Fitting Stretched Exponentials

April 4, 2008

The stretched exponential or Kohlrausch-Watts-Williams (function) often describes dynamics in heterogeneous systems characterized by a range of rates. It is of the form,

$$I = I_0 e^{-(t/\tau)^\beta}$$

The β parameter is the "stretch" parameter, and it varies between 0 and 1. A β parameter of 1 is just an ordinary exponential decay. So, the lower the β parameter is, the more heterogeneous the dynamics. Therefore β obtained from fits can act as a quantitative measure of heterogeneity. We will attempt to extract β values from our simulations and fits to determine how the heterogeneity in the fluorescence dynamics depends on various parameters such as particle size, exciton diffusion length, Förster radius, and number of dyes. Let's play with the math and see if we can come up with a way to do a straight-line fit that will give us β as the slope. If we divide both sides by I_0, and take the natural log of both sides, we get,

$$\ln\left(\frac{I}{I_0}\right) = -\left(\frac{t}{\tau}\right)^\beta$$

This function is of the form,

$$y = ax^b$$

If we perform a log-log plot of y versus x, we obtain a straight line with a slope of b. If you want a little more detail and example plots, see "log log plot" on Wikipedia. The y value at $x = 1$ gives us the a parameter.

Mapping this onto the KWW form, a plot of $\ln(-\ln I/I_0)$ versus $\ln t$ should yield a straight line of slope β. The intercept should tell us something about τ.

Let's try this out on some simulated KWW-type data in MATLAB. First we set up our time axis, τ, and β:

```
>> t = 0:200;
>> tau = 30;
>> beta = 0.76;
```

Now we make some simulated fluorescence data,

```
>> f=exp(-(t/tau).^beta);
```

Notice we used `.^` since we are raising an array to a power. Now let's do a log-log plot to see if it looks like a straight line,

```
>> loglog(t,-log(f/f(1)))
```

Yes, it looks like a straight line. Recall that in MATLAB `log` means natural (base e) log, while `log10` is the base 10 logarithm. Now we have to do a linear fit in order to extract the β parameter. This is done as follows. First we need to take the log-log of the fluorescence intensity and the log of the time axis. We will place the results in `loglogf` and `logt`.

```
>> loglogf=log(-log(f/f(1)));
>> logt = log(t);
```

Now we fit to a straight line,

```
>> pp=polyfit(logt,loglogf,1)
```

This yielded a bad fit, because the first point of `loglogf` is either `NaN` or `-Inf`. Since the first point is `f(1)/f(1)` which is 1, the log of that is zero. Then we take the log of zero, and get minus infinity. We need to remove the first point and fit the rest:

```
>> loglogf=loglogf(2:end);
>> logt=logt(2:end);
>> pp=polyfit(logt,loglogf,1)
pp =

   0.76000  -2.58491
```

That fixed it! The first parameter, which is the slope, is 0.76, which is the β parameter we started with. The intercept is somehow related to τ, but we won't worry about that for now. We might want to check the fit by plotting:

```
>> plot(logt,loglogf,'o',logt,polyval(pp,logt))
```

I have put all this together in a little m-file `logloglog.m` which hopefully might help us to analyze our simulation results, etc.