

Chem 163, Problem Set 2

Due 9/22/2022, 9 am

September 8, 2022

1 Drift relaxation in solution

- Consider a bacterium, idealized as a sphere of radius $1\ \mu\text{m}$, propelling itself at $1\ \mu\text{m/s}$. At time zero, the bacterium suddenly stops swimming and coasts to a stop, following Newton's Law of motion with the Stokes drag force. Assume the bacterium has the same density as water. What is the time constant for the bacterium to dissipate its initial momentum to the solution?
- How far does it travel before it stops?
- Now assume that the bacterium isn't swimming at all, but is purely driven by Brownian motion. Approximately what is its mean-square velocity along the x-axis? How far does it travel in a given direction before it forgets its initial direction?
- It is challenging for bacteria to swim in a straight line because rotational Brownian motion sends them off course. For a $1\ \mu\text{m}$ approximately spherical bacterium in water, what is its rotational diffusion coefficient (with units)? If the bacterium is swimming at $1\ \mu\text{m/s}$, approximately how far can it go before it "forgets" which way it was headed and is on average 30 degrees off from its initial heading?
- Now consider a typical protein molecule in solution (2.5 nm diameter, density $\rho \approx 1.3\text{g/cm}^3$). What is its momentum relaxation time? Thermal velocity? Brownian hop size? Comment on whether we are

justified in treating the motion as an ideal random walk over the time and distance for this protein to diffuse across a bacterium ($1 \mu\text{m}$).

2 Rotational diffusion times

Most biomolecules have a density of $\rho \approx 1.3\text{g/cm}^3$. Plugging in numbers appropriate for water at room temperature, verify the statement, “The rotational diffusion time in picoseconds is approximately equal to the molecular weight in AMU.”

3 Violation of the Central Limit Theorem?

In class we asserted that if you convolve a probability distribution with itself enough times, you usually get a Gaussian distribution. In this problem we will see an example where this is not true.

- First let's check that a Gaussian distribution is a fixed point of iterated convolutions. Define

$$G_1(x) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{x^2}{2\sigma_1^2}}. \quad (1)$$

and

$$G_2(x) = \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{x^2}{2\sigma_2^2}}. \quad (2)$$

Show that $G_3 = G_1 \otimes G_2$ is a Gaussian. What is the variance σ_3 of G_3 ?

- In electronically excited atoms or molecules, radiation damping gives a spectral lineshape with a Lorentzian profile:

$$L_1(x) = \frac{l_1}{\pi} \cdot \frac{1}{x^2 + l_1^2}, \quad (3)$$

where $x = \lambda - \lambda_0$ and λ is the wavelength and λ_0 is the peak wavelength. Check that $L_1(x)$ is normalized, i.e. that $\int_{-\infty}^{\infty} L_1(x) dx = 1$.

Another mechanism of broadening the spectrum is called pressure broadening, which can also lead to a Lorentzian lineshape:

$$L_2(x) = \frac{l_2}{\pi} \cdot \frac{1}{x^2 + l_2^2}. \quad (4)$$

When two line-broadening mechanisms are active, the full spectrum is the convolution of the two individual spectra. Show that

$$L_3 = L_1 \otimes L_2 \tag{5}$$

is a Lorentzian. No amount of convolving Lorentzians with Lorentzians will ever give you a Gaussian! Why isn't this a violation of the Central Limit Theorem?

4 Analyzing a Brownian trajectory

The data for this problem consists of a velocity trajectory, $v(t)$, for a free Brownian particle at room temperature (300 K). The time-step is $20\mu\text{s}$ and the velocity is expressed in units of cm/s. Write a program to calculate the autocorrelation function $\langle v(t + \tau)v(t) \rangle$, and keep the autocorrelation for τ in the range $[0, 2\text{ms}]$. As in the previous problem, the `xcov` command will be helpful. The discussion of the velocity autocorrelation of an overdamped Brownian particle, in Appendix B of Lecture Notes 4 could be helpful.

- Show a plot of the velocity autocorrelation function for your trajectory. Label your axes with the correct units. Try plotting the function on a log-y scale using the function `semilogy`.
- What is the decay constant of your autocorrelation function?
- What is the mass of the particle? What is γ , the frictional constant felt by the particle?
- Include your annotated code for calculating the autocorrelation.

5 Brownian motion in a harmonic well

The data for this problem consists of a position trajectory, $x(t)$, of a spherical particle undergoing overdamped Brownian motion in a harmonic well in water at room temperature (300K). The positions (in nm) of the spherical particle are measured with a time-step of 3 ms. You should re-use your code from problem 3.

- Show the position autocorrelation function of your trajectory. Again, label your axes.
- What is the decay constant of your correlation? How does it compare to that in 3(b)?
- Determine the radius of the particle.
- Determine the spring constant of the trap.

6 Analyzing FCS Data

In this problem, you will be given real data from a recent fluorescence correlation spectroscopy (FCS) experiment and asked to analyze it using MATLAB. The data files are available on the course web site and are formatted as text. Each file contains a list of photon counts that have been binned in 100 μs intervals. One data set was taken using fluorescent polystyrene beads, and the other was taken using fluorescently labeled single-stranded DNA. The data was taken by focusing a laser beam into a droplet on a coverslip in a microscope, and recording the number of photons emitted by the sample during each 100 μs period.

FCS normally works by fitting the autocorrelation function of intensity data of the type provided here. An autocorrelation function is defined to be

$$G(\tau) = \frac{\langle \delta I(t) \delta I(t + \tau) \rangle}{\langle I(t) \rangle^2} \quad (6)$$

where $\delta I(t) \equiv I(t) - \langle I(t) \rangle$ represents deviations from the mean intensity.

For ordinary 3D diffusion, the equation for the autocorrelation function works out to be:

$$G(\tau) = G_0 \frac{1}{(1 + (\tau/\tau_D))(1 + a^{-2}(\tau/\tau_D))^{1/2}} + G_\infty \quad (7)$$

In this equation, τ_D is the residence time of the particle within the illuminated volume; a is the ratio of the axial length of the illuminated volume to the lateral length; and $G_0 = \langle N \rangle^{-1}$, where N is the average number of particles in the illuminated volume. The parameter τ_D can be used to determine the diffusion coefficient of the particle using the formula $D = R^2/4\tau_D$, where R is the radius of the illuminated area in the lateral plane. For a spherical

particle of radius r in a fluid of viscosity η at temperature T , the Einstein relation states that the diffusion constant follows the formula:

$$D = \frac{k_B T}{6\pi\eta r} \quad (8)$$

Thus, it is possible to use FCS data to determine a particle's diffusion constant, and from that, its radius (if it is spherical). This is what we will do in this exercise.

Create a script to perform the following tasks:

- Load the data into MATLAB. Use the `load` command; see the help file for assistance.
- Compute the autocorrelation function of the data from each file. The easiest way to do this is to use the `xcov` command; refer to the help file for more information. Set the maximum lag to be one second (remember that each data point represents a bin of length $100 \mu\text{s}$), and use `'unbiased'` scaling. Store both the covariance output and the lags (which are equivalent to τ in the formulas above), and rescale the lags so they are in microseconds. Keep in mind that the `xcov` command does not divide by the square of the mean intensity; you will need to do this yourself on another line. `xcov` generates values for both negative and positive τ ; these are redundant, so you should resize the output (both the covariance and the lags) to only include values for $\tau \geq 0$.
- Fit the two autocorrelation functions you calculate to the formula above. Fitting using MATLAB is somewhat difficult, so I will guide you through the process. You will first need to create a `fittype` object using the `fittype` command (again, consult the help file). For example, if I wanted to fit data to a Gaussian distribution, I would create a `fittype` object as follows:

```
gaussfittype = fittype('1/(sqrt(2*pi*sigma^2))*exp(-(s - mu)^2-  
/(2*sigma^2))','independent','s','coefficients',{'mu','sigma'});
```

The `'independent'` property sets which variable is treated as the independent variable (in this case, `s`). The `'coefficients'` property sets which variables are treated as coefficients (in this case, `'mu'` and `'sigma'`), and their ordering. For the autocorrelation function above,

the independent variable will be τ and the coefficients will be G_0 , G_∞ , a , and τ_D .

To fit a series of data, use the `fit` command. To continue with the example above, let's say I want to perform a Gaussian fit to a series of points with coordinates in the vectors `X` and `Y`. The line I would use might look as follows:

```
gaussfit = fit(X, Y, gaussfittype, 'Startpoint', [0, 5], 'Lower', [-5, 0], 'Upper', [5, 50]);
```

Let's look at the parts of this statement one by one. `X` is the x-coordinates of the data you are fitting. `Y` is the y-coordinates. `gaussfittype` is the `fittype` object created earlier, and sets the function to which the data are being fit. `'Startpoint', [0, 5]` sets the starting values of the two coefficients of the fit; in this case, I am setting the initial values `mu = 0` and `sigma = 5`. `'Lower', [-5, 0]` and `'Upper', [5, 50]` set the lower and upper bounds for the variables being fit; in this case, I am setting the restrictions $-5 \leq \mu \leq 5$ and $0 \leq \sigma \leq 50$. For each of these properties (`'Startpoint'`, `'Upper'`, and `'Lower'`), the ordering of the coefficients matches the order in which they were listed when the `fittype` object was first defined.

For your FCS fits, do not fit the zero-lag point ($\tau = 0$), as it contains additional noise that will confound the fitting routine. Assuming you have resized the covariance output as above, this means you should only fit from the second point onwards. Set parameter restrictions as follows: $0 \leq G_0 \leq 2$, $-0.1 \leq G_\infty \leq 0.1$, $2 \leq a \leq 6$, and $0 \leq \tau_D \leq 10^6$. Set starting values to: $G_0 = 0.1$, $G_\infty = 0$, $a = 3$, and $\tau_D = 1$.

- Generate plots of the autocorrelation functions and their fits. To continue the Gaussian example from above, to plot the original curve and my fit to it, I would use a line such as `plot(X, Y, X, gaussfit(X))`. The statement `gaussfit(X)` uses the fit stored in `gaussfit` to generate y-coordinates corresponding to the x-values contained in the vector `X`.
- Use the values you obtain for τ_D to estimate the diffusion constants of the beads and DNA using the formula above. To do this, you will need to directly access the final value of the parameter τ_D from your fits. This can be done by typing the name of the fit, followed by a period and the parameter name. For instance, if I wanted to access the mean

of the Gaussian fit of above, I would type `gaussfit.mu`. Substitute your value of τ_D into the formula above for the diffusion coefficient, estimating the lateral radius to be $1.5 \mu\text{m}$. Don't forget to keep track of your units!

- Finally, use the diffusion coefficients you calculate to estimate the radius of the beads and of the DNA (if it were assumed to be spherical). The experiments were performed in water at 25°C ; the viscosity of water at that temperature is $\eta = 8.90 \times 10^{-4} \text{ Pa s}$.