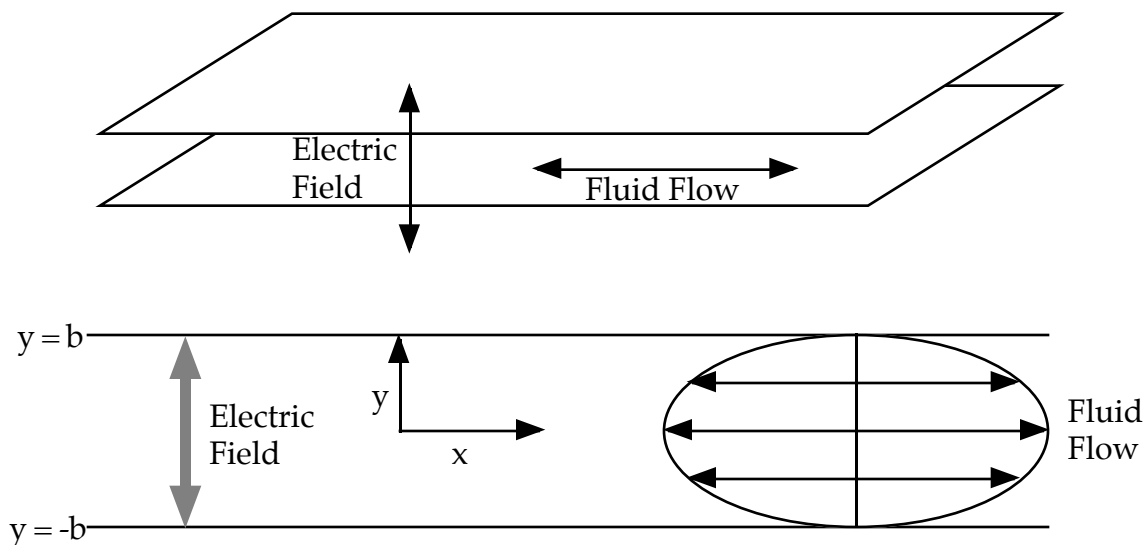


Remember: You may not discuss this with others!

### Optimization of a Binary Oscillatory Crossflow Electrophoresis separation

A key problem in the application of biotechnology is the isolation and purification of biologically active molecules in large quantities. In our laboratory we have developed a technique for separating proteins based on their electrophoretic mobility - the velocity with which proteins move when placed in an electric field - that holds great promise in improving bioseparations. In this project you will determine optimum parameters for using this technique for separating the proteins Bovine Serum Albumen (BSA) and Bovine Hemoglobin (BHb). The technique works as follows:

Consider a thin gap separating two membranes, and a solution of proteins filling this gap as depicted below. If we apply an oscillatory electric field across the gap (taken to be the y-direction), then the protein molecules will migrate up and down with an amplitude of motion proportional to their electrophoretic mobility. Thus, the time history of the solute molecules' position in the gap will be different depending on their mobilities. We may induce a separation between the molecules in the x-direction by inducing a periodic cross-flow. Because the fluid velocity is zero at the walls and a maximum in the center, molecules which have a different time history of location in the y direction will have a different average velocity in the x direction. In particular, if molecules are near the center when the flow is moving in the positive direction and near the walls during the return flow, they will on average move in the + x direction. If they are near the walls during the positive flow and in the center during the return flow, however, they will move in the - x direction. This is the basis for the separation process.



To determine the optimum operating conditions, we must first determine the average velocity of the proteins as a function of these operating parameters. This involves solving the following pair of differential equations:

$$\frac{dy}{dt} = E_0 \mu \sin(\omega t); \quad -b < y < b$$

$$\frac{dx}{dt} = \Delta x \omega \cos(2\omega t + \phi) \left(1 - \frac{y^2}{b^2}\right)$$

The parameter  $E_0$  is the amplitude of the electric field, the parameter  $\mu$  is the electrophoretic mobility,  $\omega$  is the frequency in radians/sec,  $\Delta x$  is the amplitude of the tidal oscillation (the distance the fluid swishes back and forth in the channel),  $\phi$  is the phase lag between the fluid motion and electric field, and  $b$  is the channel half-width.

The solution to these equations is complicated by the presence of the walls and by diffusion. We can simulate the effect of diffusion by simply adding a random displacement after each time step to the position of a solute molecule. The amplitude of this random displacement is given by:

$$\Delta y = (2 D \Delta t)^{1/2}$$

where  $D$  is the molecular diffusion coefficient, and  $\Delta t$  is the time step. You should multiply this amplitude by the output of the "randn" random number generator - it generates a random number with zero mean and a standard deviation of one. The presence of the walls is easily dealt with by just taking all values of  $y$  outside the boundaries at each time step and shifting them back to the nearest wall. If your time step is fairly small, this will be a good approximation.

That's pretty much it: I want you to determine the optimum frequency and phase shift (2 parameters) for separating BSA and BHb using this technique. As a criterion for determining this optimum, I want you to calculate the average velocity of BSA and BHb (averaged over one or a number of periods) and then minimize the product - negative values of the product will insure that the two proteins are going in opposite directions! The parameters which are fixed experimentally are given below:

#### Parameter Values

$$b = 0.05 \text{ cm}$$

$$E_0 = 50 \text{ V/cm}$$

$$\Delta x = 1 \text{ cm}$$

$$D = 6.5 \times 10^{-7} \text{ cm}^2/\text{s}$$

$$\mu_{\text{BSA}} = 0.625 \times 10^{-4} \text{ cm}^2/(\text{V s})$$

$$\mu_{\text{BHb}} = 1.75 \times 10^{-4} \text{ cm}^2/(\text{V s})$$

## Solution Approach:

While you can solve this problem any way you want, it makes a lot of sense to do it a bit at a time. First, work out how to integrate the differential equations, including the effect of diffusion. The trapezoidal rule with fixed step size works well here, and an adaptive integration scheme is pretty much useless. Start the problem off with  $n$  particles distributed randomly in the  $y$  direction (the `rand` command is useful here), and then plot up  $y$  vs.  $x$  for all of them (using 'o' s) so you can get a feel for how the electric field and flow field interact. Use the "drawnow" command so that you can see the positions evolve in time. Turn this graphics off (it slows the integration down), and then probe the behavior by examining the velocity product for different values of  $\omega$  and  $\phi$  in the appropriate range. After you get this to work pretty well (and your code is fairly fast to run!), probe the parameter space by determining the velocity product for an array of values of  $\omega$  and  $\phi$  (this produces a matrix of values of the velocity product). Generate a contour plot which shows the velocity product -- a chunk of code like:

```
cs = contour(omega,phi,velprod,10);
clabel(cs)
```

is useful, with the "clabel" command producing labels on the 10 contours. Type in "help contour" if you are unsure of its use. Finally, once you have a good idea of the optimum operating parameters, use an optimization routine like `fmins` to determine the precise optimum. Finish the problem off by plotting the average velocity as a function of electrophoretic mobility over the range  $[0, 3e-4]$   $\text{cm}^2/(\text{V sec})$ , marking the points corresponding to BSA and BHp.

## What to turn in:

I want you to turn in your source code and your output, with the optimum parameter values and corresponding velocities clearly marked. I also want a contour plot showing the velocity product for phase shifts in the range  $-\pi/2 < \phi < \pi/2$  and frequencies in the range  $0.02 < \omega < 0.2$  radians/sec. Finally, I want the velocity plot at the optimum parameter values as described above.

## Hints:

A few pointers that you might find useful:

- You need to have enough points and periods of oscillation to get good statistical accuracy, but not so many that it takes a long time to calculate a velocity. Remember that your contour plot and your optimization routine will have to calculate the velocity a whole lot of times!

- I found it convenient to write a function which calculates the average velocity as a function of electrophoretic mobility, and in which the parameter values  $\omega$  and  $\phi$  are passed in as global parameters.
- You can then write another function which uses this routine to determine the velocity product as a function of  $\omega$  and  $\phi$ , and which can be minimized using `fmins`.
- You can avoid the use of "if" statements by using the matlab "min" and "max" commands in an intelligent way for each of the two walls (e.g., comparing the y positions of the n molecules to a vector of -b's or +b's and keeping the appropriate minimum or maximum).
- The trapezoidal rule is easy to implement for this pair of first order equations - the equation for y is not even a function of y, so it is easy to use TR for it, and the same is true for x (just do y first).
- Be sure to integrate over an integral number of cycles at a time to get an accurate estimate of the average velocity! Fixed time steps make this easy to do.
- The average velocity over one period is just the x displacement over that period divided by the length of the period.
- Make sure your program is as parallel as possible - write the routine so it operates on all n particles simultaneously, with vectors x and y describing the position of all of the particles!
- Start the particles off randomly distributed in the y direction using the `rand` command, and then wait long enough for the y distribution to approach steady-state before calculating the average velocity. It won't take very long at all for frequencies low enough that the molecules impact the walls.
- In your final version, the velocity function should automatically determine how long to wait before the concentration distribution in the y-direction approaches steady-state. A convenient indicator is when the change in the average velocity between integrating over one cycle and the next is of the same order as the random error in the average velocity (it will never be much less than this value!). To get you started, however, you might want to begin by just assuming that the velocity reaches steady-state after one cycle (this will be correct for low frequencies, but not for high ones) and build in the corrective part later.
- After reaching steady-state, you can throw the velocities up to this point away, and then "do it for real" by calculating it again over some selected number of cycles. Again, the desired number of cycles depends on the random error in the velocity and the allowable error. For this problem, doubling the number of cycles of integration is roughly equivalent to doubling the number of molecules you are tracking, so the same error calculations in monte carlo integration used in your last homework apply.

## Final Project Evaluation Criteria:

The final project will be evaluated according to the following criteria:

50 points: Did you get the correct answer, or how close did you get?

20 points: Quality of integration algorithm. Does your program determine the velocity of all of the molecules simultaneously, or does it do one at a time? Do you have an appropriate balance between the accuracy of the integration process (in view of the randomness induced by diffusion) and the accuracy demanded by the function minimization routine? Is the number of molecules (number of cycles) you are tracking reasonable, and is it determined by the program? Is the time to approach steady-state properly accounted for and automatically determined? This is not an exclusive list, just some of the ideas to consider in writing a high quality program.

20 points: Clarity of programming. Did you use the global command in a restrained manner (remember that the global command is a meat axe - while it is extremely useful, it should also only be used when it significantly simplifies the program)? Is your program easy for us to follow? Is it well commented? Often you will have to pass your programs on to someone continuing your project in the future. If it is not clear and well commented, a program is useless to others and loses much of its value.

10 points: Clarity of presentation of graphical results (e.g., labelled axes & graphs, legend, title, etc.)

I hope that these criteria will make it easier for you to prepare the final versions of you program. Remember, do not discuss this with classmates until after 6:00PM on Thursday! You may talk to me or Mike King, however. Good Luck!

PS: Stop by next fall to see if the optimum values calculated here actually worked!