Problem Set #6

Assignment Learning Objectives

1. Use of Units within Python calculations, specifically using MetPy
2. Develop code in a modular format using definitions
3. Develop text and graphical output using MetPy and other Python libraries **Due 27 October 2020 by 3:00 pm**

Due Online

- The final script in the PSet6 repository on your GitHub account

Due Via Blackboard

- A PDF version of your code. Make sure the honor code appears in the comment block at the beginning of the script and that it has your full typed out name. This will serve as your assertion that you have upheld the honor code.

Honor Code

You must type the full honor code into each comment block to indicate that you have upheld the code. Authorized aid for this assignment is your notes and any previous programs you have written. You are highly encouraged to work with others in the class to help diagnose bugs and work out programming logic. Any copying of someone else's code IS A VIOLATION OF THE HONOR CODE, whether from a classmate, or the Internet. *Be sure to indicate with whom you have worked in the comment block of your submission.*

Background

Soundings and the parameters that can be calculated from them are essential tools that meteorologists use on a daily basis. While it is relatively easy to calculate some of these parameters by hand, it is helpful to have predefined functions that can make easy work of doing these calculations for any number of soundings. Additionally, it is useful to automate the plotting of temperature, dewpoint, and wind speed and direction on a skew-T diagram, so that a high-quality visualization can be produced quickly and on-demand. This assignment will slowly build a single program to calculate a number of variables and produce high-quality output to an end user.

Problems

1. Write a Python program that reads in a sounding data file from the Wyoming sounding archive and print the raw data to standard output. The program input should be the sounding location (three-letter identifier) and date/time in YYYYMMDDHH.

   Program Name: **soundings.py**

   Program Input: Sounding Location and Date/Time

   Program Output: Sounding Text File **(<station ID>_YYYYMMDD_HH.txt**) and Skew-T Image **(<station ID>_YYYYMMDD_HH.png**)

Test Sounding Location: SGF

Test Sounding Date/Time: 2004061212

2. Calculate the following variables for each level in the sounding and create well formatted output similar in style to that of the Wyoming site including all appropriate variables:

    - Potential Temperature
    - Mixing Ratio
    - Relative Humidity
    - Wind Speed
    - Wind Direction

    > Note: Utilize your resources well. These can all be calculated easily using the MetPy module.

3. Calculate the following levels for a surface-based parcel path and add to the text output of problem 2:

    - Lifting Condensation Level
    - Level of Free Convection
    - Equilibrium Level
    - SBCAPE
    - SBCIN

    > Note: Utilize your resources well. These can all be calculated easily using the MetPy module.

4. Create a definition to calculate Lifted Index (LI) and add to the text output of problem 2. Be sure to include an appropriate docstring for the definition, defining the input, output, what is being calculated, and how is it being calculated. For reference, the following is the equation for calculating LI,

    $LI = (T_{env} – T_{parcel})_{500}$

    > where T_{env} is the temperature of the environment (at 500 hPa) and T_{parcel} is the temperature of the surface-based parcel (at 500 hPa).

5. Create a definition to calculate the Total Totals (TT) index and add to the text output of problem 2. Be sure to include an appropriate docstring for the definition, defining the input, output, what is being calculated, and how is it being calculated. For reference, the following is the equation for calculating TT,

    $TT = T_{850} + T_{d850} - 2T_{500}$

    > where T_{850} is the temperature of the environment at 850 hPa, T_{d850} is the dewpoint temperature of the environment at 850 hPa, and T_{500} is the temperature of of the environment at 500 hPa. The TT index is derived from two parts: the vertical totals (VT) and the cross totals (CT),

    $VT = T_{850} - T_{500}$

$CT = T_{d850} - T_{500}$

$TT = VT + CT$

6. Plot the sounding on a skew-T using Python and include the surfaced-based parcel profile. Have an appropriate title on the skew-T with the station ID, date and time of the launch, and plot markers and text to identify the location of the LCL, LFC, and EL.

7. Have the program assess the likelihood of severe weather happening as a result of a given sounding. Use the following critical values for skew-T parameters to have the program make the determination if severe weather is likely to occur or not. If at least two of the parameters outlined below are satisfied, then severe weather is more likely than not to occur if thunderstorms develop issue the following statement, "WARNING: SEVERE THUNDERSTORMS ARE POSSIBLE!", otherwise issue the following statement, "NOTE: Severe Thunderstorms Not Likely To Occur."

    - CAPE >= 1500 J/kg
    - CIN >= -125 J/kg
    - LI <= -2C

Text Output Example (blue text indicating input from commandline):

```
>>>

Input Location: SGF

Input Date/Time (YYYYMMDDHH): 2004061212

SGF Observations at 12Z 12 Jun 2004
------------------------------------------------
PRES   TEMP  DWPT  RELH  MIXR  DRCT  SKNT  THTA
hPa     C      C     %   g/kg  deg   knot   K
------------------------------------------------
986.0  22.4  21.2  92.9  16.60 200     5  298.3
948.0  23.6  21.1  85.9 1 6.85 229    23  301.3
.
.
.
.
.
.
4.8   -23.3 -57.3  2.8   3.46   63   35  1148.1
4.7   -24.0 -57.6  2.9   3.41   59   35  1151.8

Station Information and sounding indices

Station identifier: SGF
Observation time: 040612/1200
```

```
    Lifted Index: -5.77
    Total Totals Index: 54.2
    SBCAPE: 1883.7 J/kg
    SBCIN: -87.1 J/kg
    LCL: 951 hPa
    LFC: 801 hPa
    EL: 207 hPa

    WARNING: SEVERE THUNDERSTORMS ARE POSSIBLE!

    >>>
```

Evaluation Criteria

Each of the following criteria will be rated from not present/completed to exemplary, having all of the elements will yield at least a 7/10 for a particular criterion. The assignment is out of 50 points.

- Efficiently coded and correct Problems 1-3 (10 points)
- Efficiently coded and correct Problems 4, 5, 6, and 7 (10 points)
- Efficiently coded and correct skew-T plot and text output file (5 points)
- Overall code is structured and styled well (5 points)
- Informative and Clear Output from running code (10 points)
- Code is well documented (informative comments/descriptions of code blocks; 5 points)
- Code was regularly updated and committed to GitHub repository (5 points)