Development and uses of the Python-ARM Radar Toolkit (Py-ART)

Jonathan Helmus[1], Scott Collis[1], Karen Johnson[2], Scott Giangrande[2], Kirk North[3] and Michael Jensen[2]

1. Argonne National Laboratory, Argonne, IL, United States.
2. Brookhaven National Laboratory, Upton, NY, United States.
3. McGill University, Montreal QC, Canada.

Abstract

The Atmospheric Radiation Measurement (ARM) climate facility routinely deals with large amounts of complex remote sensing data. The measured parameters from this remote sensing data as well as Value Added Products (VAPs) are made available to the cloud and climate modeling communities. Preparing this data for dissemination requires extensive use of computational resources and algorithms which are not well addressed by current software packages. Here we will report on our progress in the development of the Python-ARM Radar Toolkit (Py-ART) to address these needs.

Py-ART offers a powerful interpreted environment for ingesting radar data from a number of formats, correcting for aliasing and attenuation, mapping data to Cartesian grids and performing a number of geophysical retrievals on the data. The package is also capable of writing data to Climate and Forecast (CF) standard NetCDF files as well as the emerging CF-Radial format for antenna coordinate data. Py-ART is written in the Python programming language, taking advantage of the powerful scientific libraries (NumPy, SciPy, matplotlib) available for the language as well as interfacing with legacy C and FORTRAN radar code. The package will be freely available under an open-source license. Here we will focus on our recent efforts to make the package more accessible to end users by simplifying the interface, increasing code readability, and developing high-quality documentation.

Fork me on GitHub

Coming Soon: https://github.com/ARM-DOE/pyart

Ingest and writing

Py-ART has the ability to natively injest (read) radar data from MDV and CF-Radial as well as other NetCDF based formats. If the NASA TRMM Radar Software Library is installed Py-ART can also injest data from any formatted supported by the library (UF, Sigmet, Lassen, etc). Field data and instrument metadata are stored in memory as a **Radar** object which the routines in Py-ART can interact with. Data can be written out to Climate and Forecast (CF) standard NetCDF files including the CF-Radial format.

CF-Radial

Lassen

Sigmet

MDV

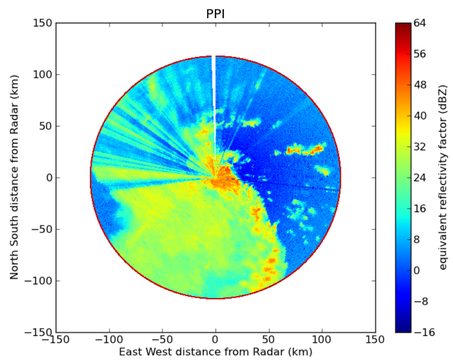
UF

Plotting

Py-ART can quickly create high quality plot of radar moments. Support for creating both plan position indicator (PPI) and range-height indicator (RHI) plots as well as plotting individual rays.

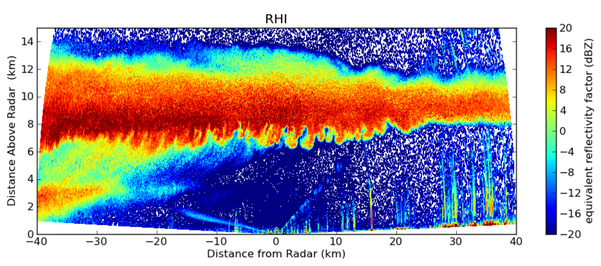
PPI

```
#! /usr/bin/env python

import matplotlib.pyplot as plt
import pyart

radar = pyart.io.read_mdv('110635.mdv')
display = pyart.graph.RadarDisplay(radar)
fig = plt.figure()
ax = fig.add_subplot(111)
display.plot_ppi('reflectivity_horizontal', 0, vmin=-16, vmax=64, title='PPI',
                 ax=ax)
fig.savefig('ppi_plot.png')
```



RHI

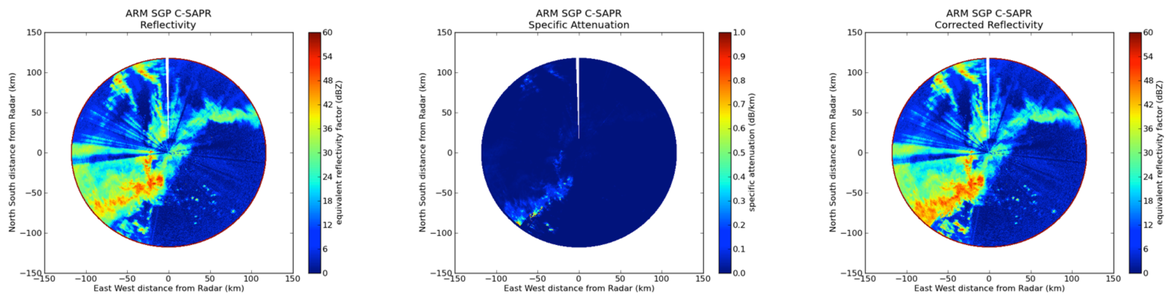
```
#! /usr/bin/env python

import matplotlib.pyplot as plt
import pyart

radar = pyart.io.read_netcdf('sgpxsaprrhicmacI5.c0.20110524.015604_NC4.nc')
display = pyart.graph.RadarDisplay(radar)

fig = plt.figure(figsize=(12, 4))
fig.subplots_adjust(hspace=0.4)
ax = fig.add_subplot(111)

display.plot_rhi('reflectivity_horizontal', 0, vmin=-20, vmax=20, title='RHI',
                 ax=ax)
display.set_limits(ylim=[0, 15])
fig.savefig('rhi_plot.png')
```



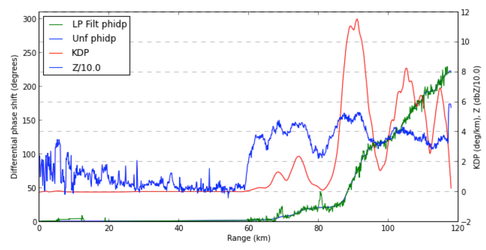
Antenna coordinate moment corrections

Py-ART includes implementations of many algorithms which can be used to correct and analyze raw radar moment information.

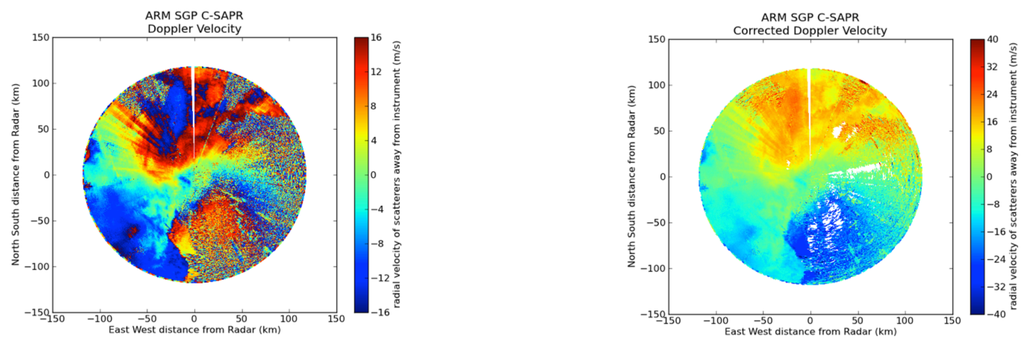
Z-PHI based attenuation correction



Phase correction using an LP algorithm



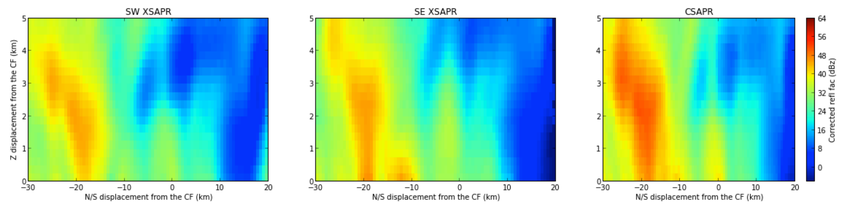
Univ. Washington 4D Velocity Dealiasing



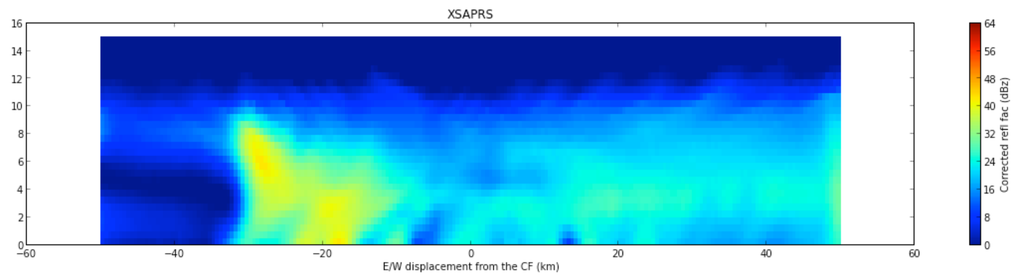
Mapping and Gridding

Py-ART contains a generalized ball-tree mapper which can be used to map data from one or multiple radars to a Cartesian grid .

Co-Grid



Multi-Grid



Additional information and Acknowledgements

Py-ART is an open source project which will be released under a BSD license, if you are interested in trying out a beta version please contact the lead developer, Jonathan Helmus (jjhelmus@anl.gov) for access to the GitHub repository and help with installing the package.

Documentation for Py-ART is available online at: http://jjhelmus.github.com/pyart/dev/index.html

Thanks go to the following for providing algorithms, code and support:
Scott Collis, Scott Giangrande, Kirk North, Alexander Ryzhkov, Matthias Steiner, Bart Kelley, Eric Bruning,
… and many others!

Py-ART would not be possible without the hard work of many other open source Scientific Python packages such as NumPy, SciPy, matplotlib, and python-netcdf4.

Argonne National Laboratory is a U.S. Department of Energy laboratory managed by U Chicago Argonne, LLC.

This poster has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. This research was supported by the Office of Biological and Environmental Research of the U.S. Department of Energy as part of the Atmospheric Radiation Measurement Climate Research Facility.