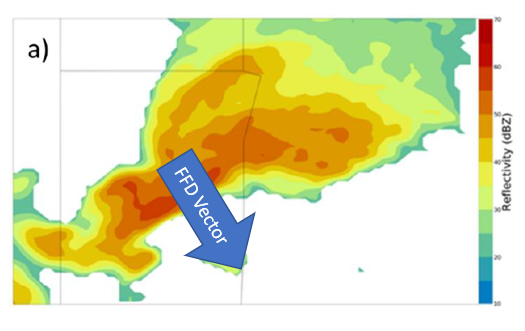
SPORK Installation Instructions

1. Download the Github repository (<https://github.com/mwilson14/SPORK-SPIN>) 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 SPORK in. To create this environment, launch the Anaconda command prompt (on Windows) or a terminal window (in Linux) and type “conda create –name test\_SPORK” (you can replace test\_SPORK 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 test\_SPORK”
4. Next, we’re going to install the modules we need to run SPORK 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’ll use to run SPORK 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 test\_SPORK 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 SPORK has been placed in and find the example notebook (in this case, SPORK2020Examples.ipynb) and click on it to launch it.
15. Try running the notebook. If it runs, SPORK is successfully installed.

**SPORK Input Parameters**

To run a case for SPORK, a number of input parameters must be set. Here’s what they all mean:

storm\_relative\_dir: This indicates the orientation of the FFD edge, which is used as an indicator of the storm’s orientation for the ZDR arc and ZDR column detections. It’s represented by the direction of this vector here:



In this case, storm\_relative\_dir would be approximately 160 degrees. This vector can be estimated from radar imagery for a given storm, and some preliminary results indicate that you may be able to estimate it by adding 90 degrees to the direction of the surface-8km shear vector since the FFD edge tends to line up along the deep-layer shear vector.

zdrlevs: ZDR threshold used to outline the ZDR arc. Generally set at 3.25 dB, but can be lowered in tropical environments where ZDR tends to be lower.

kdplevs: KDP threshold used to identify the KDP foot. Generally set at 1.5 deg/km

REFlevs: Lowest reflectivity threshold used for the storm tracking algorithm. Generally set to 45 dBZ. Can be adjusted up or down in tough storm tracking cases to help prevent track splits.

REFlev1s: Upper reflectivity threshold used for the storm tracking algorithm. Generally set to 45 dBZ. Can be adjusted up or down in tough storm tracking cases to help prevent track splits.

big\_storms: Area threshold at which the storm tracking algorithm starts looking for embedded higher reflectivities in large storm objects to split them into smaller objects. Generally set at 300 sq. km. This value can also be changed in cases with weird tracking problems.

zero\_z\_triggers: An obsolete parameter used to deal with data quality problems from the old THREDDS server. Leave this at 25, future releases will get rid of it entirely.

storm\_to\_tracks: Storm ID used to track and zoom in on a particular storm to produce storm-following loops. Set by default to -999 to plot the entire radar domain.

years: Year of start of analysis period.

months: Month of start of analysis period.

days: Day of start of analysis period.

hours: Hour, in UTC, of start of analysis period.

start\_mins: Minute at start of analysis period.

durations: Duration of analysis period, in hours.

h\_calstm: Freezing level, in meters, read from a RAP or observed sounding.

calibrations: ZDR calibration value, in dB. This value is subtracted from the ZDR field. So, for example, if ZDR is running 0.5 dB too high, this value should be 0.5. A script is available to calculate this value using scatterer-based ZDR calibration from level II radar data and a sounding-derived freezing level.

stations: Radar station the case is from. Must be a string (for example, ‘KAMA’)

**New Parameters:**

These parameters are set to default values in the function call to multi\_case\_algorithm\_2020, so you shouldn’t need to mess with them much.

storm\_motion\_dir: Sets the storm motion direction that SPORK uses to calculate and plot the separation angle. When running archived cases from the dataset I’m currently looking at, I have this set to a default of 240 and then recalculate the separation angle using the observed storm motions in post-processing, since the observed motions from these cases are calculated from the storm tracks and are thus not available before the case is run.

track\_dis: Radius, in km, that the storm tracking algorithm searches when looking for a track to append each new storm cell to. Can be useful to adjust this up a bit with very fast-moving storms. Defaults to 10 km.

GR\_mins: Minutes that each GR2 placefile is valid for. Defaults to 6, but can be adjusted up or down to improve display in GR2. Can cause lots of wackiness with SAILS scans.