In this section we will take a look at computational trajectory error. The approach that we will use is to compute a forward trajectory and then at the end of the forward trajectory we will start a backward trajectory calculation and see how close that backward trajectory ends to the original starting point. And that would be a very simple measure of the computational error that gets accumulated during the trajectory calculation.

So let’s open the graphical user interface, trajectory, set up run.   We will start again by retrieving the previous calculation and since I had not closed the graphical user interface, that filename that we saved, the original 10 m trajectory, is still in the menu.   Once you close the graphical user interface all the internal menu variables are erased except for ones that saved, the files called, that start with a name called, default, in the working directory. So in any event, let's go ahead and load that, and as you recall the only difference here was that the starting location was 10 m. Now to do this computational error calculation, we want to get away from near the ground because of the complications that are introduced when trajectories intersect a ground surface. We will review those issues further later on.

But for now let's go back and change this height back to the mid boundary layer height of 750 m above ground level and select OK.   So again we're starting at, on the 25th, at 1700, we're starting from the Dayton, Ohio, location here. This is what you should have.   We're going to go for 68 hours in a forward direction, using the vertical motion method from the input data.   But let's change the name of the output file to distinguish it from the backward trajectory calculation, which we will do after this one. So let's add underscore FWRD, for forward, this is the forward calculation.   And we will be using the same CAPTEX2_NARR meteorological data file.   So once you've set this up, go ahead and save, and now click on trajectory, run model, and notice that this menu pops up again, that we have a namelist file.   As you recall in the previous section, we were doing fraction of the boundary layer, so that the starting height was 0.5, but we are no longer interested in fractional boundary layer, so in this case, this file was left over from a previous application, so we're going to delete the file, the yellow button, and then run.

Now this calculation has completed, exit, and let’s take a look at your trajectory.   We should've seen this trajectory already several times, go ahead and display trajectory. I'm not going to change any defaults, but notice how the input file name has changed here, so you can you can change the names arbitrarily here, within the graphical user interface, but when you change it in another menu that information is passed to all the other menus where it is appropriate.   So click on execute display and here we have the mid boundary layer trajectory which you should be familiar with at this point.

The next step, I clicked on display too many times, let's quit this. And just for reference later on, why don’t we go ahead and save the forward trajectory configuration, so go back to set up run, and do a save as, and in this case, let's save this traj_fwrd_control.txt.   Obviously you can select any name you like but you'll just have to remember in future examples when we bring this up.

Alright, so now we want to do a backward calculation from that end point, the last point of the forward trajectory.   So the best way to do that is, let's get this out of the way here, and open up the working directory, hysplit4/working, and you'll see that one of the last files we created was tdump_fwrd, so that is the forward trajectory,   So go ahead and open that up with notepad or what whatever text editor you're working with, let me get this over here a little bit so we can see it, and scroll down to the end of the file, and at the end here you will see this is the last trajectory end point, on the 28th.   So, 68 hours after the starting location, after the starting time, so we take that information, and now I can transcribe it, but I'm going to use the cut and paste features of Windows, or on a Mac you can do the same, so I'm going to go select the latitude, longitude, and a height. This is height above ground, above the ground level, and you can you do edit copy here if you like, or you can do a CNTL-C, and I'm just going to go right into the setup menu, and erase this, and do a paste, in this case CNTL-V, and click on OK.   Now we know that the file that we want to start the trajectory on the 28th, at 1300, so we go up to the starting time and make this the 28th at 13, 1300 hours. The other thing we need to change is the run duration, the run duration is going to be the same, but we are going to select backward instead of forward. And all that does from the standpoint of input to HYSPLIT is that it changes the run time from positive to negative. We will leave everything else the same, but we will rename the output file to tdump_back so that we have a unique file name for the backward calculation. So go ahead and do the save. We don’t need this file any longer. And then trajectory, run model.

And let's do a display, and you can see here how the new input file name has been replaced.   We execute, and now we have the backward calculation and you can see this the star here represents the starting location, and time if you look at the axis down here for the vertical projection, goes from the 28th to the 25th, so the time variable is reversed. So this looks good, we can we can go ahead and maybe do the same thing we did with the forward trajectory, that is save the control file, so go back to setup run, save as, and leave all the same, except change this to back. So now we've got a forward and backward file that's been saved.

So the last point, those files are pretty close, but we can use the display program to see how close. So let’s open up trajectory, display, and for the input file we have dump_back, but we could just as well have multiple files here, and the trajectory plotting program will display multiple files on top of each other.

So if I go plus, by using a plus symbol, it will just take each file that follows the plus symbol, and add tdump_fwrd, and I'm going to click on execute display, and now you can see the forward and backward trajectories super-imposed upon each other.   It’s hard to tell which is which. If I look back at the starting location in Dayton, Ohio, and if I'm on a Windows machine, using Ghostview, I can right-click here, and you can see that these points are very close to each other.   So the main take away from this exercise is that computational error is not a significant factor in the trajectory calculation.

And this pretty much concludes the discussion of computational trajectory error