Setting Environment for Real-time Skew-T Plotting

This process runs on eol-hurricane as the joss user (sudo su - joss to become the joss user).

A conda virtual environment, named skewt-env, was created (by user anstett, since user joss did not have permission) to run the SHARPpy plotting to create the skew-t plots. These are the commands:

- source /opt/local/anaconda3/etc/profle.d/conda.csh (to set up hooks for conda, this activates conda only in a specific shell session).
- conda create --prefix /opt/local/anaconda3/envs/skewt-env -c conda-forge python=3.7 pyside2=5.12 numpy=1.16 matplotlib=3.2 requests python-dateutil
- conda activate /opt/local/anaconda3/envs/skewt-env
- conda install -c conda-forge sharppy

Notes:
SHARPpy software needs either python 3.6 or 3.7 and anaconda3 can run python 3.7. SHARPpy software also needs the numpy, pyside, requests, and python-dateutil packages. The SHARPpy web page said to install pyside5.12, which is known as pyside2=5.12. The software also needs the matplotlib software to create the actual plots. The version of matplotlib used was 3.2, because the newer matplotlib was giving errors. Creating the environment with the necessary packages up front was successful and then the sharppy package installed correctly. The major conflict is with the pyside package. Creating the environment with all the packages specified will take at least an hour of the "Solving environment" spinning, but it is successful.

"conda deactivate" is used to deactivate the skewt-env virtual environment.

A wiki page that describes the use of anaconda and python in EOL.

Necessary software:
anaconda3 (already installed in /opt/local/anaconda3).
SHARPpy (version 1.4.0 was used). This version is from 2020. There haven't been any recent updates to this software.

SHARPpy software requires the NumPy library. Running the full_gui.py requires the PySide library, but the plotting software does not run the full_gui.py.

See this web page for more information on the SHARPpy software. Click on the link on the right under "About" for the SHARPpy documentation.

The Real-time-plots.pl script runs a "ant sort_esc" command. This command caused a security incident of ransomware activity to be reported to CISL since I was running the command as the joss user using sudo. I sent the required information to CISL on what I was running and CISL

added this process to the Cortex allow list on eol-hurricane. CISL has some automated processes running to detect perceived security violations such as this on all servers.

The following changes were made to the SHARPpy software:
Changed
/opt/local/anaconda3/envs/skewt-env/lib/python3.7/site-packages/sharppy/plot/skew.py
In draw_title routine - changed font from 14 to 10
In draw_hodo_inset routine - changed width and height from 1.7 to 1.4 on “inset_axes = “ line
- changed fontsize from 8 to 6 on “inset_axes.text” line

python -m compileall . (compile the new version of skew.py. Use this command while in the skewt-env in the sharppy/plot directory).

Everything below was done the first time to create the plot_sounding.py program.

Downloaded example code for plotting a skew-t from:
https://sharppy.github.io/SHARPpy/auto_examples/plot_sounding.html

During development ran:
jupyter notebook --no-browser --port=8895 (on tikal, same /opt/local disk is also mounted on tsunami).
On personal mac ran:
ssh -L 8015:localhost:8895 anstett@tikal
In local browser:
localhost:8015
(enter token given when starting jupyter on tikal)

After development was finished, the jupyter notebook was converted to a python script:
ipython nbconvert --to python plot_sounding.ipynb (run ipython from /opt/local/anaconda3/bin).
where plot_sounding.ipynb is the name of the jupyter notebook. This command created plot_sounding.py.
Changes made after ipython nbconvert was run:
Add:
#!/opt/local/anaconda3/bin/python (as the first line)
for the correct matlab libraries to be included.
Change plot_sounding.py to executable.
Comment out magic line (first line in file before #! line was added).
Add the following before any matplotlib “from” statements:
import matplotlib
matplotlib.use(‘Agg’)
Add:
plt.ioff() (To turn interactive plotting off).