

## **Corrected Moments in Antenna Coordinates (CMAC) X-SAPR Technical Report**

SM Collis  
ZS Sherman

JJ Helmus  
RC Jackson

November 2022



## **DISCLAIMER**

This report was prepared as an account of work sponsored by the U.S. Government. Neither the United States nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

## **Corrected Moments in Antenna Coordinates (CMAC) X-SAPR Technical Report**

SM Collis, Argonne National Laboratory (ANL)  
Principal Investigator

JJ Helmus, ANL (now at Anaconda, Inc.)  
ZS Sherman, ANL  
RC Jackson, ANL  
Co-Investigators

November 2022

How to cite this document:

Collis, SM, JJ Helmus, ZS Sherman, and RC Jackson. 2022. Corrected Moments in Antenna Coordinates (CMAC) X-SAPR Technical Report. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-TR-283.

Work supported by the U.S. Department of Energy,  
Office of Science, Office of Biological and Environmental Research

## **Executive Summary**

In 2010 the Atmospheric Radiation Measurement user facility procured 3- and 5-cm wavelength radars for documenting the macrophysical, microphysical, and dynamical structure of precipitating systems. In order to maximize the scientific impact, the program supported the development of an application chain to correct for various phenomena in order to retrieve the “point” values of moments of the radar spectrum and polarimetric measurements. This report details the motivation, science, and progress to date as well as charting a path forward.

## **Acknowledgments**

This work would not have been possible without the support and patience of the scientific community. We specifically want to thank Scott Giangrande for his contributions of the LP code, all the radar mentors, and specifically Nitin Bharadwaj for his tireless support. Kai Muhlbauer was instrumental in accelerating the LP method to the point of usability. We also acknowledge Jordi Figueras e Ventura for providing the differential attenuation correction.

## Acronyms and Abbreviations

2D	two-dimensional
4DD	four-dimensional dealiasing
ARM	Atmospheric Radiation Measurement
CACTI	Cloud, Aerosol, and Complex Terrain Interactions
CMAC	Corrected Moments in Antenna Coordinates
CSAPR	C-band Scanning ARM Precipitation Radar
DOI	Digital Object Identifier
INTERPSONDE	Interpolated Sonde Value-Added Product
KAZR	Ka-band ARM Zenith Radar
LP	linear programming
MC3E	Mid-Latitude Convective Continental Clouds Experiment
MMCR	millimeter wavelength cloud radar
NASA	National Aeronautics and Space Administration
NBF	non-uniform beam filling
NCP	normalized coherent power
NSA	North Slope of Alaska
PPI	plan position indicator
Py-ART	Python ARM Radar Toolkit
RSL	Radar Software Library
SGP	Southern Great Plains
SQI	signal quality index
XSAPR	X-band Scanning ARM Precipitation Radar

## Contents

Executive Summary .....	iii
Acknowledgments.....	iv
Acronyms and Abbreviations .....	v
1.0 Introduction .....	1
2.0 Application Chain.....	2
2.1 Calculations Performed to Aid Identification of Scatterers at Gate.....	3
2.2 Fuzzy-Logic-Based Identification of Scatterers at Gate .....	5
2.3 Corrections and Retrievals .....	6
2.3.1 Dealiasing.....	7
2.3.2 Filtering of Measured Phase Shift between Vertical and Horizontal Polarization.....	7
2.3.3 Retrieval of Specific Attenuation.....	8
2.3.4 $Z_{dr}$ Bias and Differential Attenuation Correction.....	9
3.0 Cases of Interest at SGP .....	10
4.0 Cases of Interest at NSA.....	11
5.0 Challenges .....	12
6.0 Future Work.....	13
7.0 Output Data Format .....	13
8.0 References .....	28

## Figures

1 The application chain for Corrected Moments in Antenna Coordinates.....	2
2 Illustration of the concept of a moving filter over range gates of adjacent rays.....	3
3 Calculations of texture of radial velocity from the X-band Scanning ARM Precipitation Radar (X-SAPR) using circular statistics to avoid false texture on folds.....	4
4 Histogram of texture values for multiple volumes over a week.....	5
5 Highest score-determined categories with hard constraints for the dominant scattering process for each gate from the XSAPR alongside the (clockwise) reflectivity factor, texture, and cross-correlation ratio. ....	6
6 Raw and dealiased radial velocities from the ARM XSAPR radar collected from the NSA site. ....	7
7 A single radial of data from CSAPR highlighting the LP technique. Raw $\Psi_{DP}$ is shown in green, retrieved $\Phi_{DP}$ in black, $K_{DP}$ in red, and reflectivity (divided by 10) in blue.....	8
8 Reflectivity as measured by the radar and disdrometer offset-adjusted, attenuation-corrected reflectivity with the significant feature detection mask applied.....	9
9 An example of uncorrected $Z_{dr}$ and $Z_{dr}$ corrected for bias and differential attenuation from a PPI scan taken from the XSAPR at the ARM NSA site. ....	9

10 Convective cells heading within the vicinity of the XSAPR I6 on 1950 UTC 24 May 2018. .... 10

11 Storms passing within the I4 and I6 dual Doppler lobes while the radars were operating in dual  
Doppler mode on 855 UTC 14 July 2018. .... 10

12 Mesoscale convective snow bands with unusual movement at the ARM NSA site at 137 UTC 23  
November 2019. .... 11

13 Synoptic snow system southeast of the radar at the ARM NSA site on 1819 UTC 3 August 2020. ... 12

14 Synoptic snow north of the radar at the ARM NSA site at 2349 UTC 10 September 2020. .... 12

## Tables

1 Inputs for trapezoidal membership functions for various classes..... 5

## 1.0 Introduction

The U.S. Department of Energy’s Atmospheric Radiation Measurement user facility (ARM; Mather and Voyles 2012) has a long history of sensing clouds in the column using the millimeter wavelength cloud radar (MMCR, now Ka-Band ARM Zenith Radar or KAZR). Starting in 2010, ARM embarked on a program to better characterize the domain surrounding the column using scanning radars at millimeter and centimeter wavelengths. Processing for the MMCR and KAZR has been previously published (Kollias et al. 2013). The focus of Corrected Moments in Antenna Coordinates (CMAC) is for processing the data from the ARM X-band and C-band Scanning Precipitation Radars (X/CSAPRs). Due to the agility and lower cost of the XSAPR and CSAPRs, ARM opted not to operate the common wavelength of 10 cm (S-band), which is robust to liquid water path attenuation in all but the most severe storms. This necessitates the development of robust code for the correction of issues due to the scattering and attenuation during the two-way propagation of the radar through liquid water drops. In addition, the tradeoff between wavelength, maximum unambiguous range, and Doppler Nyquist velocity ( $V_{nyq}$ ) means the XSAPR and CSAPR alias at 12.4 and 16.52 m s<sup>-1</sup> when operating in a baseline mode (such as during the Mid-Latitude Convective Continental Clouds Experiment [MC3E; Jensen et al. 2015] and Cloud, Aerosol, and Complex Terrain Interactions [CACTI]). Due to extreme velocities of scatterers aloft and, in places such as Oklahoma with intense convection, aliasing is common and requires post-moment calculation dealiasing. There are many techniques for dealiasing Doppler velocities (e.g., James and Houze 2001). However, on testing we found these techniques to be either difficult to implement in an operational chain or lacking in robustness. When we first attempted to build a processing chain, each step made its own decision on where to conditionally run based on various measurements of “quality” such as the co-polar (zero lag) correlation coefficient  $\rho_{HV}$  and normalized coherent power (NCP, also referred to as signal quality index or SQI). These are defined as:

$$\rho_{HV}(0) = \frac{|S_{vv}S_{hh}^*|}{\sqrt{\langle |S_{hh}|^2 \rangle \langle |S_{vv}|^2 \rangle}} \quad (1)$$

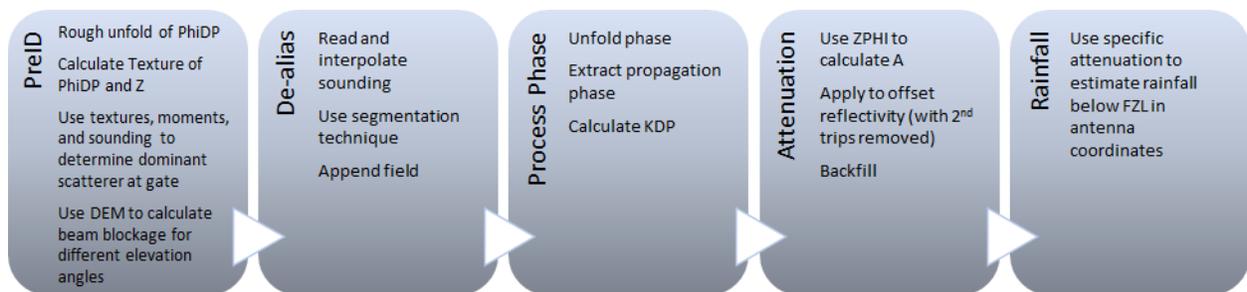
$$NCP = P_{coh}P_{DC} \quad (2)$$

Where the S terms are elements of the scattering matrix, P<sub>coh</sub> is the coherent part of the Doppler spectrum and P<sub>DC</sub> is the incoherent part. Since ARM radars use magnetron transmitters, the phase is randomized from pulse to pulse. So, when a first-trip return is mixed with a return from a scatterer beyond the maximum unambiguous range, the derived radar Doppler spectrum when averaged over many pulses is flat and the NCP is low. While the Doppler spectrum from a first trip has structure from which (depending on the method) a peak can be found and the Doppler velocity determined, the NCP approaches 1.0. However, the usefulness of NCP alone in second-trip detection breaks down in regions of high spectral width. When the spectral width approaches the  $V_{nyq}$ , even in areas of purely first trip, the NCP decreases. This is especially troublesome in regions of high convergence and divergence in convective storms, often causing false flagging of these regions. To overcome the issues of arbitrary decision making and faults in using NCP alone to detect multiple trips, our application chain, Corrected Moments in Antenna Coordinates, first attempts to identify the nature of the scattering medium at the gate. This gate-ID is performed before any corrections are applied so it is indifferent to hydrometeor identification codes (e.g., Dolan and Rutledge 2009, Wen et al. 2015, Al-Sakka et al. 2013, etc.) that seek to gain microphysical insight. Gate ID is performed for the purpose of objectively determining where

future algorithms should be applied. Since we are implementing CMAC using the Python-ARM Radar Toolkit (Py-ART; Heistermann et al. 2014, Helmus and Collis 2016) we can use the identifications to construct a gate filter.

## 2.0 Application Chain

Many algorithms exist in the scientific literature for the quality control and correction of radar data. However, given Py-ART's data model-driven approach, it is possible to design an application chain that is highly modular and task based. Each component has a particular job and can be replaced as better algorithms are published (and, ideally, code-shared). As stated in the previous section, the overarching idea behind CMAC is that a gate ID is created and determines the conditional application of algorithms. At the time of writing, implemented classes are: rain, melting layer, ice, second trip, terrain blockage, and no significant scatterer. Dealiasing, for example, would run on the set of all classes except "no significant return" while retrievals of specific attenuation would run on the class of "rain."



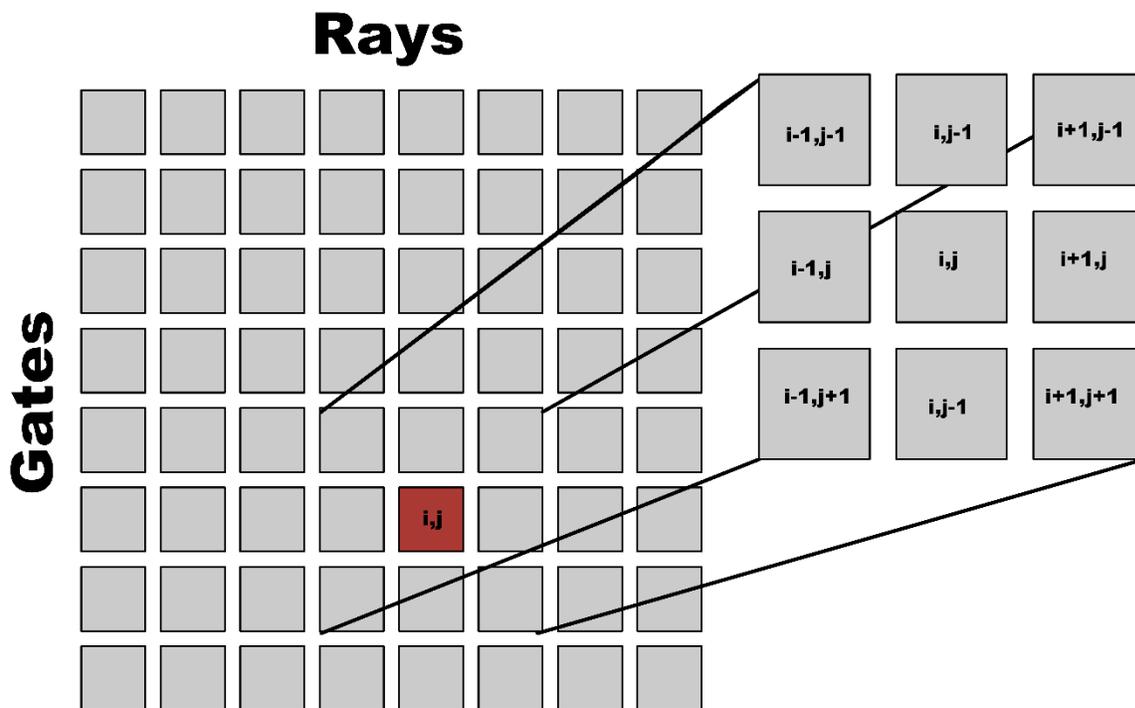
**Figure 1.** The application chain for Corrected Moments in Antenna Coordinates.

This approach requires that the gate ID is run on the pre-corrected data. However, as discussed in Section 1, radar-provided measurements alone are not sufficient to constrain the problem of gate ID, especially the identification of multiple trips. We can generate several pre-ID retrievals and inputs to constrain the problem, however, as described in Section 2.1. The Application chain for CMAC is shown in Figure 1 and can be broken down to:

- Pre-ID calculations of texture and mapping sounding data to radar gates
- Ascribing membership functions to gate classes, scoring of gates, and classification at the gate of predominant scatterer
- Dealiasing of Doppler velocities
- Extraction of propagation differential polarimetric phase from instrument-measured differential polarimetric phase
- Calculation of specific differential phase
- Calculation of specific attenuation
- Integration and application to reflectivity
- Calculation of rain rate for liquid precipitation using specific attenuation.

## 2.1 Calculations Performed to Aid Identification of Scatterers at Gate

There are two steps to add information in order to determine the dominant scattering process at the gate: mapping temperature to gate locations and using texture of radial velocity as a discriminant of significant scattering. Since Py-ART already ascribes a Cartesian displacement from the radar for each gate using a simple  $\frac{4}{3}R_e$  standard atmosphere propagation model, CMAC simply interpolates sonde data available from ARM soundings (via the Interpolated Sonde Value-Added Product: [INTERPSONDE](#)). The concept behind using the texture of the radial velocity is that when second trips (or no trips) dominate, due to the pulse-to-pulse randomized phase of a magnetron transmitter, radial velocity should vary, from gate to gate, between  $(-V_{nyq}, V_{nyq})$  randomly. As long as there is some structure to the radar Doppler spectrum, the signal processor should be able to pick a peak and determine the first moment, being the radial velocity. Thus, the gate-to-gate and azimuth-to-azimuth variation, or texture of Doppler velocity, should be able to act as a good discriminator of significant returns. The abstract concept is for a central pixel  $(i, j)$ , the points surrounding the pixel in a  $n$  by  $m$  kernel are collected as shown in Figure 2. Then the variance is calculated on the set of points and is returned as the  $(i, j)^{\text{th}}$  value in the resultant 2D (range, time/azimuth) array.



**Figure 2.** Illustration of the concept of a moving filter over range gates of adjacent rays. The center element,  $(i, j)$  is calculated by passing surrounding elements. The footprint of the surrounding elements is determined by the kernel. In many cases we use a 3x3 kernel.

The challenge comes from the desire to calculate this precorrection. Doppler folding will generate a significant signal in the texture field if done purely on radial velocity values. However, projection of radial velocity values onto a unit circle allows a smooth transition from  $(-V_{nyq}, V_{nyq})$  and there is a branch of mathematics dealing with the statistics of directions and magnitudes known as directional statistics

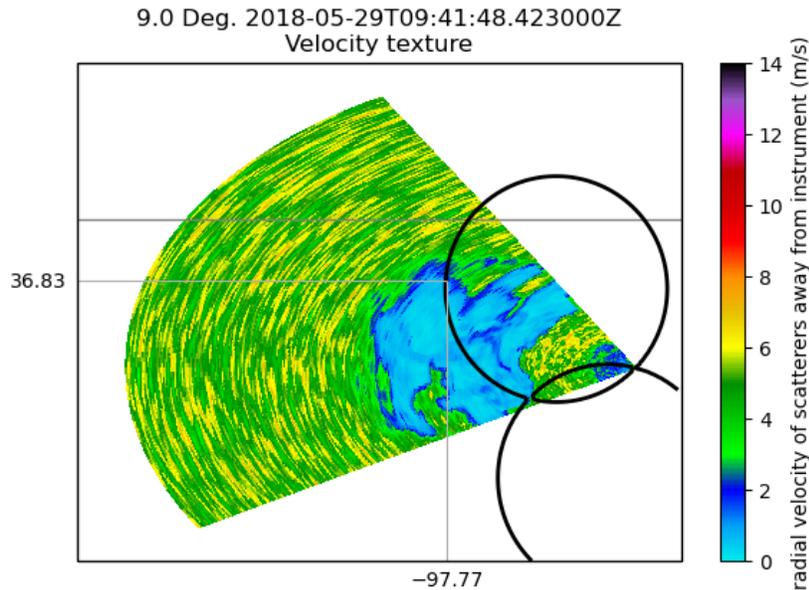
(Wikipedia 2016). Radial velocity values from the positive to the negative Nyquist velocity are projected onto a circle with  $\theta$  from 0 to  $\pi$  and the standard deviation is given by:

$$x = \cos \theta \quad (3)$$

$$y = \sin \theta \quad (4)$$

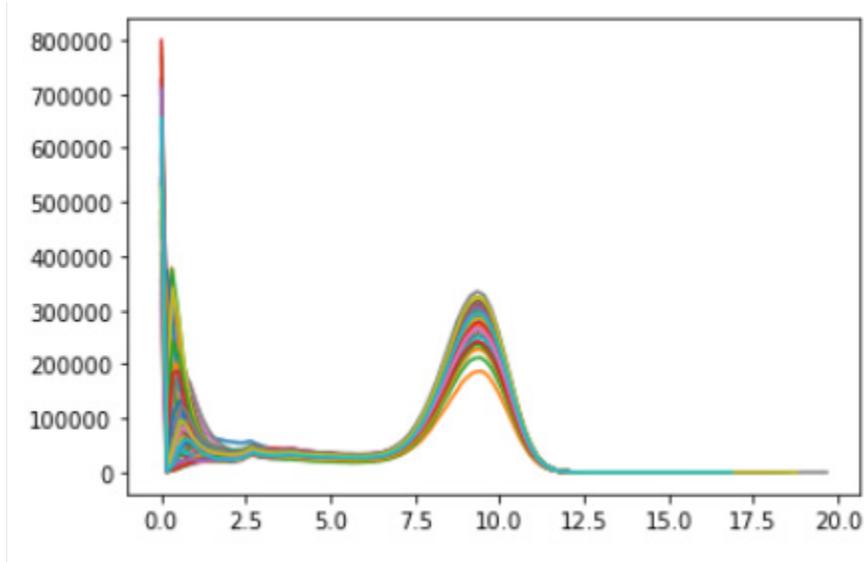
$$R = \sqrt{x_2 + y_2} \quad (5)$$

$$S = \sqrt{-2 * \log \log R} \quad (6)$$



**Figure 3.** Calculations of texture of radial velocity from the X-band Scanning ARM Precipitation Radar (XSAPR) using circular statistics to avoid false texture on folds.

There are clearly higher values of texture where there are no significant returns while texture falls quickly over the precipitation echo boundaries. However, the exact values of texture to be used in the membership function to delineate between significant and non-significant will depend on many factors that influence texture, including number of samples and signal-to-noise ratio. Plotting a histogram of texture values yields two distinctly separated populations of gates. To find the discrimination point we use Scientific Python’s (Jones et al. 2001) continuous wavelet transform-based peak-finding algorithm (Du et al. 2006) to find the location of the left and right peak. The cut off is then decided by finding the minimum value, or valley, between the two peaks. Ad hoc testing shows this to be robust even when changing radar types. We tested with X, C, and Ka-band radars, all using different configurations.



**Figure 4.** Histogram of texture values for multiple volumes over a week. The left peak corresponds to significant returns, the right to noise.

## 2.2 Fuzzy-Logic-Based Identification of Scatterers at Gate

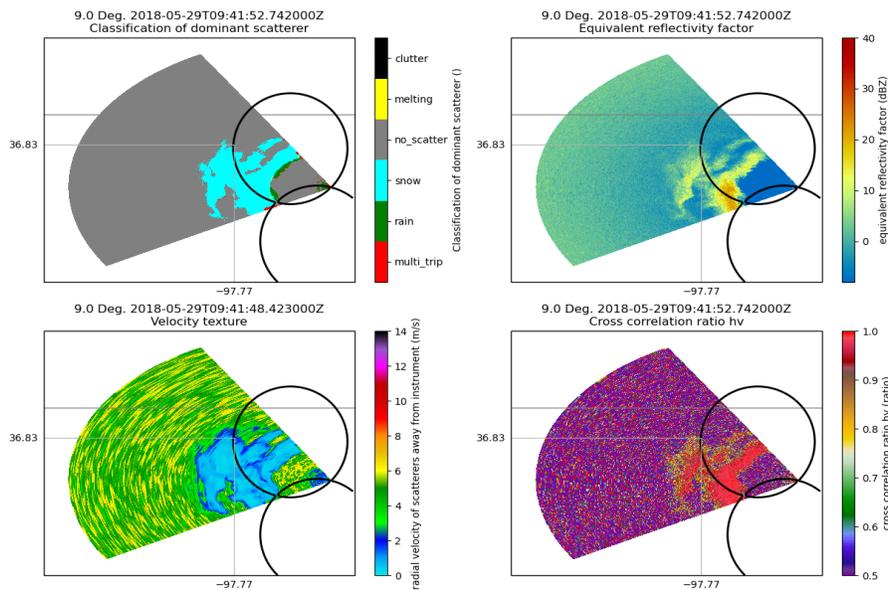
After calculating the temperature and texture of radial velocity, the next step is to identify the dominant scatterer at each gate to help CMAC choose which correction algorithms to use. While fuzzy logic has been used for particle identification extensively, few investigators have done this as a first step (pre-Kdp, etc.). A notable exception is work by Gourley et al. (2007). Preprocessing ID depends on using the moments and derived products, assuming they contain all the issues associated with unprocessed data. We use a simple scheme that associates a membership with each classification of: Melting layer, Multi-trip, Rain, and Snow. We have future plans to include gates that are contaminated by hail in the propagation path. Membership functions are shown in Table 1. At the moment, with the exception of texture, these are determined using trial and error. As we have set up a robust codebase using Py-ART and Scikit Fuzzy, we can revisit the membership functions at any time using better formulations determined using machine learning and other techniques.

**Table 1.** Inputs for trapezoidal membership functions for various classes.

Class	Texture (m/s)	$\rho_{HV}$	NCP	Temperature (C)	Height (km)	SNR (dB)
Melting	[0, 0, 2.0, 2.1], 0	[0.6, 0.65, 0.9, 0.96], 2.0	[0.4, 0.5, 1, 1], 0	[0, 0.1, 2, 4], 4	[0, 0, 25, 25], 0	[8, 10, 1000, 1000], 0
Multi trip	[2.0, 2.1, 130, 130], 4	[0.5, 0.7, 1, 1], 0	[0, 0, 0.5, 0.56], 1	[-100, -100, 100, 100], 0	[0, 0, 5, 8], 0	[5, 10, 1000, 1000], 1
Rain	[0, 0, 2.0, 2.1], 1	[0.97, 0.98, 1, 1], 1	[0.4, 0.5, 1, 1], 1	[2, 5, 100, 100], 2	[0, 0, 5, 6], 0	[8, 10, 1000, 1000], 1
Snow	[0, 0, 2.0, 2.1], 1	[0.65, 0.9, 1, 1]	[0.4, 0.5, 1, 1], 1	[-100, -100, 0.5, 4], 2	[0, 0, 25, 25], 0	[8, 10, 1000, 1000], 1
No Scatter	[2.0, 2.1, 330, 330], 2	[0, 0, 0.1, 0.2], 0	[0, 0, 0.1, 0.2], 0	[-100, -100, 100, 100], 0	[0, 0, 25, 25], 0	[-100, -100, 5, 10], 4

Figure 5 shows an example of scatterers at gate identification from a plan position indicator (PPI) tilt from the XSAPR at the ARM North Slope of Alaska (NSA) site as an organized convective system passed over the mobile site. Regions of snow scatterers are shown in cyan, rain in green (if present, but at NSA unlikely), multi-trip in red, mixed scattering in yellow (e.g., melting layer), beam blockage in brown, and no significant return in grey. Work is proceeding on determining if a radial is hail contaminated as is work on clutter identification and tagging.

Gate, or scatter ID, is used to form Py-ART Gatefilter objects that can be passed to a subsequent processing algorithm. For example, Linear programming (see Section 2.3.2) filtering of  $\phi_{DP}$  would be performed on gates identified as rain and attenuation correction (offset) on the union of rain, melting layer, and snow.



**Figure 5.** Highest score-determined categories with hard constraints for the dominant scattering process for each gate from the XSAPR alongside the (clockwise) reflectivity factor, texture, and cross-correlation ratio. These values will be used to determine what post-processing will be applied gate to gate.

## 2.3 Corrections and Retrievals

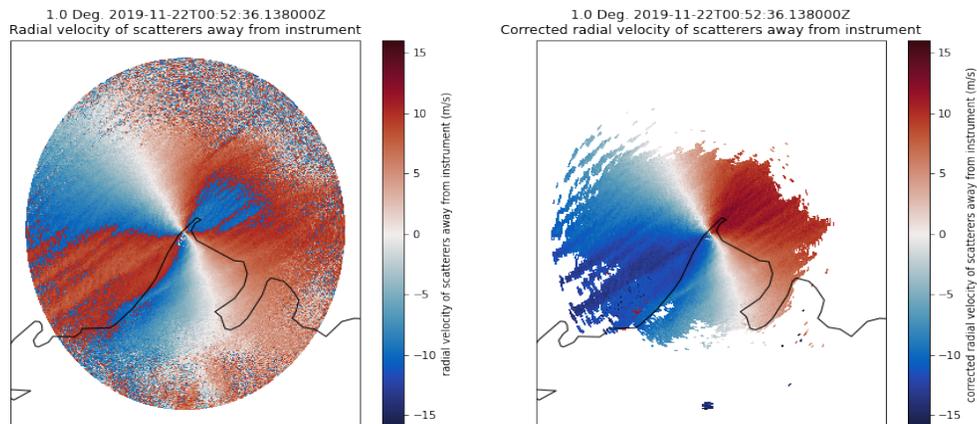
To have the greatest impact for stakeholders, the ARM radars need to provide high-quality, calibrated and corrected moments and measurements. By measurements we mean the intrinsic value, or the measurement corrected for all the issues of propagation and processing. In CMAC, this means:

- Dealiased Doppler velocities
- $\phi_{DP}$  corrected for non-uniform beam filling and phase shift on backscatter
- Specific differential phase  $K_{DP}$
- Specific attenuation
- Reflectivity corrected for liquid water path attenuation.

### 2.3.1 Dealiasing

Originally the Four Dimensional Dealiasing (4DD; James and Houze 2001) algorithm was wrapped into Py-ART using the National Aeronautics and Space Administration (NASA)’s Radar Software Library (RSL). Issues with the implementation of the paper into code led to a long discussion on issues in dealiasing (<https://github.com/ARM-DOE/pyart/issues/119>). Discussions led to two new solutions in Doppler velocity unfolding: fringe pattern-based and region-based dealiasing. Unlike the dealiasing of cloud radar data where it can be assumed scatterers move purely with the wind, dynamics creates radial velocity patterns that can move counterflow. The fringe or “phase-based” technique is an image analysis technique designed for removing fringe patterns from interferometric images. Early tests were sub-par and while the technique is added to Py-ART, it is rarely used. The region-based technique performs Doppler velocity dealiasing by finding regions of similar velocities and unfolding and merging pairs of regions until all regions are unfolded. Unfolding and merging regions is accomplished by modeling the problem as a dynamic network reduction. Figure 6 shows raw and unfolded radial velocities from the ARM XSAPR radar collected at the ARM NSA site. Unfolding was performed using the region-based technique. Even after unfolding, some velocities might be off by an integer factor of the Nyquist velocity. Therefore, a second step finds the integers  $n_i$  that minimize the cost function  $J$  given by Equation 7 related to the difference between the mean velocity field  $V_i$  of each region and winds from a rawinsonde  $V_{sounding}$ .

$$J = \sum_{i \in \text{regions}} n_i V_{nyq} V_i - V_{sounding} \quad (7)$$



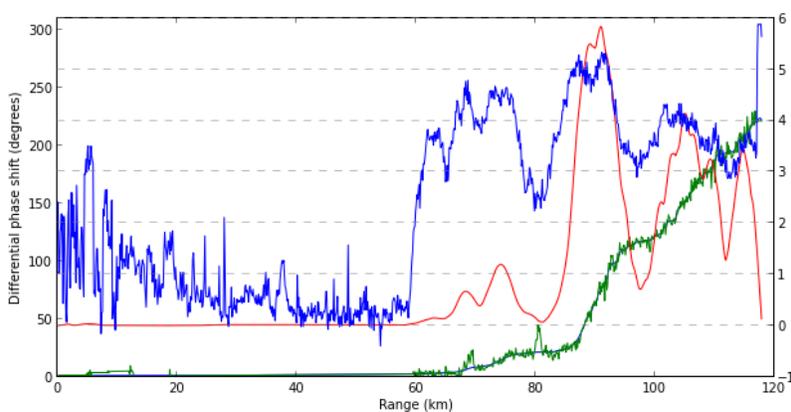
**Figure 6.** Raw and dealiased radial velocities from the ARM XSAPR radar collected from the NSA site. Unfolding was performed using the region-based approach.

### 2.3.2 Filtering of Measured Phase Shift between Vertical and Horizontal Polarization

Raw polarimetric phase shift  $\Psi_{DP}$  can be broken down into a component due to differential liquid water path ( $\Phi_{DP}$ ) and other, specific terms, due to non-uniform beam filling (NBF) and phase shift on backscatter ( $\delta$ ). Mathematically:

$$\Psi_{DP} = \Phi_{DP} + \delta + NBF \quad (8)$$

See Giangrande and Ryzhkov (2008) and references therein. In order to extract microphysical insight into the liquid (precipitating) liquid water path, it is desirable to retrieve  $\Phi_{DP}$  from the measured signal. Taking advantage of the fact that liquid water content cannot be negative and therefore we expect  $\Phi_{DP}$  to strictly increase, we can construct a filter to extract  $\Phi_{DP}$  from  $\Psi_{DP}$ . Giangrande et al. (2013) outline an objective technique that uses linear programming (LP, e.g., Helbush 1968) to create a  $\Phi_{DP}$  that is piecewise increasing and (importantly) is non-biased. That is, given a  $\Psi_{DP}$  that contains a smoothly increasing signal and a short-term variation, the algorithm will fit through the base rather than the midpoint or peak of the variation. The strength of the fit is influenced by the local reflectivity as a weak constraint. Where reflectivity is high, positivity in the gradient of  $\Phi_{DP}$  is enforced (see Giangrande and Ryzhkov 2008 for details). Once  $\Phi_{DP}$  is retrieved, the specific differential phase  $K_{DP}$  is retrieved by convoluting  $\Phi_{DP}$  with a 20-point linear ramp (a Sobel filter). This is similar in nature and ad hoc experimentation shows it to closely mimic a moving linear fit similar to that used in Bringi et al. (2002).

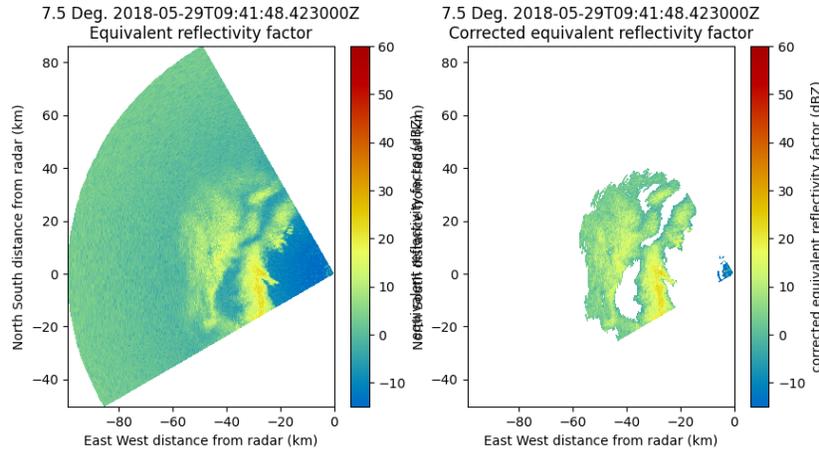


**Figure 7.** A single radial of data from CSAPR highlighting the LP technique. Raw  $\Psi_{DP}$  is shown in green, retrieved  $\Phi_{DP}$  in black,  $K_{DP}$  in red, and reflectivity (divided by 10) in blue.

Figure 7 shows a single radial of data from a CSAPR highlighting the LP technique. Raw  $\Psi_{DP}$  is shown in green, retrieved  $\Phi_{DP}$  in black,  $K_{DP}$  in red, and reflectivity (divided by 10) in blue. The retrieved  $\Phi_{DP}$  is monotonically increasing resulting in a strictly positive  $K_{DP}$ . This same method can then be applied to the XSAPR at the NSA site. LP optimization is achieved by using the CoinLP library. Initially PyGLPK was used, but with a very welcome contribution to Py-ART from Kai Muhlbauer from the University of Bonn, switching to CoinLP reduced volume processing time from 8 minutes to under a minute.

### 2.3.3 Retrieval of Specific Attenuation

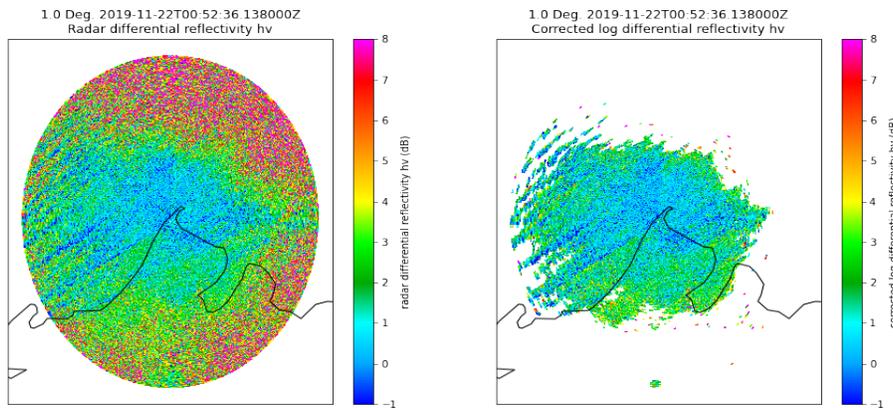
Specific attenuation  $A$  was retrieved using an adaptation of an iterative “hotspot” method as outlined in Gu et al. (2011). Using the aforementioned gate ID, a gate filter is constructed that only calculates  $A$  in regions of liquid precipitation assuming attenuation due to ice is negligible and in mixed-phase regions intractable. Occasionally clutter can throw off the  $\phi_{dp}$  calculation, which becomes apparent in the  $K_{dp}$  and  $A$  fields. In order to mitigate this, we can filter out all  $K_{dp}$  greater than  $15^\circ \text{ km}^{-1}$ .



**Figure 8.** Reflectivity as measured by the radar and disdrometer offset-adjusted, attenuation-corrected reflectivity with the significant feature detection mask applied.

Prior to the application of the attenuation correction, we apply a reflectivity offset. Figure 8 shows the original reflectivity as produced by the radar on the left and the scaled and then corrected reflectivity on the right.

### 2.3.4 $Z_{dr}$ Bias and Differential Attenuation Correction



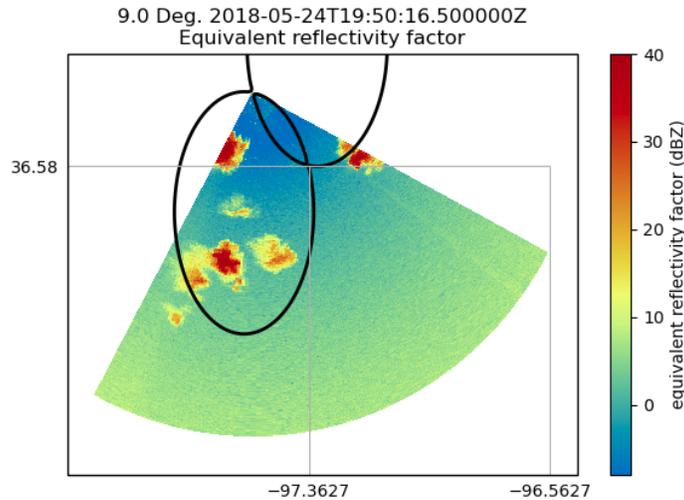
**Figure 9.** An example of uncorrected  $Z_{dr}$  and  $Z_{dr}$  corrected for bias and differential attenuation from a PPI scan taken from the XSAPR at the ARM NSA site.

In addition,  $Z_{dr}$  can also be affected by differential attenuation of the radar beam at C- and X-band wavelengths. Therefore, in addition to calculating specific attenuation, specific differential attenuation is also retrieved using the method from Gu et al. (2011) using code contributed by Jordi Figueras e Ventura from MeteoSwiss. Figure 9 shows the uncorrected  $Z_{dr}$  data on the left and the corrected  $Z_{dr}$  on the right for a 1.0-degree elevation PPI scan from the XSAPR at the ARM NSA site.

Significant  $Z_{dr}$  biases were observed from all XSAPR radars at the ARM SGP site during the period from May to September 2018. These biases varied between individual scans, preventing us from determining a  $Z_{dr}$  calibration that could be confidently applied to the data. Therefore, for the data at the

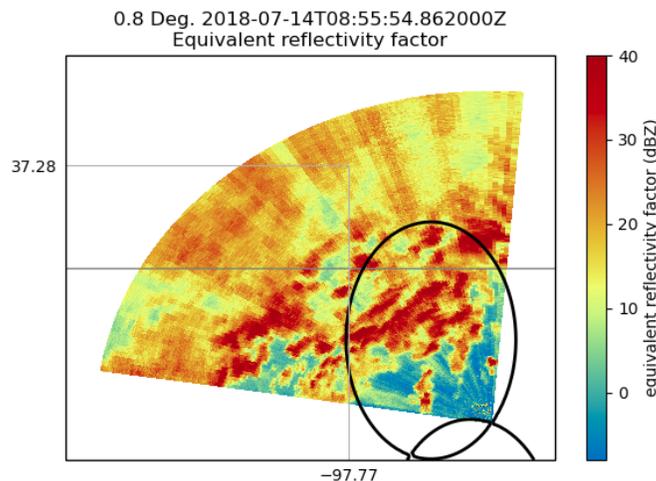
ARM SGP site, all corrected  $Z_{dr}$  data are masked and therefore not available. We include the raw  $Z_{dr}$  for analysis, but urge great caution in interpreting these  $Z_{dr}$  data.

### 3.0 Cases of Interest at SGP



**Figure 10.** Convective cells heading within the vicinity of the XSAPR I6 on 1950 UTC 24 May 2018.

During the SGP summer scanning experiment, two different scanning configurations were used: one that is favorable for retrieving the microphysical properties of cloud updrafts, and another that is more optimal for wind retrievals. While the radars were in the microphysics scanning strategy on 24 May 2018, scattered convective cells passed over the area providing an opportunity for studying microphysical processes in these developing updrafts.

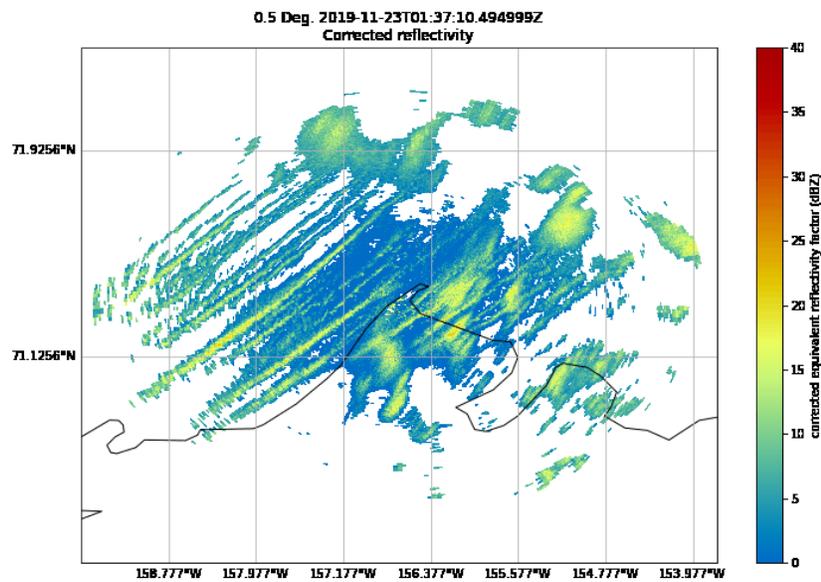


**Figure 11.** Storms passing within the I4 and I6 dual Doppler lobes while the radars were operating in dual Doppler mode on 855 UTC 14 July 2018.

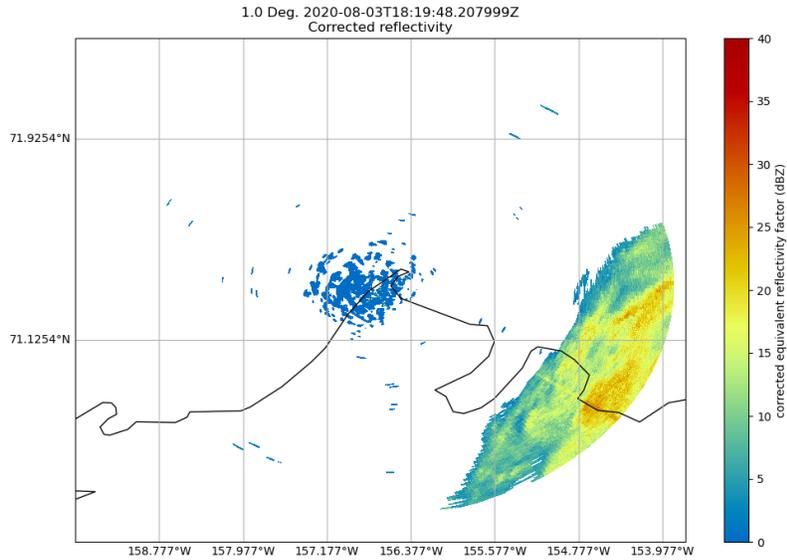
On 14 July 2018, a mesoscale convective complex approached the SGP site with convective cells present within the dual Doppler lobes of the XSAPR I4 and I6 radars. This case can provide good testing of dual Doppler wind retrievals over the area.

## 4.0 Cases of Interest at NSA

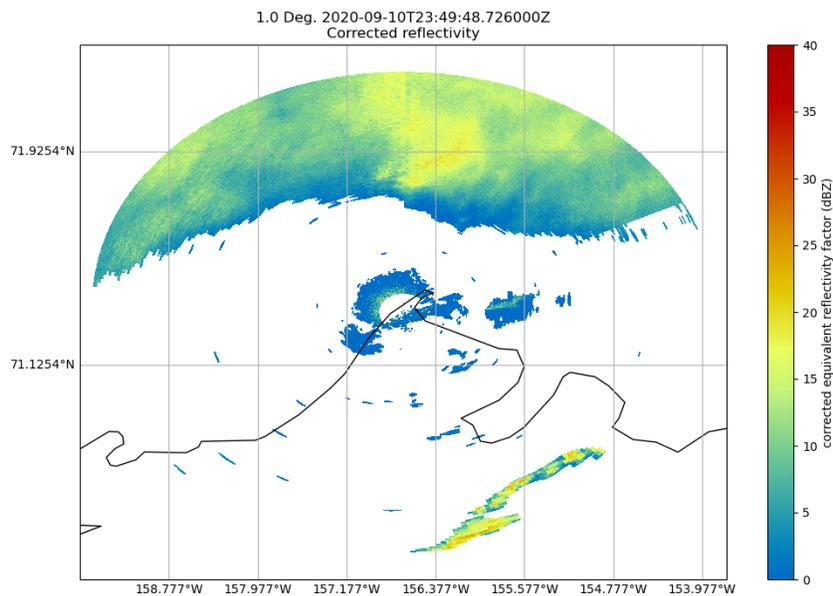
There are a couple of cases of interest, one particularly so. Figure 12 is quite a remarkable case as it contains unusual mesoscale convective snow bands on 23 November 2019. The motion is also very interesting, making that case of particular interest to those studying the extreme weather systems of the Arctic. Figures 13 and 14 are two synoptic systems, one over the sea on 3 August 2020 and the other over land on 10 September 2020.



**Figure 12.** Mesoscale convective snow bands with unusual movement at the ARM NSA site at 137 UTC 23 November 2019.



**Figure 13.** Synoptic snow system southeast of the radar at the ARM NSA site on 1819 UTC 3 August 2020.



**Figure 14.** Synoptic snow north of the radar at the ARM NSA site at 2349 UTC 10 September 2020.

## 5.0 Challenges

Initial robustness tests show we have a lot of work to do in the detection and tagging of clutter returns. Tests of the procedures on XSAPR data show an impact from clutter near the radar as well as some clutter returns further out from the radar. We were able to remove much of the pipeline clutter returns near the radar. The other challenge is in the software engineering of the LP method. It has been discovered that in

regions of extended  $\delta_{dp}$  the LP technique as it is in Giangrande and Ryzhkov (2008) underperforms. The authors have a nice solution that is difficult to implement with the currently supported LP packages. We are still actively working on this issue.

## 6.0 Future Work

Part of the future work will be to include a way to handle differing scan strategies in the same radar. CMAC is designed for this; however, the NSA data changed between every other file, which caused every other file's gate identification to be off. This data was rerun and fixed. Running this data was challenging as it is an extreme environment, but by learning, we can apply this now to future data sets of similar extremes. We would also like to improve the filtering of sea ice as well as stationary objects such as oil pipelines. In addition, some adjustments to the fuzzy logic algorithm to better distinguish between noise and second-trip echoes is required.

## 7.0 Output Data Format

```
netcdf sgpxsapremacsecI6.c1.20180607.235811 {
```

```
dimensions:
```

```
    time = UNLIMITED ; // (1092 currently)
```

```
    range = 1001 ;
```

```
    sweep = 12 ;
```

```
    string_length = 32 ;
```

```
    bound = 2 ;
```

```
variables:
```

```
    double time(time) ;
```

```
    time:units = "seconds since 2018-06-07T23:58:05Z" ;
```

```
    time:calendar = "gregorian" ;
```

```
    time:long_name = "Time offset from midnight" ;
```

```
    time:standard_name = "time" ;
```

```
    time:comment = "Coordinate variable for time. Time at the center of each ray, in fractional seconds since the global variable time_coverage_start" ;
```

```
    time:bounds = "time_bounds" ;
```

```
float range(range) ;

range:long_name = "range_to_measurement_volume" ;

range:units = "meters" ;

range:standard_name = "projection_range_coordinate" ;

range:axis = "radial_range_coordinate" ;

range:spacing_is_constant = "true" ;

range:comment = "Coordinate variable for range. Range to center of each bin." ;

range:meters_to_center_of_first_gate = 0.f ;

range:meters_between_gates = 100.f ;

range:bounds = "range_bounds" ;

float azimuth(time) ;

azimuth:long_name = "azimuth_angle_from_true_north" ;

azimuth:standard_name = "beam_azimuth_angle" ;

azimuth:axis = "radial_azimuth_coordinate" ;

azimuth:comment = "Azimuth of antenna relative to true north" ;

azimuth:units = "degree" ;

float elevation(time) ;

elevation:long_name = "elevation_angle_from_horizontal_plane" ;

elevation:standard_name = "beam_elevation_angle" ;

elevation:axis = "radial_elevation_coordinate" ;

elevation:comment = "Elevation of antenna relative to the horizontal plane" ;

elevation:units = "degree" ;

float total_power(time, range) ;

total_power:_FillValue = -9999.f ;

total_power:long_name = "Total power" ;
```

```
total_power:units = "dBZ" ;
total_power:standard_name = "equivalent_reflectivity_factor" ;
total_power:coordinates = "elevation azimuth range" ;
float reflectivity(time, range) ;
reflectivity:_FillValue = -9999.f ;
reflectivity:long_name = "Reflectivity" ;
reflectivity:units = "dBZ" ;
reflectivity:standard_name = "equivalent_reflectivity_factor" ;
reflectivity:coordinates = "elevation azimuth range" ;
float velocity(time, range) ;
velocity:_FillValue = -9999.f ;
velocity:long_name = "Mean dopper velocity" ;
velocity:standard_name = "radial_velocity_of_scatterers_away_from_instrument" ;
velocity:coordinates = "elevation azimuth range" ;
velocity:units = "m/s" ;
float spectrum_width(time, range) ;
spectrum_width:_FillValue = -9999.f ;
spectrum_width:long_name = "Doppler spectrum width" ;
spectrum_width:standard_name = "doppler_spectrum_width" ;
spectrum_width:coordinates = "elevation azimuth range" ;
spectrum_width:units = "m/s" ;
float differential_reflectivity(time, range) ;
differential_reflectivity:_FillValue = -9999.f ;
differential_reflectivity:long_name = "Differential reflectivity" ;
differential_reflectivity:units = "dB" ;
```

```
differential_reflectivity:standard_name = "log_differential_reflectivity_hv" ;
differential_reflectivity:coordinates = "elevation azimuth range" ;
float specific_differential_phase(time, range) ;

specific_differential_phase:_FillValue = -9999.f ;

specific_differential_phase:standard_name = "specific_differential_phase_hv" ;
specific_differential_phase:coordinates = "elevation azimuth range" ;
specific_differential_phase:long_name = "Specific differential phase (KDP)" ;
specific_differential_phase:units = "degrees/km" ;

float cross_correlation_ratio(time, range) ;
cross_correlation_ratio:_FillValue = -9999.f ;
cross_correlation_ratio:standard_name = "cross_correlation_ratio_hv" ;
cross_correlation_ratio:valid_max = 1. ;
cross_correlation_ratio:valid_min = 0. ;
cross_correlation_ratio:coordinates = "elevation azimuth range" ;
cross_correlation_ratio:long_name = "Cross correlation ratio (RHOHV)" ;
cross_correlation_ratio:units = "1" ;

float normalized_coherent_power(time, range) ;
normalized_coherent_power:_FillValue = -9999.f ;
normalized_coherent_power:long_name = "Normalized coherent power" ;
normalized_coherent_power:standard_name = "normalized_coherent_power" ;
normalized_coherent_power:valid_max = 1. ;
normalized_coherent_power:valid_min = 0. ;
normalized_coherent_power:comment = "Also known as signal quality index (SQI)" ;
normalized_coherent_power:coordinates = "elevation azimuth range" ;
```

```
normalized_coherent_power:units = "1" ;  
float differential_phase(time, range) ;  
differential_phase:_FillValue = -9999.f ;  
differential_phase:long_name = "Differential phase (PhiDP)" ;  
differential_phase:units = "degrees" ;  
differential_phase:standard_name = "differential_phase_hv" ;  
differential_phase:valid_max = 180. ;  
differential_phase:valid_min = -180. ;  
differential_phase:coordinates = "elevation azimuth range" ;  
double ground_clutter(time, range) ;  
ground_clutter:long_name = "Ground Clutter" ;  
ground_clutter:units = "1" ;  
ground_clutter:notes = "0: No Clutter, 1: Clutter" ;  
ground_clutter:flag_values = 0L, 1L ;  
ground_clutter:flag_meanings = "No_Clutter Clutter" ;  
float sounding_temperature(time, range) ;  
sounding_temperature:long_name = "Interpolated profile" ;  
sounding_temperature:units = "degC" ;  
sounding_temperature:standard_name = "interpolated_profile" ;  
float height(time, range) ;  
height:long_name = "Height of radar beam" ;  
height:units = "m" ;  
height:standard_name = "height" ;  
float signal_to_noise_ratio(time, range) ;  
signal_to_noise_ratio:long_name = "Signal to noise ratio" ;
```

```
signal_to_noise_ratio:units = "dB" ;

signal_to_noise_ratio:standard_name = "signal_to_noise_ratio" ;

signal_to_noise_ratio:coordinates = "elevation azimuth range" ;

double velocity_texture(time, range) ;

velocity_texture:long_name = "Mean dopper velocity" ;

velocity_texture:units = "m/s" ;

velocity_texture:standard_name = "radial_velocity_of_scatterers_away_from_instrument" ;

velocity_texture:coordinates = "elevation azimuth range" ;

int64 gate_id(time, range) ;

gate_id:long_name = "Classification of dominant scatterer" ;

gate_id:notes = "0:multi_trip,1:rain,2:snow,3:no_scatter,4:melting,5:clutter" ;

gate_id:valid_max = 0L ;

gate_id:valid_min = 0. ;

gate_id:units = "1" ;

gate_id:flag_values = 0L, 1L, 2L, 3L, 4L, 5L ;

gate_id:flag_meanings = "multi_trip rain snow no_scatter melting clutter" ;

double simulated_velocity(time, range) ;

simulated_velocity:long_name = "Simulated mean doppler velocity" ;

simulated_velocity:units = "m/s" ;

simulated_velocity:standard_name = "radial_velocity_of_scatterers_away_from_instrument" ;

simulated_velocity:coordinates = "elevation azimuth range" ;

float corrected_velocity(time, range) ;

corrected_velocity:_FillValue = -9999.f ;

corrected_velocity:long_name = "Corrected mean doppler velocity" ;

corrected_velocity:units = "m/s" ;
```

```
corrected_velocity:standard_name =  
"corrected_radial_velocity_of_scatterers_away_from_instrument" ;  
  
corrected_velocity:coordinates = "elevation azimuth range" ;  
  
corrected_velocity:valid_min = -32.085 ;  
  
corrected_velocity:valid_max = 32.085 ;  
  
double unfolded_differential_phase(time, range) ;  
  
unfolded_differential_phase:_FillValue = -9999. ;  
  
unfolded_differential_phase:long_name = "Unfolded differential propagation phase shift" ;  
  
unfolded_differential_phase:standard_name = "differential_phase_hv" ;  
  
unfolded_differential_phase:valid_max = 180. ;  
  
unfolded_differential_phase:valid_min = -180. ;  
  
unfolded_differential_phase:coordinates = "elevation azimuth range" ;  
  
unfolded_differential_phase:units = "degree" ;  
  
double corrected_differential_phase(time, range) ;  
  
corrected_differential_phase:_FillValue = -9999. ;  
  
corrected_differential_phase:long_name = "Corrected differential propagation phase shift" ;  
  
corrected_differential_phase:standard_name = "differential_phase_hv" ;  
  
corrected_differential_phase:valid_max = 400. ;  
  
corrected_differential_phase:valid_min = 0. ;  
  
corrected_differential_phase:coordinates = "elevation azimuth range" ;  
  
corrected_differential_phase:units = "degree" ;  
  
double filtered_corrected_differential_phase(time, range) ;  
  
filtered_corrected_differential_phase:_FillValue = -9999. ;  
  
filtered_corrected_differential_phase:long_name = "Filtered Corrected Differential Phase" ;  
  
filtered_corrected_differential_phase:standard_name = "differential_phase_hv" ;
```

```
filtered_corrected_differential_phase:valid_max = 400. ;
filtered_corrected_differential_phase:valid_min = 0. ;
filtered_corrected_differential_phase:coordinates = "elevation azimuth range" ;

double corrected_specific_diff_phase(time, range) ;

corrected_specific_diff_phase:_FillValue = -9999. ;
corrected_specific_diff_phase:long_name = "Specific differential phase (KDP)" ;
corrected_specific_diff_phase:units = "degrees/km" ;
corrected_specific_diff_phase:standard_name = "specific_differential_phase_hv" ;
corrected_specific_diff_phase:coordinates = "elevation azimuth range" ;

double filtered_corrected_specific_diff_phase(time, range) ;

filtered_corrected_specific_diff_phase:_FillValue = -9999. ;

filtered_corrected_specific_diff_phase:long_name = "Filtered Corrected Specific differential
phase (KDP)" ;

filtered_corrected_specific_diff_phase:units = "degrees/km" ;
filtered_corrected_specific_diff_phase:standard_name = "specific_differential_phase_hv" ;
filtered_corrected_specific_diff_phase:coordinates = "elevation azimuth range" ;

double corrected_differential_reflectivity(time, range) ;

corrected_differential_reflectivity:_FillValue = 1.e+20 ;
corrected_differential_reflectivity:long_name = "Corrected differential reflectivity" ;
corrected_differential_reflectivity:units = "dB" ;
corrected_differential_reflectivity:standard_name = "corrected_log_differential_reflectivity_hv" ;
corrected_differential_reflectivity:coordinates = "elevation azimuth range" ;

double corrected_reflectivity(time, range) ;

corrected_reflectivity:_FillValue = 1.e+20 ;
corrected_reflectivity:long_name = "Corrected reflectivity" ;
```

```
corrected_reflectivity:units = "dBZ" ;
corrected_reflectivity:standard_name = "corrected_equivalent_reflectivity_factor" ;
corrected_reflectivity:coordinates = "elevation azimuth range" ;
float height_over_iso0(time, range) ;
height_over_iso0:long_name = "Height of radar beam over freezing level" ;
height_over_iso0:units = "m" ;
height_over_iso0:standard_name = "height" ;
double specific_attenuation(time, range) ;
specific_attenuation:_FillValue = 1.e+20 ;
specific_attenuation:long_name = "Specific attenuation" ;
specific_attenuation:units = "dB/km" ;
specific_attenuation:standard_name = "specific_attenuation" ;
specific_attenuation:valid_min = 0. ;
specific_attenuation:valid_max = 1. ;
specific_attenuation:coordinates = "elevation azimuth range" ;
double path_integrated_attenuation(time, range) ;
path_integrated_attenuation:_FillValue = 1.e+20 ;
path_integrated_attenuation:long_name = "Path Integrated Attenuation" ;
path_integrated_attenuation:units = "dB" ;
path_integrated_attenuation:coordinates = "elevation azimuth range" ;
double specific_differential_attenuation(time, range) ;
specific_differential_attenuation:_FillValue = 1.e+20 ;
specific_differential_attenuation:long_name = "Specific Differential Attenuation" ;
specific_differential_attenuation:units = "dB/km" ;
specific_differential_attenuation:coordinates = "elevation azimuth range" ;
```

```
double path_integrated_differential_attenuation(time, range) ;
path_integrated_differential_attenuation:_FillValue = 1.e+20 ;
path_integrated_differential_attenuation:long_name = "Path Integrated Differential Attenuation" ;
path_integrated_differential_attenuation:units = "dB" ;
path_integrated_differential_attenuation:coordinates = "elevation azimuth range" ;

double rain_rate_A(time, range) ;
rain_rate_A:_FillValue = 1.e+20 ;
rain_rate_A:long_name = "rainfall_rate" ;
rain_rate_A:units = "mm/hr" ;
rain_rate_A:standard_name = "rainfall_rate" ;
rain_rate_A:valid_min = 0. ;
rain_rate_A:valid_max = 400. ;
rain_rate_A:coordinates = "elevation azimuth range" ;
rain_rate_A:least_significant_digit = 1L ;

double specific_attenuation(time, range) ;
specific_attenuation:_FillValue = 1.e+20 ;
specific_attenuation:long_name = "Specific attenuation" ;
specific_attenuation:units = "dB/km" ;
specific_attenuation:standard_name = "specific_attenuation" ;
specific_attenuation:valid_min = 0. ;
specific_attenuation:valid_max = 1. ;
specific_attenuation:coordinates = "elevation azimuth range" ;

double path_integrated_attenuation(time, range) ;
path_integrated_attenuation:_FillValue = 1.e+20 ;
path_integrated_attenuation:long_name = "Path Integrated Attenuation" ;
```

```
path_integrated_attenuation:units = "dB" ;
path_integrated_attenuation:coordinates = "elevation azimuth range" ;
double specific_differential_attenuation(time, range) ;
specific_differential_attenuation:_FillValue = 1.e+20 ;
specific_differential_attenuation:long_name = "Specific Differential Attenuation" ;
specific_differential_attenuation:units = "dB/km" ;
specific_differential_attenuation:coordinates = "elevation azimuth range" ;
double path_integrated_differential_attenuation(time, range) ;
path_integrated_differential_attenuation:_FillValue = 1.e+20 ;
path_integrated_differential_attenuation:long_name = "Path Integrated Differential Attenuation" ;
path_integrated_differential_attenuation:units = "dB" ;
path_integrated_differential_attenuation:coordinates = "elevation azimuth range" ;
double rain_rate_A(time, range) ;
rain_rate_A:_FillValue = 1.e+20 ;
rain_rate_A:long_name = "rainfall_rate" ;
rain_rate_A:units = "mm/hr" ;
rain_rate_A:standard_name = "rainfall_rate" ;
rain_rate_A:valid_min = 0. ;
rain_rate_A:valid_max = 400. ;
rain_rate_A:coordinates = "elevation azimuth range" ;
rain_rate_A:least_significant_digit = 1L ;

double specific_attenuation(time, range) ;
specific_attenuation:_FillValue = 1.e+20 ;
specific_attenuation:long_name = "Specific attenuation" ;
```

```
specific_attenuation:units = "dB/km" ;
specific_attenuation:standard_name = "specific_attenuation" ;
specific_attenuation:valid_min = 0. ;
specific_attenuation:valid_max = 1. ;
specific_attenuation:coordinates = "elevation azimuth range" ;
double path_integrated_attenuation(time, range) ;
path_integrated_attenuation:_FillValue = 1.e+20 ;
path_integrated_attenuation:long_name = "Path Integrated Attenuation" ;
path_integrated_attenuation:units = "dB" ;
path_integrated_attenuation:coordinates = "elevation azimuth range" ;
double specific_differential_attenuation(time, range) ;
specific_differential_attenuation:_FillValue = 1.e+20 ;
specific_differential_attenuation:long_name = "Specific Differential Attenuation" ;
specific_differential_attenuation:units = "dB/km" ;
specific_differential_attenuation:coordinates = "elevation azimuth range" ;
double path_integrated_differential_attenuation(time, range) ;
path_integrated_differential_attenuation:_FillValue = 1.e+20 ;
path_integrated_differential_attenuation:long_name = "Path Integrated Differential Attenuation" ;
path_integrated_differential_attenuation:units = "dB" ;
path_integrated_differential_attenuation:coordinates = "elevation azimuth range" ;
double rain_rate_A(time, range) ;
rain_rate_A:_FillValue = 1.e+20 ;
rain_rate_A:long_name = "rainfall_rate" ;
rain_rate_A:units = "mm/hr" ;
rain_rate_A:standard_name = "rainfall_rate" ;
```

```
rain_rate_A:valid_min = 0. ;
rain_rate_A:valid_max = 400. ;
rain_rate_A:coordinates = "elevation azimuth range" ;
rain_rate_A:least_significant_digit = 1L ;
double specific_attenuation(time, range) ;
specific_attenuation:_FillValue = 1.e+20 ;
specific_attenuation:long_name = "Specific attenuation" ;
specific_attenuation:units = "dB/km" ;
specific_attenuation:standard_name = "specific_attenuation" ;
specific_attenuation:valid_min = 0. ;
specific_attenuation:valid_max = 1. ;
specific_attenuation:coordinates = "elevation azimuth range" ;
double path_integrated_attenuation(time, range) ;
path_integrated_attenuation:_FillValue = 1.e+20 ;
path_integrated_attenuation:long_name = "Path Integrated Attenuation" ;
path_integrated_attenuation:units = "dB" ;
path_integrated_attenuation:coordinates = "elevation azimuth range" ;
double specific_differential_attenuation(time, range) ;
specific_differential_attenuation:_FillValue = 1.e+20 ;
specific_differential_attenuation:long_name = "Specific Differential Attenuation" ;
specific_differential_attenuation:units = "dB/km" ;
specific_differential_attenuation:coordinates = "elevation azimuth range" ;
double path_integrated_differential_attenuation(time, range) ;
path_integrated_differential_attenuation:_FillValue = 1.e+20 ;
path_integrated_differential_attenuation:long_name = "Path Integrated Differential Attenuation" ;
```

```
path_integrated_differential_attenuation:units = "dB" ;
path_integrated_differential_attenuation:coordinates = "elevation azimuth range" ;
double rain_rate_A(time, range) ;
rain_rate_A:_FillValue = 1.e+20 ;
rain_rate_A:long_name = "rainfall_rate" ;
rain_rate_A:units = "mm/hr" ;
rain_rate_A:standard_name = "rainfall_rate" ;
rain_rate_A:valid_min = 0. ;
rain_rate_A:valid_max = 400. ;
rain_rate_A:coordinates = "elevation azimuth range" ;
rain_rate_A:least_significant_digit = 1L ;
double specific_attenuation(time, range) ;
specific_attenuation:_FillValue = 1.e+20 ;
specific_attenuation:long_name = "Specific attenuation" ;
specific_attenuation:units = "dB/km" ;
specific_attenuation:standard_name = "specific_attenuation" ;
specific_attenuation:valid_min = 0. ;
specific_attenuation:valid_max = 1. ;
specific_attenuation:coordinates = "elevation azimuth range" ;
double path_integrated_attenuation(time, range) ;
path_integrated_attenuation:_FillValue = 1.e+20 ;
path_integrated_attenuation:long_name = "Path Integrated Attenuation" ;
path_integrated_attenuation:units = "dB" ;
path_integrated_attenuation:coordinates = "elevation azimuth range" ;
double specific_differential_attenuation(time, range) ;
```

```
specific_differential_attenuation:_FillValue = 1.e+20 ;
specific_differential_attenuation:long_name = "Specific Differential Attenuation" ;
specific_differential_attenuation:units = "dB/km" ;
specific_differential_attenuation:coordinates = "elevation azimuth range" ;
double path_integrated_differential_attenuation(time, range) ;
path_integrated_differential_attenuation:_FillValue = 1.e+20 ;
path_integrated_differential_attenuation:long_name = "Path Integrated Differential Attenuation" ;
path_integrated_differential_attenuation:units = "dB" ;
path_integrated_differential_attenuation:coordinates = "elevation azimuth range" ;
double rain_rate_A(time, range) ;
rain_rate_A:_FillValue = 1.e+20 ;
rain_rate_A:long_name = "rainfall_rate" ;
rain_rate_A:units = "mm/hr" ;
rain_rate_A:standard_name = "rainfall_rate" ;
rain_rate_A:valid_min = 0. ;
rain_rate_A:valid_max = 400. ;
rain_rate_A:coordinates = "elevation azimuth range" ;
rain_rate_A:least_significant_digit = 1L ;

:history = "created by rjackson on or-condo-c226.ornl.gov at 2021-12-13T13:15:24.662992 using
Py-ART" ;

:_NCProperties = "version=2,netcdf=4.6.2,hdf5=1.10.4" ;

:comment = "This is highly experimental and initial data. There are many known and unknown
issues. Please do not use before contacting the Translator responsible scollis@anl.gov." ;

:attributions = "This data is collected by the ARM Climate Research facility. Radar system is
operated by the radar engineering team radar@arm.gov and the data is processed by the precipitation
radar products team. LP code courtesy of Scott Giangrande BNL." ;

:Conventions = "CF/Radial 1.4" ;

:doi = "10.5439/1573362" ;
```

## 8.0 References

- Al-Sakka, H, AA Boumahmoud, B Fradon, SJ Frasier, and P Tabary. 2013. “A New Fuzzy Logic Hydrometeor Classification Scheme Applied to the French X-, C-, and S-Band Polarimetric Radars.” *Journal of Applied Meteorology and Climatology* 52(10): 2328–2344, <https://doi.org/10.1175/JAMC-D-12-0236.1>
- Bringi, VN, GJ Huang, V Chandrasekar, and E Gorgucci. 2002. “A Methodology for Estimating the Parameters of a Gamma Raindrop Size Distribution Model from Polarimetric Radar Data: Application to a Squall-Line Event from the TRMM/Brazil Campaign.” *Journal of Atmospheric and Oceanic Technology* 19(5): 633–645, [https://doi.org/10.1175/1520-0426\(2002\)019<0633:AMFETP>2.0.CO;2](https://doi.org/10.1175/1520-0426(2002)019<0633:AMFETP>2.0.CO;2)
- Dolan, B, and SA Rutledge. 2009. “A Theory-Based Hydrometeor Identification Algorithm for X-Band Polarimetric Radars.” *Journal of Atmospheric and Oceanic Technology* 26(10): 2071–2088, <https://doi.org/10.1175/2009JTECHA1208.1>
- Du, P, WA Kibbe, and SM Lin. 2006. “Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching.” *Bioinformatics* 22(17): 2059–2065, <https://doi.org/10.1093/bioinformatics/btl355>
- Gaustad, K, T Shippert, B Ermold, S Beus, J Daily, A Borsholm, and K. Fox. 2014. “A scientific data processing framework for time series netcdf data.” *Environmental Modelling & Software*, 60: 241–249, <https://doi.org/10.1016/j.envsoft.2014.06.005>
- Giangrande, SE, and AV Ryzhkov. 2008. “Estimation of rainfall based on the results of polarimetric echo classification.” *Journal of Applied Meteorology and Climatology* 47(9): 2445–2462, <https://doi.org/10.1175/2008JAMC1753.1>
- Giangrande, SE, R McGraw, and L Lei. 2013. “An Application of Linear Programming to Polarimetric Radar Differential Phase Processing.” *Journal of Atmospheric and Oceanic Technology* 30(8): 1716–1729, <https://doi.org/10.1175/JTECH-D-12-00147.1>
- Gourley, JJ, P Tabary, and J Parent du Chatelet. 2007. “A Fuzzy Logic Algorithm for the Separation of Precipitating from Nonprecipitating Echoes Using Polarimetric Radar Observations.” *Journal of Atmospheric and Oceanic Technology* 24(8): 1439–1451, <https://doi.org/10.1175/JTECH2035.1>
- Gu, JY, A Ryzhkov, P Zhang, P Neilley, M Knight, B Wolf, and DI Lee. 2011. “Polarimetric Attenuation Correction in Heavy Rain at C Band.” *Journal of Applied Meteorology and Climatology* 50(1): 39–58, <https://doi.org/10.1175/2010JAMC2258.1>
- Heistermann, M, S Collis, MJ Dixon, S Giangrande, JJ Helmus, B Kelley, J Koistinen, DB Michelson, M Peura, T Pfaff and DB Wolff. 2014. “The Emergence of Open-Source Software for the Weather Radar Community.” *Bulletin of the American Meteorological Society* 96(1): 117–128, <https://doi.org/10.1175/BAMS-D-13-00240.1>

Helbush, RE. 1968. “Linear programming applied to operational decision making in weather risk situations.” *Monthly Weather Review* 96(12): 876–882, [https://doi.org/10.1175/1520-0493\(1968\)096<0876:LPATOD>2.0.CO;2](https://doi.org/10.1175/1520-0493(1968)096<0876:LPATOD>2.0.CO;2)

Helmus, JJ, and SM Collis. 2016. “The Python ARM Radar Toolkit (Py-ART), a library for working with weather radar data in the Python programming language.” *Journal of Open Research Software* 4(1): e25, <http://doi.org/10.5334/jors.119>

James, CN, and RA Houze. 2001. “A Real-Time Four-Dimensional Doppler Dealiasing Scheme.” *Journal of Atmospheric and Oceanic Technology* 18(10):1674–1683, [https://doi.org/10.1175/1520-0426\(2001\)018<1674:ARTFDD>2.0.CO;2](https://doi.org/10.1175/1520-0426(2001)018<1674:ARTFDD>2.0.CO;2)

Jones, E, T Oliphant, and P Peterson. 2001. SciPy: Opensource scientific tools for Python. <http://www.scipy.org/>

Kollias, P, L Jo, P Borque, A Tatarevic, K Lamer, N Bharadwaj, K widener, K Johnson, and EE Clothiaux. 2013. “Scanning ARM Cloud Radars. Part II: Data Quality Control and Processing.” *Journal of Atmospheric and Oceanic Technology*, 31(3): 583–598, <https://doi.org/10.1175/JTECH-D-13-00045.1>

Mather, JH, and JW Voyles. 2012. “The Arm Climate Research Facility: A Review of Structure and Capabilities.” *Bulletin of the American Meteorological Society* 94 (3): 377–392, <https://doi.org/10.1175/BAMS-D-11-00218.1>

Varble, A, S Nesbitt, P Salio, E Avila, P Borque, P DeMott, G McFarquhar, S van den Heever, E Zipser, D Gochis, R Houze, M Jensen, P Kollias, S Kreidenweis, R Leung, K Rasmussen, D Romps, and C Williams. 2019. Cloud, Aerosol, and Complex Terrain Interactions (CACTI) Field Campaign Report. U.S. Department of Energy. DOE/SC-ARM-19-028, <https://doi.org/10.2172/1574024>

Wen, G, A Protat, PT May, X Wang, and W Moran. 2015. “A Cluster-Based Method for Hydrometeor Classification Using Polarimetric Variables. Part I: Interpretation and Analysis.” *Journal of Atmospheric and Oceanic Technology* 32(7): 1320–1340, <https://doi.org/10.1175/JTECH-D-13-00178.1>

Wikipedia. 2016: “Directional statistics.” <https://en.wikipedia.org/w/index.php?title=Directionalstatistics&oldid=705952853>



[www.arm.gov](http://www.arm.gov)

U.S. DEPARTMENT OF  
**ENERGY**

---

Office of Science