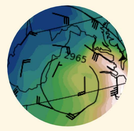
The MetPy Roadmap: Replacing Legacy Meteorological Tools

J. R. Leeman and R. May

Unidata - Boulder, CO

What is MetPy?

File Formats



Plotting

Calculations

- **23 Contributors**
- **115 Forks**
- **223 Star Gazers**

Community Driven We use feedback from scientists and educators to help plan development. Many users even contribute code to MetPy!

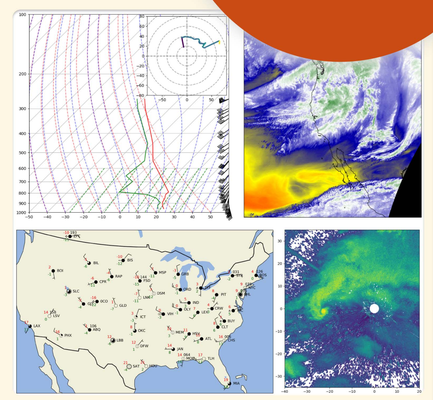
Documented Reproducibility of science is paramount in the computationally driven research world of today. We strive to document exactly how calculations are done with references to the relevant scientific literature.

Reducing time to science!

How you can learn

MetPy Mondays

Join us every Monday morning for a coffee-break sized lesson on how to use MetPy! Checkout the Unidata Developer's Blog and subscribe to our YouTube channel.



Examples are a great way to learn and get code to start you on your project. The MetPy examples and Unidata Python Gallery are great places to start!

```
# Legend
purple = mpatches.Patch(color='BlueViolet', label='Cyclonic Absolute Vorticity Advection')
yellow = mpatches.Patch(color='yellow', label='Surface MSLP < 1010 hPa')
green = mpatches.Patch(color='green', label='Surface Td > 65 F')
tan = mpatches.Patch(color='tan', label='700 hPa Dewpoint Depression > 15 C')
red_line = lines.Line2D([], [], color='red', label='Best Lifted Index (C)')
dashed_black_line = lines.Line2D([], [], linestyle='dashed', color='k',
                                 label='12-hr Surface Pressure Falls (hPa)')
black_line = lines.Line2D([], [], linestyle='solid', color='k',
                          label='12-hr 500-hPa Height Falls (m)')
plt.legend(handles=[jet300, jet500, jet850, dashed_black_line, black_line, red_line,
                    purple, tan, green, yellow], loc=3,
           title='Composite Analysis Valid: {:s}'.format(str(vtimes[0])))

plt.show()
```

Unidata has produced an online Python training course, as well as materials for our regional workshops that are freely available. These notebooks provide valuable training and examples.

Where are we going?

Challenges Facing Scientific Computing

Units MetPy uses unit aware calculations, but the scientific software ecosystem is still catching up and there are bumps to smooth out.

Data Model Meteorological data are complex. There are ungridded multi-dimensional observations, large gridded model output, and multiple coordinate systems. We are working to integrate XArray/Pandas as a data model for MetPy.

Watch milestones on GitHub for upcoming features!

Spring 2018:

- BUFR reader
- Wet bulb temperature
- Skew-T improvements
- Community contributions

Don't forget! conda update metpy

Declarative Plotting

GEMPAK

```
#! /bin/csh -f
source /Users/gempak/GEMPAK6.3.0/Gemenviron
set CURDAY = `date -u +%Y%m%d`
set FRUN = 12
set FTIME = 'f012'
set GDFILE = /models/gfs/${CURDAY}${FRUN}_gfs003.gem

set PROJ = 'str/90;-100;0'
set DEV = 'gif|us.gif|1024;768'

gdcntr <<EOF1
 GDFILE   = $GDFILE
 GDATTIM  = $FTIME
 GLEVEL   = 700
 GVCORD   = pres
 CTYPE    = f
 GFUNC    = avor(wnd)
 CONTUR   = 2
 CINT     = 2
 LINE     = 1/1
 FINT     = 10;12;14;16;18;20;22;24
 FLINE    = 101;21;22;23;5;19;17;16;15;5
 TITLE    = 31/-2/GFS ~
 CLEAR    = n
 GAREA    = us
 PROJ     = $PROJ
 DEVICE   = $DEV
 r

e
EOF1
```

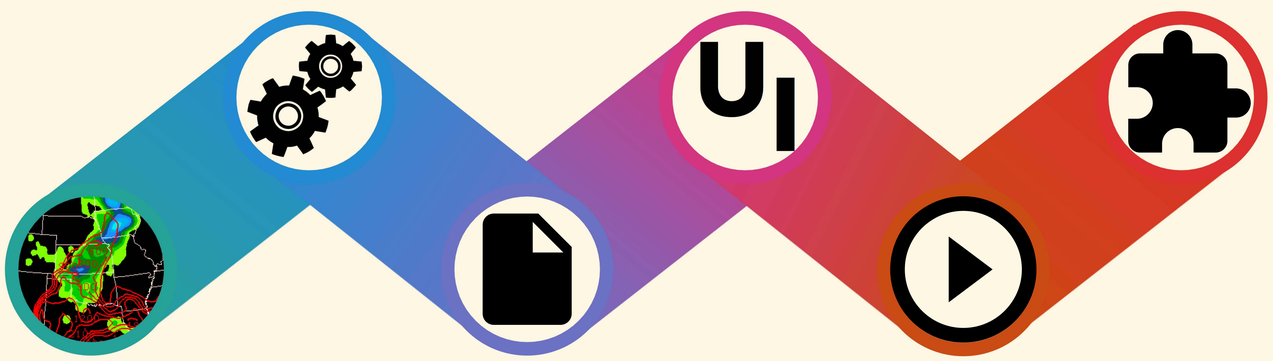
MetPy

```
cur_time = datetime.utcnow()
data_file = '/models/gfs/{0:%Y%m%d}{1}_gfs003.nc'.format(cur_time, 12)
forecast_time = cur_time + timedelta(hours=12)

config.data_file = data_file
config.datetime = forecast_time
config.level = 700
config.vertical_coord = 'pressure'
config.field = 'absolute_vorticity'
config.plot_type = FilledContour
config.intervals = range(10, 26, 2)
config.map_area = 'us'
config.projection = PolarStereographic
config.title = 'GFS {plot.datetime}'
config.output = 'us.png'
```

We are in the process of developing a declarative plotting interface, somewhat reminiscent of GEMPAK that greatly reduces the barrier to entry by writing the "boiler plate" code for you!

Tell us your suggestions!



Port GEMPAK Calculations

METAR, GRIB, BUFR

Training Materials

Checkout MetPy!

Unidata and this work are supported through grants NSF-1344155 and OAC-1740315. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.