Cross Section Analysis in MetPy

Jonathan E. Thielen [1]   Ryan M. May [2]   John R. Leeman [2]

[1]Department of Geological and Atmospheric Sciences, Iowa State University, Ames, IA   [2]Unidata/UCAR, Boulder, CO

Background

- Cross sections are a common meteorological plot, often used in analysis of:
  - Convective systems
  - Isentropic ascent
  - Tropopause folds
  - Conditional Symmetric Instability (CSI)
- The General Meteorology Package (GEMPAK) has the ability to obtain and plot such vertical cross sections through three-dimensional data, however, this functionality has been lacking until recently in MetPy.
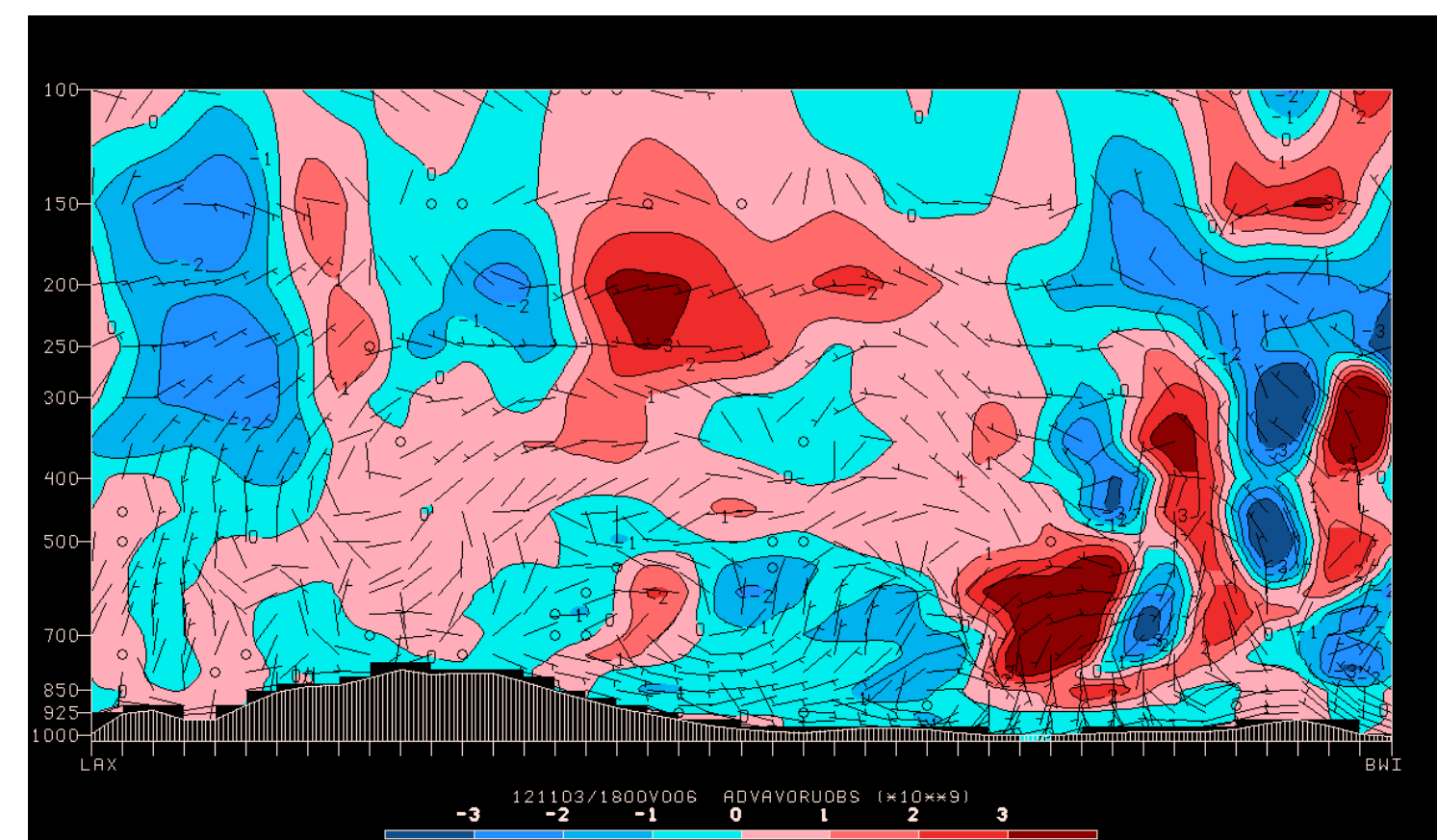
- Cross section interpolation (utilizing xarray) is now included in MetPy based on work undertaken during the 2018 Unidata Summer Internship.

xarray and MetPy

xarray is a powerful Python package that provides N-dimensional labeled arrays and datasets following the Common Data Model with many useful utility methods for indexing, inter-polating, and reducing data. MetPy is transitioning to using xar-ray as its primary data model and includes of number of helper methods via an accessor interface, such as those below:

```python
import xarray as xr
from metpy.units import units

# Parse for CRS and coordinate types
data = xr.open_dataset('data.nc').metpy.parse_cf()

# Get a Cartopy CRS
data['temperature'].metpy.cartopy_crs

# Unit and coordinate helpers
data['temperature'].metpy.sel(
    vertical=500 * units.hPa)
data['u_wind'].metpy.convert_units('knots')
```
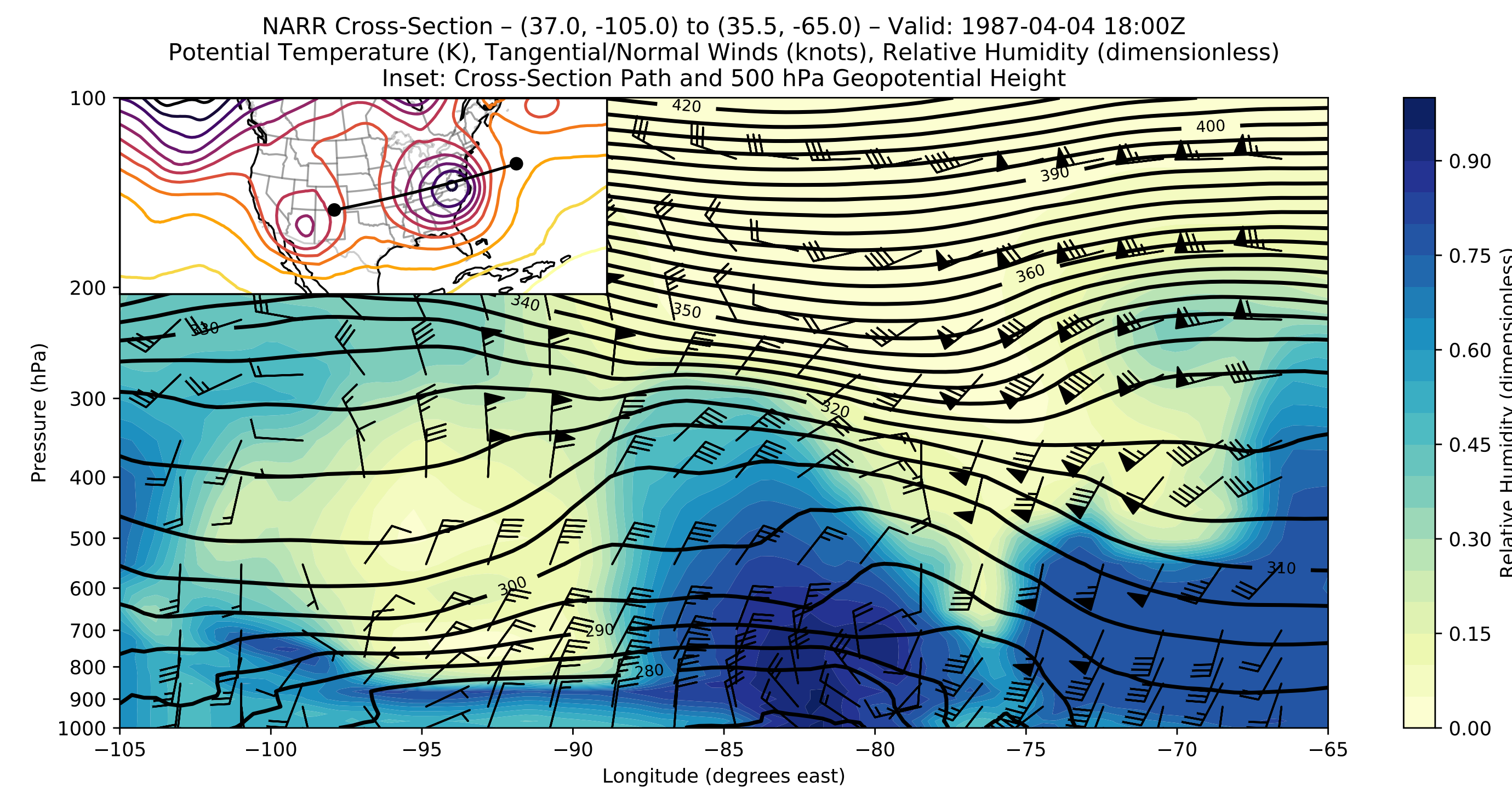

Figure 1. Basic cross section example using NARR data, with demonstration of wind components (included in MetPy documentation).
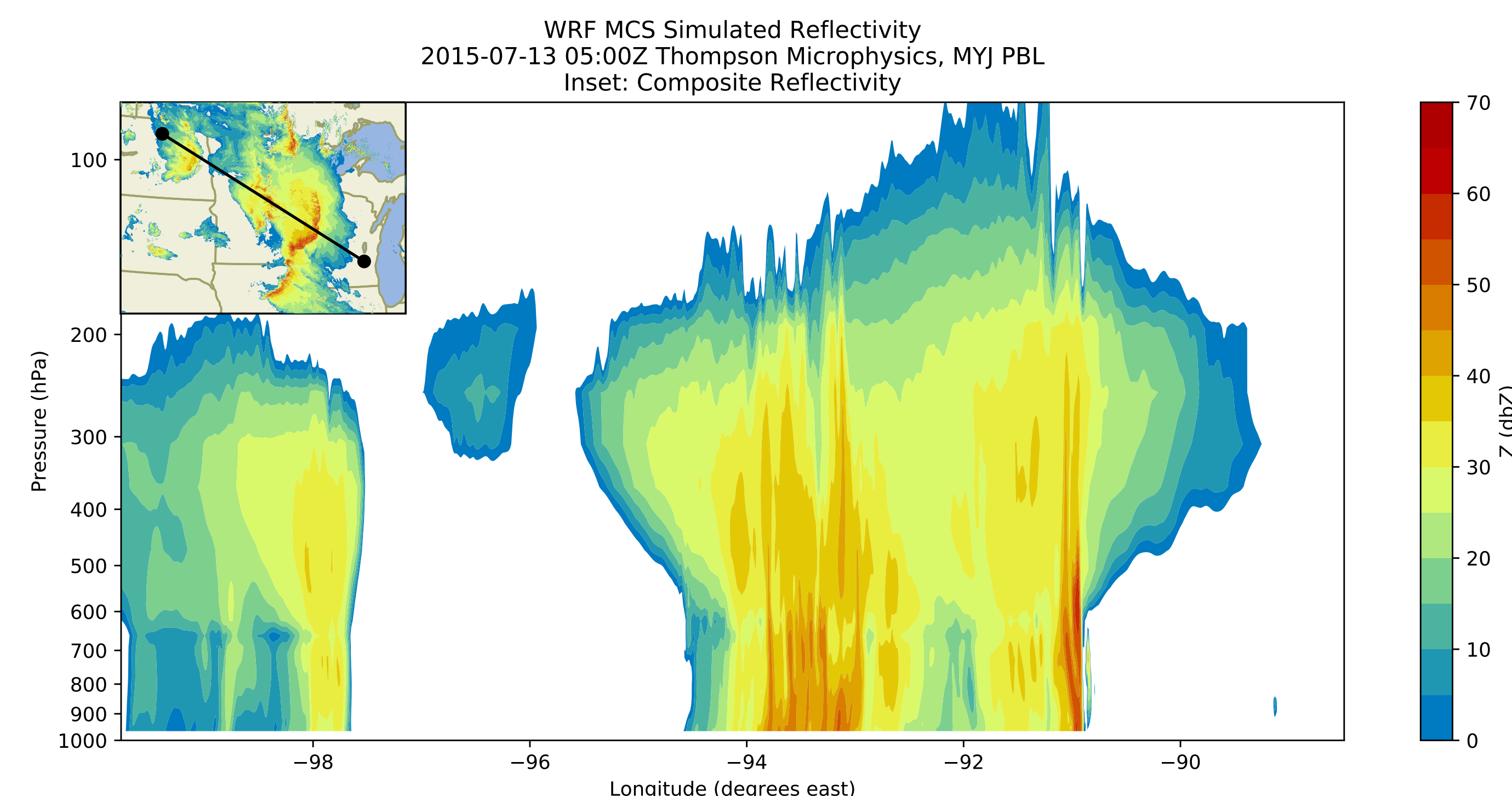

Figure 2. Cross section of simulated reflectivity through a bow echo as simulated in a 1-km WRF-ARW run.
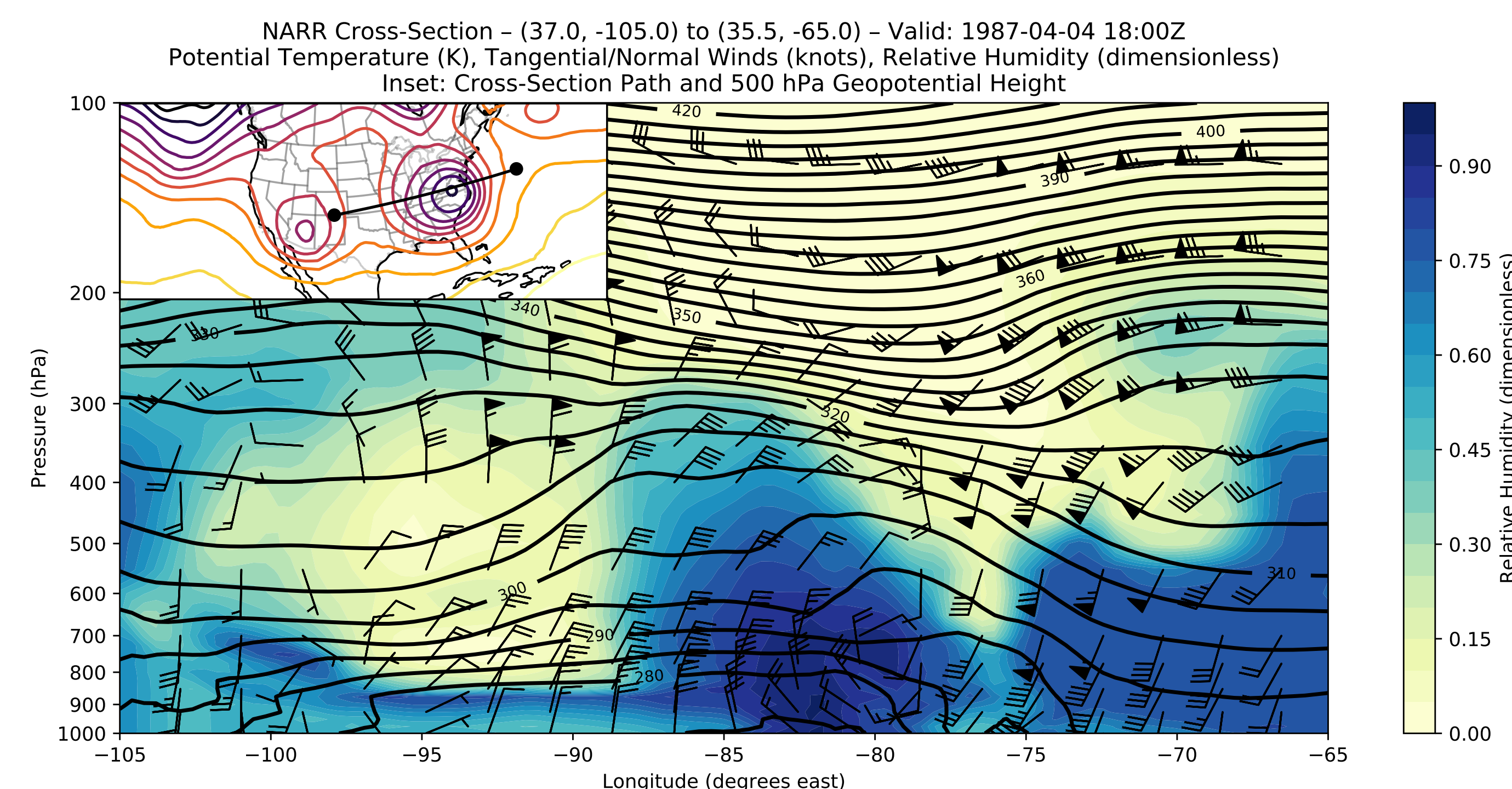

Figure 3. Cross section of Moist Potential Vorticity (MPV) for assessing CSI. (Cross-sectional absolute momentum calculation also available.)

Implementation

- Preliminaries with xarray integration
  - Construction of Cartopy CRS from CF metadata and basic unit helpers (already implemented in v0.8)
  - Systematic identification of coordinate types (from CF metadata, or regular expression parsing as a fallback) to automatically recognize which coordinate corresponds to the x, y, and vertical directions
  - Derivative calculations updated to accept and return xarray DataArrays (while being coordinate-aware)
- MetPy's `cross_section()` is currently a coordinate-type- and projection-aware wrapper for xarray interpolation
  - Takes a number of steps and a start and end point as arguments, and constructs a geodesic path along which to interpolate the input gridded data
  - Linear and nearest neighbor interpolation methods available
  - Input gridded data must be at least three-dimensional (x, y, and vertical), and any extra dimensions (such as time or ensemble member) are preserved
  - `interpolate_to_slice()` available for cross sections on arbitrary paths
- Cross section calculations (such as projected wind components and cross-sectional absolute (or "pseudoangular") momentum) are also available

Future Work

- Extending cross section interpolation to unstructured data, such as observed soundings
  - Use MetPy's interpolation methods like Cressman, Natural Neighbor, and Radial Basis Function
  - Needs easy way to access surface and upper-air observations and a reliable format to store metadata-rich unstructured data in xarray
- Making these cross section plots easier using a declarative interface (see Talk 4.1 by Ryan May, "Bringing GEMPAK-like Syntax to Python with MetPy's Declarative Plotting Interface")

Interested in finding out more or exploring some demo notebooks? Check out

```
https://github.com/jthielen/
metpy_cross_section_ams_2019
```