U.S. DEPARTMENT OF ENERGY | Office of Science

DOE/SC-ARM-TR-243

Mapped Moments to a Cartesian Grid (MMCG) Value-Added Product Report

Z Sherman
S Collis
J Hemedinger

April 2020

ARM

DISCLAIMER

DOE/SC-ARM-TR-243

Mapped Moments to a Cartesian Grid (MMCG) Value-Added Product Report

Z Sherman
S Collis
J Hemedinger
All at Argonne National Laboratory

April 2020

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

Acronyms and Abbreviations

ARM	Atmospheric Radiation Measurement
CMAC	Corrected Moments in Antenna Coordinates
dBZ	decibel relative to Z
MMCG	Mapped Moments to a Cartesian Grid
NetCDF	Network Common Data Form
OA	objective analysis
Py-ART	Python ARM Radar Toolkit
ROI	radius of influence
SGP	Southern Great Plains
VAP	value-added product
X-SAPR	X-band Scanning ARM Precipitation Radar

Contents

Figures

Tables

1.0 Introduction

Objective analysis (OA) is a method of mapping unstructured data to a structured grid. In the context of scanning radar data, OA is used to interpolate data in antenna coordinates (range, azimuth, and elevation) onto a regularly spaced Cartesian grid (Trapp and Doswell 2000). The U.S. Department of Energy Atmospheric Radiation Measurement (ARM) user facility's Mapped Moments to a Cartesian Grid (MMCG) Value-Added Product (VAP) uses the Python ARM Radar Toolkit (Py-ART), a data model-driven interactive architecture for working with weather radar data, to map the data to a Cartesian grid (Helmus and Collis 2016). MMCG, with Py-ART built in, has the ability to take radar data in antenna coordinates and map the gates to a Cartesian grid using inverse distance weight functions such as Cressman (square) and Barnes (exponential), but also can filter the data during the interpolation. MMCG also allows arbitrary formulations for the radius of influence, which are matched to particular radar scanning strategies. This creates a complex parameter space for optimizing the retention of storm structure detail while minimizing artifacts. MMCG takes data processed with ARM's Corrected Precipitation Radar Moments in Antenna Coordinates (CMAC) VAP and maps it to a Cartesian grid as the output product. A variety of fields that have been mapped to the Cartesian grid are then saved to plots to complement each grid file.

2.0 Input Data

The input data for MMCG is the CMAC-processed data from the X-band Scanning ARM Precipitation Radar (X-SAPR) network created from the I4, I5, and I6 facilities at the ARM Southern Great Plains (SGP) atmospheric observatory. Table 1 lists locations and datastream information for the input data. The full list of the input variables is given by the NetCDF header file (see Appendix A).

Table 1. MMCG input datastreams, with corresponding dates and sites.

ARM site	ARM facility	Geographic location	MMCG input datastreams	Dates
SGP	I4	Billings, Oklahoma	sgpadicmac2I4.c1	2018/08/30 to 2019/01/22
SGP	I5	Garber, Oklahoma	sgpadicmac2I5.c1	2018/08/30 to 2019/04/05
SGP	I6	Deer Creek, Oklahoma	sgpadicmac2I6.c1	2018/08/30 to 2019/02/26

3.0 Algorithm and Methodology

Radar reflectivity is in dBZ units (decibel relative to Z). The idea of interpolating in linear Z units and not on a logarithmic scale can be found in Warren and Protat (2019) with findings that interpolation of reflectivity is more accurate in *Z* than in dBZ, especially in regions of high reflectivity and strong reflectivity gradient. These authors also noted that preservation of this detail is crucial when studying large rain rate events (Warren and Protat 2019).

With this is mind, MMCG takes the radar data and converts the reflectivity back to units of $mm^6 m^{-3}$ or Z, before interpolation of the radar data to a Cartesian grid. The conversion path can be seen in Figure 1 The interpolation itself on all fields, except hydrometeor classification field, uses a weight based on Cressman (1959):

$$weights = (r2 - dist2) / (r2 + dist2)$$

r2, being the radius of influence squared and dist2, the distance between two points squared. The ROI varies as a function of the distance from the radar (in all three directions, x, y, and height) and is formulated such that, for a given scanning pattern by the radar, there are enough radar gates to provide a reasonable estimate at the Cartesian location.

The hydrometeor classification field is interpolated with nearest neighbor, as the gates each have their own identification number. Once the interpolation is completed, the fields that were interpolated in linear Z are now converted back to logarithmic units of dBZ

$$dBZ = 10\log_{10}Z$$

The grid is then written containing the new coordinate system and linear interpolated (then converted back to dBZ) fields.

3.1 Flowchart

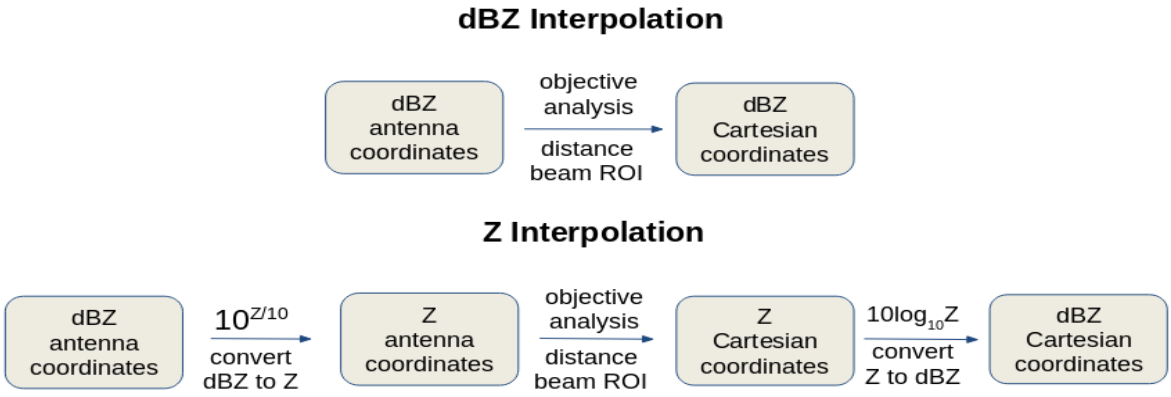

Figure 1. Flowchart of the conversion path of fields with dBZ units.

3.2 Quality Control Methods

The MMCG data does not have quality control methods applied, as the quality control was performed in the CMAC VAP package when the input datastreams for MMCG were created.

4.0 Output Data

The structure of the output data is very similar to the input data and is also in NetCDF format that complies to cfradial standards. The real difference is that the coordinate system has changed, and fields such as reflectivity and total power are interpolated in linear units and then converted to logarithmic units. The full list of the output variables is given by the NetCDF header file (see Appendix B). A full list of MMCG datastreams is found in Table 2.

Table 2. MMCG output datastreams with corresponding dates and sites.

ARM site	ARM facility	Geographic location	MMCG output datastreams	Dates
SGP	I4	Billings, Oklahoma	sgpxsaprmmcgI4.c1	2018/08/30 to 2019/01/22
SGP	I5	Garber, Oklahoma	sgpxsaprmmcgI5.c1	2018/08/30 to 2019/04/05
SGP	I6	Deer Creek, Oklahoma	sgpxsaprmmcgI6.c1	2018/08/30 to 2019/02/26

5.0 Summary

When trying to reduce artifacts while keeping structure, ROI will vary depending on scan strategy, beam spacing, and the meteorological event occurring.

6.0 Example Plots

The figures provided are a sample of a couple of the plots that are created from the sgpxsaprmmcg.c1 data sets. Figure 1 contains reflectivity that has been linear interpolated and then converted back to dBZ with cross-section views. Figure 2 contains hydrometeor classification with a cross-section view that was interpolated using nearest neighbor.

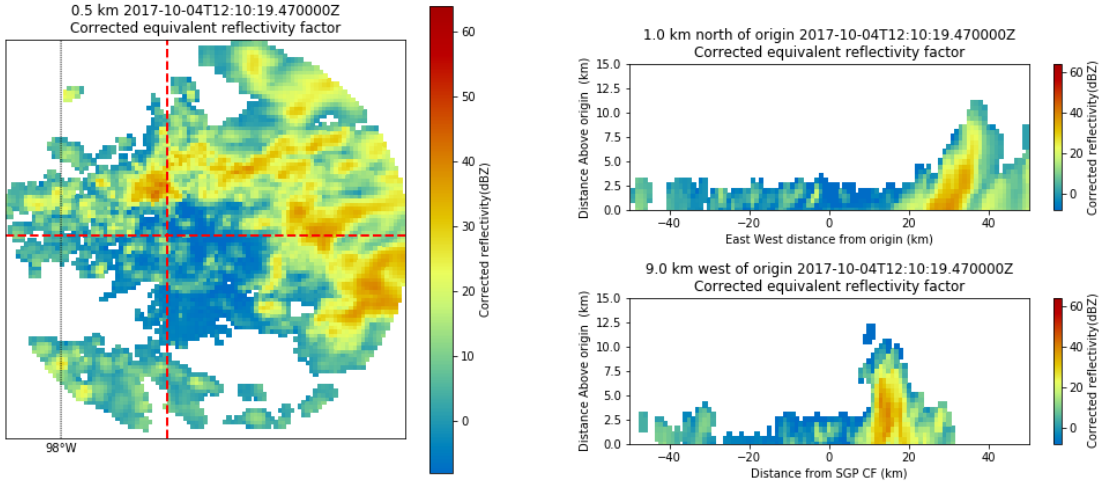

Figure 2. Radar reflectivity mapped to a Cartesian grid for October 4, 2017 using the Cressman interpolation method. Plot also contains two subplots of a latitudinal and longitudinal slice at 1km heights.

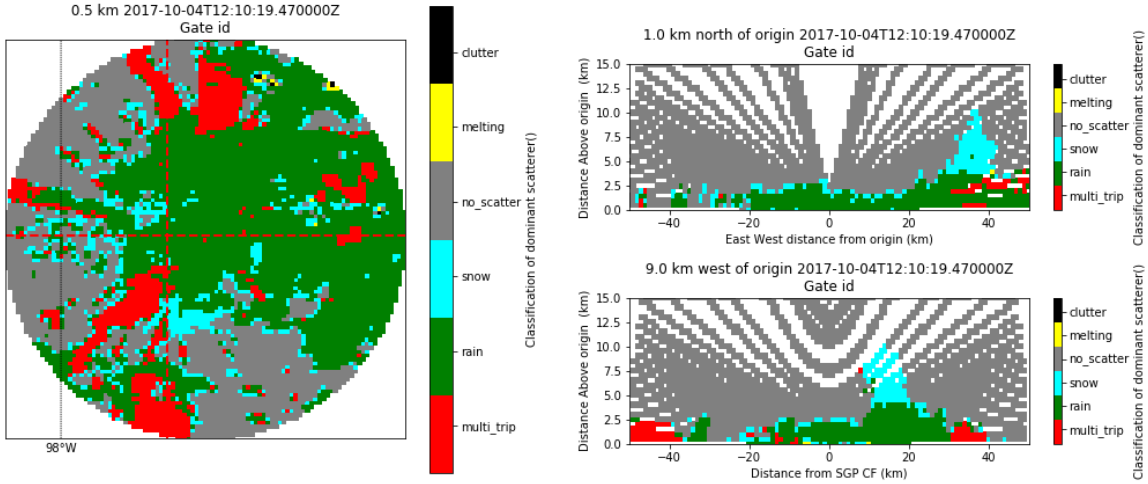

Figure 3. Hydrometeor classification mapped to a Cartesian grid for October 4, 2017 using nearest neighbor interpolation method. The plot shows IDs of ground clutter, rain, snow, multi trip, and melting. Plot also contains two subplots of a latitudinal and longitudinal slice at 1km heights.

7.0 Data Caveat/Uncertainty

Like any other gridded data grids products, MMCG data can suffer from the uncertainties introduced by interpolating/extrapolating data points to common.

8.0 References

Cressman, GP. 1959. “An operational objective analysis system.” *Monthly Weather Review* 87(10): 367–374, https://doi.org/10.1175/1520 0493(1959)087<0367:AOOAS>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), p.e25, https://doi.org/10.5334/jors.119

Trapp, RJ, and CA Doswell. 2000. “Radar Data Objective Analysis.” *Journal of Atmospheric and Oceanic Technology* 17(2): 105–120, https://doi.org/10.1175/1520-0426(2000)017<0105:RDOA>2.0.CO;2

Warren, RA, and A Protat. 2019. “Should Interpolation of Radar Reflectivity be Performed in Z or dBZ?” *Journal of Atmospheric and Oceanic Technology* 36(6): 1143–1156, https://doi.org/10.1175/JTECH-D-18-0183.1

Appendix A

Input Data

```
netcdf sgpadicmac2I4.c1.20190122.051031 {
dimensions:
        time = UNLIMITED ; // (6480 currently)
        range = 1001 ;
        sweep = 18 ;
        string_length_22 = 22 ;
variables:
        int base_time ;
                base_time:string = "22-Jan-2019,00:00:00 GMT" ;
                base_time:long_name = "Base time in Epoch" ;
                base_time:units = "seconds since 1970-1-1 0:00:00 0:00" ;
                base_time:ancillary_variables = "time_offset" ;
        double time_offset(time) ;
                time_offset:long_name = "Time offset from base_time" ;
                time_offset:units = "seconds since 2019-01-22 00:00:00 0:00" ;
                time_offset:ancillary_variables = "base_time" ;
                time_offset:calendar = "gregorian" ;
        double time(time) ;
                time:long_name = "Time offset from midnight" ;
                time:units = "seconds since 2019-01-22 00:00:00 0:00" ;
                time:calendar = "gregorian" ;
                time:comment = "Coordinate variable for time. Time at the center of each ray, in
fractional seconds since the global variable time_coverage_start" ;
                time:standard_name = "time" ;
        float range(range) ;
                range:long_name = "Range to measurement volume" ;
                range:units = "m" ;
                range:comment = "Coordinate variable for range. Range to center of each bin." ;
                range:standard_name = "projection_range_coordinate" ;
                range:meters_to_center_of_first_gate = 0.f ;
                range:meters_between_gates = 100.f ;
                range:spacing_is_constant = "true" ;
                range:axis = "radial_range_coordinate" ;
        float azimuth(time) ;
                azimuth:long_name = "Azimuth angle from true north" ;
                azimuth:units = "degree" ;
                azimuth:comment = "Azimuth of antenna relative to true north" ;
                azimuth:standard_name = "beam_azimuth_angle" ;
```

azimuth:axis = "radial_azimuth_coordinate" ;
float elevation(time) ;
elevation:long_name = "Elevation angle from horizontal plane" ;
elevation:units = "degree" ;
elevation:comment = "Elevation of antenna relative to the horizontal plane" ;
elevation:standard_name = "beam_elevation_angle" ;
elevation:axis = "radial_elevation_coordinate" ;
float reflectivity(time, range) ;
reflectivity:long_name = "Equivalent reflectivity factor" ;
reflectivity:units = "dBZ" ;
reflectivity:valid_min = -327.67f ;
reflectivity:valid_max = 327.66f ;
reflectivity:_FillValue = -9999.f ;
reflectivity:coordinates = "elevation azimuth range" ;
reflectivity:standard_name = "equivalent_reflectivity_factor" ;
float cross_correlation_ratio_hv(time, range) ;
cross_correlation_ratio_hv:long_name = "Cross correlation ratio (RhoHV)" ;
cross_correlation_ratio_hv:units = "unitless" ;
cross_correlation_ratio_hv:valid_min = 0.f ;
cross_correlation_ratio_hv:valid_max = 1.f ;
cross_correlation_ratio_hv:_FillValue = -9999.f ;
cross_correlation_ratio_hv:coordinates = "elevation azimuth range" ;
cross_correlation_ratio_hv:standard_name = "cross_correlation_ratio_hv" ;
float normalized_coherent_power(time, range) ;
normalized_coherent_power:long_name = "Normalized coherent power" ;
normalized_coherent_power:units = "unitless" ;
normalized_coherent_power:valid_min = 0.f ;
normalized_coherent_power:valid_max = 1.f ;
normalized_coherent_power:_FillValue = -9999.f ;
normalized_coherent_power:coordinates = "elevation azimuth range" ;
normalized_coherent_power:standard_name = "normalized_coherent_power" ;
float mean_doppler_velocity(time, range) ;
mean_doppler_velocity:long_name = "Mean Doppler velocity" ;
mean_doppler_velocity:units = "m/s" ;
mean_doppler_velocity:valid_min = -327.67f ;
mean_doppler_velocity:valid_max = 327.66f ;
mean_doppler_velocity:_FillValue = -9999.f ;
mean_doppler_velocity:coordinates = "elevation azimuth range" ;
mean_doppler_velocity:standard_name =
"radial_velocity_of_scatterers_away_from_instrument" ;
float total_power(time, range) ;
total_power:long_name = "Total power" ;
total_power:units = "dBZ" ;
total_power:valid_min = -327.67f ;
total_power:valid_max = 327.66f ;
total_power:_FillValue = -9999.f ;
total_power:coordinates = "elevation azimuth range" ;
total_power:standard_name = "equivalent_reflectivity_factor" ;
float spectral_width(time, range) ;
spectral_width:long_name = "Spectrum width" ;
spectral_width:units = "m/s" ;

spectral_width:valid_min = 0.01f ;
spectral_width:valid_max = 655.34f ;
spectral_width:_FillValue = -9999.f ;
spectral_width:coordinates = "elevation azimuth range" ;
spectral_width:standard_name = "doppler_spectrum_width" ;
float differential_reflectivity(time, range) ;
differential_reflectivity:long_name = "Differential reflectivity (ZDR)" ;
differential_reflectivity:units = "dB" ;
differential_reflectivity:valid_min = -327.67f ;
differential_reflectivity:valid_max = 327.66f ;
differential_reflectivity:_FillValue = -9999.f ;
differential_reflectivity:coordinates = "elevation azimuth range" ;
differential_reflectivity:standard_name = "log_differential_reflectivity_hv" ;
float specific_differential_phase(time, range) ;
specific_differential_phase:long_name = "Specific differential phase (KDP)" ;
specific_differential_phase:units = "degree/km" ;
specific_differential_phase:valid_min = -327.67f ;
specific_differential_phase:valid_max = 327.66f ;
specific_differential_phase:_FillValue = -9999.f ;
specific_differential_phase:coordinates = "elevation azimuth range" ;
specific_differential_phase:standard_name = "specific_differential_phase_hv" ;
float differential_phase(time, range) ;
differential_phase:long_name = "Differential phase (PhiDP)" ;
differential_phase:units = "degree" ;
differential_phase:valid_min = 0.f ;
differential_phase:valid_max = 359.99f ;
differential_phase:_FillValue = -9999.f ;
differential_phase:coordinates = "elevation azimuth range" ;
differential_phase:standard_name = "differential_phase_hv" ;
int sweep_number(sweep) ;
sweep_number:long_name = "Sweep number" ;
sweep_number:units = "count" ;
sweep_number:standard_name = "sweep_index_number_0_based" ;
float fixed_angle(sweep) ;
fixed_angle:long_name = "Target angle for sweep" ;
fixed_angle:units = "degree" ;
fixed_angle:standard_name = "target_fixed_angle" ;
int sweep_start_ray_index(sweep) ;
sweep_start_ray_index:long_name = "Index of first ray in sweep, 0-based" ;
sweep_start_ray_index:units = "count" ;
sweep_start_ray_index:standard_name = "index_of_first_ray_in_sweep" ;
int sweep_end_ray_index(sweep) ;
sweep_end_ray_index:long_name = "Index of last ray in sweep, 0-based" ;
sweep_end_ray_index:units = "count" ;
sweep_end_ray_index:standard_name = "index_of_last_ray_in_sweep" ;
char sweep_mode(sweep, string_length_22) ;
sweep_mode:long_name = "Sweep mode" ;
sweep_mode:units = "unitless" ;
sweep_mode:standard_name = "sweep_mode_for_sweep" ;
sweep_mode:comment = "Options are: sector coplane rhi vertical_pointing idle azimuth_surveillance elevation_surveillance sunscan pointing manual_ppi manual_rhi" ;

char prt_mode(sweep, string_length_22) ;
prt_mode:long_name = "PRT mode" ;
prt_mode:units = "unitless" ;
prt_mode:meta_group = "instrument_parameters" ;
prt_mode:options = "fixed, staggered, dual" ;
prt_mode:standard_name = "transmit_pulse_mode" ;
float nyquist_velocity(time) ;
nyquist_velocity:long_name = "Unambiguous Doppler velocity" ;
nyquist_velocity:units = "m/s" ;
nyquist_velocity:meta_group = "instrument_parameters" ;
nyquist_velocity:comments = "Unambiguous velocity" ;
nyquist_velocity:standard_name = "unambiguous_doppler_velocity" ;
float unambiguous_range(time) ;
unambiguous_range:long_name = "Unambiguous range" ;
unambiguous_range:units = "m" ;
unambiguous_range:meta_group = "instrument_parameters" ;
unambiguous_range:standard_name = "unambiguous_range" ;
float prt(time) ;
prt:long_name = "Pulse repetition time" ;
prt:units = "s" ;
prt:meta_group = "instrument_parameters" ;
prt:standard_name = "pulse_repetition_time" ;
int file_status ;
file_status:long_name = "File status" ;
file_status:units = "unitless" ;
file_status:flag_mask = 1, 2, 4, 8, 16 ;
file_status:flag_meaning = "interpolated_time_sec interpolated_time_ms approximated_time missing_rays no_kdp_variable" ;
file_status:bit_1_description = "Time in the file with second precision were interpolated to prevent repeated values. Rounding values in the time variable to the nearest second will recover the original times." ;
file_status:bit_2_description = "Time in the file with millisecond precision were interpolated to prevent repeated values. Rounding values in the time variable to the nearest millisecond will recover the original times." ;
file_status:bit_3_description = "Rays in one or more sweeps were not ordered sequentially in time in the raw file. Values in the time variable are approximated values, for exact values refer to the .00 files." ;
file_status:bit_4_description = "One or more sweeps are missing rays." ;
file_status:bit_5_description = "The specific_differential_phase variable was not collected, it contains only _FillValue." ;
char time_coverage_start(string_length_22) ;
time_coverage_start:long_name = "UTC time of first ray in the file" ;
time_coverage_start:units = "unitless" ;
time_coverage_start:standard_name = "data_volume_start_time_utc" ;
char time_coverage_end(string_length_22) ;
time_coverage_end:long_name = "UTC time of last ray in the file" ;
time_coverage_end:units = "unitless" ;
time_coverage_end:standard_name = "data_value_end_time_utc" ;
int volume_number ;
volume_number:long_name = "Volume number" ;
volume_number:units = "unitless" ;

volume_number:standard_name = "data_volume_index_number" ;
double latitude ;
latitude:long_name = "Latitude" ;
latitude:units = "degree_N" ;
latitude:standard_name = "latitude" ;
float n_samples(time) ;
n_samples:long_name = "Number of samples used to compute moments" ;
n_samples:units = "unitless" ;
n_samples:standard_name = "equivalent_reflectivity_factor" ;
n_samples:meta_group = "instrument_parameters" ;
float gate_id(time, range) ;
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" ;
gate_id:standard_name = "gate_id" ;
gate_id:valid_min = "0" ;
gate_id:valid_max = "5" ;
float SNR(time, range) ;
SNR:long_name = "Signal to noise ratio" ;
SNR:units = "dB" ;
SNR:standard_name = "signal_to_noise_ratio" ;
float velocity_texture(time, range) ;
velocity_texture:long_name = "Velocity texture" ;
velocity_texture:units = "meters_per_second" ;
velocity_texture:standard_name =
"radial_velocity_of_scatterers_away_from_instrument" ;
velocity_texture:_FillValue = -9999.f ;
velocity_texture:valid_min = -128.f ;
velocity_texture:valid_max = 128.f ;
float corrected_velocity(time, range) ;
corrected_velocity:long_name = "Corrected mean doppler velocity" ;
corrected_velocity:units = "meters_per_second" ;
corrected_velocity:standard_name =
"corrected_radial_velocity_of_scatters_away_from_instrument" ;
corrected_velocity:valid_min = -32.0849990844727 ;
corrected_velocity:valid_max = 32.0849990844727 ;
corrected_velocity:_FillValue = -9999.f ;
float unfolded_differential_phase(time, range) ;
unfolded_differential_phase:long_name = "Differential phase (PhiDP)" ;
unfolded_differential_phase:units = "degrees" ;
unfolded_differential_phase:_FillValue = -9999.f ;
unfolded_differential_phase:standard_name = "differential_phase_hv" ;
unfolded_differential_phase:valid_max = 180.f ;
unfolded_differential_phase:valid_min = -180.f ;
float corrected_differential_phase(time, range) ;
corrected_differential_phase:long_name = "Differential Phase (PhiDP)" ;
corrected_differential_phase:units = "degrees" ;
corrected_differential_phase:_FillValue = -9999.f ;
corrected_differential_phase:standard_name = "differential_phase_hv" ;
corrected_differential_phase:valid_max = 400.f ;
corrected_differential_phase:valid_min = 0.f ;

```
        float filtered_corrected_differential_phase(time, range) ;
                filtered_corrected_differential_phase:long_name = "Differential Phase (PhiDP)" ;
                filtered_corrected_differential_phase:units = "degrees" ;
                filtered_corrected_differential_phase:_FillValue = -9999.f ;
                filtered_corrected_differential_phase:standard_name = "differential_phase_hv" ;
                filtered_corrected_differential_phase:valid_max = 400.f ;
                filtered_corrected_differential_phase:valid_min = 0.f ;
        float specific_differential_attenuation(time, range) ;
                specific_differential_attenuation:long_name = "Specific differential attenuation" ;
                specific_differential_attenuation:units = "dB km-1" ;
                specific_differential_attenuation:standard_name = "specific_differential_attenuation" ;
        float specific_attenuation(time, range) ;
                specific_attenuation:long_name = "Specific attenuation" ;
                specific_attenuation:units = "dB/km" ;
                specific_attenuation:standard_name = "specific_attenuation" ;
        float corrected_specific_diff_phase(time, range) ;
                corrected_specific_diff_phase:long_name = "Specific differential phase (KDP)" ;
                corrected_specific_diff_phase:units = "degrees/km" ;
                corrected_specific_diff_phase:_FillValue = -9999.f ;
                corrected_specific_diff_phase:standard_name = "specific_differential_phase_hv" ;
        float filtered_corrected_specific_diff_phase(time, range) ;
                filtered_corrected_specific_diff_phase:long_name = "Specific differential phase (KDP)" ;
                filtered_corrected_specific_diff_phase:units = "degrees/km" ;
                filtered_corrected_specific_diff_phase:_FillValue = -9999.f ;
                filtered_corrected_specific_diff_phase:standard_name = "specific_differential_phase_hv"
;
        float path_integrated_differential_attenuation(time, range) ;
                path_integrated_differential_attenuation:long_name = "Path integrated differential
attenuation" ;
                path_integrated_differential_attenuation:units = "dB" ;
                path_integrated_differential_attenuation:standard_name =
"path_integrated_differential_attenuation" ;
        float corrected_reflectivity(time, range) ;
                corrected_reflectivity:long_name = "Corrected reflectivity" ;
                corrected_reflectivity:units = "dBZ" ;
                corrected_reflectivity:_FillValue = -9999.f ;
                corrected_reflectivity:standard_name = "corrected_equivalent_reflectivity_factor" ;
        float rain_rate_A(time, range) ;
                rain_rate_A:long_name = "rainfall_rate" ;
                rain_rate_A:units = "mm/hr" ;
                rain_rate_A:_FillValue = -9999.f ;
                rain_rate_A:valid_min = 0.f ;
                rain_rate_A:valid_max = 400.f ;
                rain_rate_A:comment = "Rain rate calculated from specific_attenuation,
R=51.3*specific_attenuation**0.81, note R=0.0 where norm coherent power < 0.4 or rhohv < 0.8" ;
                rain_rate_A:least_signficant_digit = 1 ;
        float radar_echo_classification(time, range) ;
                radar_echo_classification:long_name = "Radar echo classification" ;
                radar_echo_classification:units = "1" ;
                radar_echo_classification:standard_name = "radar_echo_classification" ;
                radar_echo_classification:flag_values = "0     1     2     3     4     5     6   255 65535" ;
```

```
		radar_echo_classification:flag_meanings = "no_data_available
non_meteorological_target rain wet_snow snow graupel hail area_not_scanned" ;
	double longitude ;
		longitude:long_name = "Longitude" ;
		longitude:units = "degree_E" ;
		longitude:standard_name = "longitude" ;
	double altitude ;
		altitude:long_name = "Altitude" ;
		altitude:units = "m" ;
		altitude:standard_name = "altitude" ;
		altitude:positive = "up" ;
	float path_integrated_attenuation(time, range) ;
		path_integrated_attenuation:long_name = "Path integrated attenuation" ;
		path_integrated_attenuation:units = "dB" ;
		path_integrated_attenuation:standard_name = "path_integrated_attenuation" ;
	float corrected_differential_reflectivity(time, range) ;
		corrected_differential_reflectivity:long_name = "Corrected differential reflectivity" ;
		corrected_differential_reflectivity:units = "dB" ;
		corrected_differential_reflectivity:standard_name = "corrected_differential_reflectivity" ;
	float ground_clutter(time, range) ;
		ground_clutter:long_name = "Ground clutter flag" ;
		ground_clutter:units = "0 = no clutter, 1 = clutter" ;
	float lat ;
		lat:long_name = "North latitude" ;
		lat:units = "degree_N" ;
		lat:standard_name = "latitude" ;
		lat:valid_min = -90.f ;
		lat:valid_max = 90.f ;
	float lon ;
		lon:long_name = "East longitude" ;
		lon:units = "degree_E" ;
		lon:standard_name = "longitude" ;
		lon:valid_min = -180.f ;
		lon:valid_max = 180.f ;
	float alt ;
		alt:long_name = "Altitude above mean sea level" ;
		alt:units = "m" ;
		alt:standard_name = "altitude" ;

// global attributes:
		:Conventions = "ARM-1.0 CF/Radial instrument_parameters" ;
		:title = "Atmospheric Radiation Measurement (ARM) program X-band Scanning ARM
Precipitation Radar (XSAPR) raw moments" ;
		:institution = "United States Department of Energy - Atmospheric Radiation
Measurement (ARM) program" ;
		:references = "See XSAPR Instrument Handbook" ;
		:source = "Atmospheric Radiation Measurement (ARM) program X-band Scanning ARM
Precipitation Radar (XSAPR)" ;
		:comment = "Data in this file has not be calibrated, corrected, or had any quality control
performed, use with caution." ;
		:instrument_name = "XSAPR" ;
```

:platform_is_mobile = "false" ;
:n_gates_vary = "false" ;
:field_names = "total_power, reflectivity, mean_doppler_velocity, spectral_width, differential_reflectivity, specific_differential_phase, cross_correlation_ratio_hv, normalized_coherent_power, differential_phase" ;
:command_line = "adi_cmac2 -R -D 1 -f I4 -s sgp -b 20190122 -e 20190123" ;
:process_version = "vap-adi_cmac2-1.0-0.dev4.dirty.el7" ;
:dod_version = "adicmac2-c1-1.0" ;
:input_source = "/data/collection/sgp/sgpxsaprI4.00/XSE190122051029.RAWKZVF" ;
:site_id = "sgp" ;
:platform_id = "adicmac2" ;
:facility_id = "I4" ;
:data_level = "c1" ;
:location_description = "Southern Great Plains (SGP), Billings, Oklahoma" ;
:datastream = "sgpadicmac2I4.c1" ;
:original_container = "sigmet" ;
:history = "created by user rjackson on machine or-condo-c215.ornl.gov at 2019-05-01 19:23:50, using vap-adi_cmac2-1.0-0.dev4.dirty.el7" ;

Appendix B

Output Data

```
netcdf sgpxsaprmmcgI5.c1.20190204.120009 {
dimensions:
        time = UNLIMITED ; // (1 currently)
        z = 31 ;
        y = 101 ;
        x = 101 ;
        nradar = 1 ;
        nradar_str_length = 5 ;
variables:
        double time(time) ;
                time:long_name = "Time offset from midnight" ;
                time:units = "seconds since 2019-02-04 00:00:00 0:00" ;
                time:standard_name = "time" ;
                time:calendar = "gregorian" ;
        double x(x) ;
                x:long_name = "X distance on the projection plane from the origin" ;
                x:units = "m" ;
                x:standard_name = "projection_x_coordinate" ;
                x:axis = "X" ;
        double y(y) ;
                y:long_name = "Y distance on the projection plane from the origin" ;
                y:units = "m" ;
                y:standard_name = "projection_y_coordinate" ;
                y:axis = "Y" ;
        double z(z) ;
                z:long_name = "Z distance on the projection plane from the origin" ;
                z:units = "m" ;
                z:standard_name = "projection_z_coordinate" ;
                z:axis = "Z" ;
                z:positive = "up" ;
        double origin_latitude(time) ;
                origin_latitude:long_name = "Latitude at grid origin" ;
                origin_latitude:units = "degree_N" ;
                origin_latitude:standard_name = "latitude" ;
                origin_latitude:valid_min = -90. ;
                origin_latitude:valid_max = 90. ;
        double origin_longitude(time) ;
                origin_longitude:long_name = "Longitude at grid origin" ;
```

origin_longitude:units = "degree_E" ;
origin_longitude:standard_name = "longitude" ;
origin_longitude:valid_min = -180. ;
origin_longitude:valid_max = 180. ;
double origin_altitude(time) ;
origin_altitude:long_name = "Altitude at grid origin" ;
origin_altitude:units = "m" ;
origin_altitude:standard_name = "altitude" ;
int projection ;
projection:proj = "pyart_aeqd" ;
projection:_include_lon_0_lat_0 = "true" ;
int ProjectionCoordinateSystem ;
ProjectionCoordinateSystem:latitude_of_projection_origin = 36.4915008544922 ;
ProjectionCoordinateSystem:longitude_of_projection_origin = -97.5938034057617 ;
ProjectionCoordinateSystem:_CoordinateTransformType = "Projection" ;
ProjectionCoordinateSystem:_CoordinateAxes = "x y z time" ;
ProjectionCoordinateSystem:_CoordinateAxesTypes = "GeoX GeoY Height Time" ;
ProjectionCoordinateSystem:grid_mapping_name = "azimuthal_equidistant" ;
ProjectionCoordinateSystem:semi_major_axis = 6370997. ;
ProjectionCoordinateSystem:inverse_flattening = 298.25 ;
ProjectionCoordinateSystem:longitude_of_prime_meridian = 0. ;
ProjectionCoordinateSystem:false_easting = 0. ;
ProjectionCoordinateSystem:false_northing = 0. ;
double radar_latitude(nradar) ;
radar_latitude:long_name = "Latitude of radars used to make the grid." ;
radar_latitude:units = "degree_N" ;
radar_latitude:standard_name = "Latitude" ;
radar_latitude:valid_min = -90. ;
radar_latitude:valid_max = 90. ;
double radar_longitude(nradar) ;
radar_longitude:long_name = "Longitude of radars used to make the grid." ;
radar_longitude:units = "degree_E" ;
radar_longitude:standard_name = "Longitude" ;
radar_longitude:valid_min = -180. ;
radar_longitude:valid_max = 180. ;
double radar_altitude(nradar) ;
radar_altitude:long_name = "Altitude of radars used to make the grid." ;
radar_altitude:units = "m" ;
radar_altitude:standard_name = "altitude" ;
double radar_time(nradar) ;
radar_time:long_name = "Time in seconds of the volume start for each radar" ;
radar_time:units = "seconds since 2019-02-04T12:00:09Z" ;
radar_time:calendar = "gregorian" ;
char radar_name(nradar, nradar_str_length) ;
radar_name:long_name = "Name of radar used to make the grid" ;
int base_time ;
base_time:long_name = "Base time in Epoch" ;
base_time:units = "seconds since 1970-1-1 0:00:00 0:00" ;
base_time:string = "04-Feb-2019,12:00:09 GMT" ;
base_time:ancillary_variables = "time_offset" ;
double time_offset(time) ;

time_offset:long_name = "Time offset from base_time" ;
time_offset:units = "seconds since 2019-02-04 00:00:00 0:00" ;
time_offset:ancillary_variables = "time_offset" ;
time_offset:calendar = "gregorian" ;
double alt ;
alt:long_name = "Altitude above mean sea level" ;
alt:units = "m" ;
alt:standard_name = "Altitude" ;
double lat ;
lat:long_name = "North Latitude" ;
lat:units = "degree_N" ;
lat:standard_name = "Latitude" ;
lat:valid_min = -90. ;
lat:valid_max = 90. ;
double lon ;
lon:long_name = "East Longitude" ;
lon:units = "degree_E" ;
lon:standard_name = "Longitude" ;
lon:valid_min = -180. ;
lon:valid_max = 180. ;
float normalized_coherent_power(time, z, y, x) ;
normalized_coherent_power:long_name = "Normalized coherent power" ;
normalized_coherent_power:units = "1" ;
normalized_coherent_power:_FillValue = -9999.f ;
normalized_coherent_power:valid_min = 0.f ;
normalized_coherent_power:valid_max = 1.f ;
normalized_coherent_power:coordinates = "elevation azimuth range" ;
normalized_coherent_power:standard_name = "normalized_coherent_power" ;
float specific_differential_attenuation(time, z, y, x) ;
specific_differential_attenuation:long_name = "Specific differential attenuation" ;
specific_differential_attenuation:units = "dB/km" ;
specific_differential_attenuation:standard_name = "specific_differential_attenuation" ;
float corrected_reflectivity(time, z, y, x) ;
corrected_reflectivity:long_name = "Corrected reflectivity" ;
corrected_reflectivity:units = "dBZ" ;
corrected_reflectivity:_FillValue = -9999.f ;
corrected_reflectivity:standard_name = "equivalent_reflectivity_factor" ;
corrected_reflectivity:comment = "This corrected_reflectivity field was interpolated linearly and then converted to logarithmic units. Using linear units during interpolation allows for the retention of storm structure and gives a more realistic estimation of convection and more." ;
float radar_echo_classification(time, z, y, x) ;
radar_echo_classification:long_name = "Radar echo classification" ;
radar_echo_classification:units = "1" ;
radar_echo_classification:standard_name = "radar_echo_classification" ;
radar_echo_classification:flag_values = "0 1 2 3 4 5 6 255 65535" ;
radar_echo_classification:flag_meanings = "no_data_available non_meteorological_target rain wet_snow snow graupel hail area_not_scanned" ;
float cross_correlation_ratio_hv(time, z, y, x) ;
cross_correlation_ratio_hv:long_name = "Cross correlation ratio (RhoHV)" ;
cross_correlation_ratio_hv:units = "1" ;
cross_correlation_ratio_hv:_FillValue = -9999.f ;

cross_correlation_ratio_hv:valid_min = 0.f ;
cross_correlation_ratio_hv:valid_max = 1.f ;
cross_correlation_ratio_hv:coordinates = "elevation azimuth range" ;
cross_correlation_ratio_hv:standard_name = "cross_correlation_ratio_hv" ;
float ground_clutter(time, z, y, x) ;
ground_clutter:long_name = "Ground clutter flag" ;
ground_clutter:units = "1" ;
ground_clutter:_FillValue = -9999.f ;
ground_clutter:flag_values = "0, 1" ;
ground_clutter:flag_meanings = "no_clutter, clutter" ;
float corrected_velocity(time, z, y, x) ;
corrected_velocity:long_name = "Corrected mean doppler velocity" ;
corrected_velocity:units = "m/s" ;
corrected_velocity:_FillValue = -9999.f ;
corrected_velocity:standard_name =
"radial_velocity_of_scatterers_away_from_instrument" ;
corrected_velocity:valid_min = -32.0849990844727 ;
corrected_velocity:valid_max = 32.0849990844727 ;
float corrected_specific_diff_phase(time, z, y, x) ;
corrected_specific_diff_phase:long_name = "Corrected specific differential phase
(KDP)" ;
corrected_specific_diff_phase:units = "degree/km" ;
corrected_specific_diff_phase:_FillValue = -9999.f ;
corrected_specific_diff_phase:standard_name = "specific_differential_phase_hv" ;
float specific_attenuation(time, z, y, x) ;
specific_attenuation:long_name = "Specific attenuation" ;
specific_attenuation:units = "dB/km" ;
specific_attenuation:standard_name = "specific_attenuation" ;
float reflectivity(time, z, y, x) ;
reflectivity:long_name = "Equivalent reflectivity factor" ;
reflectivity:units = "dBZ" ;
reflectivity:_FillValue = -9999.f ;
reflectivity:valid_min = -327.67f ;
reflectivity:valid_max = 327.66f ;
reflectivity:coordinates = "elevation azimuth range" ;
reflectivity:standard_name = "equivalent_reflectivity_factor" ;
reflectivity:comment = "This reflectivity field was interpolated linearly and then
converted to logarithmic units. Using linear units during interpolation allows for the retention of storm
structure and gives a more realistic estimation of convection and more." ;
float path_integrated_attenuation(time, z, y, x) ;
path_integrated_attenuation:long_name = "Path integrated attenuation" ;
path_integrated_attenuation:units = "dB" ;
path_integrated_attenuation:_FillValue = -9999.f ;
path_integrated_attenuation:standard_name = "path_integrated_attenuation" ;
float mean_doppler_velocity(time, z, y, x) ;
mean_doppler_velocity:long_name = "Mean Doppler velocity" ;
mean_doppler_velocity:units = "m/s" ;
mean_doppler_velocity:_FillValue = -9999.f ;
mean_doppler_velocity:valid_min = -327.67f ;
mean_doppler_velocity:valid_max = 327.66f ;
mean_doppler_velocity:coordinates = "elevation azimuth range" ;

mean_doppler_velocity:standard_name = "radial_velocity_of_scatterers_away_from_instrument" ;
float total_power(time, z, y, x) ;
total_power:long_name = "Total power" ;
total_power:units = "dBZ" ;
total_power:_FillValue = -9999.f ;
total_power:valid_min = -327.67f ;
total_power:valid_max = 327.66f ;
total_power:coordinates = "elevation azimuth range" ;
total_power:standard_name = "equivalent_reflectivity_factor" ;
total_power:comment = "This total power field was interpolated linearly and then converted to logarithmic units. Using linear units during interpolation allows for the retention of storm structure and gives a more realistic estimation of convection and more." ;
float filtered_corrected_specific_diff_phase(time, z, y, x) ;
filtered_corrected_specific_diff_phase:long_name = "Filtered specific differential phase (KDP)" ;
filtered_corrected_specific_diff_phase:units = "degree/km" ;
filtered_corrected_specific_diff_phase:_FillValue = -9999.f ;
filtered_corrected_specific_diff_phase:standard_name = "specific_differential_phase_hv" ;
float spectral_width(time, z, y, x) ;
spectral_width:long_name = "Spectrum width" ;
spectral_width:units = "m/s" ;
spectral_width:_FillValue = -9999.f ;
spectral_width:valid_min = 0.01f ;
spectral_width:valid_max = 655.34f ;
spectral_width:coordinates = "elevation azimuth range" ;
spectral_width:standard_name = "doppler_spectrum_width" ;
float path_integrated_differential_attenuation(time, z, y, x) ;
path_integrated_differential_attenuation:long_name = "Path integrated differential attenuation" ;
path_integrated_differential_attenuation:units = "dB" ;
path_integrated_differential_attenuation:_FillValue = -9999.f ;
path_integrated_differential_attenuation:standard_name = "path_integrated_differential_attenuation" ;
float specific_differential_phase(time, z, y, x) ;
specific_differential_phase:long_name = "Specific differential phase (KDP)" ;
specific_differential_phase:units = "degree/km" ;
specific_differential_phase:_FillValue = -9999.f ;
specific_differential_phase:valid_min = -327.67f ;
specific_differential_phase:valid_max = 327.66f ;
specific_differential_phase:coordinates = "elevation azimuth range" ;
specific_differential_phase:standard_name = "specific_differential_phase_hv" ;
float unfolded_differential_phase(time, z, y, x) ;
unfolded_differential_phase:long_name = "Unfolded differential phase (PhiDP)" ;
unfolded_differential_phase:units = "degree" ;
unfolded_differential_phase:_FillValue = -9999.f ;
unfolded_differential_phase:standard_name = "differential_phase_hv" ;
unfolded_differential_phase:valid_max = 180.f ;
unfolded_differential_phase:valid_min = -180.f ;
float differential_reflectivity(time, z, y, x) ;

differential_reflectivity:long_name = "Differential reflectivity (ZDR)" ;
differential_reflectivity:units = "dB" ;
differential_reflectivity:_FillValue = -9999.f ;
differential_reflectivity:valid_min = -327.67f ;
differential_reflectivity:valid_max = 327.66f ;
differential_reflectivity:coordinates = "elevation azimuth range" ;
differential_reflectivity:standard_name = "log_differential_reflectivity_hv" ;
float differential_phase(time, z, y, x) ;
differential_phase:long_name = "Differential phase (PhiDP)" ;
differential_phase:units = "degree" ;
differential_phase:_FillValue = -9999.f ;
differential_phase:valid_min = 0.f ;
differential_phase:valid_max = 359.99f ;
differential_phase:coordinates = "elevation azimuth range" ;
differential_phase:standard_name = "differential_phase_hv" ;
float gate_id(time, z, y, x) ;
gate_id:long_name = "Classification of dominant scatterer" ;
gate_id:units = "1" ;
gate_id:standard_name = "gate_id" ;
gate_id:valid_min = "0" ;
gate_id:valid_max = 5LL ;
gate_id:comment = "This gate id field has been mapped to a Cartesian grid using nearest neighbor. This may differ from the mapping method used in the other fields" ;
gate_id:flag_values = "1, 2, 3, 4, 5" ;
gate_id:flag_meanings = "multi_trip, rain, snow, no_scatter, melting, clutter" ;
float rain_rate_A(time, z, y, x) ;
rain_rate_A:long_name = "rainfall_rate" ;
rain_rate_A:units = "mm/hr" ;
rain_rate_A:_FillValue = -9999.f ;
rain_rate_A:valid_min = 0.f ;
rain_rate_A:valid_max = 400.f ;
rain_rate_A:comment = "Rain rate calculated from specific_attenuation, R=51.3*specific_attenuation**0.81, note R=0.0 where norm coherent power < 0.4 or rhohv < 0.8" ;
rain_rate_A:least_signficant_digit = 1 ;
float corrected_differential_phase(time, z, y, x) ;
corrected_differential_phase:long_name = "Corrected differential phase (PhiDP)" ;
corrected_differential_phase:units = "degree" ;
corrected_differential_phase:_FillValue = -9999.f ;
corrected_differential_phase:standard_name = "differential_phase_hv" ;
corrected_differential_phase:valid_max = 400.f ;
corrected_differential_phase:valid_min = 0.f ;
float velocity_texture(time, z, y, x) ;
velocity_texture:long_name = "Mean doppler velocity texture" ;
velocity_texture:units = "m/s" ;
velocity_texture:_FillValue = -9999.f ;
velocity_texture:standard_name = "radial_velocity_of_scatterers_away_from_instrument" ;
velocity_texture:valid_min = -128.f ;
velocity_texture:valid_max = 128.f ;
float corrected_differential_reflectivity(time, z, y, x) ;
corrected_differential_reflectivity:long_name = "Corrected differential reflectivity" ;

corrected_differential_reflectivity:units = "dB" ;
corrected_differential_reflectivity:standard_name = "corrected_differential_reflectivity" ;
float SNR(time, z, y, x) ;
SNR:long_name = "Signal to noise ratio" ;
SNR:units = "dB" ;
SNR:_FillValue = -9999.f ;
SNR:standard_name = "signal_to_noise_ratio" ;
float filtered_corrected_differential_phase(time, z, y, x) ;
filtered_corrected_differential_phase:long_name = "Filtered differential phase (PhiDP)" ;
filtered_corrected_differential_phase:units = "degree" ;
filtered_corrected_differential_phase:_FillValue = -9999.f ;
filtered_corrected_differential_phase:standard_name = "differential_phase_hv" ;
filtered_corrected_differential_phase:valid_max = 400.f ;
filtered_corrected_differential_phase:valid_min = 0.f ;
float ROI(time, z, y, x) ;
ROI:long_name = "Radius of influence for mapping" ;
ROI:units = "m" ;
ROI:_FillValue = -9999.f ;
ROI:standard_name = "radius_of_influence" ;
ROI:least_significant_digit = 1LL ;

// global attributes:
:Conventions = "ARM-1.2" ;
:vap_name = "mmcg" ;
:instrument_name = "X_SAPR Mapped Moments toa Cartesian Grid"
:process_version = "EVAL-0.5" ;
:dod_version = "v1.0" ;
:site_id = "SGP" ;
:platform_id = "xsaprmmcg" ;
:facility_id = "I5:Garber, OK" ;
:data_level = "c1" ;
:location_description = "Southern Great Plains (SGP), Garber, Oklahoma" ;
:datastream = "sgpxsaprmmcgI5.c1" ;
:doi = "" ;
:input_datastream = "sgpxsaprcmacsurI5.c1" ;
:history = "Created by jhemedinger and zsherman on or-condo-c00.ornl.gov at 2019-06-11T20:43:00.880860 using PyART" ;

www.arm.gov

U.S. DEPARTMENT OF
ENERGY

Office of Science