File: /General/MLAB-Text/Papers/centrifuge/centshort.tex

Ultracentrifuge Data Analysis

Gary D. Knott, Ph.D.
Civilized Software, Inc.
12109 Heritage Park Circle
Silver Spring, MD 20906 USA
Tel. (301) 962-3711
email: csi@civilized.com

One common application of curve-fitting occurs in the analysis of ultracentrifuge data. Suppose we load the centrifuge cell (a wedge-like container with circular-profile surfaces at each end) of an ultracentrifuge with a solution of a particular substance or substances, and then spin the cell until the molecules in solution distribute themselves in the cell according to the forces that act upon them. This equilibrium arrangement will have the complex solute distributed in the cell in an exponential fashion with relatively more of the complex solute at the bottom (which is the outer end) of the cell. Therefore the concentration of the complex solute varies at different positions along the cell (*i.e.*, at different radial positions or distances from the center of the ultracentrifuge.)

Let us denote a given radial position by r, measured in centimeters from the center of rotation, where the cell ranges from the top position at r_m to the bottom position at the outer end of the cell at r_b. Then the concentration of a single-substance solute, c, measured in grams per liter, at the radial position r is a function of r given as:

$$c(r) = c_h e^{AN(r^2 - r_h^2)},$$

where

c_h is the concentration in grams per liter of the solute at a given reference radial position, r_h. The value c_h can be computed from N and c_0, the total amount of solute in the cell measured in grams per liter, however, in practice, c_h can be determined by curve-fitting.

r_h is the given radial reference position (it is not necessary that r_h be r_b or r_m, indeed r_b is generally a poor choice).

N is the apparent molecular weight of the solute.

A is a certain constant, defined as: $(1-\bar{v}\rho)w^2/(2RT)$, where $\bar{v}$ is the partial specific volume of the solute ($1/\bar{v}$ is the density of the anhydrous solute, so $\bar{v}$ is measured in cm^3/gram), ρ is the solution density in grams/cm^3, w is the angular velocity of the rotor in radians per second, R is the gas constant ($\approx 8.314 \times 10^7$ ergs/degree/mole), and T is the absolute temperature.

Due to the non-ideal diffusion behavior of most materials, the observable or apparent molecular weight N, at a given radial position r, is approximately the following non-linear function of the actual molecular weight, $M : N \approx M/(1 + BMc(r))$, where the parameter B is the so-called virial coefficient, with $B \geq 0$.

Thus N is a function of r, and we have:

$$\begin{aligned} c(r) &= c_h \exp(AM(r^2 - r_h^2)/(1 + BMc(r))), \quad \text{or} \\ c(r) &= root(x, 0, c_{\max}, (c_h \exp(AM(r^2 - r_h^2)/(1 + MBx)) - x). \end{aligned}$$

The notation $root(x, a, b, f(x))$ denotes a value, x_0, in the interval $[a, b]$ which is a root of the expression $f(x)$, *i.e.*, x_0 is a value such that $f(x_0) = 0$. MLAB provides the root operator as a built-in function `ROOT`. The value $c_{\max}$ is any value greater than $c(r_b)$, which is an upper bound for $c(r)$.

If we have several solutes with unknown molecular weights, then the equation for the combined concentration distribution is merely the sum of the separate concentration distributions.

In general, we have: $c(r) = c_1(r) + c_2(r) + \ldots + c_n(r)$, where the gram per liter concentration of the ith solute is $c_i(r) = c_{hi} \exp(AN_i(r^2 - r_h^2))$, N_i is the apparent molecular weight of the ith solute, and c_{hi} is the grams per liter concentration of the ith solute at the reference position r_h, and A_i is the usual constant $(1-\bar{v}\rho)w^2/(2RT)$ described above, particularized for the ith solute.

As before, N_i is a function of r and the true molecular weight, M_i, namely: $N_i \approx M_i/(1 + B_i M_i c(r))$. In fact this is an approximation to the relationship:

$$N_i = M_i/(1 + \sum_{k=1}^{n} \sum_{j=1}^{\infty} B_{ikj}(M_i c_k(r))^j)$$

Usually assuming each value B_{ik1} is the same, namely B_i/n, for $1 \leq k \leq n$ and $B_{ikj} = 0$ for $1 < j$ is adequate, and this results in the simple form used above.

It is often of interest to compute the amounts of each component present in the ultracentrifuge cell directly from the gram-per-liter concentration profile function. This can be done by using the equation:

$$q_i = \int_{r_m}^{r_b} c_i(r) v dr,$$

where q_i is the total mass in grams of the given solute i, v is a constant based on the geometry of the cell, r_b is the radial position of the bottom of the cell, and r_m is the radial position of the top, or meniscus, of the solution in the cell. For a cell 1.2 cm deep with a chord length of .33 cm at the radial position $r_b = 7.2$ cm while spinning, we have $v = 1.2{\cdot}2{\cdot}\arcsin(.33{\cdot}7.2/2)$. It is important to use the most accurate possible geometry constant v and bottom radial position r_b; when spinning, r_b can be approximately 1% greater than the value measured at rest. Finally note that this computation can be confounded when the solute sticks to the the interior walls of the cell or builds up excessively at the bottom of the cell.

The mathematical modeling software MLAB can be used to compute the equilibrium constant K for the simple binding reaction $X + Y \rightleftharpoons Z$ occurring within the ultracentrifuge cell in many cases.

Let $c_1(r)$ denote the grams-per-liter concentration of the first solute substance X at radial position r, let $c_2(r)$ denote the grams-per-liter concentration of the second solute substance Y at radial position r, and let $c_3(r)$ denote the grams-per-liter concentration of the composite third solute substance Z at radial position r. Then the combined concentration distribution function is $c(r) = c_1(r) + c_2(r) + c_3(r)$, where the grams-per-liter concentration distribution of the ith solute is $c_i(r) = c_{hi} \exp(A_i N_i (r^2 - r_h^2))$ for $i = 1, 2, 3$.

Also, at any fixed radial position r, the stochiometric relationship $K = (\frac{c_3(r)}{M_3})/((\frac{c_1(r)}{M_1})(\frac{c_2(r)}{M_2}))$ holds, where M_i denotes the molecular weight of solute i for $i = 1, 2, 3$. Note that $M_3 = M_1 + M_2$. Thus,

$$c_3(r) = \frac{M_1 + M_2}{M_1 M_2} K c_1(r) c_2(r).$$

Let $g_i(r, x) := c_{hi} \exp(A_i M_i (r^2 - r_h^2)/(1 + B_i M_i x))$ for $i = 1, 2, 3$. Note $g_i(r, c(r)) = c_i(r)$. Now define $f(c_1, c_2) := c_1 + c_2 + K c_1 c_2 (M_1 + M_2)/(M_1 M_2)$. Then, $c(r) = root(x, 0, c_{\max}, f(g_1(r, x), g_2(r, x)) - x)$ where $c_{\max}$ satisfies $c_{\max} > \max_r c(r)$. We thus have an implicit model for the c vs. r data corresponding to the simple binding reaction $X + Y \rightleftharpoons Z$ occurring in the ultracentrifuge cell. The potentially-unknown parameters of this model are c_{1h}, c_{2h}, M_1, M_2, B_1, B_2, A_1, A_2 and K. MLAB curve-fitting may now

in principle be employed to estimate these parameters. In the case of a self-associative dimerization reaction, this model simplifies with $c_{1h} = c_{2h}$, $M_1 = M_2$, $A_1 = A_2$ and $B_1 = B_2$.

Now, if we measure the total concentration distribution with a UV scanner or with a Rayleigh interferometer attached to the ultracentrifuge and express the results as a table of (r, c) points, each of which gives the observed concentration c at the corresponding specified radial position r in the cell, then the root-form equation for $c(r)$ given above can be used as a model to fit these points. In this way, various parameters such as K, M_1, M_2, B_1, or B_2 can, in principle, be determined by curve-fitting. Often c_{h1} and c_{h2} are not exactly known, and are also taken as fitting parameters. Generally, the number of distinct fitting parameters must be minimized, however, in order to obtain physically reliable estimates. Concentration can be measured in any of a number of different units. Fundamentally, we have fringe numbers or optical densities, and such observations are readily convertible by scaling to grams-per-liter concentration units. Note if $K = 0$, we have two non-interacting substances in the cell, and estimating M_1, M_2, B_1, and B_2 becomes feasible, In the example given here, however, we are interested in estimating the equilibrium constant K.

Generally we cannot successfully estimate the equilibrium constant K when K is small (say $< .5 \cdot 10^4$,) because our model tends to become increasingly ill-conditioned as K decreases. The exact limit depends on the molecular weights, the initially-loaded amounts, and the virial coefficients of the reactants.

There are several more "rules of thumb" that we should remember when estimating the equlibrium constant K.

(1) The molecular weights M_1 and M_2 should be predetermined and should not be fitting parameters. Any of a number of methods can be used to determine M_1 and M_2; if sequence composition information is known, of course, it should be used to compute the corresponding molecular weight exactly. Similarly, the virial coefficients B_1 and B_2 should likewise be predetermined, if possible. (Note that once we have obtained M_1, M_2, B_1, B_2, c_{1h}, and c_{2h}, we can subtract $c_1(r) + c_2(r)$ from our data, and then fit g_3 to the result in order to estimate B_3 and c_{3h} if this is desired.)

(2) The value $c_{\max}$ should be as small as possible so as to minimize the occurrence of overflows in the derivatives of c. This implies that relatively dilute solutions of the reactants should be used. On the other hand, we need to produce an adequate amount of Z material to be represented in the $c(r)$-data. The speed of the ultracentrifuge should be chosen to obtain a distribution of mass which is neither too "flat" nor too "sharp", *i.e.*,

material should not "pile-up" at the bottom of the cell. It may sometimes be convenient to discard the data for radial positions near the bottom of the cell in order to be able to reduce the value of c_{max}. It is generally wise to discard the data near the bottom of cell in any event because the error in measuring mass concentration is large near the boundary of the cell.

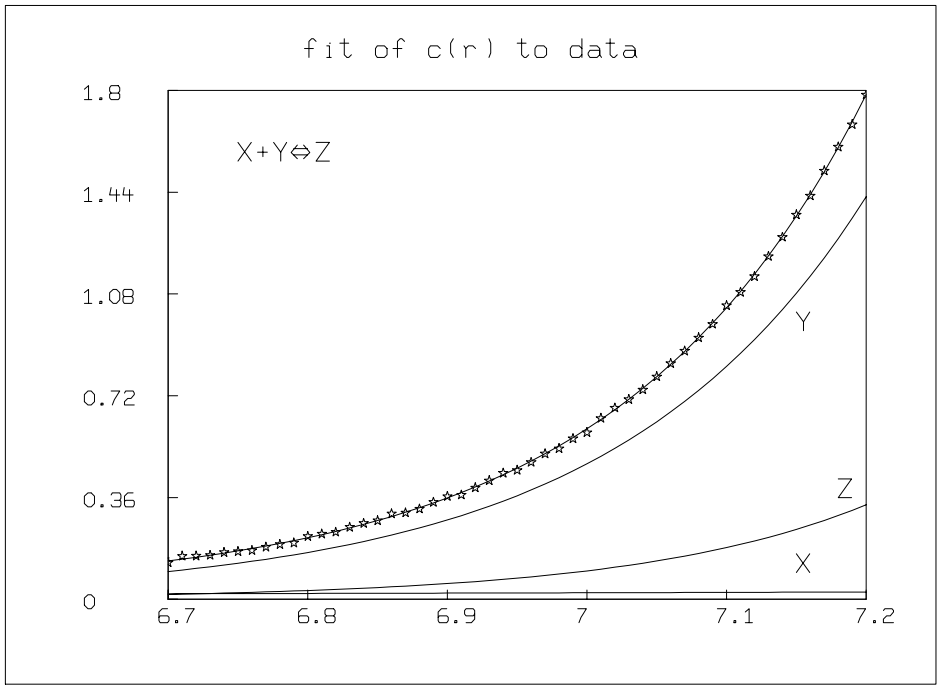
(3) It is generally useful to replace K by $exp(L)$, where $L = log(K)$, and then fit to estimate L. This does not generally provide a more exact estimate of K, but the normal theory standard errors computed for L are more likely to be meaningful than are the same quantities for K when estimated directly.

Here is an example of using MLAB to estimate the equilibrium constant for a simple binding reaction. Note we have also used the conservation-of-mass relations for substance 1 (X) and for substance 2 (Y). Choosing parameters to fit the computed masses of X and Y to the known initially-loaded masses is a valuable device to improve our parameter estimates. This can only be done, however, when accurate values of v and r_b are available. (Another device that can help in obtaining more accurate estimates is to simultaneously fit gram-per-liter concentration-profile data obtained at equilibrium for several different rotor speeds.)

```
* rm = 6.7  /* minimum radius in cm */
* rb = 7.2  /* maximum radius in cm */
* rh = rm   /* reference radius in cm */
* vc = 2*asin(.33/(2*rb))  /*angle of the cell wedge */
* vc=vc*1.2 /*volume constant, total solution vol.= (vc/2)*(rb^2-rm^2) cm^3 */

* r1 = rm:rb:.01 /* radial positions for curve-plotting */

* qv = 0.26 /* this term is 1 - zbar*rho */
* omega = 1047.2 /* corresponding to 10000 rpm machine rotation */
* R = 8.314*10^7 /* gas concentration */
* t = 298.15 /* absolute temperature */
* A1 = (qv*omega^2)/(2*R*t); A2=A1 /*Both A1 and A2 = 5.75119093E-6 */

* /* predetermined values for known parameters */
* B1 = 2*10^(-8); B2=10^(-9);
* M1 = 67000;  M2=6800

* constraints q={9<lk, lk<40, ch1>0, ch2>0, ch1+ch2<1.4}

* fct g1(r,x)=ch1*exp(A1*M1*(r^2-rh^2)/(1+M1*B1*x))
```

```
* fct g2(r,x)=ch2*exp(A2*M2*(r^2-rh^2)/(1+M2*B2*x))
* fct f(c1,c2)=c1+c2+exp(lk)*c1*c2*(M1+M2)/(M1*M2)
* fct c(r) = root(x,(ch1+ch2)/3,100000, f(g1(r,x),g2(r,x))-x)

* fct c1(r)=g1(r,c(r))
* fct c2(r)=g2(r,c(r))
* fct c3(r)=c3h(r,c(r))
* fct c3h(r,x)=x-g1(r,x)-g2(r,x)

* /*tm1, tm2  = total mass in grams of substances 1 and 2 */
* fct tm1()=integral(r,rm,rb,tm1z(r,c(r))*vc*r)
* fct tm1f(c1,c2,c)=c1+(M1/(M1+M2))*(c-c1-c2)
* fct tm1z(r,x) = tm1f(g1(r,x),g2(r,x),x)

* fct tm2()=integral(r,rm,rb,tm2z(r,c(r))*vc*r)
* fct tm2f(c1,c2,c)=c2+(M2/(M1+M2))*(c-c1-c2)
* fct tm2z(r,x) = tm2f(g1(r,x),g2(r,x),x)

* /*specify the total mass in grams of substances 1 and 2 */
* mass1[1]= .113689839;
* mass2[1]= .00613855051

* /* read-in the r vs. c(r) data to be modeled. */
* data = read(data.dat,1000,2)
* n=nrows(data);
*
* /* define the weights for fitting and normalize them to sum to 1 */
* zw=ewt(data); zw=zw/rowsum(zw)

* /*set initial guesses for k, ch1,and ch2 */
* k=10000; lk=log(k)
* ch1=.15;
* ch2=.035

* fit(lk,ch1,ch2), c to data with weight zw, tm1 to mass1, tm2 to mass2,\
: constraints q

 final parameter values
      value               error                dependency    parameter
 10.92859862         0.1533653124            0.9980490348   LK
```

```
  0.09847696449      0.003231742687          0.9986427864   CH1
  0.01982622037      0.001226384415          0.9914474897   CH2
4 iterations
CONVERGED
best weighted sum of squares = 2.386287e-05
weighted root mean square error = 6.908382e-04
weighted deviation fraction = 5.217909e-03
R squared = 9.998792e-01
no active constraints

* exp(lk)  /* k=exp(lk) */
    = 55748.1007
* tm1()    /* estimated total mass in grams of X */
    = .113740813
* tm2()    /* estimated total mass in grams of Y */
    = 6.25476406E-3

* draw data, lt none, pt star, ptsize 0.01
* draw points(c,r1)
* top title "fit of c(r) to data"
* draw points(c1,r1), color red
* draw points(c2,r1), color green
* draw points(c3,r1), color yellow
* title "Y" at (7.15,.95) world
* title "Z" at (7.18,.36) world
* title "X" at (7.15,.095) world
* title "X+Y'2T='RZ" at (6.75,1.55) world
* view
```



The example above demonstrates the type of models involved in the analysis of data obtained from the use of an ultracentrifuge. There are many other useful experiments, with other equations (or differential equations) describing the phenomena involved. MLAB is often useful in analyzing such experiments, and may well succeed where other software fails.