small corrections and choice of windowing function
 * V2 History
 * V1 Ancient History
 *
 *
 *
 *
 *
 *
 *****/

#include <iostream>
#include <cmath>
#include <complex>
#include "fftw3.h"
#include <hdf5.h>
#include <H5LT.h>
#include "readinput.h"

//Add colour output to make things more readable
//Check if in cygwin, and alter coloura appropriately.
#if defined(__GNUC__) && defined(__CYGWIN__)
#define redc "\033[22;31m"
#define normc "\033[01;37m"
#define bluec "\033[22;34m"
#define yellowc "\033[01;33m"
#else
#define redc "\033[22;31m"
#define normc "\033[01;37m"
#define bluec "\033[22;34m"
#define yellowc "\033[01;33m"
#endif

//Max voltage before warning
#define maxV 9.98

//Allocate a 2d array compatible with hdf5 library
using namespace std;
template < typename T >
T **Allocate2DArray( int nRows, int nCols)
{
    T **ppi;
    T *pool;
    T *curPtr;

```

```

    //(step 1) allocate memory for array of elements of column

    ppi = new T*[nRows];

    //(step 2) allocate memory for array of elements of each row
    pool = new T [nRows * nCols];

    // Now point the pointers in the right place
    curPtr = pool;
    for( int i = 0; i < nRows; i++)
    {
        *(ppi + i) = curPtr;
        curPtr += nCols;
    }
    return ppi;
}

//Deallocate the 2d array
template < typename T >
void Free2DArray(T** Array)
{
    delete [] *Array;
    delete [] Array;
}

//Initialize function definitions
long int power(int x, int n);
long count_lines (string filename);
double average(double * array, int i, int f);
inline double hann(int n, int N);
void version(void);
void help(string filename);
inline double trianglew(int n, int N);
inline double sinew(int n, int N);
inline double bartlettthann(int n, int N);
inline double fourtBlackHarr(int n, int N);
inline double fourtBlackHarr90db(int n, int N);
inline double Hamming(int n, int N);
inline double flattop(int n, int N);
double window(int n, int N, int argc, char* argv[]);
inline void countr(string red);

int main(int argc, char* argv[]) {
    cout << endl<<endl;
    // Check input for version call
    if (readinputb(argc,argv,"V")){
        version();
    }
    bool showHelp=false;
    //Check input for help call
    if (readinputb(argc,argv,"H") || readinputb(argc,argv,"h") )
        showHelp=true;
    //Initialize all variables
    long warn=0;
    int bins=0;
    bool Force=false;
    int nan=0;
    string filename, outfilename, infofilename, xfilename, yfilename, zfilename;
    bool shift = !readinputb(argc,argv,"S");
    bool welch = true;
    int segments = 1;
    int overlap= 0;
    if(welch){
        segments = readinputi(argc,argv,"s",1);
        overlap=readinputi(argc,argv,"l",0);
    }
    double shiftx=0,shifty=0;
    //Make filename variables and set up program depending on number of inputs
    if(2 == argc ){
        filename=argv[1];
        infofilename=filename + ".txt";
        if ( !file_exists(infofilename))
            help(argv[0]);
        outfilename=filename;
        xfilename=filename + "_x.hws";
        yfilename=filename + "_y.hws";
        zfilename=filename + "_z.hws";
        bins=0;
    }
    else if (5>argc)
        showHelp=true;
    else{
        bins=readinputi(argc,argv,"b",0);
    }
}

```

```

    Force = readinputb(argc,argv,"F");
    filename=readinputs(argc,argv,"i","infile");
    outfilename=readinputs(argc,argv,"o","outfile");
    infofilename = filename + ".txt";
    xfilename =filename + "-x.hws";
    yfilename =filename + "-y.hws";
    zfilename =filename + "-z.hws";
}
string outfilename=outfilename+"_psd.txt";
if(showHelp)
    help(argv[0]);
//Set variables from stdin
bool normalize = !readinputb(argc,argv,"n");
bool nocomment = readinputb(argc,argv,"D");
int find_max = readinputb(argc,argv,"M");
// Check all files exist
if( !file_exists(infofilename)){
    cout <<"Please specify a valid input information file"<<endl<<
        "e.g. E01.txt " <<endl;
    exit(EXIT.FAILURE);
}
if( !file_exists(xfilename) || !file_exists(yfilename) ||
    !file_exists(zfilename) ){
    cout <<"Please verify valid data files"<<endl<<
        "e.g. E01.x.txt, E01.y.txt and E01.z.txt " <<endl;
    exit(EXIT.FAILURE);
}
FILE * InfoFile;
InfoFile = fopen(infofilename.c_str(),"r");
cout << "Reading infofile:"<<infofilename<<endl;
// Create a variable to hold experimental info
char test[200];
//Read last line (bit silly but works)
for(int i =0;i<99;i++){
    fgets(test,99,InfoFile);
}
// initialize variables used in data analysis
int samples,version=-1;
double tempavg,tempi,tempf,bandwidth,time,particlesize,drivefreq,drivedist,
    power,samples_d;
char version_ascii=0;
//Determine length of line (1st or second version
for(int j=0;version<1;j++){
    version_ascii=test[strlen(test)-1-j];
    if(version_ascii<'1' || version_ascii>'9')
        version_ascii='0';
    version=version_ascii-48;
    if (j>5 || version>3){
        cout << "Wrong specifications file, please correct"<<endl;
        cout << "Final line should be:"<<endl;
        cout << "\"Sampling Freq\" \"Time\" \"Avg Temp\" \"I temp\"\"
            <<"\"F temp\" \"Version\""<<endl;
        exit(EXIT.FAILURE);
    }
}
//Read input file, depending on version
if(version==3)
    sscanf(test,"%le %le %le %le %le %le %le %le %le %d",
        &samples_d,&time,&tempavg,&tempf,&tempf,&particlesize,
        &drivefreq,&drivedist,&power,&version);
else if(version==2)
    sscanf(test,"%le %le %le %le %le %le %le %le %d",
        &samples_d,&time,&tempavg,&tempf,&tempf,&particlesize,
        &drivefreq,&drivedist,&version);
else if (version==1)
    sscanf(test,"%le %le %le %le %le %d",&samples_d,&time,&tempavg,
        &tempf,&tempf,&version);
samples=(int)samples_d;
//read from the test variable and read info
fclose(InfoFile);
//If we are splitting up the datafiles, find size of segments
long orglines=(long)(samples*time);
long lines=orglines/segments;
long freq=samples;
long overlaplines=orglines/segments*overlap/100;
long discardedpoints=orglines+overlaplines*segments-lines*segments;
while(discardedpoints>lines){
    segments++;
    discardedpoints=orglines+overlaplines*segments-lines*segments;
    cout << "Added segment, due to overlap"<<endl;
}
//long lines=count_lines(filename);

```

```

cout << "\n\nFFT of Optical tweezer data\n"<<
  "Note that the number of bins is taken as an approximate\n"<<
  "Sampling frequency : " << freq << endl<<
  "In Filename:"<<filename<<"\nOutfilename : "<<outfilename<<endl<<
  "Sampling rate:"<<samples<<" samples/s"<<endl<<
  "Sampling time:"<<time<<" s"<<endl<<
  "File version:"<<version<<endl<<endl;
if(!normalize)
  cout << bluec<<"You have choosen not to normaize the data"
    <<normc<<endl<<endl;
double * freq_list= new double[lines/2];
cout << "Generating frequency data";
for(int i=0;i<lines/2;i++){
  freq_list[i]=samples*(double)i/lines;
}
cout <<" , done"<<endl<<endl;

hid_t      file_id ;
herr_t     status;
bandwidth=(double)lines*(double)lines/((double)time/2.0;
/*****
 *      Import Z Data and take FFT
 *      Conjugate and save to standard array
 *
 *****/
double * zdata = new double[lines/2];
double * raw_z_data = new double[orglines];
cout << "Loading Zdata into memory: " << zfilename;
// Open HDF File
file_id = H5Fopen (zfilename.c_str(), H5F_ACC_RDONLY, H5P_DEFAULT);
status = H5LTread_dataset_double(file_id ,
  "/wfm_group0/axes/axis1/data_vector/data", raw_z_data);
status = H5Fclose (file_id);
std::complex<double>* inz=new std::complex<double> [lines];
std::complex<double>* outz=new std::complex<double> [lines];
cout << " , Starting FFT\n";
//Segment and analyze
for(int segs=0;segs<segments;segs++){
  for(int i=0;i<lines;i++){
    inz[i].real()=raw_z_data[i]*window(i,lines,argc,argv);
    if(raw_z_data[i]>9.98)
      warn++;
    inz[i].imag()=0.0;
  }
  //Perform FFT
  fftw_plan pz = fftw_plan_dft_1d(lines ,
    reinterpret_cast<fftw_complex*>(inz),
    reinterpret_cast<fftw_complex*>(outz),
    FFTW_FORWARD, FFTW_ESTIMATE);
  fftw_execute(pz);
  for(int i=0;i<lines/2;i++){
    zdata[i]=real(outz[i]*conj(outz[i]))/bandwidth;
  }
  fftw_destroy_plan(pz);
}
delete [] inz;
delete [] outz;
cout << "FFT'd and conjugated Z data"<<endl<<endl;
/*****
 *      Import X Data and take FFT
 *      Conjugate and save to standard array
 *
 *****/
std::complex<double>* inx=new std::complex<double> [lines];
std::complex<double>* outx=new std::complex<double> [lines];
double * xdata = new double[lines/2];
for(int i=0;i<lines/2;i++){
  xdata[i]=0;
double * tempdata = new double[lines];
double * rawxdata = new double[orglines];
cout << "Loading Xdata into memory: " << xfilename;
file_id = H5Fopen (xfilename.c_str(), H5F_ACC_RDONLY, H5P_DEFAULT);
status = H5LTread_dataset_double(file_id ,
  "/wfm_group0/axes/axis1/data_vector/data", rawxdata);
status = H5Fclose (file_id);
cout << " , Starting FFT\n";
for(int segs=0;segs<segments;segs++){
  int segstart=segs*lines-segs*overlapplines;
  for(int p=0;p<lines;p++){
    tempdata[p]=rawxdata[p+segstart];
  }
  if(shift)
    shiftx=average(tempdata,0,lines-1);

```

```

    if(normalize)
        for(int i=0;i<lines;i++){
            if(raw_z_data[i]==0){
                cout << "Please rerun without normalization,"
                    <<"divide by zero"<<endl;
                exit(EXIT.FAILURE);
            }
            inx[i].real()=(tempdata[i]-shiftx)/raw_z_data[i+segstart]*
                window(i,lines,argc,argv);
            inx[i].imag()=0.0;
        }
    else
        for(int i=0;i<lines;i++){
            inx[i].real()=(tempdata[i]-shiftx)*window(i,lines,argc,argv);
            inx[i].imag()=0.0;
        }

    fftw_plan px = fftw_plan_dft_1d(lines,
        reinterpret_cast<fftw_complex*>(inx),
        reinterpret_cast<fftw_complex*>(outx),
        FFTW_FORWARD, FFTW_ESTIMATE);
    fftw_execute(px);
    for(int i=0;i<lines/2;i++){
        xdata[i]+=real(outx[i]*conj(outx[i]))/bandwidth;
    }
    fftw_destroy_plan(px);
}
delete [] rawxdata;
delete [] inx;
delete [] outx;
cout << "FFT'd and conjugated X data"<<endl<<endl;
/*****
*
*   Import Y Data and take FFT
*   Conjugate and save to standard array
*
*****/
std::complex<double>* iny=new std::complex<double> [lines];
std::complex<double>* outy=new std::complex<double> [lines];
double * ydata = new double [lines/2];
for(int i=0;i<lines/2;i++){
    ydata[i]=0;
}
double * rawydata = new double [orglines];
cout << "Loading Ydata into memory: " << yfilename;
file_id = H5Fopen (yfilename.c_str(), H5F_ACC_RDONLY, H5P_DEFAULT);
status = H5LTread_dataset_double (file_id,
    "/wfm_group0/axes/axis1/data_vector/data",rawydata);
status = H5Fclose (file_id);
tempdata[0]=5;
cout << "Starting FFT\n";
for(int segs=0;segs<segments;segs++){
    int segstart=segs*lines-segs*overlaplines;
    for(int p=0;p<lines;p++){
        tempdata[p]=rawydata[p+segstart];
    }
    if(shift)
        shifty = average(tempdata,0,lines-1);
    if(normalize)
        for(int i=0;i<lines;i++){
            if(raw_z_data[i]==0){
                cout << "Please rerun without normalization, divide by zero"
                    <<endl;
                exit(EXIT.FAILURE);
            }
            iny[i].real()=(tempdata[i]-shifty)/raw_z_data[i+segstart]*
                window(i,lines,argc,argv);
            iny[i].imag()=0.0;
        }
    else
        for(int i=0;i<lines;i++){
            iny[i].real()=(tempdata[i]-shifty)*window(i,lines,argc,argv);
            iny[i].imag()=0.0;
        }

    fftw_plan py = fftw_plan_dft_1d(lines,
        reinterpret_cast<fftw_complex*>(iny),
        reinterpret_cast<fftw_complex*>(outy),
        FFTW_FORWARD, FFTW_ESTIMATE);
    fftw_execute(py);
    for(int i=0;i<lines/2;i++){
        ydata[i]+=real(outy[i]*conj(outy[i]))/bandwidth;
    }
    fftw_destroy_plan(py);
}
delete [] tempdata;

```

```

delete [] rawydata;
delete [] iny;
delete [] outy;
cout << "FFT'd and conjugated Y data"<<endl<<endl;
for (int i =0;i<lines/2;i++){
    ydata[i]=ydata[i]/segments;
    xdata[i]=xdata[i]/segments;
}

FILE * outfile;
outfile = fopen(outfilename.c_str(),"w");
double xmax_freq=0,xmax_value=0;
double ymax_freq=0,ymax_value=0;

/*****
 * Find Max
 * *****/

if (find_max){
    for (int i=0;i<lines/2;i++){
        if (freq_list[i]>12 && freq_list[i]<42){
            if (xdata[i]>xmax_value){
                xmax_freq=freq_list[i];
                xmax_value=xdata[i];
            }
        }
    }
    for (int i=0;i<lines/2;i++){
        if (freq_list[i]>12 && freq_list[i]<42){
            if (ydata[i]>ymax_value){
                ymax_freq=freq_list[i];
                ymax_value=ydata[i];
            }
        }
    }
    printf("XMax at %g Hz with %g\n",xmax_freq,xmax_value);
    printf("YMax at %g Hz with %g\n",ymax_freq,ymax_value);
}

/*****
 *
 * Write comments to file
 *
 * *****/

if (!nocoment){
    if ((warn>0) && normalize)
        fprintf(outfile,"%sWARNING NORMALIZATION MAY NOT BE CORRECT,"
            <<"VOLTAGE Exceeded %gV %ld times (%ld%)\n",
            maxV,warn,warn*100/lines);
    fprintf(outfile,"%s %s\n","Source file name: ",__FILE__);
    fprintf(outfile,"%s %s %s %s\n","Compiled at: ",__DATE__,
        " ",__TIME__, " with GNU g++ ");
    fprintf(outfile,"%s %d.%d.%d\n", __GNUG__, __GNUG_MINOR__,
        <<"",__GNUC_PATCHLEVEL__);
    fprintf(outfile,"%s %s\n","FFTW Version: ",fftw_version);
    fprintf(outfile,"%s %d.%d.%d.%s\n","HDF5 Version: ",
        H5.VERS.MAJOR,H5.VERS.MINOR,H5.VERS.RELEASE,H5.VERS.SUBRELEASE);
    if (Force)
        fprintf(outfile,"%sInput version not correct, user override\n");
    if (normalize)
        fprintf(outfile,"%sData was normalized using Z\n");
    if (find_max)
        fprintf(outfile,"%sMax at %g Hz with %g\n",xmax_freq,xmax_value);
    switch (readinputc(argc,argv,"W","Q")){
        case 'H':
            fprintf(outfile,"%sWindowing Function: Hann\n");
            break;
        case 'T':
            fprintf(outfile,"%sWindowing Function: Triangle\n");
            break;
        case '9':
            fprintf(outfile,"%sWindowing Function: 4term BlackmannHarris(90db)\n");
            break;
        case '4':
            fprintf(outfile,"%sWindowing Function: 4term BlackmannHarris(72db)\n");
            break;
        case 'B':
            fprintf(outfile,"%sWindowing Function: Barlett-Hann\n");
            break;
        case '1':
            fprintf(outfile,"%sWindowing Function: None/rectangular\n");
            break;
    }
}

```



```

        case 'S':
            fprintf(outfile, "%s\n", "Windowing Function: Sine");
            break;
        case 'A':
            fprintf(outfile, "%s\n", "Windowing Function: Hamming");
            break;
        case 'F':
            fprintf(outfile, "%s\n", "Windowing Function: Flat top");
            break;
        default:
            fprintf(outfile, "%s\n", "Windowing Function: Hamming(Default)");
    }
    if (nan > 0)
        fprintf(outfile, "\n\nWARNING ENCOUNTERED NaN's, %d times\n", nan);
}

/*****
 *
 * Put data in correct array for writing to file.
 *
 * *****/
double * fxdata = new double[lines/2];
double * fydata = new double[lines/2];
double * fzdata = new double[lines/2];
double * ffdata = new double[lines/2];
int numlines=0;
for (int i=0; i<lines/2; i++)
    fxdata[i]=fydata[i]=fzdata[i]=ffdata[i]=0.0;
if (bins == 0) {
    if (!nocoment)
        fprintf(outfile, "%s\n", "Binning was not performed");
    numlines=lines/2;
    for (int i=0; i<lines/2; i++) {
        ffdata[i]=freq_list[i];
        fxdata[i]=xdata[i];
        fydata[i]=ydata[i];
        fzdata[i]=zdata[i];
    }
}
else {
    double min=freq_list[1];
    double max=lines/2;
    double delta=log(max/min)/bins;
    int * count=new int[lines/2];
    for (int i=0; i<lines/2; i++) {
        count[i]=0;
    }
    int bin=0;
    ffdata[0]=freq_list[0];
    fxdata[0]=xdata[0];
    fydata[0]=ydata[0];
    fzdata[0]=zdata[0];
    count[0]=1;
    int i;
    if (drivefreq <= 0)
        drivefreq=20;

    for (i=1; ffdata[i-1]<drivefreq+4; i++) {
        ffdata[i]=freq_list[i];
        fxdata[i]=xdata[i];
        fydata[i]=ydata[i];
        fzdata[i]=zdata[i];
        count[i]++;
    }
    for (; i<lines/2; i++) {
        bin=(int)(log((double)i/min)/delta);
        ffdata[bin]+=freq_list[i];
        fxdata[bin]+=xdata[i];
        fydata[bin]+=ydata[i];
        fzdata[bin]+=zdata[i];
        count[bin]++;
    }
    int j=0;
    for (int i=0; i<lines/2; i++) {
        ffdata[j]=ffdata[i]/count[i];
        if (isnan(ffdata[j]))
            j--;
        else {
            fxdata[j]=fxdata[i]/count[i];
            fydata[j]=fydata[i]/count[i];
            fzdata[j]=fzdata[i]/count[i];
        }
        j++;
    }
}

```

```

    }
    numlines=j;
    //if (!nocoment)
    //    fprintf(outfile,"%%Binning was performed with %d specified"
    //            <<"restulting in a binning coefficient of %g\n\n",
    //            bins, bincoef);

    delete [] xdata;
    delete [] ydata;
    delete [] zdata;
    delete [] raw_z_data;
    delete [] freq_list;

    /*****
    *
    * Write to file
    *
    * *****/
    cout << "Saving data..." << endl;
    if (readinputb(argc,argv,"t")){
        for (int i=0;i<numlines;i++){
            if (ffdata[i]-0.00001>ffdata[i-1])
                fprintf(outfile,"%20.9e,%20.9e,%20.9e,%20.9e\n",
                        ffdata[i],fxdata[i],fydata[i],fzdata[i]);
        }
    }
    else{
        hid_t file_oid,dataspaceo,dataset_oid;
        hsize_t dimso[2];
        herr_t statuso;
        double ** datx=Allocate2DArray<double>(4,numlines);
        for (int i=0;i<numlines;i++){
            datx[0][i]=ffdata[i];
            datx[1][i]=fxdata[i];
            datx[2][i]=fydata[i];
            datx[3][i]=fzdata[i]; // freq_list[i];
        }
        string h5file=outfilename+"_psd.hdf5";
        file_oid=H5Fcreate(h5file.c_str(),H5F_ACC_TRUNC,H5P_DEFAULT,H5P_DEFAULT);
        dimso[0]=4;
        dimso[1]=numlines;
        dataspaceo=H5Screate_simple(2,dimso,NULL);
        dataset_oid=H5Dcreate(file_oid,"dset",H5T_IEEE_F64BE,dataspaceo,
                             H5P_DEFAULT);
        statuso=H5Dwrite(dataset_oid,H5T_NATIVE_DOUBLE,H5S_ALL,H5S_ALL,
                         H5P_DEFAULT,&datx[0][0]);
        statuso=H5Dclose(dataset_oid);
        statuso=H5Sclose(dataspaceo);
        statuso=H5Fclose(file_oid);
        Free2DArray(datx);
    }

    /*****
    Delete stuff
    *****/

    delete [] ffdata;
    delete [] fxdata;
    delete [] fydata;
    delete [] fzdata;
    fclose(outfile);
    if (warn && normalize)
        cout << redc
              << "\n\nWARNING NORMALIZATION MAY NOT BE CORRECT, VOLTAGE EXceeded "
              << maxV << "V " << warn << " times (" << warn*100/lines
              << "%%)\n\n" << normc;
    if (nan > 0)
        cout << redc << "\n\nWARNING ENCOUNTERED NaN's, " << nan
              << " times\n\n" << normc;
    return 0;
}

//Custom power function which takes two integers
long int power(int x, int n) {
    long int sol=x;
    for (int i=1;i<n;i++){
        sol*=x;
    }
    return sol;
}

//Find how many files a file contains
long count_lines (string filename)

```

```

{
    FILE *wdata;
    wdata = fopen(filename.c_str(),"r");
    int c;
    unsigned long num_lines = 0;

    while ((c = getc (wdata)) != EOF)
        if (c == '\n')
            num_lines++;
    fclose(wdata);
    return num_lines;
}

//Average an array
double average(double * array,int i,int f){
    double sum=0;
    for (int x=i;x<=f;x++){
        sum+=array[x];
    }
    sum/=(f-i+1);
    return sum;
}

//Return windowed function
double window(int n,int N, int argc, char* argv[]){
    /* window with the appropriate windowing function */
    switch (readinputc(argc,argv,"W",'Q')){ //No default return
        case 'H':
            return hann(n,N);
            break;
        case 'T':
            return trianglew(n,N);
            break;
        case 'B':
            return bartlettthann(n,N);
            break;
        case 'R':
            return 1;
            break;
        case 'S':
            return sinew(n,N);
            break;
        case '4':
            return fourtBlackHarr(n,N);
            break;
        case '9':
            return fourtBlackHarr90db(n,N);
            break;
        case 'A':
            return Hamming(n,N);
            break;
        case 'F':
            return flattop(n,N);
            break;
        default:
            return Hamming(n,N);
    }
}

/*****
 *
 * Windowing functions
 *
 *****/

inline double Hamming(int n,int N){
    return 0.54 -0.46*cos(2*M_PI*n/N);
}

inline double fourtBlackHarr(int n,int N){
    return 0.40217 -0.49703*cos(2*M_PI*n/N)+0.09392*cos(4*M_PI*n/N)
        -0.00183*cos(6*M_PI*n/N);
}

inline double fourtBlackHarr90db(int n,int N){
    return 0.35875 -0.48829*cos(2*M_PI*n/N)+0.14128*cos(4*M_PI*n/N)
        -0.01168*cos(6*M_PI*n/N);
}

inline double hann(int n,int N){
    /* calculates the hann coefficients
     * w(n)=0.5*(1-cos(2*pi*n/N)) */
    return 0.5*(1-cos(2*M_PI*(double)n/((double)N)));
}

```

```

inline double sinew(int n, int N){
    /* Calculate the sine window
    * w(n) = sin(pi*n/(N-1)) */
    return sin(M_PI*n/(N-1));
}
inline double trianglew(int n,int N){
    /* Calculate the triangle window coeffiecntes
    * w(n)=2/N*(N/2-abs(n-(N-1)/2)) */
    return 2.0/N*(N/2.0-fabs(n-(N-1.0)/2.0));
}

inline double bartlettthann(int n,int N){
    /* calculate the barlett-hann window
    * w(n)= 0.62--0.48*abs(n/(N-1)-0.5)-0.38*cos(2*pi*n/(N-1)) */
    return 0.62-0.48*fabs(n/(N-1)-0.5)-0.38*cos(2*M_PI*n/(N-1));
}

inline double flattop(int n, int N){
    /* Calculate the flattop window
    * w(n) = a0 - a1 cos(2 pi n/(N-1)) +a2 cos(4 pi n/(N-1))
    -a3cos(6 pi n)/(N-1))+a4 cos(8 pi n/(N-1)) */
    return 1 -1.93*cos(2*M_PI*n/(N-1))+1.29*cos(4*M_PI*n/(N-1))
    +0.388*cos(6*M_PI*n/(N-1))+0.032*cos(8*M_PI*n/(N-1));
}

//Return version information
void version(){
    cout <<
    "Source file name: "<< __FILE__ <<endl<<
    "Compiled at: "<< __DATE__ <<" "<< __TIME__<< " with GNU g++ "
    << __GNUC__ << " " << __GNUC_MINOR__ << " "
    << __GNUC_PATCHLEVEL__ << endl << endl<<
    "FFTW Version: "<<fftw_version<<endl<<
    "HDF5 Version: "<<H5_VERS_MAJOR<<" "<<H5_VERS_MINOR<<" "
    <<H5_VERS_RELEASE<<" "<<H5_VERS_SUBRELEASE<<endl<<endl<<
    exit(EXIT.SUCCESS);
}

//Return help information
void help(string filename){
    cout << "Minimum call: "<< filename<< " -i infile -o outfile.txt"
    << endl << endl <<
    "General:\n"<<
    " -i input file name, without extension or channel\n"<<
    " -o output file name, where the psd will be placed"
    <<"it will be appended by _psd\n"<<
    " -t do not write hdf5 file , write to txt file\n"<<
    " -F Force using data even if version can't be confirmed\n"<<
    " -H Show help\n"<<
    " -V Version info"<<
    endl<<
    "Options Raw data:\n" <<
    " -n do not normalize the output with z voltage\n"<<
    " -S do not center around 0\n"<<
    " -W Window functions:"<<
    " H hann window\n"<<
    " B bartlett-Hann window\n"<<
    " T triangle window \n"<<
    " l no window function\n"<<
    " S sine window\n"<<
    " 4 4 term blackmannHarris (72db)\n"<<
    " 9 4 term blackmannHarris (90db)\n"<<
    " A Hamming\n"<<
    " F Flat-top\n"<<
    " -S # Segmented windowing\n"
    " -l # with # segments and #%% overlap\n"<<
    endl<<
    "Options PSD Data\n"<<
    " -b Number of bins to use, 0 mean no binning,"
    <<"this is used as an approximate\n"<<
    " -D make output datafile comment free\n"<<
    " -M report the maximum frequency and value \n"<<
    endl<<
    exit(EXIT.SUCCESS);
}

/* *****
* String colors :
* 033[22;30m - black

```

```

\033[22;31m - red
\033[22;32m - green
\033[22;33m - brown
\033[22;34m - blue
\033[22;35m - magenta
\033[22;36m - cyan
\033[22;37m - gray
\033[01;30m - dark gray
\033[01;31m - light red
\033[01;32m - light green
\033[01;33m - yellow
\033[01;34m - light blue
\033[01;35m - light magenta
\033[01;36m - light cyan
\033[01;37m - white
*****/

inline void coutr(string red){
    cout << redc <<red <<normc;
}

```

A.3 Code used to verify Fast Fourier transform coefficients

Due to some uncertainty about mixing of the Fourier coefficients, a small segment of the power spectral density code was taken and its own program was made. It creates an array of the Gaussian function, $y(x) = e^{-\lambda x^2}$ and Fourier transforms it. The program creates 3 files, Raw.txt which contains the raw waveform, FFT.txt which contains the fast Fourier transform of the raw wave, and finally PSD.txt which contains the power spectral density calculated in the same way as section A.2

```

/*
 * Program to create a Fourier transform using the FFTW library
 * 3 parameters that can be adjusted
 *     time: sampling time
 *     srates: sampling rate
 *     alpha: the parameter alpha for the gaussian
 *
 * To compile:
 * g++ fft.c -g -O0 -lfftw3 -o fft
 */
#include <iostream>
#include <complex>
#include <cmath>
#include "fftw3.h"
#define time 400
#define srates 2048
#define alpha 0.5
using namespace std;
int main (int){
    int lines=srates*time;
    double * x = new double[lines];
    double * t = new double[lines];
    double * f = new double[lines/2];
    //Create the gaussian wave y=exp(-alpha x^2)
    for(int i=0;i<lines;i++){
        t[i]=(double)i/srates;
        x[i]=sqrt(M_PI/alpha)*exp(-alpha*t[i]*t[i]);
    }

    //Calculate the Fourier transform
    complex<double>* inz=new complex<double> [lines];
    complex<double>* outz=new complex<double> [lines];
    for(int l=0;l<time*srates;l++){
        inz[l].real()=x[l];
        inz[l].imag()=0;
    }
    fftw_plan pz = fftw_plan_dft_1d(lines,reinterpret_cast<fftw_complex*>(inz),
        reinterpret_cast<fftw_complex*>(outz),

```

```
        FFTW_FORWARD, FFTW_ESTIMATE);
fftw_execute(pz);
for(int m=0;m<lines/2;m++){
    f[m]=1.0*(double)m/srate;
}

//Write the original signal to file:
FILE * outfile;
outfile = fopen("RAW.txt","w");
for(int k=0;k<lines;k++){
    fprintf(outfile,"%e %e\n",t[k],x[k]);
}
fclose(outfile);

//Write Fourier transform to file
FILE * outfile2;
outfile2 = fopen("FFT.txt","w");
for(int k=0;k<60000;k+=1){
    fprintf(outfile2,"%e %e\n",f[k],real(outh[k]));
}
fclose(outfile2);

//Write power spectrum to file
double bandwidth=(double)lines*(double)lines/(double)time/2.0;
FILE * outfile3;
outfile3 = fopen("PSD.txt","w");
for(int k=0;k<60000;k+=1){
    fprintf(outfile3,"%e %e\n",f[k],real(outh[k]*conj(outh[k])));
}
fclose(outfile3);
}
```

A.4 Labview code

Labview was chosen as the primary program for data acquisition as it is able to provide a live view of any signals captured as well as being able to process the data before exporting it for further processing.

A.4.1 Code for viewing live data

The code for live data acquisition can be seen in Figure 58. The user enters a sampling rate, typically quite low as the computer has problems keeping up with larger sampling rates, and the number of samples to acquire before showing a data updates. The user will be presented with a power spectral density along with live waveforms of the signals.

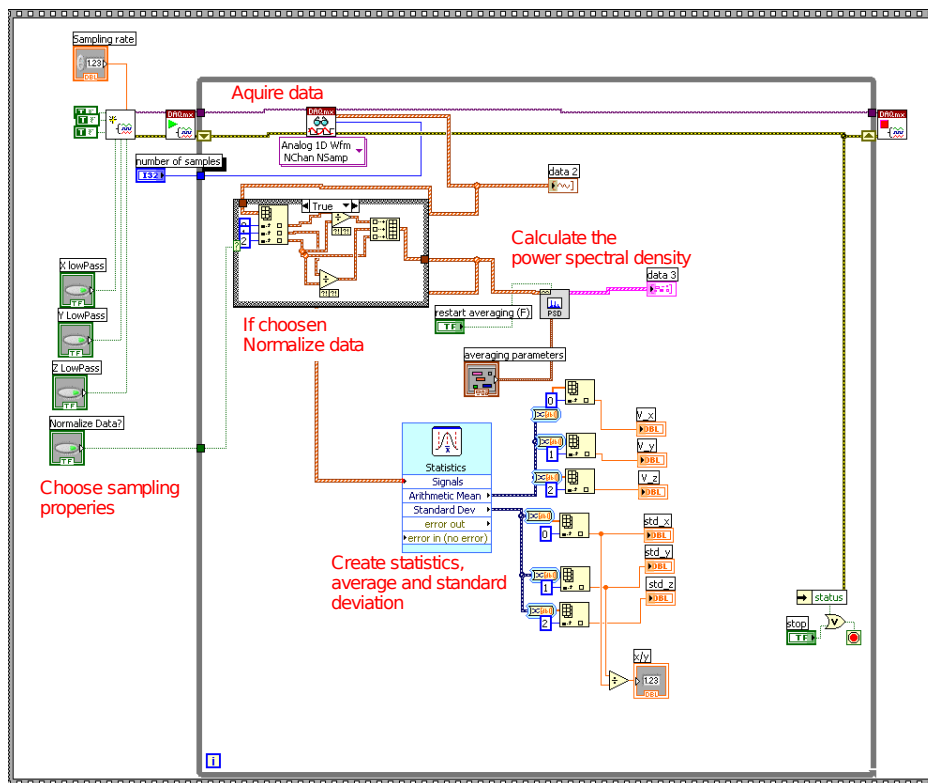


Figure 58: Labview code for viewing live data

A.4.2 Code for acquiring experimental data set

The code for continuous acquisition can be seen in Figure 59. The user specifies the time, and sample rate and optionally can provide additional

experimental data. This information along with the temperature for the experiment are written to an information file. The experimental data is written to a binary data file, of type HDF5, this format was chosen as Matlab and Labview are both naively able to read and write this data and we were able to avoid the large file size that accompanies raw ASCII files.

A.4.3 Code for initialising DAQ

Both routines listed in Section A.4.1 and A.4.2 make use of the code in Figure 60 for initialising the DAQ.

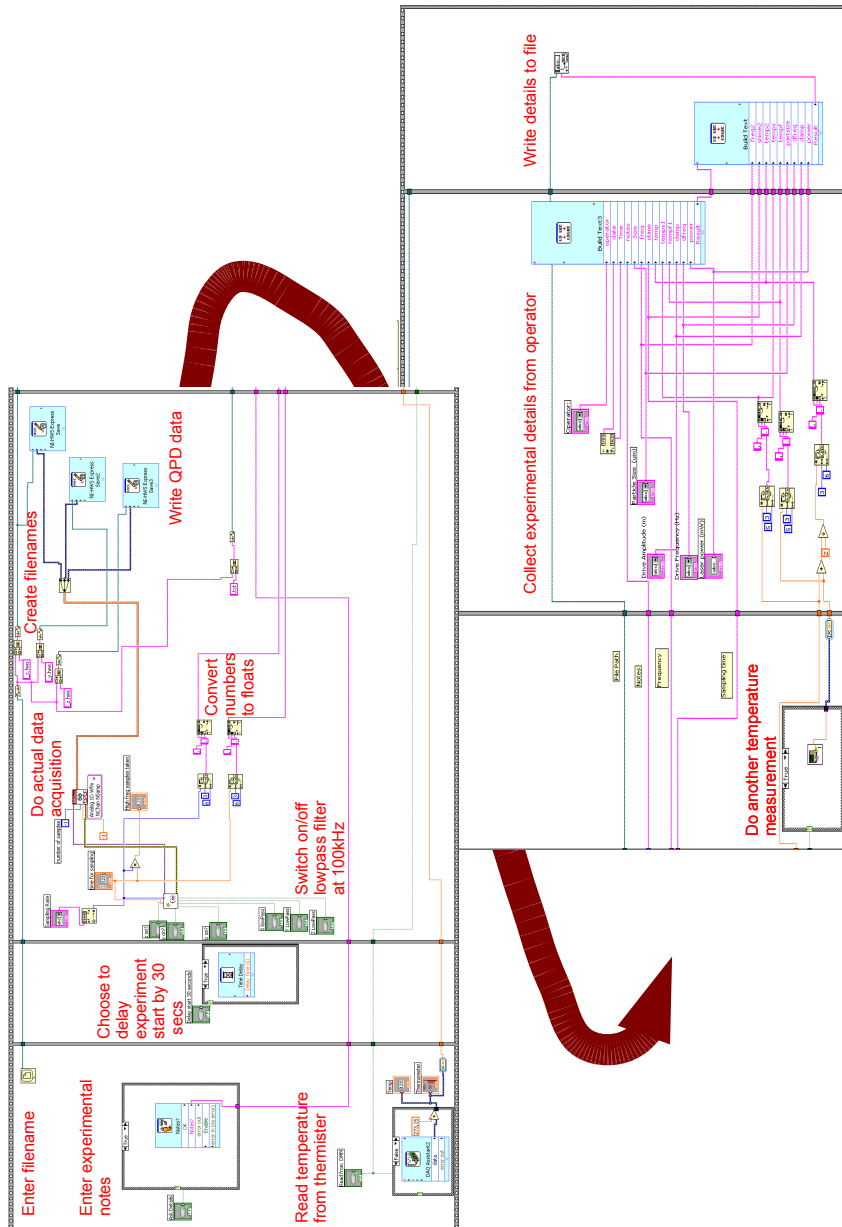


Figure 59: Lab view code for acquiring samples for a specified time and sample rate

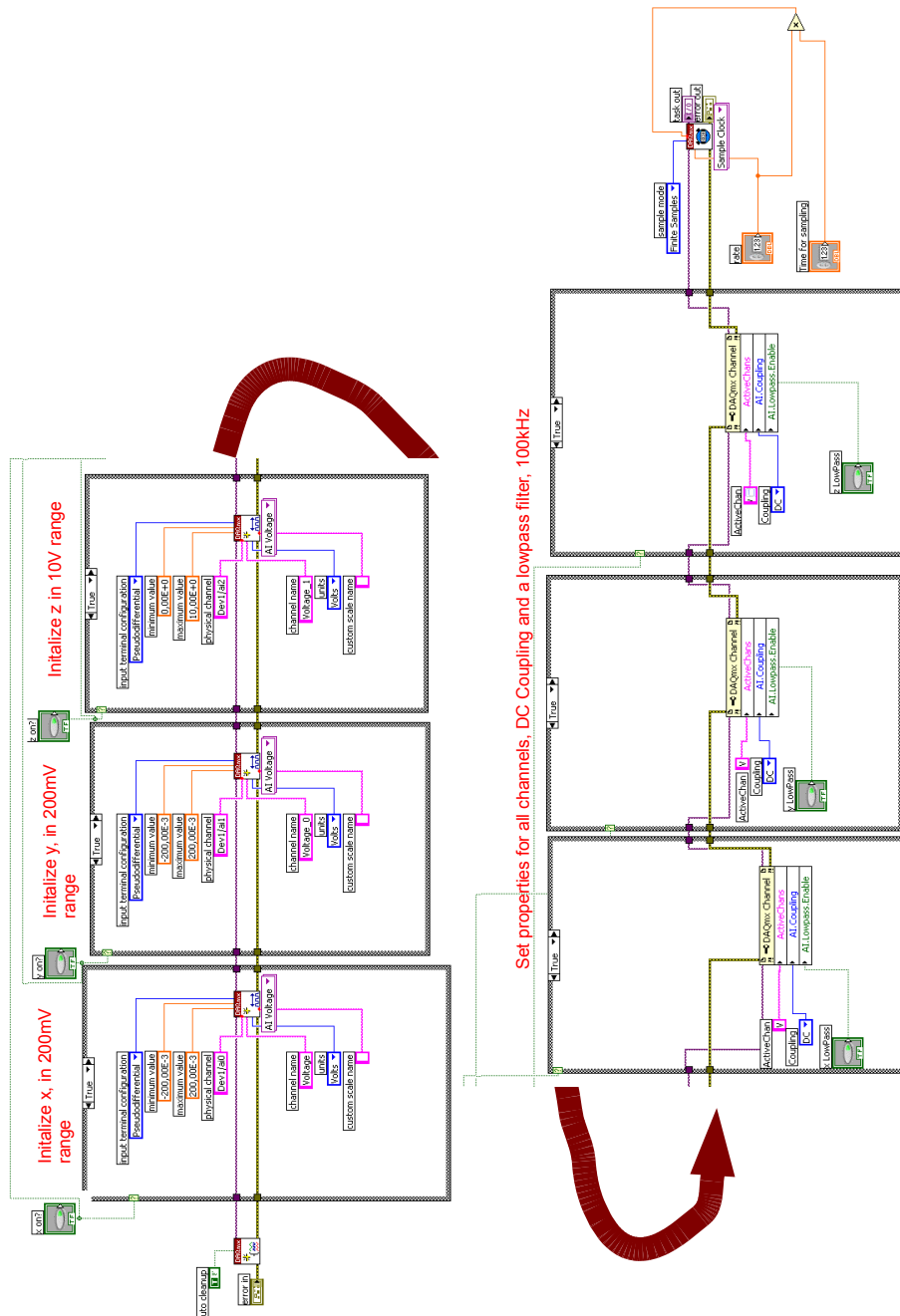


Figure 60: Labview code for initialising the DAQ

A.5 Matlab code

A.5.1 Code for finding physical properties of system

```

function doanalysis(E,fname);
% E is an array with three collums
%      1: Frequency
%      2: Data
%      3: Data with noise removed (for fit)
% fdrive is the driving frequency for the stage
% starttime is the frequency from. which to start the fit
% plot
%      0 Do not plot data
%      1 Plot data
%      2 Plot data on current figure with different colours
% plotfrom
%      Plot from frequency

if nargin < 2
    help doanalysis
    return
end

Info = getfield(importdata(fname,' ', 14),'data');
Version=Info(length(Info));
if Version<3
    fprintf('Wrong file version, only compatible with version 3\n');
    return
end

if size(E,2)>3
    if(size(E,2)<5)
        fprintf('\nInvalid number of collums\n\n');
        return
    end
end

%Variables that should be adjusted
%Defaults from exp
Sfreq=Info(1);
tmsr=1/(E(2,1)-E(1,1)); % As welch will change length
temp=Info(3);
diameter=Info(6)*1e-6;%/1.24
fdrive=Info(7);
amplitude=Info(8);
power1=Info(9);

if power1>1
    power1=power1/1000;
end

%Plot defaults
starttime=25;
plotfrom=1;
plot=1;
endtime=10000;
R_cover=60.84e-6; % Distance to cover slip
density=1.96e-3; % Density of bead in kg/m^3
density_liquid=1e-3; % Density of water in kg/m^3
wavelength=628.9e-9; % Laser wavelength

% Constants
kb=1.38062e-23;
radius=diameter/2;
%temp adjusted for beam intensity
realtemp=temp+3.8*(log(2*pi*R_cover/wavelength)-1)*power1;
viscosity=viscosity_calc(temp);
gamma=6*pi*viscosity*radius;

loren=inline('beta(1)./(pi^2*((beta(2)^2+x.^2)))','beta','x');
startvo=find(E(:,1)>starttime,1,'first');
endval=find(E(:,1)>endtime,1,'first');
[coef,r,J]=nlinfit(E(startvo:endval,1),E(startvo:endval,3),loren,[2,150]);
if size(E,1)>3000
    fprintf('\n\nNLINFIT USED!!!\n\n');
else
    options = optimset('TolFun',1e-16);
    [coef,resnorm,r,exitflag,output,lambda,J]=
        lsqcurvefit(loren,[abs(coef(1)),abs(coef(2))],
            E(startvo:endval,1),E(startvo:endval,3),[0,0],[],options);
end

```

```

Dvolt=coef(1);
fc=coef(2);
if plot>0
    startv=find(E(:,1)>plotfrom,1,'first');
    [yfit,delta]=nlpredci(loren,E(startv:end,1),coef,r,J);
    if plot~=2
        loglog(E(startv:end,1),E(startv:end,2),'b',E(startv:end,1),
            yfit,'r',E(startv:end,1),yfit+delta,'r:',E(startv:end,1),
            yfit-delta,'r:');
    else
        loglog(E(startv:end,1),E(startv:end,2),'k',E(startv:end,1),
            (Dvolt./(pi^2*(fc.^2+E(startv:end,1).^2))), 'c');
    end

%Theoretical fit using fc ;
f=E(:,1);
nu=1e-6; % kinematic viscosity m^2/s
mstar= 4/3*radius^2*density+2/3*pi*radius^3*density_liquid;
fm=gamma/(2*pi*mstar);
fnu=nu/(pi*radius^2);
sqrt_fnu=sqrt(abs(f)/fnu);
Reg=1+sqrt_fnu-3*radius/(16*R_cover)+3.*radius/(4.*R_cover).*
    exp(-2*R_cover/radius.*sqrt_fnu).*cos(2*R_cover/radius.*sqrt_fnu);
Img=-sqrt_fnu+3*radius/(4*R_cover)*exp(-2*R_cover/radius.*sqrt_fnu).*
    sin(2*R_cover/radius.*sqrt_fnu);
Ph = (Dvolt/(pi^2)*Reg)./(fc+f.*Img-f.^2/fm).^2+(f.*Reg).^2);
Ph_resp = (Dvolt*amplitude*fdrive*sqrt(Reg.^2+Img.^2)).^2./
    (2*(fc+f.*Img-f.^2/fm).^2+(f.*Reg).^2);
end

%General stuff
S0=Dvolt/fc^2/pi^2; %Using P(F)=D/(pi^2*(Fc^2+F^2) and setting F to zero
Wth=amplitude^2/(2*(1+fc^2/fdrive^2));
fdrivepos=find(E(:,1)==fdrive,1,'first');
peakval=E(fdrivepos,2);
peakval=peakval+E(fdrivepos+1,2);
peakval=peakval+E(fdrivepos-1,2);
Wex=(peakval-3*Dvolt/(fdrive^2+fc^2)/pi^2)/tmsr; %Peak times fdrive
beta=sqrt(Wth/Wex);

%Lorentzian stuff
gamma_loren=kb*realtemp/(pi*pi*S0*fc^2);
gamma_loren_m=kb*realtemp/(pi*pi*S0*fc^2)/beta^2;
viscosity_loren=gamma_loren/6/pi/radius*2;
viscosity_loren_m=gamma_loren/6/pi/radius*2/beta^2;
beta_loren=sqrt(gamma_loren/gamma);
kappa1=2*pi*fc*gamma;
kappa_loren=2*kb*realtemp/(pi*S0*fc);
kappa_loren_m=2*kb*realtemp/(pi*S0*fc)/beta^2;

% Dual calibration stuff

kappa2=2*pi*fc*1.38e-23*realtemp/beta^2/Dvolt;
gammaDC=1.38e-23*realtemp/beta^2/Dvolt;
viscosityDC=gammaDC/(6*pi*radius);

fprintf('\n\n');
fprintf('Tabulated Values:\n');
fprintf('\tTheoretical Viscosity : %20e Pa s ( kg m^1 s^-1)\n',viscosity);
fprintf('\tTheoretical gamma : %20e Ns/m\n',gamma);
fprintf('\tTheoretical D : %20e m^2/s\n',kb*temp/gamma);
fprintf('\n');

fprintf('General:\n');
fprintf('\tPeak height : %20e V^2 at %10g Hz\n',
    peakval,fdrive);
fprintf('\tMeasured temp : %20e K\n',temp);
fprintf('\tTheoretical Local heating : %20e K\n',realtemp-temp);
fprintf('\tAdjusted Temp : %20e K\n',realtemp);
fprintf('\tMeasurement time : %20e s\n',tmsr);
fprintf('\n');

fprintf('Fit parameters:\n');
fprintf('\tFc : %20e Hz\n',fc);
fprintf('\tS0 : %20e V^2/Hz\n',S0);
fprintf('\tBeta : %20e m/V\n',Wth,Wex);
fprintf('\tBeta (from Gamma) : %20e m/V\n',beta_loren);
fprintf('\n');

fprintf('Lorentzian Analysis:\n');
fprintf('\tKappa using Gamma Theoretical : %20e N/m %20e pN/nm,+
    %20e pN/um\n',kappa1,kappa1*1e12/1e9,kappa1*1e12/1e6);

```

```

fprintf('\tGamma(exp)                : %20e Js/V^2      %20e Ns/m\n',
        gamma_loren, gamma_loren_m);
fprintf('\tViscosity (w Gamma(exp))   : %20e Pa s      %20e N/(V^2 Hz)\n',
        viscosity_loren_m, viscosity_loren);
fprintf('\tKappa (w Gamma(exp))       : %20e N/m        %20e J /V^2      '+
        '%20e pN/nm\n', kappa_loren_m, kappa_loren, kappa_loren_m*1e12/1e9);
fprintf('\n');

fprintf('Dual calibration\n');
fprintf('\tDvoltage                    : %20e V^2/s\n', Dvoltage);
fprintf('\tDamping                    : %20e m^2/s\n', Ddamping*beta^2);
fprintf('\tKappa                      : %20e N/m,        %20e pN/nm      '+
        '%20e pN/um\n', kappa2, kappa2*1e12/1e9, kappa2*1e12/1e6);
fprintf('\tGamma                    : %20e Ns/m\n', gammaDC);
fprintf('\tViscosity                    : %20e Pa s (%g %g)\n', viscosityDC,
        (viscosityDC)/viscosity*100);
fprintf('\n');

```

A.5.2 Code for importing Labview HWS data

```

function data=importHWS( basefilename, sfreq, time )

if nargin <3
    fprintf('Usage: importHWS(basefilename, sampling freq, time)\n');
    return
end

fprintf('Generating Time axis\n');
dsize=sfreq*time;
timeincrement=1/sfreq;
data(:,1)=1:dsize;
data(:,1)=data(:,1)*timeincrement;
data(:,2)=hdf5read([basefilename '_x.hws'],
    '/wfm_group0/axes/axis1/data_vector/data');
fprintf('Read X data\n');
data(:,3)=hdf5read([basefilename '_y.hws'],
    '/wfm_group0/axes/axis1/data_vector/data');
fprintf('Read Y data\n');
data(:,4)=hdf5read([basefilename '_z.hws'],
    '/wfm_group0/axes/axis1/data_vector/data');
fprintf('Read Z data\n');

fprintf('Done\n');

```

B Micelle preparation

Initial ideas for this project involved making measurements on sponge like micells; unfortunately due to time constraints this was not possible. The following sections contains instructions on how the micelle solution is made.

B.1 Preparation of Entangled Worm like Micells

The surfactant solution that we will be preparing is a binary mixture made of Cetylpyridinium chloride and sodium salicylate diluted in a 0.5M brine solution to make a 10% solution by weight. The Sodim salicylate (NaSAL) and Cetylpyridinium (CPyCl) chloride will be added so that there is a 1:2 molar ratio.[7] The brine solution added is a 0.5M solution, so as to keep the electrostatic screening length constant over a whole range of mixture concentrations.

B.2 Calculations:

As mentioned above, we want a 2:1 ratio of [CPyCl]:[SAL]. This can be done by calculating the number of moles required of each CPyCl and SAL, essentially working backwards from the final mass. Assuming we want a total mass of M_{tot} and a 10% solution, and the Relative Molecular Mass (RMM) is $RMM_{2CPy,SAL}$ the number of mole of 2CPy+SAL is

$$[2CPy, SAL] = \frac{0.1 * M_{tot}}{RMM_{2CPyCl} + RMM_{NaSAL}} \quad (207)$$

The number of moles of each compound is then:

$$[CPy] = 2 * [2CPyCl, NaSAL] \quad (208)$$

$$[SAL] = [2CPyCl, NaSAL] \quad (209)$$

Since each of these compounds have a one to one ratio of their ions, we can calculate the mass of each compound simply by multiplying by their respective RMM:

$$\begin{aligned} M_{CPyCl} &= [CPy] * RMM_{CPy} \\ &= 2 * \frac{0.1 * M_{tot} * RMM_{CPyCl}}{RMM_{2CPyCl} + RMM_{NaSAL}} \end{aligned} \quad (210)$$

$$\begin{aligned} M_{NaSAL} &= [SAL] * RMM_{SAL} \\ &= \frac{0.1 * M_{tot} * RMM_{NaSAL}}{RMM_{2CPyCl} + RMM_{NaSAL}} \end{aligned} \quad (211)$$

We then add the 0.5M brine so as to make a 10% ratio of surfactant to brine. We already know the number of moles for Cpy and SAL, and can easily calculate their masses:

$$\begin{aligned} M_{CPy} &= 2 * [2CPy, SAL] * RMM_{CPy} \\ &= 2 * \frac{0.1 * M_{tot} * RMM_{CPy}}{RMM_{2CPyCl} + RMM_{NaSAL}} \end{aligned} \quad (212)$$

$$\begin{aligned} M_{SAL} &= [2CPy, SAL] * RMM_{SAL} \\ &= \frac{0.1 * M_{tot} * RMM_{SAL}}{RMM_{2CPyCl} + RMM_{NaSAL}} \end{aligned} \quad (213)$$

To find the mass of salt we find what 90% corresponds to:

$$M_{NaCL_{aq}} = M_{SAL+CPy} * 9; \quad (214)$$

B.2.1 Molecular Masses:

NaSAL has a molecular formula of HOC_6H_4COONa which corresponds to a molecular mass of 160.10u. CPyCl has a molecular formula of $C_{21}H_{38}ClN \cdot H_2O$, which corresponds to a molecular mass of 358.00u. When in solution the SAL and CPy have a molecular mass of 137.11u and 322.55u respectively. M_{tot} will be $137.11 + 2 \cdot 322.55 = 876.13$.

B.2.2 Preparation of the Worm like micells:

To make roughly 90g of micells, using equations Eq. (210) and Eq. (211):

$$M_{CPyCl} = 2 * \frac{0.1 * 100g * 358.00g/mol}{2 * 358.00g/mol + 160.10g/mol} = 8.172g$$

$$M_{NaSAL} = \frac{0.1 * 100g * 160.10g/mol}{2 * 358.00g/mol + 160.10g/mol} = 1.827g$$

Using Eq. (212) and Eq. (213) we can find the total mass of CPy and SAL ions:

$$M_{CPy} = 2 * \frac{0.1 * 100g * 322.55g/mol}{2 * 358.00g/mol + 160.10g/mol} = 7.363g$$

$$M_{NaSAL} = \frac{0.1 * 100g * 137.11g/mol}{2 * 358.00g/mol + 160.10g/mol} = 1.565g$$

Using Eq. (214) this then gives us a mass of brine:

$$M_{NaCl_{aq}} = (M_{CPy} + M_{SAL}) * 9 = (7.363g + 1.565g) * 9 = 80.352g$$

Making a total of 89.28g of the worm like micells

References

- [1] A. J. D. A. J. Cox and J. Linden. An experiment to measure mie and rayleigh total scattering cross sections. *The American Journal of Physics*, 70:620–625, 2002.
- [2] A. Ashkin. The pressure of light. *Scientific American*, 226:63–71, 1972.
- [3] A. Ashkin. Forces of a single-beam gradient laser trap on a dielectric sphere in the ray optics regime. *Journal of Biophysics*, 61:569–582, 1992.
- [4] J. P. Barton and D. R. Alexander. Fifth-order corrected electromagnetic field components for a fundamental gaussian beam. *Journal of Applied Physics*, 66(7):2800–2802, 1989.
- [5] J. P. Barton, D. R. Alexander, and S. A. Schaub. Internal and near-surface electromagnetic fields for a spherical particle irradiated by a focused laser beam. *Journal of Applied Physics*, 64(4):1632–1639, 1988.
- [6] J. Bechhoefer and S. Wilson. Faster, cheaper, safer optical tweezers for the undergraduate laboratory. *American Journal of Physics*, 70:393–400, 2002.
- [7] J.-F. Berret, J. Appell, and G. Porte. Linear rheology of entangled worm-like micelles. *Langmuir*, 9:2851–2854, 1993.
- [8] S. M. Block. *Cells*, volume 2. Cold Spring Harbour Laboratory Press, 1994.
- [9] E. Butkov. *Mathematical Physics*. Addison-Wesley Publishing Company, 1968.
- [10] R. Castberg. Computer code used in project, 2008. http://rene.castberg.org/OT_code.zip and http://folk.uio.no/renec/OT_code.zip.
- [11] C. F. S. Frederick Gittes. *Signals and Noise in Micro mechanical Measurements*. Academic Press, 1998. Methods in cell Biology, Vol 55.
- [12] F. J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66:51–83, 1978.
- [13] J. A. Käs. Optical forces. website, 2007. http://www.uni-leipzig.de/~pwm/kas/modul_opticalforces/opticalforces.html.

- [14] H. F. Kirstine Berg-Sørensen. Power spectrum analysis for optical tweezers. *Review of Scientific Instruments*, 75:594–612, 2003.
- [15] Leica microsystems. *Leica HCS Objectives, Version 5.3*. 2006. <http://www.leica-microsystems.com/Leica-Microsystems>.
- [16] LEXEL LASER. *Model 95 ION LASER Manual*.
- [17] D. R. Lide, editor. *CRC Handbook of Chemistry and Physics*. CRC Press, 84th edition edition, 2003. pages 8 & 77.
- [18] K. G. Måløy. Lecture notes; experimental physics, FYS4420, 2006.
- [19] National Instruments. *S Series Multifunction DAQ 12 or 16-bit, 1 to 10MS/s, 4 Analog Inputs*. <http://sine.ni.com/nips/cds/view/p/lang/en/nid/11943>National Instruments Website.
- [20] National Instruments. *Aliasing and Sampling at Frequencies above the Nyquist Frequency*, 2008. <http://zone.ni.com/devzone/cda/tut/p/id/3000>.
- [21] Microscopyu, the source for microscopy education. Website. <http://www.microscopyu.com/Microscopyu.com>.
- [22] Pacific Silicon Sensor Inc. *Quad photo diode sum and difference amplifier specifications, QP50-6SD2*. <http://www.pacific-sensor.com/pdf/QP506SD2.pdf>QP50-5SD2 Data sheet.
- [23] Physik Instrumente (PI) GmbH & Co. *P-517 Data sheets*. <http://www.physikinstrumente.com>PI-Website.
- [24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.
- [25] N. H. R. Kubo, M. Toda. *Statistical Physics II*. Springer-Verlag, Berline, 1985.
- [26] K. Rottmann. *Matematisk Formelsamling*. Spektrum forlag, 2006.
- [27] S. F. Tolic-Norrelykke, E. Schaeffer, J. Howard, F. S. Pavone, F. Juelicher, and H. Flyvbjerg. Calibration of optical tweezers with positional detection in the back-focal-plane. *Review of Scientific Instruments*, 2006.

- [28] Wikipedia. Optical tweezers. http://en.wikipedia.org/wiki/Optical_tweezers, 2008. [Online; accessed 12-July-2008].

A huge thanks goes to:

My supervisor Knut Jørgen Måløy.
Grunde Løvoll, Henning Knudsen and Bjørnar Sandnes

For their advice and taking the time to proof read this thesis.

Gunnar, Louisa and Peter
Castberg

For taking the time to proof read this thesis.

And to the rest of the Complex group for their support and
advice.