U.S. DEPARTMENT OF ENERGY | Office of Science

DOE/SC-ARM-TR 313

Colorado State University (CSU) X-Band Precipitation Radar Plan Position Indicator Data Processed with Corrected Moments in Antenna Coordinates (CMAC) Value-Added Product Report

JR O'Brien
RC Jackson
SM Collis
BA Raut
D Feldman
M Grover
ZS Sherman
A Theisen
M Tuftedal

December 2024

ARM

Colorado State University (CSU) X-Band Precipitation Radar Plan Position Indicator Data Processed with Corrected Moments in Antenna Coordinates (CMAC) Value-Added Product Report

JR O'Brien, Argonne National Laboratory (ANL)
Principal Investigator

M Grover, ANL
RC Jackson, ANL
ZS Sherman, ANL
SM Collis, ANL
A Theisen, ANL
BA Rout, Northwestern University
M Tuftedal, ANL
D Feldman, Lawrence Berkeley National Laboratory
Co-Investigators

December 2024

How to cite this document:

O'Brien, JR, M Grover, RC Jackson, ZS Sherman, SM Collis, A Theisen, BA Raut, M Tuftedal, and D Feldman. 2024. Colorado State University (CSU) X-Band Precipitation Radar Plan Position Indicator Data Processed with Corrected Moments in Antenna Coordinates (CMAC) Value-Added Product Report. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-TR-313.

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

Executive Summary

In 2010 the U.S. Department of Energy (DOE) Atmospheric Radiation Measurement (ARM) user facility procured 3- and 5-cm wavelength radars for documenting the macrophysical, microphysical, and dynamical structure of precipitating systems. To maximize the scientific impact, ARM 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.

We have now used the lessons learned from the processing of the 3- and 5-cm wavelength radars obtained by ARM to help process X-band radar data from the Surface Atmospheric Integrated Field Laboratory (SAIL) field campaign. 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 specially thank Scott Giangrande for contributing the Linear Programming code and Kai Muhlbauer's effort in accelerating the method to the point of usability. We also thank all the radar mentors and 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
CSU	Colorado State University
DOE	U.S. Department of Energy
ID	identification
KAZR	Ka-Band ARM Zenith Radar
LDQUANTS	Laser Disdrometer Quantities Value-Added Product
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
PPI	plan position indicator
Py-ART	Python-ARM Radar Toolkit
RSL	Radar Software Library
SAIL	Surface Atmosphere Integrated Field Laboratory
SQI	signal quality index
XSAPR	X-Band Scanning ARM Precipitation Radar

Contents

Figures

Tables

1.0 Introduction

The Atmospheric Radiation Measurement user facility (Mather and Voyles 2012) has a long history of sensing clouds in the column using the millimeter 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 original focus of CMAC was to process data from the ARM X-Band and C-Band Scanning Precipitation Radars (X/CSAPRs). However, the focus has now shifted to incorporate additional radars at these wavelengths (such as the Colorado State University [CSU] X-band Precipitation Radar deployed for the Surface Atmosphere Integrated Field Laboratory [SAIL]), with additional data quality processing specific to the ARM scientific needs for a given deployment . Due to the agility and lower cost of the X-band and C-band scanning radars, the program 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^{-1} when operating in a baseline mode (such as during the Mid-Latitude Convective Continental Clouds Experiment (MC3E; Jensen et al. 2015) and the Cloud, Aerosol, and Complex Terrain Interactions [CACTI] campaign). 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 ρ_{HV} and normalized coherent power (NCP, also referred to as signal quality index or SQI). These are defined as:

$$\rho HV(0) = \frac{|S_{vv}S_{hh}^{*}|}{\sqrt{<|S_{hh}|^2><|S_{vv}|^2>}} \quad (1)$$

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

Where the S terms are elements of the scattering matrix, P_{coh} is the coherent part of the Doppler spectrum, and P_{DC} 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, the Doppler velocity can be determined and 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 that 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."

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 insufficient to constrain the problem of gate-ID, especially the identification of multiple trips. We can, however, generate several pre-ID retrievals and inputs to constrain the problem, 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
- Integrate and apply to reflectivity
- Calculate rain rate for liquid precipitation using specific attenuation
- Calculate snowfall rate from reflectivity.

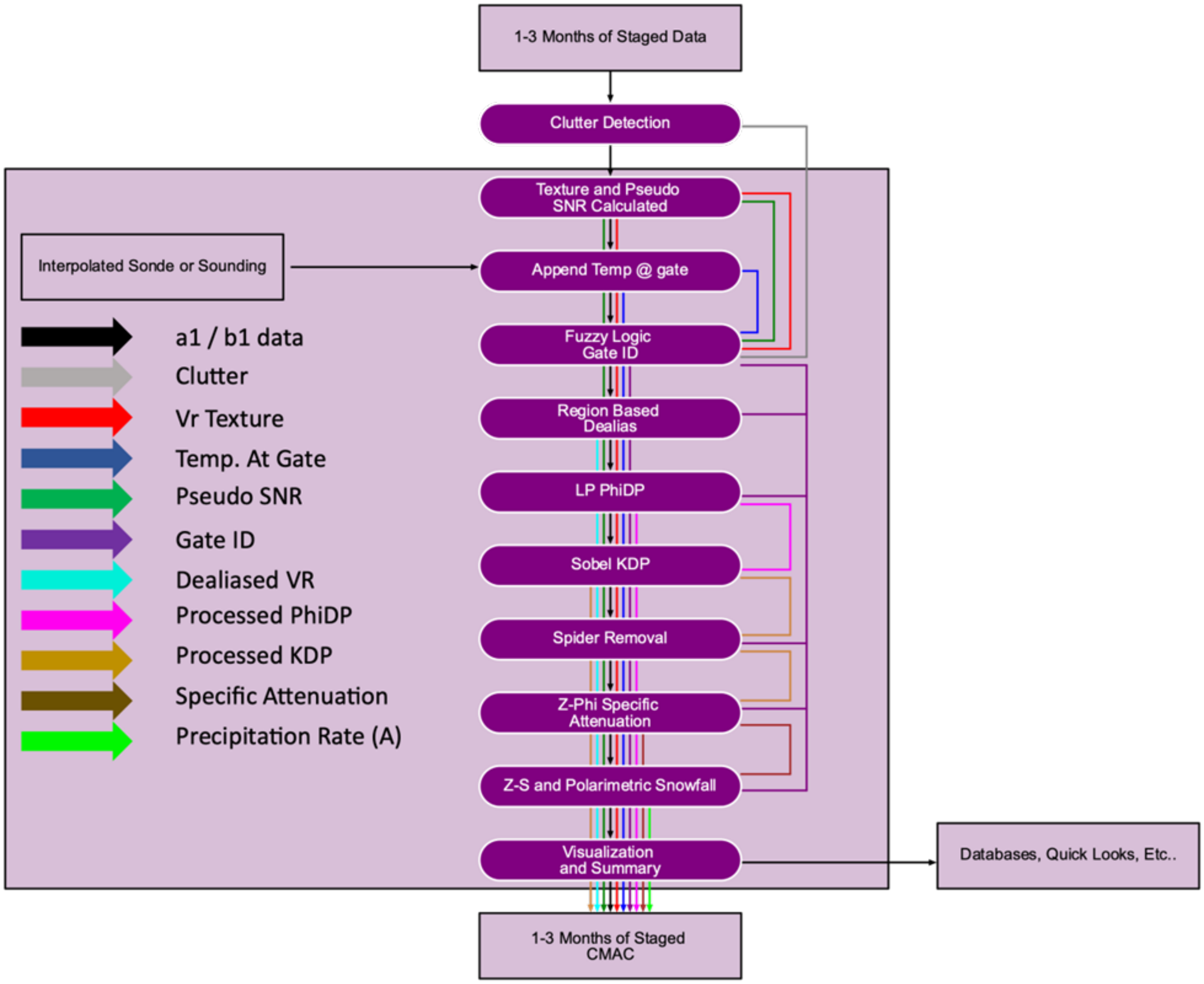

Figure 1. The application chain for Corrected Moments in Antenna Coordinates for the ARM SAIL experiment.

2.1 Calculations Performed to Aid Identification of Scatterers at Gate

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 product http://dx.doi.org/10.5439/1095316). 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 identify 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 a good discriminator of significant returns. The abstract concept is that a central pixel *(i, j)*, the points surrounding the pixel in an $n \times m$ kernel, are collected as shown in Figure 2. Then the variance is calculated on the set of points and is returned as the $(i, j)^{th}$ value in the resultant 2D (range, time/azimuth) array.

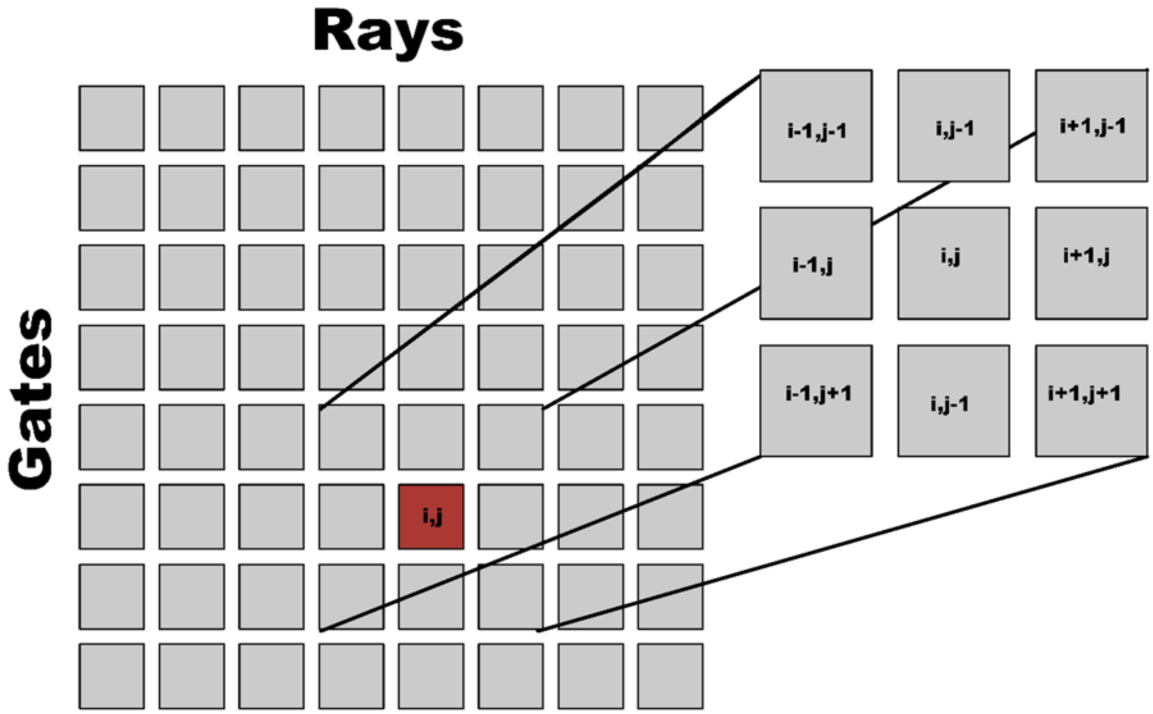

Figure 2. Illustration of the concept of a moving filter overrange gates of adjacent rays. The center element, *(i, j),* is calculated by passing surrounding elements. The footprint of the surrounding elements is determined by the kernel. In many cases we use a 3x3 kernel.

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 θ from 0 to π and the standard deviation is given by:

$$x = \cos\theta \tag{3}$$

$$y = \sin\theta \tag{4}$$

$$R = \sqrt{x_2 + y_2} \tag{5}$$

$$S = \sqrt{-2 * log\, log\, R} \tag{6}$$

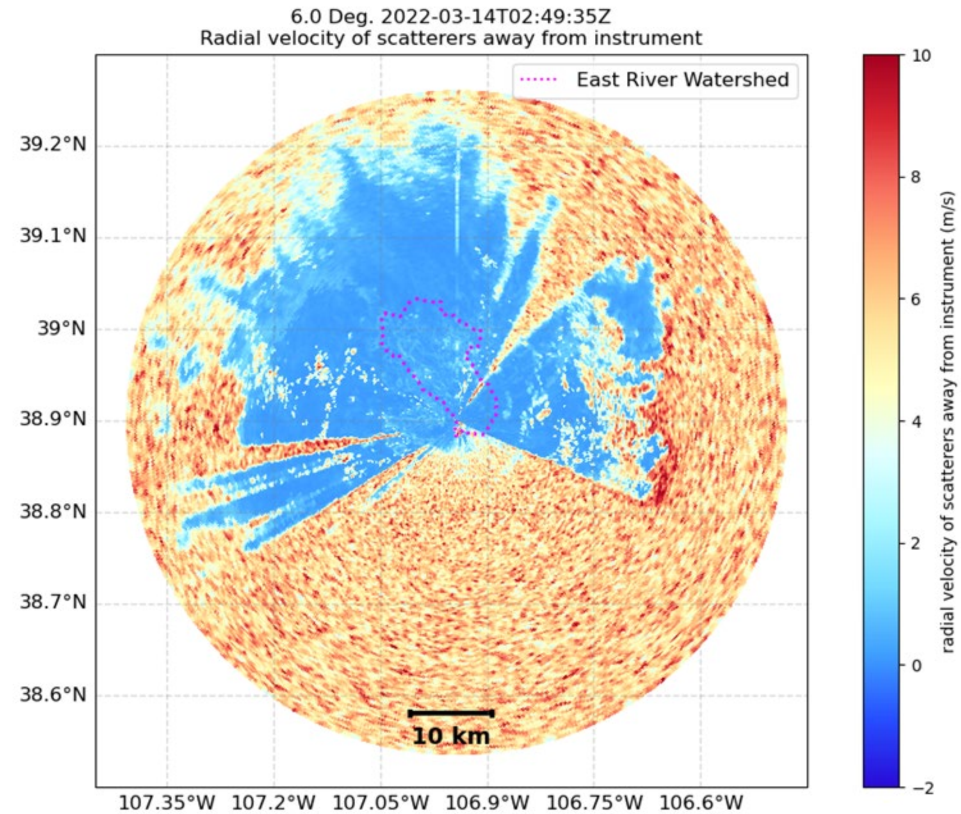

Figure 3. Calculations of texture of radial velocity from the CSU X-Band Precipitation Radar (XPRECIPRADAR) using circular statistics to avoid false texture on folds.

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.

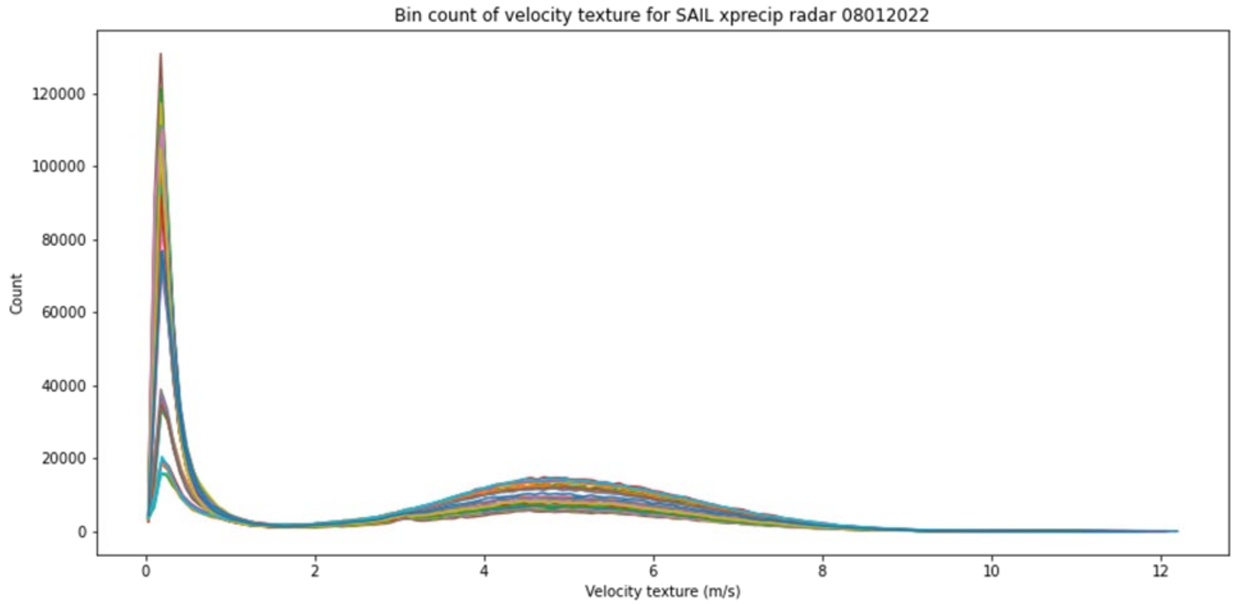

Figure 4. Histogram of texture values for individual radar volume scans from 1 August 2022 during SAIL. The left peak corresponds to significant returns, the right to noise.

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 extensively used for particle identification, few investigators have done this as a first step (pre K_{dp}, 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.

Figure 5 shows an example of scatterers at gate identification from a plan position indicator (PPI) tilt from the XPRECIPRADAR during SAIL. Regions of snow scatterers are shown in cyan, rain in green, 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 subsequent processing algorithms. For example, Linear Programming (see Section 2.2.2) filtering of φ_{DP} would be performed on gates identified as rain and attenuation correction (offset) on the union of rain, melting layer, and snow.

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

Class	Texture (m/s)	ρ_{HV}	NCP	Temperature (C)	Height (km)	SNR (dB)
Melting	[0, 0, 2.4, 2.5], 0	[0.6, 0.65, 0.9, 0.96], 3	[0.4, 0.5, 1, 1], 0	[0, 0.1, 2, 4], 4	[0, 0, 25, 25], 0	[20, 22, 1000, 1000], 0
Multi-trip	[7.7, 10, 130, 130], 4	[0.7, 0.8, 1, 1], 0	[0, 0, 0.3, 0.35], 0	[-100, -100, 100,100], 0	[0, 0, 5, 8], 0	[20, 22, 1000, 1000], 1
Rain	[0, 0, 2.4, 2.5], 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	[20, 22, 1000, 1000], 1
Snow	[0, 0, 2.4, 2.5], 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	[20, 22, 1000, 1000], 1
No Scatter	[0, 0, 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, 20, 22], 4

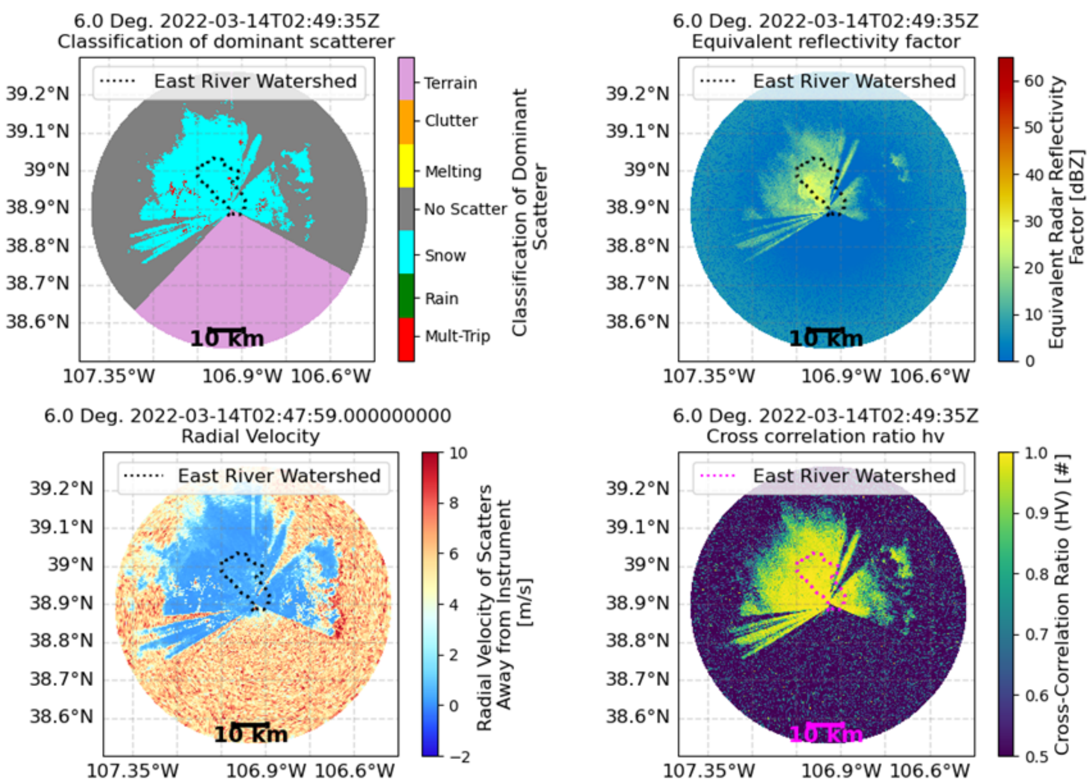

Figure 5. Highest-score-determined categories with hard constraints for the dominant scattering process for each gate from the XPRECIPRADAR alongside (clockwise) reflectivity factor, cross-correlation ratio, and velocity texture. These values will be used to determine what post-processing will be applied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

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

- Dealiased doppler velocities
- φ_{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.

2.2.1 Dealising

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). Problems with the implementation of the paper into code led to a long discussion on issues in dealiasing (see 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 CSU XPRECIPRADAR collected during SAIL. 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 $V_{sounding}$.

$$J = \sum_{i \in regions} n_i V_{nyq} V_i - V_{sounding} \tag{7}$$

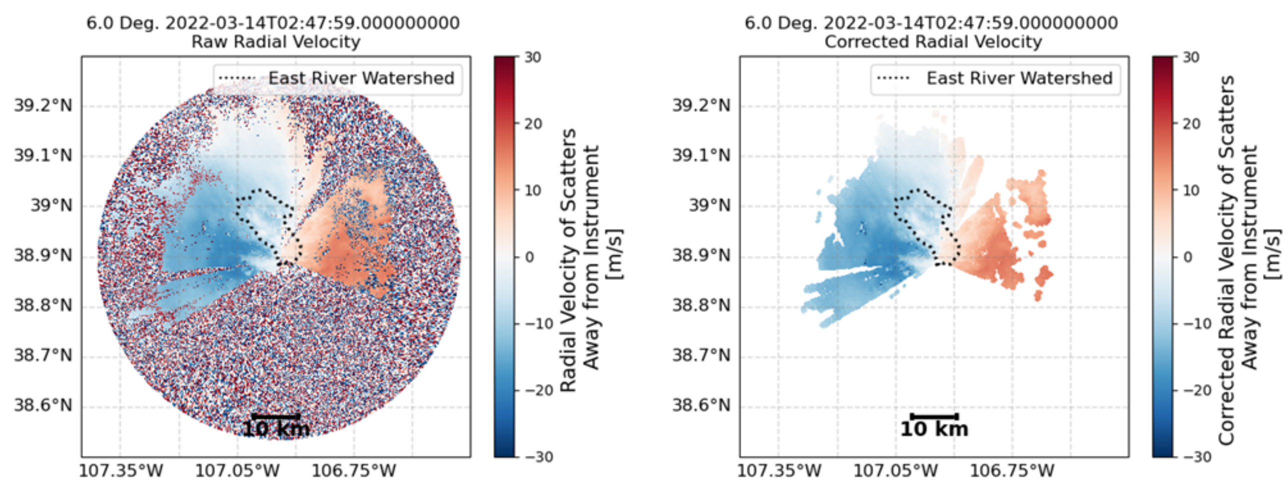

Figure 6. Raw (left) and dealiased (right) radial velocities from the CSU XPRECIPRADAR collected during SAIL. Unfolding was performed using the region-based approach.

2.2.2 Filtering of Measured Phase Shift between Vertical and Horizontal Polarization

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

$$\Psi_{DP} = \Phi_{DP} + \delta + NBF \tag{8}$$

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 Φ_{DP} from the measured signal. Taking advantage of the fact that liquid water content cannot be negative and therefore we expect Φ_{DP} to strictly increase, we can construct a filter to extract Φ_{DP} from Ψ_{DP}. Giangrande et al. (2013) outlines an objective technique that uses Linear Programming (LP; i.e., Helbush 1968) to create a Φ_{DP} that is piecewise increasing and (importantly) is non-biased. That is, given a Ψ_{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. The strength of the fit is influenced by the local reflectivity as a weak constraint (see Giangrande and Ryzhkov [2008] for details). Once Φ_{DP} is retrieved, the specific differential phase K_{DP} is retrieved by convoluting Φ_{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 shows a single radial of data from the CSAPR highlighting the LP technique. Raw Ψ_{DP} is shown in green, retrieved Φ_{DP} in black, K_{DP} in red, and reflectivity (divided by 10) in blue. The retrieved Φ_{DP} is monotonically increasing resulting in a strictly positive K_{DP}. 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 eight minutes to under a minute.

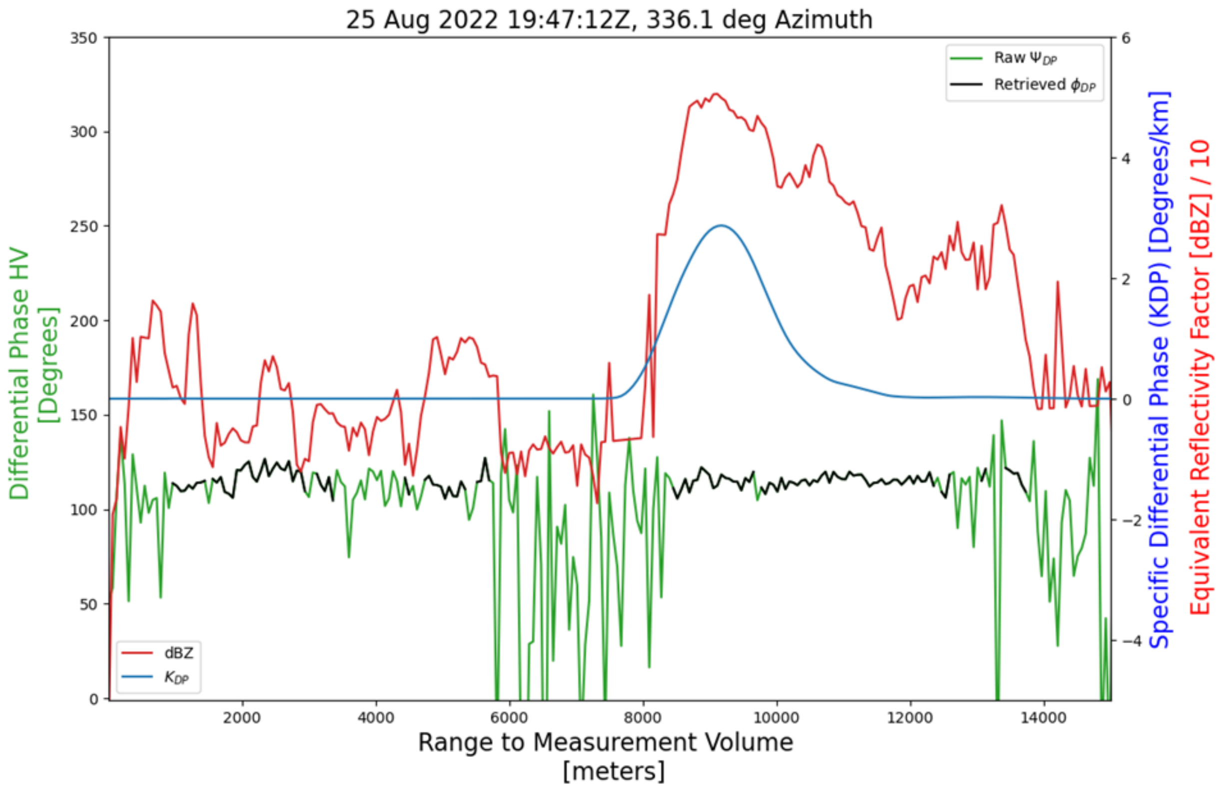

Figure 7. A single radial of data from CSU XPRECIPRADAR highlighting the LP technique. Raw *ΦDP* is shown in green, unfolded *ΦDP* in black, K_{DP} in blue, and reflectivity (divided by 10) in red.

2.2.3 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 φ_{dp} calculation, which becomes apparent in the K_{dp} and A fields. To mitigate this, we can filter out all K_{dp} greater than 15° km^{-1}.

Before application of the attenuation correction, we apply a reflectivity offset. For the data from SAIL, we scale by comparing to a disdrometer measurement; in the future, data will be provided in calibrated form using end-to-end means. Figure 8 shows the original reflectivity as produced by the radar on the left and the scaled and then corrected reflectivity on the right.

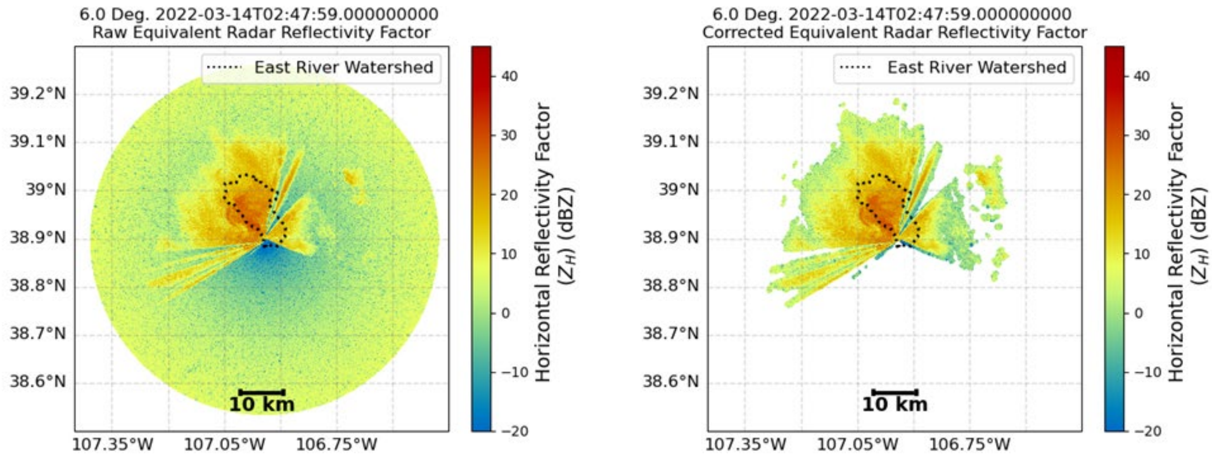

Figure 8. Reflectivity as measured by the radar (left) and disdrometer offset adjusted attenuation corrected reflectivity with the significant feature detection mask applied (right).

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.

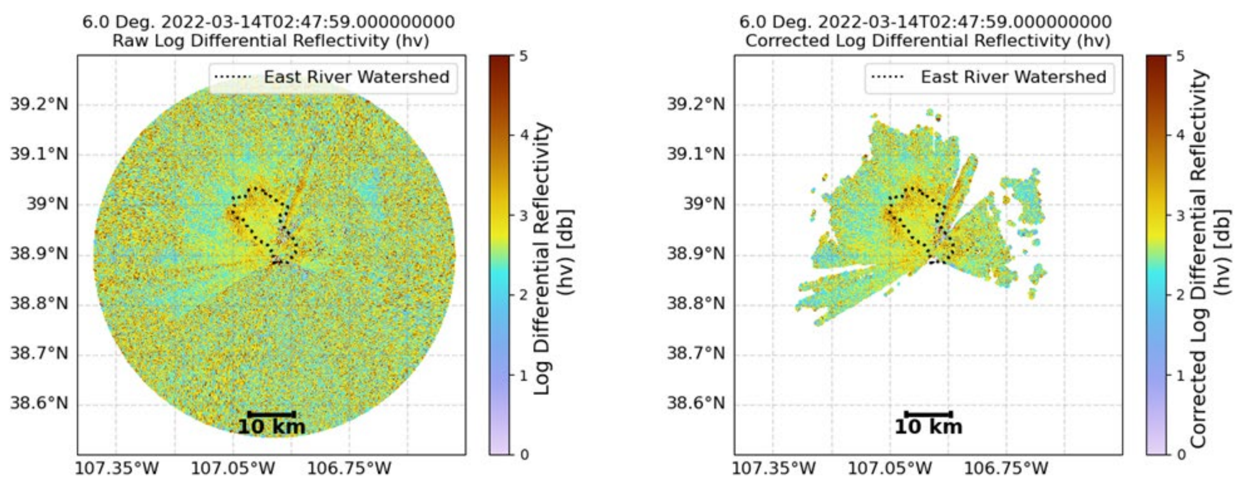

Figure 9. An example of (left) uncorrected Z_{dr} and (right) Z_{dr} corrected for bias and differential attenuation from a PPI scan taken from the XPRECIPRADAR during SAIL.

2.3 Beam Blockage

Radars in complex terrain often suffer from beam blockage issues. When the path of the radar beam encounters an obstacle such as mountains, a fraction of, or sometimes the full transmitting power of the radar, is inhibited from measuring the atmosphere. This can cause a reduction in the power return received by the radar, which can cause underestimation, or worse, blind spots.

Due to the mountain range located south of the radar during SAIL, the radar has limited view of the southern coverage at lower elevation angles. Beam-blockage maps were generated for the different elevation angles of the radar using *wradlib*. The cumulative beam-blockage fraction for the elevation angles affected by the terrain are shown in Figure 10, where 0.0 (white) means the radar has a clear view and 1.0 (red) means there is total beam blockage.

The beam blockage map is also used in determining the dominant scattering process for the gates, where gates that show more than 30% beam blockage are flagged as "terrain blockage" (refer to the classification in the bottom row of Figure 5).

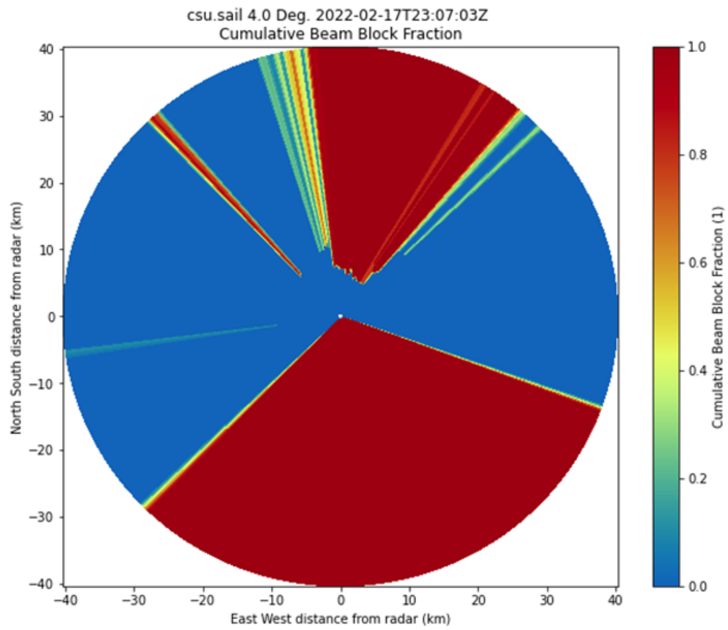

Figure 10. Cumulative beam blockage for the CSU XPRECIPRADAR.

2.4 Snowfall Retrievals

To provide accurate precipitation estimates for the Upper Colorado River Basin, estimated snowfall rates are desired by the SAIL community. However, accurate measurements of snowfall within complex terrain from radar are difficult to achieve due to the diversity of hydrometeor characteristics such as crystal habit and distribution of hydrometeor sizes. To estimate snowfall from radar, empirical relationships of the equivalent radar reflectivity factor (Z_e) to liquid-equivalent snowfall rates ($Z_e = aS^b$) are typically applied. The coefficients *a* and *b* are carefully chosen for the environmental conditions of the observations. For SAIL, instead of determining one relationship to relate to each event, an ensemble approach with multiple *a* and *b* coefficients is used. This approach is designed to accurately describe the uncertainty within the precipitation estimates of the region. Taken from Bukovčić et al. (2018), and shown in Table 2, four initial empirical relationships have been chosen to represent the spread within snowfall estimates for the region. Additional relationships are expected to be eventually included upon collaboration with the SAIL community and analysis into more cases throughout the duration of the field experiment. Figure 11 contains the estimated snowfall rates calculated by applying the empirical relationships to the CMAC-corrected observations for a snowfall event in March 2022, highlighting the spread within the four relationships.

Table 2. Empirical relationships used to calculate estimated snowfall rates from radar.

Source	Z(S)	A Coefficient	B Coefficient	Radar Band
Wolfe and Snider (2012)	$Z = 110S^2$	110	2	S
WSR-88D High Plains	$Z = 130S^2$	130	2	S
Braham (1990) 1	$Z = 67S^{1.28}$	67	1.28	X
Braham (1990) 2	$Z = 114S^{1.39}$	114	1.39	X

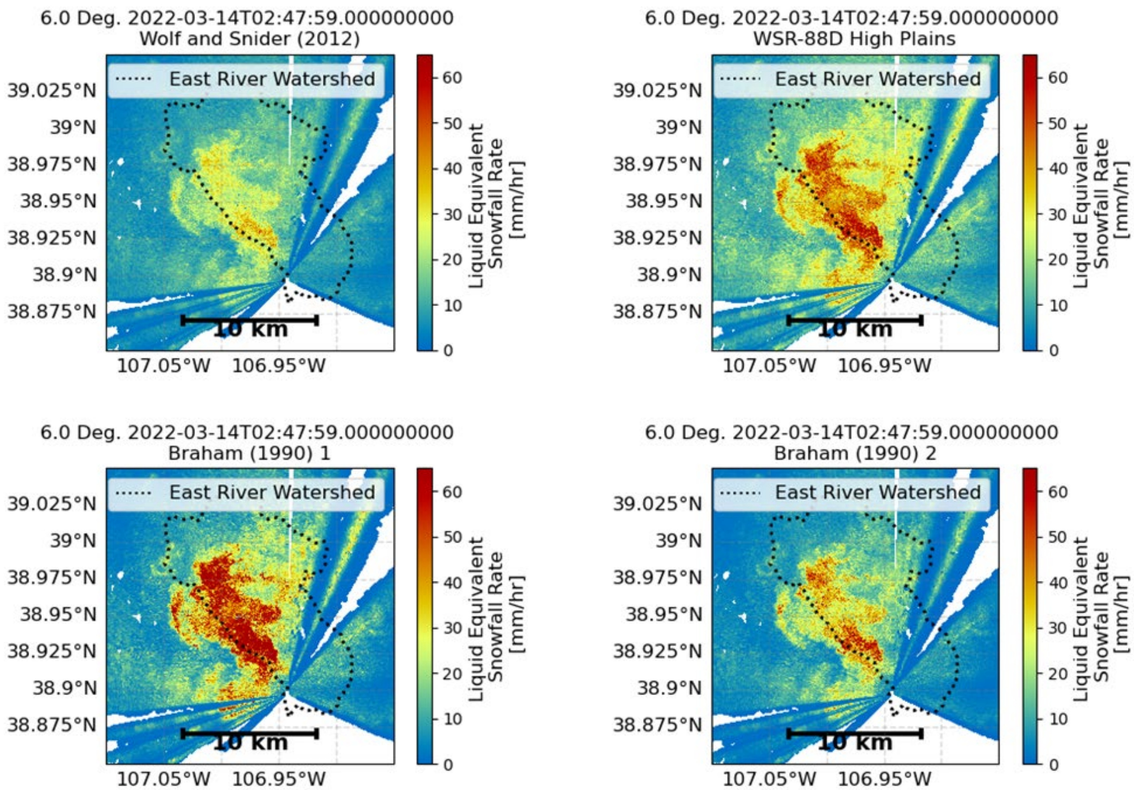

Figure 11. Estimated snowfall rates calculated from CSU XPRECIPRADAR CMAC-corrected observations for 14 March 2022.

2.5 Rainfall Retrievals

To validate the relationships used within CMAC for reflectivity R(Z)- and attenuation R(A)-based rainfall estimates, comparison of the CMAC rainfall estimates to the Pluvio2 weighing bucket rain gauge and ARM's Laser Disdrometer Quantities Value-Added Product (LDQUANTS) at the ARM SAIL M1 site was conducted for August 2022 (Figures 12 and 13). As shown in Figure 13, attenuation-based rainfall estimates were found to routinely underestimate rainfall compared to both in situ observations. Through investigation of the laser disdrometer observations, we determined that the vast majority of warm-phase precipitation at the M1 site was drizzle, where attenuation-based rainfall calculations are known to perform poorly. The CMAC reflectivity-based rainfall estimates are recommended for use for warm-phase precipitation during SAIL.

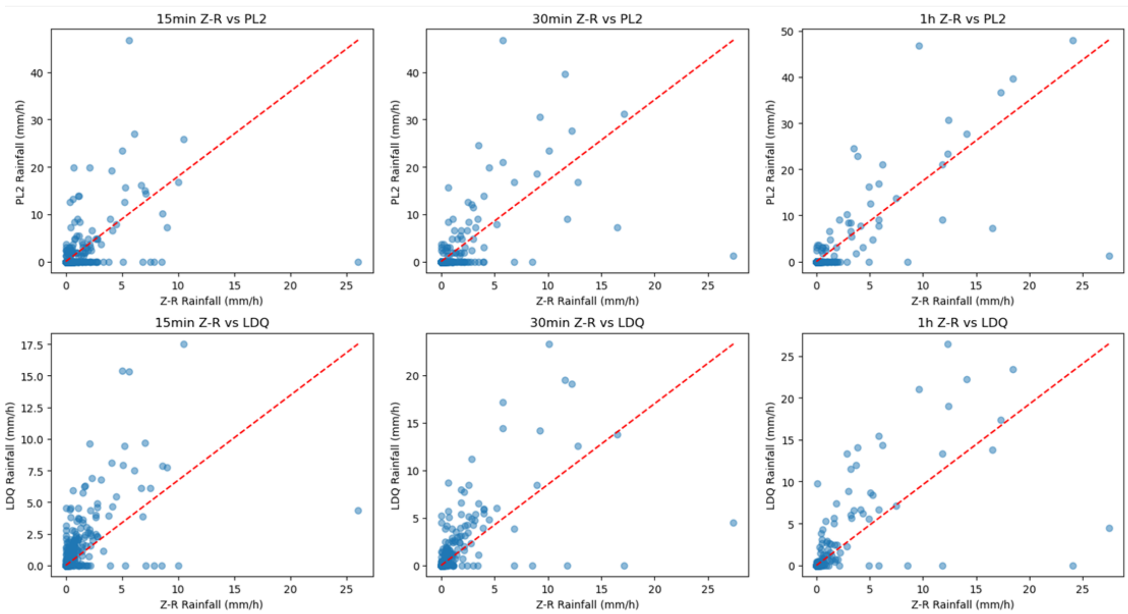

Figure 12. Comparison of CMAC reflectivity-based rainfall estimates to the Pluvio2 weighing bucket rain gauge and LDQUANTS datastreams for 15-minute, 30-minute, and 1-hour accumulations for August 2022.

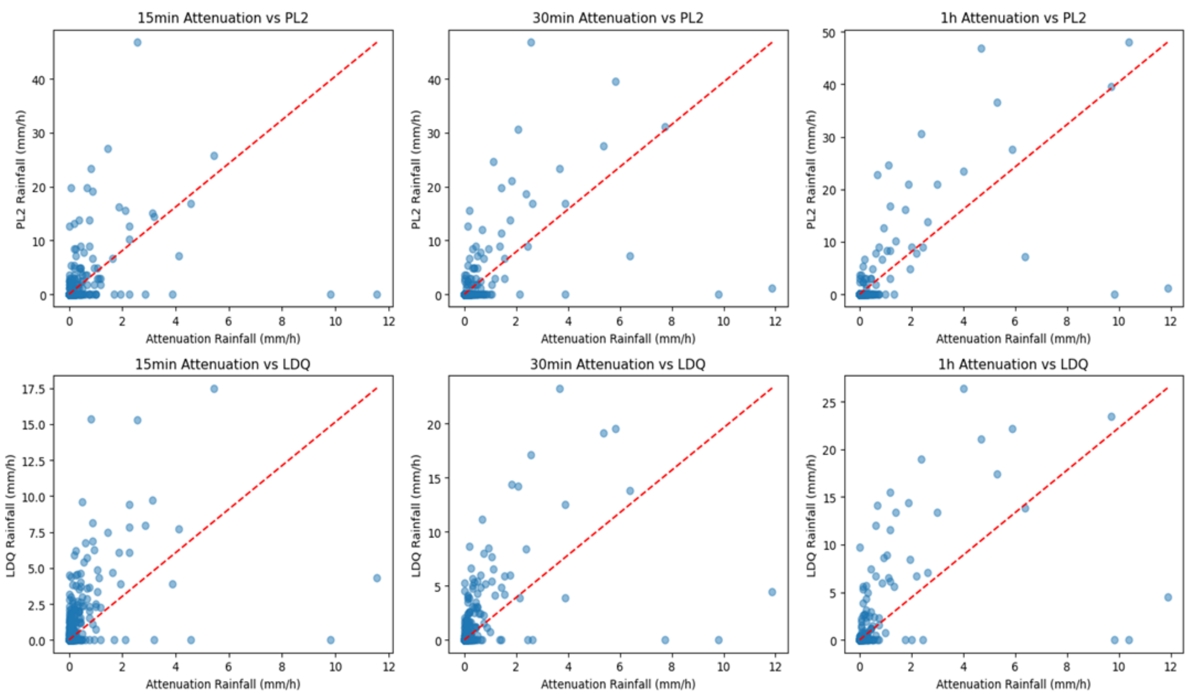

Figure 13. Comparison of CMAC attenuation-based rainfall estimates to the Pluvio2 weighing bucket rain gauge and LDQUANTS datastreams for 15-minute, 30-minute, and 1-hour accumulations for August 2022.

3.0 Open Science Documentation

To encourage the SAIL and atmospheric science community to collaborate with this product, a repository was created to hold workflow examples. Examples highlighting products derived from the XPRECIPRADAR CMAC-corrected observations are also included, as well as highlights of unique events from the SAIL field experiment. Users are encouraged to review this repository if they are interested in reproducing the outlined methodology or interested in viewing the figures created within this document. Users are also encouraged to submit their own examples of unique SAIL events that may be of interest. The SAIL open science documentation is available at https://arm-development.github.io/sail-xprecip-radar.

4.0 Challenges

Initial robustness tests show we have a lot of work to do in the detection and tagging of clutter returns. We are currently working on techniques that look at the mean and variance of reflectivity in non-precipitating regions to diagnose clutter. However, this is challenging because anomalous propagation exacerbates the clutter issue and is often present as a convective system cools and moistens the boundary layer. The other challenge is in the software engineering of the LP method. It has been discovered that in regions of extended δ_{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 actively working on this issue.

5.0 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

Bukovčić, P, A Ryzhkov, D Zrnić, and G Zhang. 2018. "Polarimetric Radar Relations for Quantification of Snow Based on Disdrometer Data." *Journal of Applied Meteorology and Climatology* 57(1): 103–120, https://doi.org/10.1175/JAMC-D-17-0090.1

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

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: 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

Jones, E, T Oliphant, and P Peterson. 2001. "SciPy: Opensource scientific tools for Python." http://www.scipy.org/, [Online; accessed 2016-03-02].

Kollias, P, I 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 — wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Directionalstatistics&oldid=705952853, [Online; accessed 1-March-2016]

Appendix A

Output Data

```
netcdf xprecipradarppicmac2.c1{
dimensions:
	time = 1 ;
	range = 668 ;
	sweep = 8 ;
	string_length = 192 ;
variables:
	int64 time(time) ;
		time:long_name = "Time in Seconds from Volume Start" ;
		time:units = "seconds since 1970-01-01T00:00:00Z" ;
		time:standard_name = "time" ;
		time:calendar = "standard" ;
	int64 range(range) ;
		range:long_name = "Range to measurement volume" ;
		range:units = "meter" ;
		range:standard_name = "projection_range_coordinate" ;
		range:spacing_is_constant = "true" ;
		range:meters_to_center_of_first_gate = "-112.6891" ;
		range:meters_between_gates = "59.94095" ;
		range:axis = "radial_range_coordinate" ;
	double azimuth(time) ;
		azimuth:_FillValue = -9999. ;
		azimuth:long_name = "Azimuth Angle from True North" ;
		azimuth:units = "degree" ;
		azimuth:axis = "radial_azimuth_coordinate" ;
		azimuth:standard_name = "sensor_to_target_azimuth_angle" ;
	double elevation(time) ;
		elevation:_FillValue = -9999. ;
		elevation:long_name = "Elevation angle from horizontal plane" ;
		elevation:units = "degree" ;
		elevation:standard_name = "sensor_to_target_elevation_angle" ;
		elevation:axis = "radial_elevation_coordinate" ;
	double DBZ(time, range) ;
		DBZ:_FillValue = -32768. ;
		DBZ:long_name = "Equaivalent_radar_reflectiivity_factor" ;
		DBZ:units = "dBZ" ;
		DBZ:standard_name = "equivalent_reflectivity_factor" ;
		DBZ:coordinates = "elevation azimuth range" ;
	double VEL(time, range) ;
```

```
        VEL:_FillValue = -32768. ;
        VEL:long_name = "Radial Doppler Velocity, Positive for Motion Away from Instrument" ;
        VEL:units = "m/s" ;
        VEL:standard_name = "radial_velocity_of_scatterers_away_from_instruments" ;
        VEL:coordinates = "elevation azimuth range" ;
double WIDTH(time, range) ;
        WIDTH:_FillValue = -32768. ;
        WIDTH:long_name = "Spectral Width" ;
        WIDTH:units = "m/s" ;
        WIDTH:standard_name = "doppler_spectrum_width" ;
        WIDTH:coordinates = "elevation azimuth range" ;
double ZDR(time, range) ;
        ZDR:_FillValue = -32768. ;
        ZDR:long_name = "Differential Reflectivity" ;
        ZDR:units = "dB" ;
        ZDR:standard_name = "log_differential_reflectivity_hv" ;
        ZDR:coordinates = "elevation azimuth range" ;
double PHIDP(time, range) ;
        PHIDP:_FillValue = -32768. ;
        PHIDP:long_name = "Differential Phase" ;
        PHIDP:units = "degree" ;
        PHIDP:standard_name = "differential_phase_hv" ;
        PHIDP:coordinates = "elevation azimuth range" ;
double RHOHV(time, range) ;
        RHOHV:_FillValue = -32768. ;
        RHOHV:long_name = "Cross-Polar Correlation Ratio" ;
        RHOHV:units = "1" ;
        RHOHV:standard_name = "cross_correlation_ratio_hv" ;
        RHOHV:coordinates = "elevation azimuth range" ;
double NCP(time, range) ;
        NCP:_FillValue = -32768. ;
        NCP:long_name = "Normalized Coherent Power, also known as SQI" ;
        NCP:units = "1" ;
        NCP:standard_name = "normalized_coherent_power" ;
        NCP:coordinates = "elevation azimuth range" ;
double DBZhv(time, range) ;
        DBZhv:_FillValue = -32768. ;
        DBZhv:long_name = "Equivalent Reflectivity Factor HV" ;
        DBZhv:units = "dBZ" ;
        DBZhv:standard_name = "equivalent_reflectivity_factor_hv" ;
        DBZhv:coordinates = "elevation azimuth range" ;
double cbb_flag(time, range) ;
        cbb_flag:_FillValue = NaN ;
        cbb_flag:long_name = "Cumulative Beam Block Fraction Flag" ;
        cbb_flag:units = "1" ;
        cbb_flag:coordinates = "elevation azimuth range" ;
        cbb_flag:comment = "Cumulative beam block flag due to terrain." ;
double sounding_temperature(time, range) ;
```

```
        sounding_temperature:_FillValue = NaN ;
        sounding_temperature:long_name = "Interpolated profile" ;
        sounding_temperature:units = "degC" ;
        sounding_temperature:standard_name = "interpolated_profile" ;
        sounding_temperature:missing_value = "-9999" ;
    double height(time, range) ;
        height:_FillValue = NaN ;
        height:long_name = "Height of radar beam" ;
        height:units = "m" ;
        height:standard_name = "height" ;
        height:missing_value = "-9999" ;
    double signal_to_noise_ratio(time, range) ;
        signal_to_noise_ratio:_FillValue = -32768. ;
        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:_FillValue = NaN ;
        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" ;
        velocity_texture:missing_value = "-9999" ;
    double gate_id(time, range) ;
        gate_id:_FillValue = NaN ;
        gate_id:long_name = "Classification of dominant scatterer" ;
        gate_id:units = "1" ;
        gate_id:notes =
"0:multi_trip,1:rain,2:snow,3:no_scatter,4:melting,5:clutter,6:terrain_blockage" ;
        gate_id:valid_max = "6" ;
        gate_id:valid_min = "0" ;
        gate_id:flag_values = "0, 1, 2, 3, 4, 5, 6" ;
        gate_id:flag_meanings = "multi_trip rain snow no_scatter melting clutter terrain_blockage" ;
    double simulated_velocity(time, range) ;
        simulated_velocity:_FillValue = NaN ;
        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" ;
    double corrected_velocity(time, range) ;
        corrected_velocity:_FillValue = -32768. ;
        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 = "-79.5" ;
                corrected_velocity:valid_max = "79.5" ;
        double unfolded_differential_phase(time, range) ;
                unfolded_differential_phase:_FillValue = -32768. ;
                unfolded_differential_phase:long_name = "Unfolded differential propagation phase shift" ;
                unfolded_differential_phase:units = "degree" ;
                unfolded_differential_phase:standard_name = "differential_phase_hv" ;
                unfolded_differential_phase:coordinates = "elevation azimuth range" ;
        double corrected_differential_phase(time, range) ;
                corrected_differential_phase:_FillValue = -32768. ;
                corrected_differential_phase:long_name = "Corrected differential propagation phase shift" ;
                corrected_differential_phase:units = "degree" ;
                corrected_differential_phase:standard_name = "differential_phase_hv" ;
                corrected_differential_phase:coordinates = "elevation azimuth range" ;
                corrected_differential_phase:valid_min = "0.0" ;
                corrected_differential_phase:valid_max = "400.0" ;
        double filtered_corrected_differential_phase(time, range) ;
                filtered_corrected_differential_phase:_FillValue = -32768. ;
                filtered_corrected_differential_phase:long_name = "Filtered Corrected Differential Phase" ;
                filtered_corrected_differential_phase:units = "degree" ;
                filtered_corrected_differential_phase:standard_name = "differential_phase_hv" ;
                filtered_corrected_differential_phase:coordinates = "elevation azimuth range" ;
                filtered_corrected_differential_phase:valid_min = "0.0" ;
                filtered_corrected_differential_phase:valid_max = "400.0" ;
        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" ;
	double height_over_iso0(time, range) ;
		height_over_iso0:_FillValue = NaN ;
		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 = "0.0" ;
		specific_attenuation:valid_max = "1.0" ;
		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 from Specific Attenuation" ;
		rain_rate_A:units = "mm/hr" ;
		rain_rate_A:standard_name = "rainfall_rate" ;
		rain_rate_A:valid_min = "0.0" ;
		rain_rate_A:valid_max = "400.0" ;
		rain_rate_A:coordinates = "elevation azimuth range" ;
		rain_rate_A:least_significant_digit = "1" ;
		rain_rate_A:comment = "Rain rate calculated from specific_attenuation,
R=43.5*specific_attenuation**0.79, note R=0.0 where norm coherent power < 0.4 or rhohv < 0.8" ;
	double snow_rate_ws2012(time, range) ;
		snow_rate_ws2012:_FillValue = 1.e+20 ;
		snow_rate_ws2012:long_name = "Snowfall rate from Z using Wolf and Snider (2012)" ;
		snow_rate_ws2012:units = "mm/h" ;
		snow_rate_ws2012:standard_name = "snowfall_rate" ;
		snow_rate_ws2012:coordinates = "elevation azimuth range" ;
```

```
        snow_rate_ws2012:valid_min = "0" ;
        snow_rate_ws2012:valid_max = "500" ;
        snow_rate_ws2012:swe_ratio = "13.699" ;
        snow_rate_ws2012:A = "110" ;
        snow_rate_ws2012:B = "2" ;
    double snow_rate_ws88diw(time, range) ;
        snow_rate_ws88diw:_FillValue = 1.e+20 ;
        snow_rate_ws88diw:long_name = "Snowfall rate from Z using WSR 88D High Plains" ;
        snow_rate_ws88diw:units = "mm/h" ;
        snow_rate_ws88diw:standard_name = "snowfall_rate" ;
        snow_rate_ws88diw:coordinates = "elevation azimuth range" ;
        snow_rate_ws88diw:valid_min = "0" ;
        snow_rate_ws88diw:valid_max = "500" ;
        snow_rate_ws88diw:swe_ratio = "13.699" ;
        snow_rate_ws88diw:A = "40" ;
        snow_rate_ws88diw:B = "2" ;
    double snow_rate_m2009_1(time, range) ;
        snow_rate_m2009_1:_FillValue = 1.e+20 ;
        snow_rate_m2009_1:long_name = "Snowfall rate from Z using Matrosov et al.(2009)
Braham(1990) 1" ;
        snow_rate_m2009_1:units = "mm/h" ;
        snow_rate_m2009_1:standard_name = "snowfall_rate" ;
        snow_rate_m2009_1:coordinates = "elevation azimuth range" ;
        snow_rate_m2009_1:valid_min = "0" ;
        snow_rate_m2009_1:valid_max = "500" ;
        snow_rate_m2009_1:swe_ratio = "13.699" ;
        snow_rate_m2009_1:A = "67" ;
        snow_rate_m2009_1:B = "1.28" ;
    double snow_rate_m2009_2(time, range) ;
        snow_rate_m2009_2:_FillValue = 1.e+20 ;
        snow_rate_m2009_2:long_name = "Snowfall rate from Z using Matrosov et al.(2009)
Braham(1990) 2" ;
        snow_rate_m2009_2:units = "mm/h" ;
        snow_rate_m2009_2:standard_name = "snowfall_rate" ;
        snow_rate_m2009_2:coordinates = "elevation azimuth range" ;
        snow_rate_m2009_2:valid_min = "0" ;
        snow_rate_m2009_2:valid_max = "500" ;
        snow_rate_m2009_2:swe_ratio = "13.699" ;
        snow_rate_m2009_2:A = "114" ;
        snow_rate_m2009_2:B = "1.39" ;
    double sweep_number(sweep) ;
        sweep_number:_FillValue = -9999. ;
        sweep_number:long_name = "Sweep index number 0 based" ;
        sweep_number:units = "1" ;
    double fixed_angle(sweep) ;
        fixed_angle:_FillValue = -9999. ;
        fixed_angle:long_name = "Ray Target Fixed Angle" ;
        fixed_angle:units = "degree" ;
```

```
double sweep_start_ray_index(sweep) ;
        sweep_start_ray_index:_FillValue = -9999. ;
        sweep_start_ray_index:long_name = "Index of First Ray in Sweep" ;
        sweep_start_ray_index:units = "1" ;
double sweep_end_ray_index(sweep) ;
        sweep_end_ray_index:_FillValue = -9999. ;
        sweep_end_ray_index:long_name = "Index of End Ray in Sweep" ;
        sweep_end_ray_index:units = "1" ;
double sweep_mode(sweep, string_length) ;
        sweep_mode:_FillValue = NaN ;
        sweep_mode:long_name = "Scan Mode of Sweep" ;
        sweep_mode:units = "1" ;
double nyquist_velocity(time) ;
        nyquist_velocity:_FillValue = -9999. ;
        nyquist_velocity:long_name = "Nyquist velocity" ;
        nyquist_velocity:units = "m/s" ;
        nyquist_velocity:standard_name = "nyquist_velocity" ;
double time_coverage_start(string_length) ;
        time_coverage_start:_FillValue = NaN ;
        time_coverage_start:long_name = "UTC time of first ray in the file" ;
        time_coverage_start:units = "1" ;
double time_coverage_end(string_length) ;
        time_coverage_end:_FillValue = NaN ;
        time_coverage_end:long_name = "UTC time of last ray in the file" ;
        time_coverage_end:units = "1" ;
double time_reference(string_length) ;
        time_reference:_FillValue = NaN ;
        time_reference:long_name = "UTC time reference" ;
        time_reference:units = "1" ;
double volume_number ;
        volume_number:_FillValue = NaN ;
        volume_number:long_name = "Volume number" ;
        volume_number:units = "1" ;
double latitude ;
        latitude:_FillValue = -9999. ;
        latitude:long_name = "Latitude" ;
        latitude:units = "degree_N" ;
        latitude:standard_name = "latitude" ;
        latitude:valid_min = "-90.0" ;
        latitude:valid_max = "90.0" ;
double longitude ;
        longitude:_FillValue = -9999. ;
        longitude:long_name = "Longitude" ;
        longitude:units = "degree_E" ;
        longitude:standard_name = "longitude" ;
        longitude:valid_min = "-180.0" ;
        longitude:valid_max = "180.0" ;
double altitude ;
```

```
                    altitude:_FillValue = -9999. ;
                    altitude:long_name = "Altitude" ;
                    altitude:units = "m" ;
                    altitude:standard_name = "altitude" ;

// global attributes:
                    :command_line = "" ;
                    :Conventions = "ARM-1.3 CF/Radial instrument_parameters" ;
                    :process_version = "" ;
                    :dod_version = "" ;
                    :input_datastreams = "" ;
                    :site_id = "" ;
                    :platform_id = "" ;
                    :facility_id = "" ;
                    :data_level = "" ;
                    :location_description = "" ;
                    :datastream = "" ;
                    :institution = "U.S. Department of Energy Atmospheric Radiation Measurement (ARM)
Climate Research Facility" ;
                    :references = "See XPRECIPRADAR Instrument Handbook" ;
                    :doi = "10.5439/1883164" ;
                    :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." ;
                    :vap_name = "cmac" ;
                    :known_issues = "False phidp jumps in insect regions. Still uses old Giangrande code. Issues
with some snow below melting layer." ;
                    :developers = "Robert Jackson, ANL. Zachary Sherman, ANL. Maxwell Grover, ANL.
Joseph O'Brien, ANL." ;
                    :translator = "https://www.arm.gov/capabilities/instruments/xprecipradar" ;
                    :mentors = "https://www.arm.gov/connect-with-arm/organization/instrument-
mentors/list#xprecipradar" ;
                    :source = "Colorado State University\'s X-Band Precipitation Radar (XPRECIPRADAR)
(DOI: 10.5439/1844501)" ;
                    :input_datastream = "xprecipradarS2.00" ;
                    :field_names = "DBZ, VEL, WIDTH, ZDR, PHIDP, RHOHV, NCP, DBZhv, cbb_flag,
sounding_temperature, height, signal_to_noise_ratio, velocity_texture, gate_id, simulated_velocity,
corrected_velocity, unfolded_differential_phase, corrected_differential_phase,
filtered_corrected_differential_phase, corrected_specific_diff_phase, filtered_corrected_specific_diff_phase,
corrected_differential_reflectivity, corrected_reflectivity, height_over_iso0, specific_attenuation,
path_integrated_attenuation, specific_differential_attenuation, path_integrated_differential_attenuation,
rain_rate_A, snow_rate_ws2012, snow_rate_ws88diw, snow_rate_m2009_1, snow_rate_m2009_2" ;
                    :history = "" ;
}
```

www.arm.gov

U.S. DEPARTMENT OF
ENERGY

Office of Science