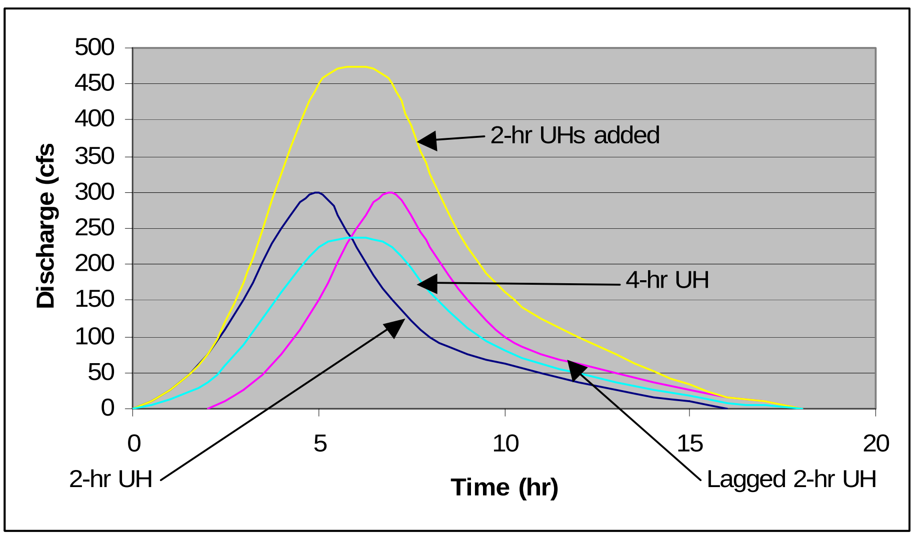
Lecture 21—Screwing around with unit hydrographs

We ended up last time with a fairly simply (almost *contrived*) example of making a unit hydrograph. We left off with a few questions:

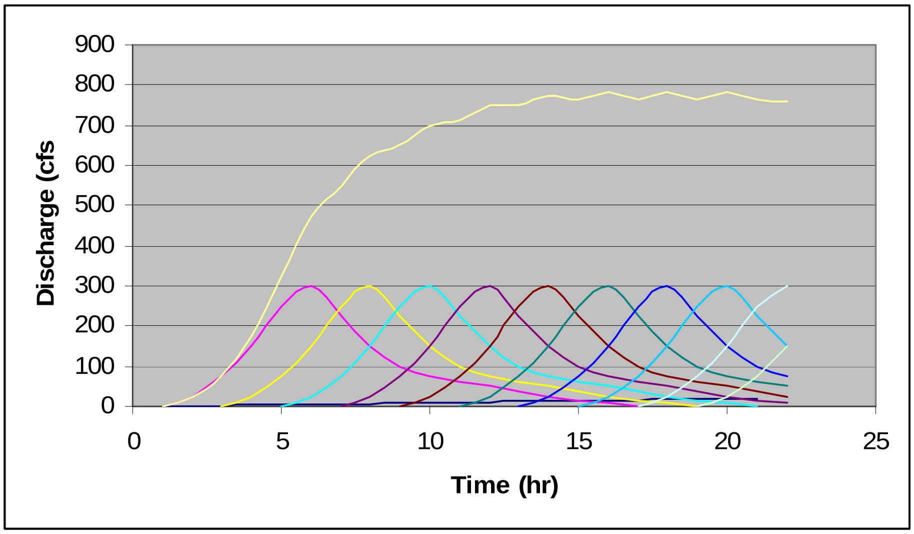
- What if I wanted to make a UH with a time duration different from the rainfall (or not easily computed from the excess rainfall?)
- What if I wanted to use *actual rainfall* instead of the highly contrived version we saw?

The first one is easy, relatively. Turns out that UHs are linear—this means you can add and subtract UHs to your heart's content. Suppose, for example, that you have a 2-hour UH, and you want a 4-hour UH. What you do is take the 2-hour UH, *lag* it by two hours, add the two, and divide by two.

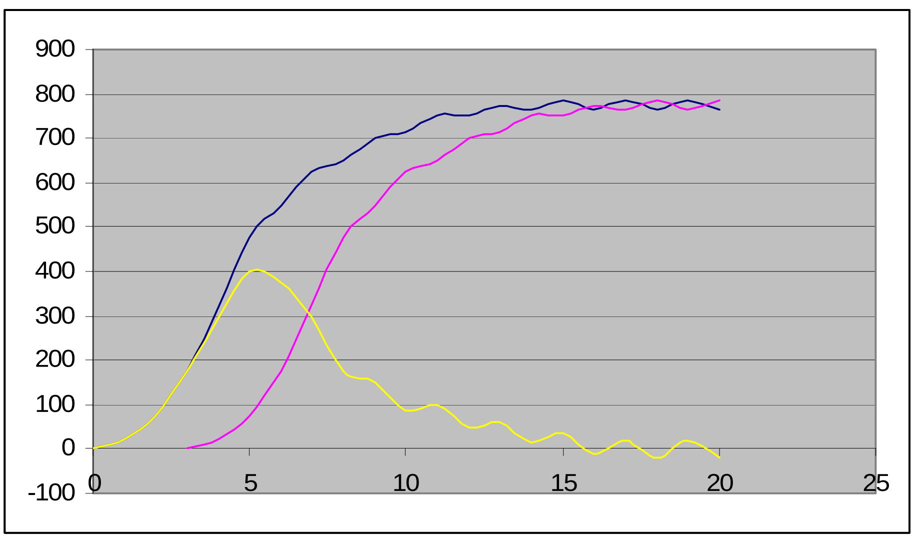



The result is the four-hour UH. Now, there's a problem—you're restricted to integers (you can't start, for example, with the 1.5-hour UH because you'd need to add together 1 and a half of them). It's also a little clumsy. However, there's a method that easily allows making UHs of any duration at your leisure.

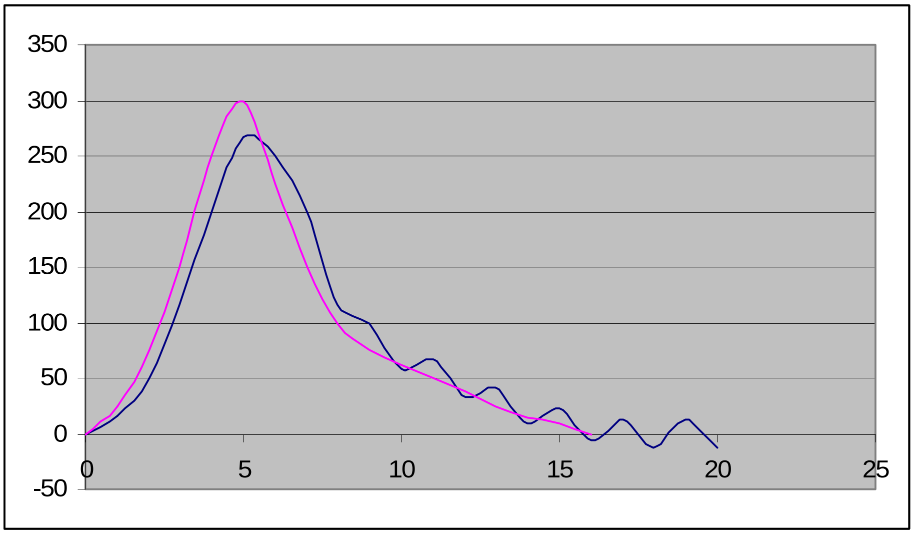
Suppose we took, say, a 2-hour UH, and offset it by 2 hours, and summed it, repeatedly. The result would be an S-shaped curve that approaches equilibrium (this is at least slightly familiar—it's effectively the equilibrium hydrograph for continuous excess rainfall of duration 1/*D* in/hr, where *D* is the duration of rain in the hydrograph). Curiously, we'll call this curve the S-curve, and the method of generating UHs the S-curve method.




From here it's easy. Simply lag the *S-curve* by the duration you wanted (say, 3 hours), and take the difference between the two S-curves.




To get the vertical axis scale right, you have to multiply everything by the ratio of the original duration over the new duration (here 2/3).




Some important thoughts on this:

- It's ideal for doing on a spreadsheet (or in FORTRAN, which is why it was invented)
- You'll notice the new UH tends to have wiggles in it. These are numerical errors caused by truncation, and are commonly smoothed out by hand.

Ok. That wasn't too bad. You may ask yourself, "why did we do this?" and that's a legitimate question. Mostly it's to demonstrate an important fact of UHs. They're linear, so you can add them together. This is about to become important, and will be vital to generation of *synthetic* UHs later.

Up to now, unit hydrographs haven't been terribly satisfying. We're left asking such important questions as, "what are these things good for?" and saying such important statements as "that assumption is crap." Ok, here's the first thing UHs are good for. Suppose we had an excess rainfall hyetograph and a 1-hour UH for the area, and we'd like to know what the actual storm hydrograph might look like. This might become important where rainfall intensity can be known (say from Doppler radar) or a "design rainfall" requested (see, for example, pages 58-63 in your book). The procedure is a *lot* like the time-area method we dealt with two days ago. Basically, if you have the excess rainfall intensity in one hour increments, and the 1-hour UH, you can multiply the 1-hour UH by the rainfall intensity for each time interval, lag each resulting UH by the appropriate amount, and sum. In math:

$$Q_n = \sum_{i=1}^{n} P_i U_{n-i+1} = P_n U_1 + P_{n-1} U_2 + ... + P_1 U_{n-i+1}$$

Where $P_i$ is the rainfall excess at time increment *i* and $U_i$ is the unit hydrograph ordinate at time increment *i*. This process is called *unit hydrograph convolution*, and the equation is the *convolution equation*. It actually looks a whole lot easier on a spreadsheet—all you do is take the UH down one column in 1-hour chunks (and normalize it by the rainfall intensity) then lag each successive UH until the rain stops, then sum.

Ok, so now we have something! If we can make a UH, we can construct (and therefore *predict*) hydrographs based on hyetographs! While these aren't perfect, they can be *calibrated* for watersheds, with a goal of being able to adequately predict flooding. This is important.

So where did that UH come from? Well, one other consequence of linearity is that if you can add them up, you can also subtract them. *Therefore*, if we had an actual storm hydrograph and an excess rainfall hyetograph, we could *produce* the UH from that! This would be called, naturally enough, *deconvolution*. And yes, unfortunately, just as integration is nastier than differentiation, deconvolution is nastier than convolution. Let's suppose we had a 4-hour rainfall that resulted in 7 hours of runoff. *Deconvolution* would result in the following equations for the UH:

$$
\begin{aligned}
Q_1 &= P_1U_1 \\
Q_2 &= P_2U_1 + P_1U_2 \\
Q_3 &= P_3U_1 + P_2U_2 + P_1U_3 \\
Q_4 &= P_4U_1 + P_3U_2 + P_2U_3 + P_1U_4 \\
Q_5 &= P_4U_2 + P_3U_3 + P_2U_4 + P_1U_5 \\
Q_6 &= P_4U_3 + P_3U_4 + P_2U_5 + P_1U_6 \\
Q_7 &= P_4U_4 + P_3U_5 + P_2U_6 + P_1U_7 \\
Q_8 &= P_4U_5 + P_3U_6 + P_2U_7 \\
Q_9 &= P_4U_6 + P_3U_7 \\
Q_{10} &= P_4U_7
\end{aligned}
$$

Where Q is the storm hydrograph, P is the excess rainfall hyetograph, and U is the UH. Ugly! However, you could solve these equations one at a time, right? Note that you wouldn't even have to solve the last three, because you'd already have all seven of the U ordinates.

By the way, we've been dancing around linear algebra here. This would be much simpler if we expressed this in matrix form:

$$[Q] = [P][U]$$

where, in this example, Q is a 1x10 matrix, P a 1x4, and U a 1x7. Suffice it to say that linear algebraic techniques are at play here, and we'll leave it at that. The upshot is that deconvolution is hard, so people try to pick small, well-behaved storms for deconvolution, or avoid the whole mess entirely and stick with *synthetic* unit hydrographs.