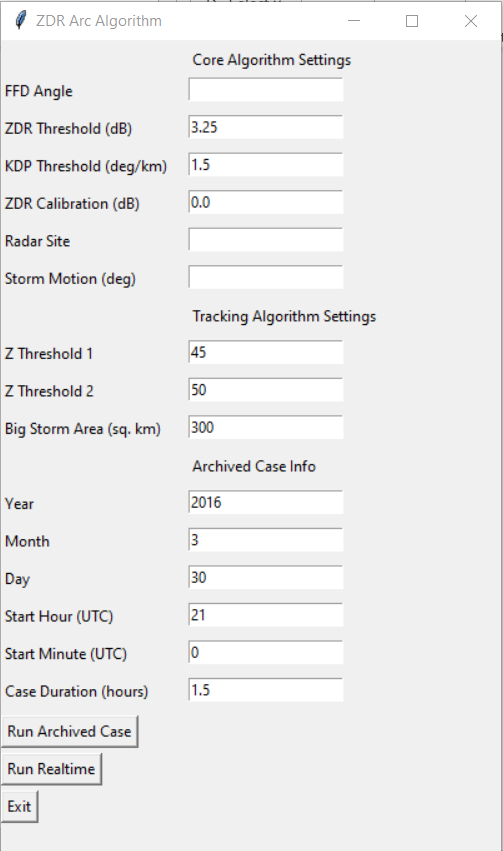
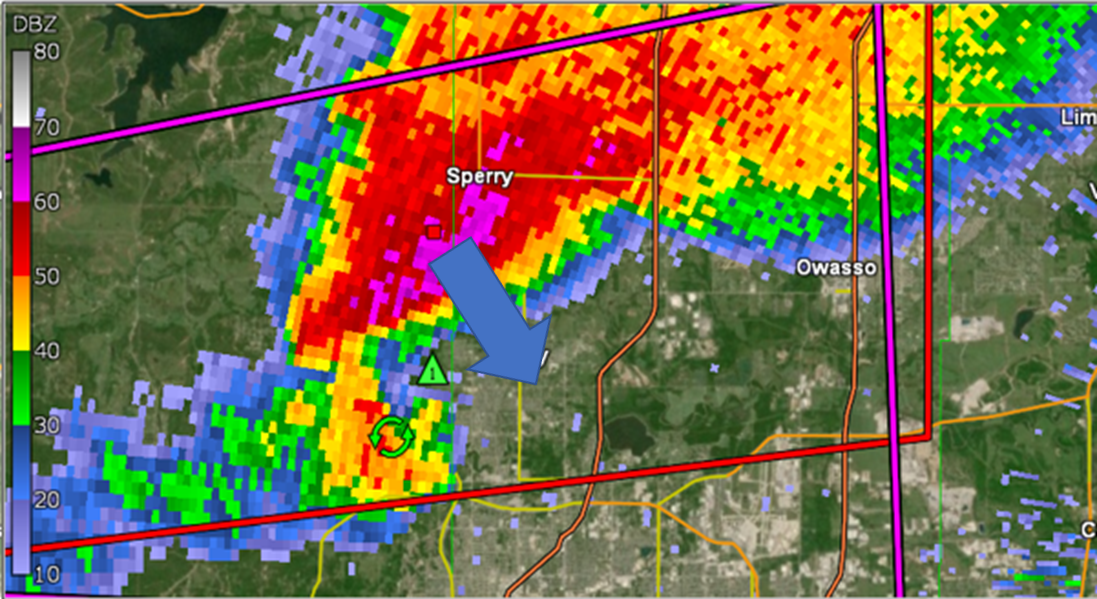
ZDR Arc algorithm Installation Instructions

1. Download the Github repository (<https://github.com/mwilson14/ZDRArcAlgorithm>) and extract the files from the zipped folder into the directory you plan on running the program in.
2. The next several steps involve creating a conda environment to run the algorithm in. To create this environment, launch the Anaconda command prompt (on Windows) or a terminal window (in Linux) and type “conda create –name ZDR\_ARC” (you can replace ZDR\_ARC with a different name you want to call the environment). Make sure you’ve got Anaconda installed first.
3. Activate your new environment with “conda activate ZDR\_ARC”
4. Next, we’re going to install the modules we need to run the algorithm in this environment. The commands that follow each install a module we’ll need.
5. Enter “conda install jupyter” This installs jupyter notebooks, which we can use to run the algorithm later.
6. Enter “conda install -c conda-forge arm\_pyart” This brings in Pyart, which will be used to read and work with the radar data.
7. Enter “conda install -c conda-forge metpy”
8. Enter “conda install scikit-learn”
9. Enter “conda install -c conda-forge siphon”
10. Enter “pip install nexradaws”
11. Enter “conda install numba”
12. Enter “conda install scikit-image”
13. Open another terminal and activate the ZDR\_ARC environment as in step 3. Launch a jupyter notebook from the terminal by entering “jupyter notebook”
14. A browser window should pop up with a jupyter notebook showing all of the directories within whatever folder the notebook was launched from. Navigate through this to the folder the algorithm has been placed in and find the example notebook (in this case, Algorithm\_demonstration.ipynb) and click on it to launch it.
15. Try running the notebook. If it runs, the algorithm is successfully installed.

Running real-time or archived cases with the GUI

1. Open another terminal and activate the ZDR\_ARC environment. Navigate to the directory that the algorithm is in.
2. Type “python gui\_test.py”. This window should pop up: 
3. To run a realtime case, open GR2 analyst and find the supercell you’re interested in looking at. Draw a vector across the FFD perpendicular to the edge of the reflectivity gradient (you can use the marker tool for this if you put a marker in the core and move the mouse across the gradient to get the direction, as shown below) Enter the direction of this vector in the FFD angle box. If storms are not yet present, the FFD angle can often be estimated by subtracting ~90 degrees from the direction of the sfc-8km bulk shear vector.



1. Enter the radar site and storm motion in their respective boxes.
2. Hit the “run realtime” button.
3. The GUI will probably freeze or say “not responding”, but that means it’s working. In a bit, you should have a file called “REALTIMEK[radar site][date]\_Placefile.txt in the directory you’re running the algorithm in. You should be able to add this placefile to GR2 and have output updating in real-time. Due to varying times between radar scans in different VCPs, occasionally more than one time frame of output is visible for a given radar scan, and the output for the most recent scan often disappears ~30 seconds to a minute after appearing and reappears when the next scan comes in. I’m not quite sure why this happens but fixing it is in the plans!

Archived Case Instructions:

Same as realtime except you’ll need to enter a valid time range in the “archived case info” section and use the “run archived case” button instead of “run realtime”.