In this section we will continue to apply the source receptor approach but in a different application; we will try to figure out the time variation of the release, given a source location that is known. And we're approaching it from a more generic sense, so rather than call this the source receptor matrix, we're calling it the coefficient matrix, which is a term common in linear algebra. And what I mean by that is, you've seen this equation before, in that if we look at each receptor location, that is the sum, the concentration at each receptor location, is the sum of the dilution factors, that is the calculation by the dispersion model, times the source term at that source, the product of those two is the contribution at the receptor from that source. So we go through I sources over J receptors. And the D matrix is called a coefficient matrix. The vector, source vector here, and the receptor vector, here. Now this equation can be rearranged, so that we can solve for the source term, where the source term is the matrix product of the inverse matrix, of the coefficient matrix, times the receptor vector. So source vector equals the inverse of the coefficient matrix, the dilution factors, times the receptor vector, the receptor vector in this case being the measurements. So this is a linear algebra problem and if it's a square matrix, it certainly can be easily solved, if it is not a square matrix, and that is normally the case, other approaches need to be applied, such as singular value decomposition. This is coded in the HYSPLIT package, there is a post-processing program that you can use to solve for the source matrix, for the source vector I should say, and we will do this for the CAPTEX release rate. We will try to determine the release rate.

A coefficient matrix does not necessarily have to be just different source locations, geographically different, and a group of receptor locations.   It can also be different release times, so the source vector could represent different release times, it doesn’t necessarily have to represent different release locations. So that’s why I introduced this as a more generic form of the source-receptor matrix.   So what we want to do is go ahead and do a CAPTEX simulation for each release hour.   So we want a unique simulation for each hour over some period where know the release occurred.   We're not just going to release at those times that we know, but we are going to release over a range of times that starts before and ends after the known release time.

So to start this, let's go ahead and retrieve, and I think before we do that we're going to do that we’re going to press reset because this is very different from the last calculation.   And we want to retrieve the original CAPTEX CONTROL and name list files and it is called captex_control.txt and for the name list, it is the captex_setup.txt file.   And since we will be doing multiple simulations, let's cut down on the particle number, we will not be going to 68 hours, so open up menu #4 and let's make that just 20000 rather than 50000.

Back to the setup menu.   Now we know the release started at 17 but we know the meteorology starts at 15, so let's start there.   And let's run for 21 hours, since we are starting two hours earlier, previously we were doing 19 hours to go to the end of the three hour sampling, but we're starting two hours earlier, so we need to run two hours longer to go to the end of the three-hour sampling. Open up the pollutant menu and we do not know the emission rate, so we will use a unit emission and since we will be doing a simulation for every hour, each simulation will only have a one hour emission.   So that is we do a simulation starting at 15 UTC with a one hour release that goes through the end of the sampling period, and then we start the simulation again at 16 Z with a one hour release going through the end of the sampling period, and so on. Each time we will do a simulation that is one hour shorter. Therefore we will have multiple simulations, each one representing a different release time, and in the end we will solve this coefficient matrix to determine which releases gave us the best fit, or which simulations give us the best fit with the measurements.   Save that and let's also make the grid a little finer to give us a better looking product, or more representative output, because we are working with the close-in samples and we will give this output file unique let name, for instance TCM, for transfer coefficient matrix.   And I think that should be it, save, save, and save.

Now the simulation, you could certainly do these manually, each time going back and configuring a new simulation starting one hour later and running for one hour less. But there is a script available under special runs called daily. And this is very similar to what we did under trajectories, where it's going to start calculations, you're at 15, and we'll start one every hour, and let's only go, we don't need to cover the whole period, but let's just bracket the known release period. So we could end for instance, nine hours after starting; that would end at 23 UTC on the 25th. So that gives us about two or three hours, two hours before and three hours after the release terminated. And we do want to shorten every simulation by one hour, because all simulations need to end on the 26th at 12 UTC. And the output files will have a month day and hour appended to their name, so that we can identify each simulation.

At this point execute, now this will take a little bit of time, so you can probably go and catch a, get a cup of tea or coffee while this is running. We could've gotten results more quickly if we reduced the particle number, but this simulation, for this simulation, we’re going to try to achieve a little more accuracy. As you might guess, each, with each new simulation, the duration is one hour less, so therefore they will subsequently be running faster, each additional simulation will be a little bit faster, more quickly completed than the previous one.

If I were to go to the working directory while this calculation is occurring, you could see the files being generated, for instance this is the file for the release at 17, the binary, the binary concentration output file, for the release at 18, and we're going to go through 23 UTC, that will be the last file that's generated.

It is now working on the last simulation, and we’ve completed. The next step is to go to the concentration utilities, to the transfer coefficient menu, and we're going to look at the singular value decomposition solution. And we need to populate the menus, so we know that TCM is our base name. I'm not sure what else is in there, so let's continue on, and this becomes the pattern for the wildcard search for determining all files that start with TCM0925, the remaining two digits are the hour of the release. And we click on create and we created an INFILE and like we've done in past examples, let's make sure that in the working directory, this INFILE does contain the files we want, and indeed it does. These are the hourly simulations. And the next up is selecting the measured data file, the three hour data. And the units conversion for picograms is correct.

And we're going to create an output file, a comma delimited output file of c2array and let's do that. And this output file contains the coefficient matrix. That is this would be the, except for the last column, just from here and then down. These are the dilution factors for, this is time, in days since the year 1900, days since year 1900, is the standard format for spreadsheets, so that if you were to import this into say Excel, you can turn this into a real date by just changing the numeric format, and this will be the time of the release, the first release 15 Z, 16 Z, 17Z, and so on until 23. The last column, well I should say each row here, alright, represents the measurements, the three-hour measurements, whether they be zero or something else in picograms per cubic meter. The stations are not identified, but you could go to the file and find out, to the measurement data file, and determine which station these are. And this preprocessing program that you call when you hit the create button, the transfer coefficient matrix create button, looked at each simulation, this is the 15 Z simulation, and determined what the correct, what the dilution factor was, what was the unit source concentration prediction, this simulation to this sampling location, and so on, down for each row. So the dilution factor for this 15 Z release was this number and if you would multiply this number by the release amount, 67000, we would get the contribution in picograms to this measured value. If you were to add up all the release times that would be the total contribution. So this is the matrix.

The next step is simply to solve and you can see from the output file, this is, this information is written to a file call source.txt here. But the results aren't actually very good. We know that, we don't convert the units here, 15, 16, 17, so this start 17, this is the first real release time, 17, 18, 19, so these are the three release times and negative in the solutions sense is zero. So the only one that's close to reality at 58000, is the last one, and you can see this, and very large numbers and zero numbers and numbers that don't really make much sense. So the solution isn't really very good and the problem with the matrix solution is that it doesn’t, the coefficient matrix solution, it doesn't tolerate errors and uncertainties.

As an example we could take a look at this matrix that we just created here, and we know that the, 17, 18, and 19 UTC runs, were the only runs with releases, we could take through actual release rate, 67,000 for instance, multiply

67,000 by this number, by this number, and by this number, and add them together and we would get the model prediction for that particular sample. So, just as an exercise and you don't really need to do this, we've done this for you, and we've done this to give us what we're calling synthetic data or hypothetical data. So if we do not know the real release rate, here's the same, here's the same matrix, so here's the 67,000. We could multiply that by, get the product in this column, this column, and this column, add them together across, everything else is zero, so there's no contribution, and this would give us then the projected, if you will, the model predicted concentration, if the model were the truth. What I’m getting at here is if we were to use the model predicted concentrations as truth, how different are they from the real measured data? When we compare them side-by-side and that's what you're seeing in these last two columns here, this was the measurement but this is the model prediction. So you can see that they're all, when they’re high, they're both, they are bouncing around up high and low within a range that we normally would not find unusual. That there is uncertainty in the model calculations and if I were to look at these numbers as being the model performance compared to these measurements, the model predictions, the model measured data, it doesn't look that bad. It's really within certain acceptable limits but it does not give us a simple solution.

So we have, just as an example, created a pseudo data file, a hypothetical data file of the three hour measurements that are really just model predictions. So, if you will, perfect measurements, so that they would match the model.   Just to test the, to test the computational solutions and we call this the hypo, CAPTEX 2, hypotheticals and if you were to compare these two files with each other, you can see they're not that much different.   So if we were to pretend that this was a, the real measured data file, this hypothetical instead rather than the true measured data, and create the new coefficient matrix and solve it, lo and behold, you can see that we get something pretty close to 67,000 for the three release hours and then pretty much zero after that and there is some suggestion of contributions from the earlier releases.   So this is really the result of the atmospheric dispersion and the inevitability of having some of the releases mixed together within the three hour sampling data.   Had we had shorter duration sampling data this probably would not have been an issue.   But the point of this is that the solution results are very sensitive to errors in the model as well as errors and uncertainty in the measurements as well.

And let’s then exit this, and the point is that what you're doing with the solutions is that you have to be careful and edit data and remove uncertainties as much as possible. The other issue here is we are working with just a very small subset of data.   If we had hundreds of measurements, the situation would be completely different. So we are working with a very small data set and small differences in the model prediction versus the measurements can have an outsized effect on the results. So I don't want to discourage you from trying these approaches but they would work much better with larger sets of data.

And that concludes the coefficient matrix solution discussion. In the next section, and do not delete the simulations, so you do not have to run them again, we will look at another approach to solving the coefficient matrix, that accounts for some of the uncertainties in the model predictions as well as the measurements.

And this concludes our discussion.