Open radar science

Using community coding paradigms to enable more people to do more with polarimetric radar measurements.
Outline

- What is a community coding paradigm? How do we make this happen and why do we bother?
- The Art and Zen of the Python ARM Radar Toolkit, Py-ART. Get the data model right and the rest will follow.
- Py-ART in the wild:
  - In facility infrastructure: Within the Department of Energy for high quality radar products.
  - In science: Using Py-ART to look at KDP columns as a proxy for vertical velocity.
  - In climate: Using Py-ART to validate regionally refined climate models.
- Uses Py-ART
  - ARTView
  - CSURadartools
  - Coming soon: 3D winds
- Finally, really why bother: Some highlighted third party contributions.
Community coding

- Most code is, my necessity, community based. Large software projects usually have more than one interacting coder working together.
- However, in the context of this talk a Community maintained package is one that, through various social coding platforms (GitHub, BitBucket) allows solicited and unsolicited contributions to be submitted for consideration for insertion into the package.
Art and Zen of Py-ART

- Py-ART is a data-model driven open architecture for working with radar data in the Python programming language.
- It is a community codebase which uses GitHub to allow users to interact with the code including submitting “Pull Requests”.
- Funded by ARM covering around 0.3FTE.
- Designed mainly for analysis and visualization and not operational software, although it could be and is used in operations by some groups.
- Py-ART has can read a large number of formats and the data model is based on the CF-Radial standard.
- 100’s of users, 1000’s of downloads, 15 contributors, 3 funded by ARM.
The US Department of Energy’s Atmospheric Radiation Measurement (ARM) program manages a number of sites world wide including a Supersite in Northern Oklahoma.

ARM has three X-Band systems, one C-Band system, a Ka/W band system and a vertically pointing Ka band system.

We found “off the shelf solutions” to be very difficult to work with. Either too closed or too difficult to adapt.

The next few slides show techniques we used to get a handle on second trip returns.
The second trip problem
The second trip problem
Enter texture

ARM SGP C-SAPR 1.9 Deg. 2011-05-20T11:01:00Z
Radial velocity of scatterers away from instrument

radial velocity of scatterers away from instrument (meters_per_second)

37°N

36°N

98°W

97°W
Texture calculation using circular statistics

- Magnetron systems have randomized phase on transmit.
- The texture of radial velocity is an ideal indicator of the existence of a valid first trip echo.
- We want to work on the pre-dealiased data and the aliasing jump will cause false texture.
- Answer: Calculate texture using circular statistics so going through nyquist is just going through.
- Scipy’s ndimage module is then used to convolve this with the radar image.

```
angles = np.asarray(angles)
x = np.cos(angles)
y = np.sin(angles)
norm = np.sqrt(x.mean()**2 + y.mean()**2)
return np.sqrt(-2 * np.log(norm))
```


Scott Collis - Open Radar Science - Australian Radar Workshop 2016
Enter texture

ARM SGP C-SAPR 1.9 Deg. 2011-05-20T11:01:00Z
Radial velocity of scatterers away from instrument

37°N

36°N

98°W

97°W

radial velocity of scatterers away from instrument (meters_per_second)
Enter texture
Enter texture
Enter texture
Py-ART in the wild: Looking at KDP just above the Freezing layer

- Vertical velocity is one of the most important measurements for understanding physical processes in the atmosphere.
- A common technique uses two Doppler radars and performs variational analysis to retrieve wind motions.
- But, what if we could look for *fingerprints* of vertical motions? This follows on from work by May and Rajopadhyaya who used profilers to show electrical activity was a good proxy.
- In this study we looked at regions of Specific Differential Phase above the freezing level which is a symptom of large super cooled liquid water droplets.
- We used Py-ART’s Linear Programming phase processing in order to retrieve high quality phase data.
- See van Lier-Walqui et al.


Scott Collis - Open Radar Science - Australian Radar Workshop 2016
Multi-Doppler Updraft & C-SAPR $K_{dp}$ statistics, May 23, 2011

- **Updraft mass flux (kg s$^{-1}$)**
  - Lines for different temperatures: 0°C, -10°C, -20°C, -30°C

- **$K_{dp}$ vol (km$^3$)**
  - Lines for different conditions: $K_{dp}$ > 1.5 km$^{-1}$ vol above melt, $K_{dp}$ supercooled "mass"

**Time (UTC):**
- From 21:00 to 23:00
http://www.giss.nasa.gov/staff/mvanlier-walqui/houston_tracking/khgx_track_20130608_009.html
New kid on the block: Climate Model Development and Validation (CMDV)

- As we run ACME in RRM mode at resolutions approaching 10km large scale storm dynamics and macrophysics become resolved.
- We will be using ARM and supporting observations to build a picture of what is happening in the column and surrounding domain.
- Work will focus primarily on the decadal record collected at the ARM TWP site in Darwin and, working with the rest of the team expanded to other sites.
The key science question is: When ACME is run at high resolution how well are explicitly resolved mesoscale features simulated?

In the column we plan to work with Laura Riihimaka (PNNL) to integrate the ARM cloud classification VAP with precipitation info.

We will use large scale data (sounding, VARANAL) to stratify into forcing regimes, likely using Pope et al. 2009 as a starting point.

For each regime we will compare high resolution data (rainfall, 3DVAR winds) to high resolution models and “de-resolved” 13km data to ACME.
Others!

- NASA Langley: Microphysics and lightning physics
- University of Wyoming: Data quality analysis, airborne radar toolkit
- SIMEPAR, Brazilian met agency using Py-ART to pre-process
- Meteo-Swiss: Polarimetric processing of X-Band data
- Australian Bureau of Meteorology: Both research (ASR) and dealiasing.
- University of Queensland: Bushfire dynamics and ash microphysics using X-Band radar.
- MANY university groups and agencies.
Works with Py-ART

- ARTView [https://github.com/nguy/artview](https://github.com/nguy/artview)
- SingleDop [https://github.com/nasa/SingleDop](https://github.com/nasa/SingleDop)
- PyTDA [https://github.com/nasa/PyTDA](https://github.com/nasa/PyTDA)
- DualPol [https://github.com/nasa/DualPol](https://github.com/nasa/DualPol)
- Coming soon! Multi-Doppler retrievals!
So, really why bother?

- Our hard work has paid off big time.
- We have had many contributions from the community which is now in use on our own systems.
- Additional tools like unit testing, continuous integration, automated documentation coverage etc. Really helps.
- We also did a lot of work with outreach, courses and reaching out to a larger community.
- We encourage contributions and we are currently working on a 5 year roadmap to help guide resource prioritization
- A standard data model is everything and is the reason for Py-ART’s success.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Status</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>#598</td>
<td>Hydrometeor classification and QPE</td>
<td>✔️</td>
<td>meteoswiss-mdr</td>
<td>Sep 24</td>
</tr>
<tr>
<td>#583</td>
<td>Computation of secondary moments, noise correction of RhoHV and pseudo-PPI from RHI volume</td>
<td>✗</td>
<td>jfigui</td>
<td>Sep 17</td>
</tr>
<tr>
<td>#555</td>
<td>Added rainbow_wrl.py. Function to read rainbow5 files</td>
<td>✔️</td>
<td>jfigui</td>
<td>Jul 30</td>
</tr>
<tr>
<td>#538</td>
<td>Finding objects in scans and despeckling</td>
<td>✔️</td>
<td>tjlang</td>
<td>Jun 9</td>
</tr>
<tr>
<td>#511</td>
<td>GeoTIFF output</td>
<td>✔️</td>
<td>tjlang</td>
<td>Apr 2</td>
</tr>
<tr>
<td>#464</td>
<td>ENH: Default field specific colormaps and luminosity when plotting</td>
<td>✔️</td>
<td>jjhelmus</td>
<td>Feb 6</td>
</tr>
<tr>
<td>#459</td>
<td>ENH: variational KDP retrievals</td>
<td>✔️</td>
<td>kirknorth</td>
<td>Mar 30</td>
</tr>
<tr>
<td>#456</td>
<td>Add radar_utils</td>
<td>✔️</td>
<td>nguy</td>
<td>Jan 24</td>
</tr>
<tr>
<td>#453</td>
<td>Time-height displays</td>
<td>✔️</td>
<td>nguy</td>
<td>Jan 20</td>
</tr>
</tbody>
</table>
Thank you!

www.arm.gov

arm-doe.github.io/pyart

scollis@anl.gov

@cyclogenesis_au

Argonne National Laboratory’s work was supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research (OBER), under Contract DE-AC02-06CH11357. The work has also been supported by the OBER of the DOE as part of the ARM Program.