U.S. DEPARTMENT OF ENERGY | Office of Science

DOE/SC-ARM-TR-314

ARM Trajectories Data Set Value-Added Product Report

I Silber
MR Kieburtz
F Mei
JM Comstock
KL Gaustad

January 2025

ARM

DISCLAIMER

DOE/SC-ARM-TR-314

ARM Trajectories Data Set Value-Added Product Report

I Silber
JM Comstock
MR Kieburtz
KL Gaustad
F Mei
All at Pacific Northwest National Laboratory

January 2025

How to cite this document:

Silber, I, JM Comstock, MR Kieburtz, KL Gaustad, and F Mei. 2025. ARM Trajectories Data Set Value-Added Product Report. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-TR-314.

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

Executive Summary

The U.S. Department of Energy Atmospheric Radiation Measurement (ARM) user facility's ARM Trajectories Data Set (ARMTRAJ) Value-Added Product (VAP) provides trajectory data sets initialized at ARM deployment coordinates and configured using ARM data sets. The four trajectory data sets support aerosol, cloud, and planetary boundary-layer research. Trajectory calculations use the Hybrid Single-Particle Lagrangian Integrated Trajectory (HYSPLIT) model informed by the European Centre for Medium-Range Weather Forecasts (ECMWF) fifth-generation atmospheric reanalysis (ERA5) data set at its highest spatial resolution (~31 km). HYSPLIT also runs at multiple initial starting locations surrounding ARM deployments (in latitude/longitude and/or vertical coordinates), facilitating an ensemble for each sample in the data sets. The ensemble mean and variability reported in ARMTRAJ improve the fidelity and provide uncertainty estimates of trajectory coordinates, thermodynamic properties, and other output fields.

Acronyms and Abbreviations

AAF	ARM Aerial Facility
AGL	above ground level
AMSL	above mean sea level
ARL	Air Resources Laboratory
ARM	Atmospheric Radiation Measurement
ARMTRAJ	ARM Trajectories Data Set Value-Added Product
ARSCL	Active Remote Sensing of Clouds Value-Added Product
CLD	cloud
ECMWF	European Centre for Medium-Range Weather Forecasts
EPCAPE	Eastern Pacific Cloud Aerosol Precipitation Experiment
ERA5	ECMWF fifth-generation atmospheric reanalysis
FT	free troposphere
HYSPLIT	Hybrid Single-Particle Lagrangian Integrated Trajectory
IFS	Integrated Forecasting System
MET	ARM surface meteorological system
NetCDF	Network Common Data Form
PBL	planetary boundary layer
PBLH	planetary boundary-layer height
PBLHT	Planetary Boundary Layer Height Value-Added Product
PDF	probability density function
RH	relative humidity
SGP	Southern Great Plains
SONDE	balloon-borne sounding system
TBS	tethered balloon system
UAS	uncrewed aerial system
UTC	Coordinated Universal Time
VAP	value-added product

Contents

Figures

Tables

1.0 Introduction

There is a growing recognition in the atmospheric science community of the importance of Lagrangian considerations wherever atmospheric dynamics play a crucial role. This recognition is manifested in the implementation of Lagrangian components in numerous aerosol and cloud studies. Better characterization of cloud life cycles, for example, often necessitates knowledge about the hysteresis and origin of cloudy airmasses. Airmass trajectory calculations can inform on potential cloud formation and evolution mechanisms (e.g., Mohrmann et al. 2019, Silber and Shupe 2022, Svensson et al. 2023), and support modeling studies by providing boundary conditions and observational benchmarks (e.g., Silber et al. 2019, Tornow et al. 2022), especially in cases with either upwind or downwind ground-based site overpass (e.g., Ali and Pithan 2020). Airmass trajectory calculations are also useful for understanding aerosol hysteresis and potential indirect effects. For example, trajectories can be used to estimate aerosol source origin and quantify aerosol processing during short-to-long-range transport to ground-based measurement stations (e.g., Hawkins and Russell 2010, Liu et al. 2011, Zheng et al. 2020).

ARM field campaigns, fixed sites, and the ARM Aerial Facility (AAF) provide high-end, unique measurement data sets, which trajectory calculations can augment. Trajectory data products for different ARM sites and deployments can enhance the usability of ARM data sets as they close some gaps ensuing from the fixed-location (Eulerian) nature of many ARM measurements. Characterization of airmass history facilitates clustering and/or classification of specific events, which could strengthen the results of studies focusing on processes with weak direct signals and provide metrics applicable for categorizing epochs and periods of interest (e.g., Silber and Shupe 2022). Here, we describe the ARMTRAJ VAP, which includes trajectory data sets supporting cloud, aerosol, and planetary boundary-layer (PBL) studies relying on ARM data. Additional information about ARMTRAJ can be found in a recent data article (Silber et al. 2025).

2.0 Processing and Data Sets

The ARMTRAJ VAP uses the HYSPLIT model (Stein et al. 2015) informed by the ECMWF ERA5 reanalysis (Hersbach et al. 2020) at its highest spatial resolution (0.25 degrees; ~31 km). The VAP data sets differ in their purpose and, therefore, in the initialization of the HYSPLIT model runs, relying on information from different ARM data sets (see sub-section below).

All ARMTRAJ data sets include ensemble runs initialized at relatively small horizontal and vertical offsets from ARM site coordinates. The vertical offset differs between data sets. However, except for the AAF trajectory data set (see sect. 2.5), the horizontal offsets are fixed at ± 7.5 km in the east-west and north-south directions, thereby defining a 3×3 horizontal grid around ARM sites and rendering ensemble sizes of multiplicities of nine. This ensemble starting horizontal extent covers roughly half the horizontal dimension of ERA5 grid cells, allowing the evaluation of ensembles' physical variability yet keeping them initially constrained to the site vicinity. The fixed geodetic distance in metric units rather than in arc degrees is used to ensure ARMTRAJ's ensemble configuration consistency when initialized in different geographic regions, such as ARM's North Slope of Alaska site, where a given longitudinal arc length translates to shorter geodetic distances relative to lower-latitude ARM sites such as the Eastern North Atlantic observatory in the Azores. ARMTRAJ reports the mean and standard deviation of all airmass

coordinate and thermodynamic variables, as well as most surface attribute variables. The ensemble means increase the fidelity of trajectory results, whereas the standard deviation of variables can be treated as a measure of trajectory estimated uncertainty.

The five ARMTRAJ data sets are ARMTRAJ-SFC, ARMTRAJ-CLD, ARMTRAJ-PBL, ARMTRAJ-ARSCL, and ARMTRAJ-AAF, which support surface and airborne in situ measurement, liquid-bearing cloud layer, PBL, and primary cloud deck analyses, respectively. Properties such as temporal resolution and trajectory periods of each data set are given in Table 1. The following sub-sections briefly describe each of these data sets.

Table 1. ARMTRAJ data set summary.

Dataset name	Initialization time	Free tropospheric run	Ensemble members	Back trajectory period [hours]	Forward trajectory period [hours]
ARMTRAJ-SFC	00, 03, 06, 09, 12, 15, 18, and 21 UTC	No	18	240	-
ARMTRAJ-CLD	SONDE release rounded to the nearest hour	No	27*	120	120
ARMTRAJ-PBL	SONDE release rounded to the nearest hour	Yes	99**	120	-
ARMTRAJ-ARSCL	00, 03, 06, 09, 12, 15, 18, and 21 UTC	Yes	99**	120	120
ARMTRAJ-AAF	Hours during which AAF systems are operated	No***	25****	120	-

* Per detected liquid-bearing cloud layer (see section 2.2)

** Ensemble size of 9 in free-tropospheric runs (see section 2.3)

*** Includes an equivalent surface run (same initial coordinates as the AAF system) (see section 2.5)

**** Per flight altitude (see section 2.5)

2.1 ARMTRAJ-SFC – Surface Trajectory Data Set

The ARMTRAJ-SFC data set is designed to support research using ARM's surface measurements, with an emphasis on aerosol observations. For a given day at a given ARM site, ARMTRAJ-SFC is initialized at the surface in three-hour intervals, with each run including a 240-hour (10-day) back trajectory. The ARMTRAJ-SFC ensemble is initialized using two starting heights (surface and 50 m AGL), rendering 18 ensemble members. ARMTRAJ-SFC data files are supplemented with 1-hour mean and standard deviation values (starting at trajectory initialization time) of surface observations from the corresponding ARM site Surface Meteorology System (MET; Kyrouac and Tuftedal 2024).

2.2 ARMTRAJ-CLD –Liquid-Bearing Cloud Layer Trajectory Data Set

ARMTRAJ-CLD aims to augment liquid-bearing cloud studies from warm to mixed-phase clouds. The 120-hour (five-day) back and forward trajectories reported in this data set, among other uses, support

analyses of cloud formation mechanisms and cloudy airmass hysteresis and provide essential information for initializing and constraining modeling studies.

ARMTRAJ-CLD's initialization depends on ARM radiosonde measurements (SONDE; Holdridge 2020) for determining liquid-bearing cloud layer occurrence. Data set files are not generated for clear-sky periods. Therefore, this data set has the same starting times as the SONDE rounded to the nearest hour. The main advantage of radiosonde-based cloud detection is that full tropospheric profiles can be examined. Liquid-bearing cloud layers are determined from the SONDE relative humidity (RH) profiles using the following steps:

1. Set SONDE samples as "cloud" if RH values exceed 96%. This threshold value considers the Vaisala RS-41 radiosonde 4% RH uncertainty (see Holdridge 2020) and is consistent with previous comparisons to cloud layer detections based on other instruments (Silber et al. 2020, Figure S1; e.g., Silber and Shupe 2022, Stanford et al. 2023, Appendix D).
2. Concatenate "cloud" samples distant by less than 50 m from each other.
3. Remove resulting layers if their total thickness (including the thickness of the cloud-top sample) is smaller than 25 m.

ARMTRAJ-CLD reports trajectories for the center height of up to 10 detected overlying liquid-bearing layers per initialization time step. ARMTRAJ-CLD ensemble statistics are based on 27 members per detected cloud layer, with each ensemble consisting of three vertical starting heights (cloud layer center and center ± 50 m). The data set also provides the detected liquid-bearing cloud layer boundaries and the relevant SONDE thermodynamic and wind measurements to support cloud-related analysis further.

2.3 ARMTRAJ-PBL – Planetary Boundary-Layer Trajectory Data Set

ARMTRAJ-PBL, which could support PBL cloud and aerosol research in addition to other PBL research topics, includes 120-hour (five-day) back trajectory calculations for the base (surface), middle, and top of the PBL (i.e., ARM's Planetary Boundary Layer Height [PBLHT] VAP). ARMTRAJ-PBL also includes free-tropospheric runs for each trajectory starting time, initialized 200 m above the reported PBLH to support and augment studies focusing on free-tropospheric entrainment effects.

The PBLH data used for HYSPLIT initialization is taken from ARM's PBLHT VAP (PBLHSONDE1MCFARL datastream; Sivaraman et al. 2013). Specifically, the implemented PBLH is determined from SONDE measurements using a bulk Richardson number method (Troen and Mahrt 1986, Vogelezang and Holtslag 1996) with a critical threshold value of 0.25, the same method and threshold value implemented in the ERA5 PBLH diagnostics used for HYSPLIT calculations. Because the PBLHT data product depends on sounding data, ARMTRAJ-PBL trajectories are initialized at radiosonde release times, rounded to the nearest hour. Depending on SONDE availability, this results in two to four trajectory starting times per day.

The ensemble in the ARMTRAJ-PBL data set consists of 11 equally distant heights from the surface to the PBLH, totaling 99 ensemble members. This extensive ensemble configuration ameliorates the lack of explicit mixing in the ECMWF Integrated Forecasting System (IFS) model driving ERA5, the limited near-surface resolution (~250 m) at the ERA5 pressure level grid, and the lowest PBLH of 100 m AGL

set in HYSPLIT calculations. The free-tropospheric ensemble is initialized at a single height, resulting in nine ensemble members.

2.4 ARMTRAJ-ARSCL – Primary Cloud Deck Trajectory Data Set

Many studies often focus on primary cloud decks, referred to here as the optically and geometrically thickest cloud decks in atmospheric columns. Those primary cloud decks typically produce a significant radiative effect, impacting the surface and atmospheric energy budgets. ARMTRAJ-ARSCL's objective is to support studies focusing on those cloud decks while still providing analysis flexibility by running the trajectory calculations five days backward and forward in time. As suggested by the data set's name, data from ARM's Active Remote Sensing of Clouds VAP (ARSCL; Clothiaux et al. 2001) provide the primary cloud deck boundaries for the HYSPLIT model initialization.

The ARMTRAJ-ARSCL data set is initialized every three hours, similar to ARMTRAJ-SFC. (Clear-sky periods are excluded from the data set). The cloud deck base is determined as the 1-hour mean (starting at the initialization time stamps) cloud base height (using the "cloud_base_best_estimate" ARSCL field). The cloud deck top is set as the 1-hour mean first radar top (first radar echo with an overlying clear-sky range gate), the samples of which are included in the averaging only if, at a given time step, they are above the cloud deck base. Trajectory calculations are also made for the free troposphere to address community interest in processes such as cloud-top entrainment. Because the cloud top can be fairly variable over a 1-hour period, the free-tropospheric height is set as the sum of the one-hour mean cloud top, its 1-hour standard deviation (using the same samples as in the first radar-top averaging), and 200 m.

Similar to ARMTRAJ-PBL, ARMTRAJ-ARSCL reports trajectory data for the cloud deck base, middle, and top and includes 99-member ensemble results using 11 equally distant heights between the cloud deck base and top. (nine-member ensemble statistics are reported for the free troposphere). Hourly means of auxiliary data used and reported by ARSCL, such as hydrometeor field boundaries and liquid water path retrieved by ARMs microwave radiometer (Morris 2019), are included in ARMTRAJ-ARSCL data files.

2.5 ARMTRAJ-AAF – Trajectory Data Set for AAF Flights

The AAF operates a crewed aircraft, a tethered balloon system (TBS), and uncrewed aerial systems (UASs). Those ARM facilities (e.g., Dexheimer et al. 2024, Mather 2024, Mei et al. 2022, Schmid and Ivey 2016) obtain primarily in situ measurements, which offer unique insights into atmospheric structure and composition. Vertical and/or horizontal gradients or transitions in certain variables (e.g., aerosol concentration and composition) may indicate different airmass chemistry and source origins, among other properties. Therefore, such observations can benefit from back trajectory information, which suggests plausible hysteresis of airmass characteristics and potential source origins.

The ARMTRAJ-AAF data set supports research using AAF measurements by providing five-day back-trajectories. The data set is initialized at facility operation times in one-hour increments. The AAF UAS and aircraft coordinates for HYSPLIT initialization are extracted from the AAFNAVAIMS datastream, whereas the TBSIMET datastream provides the TBS coordinates. Trajectories are initialized at 11 equally distant altitudes AMSL between a facility's first and 99th altitude percentile over each one-hour period. Using altitude percentiles instead of a system's minimum and maximum altitude values

mitigates the potential influence of sparse, incorrectly reported altitudes (e.g., due to lack of Global Positioning System lock). The initial trajectory coordinates are set as the mean reported latitude and longitude over each one-hour period. To supplement studies contrasting surface and elevated airmass properties, ARMTRAJ-AAF also includes surface runs for each trajectory starting time, initialized at the corresponding coordinates at an altitude equal to the linearly interpolated ERA5 surface level.

Trajectory ensembles with 25 members are generated for each initialized altitude as well as for the surface run. To consider the moving nature of the AAF systems, the ensemble grid is set to be larger than in the other ARMTRAJ data sets with a 5x5 configuration. Unlike the other ARMTRAJ data sets, the distance between neighboring ensemble grid points is not fixed; it is set, for each one-hour period, as the larger value between a (latitude or longitude) coordinate standard deviation over that one-hour period (converted from arc degrees to kilometers) and 3.75 km (half the 7.5 km used in the other data sets). Note that this distance between grid points is calculated separately for the latitude and longitude dimensions and can, therefore, result in a rectangular rather than square grid shape, depending on the AAF system flight path. The neighboring grid point distance is limited to 50 km, rendering the effective ensemble dimensions between 15x15 km, as in the other ARMTRAJ data sets, and 100x100 km. This dynamic ensemble grid ensures applicability to all AAF systems, with the minimum ensemble dimensions being relevant to all TBS flights and most UAS flights and the larger ensemble grid dimensions being relevant to the vast majority of AAF aircraft flights. Hourly means, standard deviations, minima, and maxima of AAF system coordinates (latitude, longitude, and altitude), as well as the complete AAF system time series from the AAF datastreams are included in ARMTRAJ-AAF data files.

3.0 Input Data

The primary inputs for ARMTRAJ are ERA5 reanalysis data files converted to a dedicated Air Resources Laboratory (ARL) format extending to periods matching the trajectory periods (see Table 1), and ARM datastreams used for HYSPLIT initialization. These files are: MET for ARMTRAJ-SFC, SONDEWNPN for ARMTRAJ-CLD, PBLHSONDE1MCFARL for ARMTRAJ-PBL, ARSCL for ARMTRAJ-ARSCL, and AAFNAVAIMS and TBSIMET for ARMTRAJ-AAF.

4.0 Output Data

ARMTRAJ produces five daily output files; one for each of the current five data sets (ARMTRAJ-SFC, ARMTRAJ-CLD, ARMTRAJ-PBL, ARMTRAJ-ARSCL, and ARMTRAJ-AAF). The corresponding names of the output files are:

`SSSarmtrajsfcXX.c1.YYYYMMDD.hhmmss.nc`

`SSSarmtrajcldXX.c1.YYYYMMDD.hhmmss.nc`

`SSSarmtrajpblXX.c1.YYYYMMDD.hhmmss.nc`

`SSSarmtrajarsclXX.c1.YYYYMMDD.hhmmss.nc`

and

`SSSarmtrajaafXX.c1.YYYYMMDD.hhmmss.nc`

Where:

- SSS is the site
- XX is the facility
- YYYY is the year
- MM is the month
- DD is the day
- hh is the hour
- mm is the minute
- ss is the second

Each output file comprises numerous variables interpolated from ERA5 describing the airmass trajectory path, thermodynamic properties, and surface properties in the airmass column. Those variables include:

- airmass latitude (latitude), longitude (longitude), height AGL (height), and altitude AMSL (altitude)

- Airmass pressure (pres), temperature (temp), specific humidity (qv), relative humidity (rh), relative humidity with respect to ice (rhi), potential temperature (theta), equivalent potential temperature (theta_e), virtual potential temperature (theta_v), and mean hourly ascent rate (wvert)

- (In the airmass column) planetary boundary-layer height (PBLH), surface altitude (surf_altitude), and airmass height to PBLH ratio (height_to_pblh_ratio)

- (In the airmass column) land/sea cover (land_sea_mask), daily sea-ice cover (sea_ice_cover), low vegetation type and cover (low_vegetation_type, low_vegetation_cover), high vegetation type and cover (high_vegetation_type, high_vegetation_cover), and soil type (soil_type)

- (In the airmass column) Subgrid-scale orography angle (sgs_orography_angle), distortion (sgs_orography_anisotropy), standard deviation (sgs_orography_std), and slope (sgs_orography_slope).

Free-tropospheric air trajectory fields (in ARMTRAJ-PBL and ARMTRAJ-ARSCL) have a '_ft' suffix, and surface air trajectory fields (in ARMTRAJ-AAF) have a '_sfc' suffix. Ensemble fields have the '_ens' suffix followed by a corresponding statistic ('_mean', '_std', '_min', and '_max' for the mean, standard deviation, minimum, and maximum, respectively).

Output fields also include variables that could support the analysis of ARMTRAJ data, which are harvested from the associated ARM data sets (see Section 0).

A complete lists of output variables are given in the sample NetCDF headers in Appendix A, Appendix B, Appendix C, Appendix D, and Appendix E.

5.0 Example Plots

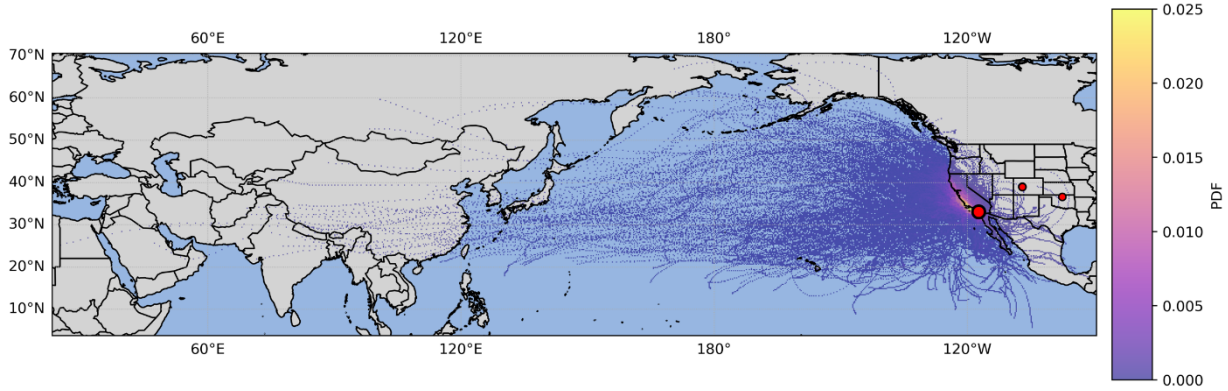

Figure 1. Probability density function (PDF) of 24-96 hour (one-four day) back-trajectory ensemble mean samples initialized in cloud decks detected over the Eastern Pacific Cloud Aerosol Precipitation Experiment (EPCAPE) deployment in La Jolla, California between March 1, 2023 and February 13, 2024. The large red marker denotes the EPCAPE main deployment site. PDFs (bin dimensions of 0.25×0.25 degrees) were generated using the ARMTRAJ-ARSCL data set.

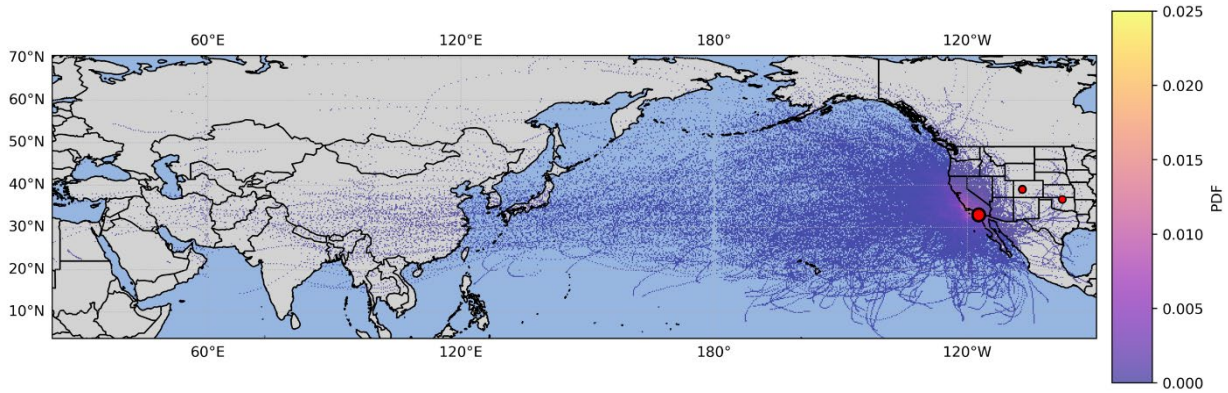

Figure 2. Same as Figure 1, but for free-tropospheric air just above the detected cloud decks. Note the more extensive spread relative to the PDF presented in Figure 1.

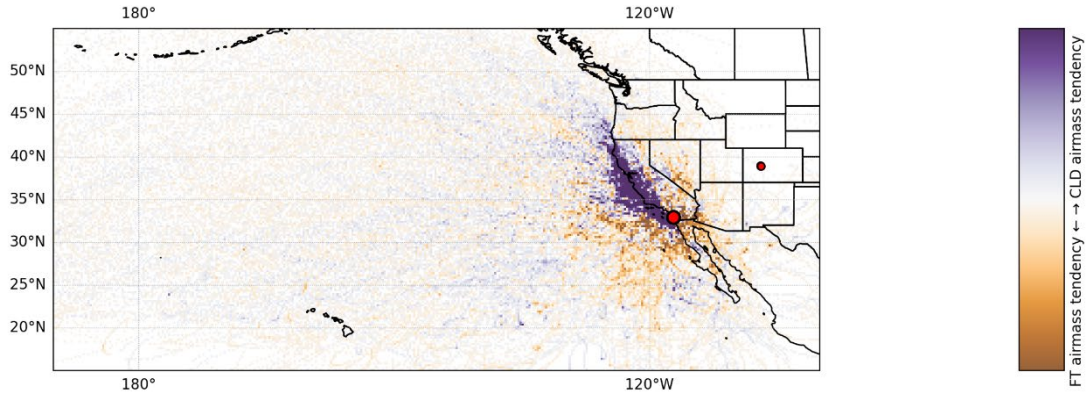

Figure 3. Cloudy and free-tropospheric airmass source origin tendencies, qualitatively demonstrated by the difference between the PDFs depicted above.

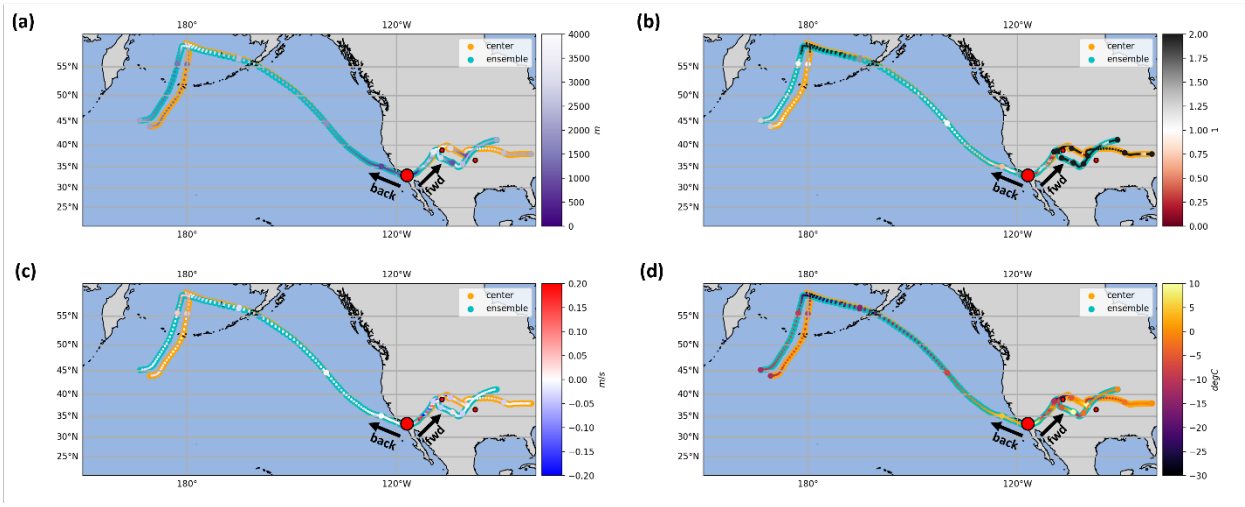

Figure 4. 120-hour (five-day) airmass back and forward trajectories and properties for the lowest liquid-bearing cloud layer (cloud base at 772 m AGL) detected over the EPCAPE main deployment site on March 1, 2023. The illustrated trajectories represent the site coordinate initialization and the ensemble mean (see legend). (a) Airmass altitude AMSL, (b) airmass height to PBLH ratio (values smaller than one suggest airmass within the PBL), (c) airmass one-hour mean vertical motion, and (d) airmass temperature. Larger markers along trajectories designate 24-hour intervals. The larger red marker denotes the EPCAPE main deployment site.

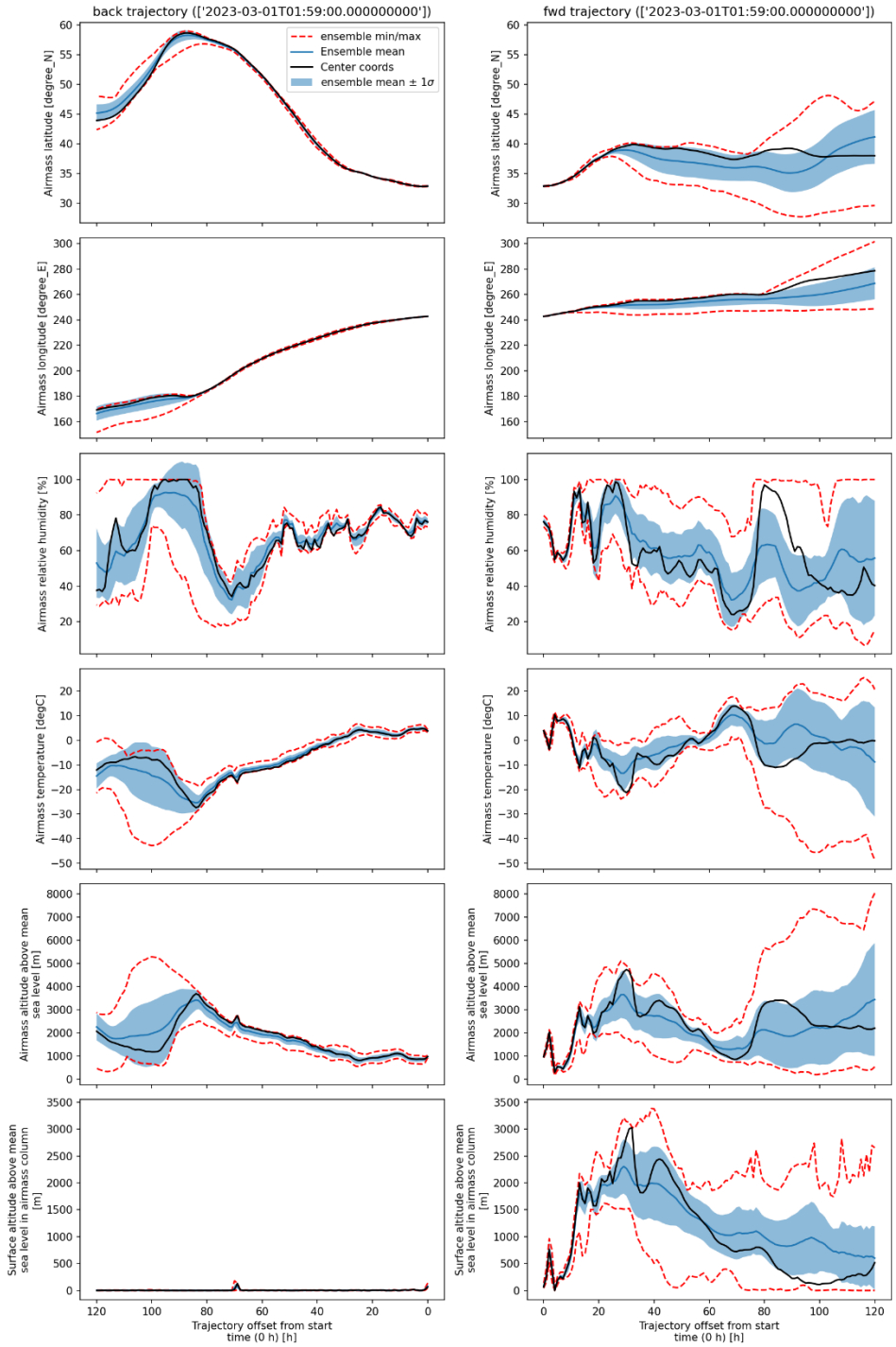

Figure 5. Time series of the 120-hour (five-day) airmass back (left column) and forward (right column) trajectory variables for the same event presented in Figure 4 (see legend for details). (From top to bottom) Airmass latitude, longitude, relative humidity, temperature, altitude, and surface altitude in airmass column.

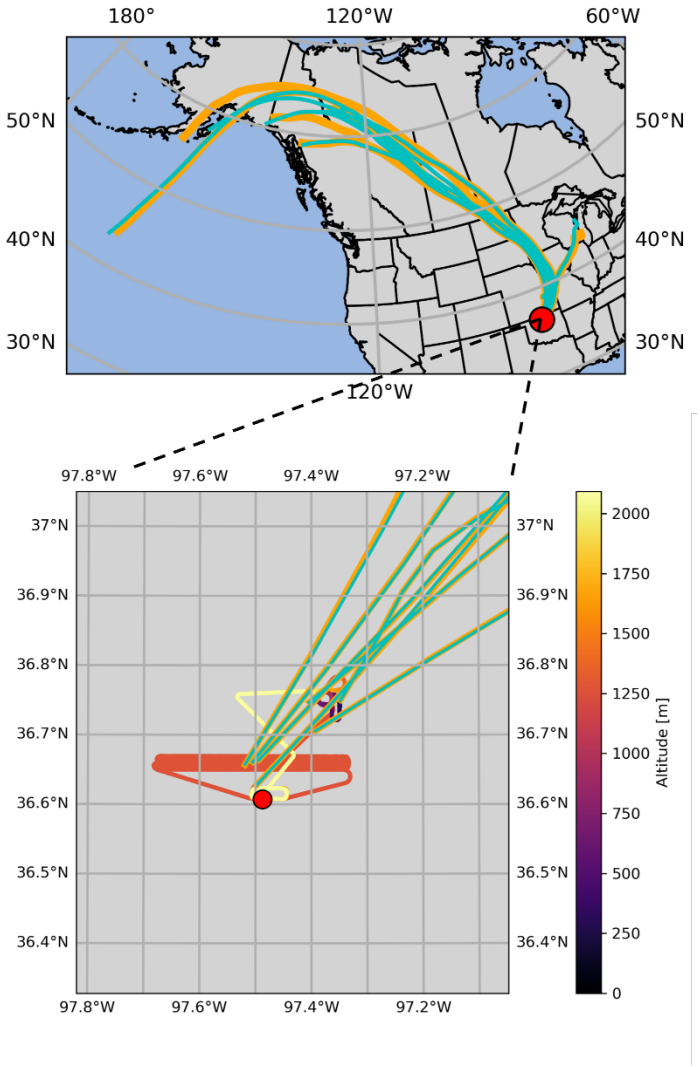

Figure 6. (Top) 120-hour (five-day) airmass back trajectories for the ArcticShark UAS ~seven-hour flight on August 30, 2023, over the ARM Southern Great Plains (SGP) central site (red marker). The orange curves designate the seven trajectories initialized between 14 and 20 UTC at the ArcticShark's one-hour mean latitude and longitude coordinates in the middle between the 1st and 99th UAS flight altitude percentiles during each one-hour window. The cyan curves illustrate the corresponding ensemble mean trajectories. (Bottom) Same, but zooming in on the ArcticShark flight path with the color scale illustrating the UAS flight altitude. Note the difference in trajectory initialization latitude and longitude, which have, in this case, some offset relative to the SGP site.

6.0 References

Ali, SM, and F Pithan. 2020. "Following moist intrusions into the Arctic using SHEBA observations in a Lagrangian perspective." *Quarterly Journal of the Royal Meteorological Society* 146(732): 3522–33, https://doi.org/10.1002/qj.3859

Clothiaux, EE, MA Miller, RC Perez, DD Turner, KP Moran, BE Martner, TP Ackerman, GG Mace, RT Marchand, KB Widener, DJ Rodriguez, T Uttal, JH Mather, CJ Flynn, KL Gaustad, and B Ermold. 2001. The ARM Millimeter Wave Cloud Radars (MMCRs) and the Active Remote Sensing of Clouds (ARSCL) Value Added Product (VAP). U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. ARM VAP-002.1. https://doi.org/10.2172/1808567

Dexheimer, D, G Whitson, Z Cheng, J Sammon, K Gaustad, F Mei, and C Longbottom. 2024. Tethered Balloon System (TBS) Instrument Handbook. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-TR-206. https://doi.org/10.2172/1415858

Hawkins, LN, and LM Russell. 2010. "Oxidation of Ketone Groups in Transported Biomass Burning Aerosol from the 2008 Northern California Lightning Series Fires." *Atmospheric Environment* 44(34): 4142–4154, https://doi.org/10.1016/j.atmosenv.2010.07.036

Hersbach, H, B Bell, P Berrisford, S Hirahara, A Horányi, J Muñoz-Sabater, J Nicolas, C Peubey, R Radu, D Schepers, A Simmons, C Soci, S Abdalla, X Abellan, G Balsamo, P Bechtold, G Biavati, J Bidlot, M Bonavita, G De Chiara, P Dahlgren, D Dee, M Diamantakis, R Dragani, J Flemming, R Forbes, M Fuentes, A Geer, L Haimberger, S Healy, RJ Hogan, E Hólm, M Janisková, S Keeley, P Laloyaux, P Lopez, C Lupu, G Radnoti, P de Rosnay, I Rozum, F Vamborg, S Villaume, and J-N Thépaut. 2020. "The ERA5 Global Reanalysis." *Quarterly Journal of the Royal Meteorological Society* 146(730): 1999–2049, https://doi.org/10.1002/qj.3803

Holdridge, D. 2020. Balloon-Borne Sounding System (SONDE) Instrument Handbook. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-TR-029.

Kyrouac, J, and M Tuftedal. 2011. Surface Meteorological System (MET) Instrument Handbook. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-TR-086. https://doi.org/10.2172/1007926

Liu, S, DA Day, JE Shields, and LM Russell. 2011. "Ozone-Driven Daytime Formation of Secondary Organic Aerosol Containing Carboxylic Acid Groups and Alkane Groups." *Atmospheric Chemistry and Physics* 11(16): 8321–8341, https://doi.org/10.5194/acp-11-8321-2011

Mather, J. 2024. Atmospheric Radiation Measurement (ARM) Management Plan. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-13-022. https://doi.org/10.2172/1253897

Mei, F, MS Pekour, D Dexheimer, G de Boer, R Cook, J Tomlinson, B Schmid, LA Goldberger, R Newsom, and JD Fast. 2022. "Observational Data from Uncrewed Systems over Southern Great Plains." *Earth System Science Data* 14(7): 3423–3438, https://doi.org/10.5194/essd-14-3423-2022

Mohrmann, J, CS Bretherton, IL McCoy, J McGibbon, R Wood, V Ghate, B Albrecht, M Sarkar, P Zuidema, and R Palikonda. 2019. "Lagrangian Evolution of the Northeast Pacific Marine Boundary Layer Structure and Cloud during CSET." *Monthly Weather Review* 147(12): 4681–4700, https://doi.org/10.1175/MWR-D-19-0053.1

Morris, VR. 2019. Microwave Radiometer (MWR) Handbook. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-TR-016.

Schmid, B, and M Ivey. 2016. ARM Unmanned Aerial Systems Implementation Plan. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-16-054.

Silber, I, JM Comstock, MR Kieburtz, and LM Russell. 2025. "ARMTRAJ: A Set of Multipurpose Trajectory Datasets Augmenting the Atmospheric Radiation Measurement (ARM) User Facility Measurements." *Earth System Science Data* 17(1): 29–42, https://doi.org/10.5194/essd-17-29-2025

Silber, I, AM Fridlind, J Verlinde, AS Ackerman, Y-S Chen, DH Bromwich, S-H Wang, M Cadeddu, and EW Eloranta. 2019. "Persistent Supercooled Drizzle at Temperatures below -25°C Observed at McMurdo Station, Antarctica." *Journal of Geophysical Research– Atmospheres* 124(20): 10878–10895, https://doi.org/10.1029/2019JD030882

Silber, I, AM Fridlind, J Verlinde, LM Russell, and AS Ackerman. 2020. "Nonturbulent Liquid-Bearing Polar Clouds: Observed Frequency of Occurrence and Simulated Sensitivity to Gravity Waves." *Geophysical Research Letters* 47(10): e2020GL087099, https://doi.org/10.1029/2020GL087099

Silber, I, and MD Shupe. 2022. "Insights on Sources and Formation Mechanisms of Liquid-Bearing Clouds over MOSAiC Examined from a Lagrangian Framework." *Elementa: Science of the Anthropocene* 10(1): 000071, https://doi.org/10.1525/elementa.2021.000071

Sivaraman, C, S McFarlane, E Chapman, M Jensen, T Toto (Fairless), S Liu, and M Fischer. 2013. Planetary Boundary Layer Height (PBL) Value Added Product (VAP): Radiosonde Retrievals. U.S. Department of Energy, Atmospheric Radiation Measurement user facility, Richland, Washington. DOE/SC-ARM-TR-132. https://doi.org/10.2172/1808688

Stanford, MW, AM Fridlind, I Silber, AS Ackerman, G Cesana, J Mülmenstädt, A Protat, S Alexander, and A McDonald. 2023. "Earth-System-Model Evaluation of Cloud and Precipitation Occurrence for Supercooled and Warm Clouds over the Southern Ocean's Macquarie Island." *Atmospheric Chemistry and Physics* 23(16): 9037–9069, https://doi.org/10.5194/acp-23-9037-2023

Stein, AF, RR Draxler, GD Rolph, BJB Stunder, MD Cohen, and F Ngan. 2015. "NOAA's HYSPLIT Atmospheric Transport and Dispersion Modeling System." *Bulletin of the American Meteorological Society* 96(12): 2059–2077, https://doi.org/10.1175/BAMS-D-14-00110.1

Svensson, G, S Murto, MD Shupe, F Pithan, L Magnusson, JJ Day, JD Doyle, IA Renfrew, T Spengler, and T Vihma. 2023. "Warm Air Intrusions Reaching the MOSAiC Expedition in April 2020—The YOPP Targeted Observing Period (TOP)." *Elementa: Science of the Anthropocene* 11(1): 00016, https://doi.org/10.1525/elementa.2023.00016

Tornow, F, AS Ackerman, AM Fridlind, B Cairns, EC Crosbie, S Kirschler, RH Moore, D Painemal, CE Robinson, C Seethala, MA Shook, C Voigt, EL Winstead, LD Ziemba, P Zuidema, and A Sorooshian. 2022. "Dilution of Boundary Layer Cloud Condensation Nucleus Concentrations by Free Tropospheric Entrainment during Marine Cold Air Outbreaks." *Geophysical Research Letters* 49(11): e2022GL098444, https://doi.org/https://doi.org/10.1029/2022GL098444

Troen, IB, and L Mahrt. 1986. "A Simple Model of the Atmospheric Boundary Layer; Sensitivity to Surface Evaporation." *Boundary-Layer Meteorology* 37(1): 129–148, https://doi.org/10.1007/BF00122760

Vogelezang, DHP, and AAM Holtslag. 1996. "Evaluation and Model Impacts of Alternative Boundary-Layer Height Formulations." *Boundary-Layer Meteorology* 81(3): 245–269, https://doi.org/10.1007/BF02430331

Zheng, G, AJ Sedlacek, AC Aiken, Y Feng, TB Watson, S Raveh-Rubin, J Uin, ER Lewis, and J Wang. 2020. "Long-Range Transported North American Wildfire Aerosols Observed in Marine Boundary Layer of Eastern North Atlantic." *Environment International* 139: 105680, https://doi.org/10.1016/j.envint.2020.105680

Appendix A

ARMTRAJ-SFC Output Data

```
netcdf epcarmtrajsfcM1.c1.20230820.000000 {
dimensions:
        time = UNLIMITED ; // (8 currently)
        trajectory_time = 241 ;
        trajectory_type = 1 ;
variables:
        int trajectory_time(trajectory_time) ;
                trajectory_time:long_name = "Trajectory offset from start time (0 h)" ;
                trajectory_time:units = "h" ;
        int trajectory_type(trajectory_type) ;
                trajectory_type:long_name = "-1 - back trajectories, 1 - fwd trajectories" ;
                trajectory_type:units = "1" ;
        double date(time, trajectory_time) ;
                date:_FillValue = -9999. ;
                date:long_name = "Trajectory date per trajectory type and time step" ;
                date:units = "seconds since 1970-01-01" ;
        double latitude(time, trajectory_time) ;
                latitude:_FillValue = -9999. ;
                latitude:long_name = "Airmass latitude" ;
                latitude:units = "degree_N" ;
                latitude:standard_name = "grid_latitude" ;
        double longitude(time, trajectory_time) ;
                longitude:_FillValue = -9999. ;
                longitude:long_name = "Airmass longitude" ;
                longitude:units = "degree_E" ;
                longitude:standard_name = "grid_longitude" ;
        double height(time, trajectory_time) ;
                height:_FillValue = -9999. ;
                height:long_name = "Airmass height above ground level" ;
                height:units = "m" ;
                height:standard_name = "height" ;
        double pres(time, trajectory_time) ;
                pres:_FillValue = -9999. ;
                pres:long_name = "Airmass pressure" ;
```

pres:units = "hPa" ;
pres:standard_name = "air_pressure" ;
double theta(time, trajectory_time) ;
theta:_FillValue = -9999. ;
theta:long_name = "Airmass potential temperature" ;
theta:units = "K" ;
theta:standard_name = "air_potential_temperature" ;
double temp(time, trajectory_time) ;
temp:_FillValue = -9999. ;
temp:long_name = "Airmass temperature" ;
temp:units = "degC" ;
temp:standard_name = "air_temperature" ;
double pblh(time, trajectory_time) ;
pblh:_FillValue = -9999. ;
pblh:long_name = "Planetary boundary layer height in airmass column" ;
pblh:units = "m" ;
pblh:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh(time, trajectory_time) ;
rh:_FillValue = -9999. ;
rh:long_name = "Airmass relative humidity" ;
rh:units = "%" ;
rh:standard_name = "relative_humidity" ;
double surf_altitude(time, trajectory_time) ;
surf_altitude:_FillValue = -9999. ;
surf_altitude:long_name = "Surface altitude above mean sea level in airmass column" ;
surf_altitude:units = "m" ;
surf_altitude:standard_name = "surface_altitude" ;
double qv(time, trajectory_time) ;
qv:_FillValue = -9999. ;
qv:long_name = "Airmass specific humidity" ;
qv:units = "g/kg" ;
qv:standard_name = "specific_humidity" ;
double rhi(time, trajectory_time) ;
rhi:_FillValue = -9999. ;
rhi:long_name = "Airmass relative humidity with respect to ice" ;
rhi:units = "%" ;
double theta_e(time, trajectory_time) ;
theta_e:_FillValue = -9999. ;
theta_e:long_name = "Airmass equivalent potential temperature (excluding condensate from the calculation)" ;
theta_e:units = "K" ;
theta_e:standard_name = "air_equivalent_potential_temperature" ;
double theta_v(time, trajectory_time) ;
theta_v:_FillValue = -9999. ;
theta_v:long_name = "Airmass virtual potential temperature" ;
theta_v:units = "K" ;

```
        double altitude(time, trajectory_time) ;
                altitude:_FillValue = -9999. ;
                altitude:long_name = "Airmass altitude above mean sea level" ;
                altitude:units = "m" ;
                altitude:standard_name = "height_above_mean_sea_level" ;
        double wvert(time, trajectory_time) ;
                wvert:_FillValue = -9999. ;
                wvert:long_name = "Mean hourly airmass ascent rate" ;
                wvert:units = "m/s" ;
                wvert:standard_name = "upward_air_velocity" ;
        double height_to_pblh_ratio(time, trajectory_time) ;
                height_to_pblh_ratio:_FillValue = -9999. ;
                height_to_pblh_ratio:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above
PBL; <1 - airmass within PBL)" ;
                height_to_pblh_ratio:units = "1" ;
        double latitude_ens_mean(time, trajectory_time) ;
                latitude_ens_mean:_FillValue = -9999. ;
        double latitude_ens_std(time, trajectory_time) ;
                latitude_ens_std:_FillValue = -9999. ;
        double latitude_ens_min(time, trajectory_time) ;
                latitude_ens_min:_FillValue = -9999. ;
        double latitude_ens_max(time, trajectory_time) ;
                latitude_ens_max:_FillValue = -9999. ;
        double longitude_ens_mean(time, trajectory_time) ;
                longitude_ens_mean:_FillValue = -9999. ;
        double longitude_ens_std(time, trajectory_time) ;
                longitude_ens_std:_FillValue = -9999. ;
        double longitude_ens_min(time, trajectory_time) ;
                longitude_ens_min:_FillValue = -9999. ;
        double longitude_ens_max(time, trajectory_time) ;
                longitude_ens_max:_FillValue = -9999. ;
        double height_ens_mean(time, trajectory_time) ;
                height_ens_mean:_FillValue = -9999. ;
        double height_ens_std(time, trajectory_time) ;
                height_ens_std:_FillValue = -9999. ;
        double height_ens_min(time, trajectory_time) ;
                height_ens_min:_FillValue = -9999. ;
        double height_ens_max(time, trajectory_time) ;
                height_ens_max:_FillValue = -9999. ;
        double pres_ens_mean(time, trajectory_time) ;
                pres_ens_mean:_FillValue = -9999. ;
        double pres_ens_std(time, trajectory_time) ;
                pres_ens_std:_FillValue = -9999. ;
        double pres_ens_min(time, trajectory_time) ;
                pres_ens_min:_FillValue = -9999. ;
        double pres_ens_max(time, trajectory_time) ;
```

pres_ens_max:_FillValue = -9999. ;
double theta_ens_mean(time, trajectory_time) ;
theta_ens_mean:_FillValue = -9999. ;
double theta_ens_std(time, trajectory_time) ;
theta_ens_std:_FillValue = -9999. ;
double theta_ens_min(time, trajectory_time) ;
theta_ens_min:_FillValue = -9999. ;
double theta_ens_max(time, trajectory_time) ;
theta_ens_max:_FillValue = -9999. ;
double temp_ens_mean(time, trajectory_time) ;
temp_ens_mean:_FillValue = -9999. ;
double temp_ens_std(time, trajectory_time) ;
temp_ens_std:_FillValue = -9999. ;
double temp_ens_min(time, trajectory_time) ;
temp_ens_min:_FillValue = -9999. ;
double temp_ens_max(time, trajectory_time) ;
temp_ens_max:_FillValue = -9999. ;
double pblh_ens_mean(time, trajectory_time) ;
pblh_ens_mean:_FillValue = -9999. ;
double pblh_ens_std(time, trajectory_time) ;
pblh_ens_std:_FillValue = -9999. ;
double pblh_ens_min(time, trajectory_time) ;
pblh_ens_min:_FillValue = -9999. ;
double pblh_ens_max(time, trajectory_time) ;
pblh_ens_max:_FillValue = -9999. ;
double rh_ens_mean(time, trajectory_time) ;
rh_ens_mean:_FillValue = -9999. ;
double rh_ens_std(time, trajectory_time) ;
rh_ens_std:_FillValue = -9999. ;
double rh_ens_min(time, trajectory_time) ;
rh_ens_min:_FillValue = -9999. ;
double rh_ens_max(time, trajectory_time) ;
rh_ens_max:_FillValue = -9999. ;
double surf_altitude_ens_mean(time, trajectory_time) ;
surf_altitude_ens_mean:_FillValue = -9999. ;
double surf_altitude_ens_std(time, trajectory_time) ;
surf_altitude_ens_std:_FillValue = -9999. ;
double surf_altitude_ens_min(time, trajectory_time) ;
surf_altitude_ens_min:_FillValue = -9999. ;
double surf_altitude_ens_max(time, trajectory_time) ;
surf_altitude_ens_max:_FillValue = -9999. ;
double qv_ens_mean(time, trajectory_time) ;
qv_ens_mean:_FillValue = -9999. ;
double qv_ens_std(time, trajectory_time) ;
qv_ens_std:_FillValue = -9999. ;
double qv_ens_min(time, trajectory_time) ;

qv_ens_min:_FillValue = -9999. ;
double qv_ens_max(time, trajectory_time) ;
qv_ens_max:_FillValue = -9999. ;
double rhi_ens_mean(time, trajectory_time) ;
rhi_ens_mean:_FillValue = -9999. ;
double rhi_ens_std(time, trajectory_time) ;
rhi_ens_std:_FillValue = -9999. ;
double rhi_ens_min(time, trajectory_time) ;
rhi_ens_min:_FillValue = -9999. ;
double rhi_ens_max(time, trajectory_time) ;
rhi_ens_max:_FillValue = -9999. ;
double theta_e_ens_mean(time, trajectory_time) ;
theta_e_ens_mean:_FillValue = -9999. ;
double theta_e_ens_std(time, trajectory_time) ;
theta_e_ens_std:_FillValue = -9999. ;
double theta_e_ens_min(time, trajectory_time) ;
theta_e_ens_min:_FillValue = -9999. ;
double theta_e_ens_max(time, trajectory_time) ;
theta_e_ens_max:_FillValue = -9999. ;
double theta_v_ens_mean(time, trajectory_time) ;
theta_v_ens_mean:_FillValue = -9999. ;
double theta_v_ens_std(time, trajectory_time) ;
theta_v_ens_std:_FillValue = -9999. ;
double theta_v_ens_min(time, trajectory_time) ;
theta_v_ens_min:_FillValue = -9999. ;
double theta_v_ens_max(time, trajectory_time) ;
theta_v_ens_max:_FillValue = -9999. ;
double altitude_ens_mean(time, trajectory_time) ;
altitude_ens_mean:_FillValue = -9999. ;
double altitude_ens_std(time, trajectory_time) ;
altitude_ens_std:_FillValue = -9999. ;
double altitude_ens_min(time, trajectory_time) ;
altitude_ens_min:_FillValue = -9999. ;
double altitude_ens_max(time, trajectory_time) ;
altitude_ens_max:_FillValue = -9999. ;
double wvert_ens_mean(time, trajectory_time) ;
wvert_ens_mean:_FillValue = -9999. ;
double wvert_ens_std(time, trajectory_time) ;
wvert_ens_std:_FillValue = -9999. ;
double wvert_ens_min(time, trajectory_time) ;
wvert_ens_min:_FillValue = -9999. ;
double wvert_ens_max(time, trajectory_time) ;
wvert_ens_max:_FillValue = -9999. ;
double height_to_pblh_ratio_ens_mean(time, trajectory_time) ;
height_to_pblh_ratio_ens_mean:_FillValue = -9999. ;
double height_to_pblh_ratio_ens_std(time, trajectory_time) ;

height_to_pblh_ratio_ens_std:_FillValue = -9999. ;
double height_to_pblh_ratio_ens_min(time, trajectory_time) ;
height_to_pblh_ratio_ens_min:_FillValue = -9999. ;
double height_to_pblh_ratio_ens_max(time, trajectory_time) ;
height_to_pblh_ratio_ens_max:_FillValue = -9999. ;
double sgs_orography_angle(time, trajectory_time) ;
sgs_orography_angle:_FillValue = -9999. ;
sgs_orography_angle:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_angle:units = "radian" ;
double sgs_orography_angle_ens_mean(time, trajectory_time) ;
sgs_orography_angle_ens_mean:_FillValue = -9999. ;
sgs_orography_angle_ens_mean:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_angle_ens_mean:units = "radian" ;
double sgs_orography_anisotropy(time, trajectory_time) ;
sgs_orography_anisotropy:_FillValue = -9999. ;
sgs_orography_anisotropy:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_anisotropy:units = "1" ;
double sgs_orography_anisotropy_ens_mean(time, trajectory_time) ;
sgs_orography_anisotropy_ens_mean:_FillValue = -9999. ;
sgs_orography_anisotropy_ens_mean:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_anisotropy_ens_mean:units = "1" ;
double sgs_orography_std(time, trajectory_time) ;
sgs_orography_std:_FillValue = -9999. ;
sgs_orography_std:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_std:units = "m" ;
double sgs_orography_std_ens_mean(time, trajectory_time) ;
sgs_orography_std_ens_mean:_FillValue = -9999. ;
sgs_orography_std_ens_mean:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_std_ens_mean:units = "m" ;
double sgs_orography_slope(time, trajectory_time) ;
sgs_orography_slope:_FillValue = -9999. ;
sgs_orography_slope:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_slope:units = "1" ;

double sgs_orography_slope_ens_mean(time, trajectory_time) ;
sgs_orography_slope_ens_mean:_FillValue = -9999. ;
sgs_orography_slope_ens_mean:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_slope_ens_mean:units = "1" ;
double land_sea_mask(time, trajectory_time) ;
land_sea_mask:_FillValue = -9999. ;
land_sea_mask:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land)" ;
land_sea_mask:units = "1" ;
double land_sea_mask_ens_mean(time, trajectory_time) ;
land_sea_mask_ens_mean:_FillValue = -9999. ;
land_sea_mask_ens_mean:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along ensemble mean trajectory)" ;
land_sea_mask_ens_mean:units = "1" ;
float high_vegetation_type(time, trajectory_time) ;
high_vegetation_type:_FillValue = -9999.f ;
high_vegetation_type:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation)" ;
high_vegetation_type:units = "1" ;
float high_vegetation_type_ens_mean(time, trajectory_time) ;
high_vegetation_type_ens_mean:_FillValue = -9999.f ;
high_vegetation_type_ens_mean:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along ensemble mean trajectory)" ;
high_vegetation_type_ens_mean:units = "1" ;
float low_vegetation_type(time, trajectory_time) ;
low_vegetation_type:_FillValue = -9999.f ;
low_vegetation_type:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation)" ;
low_vegetation_type:units = "1" ;
float low_vegetation_type_ens_mean(time, trajectory_time) ;
low_vegetation_type_ens_mean:_FillValue = -9999.f ;
low_vegetation_type_ens_mean:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along ensemble mean trajectory)" ;
low_vegetation_type_ens_mean:units = "1" ;
double high_vegetation_cover(time, trajectory_time) ;
high_vegetation_cover:_FillValue = -9999. ;

high_vegetation_cover:long_name = "High vegetation cover fraction in airmass column" ;
high_vegetation_cover:units = "1" ;
double high_vegetation_cover_ens_mean(time, trajectory_time) ;
high_vegetation_cover_ens_mean:_FillValue = -9999. ;
high_vegetation_cover_ens_mean:long_name = "High vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
high_vegetation_cover_ens_mean:units = "1" ;
double low_vegetation_cover(time, trajectory_time) ;
low_vegetation_cover:_FillValue = -9999. ;
low_vegetation_cover:long_name = "Low vegetation cover fraction in airmass column" ;
low_vegetation_cover:units = "1" ;
double low_vegetation_cover_ens_mean(time, trajectory_time) ;
low_vegetation_cover_ens_mean:_FillValue = -9999. ;
low_vegetation_cover_ens_mean:long_name = "Low vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
low_vegetation_cover_ens_mean:units = "1" ;
float soil_type(time, trajectory_time) ;
soil_type:_FillValue = -9999.f ;
soil_type:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point)" ;
soil_type:units = "1" ;
soil_type:standard_name = "soil_type" ;
float soil_type_ens_mean(time, trajectory_time) ;
soil_type_ens_mean:_FillValue = -9999.f ;
soil_type_ens_mean:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along ensemble mean trajectory)" ;
soil_type_ens_mean:units = "1" ;
soil_type_ens_mean:standard_name = "soil_type" ;
double sea_ice_cover(time, trajectory_time) ;
sea_ice_cover:_FillValue = -9999. ;
sea_ice_cover:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column" ;
sea_ice_cover:units = "1" ;
sea_ice_cover:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ens_mean(time, trajectory_time) ;
sea_ice_cover_ens_mean:_FillValue = -9999. ;
sea_ice_cover_ens_mean:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along ensemble mean trajectory)" ;
sea_ice_cover_ens_mean:units = "1" ;
sea_ice_cover_ens_mean:standard_name = "sea_ice_area_fraction" ;
float temp_1h_mean_met(time) ;
temp_1h_mean_met:_FillValue = -9999.f ;
temp_1h_mean_met:long_name = "Temperature mean, 1-hour mean" ;

temp_1h_mean_met:units = "degC" ;
temp_1h_mean_met:valid_min = -40.f ;
temp_1h_mean_met:valid_max = 50.f ;
temp_1h_mean_met:valid_delta = 20.f ;
temp_1h_mean_met:standard_name = "air_temperature" ;
float temp_1h_std_met(time) ;
temp_1h_std_met:_FillValue = -9999.f ;
temp_1h_std_met:long_name = "Temperature mean, 1-hour std" ;
temp_1h_std_met:units = "degC" ;
temp_1h_std_met:valid_min = -40.f ;
temp_1h_std_met:valid_max = 50.f ;
temp_1h_std_met:valid_delta = 20.f ;
temp_1h_std_met:standard_name = "air_temperature" ;
float rh_1h_mean_met(time) ;
rh_1h_mean_met:_FillValue = -9999.f ;
rh_1h_mean_met:long_name = "Relative humidity mean, 1-hour mean" ;
rh_1h_mean_met:units = "%" ;
rh_1h_mean_met:valid_min = -2.f ;
rh_1h_mean_met:valid_max = 104.f ;
rh_1h_mean_met:valid_delta = 30.f ;
rh_1h_mean_met:standard_name = "relative_humidity" ;
float rh_1h_std_met(time) ;
rh_1h_std_met:_FillValue = -9999.f ;
rh_1h_std_met:long_name = "Relative humidity mean, 1-hour std" ;
rh_1h_std_met:units = "%" ;
rh_1h_std_met:valid_min = -2.f ;
rh_1h_std_met:valid_max = 104.f ;
rh_1h_std_met:valid_delta = 30.f ;
rh_1h_std_met:standard_name = "relative_humidity" ;
float pres_1h_mean_met(time) ;
pres_1h_mean_met:_FillValue = -9999.f ;
pres_1h_mean_met:long_name = "Atmospheric pressure, 1-hour mean" ;
pres_1h_mean_met:units = "kPa" ;
pres_1h_mean_met:valid_min = 80.f ;
pres_1h_mean_met:valid_max = 110.f ;
pres_1h_mean_met:valid_delta = 1.f ;
pres_1h_mean_met:standard_name = "surface_air_pressure" ;
float pres_1h_std_met(time) ;
pres_1h_std_met:_FillValue = -9999.f ;
pres_1h_std_met:long_name = "Atmospheric pressure, 1-hour std" ;
pres_1h_std_met:units = "kPa" ;
pres_1h_std_met:valid_min = 80.f ;
pres_1h_std_met:valid_max = 110.f ;
pres_1h_std_met:valid_delta = 1.f ;
pres_1h_std_met:standard_name = "surface_air_pressure" ;
float tbrg_precip_1h_total(time) ;

tbrg_precip_1h_total:_FillValue = -9999.f ;
tbrg_precip_1h_total:long_name = "TBRG precipitation total, corrected, 1-hour total" ;
tbrg_precip_1h_total:units = "mm" ;
tbrg_precip_1h_total:valid_min = 0.f ;
tbrg_precip_1h_total:valid_max = 10.f ;
double org_precip_1h_total(time) ;
org_precip_1h_total:_FillValue = -9999. ;
org_precip_1h_total:long_name = "ORG precipitation 1-hour total" ;
org_precip_1h_total:units = "mm" ;
org_precip_1h_total:valid_min = 0.f ;
org_precip_1h_total:valid_max = 500.f ;
float wspd_1h_mean_met(time) ;
wspd_1h_mean_met:_FillValue = -9999.f ;
wspd_1h_mean_met:long_name = "Wind speed arithmetic mean, 1-hour mean" ;
wspd_1h_mean_met:units = "m/s" ;
wspd_1h_mean_met:valid_min = 0.f ;
wspd_1h_mean_met:valid_max = 60.f ;
wspd_1h_mean_met:valid_delta = 20.f ;
float wspd_1h_std_met(time) ;
wspd_1h_std_met:_FillValue = -9999.f ;
wspd_1h_std_met:long_name = "Wind speed arithmetic mean, 1-hour std" ;
wspd_1h_std_met:units = "m/s" ;
wspd_1h_std_met:valid_min = 0.f ;
wspd_1h_std_met:valid_max = 60.f ;
wspd_1h_std_met:valid_delta = 20.f ;
double wdir_1h_mean_met(time) ;
wdir_1h_mean_met:_FillValue = -9999. ;
wdir_1h_mean_met:long_name = "Wind direction vector mean, 1-hour mean" ;
wdir_1h_mean_met:units = "degree" ;
wdir_1h_mean_met:valid_min = 0.f ;
wdir_1h_mean_met:valid_max = 360.f ;
wdir_1h_mean_met:standard_name = "wind_from_direction" ;
double wdir_1h_std_met(time) ;
wdir_1h_std_met:_FillValue = -9999. ;
wdir_1h_std_met:long_name = "Wind direction vector mean, 1-hour std" ;
wdir_1h_std_met:units = "degree" ;
wdir_1h_std_met:valid_min = 0.f ;
wdir_1h_std_met:valid_max = 360.f ;
wdir_1h_std_met:standard_name = "wind_from_direction" ;
double time(time) ;
time:long_name = "Time in seconds since volume start" ;
time:units = "seconds since 2023-08-20 00:00:00 0:00" ;
double time_offset(time) ;
time_offset:_FillValue = -9999. ;
time_offset:long_name = "Time offset from base_time" ;
time_offset:units = "seconds since 2023-08-20 00:00:00 0:00" ;

```
        int base_time ;
                base_time:_FillValue = -9999 ;
                base_time:long_name = "Base time in Epoch" ;
                base_time:units = "seconds since 1970-01-01 0:00:00 0:00" ;
        double lat ;
                lat:_FillValue = -9999. ;
                lat:long_name = "latitude" ;
                lat:units = "degree_N" ;
        double lon ;
                lon:_FillValue = -9999. ;
                lon:long_name = "longitude" ;
                lon:units = "degree_E" ;
        double alt ;
                alt:_FillValue = -9999. ;
                alt:long_name = "Altitude above mean sea level" ;
                alt:units = "m" ;

// global attributes:
                :mode = "sfc" ;
                :ensemble_size = "18" ;
                :ensemble_dx_from_center = "[-7.5  0.   7.5] km" ;
                :ensemble_dy_from_center = "[-7.5  0.   7.5] km" ;
                :ensemble_dz_from_center = "[ 0 50] m" ;
                :used_era5_resolution = "0.25 degrees" ;
                :total_trajectory_hours = "[-240]" ;
                :number_of_starting_center_points_per_time_step = "[1 1 1 1 1 1 1 1]" ;
                :title = "Back trajectories for ARM surface deployments supporting aerosol studies" ;
                :summary = "The ARMTRAJ VAP provides trajectory datasets initialized at ARM
deployment coordinates and configured using ARM datasets. The trajectory datasets support aerosol,
cloud, and planetary boundary layer research. Trajectory calculations use the HYSPLIT model informed
by the ERA5 reanalysis dataset at its highest spatial resolution (~31 km). For each sample in each of the
datasets, HYSPLIT also runs at multiple starting locations surrounding ARM deployments, enabling an
ensemble of runs from which the mean and variability (estimated uncertainty) of each sample\'s trajectory
coordinates, thermodynamic properties, or other fields are reported." ;
                :keywords = "ARM, ground-based measurements, ground-based remote sensing, clouds,
aerosol, PBL, trajectories, Lagrangian analysis" ;
                :doi = "10.5439/2309850" ;
                :location_description = "Eastern Pacific Cloud Aerosol Precipitation Experiment
(EPCAPE), Scripps Pier, La Jolla, CA" ;
                :dod_version = "v1.0" ;
                :command_line = "python armtraj_run.py" ;
                :datastream = "epcarmtrajsfcM1.c1" ;
                :data_level = "c1" ;
                :facility_id = "M1" ;
                :site_id = "epc" ;
                :platform_id = "armtrajsfc" ;
```

:history = "created by user isilber1 on machine dev-proc2 at 27-Sep-2024,15:26:15" ;
}

Appendix B

ARMTRAJ-CLD Output Data

```
netcdf epcarmtrajcldM1.c1.20230820.000000 {
dimensions:
        time = UNLIMITED ; // (4 currently)
        vert_layer = 10 ;
        trajectory_time = 121 ;
        trajectory_type = 2 ;
        time_dim_sonde = 10000 ;
variables:
        int vert_layer(vert_layer) ;
                vert_layer:long_name = "Vertical layer index (bottom-up)" ;
                vert_layer:units = "1" ;
        int trajectory_time(trajectory_time) ;
                trajectory_time:long_name = "Trajectory offset from start time (0 h)" ;
                trajectory_time:units = "h" ;
        int trajectory_type(trajectory_type) ;
                trajectory_type:long_name = "-1 - back trajectories, 1 - fwd trajectories" ;
                trajectory_type:units = "1" ;
        double date(time, trajectory_type, trajectory_time) ;
                date:_FillValue = -9999. ;
                date:long_name = "Trajectory date per trajectory type and time step" ;
                date:units = "seconds since 1970-01-01" ;
        double latitude(time, trajectory_type, trajectory_time, vert_layer) ;
                latitude:_FillValue = -9999. ;
                latitude:long_name = "Airmass latitude" ;
                latitude:units = "degree_N" ;
                latitude:standard_name = "grid_latitude" ;
        double longitude(time, trajectory_type, trajectory_time, vert_layer) ;
                longitude:_FillValue = -9999. ;
                longitude:long_name = "Airmass longitude" ;
                longitude:units = "degree_E" ;
                longitude:standard_name = "grid_longitude" ;
        double height(time, trajectory_type, trajectory_time, vert_layer) ;
                height:_FillValue = -9999. ;
                height:long_name = "Airmass height above ground level" ;
```

height:units = "m" ;
height:standard_name = "height" ;
double pres(time, trajectory_type, trajectory_time, vert_layer) ;
pres:_FillValue = -9999. ;
pres:long_name = "Airmass pressure" ;
pres:units = "hPa" ;
pres:standard_name = "air_pressure" ;
double theta(time, trajectory_type, trajectory_time, vert_layer) ;
theta:_FillValue = -9999. ;
theta:long_name = "Airmass potential temperature" ;
theta:units = "K" ;
theta:standard_name = "air_potential_temperature" ;
double temp(time, trajectory_type, trajectory_time, vert_layer) ;
temp:_FillValue = -9999. ;
temp:long_name = "Airmass temperature" ;
temp:units = "degC" ;
temp:standard_name = "air_temperature" ;
double pblh(time, trajectory_type, trajectory_time, vert_layer) ;
pblh:_FillValue = -9999. ;
pblh:long_name = "Planetary boundary layer height in airmass column" ;
pblh:units = "m" ;
pblh:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh(time, trajectory_type, trajectory_time, vert_layer) ;
rh:_FillValue = -9999. ;
rh:long_name = "Airmass relative humidity" ;
rh:units = "%" ;
rh:standard_name = "relative_humidity" ;
double surf_altitude(time, trajectory_type, trajectory_time, vert_layer) ;
surf_altitude:_FillValue = -9999. ;
surf_altitude:long_name = "Surface altitude above mean sea level in airmass column" ;
surf_altitude:units = "m" ;
surf_altitude:standard_name = "surface_altitude" ;
double qv(time, trajectory_type, trajectory_time, vert_layer) ;
qv:_FillValue = -9999. ;
qv:long_name = "Airmass specific humidity" ;
qv:units = "g/kg" ;
qv:standard_name = "specific_humidity" ;
double rhi(time, trajectory_type, trajectory_time, vert_layer) ;
rhi:_FillValue = -9999. ;
rhi:long_name = "Airmass relative humidity with respect to ice" ;
rhi:units = "%" ;
double theta_e(time, trajectory_type, trajectory_time, vert_layer) ;
theta_e:_FillValue = -9999. ;
theta_e:long_name = "Airmass equivalent potential temperature (excluding condensate from the calculation)" ;
theta_e:units = "K" ;

```
        theta_e:standard_name = "air_equivalent_potential_temperature" ;
    double theta_v(time, trajectory_type, trajectory_time, vert_layer) ;
        theta_v:_FillValue = -9999. ;
        theta_v:long_name = "Airmass virtual potential temperature" ;
        theta_v:units = "K" ;
    double altitude(time, trajectory_type, trajectory_time, vert_layer) ;
        altitude:_FillValue = -9999. ;
        altitude:long_name = "Airmass altitude above mean sea level" ;
        altitude:units = "m" ;
        altitude:standard_name = "height_above_mean_sea_level" ;
    double wvert(time, trajectory_type, trajectory_time, vert_layer) ;
        wvert:_FillValue = -9999. ;
        wvert:long_name = "Mean hourly airmass ascent rate" ;
        wvert:units = "m/s" ;
        wvert:standard_name = "upward_air_velocity" ;
    double height_to_pblh_ratio(time, trajectory_type, trajectory_time, vert_layer) ;
        height_to_pblh_ratio:_FillValue = -9999. ;
        height_to_pblh_ratio:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above
PBL; <1 - airmass within PBL)" ;
        height_to_pblh_ratio:units = "1" ;
    double latitude_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
        latitude_ens_mean:_FillValue = -9999. ;
    double latitude_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
        latitude_ens_std:_FillValue = -9999. ;
    double latitude_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
        latitude_ens_min:_FillValue = -9999. ;
    double latitude_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
        latitude_ens_max:_FillValue = -9999. ;
    double longitude_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
        longitude_ens_mean:_FillValue = -9999. ;
    double longitude_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
        longitude_ens_std:_FillValue = -9999. ;
    double longitude_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
        longitude_ens_min:_FillValue = -9999. ;
    double longitude_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
        longitude_ens_max:_FillValue = -9999. ;
    double height_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
        height_ens_mean:_FillValue = -9999. ;
    double height_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
        height_ens_std:_FillValue = -9999. ;
    double height_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
        height_ens_min:_FillValue = -9999. ;
    double height_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
        height_ens_max:_FillValue = -9999. ;
    double pres_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
        pres_ens_mean:_FillValue = -9999. ;
```

double pres_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
pres_ens_std:_FillValue = -9999. ;
double pres_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
pres_ens_min:_FillValue = -9999. ;
double pres_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
pres_ens_max:_FillValue = -9999. ;
double theta_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
theta_ens_mean:_FillValue = -9999. ;
double theta_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
theta_ens_std:_FillValue = -9999. ;
double theta_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
theta_ens_min:_FillValue = -9999. ;
double theta_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
theta_ens_max:_FillValue = -9999. ;
double temp_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
temp_ens_mean:_FillValue = -9999. ;
double temp_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
temp_ens_std:_FillValue = -9999. ;
double temp_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
temp_ens_min:_FillValue = -9999. ;
double temp_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
temp_ens_max:_FillValue = -9999. ;
double pblh_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
pblh_ens_mean:_FillValue = -9999. ;
double pblh_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
pblh_ens_std:_FillValue = -9999. ;
double pblh_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
pblh_ens_min:_FillValue = -9999. ;
double pblh_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
pblh_ens_max:_FillValue = -9999. ;
double rh_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
rh_ens_mean:_FillValue = -9999. ;
double rh_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
rh_ens_std:_FillValue = -9999. ;
double rh_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
rh_ens_min:_FillValue = -9999. ;
double rh_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
rh_ens_max:_FillValue = -9999. ;
double surf_altitude_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
surf_altitude_ens_mean:_FillValue = -9999. ;
double surf_altitude_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
surf_altitude_ens_std:_FillValue = -9999. ;
double surf_altitude_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
surf_altitude_ens_min:_FillValue = -9999. ;
double surf_altitude_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
surf_altitude_ens_max:_FillValue = -9999. ;

double qv_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
qv_ens_mean:_FillValue = -9999. ;
double qv_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
qv_ens_std:_FillValue = -9999. ;
double qv_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
qv_ens_min:_FillValue = -9999. ;
double qv_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
qv_ens_max:_FillValue = -9999. ;
double rhi_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
rhi_ens_mean:_FillValue = -9999. ;
double rhi_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
rhi_ens_std:_FillValue = -9999. ;
double rhi_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
rhi_ens_min:_FillValue = -9999. ;
double rhi_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
rhi_ens_max:_FillValue = -9999. ;
double theta_e_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
theta_e_ens_mean:_FillValue = -9999. ;
double theta_e_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
theta_e_ens_std:_FillValue = -9999. ;
double theta_e_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
theta_e_ens_min:_FillValue = -9999. ;
double theta_e_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
theta_e_ens_max:_FillValue = -9999. ;
double theta_v_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
theta_v_ens_mean:_FillValue = -9999. ;
double theta_v_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
theta_v_ens_std:_FillValue = -9999. ;
double theta_v_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
theta_v_ens_min:_FillValue = -9999. ;
double theta_v_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
theta_v_ens_max:_FillValue = -9999. ;
double altitude_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
altitude_ens_mean:_FillValue = -9999. ;
double altitude_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
altitude_ens_std:_FillValue = -9999. ;
double altitude_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
altitude_ens_min:_FillValue = -9999. ;
double altitude_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
altitude_ens_max:_FillValue = -9999. ;
double wvert_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
wvert_ens_mean:_FillValue = -9999. ;
double wvert_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
wvert_ens_std:_FillValue = -9999. ;
double wvert_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
wvert_ens_min:_FillValue = -9999. ;

double wvert_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
wvert_ens_max:_FillValue = -9999. ;
double height_to_pblh_ratio_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
height_to_pblh_ratio_ens_mean:_FillValue = -9999. ;
double height_to_pblh_ratio_ens_std(time, trajectory_type, trajectory_time, vert_layer) ;
height_to_pblh_ratio_ens_std:_FillValue = -9999. ;
double height_to_pblh_ratio_ens_min(time, trajectory_type, trajectory_time, vert_layer) ;
height_to_pblh_ratio_ens_min:_FillValue = -9999. ;
double height_to_pblh_ratio_ens_max(time, trajectory_type, trajectory_time, vert_layer) ;
height_to_pblh_ratio_ens_max:_FillValue = -9999. ;
double sgs_orography_angle(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_angle:_FillValue = -9999. ;
sgs_orography_angle:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_angle:units = "radian" ;
double sgs_orography_angle_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_angle_ens_mean:_FillValue = -9999. ;
sgs_orography_angle_ens_mean:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_angle_ens_mean:units = "radian" ;
double sgs_orography_anisotropy(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_anisotropy:_FillValue = -9999. ;
sgs_orography_anisotropy:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_anisotropy:units = "1" ;
double sgs_orography_anisotropy_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_anisotropy_ens_mean:_FillValue = -9999. ;
sgs_orography_anisotropy_ens_mean:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_anisotropy_ens_mean:units = "1" ;
double sgs_orography_std(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_std:_FillValue = -9999. ;
sgs_orography_std:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_std:units = "m" ;
double sgs_orography_std_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_std_ens_mean:_FillValue = -9999. ;
sgs_orography_std_ens_mean:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_std_ens_mean:units = "m" ;
double sgs_orography_slope(time, trajectory_type, trajectory_time, vert_layer) ;

sgs_orography_slope:_FillValue = -9999. ;
sgs_orography_slope:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_slope:units = "1" ;
double sgs_orography_slope_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_slope_ens_mean:_FillValue = -9999. ;
sgs_orography_slope_ens_mean:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_slope_ens_mean:units = "1" ;
double land_sea_mask(time, trajectory_type, trajectory_time, vert_layer) ;
land_sea_mask:_FillValue = -9999. ;
land_sea_mask:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land)" ;
land_sea_mask:units = "1" ;
double land_sea_mask_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
land_sea_mask_ens_mean:_FillValue = -9999. ;
land_sea_mask_ens_mean:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along ensemble mean trajectory)" ;
land_sea_mask_ens_mean:units = "1" ;
float high_vegetation_type(time, trajectory_type, trajectory_time, vert_layer) ;
high_vegetation_type:_FillValue = -9999.f ;
high_vegetation_type:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation)" ;
high_vegetation_type:units = "1" ;
float high_vegetation_type_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
high_vegetation_type_ens_mean:_FillValue = -9999.f ;
high_vegetation_type_ens_mean:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along ensemble mean trajectory)" ;
high_vegetation_type_ens_mean:units = "1" ;
float low_vegetation_type(time, trajectory_type, trajectory_time, vert_layer) ;
low_vegetation_type:_FillValue = -9999.f ;
low_vegetation_type:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation)" ;
low_vegetation_type:units = "1" ;
float low_vegetation_type_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
low_vegetation_type_ens_mean:_FillValue = -9999.f ;
low_vegetation_type_ens_mean:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops,

11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along ensemble mean trajectory)" ;
low_vegetation_type_ens_mean:units = "1" ;
double high_vegetation_cover(time, trajectory_type, trajectory_time, vert_layer) ;
high_vegetation_cover:_FillValue = -9999. ;
high_vegetation_cover:long_name = "High vegetation cover fraction in airmass column" ;
high_vegetation_cover:units = "1" ;
double high_vegetation_cover_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
high_vegetation_cover_ens_mean:_FillValue = -9999. ;
high_vegetation_cover_ens_mean:long_name = "High vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
high_vegetation_cover_ens_mean:units = "1" ;
double low_vegetation_cover(time, trajectory_type, trajectory_time, vert_layer) ;
low_vegetation_cover:_FillValue = -9999. ;
low_vegetation_cover:long_name = "Low vegetation cover fraction in airmass column" ;
low_vegetation_cover:units = "1" ;
double low_vegetation_cover_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
low_vegetation_cover_ens_mean:_FillValue = -9999. ;
low_vegetation_cover_ens_mean:long_name = "Low vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
low_vegetation_cover_ens_mean:units = "1" ;
float soil_type(time, trajectory_type, trajectory_time, vert_layer) ;
soil_type:_FillValue = -9999.f ;
soil_type:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point)" ;
soil_type:units = "1" ;
soil_type:standard_name = "soil_type" ;
float soil_type_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
soil_type_ens_mean:_FillValue = -9999.f ;
soil_type_ens_mean:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along ensemble mean trajectory)" ;
soil_type_ens_mean:units = "1" ;
soil_type_ens_mean:standard_name = "soil_type" ;
double sea_ice_cover(time, trajectory_type, trajectory_time, vert_layer) ;
sea_ice_cover:_FillValue = -9999. ;
sea_ice_cover:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column" ;
sea_ice_cover:units = "1" ;
sea_ice_cover:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ens_mean(time, trajectory_type, trajectory_time, vert_layer) ;
sea_ice_cover_ens_mean:_FillValue = -9999. ;
sea_ice_cover_ens_mean:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along ensemble mean trajectory)" ;

sea_ice_cover_ens_mean:units = "1" ;
sea_ice_cover_ens_mean:standard_name = "sea_ice_area_fraction" ;
int time_dim_sonde(time_dim_sonde) ;
time_dim_sonde:long_name = "sonde time (sample) dimension entries" ;
time_dim_sonde:units = "1" ;
double rh_sonde(time, time_dim_sonde) ;
rh_sonde:_FillValue = -9999. ;
rh_sonde:long_name = "Relative Humidity" ;
rh_sonde:units = "%" ;
rh_sonde:valid_min = 0.f ;
rh_sonde:valid_max = 100.f ;
rh_sonde:resolution = 1.f ;
rh_sonde:standard_name = "relative_humidity" ;
double pres_sonde(time, time_dim_sonde) ;
pres_sonde:_FillValue = -9999. ;
pres_sonde:long_name = "Pressure" ;
pres_sonde:units = "hPa" ;
pres_sonde:valid_min = 0.f ;
pres_sonde:valid_max = 1100.f ;
pres_sonde:valid_delta = 10.f ;
pres_sonde:resolution = 0.1f ;
pres_sonde:standard_name = "air_pressure" ;
double temp_sonde(time, time_dim_sonde) ;
temp_sonde:_FillValue = -9999. ;
temp_sonde:long_name = "Dry Bulb Temperature" ;
temp_sonde:units = "degC" ;
temp_sonde:valid_min = -90.f ;
temp_sonde:valid_max = 50.f ;
temp_sonde:valid_delta = 10.f ;
temp_sonde:resolution = 0.1f ;
temp_sonde:standard_name = "air_temperature" ;
double alt_sonde(time, time_dim_sonde) ;
alt_sonde:_FillValue = -9999. ;
alt_sonde:long_name = "Altitude above mean sea level" ;
alt_sonde:units = "m" ;
alt_sonde:standard_name = "altitude" ;
double time_sonde(time, time_dim_sonde) ;
time_sonde:_FillValue = -9999. ;
time_sonde:long_name = "Time offset from midnight" ;
time_sonde:units = "s" ;
time_sonde:standard_name = "time" ;
double wspd_sonde(time, time_dim_sonde) ;
wspd_sonde:_FillValue = -9999. ;
wspd_sonde:long_name = "Wind Speed" ;
wspd_sonde:units = "m/s" ;
wspd_sonde:valid_min = 0.f ;

wspd_sonde:valid_max = 100.f ;
wspd_sonde:resolution = 0.1f ;
wspd_sonde:standard_name = "wind_speed" ;
double wdir_sonde(time, time_dim_sonde) ;
wdir_sonde:_FillValue = -9999. ;
wdir_sonde:long_name = "Wind Direction" ;
wdir_sonde:units = "degree" ;
wdir_sonde:valid_min = 0.f ;
wdir_sonde:valid_max = 360.f ;
wdir_sonde:resolution = 1.f ;
wdir_sonde:standard_name = "wind_from_direction" ;
int liq_cld_exist(time, time_dim_sonde) ;
liq_cld_exist:_FillValue = -9999 ;
liq_cld_exist:long_name = "1 - liquid droplets in grid cell, 0 - no liquid droplets in grid cell; determined using sonde RH threshold of 96.0%; layers distant by less than 50.0 meters were concatenated; layers thinner than 25.0 meters were removed" ;
liq_cld_exist:units = "1" ;
int liq_cld_base(time, vert_layer) ;
liq_cld_base:_FillValue = -9999 ;
liq_cld_base:long_name = "Liquid cloud base altitude; determined using sonde RH threshold of 96.0%; layers distant by less than 50.0 meters were concatenated; layers thinner than 25.0 meters were removed" ;
liq_cld_base:units = "m" ;
int liq_cld_top(time, vert_layer) ;
liq_cld_top:_FillValue = -9999 ;
liq_cld_top:long_name = "Liquid cloud top altitude; determined using sonde RH threshold of 96.0%; layers distant by less than 50.0 meters were concatenated; layers thinner than 25.0 meters were removed" ;
liq_cld_top:units = "m" ;
double time(time) ;
time:long_name = "Time in seconds since volume start" ;
time:units = "seconds since 2023-08-20 00:00:00 0:00" ;
double time_offset(time) ;
time_offset:_FillValue = -9999. ;
time_offset:long_name = "Time offset from base_time" ;
time_offset:units = "seconds since 2023-08-20 00:00:00 0:00" ;
int base_time ;
base_time:_FillValue = -9999 ;
base_time:long_name = "Base time in Epoch" ;
base_time:units = "seconds since 1970-01-01 0:00:00 0:00" ;
double lat ;
lat:_FillValue = -9999. ;
lat:long_name = "latitude" ;
lat:units = "degree_N" ;
double lon ;
lon:_FillValue = -9999. ;

lon:long_name = "longitude" ;
lon:units = "degree_E" ;
double alt ;
alt:_FillValue = -9999. ;
alt:long_name = "Altitude above mean sea level" ;
alt:units = "m" ;

// global attributes:
:mode = "cld" ;
:ensemble_size = "27" ;
:ensemble_dx_from_center = "[-7.5 0.   7.5] km" ;
:ensemble_dy_from_center = "[-7.5 0.   7.5] km" ;
:ensemble_dz_from_center = "[-50   0 50] m" ;
:used_era5_resolution = "0.25 degrees" ;
:total_trajectory_hours = "[-120, 120]" ;
:number_of_starting_center_points_per_time_step = "[1 1 5 4]" ;
:title = "Back and forward trajectories for liquid-bearing cloud layers in full tropospheric profiles" ;
:summary = "The ARMTRAJ VAP provides trajectory datasets initialized at ARM deployment coordinates and configured using ARM datasets. The trajectory datasets support aerosol, cloud, and planetary boundary layer research. Trajectory calculations use the HYSPLIT model informed by the ERA5 reanalysis dataset at its highest spatial resolution (~31 km). For each sample in each of the datasets, HYSPLIT also runs at multiple starting locations surrounding ARM deployments, enabling an ensemble of runs from which the mean and variability (estimated uncertainty) of each sample\'s trajectory coordinates, thermodynamic properties, or other fields are reported." ;
:keywords = "ARM, ground-based measurements, ground-based remote sensing, clouds, aerosol, PBL, trajectories, Lagrangian analysis" ;
:doi = "10.5439/2309851" ;
:location_description = "Eastern Pacific Cloud Aerosol Precipitation Experiment (EPCAPE), Scripps Pier, La Jolla, CA" ;
:dod_version = "v1.0" ;
:command_line = "python armtraj_run.py" ;
:datastream = "epcarmtrajcldM1.c1" ;
:data_level = "c1" ;
:facility_id = "M1" ;
:site_id = "epc" ;
:platform_id = "armtrajcld" ;
:history = "created by user isilber1 on machine dev-proc2 at 27-Sep-2024,15:26:37" ;
}

Appendix C

ARMTRAJ-PBL Output Data

```
netcdf epcarmtrajpblM1.c1.20230820.000000 {
dimensions:
        time = UNLIMITED ; // (4 currently)
        vert_layer = 3 ;
        trajectory_time = 121 ;
        trajectory_type = 1 ;
variables:
        int vert_layer(vert_layer) ;
                vert_layer:long_name = "Vertical layer index (bottom-up) (0 - PBL base, 5 - PBL mid,
10 - PBL top)" ;
                vert_layer:units = "1" ;
        int trajectory_time(trajectory_time) ;
                trajectory_time:long_name = "Trajectory offset from start time (0 h)" ;
                trajectory_time:units = "h" ;
        int trajectory_type(trajectory_type) ;
                trajectory_type:long_name = "-1 - back trajectories, 1 - fwd trajectories" ;
                trajectory_type:units = "1" ;
        double date(time, trajectory_time) ;
                date:_FillValue = -9999. ;
                date:long_name = "Trajectory date per trajectory type and time step" ;
                date:units = "seconds since 1970-01-01" ;
        double latitude(time, trajectory_time, vert_layer) ;
                latitude:_FillValue = -9999. ;
                latitude:long_name = "Airmass latitude" ;
                latitude:units = "degree_N" ;
                latitude:standard_name = "grid_latitude" ;
        double longitude(time, trajectory_time, vert_layer) ;
                longitude:_FillValue = -9999. ;
                longitude:long_name = "Airmass longitude" ;
                longitude:units = "degree_E" ;
                longitude:standard_name = "grid_longitude" ;
        double height(time, trajectory_time, vert_layer) ;
                height:_FillValue = -9999. ;
                height:long_name = "Airmass height above ground level" ;
```

height:units = "m" ;
height:standard_name = "height" ;
double pres(time, trajectory_time, vert_layer) ;
pres:_FillValue = -9999. ;
pres:long_name = "Airmass pressure" ;
pres:units = "hPa" ;
pres:standard_name = "air_pressure" ;
double theta(time, trajectory_time, vert_layer) ;
theta:_FillValue = -9999. ;
theta:long_name = "Airmass potential temperature" ;
theta:units = "K" ;
theta:standard_name = "air_potential_temperature" ;
double temp(time, trajectory_time, vert_layer) ;
temp:_FillValue = -9999. ;
temp:long_name = "Airmass temperature" ;
temp:units = "degC" ;
temp:standard_name = "air_temperature" ;
double pblh(time, trajectory_time, vert_layer) ;
pblh:_FillValue = -9999. ;
pblh:long_name = "Planetary boundary layer height in airmass column" ;
pblh:units = "m" ;
pblh:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh(time, trajectory_time, vert_layer) ;
rh:_FillValue = -9999. ;
rh:long_name = "Airmass relative humidity" ;
rh:units = "%" ;
rh:standard_name = "relative_humidity" ;
double surf_altitude(time, trajectory_time, vert_layer) ;
surf_altitude:_FillValue = -9999. ;
surf_altitude:long_name = "Surface altitude above mean sea level in airmass column" ;
surf_altitude:units = "m" ;
surf_altitude:standard_name = "surface_altitude" ;
double qv(time, trajectory_time, vert_layer) ;
qv:_FillValue = -9999. ;
qv:long_name = "Airmass specific humidity" ;
qv:units = "g/kg" ;
qv:standard_name = "specific_humidity" ;
double rhi(time, trajectory_time, vert_layer) ;
rhi:_FillValue = -9999. ;
rhi:long_name = "Airmass relative humidity with respect to ice" ;
rhi:units = "%" ;
double theta_e(time, trajectory_time, vert_layer) ;
theta_e:_FillValue = -9999. ;
theta_e:long_name = "Airmass equivalent potential temperature (excluding condensate from the calculation)" ;
theta_e:units = "K" ;

```
                theta_e:standard_name = "air_equivalent_potential_temperature" ;
        double theta_v(time, trajectory_time, vert_layer) ;
                theta_v:_FillValue = -9999. ;
                theta_v:long_name = "Airmass virtual potential temperature" ;
                theta_v:units = "K" ;
        double altitude(time, trajectory_time, vert_layer) ;
                altitude:_FillValue = -9999. ;
                altitude:long_name = "Airmass altitude above mean sea level" ;
                altitude:units = "m" ;
                altitude:standard_name = "height_above_mean_sea_level" ;
        double wvert(time, trajectory_time, vert_layer) ;
                wvert:_FillValue = -9999. ;
                wvert:long_name = "Mean hourly airmass ascent rate" ;
                wvert:units = "m/s" ;
                wvert:standard_name = "upward_air_velocity" ;
        double height_to_pblh_ratio(time, trajectory_time, vert_layer) ;
                height_to_pblh_ratio:_FillValue = -9999. ;
                height_to_pblh_ratio:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above
PBL; <1 - airmass within PBL)" ;
                height_to_pblh_ratio:units = "1" ;
        double latitude_ft(time, trajectory_time) ;
                latitude_ft:_FillValue = -9999. ;
                latitude_ft:long_name = "Free-troposphere (at initialization) airmass latitude" ;
                latitude_ft:units = "degree_N" ;
                latitude_ft:standard_name = "grid_latitude" ;
        double longitude_ft(time, trajectory_time) ;
                longitude_ft:_FillValue = -9999. ;
                longitude_ft:long_name = "Free-troposphere (at initialization) airmass longitude" ;
                longitude_ft:units = "degree_E" ;
                longitude_ft:standard_name = "grid_longitude" ;
        double height_ft(time, trajectory_time) ;
                height_ft:_FillValue = -9999. ;
                height_ft:long_name = "Free-troposphere (at initialization) airmass height above ground
level" ;
                height_ft:units = "m" ;
                height_ft:standard_name = "height" ;
        double pres_ft(time, trajectory_time) ;
                pres_ft:_FillValue = -9999. ;
                pres_ft:long_name = "Free-troposphere (at initialization) airmass pressure" ;
                pres_ft:units = "hPa" ;
                pres_ft:standard_name = "air_pressure" ;
        double theta_ft(time, trajectory_time) ;
                theta_ft:_FillValue = -9999. ;
                theta_ft:long_name = "Free-troposphere (at initialization) airmass potential temperature"
;
                theta_ft:units = "K" ;
```

theta_ft:standard_name = "air_potential_temperature" ;
double temp_ft(time, trajectory_time) ;
temp_ft:_FillValue = -9999. ;
temp_ft:long_name = "Free-troposphere (at initialization) airmass temperature" ;
temp_ft:units = "degC" ;
temp_ft:standard_name = "air_temperature" ;
double pblh_ft(time, trajectory_time) ;
pblh_ft:_FillValue = -9999. ;
pblh_ft:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column" ;
pblh_ft:units = "m" ;
pblh_ft:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh_ft(time, trajectory_time) ;
rh_ft:_FillValue = -9999. ;
rh_ft:long_name = "Free-troposphere (at initialization) airmass relative humidity" ;
rh_ft:units = "%" ;
rh_ft:standard_name = "relative_humidity" ;
double surf_altitude_ft(time, trajectory_time) ;
surf_altitude_ft:_FillValue = -9999. ;
surf_altitude_ft:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column" ;
surf_altitude_ft:units = "m" ;
surf_altitude_ft:standard_name = "surface_altitude" ;
double qv_ft(time, trajectory_time) ;
qv_ft:_FillValue = -9999. ;
qv_ft:long_name = "Free-troposphere (at initialization) airmass specific humidity" ;
qv_ft:units = "g/kg" ;
qv_ft:standard_name = "specific_humidity" ;
double rhi_ft(time, trajectory_time) ;
rhi_ft:_FillValue = -9999. ;
rhi_ft:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice" ;
rhi_ft:units = "%" ;
double theta_e_ft(time, trajectory_time) ;
theta_e_ft:_FillValue = -9999. ;
theta_e_ft:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation)" ;
theta_e_ft:units = "K" ;
theta_e_ft:standard_name = "air_equivalent_potential_temperature" ;
double theta_v_ft(time, trajectory_time) ;
theta_v_ft:_FillValue = -9999. ;
theta_v_ft:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature" ;
theta_v_ft:units = "K" ;
double altitude_ft(time, trajectory_time) ;
altitude_ft:_FillValue = -9999. ;

altitude_ft:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level" ;
altitude_ft:units = "m" ;
altitude_ft:standard_name = "height_above_mean_sea_level" ;
double wvert_ft(time, trajectory_time) ;
wvert_ft:_FillValue = -9999. ;
wvert_ft:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate" ;
wvert_ft:units = "m/s" ;
wvert_ft:standard_name = "upward_air_velocity" ;
double height_to_pblh_ratio_ft(time, trajectory_time) ;
height_to_pblh_ratio_ft:_FillValue = -9999. ;
height_to_pblh_ratio_ft:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl)" ;
height_to_pblh_ratio_ft:units = "1" ;
double latitude_ens_mean(time, trajectory_time) ;
latitude_ens_mean:_FillValue = -9999. ;
latitude_ens_mean:long_name = "Airmass latitude ensemble mean" ;
latitude_ens_mean:units = "degree_N" ;
latitude_ens_mean:standard_name = "grid_latitude" ;
double latitude_ens_std(time, trajectory_time) ;
latitude_ens_std:_FillValue = -9999. ;
latitude_ens_std:long_name = "Airmass latitude ensemble std" ;
latitude_ens_std:units = "degree_N" ;
latitude_ens_std:standard_name = "grid_latitude" ;
double latitude_ens_min(time, trajectory_time) ;
latitude_ens_min:_FillValue = -9999. ;
latitude_ens_min:long_name = "Airmass latitude ensemble min" ;
latitude_ens_min:units = "degree_N" ;
latitude_ens_min:standard_name = "grid_latitude" ;
double latitude_ens_max(time, trajectory_time) ;
latitude_ens_max:_FillValue = -9999. ;
latitude_ens_max:long_name = "Airmass latitude ensemble max" ;
latitude_ens_max:units = "degree_N" ;
latitude_ens_max:standard_name = "grid_latitude" ;
double longitude_ens_mean(time, trajectory_time) ;
longitude_ens_mean:_FillValue = -9999. ;
longitude_ens_mean:long_name = "Airmass longitude ensemble mean" ;
longitude_ens_mean:units = "degree_E" ;
longitude_ens_mean:standard_name = "grid_longitude" ;
double longitude_ens_std(time, trajectory_time) ;
longitude_ens_std:_FillValue = -9999. ;
longitude_ens_std:long_name = "Airmass longitude ensemble std" ;
longitude_ens_std:units = "degree_E" ;
longitude_ens_std:standard_name = "grid_longitude" ;
double longitude_ens_min(time, trajectory_time) ;

longitude_ens_min:_FillValue = -9999. ;
longitude_ens_min:long_name = "Airmass longitude ensemble min" ;
longitude_ens_min:units = "degree_E" ;
longitude_ens_min:standard_name = "grid_longitude" ;
double longitude_ens_max(time, trajectory_time) ;
longitude_ens_max:_FillValue = -9999. ;
longitude_ens_max:long_name = "Airmass longitude ensemble max" ;
longitude_ens_max:units = "degree_E" ;
longitude_ens_max:standard_name = "grid_longitude" ;
double height_ens_mean(time, trajectory_time) ;
height_ens_mean:_FillValue = -9999. ;
height_ens_mean:long_name = "Airmass height above ground level ensemble mean" ;
height_ens_mean:units = "m" ;
height_ens_mean:standard_name = "height" ;
double height_ens_std(time, trajectory_time) ;
height_ens_std:_FillValue = -9999. ;
height_ens_std:long_name = "Airmass height above ground level ensemble std" ;
height_ens_std:units = "m" ;
height_ens_std:standard_name = "height" ;
double height_ens_min(time, trajectory_time) ;
height_ens_min:_FillValue = -9999. ;
height_ens_min:long_name = "Airmass height above ground level ensemble min" ;
height_ens_min:units = "m" ;
height_ens_min:standard_name = "height" ;
double height_ens_max(time, trajectory_time) ;
height_ens_max:_FillValue = -9999. ;
height_ens_max:long_name = "Airmass height above ground level ensemble max" ;
height_ens_max:units = "m" ;
height_ens_max:standard_name = "height" ;
double pres_ens_mean(time, trajectory_time) ;
pres_ens_mean:_FillValue = -9999. ;
pres_ens_mean:long_name = "Airmass pressure ensemble mean" ;
pres_ens_mean:units = "hPa" ;
pres_ens_mean:standard_name = "air_pressure" ;
double pres_ens_std(time, trajectory_time) ;
pres_ens_std:_FillValue = -9999. ;
pres_ens_std:long_name = "Airmass pressure ensemble std" ;
pres_ens_std:units = "hPa" ;
pres_ens_std:standard_name = "air_pressure" ;
double pres_ens_min(time, trajectory_time) ;
pres_ens_min:_FillValue = -9999. ;
pres_ens_min:long_name = "Airmass pressure ensemble min" ;
pres_ens_min:units = "hPa" ;
pres_ens_min:standard_name = "air_pressure" ;
double pres_ens_max(time, trajectory_time) ;
pres_ens_max:_FillValue = -9999. ;

pres_ens_max:long_name = "Airmass pressure ensemble max" ;
pres_ens_max:units = "hPa" ;
pres_ens_max:standard_name = "air_pressure" ;
double theta_ens_mean(time, trajectory_time) ;
theta_ens_mean:_FillValue = -9999. ;
theta_ens_mean:long_name = "Airmass potential temperature ensemble mean" ;
theta_ens_mean:units = "K" ;
theta_ens_mean:standard_name = "air_potential_temperature" ;
double theta_ens_std(time, trajectory_time) ;
theta_ens_std:_FillValue = -9999. ;
theta_ens_std:long_name = "Airmass potential temperature ensemble std" ;
theta_ens_std:units = "K" ;
theta_ens_std:standard_name = "air_potential_temperature" ;
double theta_ens_min(time, trajectory_time) ;
theta_ens_min:_FillValue = -9999. ;
theta_ens_min:long_name = "Airmass potential temperature ensemble min" ;
theta_ens_min:units = "K" ;
theta_ens_min:standard_name = "air_potential_temperature" ;
double theta_ens_max(time, trajectory_time) ;
theta_ens_max:_FillValue = -9999. ;
theta_ens_max:long_name = "Airmass potential temperature ensemble max" ;
theta_ens_max:units = "K" ;
theta_ens_max:standard_name = "air_potential_temperature" ;
double temp_ens_mean(time, trajectory_time) ;
temp_ens_mean:_FillValue = -9999. ;
temp_ens_mean:long_name = "Airmass temperature ensemble mean" ;
temp_ens_mean:units = "degC" ;
temp_ens_mean:standard_name = "air_temperature" ;
double temp_ens_std(time, trajectory_time) ;
temp_ens_std:_FillValue = -9999. ;
temp_ens_std:long_name = "Airmass temperature ensemble std" ;
temp_ens_std:units = "degC" ;
temp_ens_std:standard_name = "air_temperature" ;
double temp_ens_min(time, trajectory_time) ;
temp_ens_min:_FillValue = -9999. ;
temp_ens_min:long_name = "Airmass temperature ensemble min" ;
temp_ens_min:units = "degC" ;
temp_ens_min:standard_name = "air_temperature" ;
double temp_ens_max(time, trajectory_time) ;
temp_ens_max:_FillValue = -9999. ;
temp_ens_max:long_name = "Airmass temperature ensemble max" ;
temp_ens_max:units = "degC" ;
temp_ens_max:standard_name = "air_temperature" ;
double pblh_ens_mean(time, trajectory_time) ;
pblh_ens_mean:_FillValue = -9999. ;

pblh_ens_mean:long_name = "Planetary boundary layer height in airmass column ensemble mean" ;
pblh_ens_mean:units = "m" ;
pblh_ens_mean:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ens_std(time, trajectory_time) ;
pblh_ens_std:_FillValue = -9999. ;
pblh_ens_std:long_name = "Planetary boundary layer height in airmass column ensemble std" ;
pblh_ens_std:units = "m" ;
pblh_ens_std:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ens_min(time, trajectory_time) ;
pblh_ens_min:_FillValue = -9999. ;
pblh_ens_min:long_name = "Planetary boundary layer height in airmass column ensemble min" ;
pblh_ens_min:units = "m" ;
pblh_ens_min:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ens_max(time, trajectory_time) ;
pblh_ens_max:_FillValue = -9999. ;
pblh_ens_max:long_name = "Planetary boundary layer height in airmass column ensemble max" ;
pblh_ens_max:units = "m" ;
pblh_ens_max:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh_ens_mean(time, trajectory_time) ;
rh_ens_mean:_FillValue = -9999. ;
rh_ens_mean:long_name = "Airmass relative humidity ensemble mean" ;
rh_ens_mean:units = "%" ;
rh_ens_mean:standard_name = "relative_humidity" ;
double rh_ens_std(time, trajectory_time) ;
rh_ens_std:_FillValue = -9999. ;
rh_ens_std:long_name = "Airmass relative humidity ensemble std" ;
rh_ens_std:units = "%" ;
rh_ens_std:standard_name = "relative_humidity" ;
double rh_ens_min(time, trajectory_time) ;
rh_ens_min:_FillValue = -9999. ;
rh_ens_min:long_name = "Airmass relative humidity ensemble min" ;
rh_ens_min:units = "%" ;
rh_ens_min:standard_name = "relative_humidity" ;
double rh_ens_max(time, trajectory_time) ;
rh_ens_max:_FillValue = -9999. ;
rh_ens_max:long_name = "Airmass relative humidity ensemble max" ;
rh_ens_max:units = "%" ;
rh_ens_max:standard_name = "relative_humidity" ;
double surf_altitude_ens_mean(time, trajectory_time) ;
surf_altitude_ens_mean:_FillValue = -9999. ;
surf_altitude_ens_mean:long_name = "Surface altitude above mean sea level in airmass column ensemble mean" ;

surf_altitude_ens_mean:units = "m" ;
surf_altitude_ens_mean:standard_name = "surface_altitude" ;
double surf_altitude_ens_std(time, trajectory_time) ;
surf_altitude_ens_std:_FillValue = -9999. ;
surf_altitude_ens_std:long_name = "Surface altitude above mean sea level in airmass column ensemble std" ;
surf_altitude_ens_std:units = "m" ;
surf_altitude_ens_std:standard_name = "surface_altitude" ;
double surf_altitude_ens_min(time, trajectory_time) ;
surf_altitude_ens_min:_FillValue = -9999. ;
surf_altitude_ens_min:long_name = "Surface altitude above mean sea level in airmass column ensemble min" ;
surf_altitude_ens_min:units = "m" ;
surf_altitude_ens_min:standard_name = "surface_altitude" ;
double surf_altitude_ens_max(time, trajectory_time) ;
surf_altitude_ens_max:_FillValue = -9999. ;
surf_altitude_ens_max:long_name = "Surface altitude above mean sea level in airmass column ensemble max" ;
surf_altitude_ens_max:units = "m" ;
surf_altitude_ens_max:standard_name = "surface_altitude" ;
double qv_ens_mean(time, trajectory_time) ;
qv_ens_mean:_FillValue = -9999. ;
qv_ens_mean:long_name = "Airmass specific humidity ensemble mean" ;
qv_ens_mean:units = "g/kg" ;
qv_ens_mean:standard_name = "specific_humidity" ;
double qv_ens_std(time, trajectory_time) ;
qv_ens_std:_FillValue = -9999. ;
qv_ens_std:long_name = "Airmass specific humidity ensemble std" ;
qv_ens_std:units = "g/kg" ;
qv_ens_std:standard_name = "specific_humidity" ;
double qv_ens_min(time, trajectory_time) ;
qv_ens_min:_FillValue = -9999. ;
qv_ens_min:long_name = "Airmass specific humidity ensemble min" ;
qv_ens_min:units = "g/kg" ;
qv_ens_min:standard_name = "specific_humidity" ;
double qv_ens_max(time, trajectory_time) ;
qv_ens_max:_FillValue = -9999. ;
qv_ens_max:long_name = "Airmass specific humidity ensemble max" ;
qv_ens_max:units = "g/kg" ;
qv_ens_max:standard_name = "specific_humidity" ;
double rhi_ens_mean(time, trajectory_time) ;
rhi_ens_mean:_FillValue = -9999. ;
rhi_ens_mean:long_name = "Airmass relative humidity with respect to ice ensemble mean" ;
rhi_ens_mean:units = "%" ;
double rhi_ens_std(time, trajectory_time) ;

rhi_ens_std:_FillValue = -9999. ;
rhi_ens_std:long_name = "Airmass relative humidity with respect to ice ensemble std" ;
rhi_ens_std:units = "%" ;
double rhi_ens_min(time, trajectory_time) ;
rhi_ens_min:_FillValue = -9999. ;
rhi_ens_min:long_name = "Airmass relative humidity with respect to ice ensemble min" ;
rhi_ens_min:units = "%" ;
double rhi_ens_max(time, trajectory_time) ;
rhi_ens_max:_FillValue = -9999. ;
rhi_ens_max:long_name = "Airmass relative humidity with respect to ice ensemble max"
;
rhi_ens_max:units = "%" ;
double theta_e_ens_mean(time, trajectory_time) ;
theta_e_ens_mean:_FillValue = -9999. ;
theta_e_ens_mean:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble mean" ;
theta_e_ens_mean:units = "K" ;
theta_e_ens_mean:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ens_std(time, trajectory_time) ;
theta_e_ens_std:_FillValue = -9999. ;
theta_e_ens_std:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble std" ;
theta_e_ens_std:units = "K" ;
theta_e_ens_std:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ens_min(time, trajectory_time) ;
theta_e_ens_min:_FillValue = -9999. ;
theta_e_ens_min:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble min" ;
theta_e_ens_min:units = "K" ;
theta_e_ens_min:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ens_max(time, trajectory_time) ;
theta_e_ens_max:_FillValue = -9999. ;
theta_e_ens_max:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble max" ;
theta_e_ens_max:units = "K" ;
theta_e_ens_max:standard_name = "air_equivalent_potential_temperature" ;
double theta_v_ens_mean(time, trajectory_time) ;
theta_v_ens_mean:_FillValue = -9999. ;
theta_v_ens_mean:long_name = "Airmass virtual potential temperature ensemble mean" ;
theta_v_ens_mean:units = "K" ;
double theta_v_ens_std(time, trajectory_time) ;
theta_v_ens_std:_FillValue = -9999. ;
theta_v_ens_std:long_name = "Airmass virtual potential temperature ensemble std" ;
theta_v_ens_std:units = "K" ;
double theta_v_ens_min(time, trajectory_time) ;
theta_v_ens_min:_FillValue = -9999. ;

```
        theta_v_ens_min:long_name = "Airmass virtual potential temperature ensemble min" ;
        theta_v_ens_min:units = "K" ;
    double theta_v_ens_max(time, trajectory_time) ;
        theta_v_ens_max:_FillValue = -9999. ;
        theta_v_ens_max:long_name = "Airmass virtual potential temperature ensemble max" ;
        theta_v_ens_max:units = "K" ;
    double altitude_ens_mean(time, trajectory_time) ;
        altitude_ens_mean:_FillValue = -9999. ;
        altitude_ens_mean:long_name = "Airmass altitude above mean sea level ensemble mean"
;
        altitude_ens_mean:units = "m" ;
        altitude_ens_mean:standard_name = "height_above_mean_sea_level" ;
    double altitude_ens_std(time, trajectory_time) ;
        altitude_ens_std:_FillValue = -9999. ;
        altitude_ens_std:long_name = "Airmass altitude above mean sea level ensemble std" ;
        altitude_ens_std:units = "m" ;
        altitude_ens_std:standard_name = "height_above_mean_sea_level" ;
    double altitude_ens_min(time, trajectory_time) ;
        altitude_ens_min:_FillValue = -9999. ;
        altitude_ens_min:long_name = "Airmass altitude above mean sea level ensemble min" ;
        altitude_ens_min:units = "m" ;
        altitude_ens_min:standard_name = "height_above_mean_sea_level" ;
    double altitude_ens_max(time, trajectory_time) ;
        altitude_ens_max:_FillValue = -9999. ;
        altitude_ens_max:long_name = "Airmass altitude above mean sea level ensemble max" ;
        altitude_ens_max:units = "m" ;
        altitude_ens_max:standard_name = "height_above_mean_sea_level" ;
    double wvert_ens_mean(time, trajectory_time) ;
        wvert_ens_mean:_FillValue = -9999. ;
        wvert_ens_mean:long_name = "Mean hourly airmass ascent rate ensemble mean" ;
        wvert_ens_mean:units = "m/s" ;
        wvert_ens_mean:standard_name = "upward_air_velocity" ;
    double wvert_ens_std(time, trajectory_time) ;
        wvert_ens_std:_FillValue = -9999. ;
        wvert_ens_std:long_name = "Mean hourly airmass ascent rate ensemble std" ;
        wvert_ens_std:units = "m/s" ;
        wvert_ens_std:standard_name = "upward_air_velocity" ;
    double wvert_ens_min(time, trajectory_time) ;
        wvert_ens_min:_FillValue = -9999. ;
        wvert_ens_min:long_name = "Mean hourly airmass ascent rate ensemble min" ;
        wvert_ens_min:units = "m/s" ;
        wvert_ens_min:standard_name = "upward_air_velocity" ;
    double wvert_ens_max(time, trajectory_time) ;
        wvert_ens_max:_FillValue = -9999. ;
        wvert_ens_max:long_name = "Mean hourly airmass ascent rate ensemble max" ;
        wvert_ens_max:units = "m/s" ;
```

wvert_ens_max:standard_name = "upward_air_velocity" ;
double height_to_pblh_ratio_ens_mean(time, trajectory_time) ;
height_to_pblh_ratio_ens_mean:_FillValue = -9999. ;
height_to_pblh_ratio_ens_mean:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL) ensemble mean" ;
height_to_pblh_ratio_ens_mean:units = "1" ;
double height_to_pblh_ratio_ens_std(time, trajectory_time) ;
height_to_pblh_ratio_ens_std:_FillValue = -9999. ;
height_to_pblh_ratio_ens_std:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL) ensemble std" ;
height_to_pblh_ratio_ens_std:units = "1" ;
double height_to_pblh_ratio_ens_min(time, trajectory_time) ;
height_to_pblh_ratio_ens_min:_FillValue = -9999. ;
height_to_pblh_ratio_ens_min:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL) ensemble min" ;
height_to_pblh_ratio_ens_min:units = "1" ;
double height_to_pblh_ratio_ens_max(time, trajectory_time) ;
height_to_pblh_ratio_ens_max:_FillValue = -9999. ;
height_to_pblh_ratio_ens_max:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL) ensemble max" ;
height_to_pblh_ratio_ens_max:units = "1" ;
double latitude_ft_ens_mean(time, trajectory_time) ;
latitude_ft_ens_mean:_FillValue = -9999. ;
latitude_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass latitude ensemble mean" ;
latitude_ft_ens_mean:units = "degree_N" ;
latitude_ft_ens_mean:standard_name = "grid_latitude" ;
double latitude_ft_ens_std(time, trajectory_time) ;
latitude_ft_ens_std:_FillValue = -9999. ;
latitude_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass latitude ensemble std" ;
latitude_ft_ens_std:units = "degree_N" ;
latitude_ft_ens_std:standard_name = "grid_latitude" ;
double latitude_ft_ens_min(time, trajectory_time) ;
latitude_ft_ens_min:_FillValue = -9999. ;
latitude_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass latitude ensemble min" ;
latitude_ft_ens_min:units = "degree_N" ;
latitude_ft_ens_min:standard_name = "grid_latitude" ;
double latitude_ft_ens_max(time, trajectory_time) ;
latitude_ft_ens_max:_FillValue = -9999. ;
latitude_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass latitude ensemble max" ;
latitude_ft_ens_max:units = "degree_N" ;
latitude_ft_ens_max:standard_name = "grid_latitude" ;
double longitude_ft_ens_mean(time, trajectory_time) ;

longitude_ft_ens_mean:_FillValue = -9999. ;
longitude_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass longitude ensemble mean" ;
longitude_ft_ens_mean:units = "degree_E" ;
longitude_ft_ens_mean:standard_name = "grid_longitude" ;
double longitude_ft_ens_std(time, trajectory_time) ;
longitude_ft_ens_std:_FillValue = -9999. ;
longitude_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass longitude ensemble std" ;
longitude_ft_ens_std:units = "degree_E" ;
longitude_ft_ens_std:standard_name = "grid_longitude" ;
double longitude_ft_ens_min(time, trajectory_time) ;
longitude_ft_ens_min:_FillValue = -9999. ;
longitude_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass longitude ensemble min" ;
longitude_ft_ens_min:units = "degree_E" ;
longitude_ft_ens_min:standard_name = "grid_longitude" ;
double longitude_ft_ens_max(time, trajectory_time) ;
longitude_ft_ens_max:_FillValue = -9999. ;
longitude_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass longitude ensemble max" ;
longitude_ft_ens_max:units = "degree_E" ;
longitude_ft_ens_max:standard_name = "grid_longitude" ;
double height_ft_ens_mean(time, trajectory_time) ;
height_ft_ens_mean:_FillValue = -9999. ;
height_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass height above ground level ensemble mean" ;
height_ft_ens_mean:units = "m" ;
height_ft_ens_mean:standard_name = "height" ;
double height_ft_ens_std(time, trajectory_time) ;
height_ft_ens_std:_FillValue = -9999. ;
height_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass height above ground level ensemble std" ;
height_ft_ens_std:units = "m" ;
height_ft_ens_std:standard_name = "height" ;
double height_ft_ens_min(time, trajectory_time) ;
height_ft_ens_min:_FillValue = -9999. ;
height_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass height above ground level ensemble min" ;
height_ft_ens_min:units = "m" ;
height_ft_ens_min:standard_name = "height" ;
double height_ft_ens_max(time, trajectory_time) ;
height_ft_ens_max:_FillValue = -9999. ;
height_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass height above ground level ensemble max" ;
height_ft_ens_max:units = "m" ;

height_ft_ens_max:standard_name = "height" ;
double pres_ft_ens_mean(time, trajectory_time) ;
pres_ft_ens_mean:_FillValue = -9999. ;
pres_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass pressure ensemble mean" ;
pres_ft_ens_mean:units = "hPa" ;
pres_ft_ens_mean:standard_name = "air_pressure" ;
double pres_ft_ens_std(time, trajectory_time) ;
pres_ft_ens_std:_FillValue = -9999. ;
pres_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass pressure ensemble std" ;
pres_ft_ens_std:units = "hPa" ;
pres_ft_ens_std:standard_name = "air_pressure" ;
double pres_ft_ens_min(time, trajectory_time) ;
pres_ft_ens_min:_FillValue = -9999. ;
pres_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass pressure ensemble min" ;
pres_ft_ens_min:units = "hPa" ;
pres_ft_ens_min:standard_name = "air_pressure" ;
double pres_ft_ens_max(time, trajectory_time) ;
pres_ft_ens_max:_FillValue = -9999. ;
pres_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass pressure ensemble max" ;
pres_ft_ens_max:units = "hPa" ;
pres_ft_ens_max:standard_name = "air_pressure" ;
double theta_ft_ens_mean(time, trajectory_time) ;
theta_ft_ens_mean:_FillValue = -9999. ;
theta_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass potential temperature ensemble mean" ;
theta_ft_ens_mean:units = "K" ;
theta_ft_ens_mean:standard_name = "air_potential_temperature" ;
double theta_ft_ens_std(time, trajectory_time) ;
theta_ft_ens_std:_FillValue = -9999. ;
theta_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass potential temperature ensemble std" ;
theta_ft_ens_std:units = "K" ;
theta_ft_ens_std:standard_name = "air_potential_temperature" ;
double theta_ft_ens_min(time, trajectory_time) ;
theta_ft_ens_min:_FillValue = -9999. ;
theta_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass potential temperature ensemble min" ;
theta_ft_ens_min:units = "K" ;
theta_ft_ens_min:standard_name = "air_potential_temperature" ;
double theta_ft_ens_max(time, trajectory_time) ;
theta_ft_ens_max:_FillValue = -9999. ;

theta_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass potential temperature ensemble max" ;
theta_ft_ens_max:units = "K" ;
theta_ft_ens_max:standard_name = "air_potential_temperature" ;
double temp_ft_ens_mean(time, trajectory_time) ;
temp_ft_ens_mean:_FillValue = -9999. ;
temp_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass temperature ensemble mean" ;
temp_ft_ens_mean:units = "degC" ;
temp_ft_ens_mean:standard_name = "air_temperature" ;
double temp_ft_ens_std(time, trajectory_time) ;
temp_ft_ens_std:_FillValue = -9999. ;
temp_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass temperature ensemble std" ;
temp_ft_ens_std:units = "degC" ;
temp_ft_ens_std:standard_name = "air_temperature" ;
double temp_ft_ens_min(time, trajectory_time) ;
temp_ft_ens_min:_FillValue = -9999. ;
temp_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass temperature ensemble min" ;
temp_ft_ens_min:units = "degC" ;
temp_ft_ens_min:standard_name = "air_temperature" ;
double temp_ft_ens_max(time, trajectory_time) ;
temp_ft_ens_max:_FillValue = -9999. ;
temp_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass temperature ensemble max" ;
temp_ft_ens_max:units = "degC" ;
temp_ft_ens_max:standard_name = "air_temperature" ;
double pblh_ft_ens_mean(time, trajectory_time) ;
pblh_ft_ens_mean:_FillValue = -9999. ;
pblh_ft_ens_mean:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column ensemble mean" ;
pblh_ft_ens_mean:units = "m" ;
pblh_ft_ens_mean:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ft_ens_std(time, trajectory_time) ;
pblh_ft_ens_std:_FillValue = -9999. ;
pblh_ft_ens_std:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column ensemble std" ;
pblh_ft_ens_std:units = "m" ;
pblh_ft_ens_std:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ft_ens_min(time, trajectory_time) ;
pblh_ft_ens_min:_FillValue = -9999. ;
pblh_ft_ens_min:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column ensemble min" ;
pblh_ft_ens_min:units = "m" ;
pblh_ft_ens_min:standard_name = "atmosphere_boundary_layer_thickness" ;

double pblh_ft_ens_max(time, trajectory_time) ;
pblh_ft_ens_max:_FillValue = -9999. ;
pblh_ft_ens_max:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column ensemble max" ;
pblh_ft_ens_max:units = "m" ;
pblh_ft_ens_max:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh_ft_ens_mean(time, trajectory_time) ;
rh_ft_ens_mean:_FillValue = -9999. ;
rh_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass relative humidity ensemble mean" ;
rh_ft_ens_mean:units = "%" ;
rh_ft_ens_mean:standard_name = "relative_humidity" ;
double rh_ft_ens_std(time, trajectory_time) ;
rh_ft_ens_std:_FillValue = -9999. ;
rh_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass relative humidity ensemble std" ;
rh_ft_ens_std:units = "%" ;
rh_ft_ens_std:standard_name = "relative_humidity" ;
double rh_ft_ens_min(time, trajectory_time) ;
rh_ft_ens_min:_FillValue = -9999. ;
rh_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass relative humidity ensemble min" ;
rh_ft_ens_min:units = "%" ;
rh_ft_ens_min:standard_name = "relative_humidity" ;
double rh_ft_ens_max(time, trajectory_time) ;
rh_ft_ens_max:_FillValue = -9999. ;
rh_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass relative humidity ensemble max" ;
rh_ft_ens_max:units = "%" ;
rh_ft_ens_max:standard_name = "relative_humidity" ;
double surf_altitude_ft_ens_mean(time, trajectory_time) ;
surf_altitude_ft_ens_mean:_FillValue = -9999. ;
surf_altitude_ft_ens_mean:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column ensemble mean" ;
surf_altitude_ft_ens_mean:units = "m" ;
surf_altitude_ft_ens_mean:standard_name = "surface_altitude" ;
double surf_altitude_ft_ens_std(time, trajectory_time) ;
surf_altitude_ft_ens_std:_FillValue = -9999. ;
surf_altitude_ft_ens_std:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column ensemble std" ;
surf_altitude_ft_ens_std:units = "m" ;
surf_altitude_ft_ens_std:standard_name = "surface_altitude" ;
double surf_altitude_ft_ens_min(time, trajectory_time) ;
surf_altitude_ft_ens_min:_FillValue = -9999. ;
surf_altitude_ft_ens_min:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column ensemble min" ;

surf_altitude_ft_ens_min:units = "m" ;
surf_altitude_ft_ens_min:standard_name = "surface_altitude" ;
double surf_altitude_ft_ens_max(time, trajectory_time) ;
surf_altitude_ft_ens_max:_FillValue = -9999. ;
surf_altitude_ft_ens_max:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column ensemble max" ;
surf_altitude_ft_ens_max:units = "m" ;
surf_altitude_ft_ens_max:standard_name = "surface_altitude" ;
double qv_ft_ens_mean(time, trajectory_time) ;
qv_ft_ens_mean:_FillValue = -9999. ;
qv_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass specific humidity ensemble mean" ;
qv_ft_ens_mean:units = "g/kg" ;
qv_ft_ens_mean:standard_name = "specific_humidity" ;
double qv_ft_ens_std(time, trajectory_time) ;
qv_ft_ens_std:_FillValue = -9999. ;
qv_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass specific humidity ensemble std" ;
qv_ft_ens_std:units = "g/kg" ;
qv_ft_ens_std:standard_name = "specific_humidity" ;
double qv_ft_ens_min(time, trajectory_time) ;
qv_ft_ens_min:_FillValue = -9999. ;
qv_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass specific humidity ensemble min" ;
qv_ft_ens_min:units = "g/kg" ;
qv_ft_ens_min:standard_name = "specific_humidity" ;
double qv_ft_ens_max(time, trajectory_time) ;
qv_ft_ens_max:_FillValue = -9999. ;
qv_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass specific humidity ensemble max" ;
qv_ft_ens_max:units = "g/kg" ;
qv_ft_ens_max:standard_name = "specific_humidity" ;
double rhi_ft_ens_mean(time, trajectory_time) ;
rhi_ft_ens_mean:_FillValue = -9999. ;
rhi_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice ensemble mean" ;
rhi_ft_ens_mean:units = "%" ;
double rhi_ft_ens_std(time, trajectory_time) ;
rhi_ft_ens_std:_FillValue = -9999. ;
rhi_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice ensemble std" ;
rhi_ft_ens_std:units = "%" ;
double rhi_ft_ens_min(time, trajectory_time) ;
rhi_ft_ens_min:_FillValue = -9999. ;
rhi_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice ensemble min" ;

rhi_ft_ens_min:units = "%" ;
double rhi_ft_ens_max(time, trajectory_time) ;
rhi_ft_ens_max:_FillValue = -9999. ;
rhi_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice ensemble max" ;
rhi_ft_ens_max:units = "%" ;
double theta_e_ft_ens_mean(time, trajectory_time) ;
theta_e_ft_ens_mean:_FillValue = -9999. ;
theta_e_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble mean" ;
theta_e_ft_ens_mean:units = "K" ;
theta_e_ft_ens_mean:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ft_ens_std(time, trajectory_time) ;
theta_e_ft_ens_std:_FillValue = -9999. ;
theta_e_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble std" ;
theta_e_ft_ens_std:units = "K" ;
theta_e_ft_ens_std:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ft_ens_min(time, trajectory_time) ;
theta_e_ft_ens_min:_FillValue = -9999. ;
theta_e_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble min" ;
theta_e_ft_ens_min:units = "K" ;
theta_e_ft_ens_min:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ft_ens_max(time, trajectory_time) ;
theta_e_ft_ens_max:_FillValue = -9999. ;
theta_e_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble max" ;
theta_e_ft_ens_max:units = "K" ;
theta_e_ft_ens_max:standard_name = "air_equivalent_potential_temperature" ;
double theta_v_ft_ens_mean(time, trajectory_time) ;
theta_v_ft_ens_mean:_FillValue = -9999. ;
theta_v_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature ensemble mean" ;
theta_v_ft_ens_mean:units = "K" ;
double theta_v_ft_ens_std(time, trajectory_time) ;
theta_v_ft_ens_std:_FillValue = -9999. ;
theta_v_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature ensemble std" ;
theta_v_ft_ens_std:units = "K" ;
double theta_v_ft_ens_min(time, trajectory_time) ;
theta_v_ft_ens_min:_FillValue = -9999. ;
theta_v_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature ensemble min" ;
theta_v_ft_ens_min:units = "K" ;
double theta_v_ft_ens_max(time, trajectory_time) ;

theta_v_ft_ens_max:_FillValue = -9999. ;
theta_v_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature ensemble max" ;
theta_v_ft_ens_max:units = "K" ;
double altitude_ft_ens_mean(time, trajectory_time) ;
altitude_ft_ens_mean:_FillValue = -9999. ;
altitude_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level ensemble mean" ;
altitude_ft_ens_mean:units = "m" ;
altitude_ft_ens_mean:standard_name = "height_above_mean_sea_level" ;
double altitude_ft_ens_std(time, trajectory_time) ;
altitude_ft_ens_std:_FillValue = -9999. ;
altitude_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level ensemble std" ;
altitude_ft_ens_std:units = "m" ;
altitude_ft_ens_std:standard_name = "height_above_mean_sea_level" ;
double altitude_ft_ens_min(time, trajectory_time) ;
altitude_ft_ens_min:_FillValue = -9999. ;
altitude_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level ensemble min" ;
altitude_ft_ens_min:units = "m" ;
altitude_ft_ens_min:standard_name = "height_above_mean_sea_level" ;
double altitude_ft_ens_max(time, trajectory_time) ;
altitude_ft_ens_max:_FillValue = -9999. ;
altitude_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level ensemble max" ;
altitude_ft_ens_max:units = "m" ;
altitude_ft_ens_max:standard_name = "height_above_mean_sea_level" ;
double wvert_ft_ens_mean(time, trajectory_time) ;
wvert_ft_ens_mean:_FillValue = -9999. ;
wvert_ft_ens_mean:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate ensemble mean" ;
wvert_ft_ens_mean:units = "m/s" ;
wvert_ft_ens_mean:standard_name = "upward_air_velocity" ;
double wvert_ft_ens_std(time, trajectory_time) ;
wvert_ft_ens_std:_FillValue = -9999. ;
wvert_ft_ens_std:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate ensemble std" ;
wvert_ft_ens_std:units = "m/s" ;
wvert_ft_ens_std:standard_name = "upward_air_velocity" ;
double wvert_ft_ens_min(time, trajectory_time) ;
wvert_ft_ens_min:_FillValue = -9999. ;
wvert_ft_ens_min:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate ensemble min" ;
wvert_ft_ens_min:units = "m/s" ;
wvert_ft_ens_min:standard_name = "upward_air_velocity" ;

double wvert_ft_ens_max(time, trajectory_time) ;
wvert_ft_ens_max:_FillValue = -9999. ;
wvert_ft_ens_max:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate ensemble max" ;
wvert_ft_ens_max:units = "m/s" ;
wvert_ft_ens_max:standard_name = "upward_air_velocity" ;
double height_to_pblh_ratio_ft_ens_mean(time, trajectory_time) ;
height_to_pblh_ratio_ft_ens_mean:_FillValue = -9999. ;
height_to_pblh_ratio_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble mean" ;
height_to_pblh_ratio_ft_ens_mean:units = "1" ;
double height_to_pblh_ratio_ft_ens_std(time, trajectory_time) ;
height_to_pblh_ratio_ft_ens_std:_FillValue = -9999. ;
height_to_pblh_ratio_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble std" ;
height_to_pblh_ratio_ft_ens_std:units = "1" ;
double height_to_pblh_ratio_ft_ens_min(time, trajectory_time) ;
height_to_pblh_ratio_ft_ens_min:_FillValue = -9999. ;
height_to_pblh_ratio_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble min" ;
height_to_pblh_ratio_ft_ens_min:units = "1" ;
double height_to_pblh_ratio_ft_ens_max(time, trajectory_time) ;
height_to_pblh_ratio_ft_ens_max:_FillValue = -9999. ;
height_to_pblh_ratio_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble max" ;
height_to_pblh_ratio_ft_ens_max:units = "1" ;
double sgs_orography_angle(time, trajectory_time, vert_layer) ;
sgs_orography_angle:_FillValue = -9999. ;
sgs_orography_angle:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_angle:units = "radian" ;
double sgs_orography_angle_ens_mean(time, trajectory_time) ;
sgs_orography_angle_ens_mean:_FillValue = -9999. ;
sgs_orography_angle_ens_mean:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_angle_ens_mean:units = "radian" ;
double sgs_orography_angle_ft(time, trajectory_time) ;
sgs_orography_angle_ft:_FillValue = -9999. ;
sgs_orography_angle_ft:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere trajectory)" ;
sgs_orography_angle_ft:units = "radian" ;
double sgs_orography_angle_ft_ens_mean(time, trajectory_time) ;
sgs_orography_angle_ft_ens_mean:_FillValue = -9999. ;

sgs_orography_angle_ft_ens_mean:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere ensemble mean trajectory)" ;

sgs_orography_angle_ft_ens_mean:units = "radian" ;

double sgs_orography_anisotropy(time, trajectory_time, vert_layer) ;

sgs_orography_anisotropy:_FillValue = -9999. ;

sgs_orography_anisotropy:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;

sgs_orography_anisotropy:units = "1" ;

double sgs_orography_anisotropy_ens_mean(time, trajectory_time) ;

sgs_orography_anisotropy_ens_mean:_FillValue = -9999. ;

sgs_orography_anisotropy_ens_mean:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;

sgs_orography_anisotropy_ens_mean:units = "1" ;

double sgs_orography_anisotropy_ft(time, trajectory_time) ;

sgs_orography_anisotropy_ft:_FillValue = -9999. ;

sgs_orography_anisotropy_ft:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere trajectory)" ;

sgs_orography_anisotropy_ft:units = "1" ;

double sgs_orography_anisotropy_ft_ens_mean(time, trajectory_time) ;

sgs_orography_anisotropy_ft_ens_mean:_FillValue = -9999. ;

sgs_orography_anisotropy_ft_ens_mean:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere ensemble mean trajectory)" ;

sgs_orography_anisotropy_ft_ens_mean:units = "1" ;

double sgs_orography_std(time, trajectory_time, vert_layer) ;

sgs_orography_std:_FillValue = -9999. ;

sgs_orography_std:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;

sgs_orography_std:units = "m" ;

double sgs_orography_std_ens_mean(time, trajectory_time) ;

sgs_orography_std_ens_mean:_FillValue = -9999. ;

sgs_orography_std_ens_mean:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;

sgs_orography_std_ens_mean:units = "m" ;

double sgs_orography_std_ft(time, trajectory_time) ;

sgs_orography_std_ft:_FillValue = -9999. ;

sgs_orography_std_ft:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere trajectory)" ;

sgs_orography_std_ft:units = "m" ;

double sgs_orography_std_ft_ens_mean(time, trajectory_time) ;
sgs_orography_std_ft_ens_mean:_FillValue = -9999. ;
sgs_orography_std_ft_ens_mean:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere ensemble mean trajectory)" ;
sgs_orography_std_ft_ens_mean:units = "m" ;
double sgs_orography_slope(time, trajectory_time, vert_layer) ;
sgs_orography_slope:_FillValue = -9999. ;
sgs_orography_slope:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_slope:units = "1" ;
double sgs_orography_slope_ens_mean(time, trajectory_time) ;
sgs_orography_slope_ens_mean:_FillValue = -9999. ;
sgs_orography_slope_ens_mean:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_slope_ens_mean:units = "1" ;
double sgs_orography_slope_ft(time, trajectory_time) ;
sgs_orography_slope_ft:_FillValue = -9999. ;
sgs_orography_slope_ft:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere trajectory)" ;
sgs_orography_slope_ft:units = "1" ;
double sgs_orography_slope_ft_ens_mean(time, trajectory_time) ;
sgs_orography_slope_ft_ens_mean:_FillValue = -9999. ;
sgs_orography_slope_ft_ens_mean:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere ensemble mean trajectory)" ;
sgs_orography_slope_ft_ens_mean:units = "1" ;
double land_sea_mask(time, trajectory_time, vert_layer) ;
land_sea_mask:_FillValue = -9999. ;
land_sea_mask:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land)" ;
land_sea_mask:units = "1" ;
double land_sea_mask_ens_mean(time, trajectory_time) ;
land_sea_mask_ens_mean:_FillValue = -9999. ;
land_sea_mask_ens_mean:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along ensemble mean trajectory)" ;
land_sea_mask_ens_mean:units = "1" ;
double land_sea_mask_ft(time, trajectory_time) ;
land_sea_mask_ft:_FillValue = -9999. ;
land_sea_mask_ft:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along free-troposphere trajectory)" ;
land_sea_mask_ft:units = "1" ;
double land_sea_mask_ft_ens_mean(time, trajectory_time) ;

land_sea_mask_ft_ens_mean:_FillValue = -9999. ;
land_sea_mask_ft_ens_mean:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along free-troposphere ensemble mean trajectory)" ;
land_sea_mask_ft_ens_mean:units = "1" ;
float high_vegetation_type(time, trajectory_time, vert_layer) ;
high_vegetation_type:_FillValue = -9999.f ;
high_vegetation_type:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation)" ;
high_vegetation_type:units = "1" ;
float high_vegetation_type_ens_mean(time, trajectory_time) ;
high_vegetation_type_ens_mean:_FillValue = -9999.f ;
high_vegetation_type_ens_mean:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along ensemble mean trajectory)" ;
high_vegetation_type_ens_mean:units = "1" ;
float high_vegetation_type_ft(time, trajectory_time) ;
high_vegetation_type_ft:_FillValue = -9999.f ;
high_vegetation_type_ft:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along free-troposphere trajectory)" ;
high_vegetation_type_ft:units = "1" ;
float high_vegetation_type_ft_ens_mean(time, trajectory_time) ;
high_vegetation_type_ft_ens_mean:_FillValue = -9999.f ;
high_vegetation_type_ft_ens_mean:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along free-troposphere ensemble mean trajectory)" ;
high_vegetation_type_ft_ens_mean:units = "1" ;
float low_vegetation_type(time, trajectory_time, vert_layer) ;
low_vegetation_type:_FillValue = -9999.f ;
low_vegetation_type:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation)" ;
low_vegetation_type:units = "1" ;
float low_vegetation_type_ens_mean(time, trajectory_time) ;
low_vegetation_type_ens_mean:_FillValue = -9999.f ;
low_vegetation_type_ens_mean:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along ensemble mean trajectory)" ;

low_vegetation_type_ens_mean:units = "1" ;
float low_vegetation_type_ft(time, trajectory_time) ;
low_vegetation_type_ft:_FillValue = -9999.f ;
low_vegetation_type_ft:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along free-troposphere trajectory)" ;
low_vegetation_type_ft:units = "1" ;
float low_vegetation_type_ft_ens_mean(time, trajectory_time) ;
low_vegetation_type_ft_ens_mean:_FillValue = -9999.f ;
low_vegetation_type_ft_ens_mean:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along free-troposphere ensemble mean trajectory)" ;
low_vegetation_type_ft_ens_mean:units = "1" ;
double high_vegetation_cover(time, trajectory_time, vert_layer) ;
high_vegetation_cover:_FillValue = -9999. ;
high_vegetation_cover:long_name = "High vegetation cover fraction in airmass column" ;
high_vegetation_cover:units = "1" ;
double high_vegetation_cover_ens_mean(time, trajectory_time) ;
high_vegetation_cover_ens_mean:_FillValue = -9999. ;
high_vegetation_cover_ens_mean:long_name = "High vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
high_vegetation_cover_ens_mean:units = "1" ;
double high_vegetation_cover_ft(time, trajectory_time) ;
high_vegetation_cover_ft:_FillValue = -9999. ;
high_vegetation_cover_ft:long_name = "High vegetation cover fraction in airmass column (along free-troposphere trajectory)" ;
high_vegetation_cover_ft:units = "1" ;
double high_vegetation_cover_ft_ens_mean(time, trajectory_time) ;
high_vegetation_cover_ft_ens_mean:_FillValue = -9999. ;
high_vegetation_cover_ft_ens_mean:long_name = "High vegetation cover fraction in airmass column (along free-troposphere ensemble mean trajectory)" ;
high_vegetation_cover_ft_ens_mean:units = "1" ;
double low_vegetation_cover(time, trajectory_time, vert_layer) ;
low_vegetation_cover:_FillValue = -9999. ;
low_vegetation_cover:long_name = "Low vegetation cover fraction in airmass column" ;
low_vegetation_cover:units = "1" ;
double low_vegetation_cover_ens_mean(time, trajectory_time) ;
low_vegetation_cover_ens_mean:_FillValue = -9999. ;
low_vegetation_cover_ens_mean:long_name = "Low vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
low_vegetation_cover_ens_mean:units = "1" ;
double low_vegetation_cover_ft(time, trajectory_time) ;
low_vegetation_cover_ft:_FillValue = -9999. ;

low_vegetation_cover_ft:long_name = "Low vegetation cover fraction in airmass column (along free-troposphere trajectory)" ;
low_vegetation_cover_ft:units = "1" ;
double low_vegetation_cover_ft_ens_mean(time, trajectory_time) ;
low_vegetation_cover_ft_ens_mean:_FillValue = -9999. ;
low_vegetation_cover_ft_ens_mean:long_name = "Low vegetation cover fraction in airmass column (along free-troposphere ensemble mean trajectory)" ;
low_vegetation_cover_ft_ens_mean:units = "1" ;
float soil_type(time, trajectory_time, vert_layer) ;
soil_type:_FillValue = -9999.f ;
soil_type:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point)" ;
soil_type:units = "1" ;
soil_type:standard_name = "soil_type" ;
float soil_type_ens_mean(time, trajectory_time) ;
soil_type_ens_mean:_FillValue = -9999.f ;
soil_type_ens_mean:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along ensemble mean trajectory)" ;
soil_type_ens_mean:units = "1" ;
soil_type_ens_mean:standard_name = "soil_type" ;
float soil_type_ft(time, trajectory_time) ;
soil_type_ft:_FillValue = -9999.f ;
soil_type_ft:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along free-troposphere trajectory)" ;
soil_type_ft:units = "1" ;
soil_type_ft:standard_name = "soil_type" ;
float soil_type_ft_ens_mean(time, trajectory_time) ;
soil_type_ft_ens_mean:_FillValue = -9999.f ;
soil_type_ft_ens_mean:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along free-troposphere ensemble mean trajectory)" ;
soil_type_ft_ens_mean:units = "1" ;
soil_type_ft_ens_mean:standard_name = "soil_type" ;
double sea_ice_cover(time, trajectory_time, vert_layer) ;
sea_ice_cover:_FillValue = -9999. ;
sea_ice_cover:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column" ;
sea_ice_cover:units = "1" ;
sea_ice_cover:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ens_mean(time, trajectory_time) ;
sea_ice_cover_ens_mean:_FillValue = -9999. ;
sea_ice_cover_ens_mean:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along ensemble mean trajectory)" ;

sea_ice_cover_ens_mean:units = "1" ;
sea_ice_cover_ens_mean:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ft(time, trajectory_time) ;
sea_ice_cover_ft:_FillValue = -9999. ;
sea_ice_cover_ft:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along free-troposphere trajectory)" ;
sea_ice_cover_ft:units = "1" ;
sea_ice_cover_ft:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ft_ens_mean(time, trajectory_time) ;
sea_ice_cover_ft_ens_mean:_FillValue = -9999. ;
sea_ice_cover_ft_ens_mean:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along free-troposphere ensemble mean trajectory)" ;
sea_ice_cover_ft_ens_mean:units = "1" ;
sea_ice_cover_ft_ens_mean:standard_name = "sea_ice_area_fraction" ;
float pbl_height_heffter(time) ;
pbl_height_heffter:_FillValue = -9999.f ;
pbl_height_heffter:long_name = "Planetary boundary layer height above mean sea level calculated using the Heffter (1980) method" ;
pbl_height_heffter:units = "m" ;
pbl_height_heffter:standard_name = "atmosphere_boundary_layer_thickness" ;
pbl_height_heffter:valid_min = 13.1f ;
pbl_height_heffter:valid_max = 4013.1f ;
pbl_height_heffter:comment = "valid_min is site elevation; valid_max is site elevation plus 4000 m" ;
double pbl_regime_type_liu_liang(time) ;
pbl_regime_type_liu_liang:_FillValue = -9999. ;
pbl_regime_type_liu_liang:long_name = "Planetary boundary layer regime type determined by Liu and Liang (2010) method" ;
pbl_regime_type_liu_liang:units = "1" ;
pbl_regime_type_liu_liang:flag_0_description = "Neutral boundary layer" ;
pbl_regime_type_liu_liang:flag_1_description = "Stable boundary layer" ;
pbl_regime_type_liu_liang:flag_2_description = "Convective boundary layer" ;
float pbl_height_liu_liang(time) ;
pbl_height_liu_liang:_FillValue = -9999.f ;
pbl_height_liu_liang:long_name = "Planetary boundary layer height above mean sea level calculated by Liu and Liang (2010) method" ;
pbl_height_liu_liang:units = "m" ;
pbl_height_liu_liang:standard_name = "atmosphere_boundary_layer_thickness" ;
pbl_height_liu_liang:valid_min = 13.1f ;
pbl_height_liu_liang:valid_max = 4013.1f ;
pbl_height_liu_liang:comment = "valid_min is site elevation; valid_max is site elevation plus 4000 m" ;
float pbl_height_bulk_richardson_pt25(time) ;
pbl_height_bulk_richardson_pt25:_FillValue = -9999.f ;
pbl_height_bulk_richardson_pt25:long_name = "Planetary boundary layer height above mean sea level calculated from bulk Richardson number using critical threshold of 0.25" ;

```
                pbl_height_bulk_richardson_pt25:units = "m" ;
                pbl_height_bulk_richardson_pt25:standard_name =
"atmosphere_boundary_layer_thickness" ;
                pbl_height_bulk_richardson_pt25:valid_min = 13.1f ;
                pbl_height_bulk_richardson_pt25:valid_max = 4013.1f ;
                pbl_height_bulk_richardson_pt25:comment = "valid_min is site elevation; valid_max is
site elevation plus 4000 m" ;
        float pbl_height_bulk_richardson_pt5(time) ;
                pbl_height_bulk_richardson_pt5:_FillValue = -9999.f ;
                pbl_height_bulk_richardson_pt5:long_name = "Planetary boundary layer height above
mean sea level calculated from bulk Richardson number using critical threshold of 0.5" ;
                pbl_height_bulk_richardson_pt5:units = "m" ;
                pbl_height_bulk_richardson_pt5:standard_name =
"atmosphere_boundary_layer_thickness" ;
                pbl_height_bulk_richardson_pt5:valid_min = 13.1f ;
                pbl_height_bulk_richardson_pt5:valid_max = 4013.1f ;
                pbl_height_bulk_richardson_pt5:comment = "valid_min is site elevation; valid_max is
site elevation plus 4000 m" ;
        double time(time) ;
                time:long_name = "Time in seconds since volume start" ;
                time:units = "seconds since 2023-08-20 00:00:00 0:00" ;
        double time_offset(time) ;
                time_offset:_FillValue = -9999. ;
                time_offset:long_name = "Time offset from base_time" ;
                time_offset:units = "seconds since 2023-08-20 00:00:00 0:00" ;
        int base_time ;
                base_time:_FillValue = -9999 ;
                base_time:long_name = "Base time in Epoch" ;
                base_time:units = "seconds since 1970-01-01 0:00:00 0:00" ;
        double lat ;
                lat:_FillValue = -9999. ;
                lat:long_name = "latitude" ;
                lat:units = "degree_N" ;
        double lon ;
                lon:_FillValue = -9999. ;
                lon:long_name = "longitude" ;
                lon:units = "degree_E" ;
        double alt ;
                alt:_FillValue = -9999. ;
                alt:long_name = "Altitude above mean sea level" ;
                alt:units = "m" ;

// global attributes:
                :mode = "pbl" ;
                :arm_pblh_algorithm = "pbl_height_bulk_richardson_pt25" ;
                :ensemble_size = "99" ;
```

:ensemble_size_FT = "9" ;
:ensemble_dx_from_center = "[-7.5 0.   7.5] km" ;
:ensemble_dy_from_center = "[-7.5 0.   7.5] km" ;
:ensemble_dz_from_center = "[0] m" ;
:used_era5_resolution = "0.25 degrees" ;
:total_trajectory_hours = "[-120]" ;
:number_of_starting_center_points_per_time_step = "[11 11 11 11]" ;
:title = "Back trajectories for PBL and related aerosol and cloud studies" ;
:summary = "The ARMTRAJ VAP provides trajectory datasets initialized at ARM deployment coordinates and configured using ARM datasets. The trajectory datasets support aerosol, cloud, and planetary boundary layer research. Trajectory calculations use the HYSPLIT model informed by the ERA5 reanalysis dataset at its highest spatial resolution (~31 km). For each sample in each of the datasets, HYSPLIT also runs at multiple starting locations surrounding ARM deployments, enabling an ensemble of runs from which the mean and variability (estimated uncertainty) of each sample\'s trajectory coordinates, thermodynamic properties, or other fields are reported." ;
:keywords = "ARM, ground-based measurements, ground-based remote sensing, clouds, aerosol, PBL, trajectories, Lagrangian analysis" ;
:doi = "10.5439/2309848" ;
:location_description = "Eastern Pacific Cloud Aerosol Precipitation Experiment (EPCAPE), Scripps Pier, La Jolla, CA" ;
:dod_version = "v1.0" ;
:command_line = "python armtraj_run.py" ;
:datastream = "epcarmtrajpblM1.c1" ;
:data_level = "c1" ;
:facility_id = "M1" ;
:site_id = "epc" ;
:platform_id = "armtrajpbl" ;
:history = "created by user isilber1 on machine dev-proc2 at 27-Sep-2024,15:26:15" ;
}

Appendix D

ARMTRAJ-ARSCL Output Data

```
netcdf epcarmtrajarsclM1.c1.20230820.000000 {
dimensions:
        time = UNLIMITED ; // (8 currently)
        vert_layer = 3 ;
        trajectory_time = 121 ;
        trajectory_type = 2 ;
        layer = 10 ;
        range = 596 ;
variables:
        int vert_layer(vert_layer) ;
                vert_layer:long_name = "Vertical layer index (bottom-up) (0 - cloud deck base, 5 - cloud
deck mid, 10 - cloud deck top)" ;
                vert_layer:units = "1" ;
        int trajectory_time(trajectory_time) ;
                trajectory_time:long_name = "Trajectory offset from start time (0 h)" ;
                trajectory_time:units = "h" ;
        int trajectory_type(trajectory_type) ;
                trajectory_type:long_name = "-1 - back trajectories, 1 - fwd trajectories" ;
                trajectory_type:units = "1" ;
        double date(time, trajectory_type, trajectory_time) ;
                date:_FillValue = -9999. ;
                date:long_name = "Trajectory date per trajectory type and time step" ;
                date:units = "seconds since 1970-01-01" ;
        double latitude(time, trajectory_type, trajectory_time, vert_layer) ;
                latitude:_FillValue = -9999. ;
                latitude:long_name = "Airmass latitude" ;
                latitude:units = "degree_N" ;
                latitude:standard_name = "grid_latitude" ;
        double longitude(time, trajectory_type, trajectory_time, vert_layer) ;
                longitude:_FillValue = -9999. ;
                longitude:long_name = "Airmass longitude" ;
                longitude:units = "degree_E" ;
                longitude:standard_name = "grid_longitude" ;
        double height(time, trajectory_type, trajectory_time, vert_layer) ;
```

height:_FillValue = -9999. ;
height:long_name = "Airmass height above ground level" ;
height:units = "m" ;
height:standard_name = "height" ;
double pres(time, trajectory_type, trajectory_time, vert_layer) ;
pres:_FillValue = -9999. ;
pres:long_name = "Airmass pressure" ;
pres:units = "hPa" ;
pres:standard_name = "air_pressure" ;
double theta(time, trajectory_type, trajectory_time, vert_layer) ;
theta:_FillValue = -9999. ;
theta:long_name = "Airmass potential temperature" ;
theta:units = "K" ;
theta:standard_name = "air_potential_temperature" ;
double temp(time, trajectory_type, trajectory_time, vert_layer) ;
temp:_FillValue = -9999. ;
temp:long_name = "Airmass temperature" ;
temp:units = "degC" ;
temp:standard_name = "air_temperature" ;
double pblh(time, trajectory_type, trajectory_time, vert_layer) ;
pblh:_FillValue = -9999. ;
pblh:long_name = "Planetary boundary layer height in airmass column" ;
pblh:units = "m" ;
pblh:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh(time, trajectory_type, trajectory_time, vert_layer) ;
rh:_FillValue = -9999. ;
rh:long_name = "Airmass relative humidity" ;
rh:units = "%" ;
rh:standard_name = "relative_humidity" ;
double surf_altitude(time, trajectory_type, trajectory_time, vert_layer) ;
surf_altitude:_FillValue = -9999. ;
surf_altitude:long_name = "Surface altitude above mean sea level in airmass column" ;
surf_altitude:units = "m" ;
surf_altitude:standard_name = "surface_altitude" ;
double qv(time, trajectory_type, trajectory_time, vert_layer) ;
qv:_FillValue = -9999. ;
qv:long_name = "Airmass specific humidity" ;
qv:units = "g/kg" ;
qv:standard_name = "specific_humidity" ;
double rhi(time, trajectory_type, trajectory_time, vert_layer) ;
rhi:_FillValue = -9999. ;
rhi:long_name = "Airmass relative humidity with respect to ice" ;
rhi:units = "%" ;
double theta_e(time, trajectory_type, trajectory_time, vert_layer) ;
theta_e:_FillValue = -9999. ;

theta_e:long_name = "Airmass equivalent potential temperature (excluding condensate from the calculation)" ;
theta_e:units = "K" ;
theta_e:standard_name = "air_equivalent_potential_temperature" ;
double theta_v(time, trajectory_type, trajectory_time, vert_layer) ;
theta_v:_FillValue = -9999. ;
theta_v:long_name = "Airmass virtual potential temperature" ;
theta_v:units = "K" ;
double altitude(time, trajectory_type, trajectory_time, vert_layer) ;
altitude:_FillValue = -9999. ;
altitude:long_name = "Airmass altitude above mean sea level" ;
altitude:units = "m" ;
altitude:standard_name = "height_above_mean_sea_level" ;
double wvert(time, trajectory_type, trajectory_time, vert_layer) ;
wvert:_FillValue = -9999. ;
wvert:long_name = "Mean hourly airmass ascent rate" ;
wvert:units = "m/s" ;
wvert:standard_name = "upward_air_velocity" ;
double height_to_pblh_ratio(time, trajectory_type, trajectory_time, vert_layer) ;
height_to_pblh_ratio:_FillValue = -9999. ;
height_to_pblh_ratio:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL)" ;
height_to_pblh_ratio:units = "1" ;
double latitude_ft(time, trajectory_type, trajectory_time) ;
latitude_ft:_FillValue = -9999. ;
latitude_ft:long_name = "Free-troposphere (at initialization) airmass latitude" ;
latitude_ft:units = "degree_N" ;
latitude_ft:standard_name = "grid_latitude" ;
double longitude_ft(time, trajectory_type, trajectory_time) ;
longitude_ft:_FillValue = -9999. ;
longitude_ft:long_name = "Free-troposphere (at initialization) airmass longitude" ;
longitude_ft:units = "degree_E" ;
longitude_ft:standard_name = "grid_longitude" ;
double height_ft(time, trajectory_type, trajectory_time) ;
height_ft:_FillValue = -9999. ;
height_ft:long_name = "Free-troposphere (at initialization) airmass height above ground level" ;
height_ft:units = "m" ;
height_ft:standard_name = "height" ;
double pres_ft(time, trajectory_type, trajectory_time) ;
pres_ft:_FillValue = -9999. ;
pres_ft:long_name = "Free-troposphere (at initialization) airmass pressure" ;
pres_ft:units = "hPa" ;
pres_ft:standard_name = "air_pressure" ;
double theta_ft(time, trajectory_type, trajectory_time) ;
theta_ft:_FillValue = -9999. ;

theta_ft:long_name = "Free-troposphere (at initialization) airmass potential temperature" ;
theta_ft:units = "K" ;
theta_ft:standard_name = "air_potential_temperature" ;
double temp_ft(time, trajectory_type, trajectory_time) ;
temp_ft:_FillValue = -9999. ;
temp_ft:long_name = "Free-troposphere (at initialization) airmass temperature" ;
temp_ft:units = "degC" ;
temp_ft:standard_name = "air_temperature" ;
double pblh_ft(time, trajectory_type, trajectory_time) ;
pblh_ft:_FillValue = -9999. ;
pblh_ft:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column" ;
pblh_ft:units = "m" ;
pblh_ft:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh_ft(time, trajectory_type, trajectory_time) ;
rh_ft:_FillValue = -9999. ;
rh_ft:long_name = "Free-troposphere (at initialization) airmass relative humidity" ;
rh_ft:units = "%" ;
rh_ft:standard_name = "relative_humidity" ;
double surf_altitude_ft(time, trajectory_type, trajectory_time) ;
surf_altitude_ft:_FillValue = -9999. ;
surf_altitude_ft:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column" ;
surf_altitude_ft:units = "m" ;
surf_altitude_ft:standard_name = "surface_altitude" ;
double qv_ft(time, trajectory_type, trajectory_time) ;
qv_ft:_FillValue = -9999. ;
qv_ft:long_name = "Free-troposphere (at initialization) airmass specific humidity" ;
qv_ft:units = "g/kg" ;
qv_ft:standard_name = "specific_humidity" ;
double rhi_ft(time, trajectory_type, trajectory_time) ;
rhi_ft:_FillValue = -9999. ;
rhi_ft:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice" ;
rhi_ft:units = "%" ;
double theta_e_ft(time, trajectory_type, trajectory_time) ;
theta_e_ft:_FillValue = -9999. ;
theta_e_ft:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation)" ;
theta_e_ft:units = "K" ;
theta_e_ft:standard_name = "air_equivalent_potential_temperature" ;
double theta_v_ft(time, trajectory_type, trajectory_time) ;
theta_v_ft:_FillValue = -9999. ;
theta_v_ft:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature" ;

```
                theta_v_ft:units = "K" ;
        double altitude_ft(time, trajectory_type, trajectory_time) ;
                altitude_ft:_FillValue = -9999. ;
                altitude_ft:long_name = "Free-troposphere (at initialization) airmass altitude above mean
sea level" ;
                altitude_ft:units = "m" ;
                altitude_ft:standard_name = "height_above_mean_sea_level" ;
        double wvert_ft(time, trajectory_type, trajectory_time) ;
                wvert_ft:_FillValue = -9999. ;
                wvert_ft:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent
rate" ;
                wvert_ft:units = "m/s" ;
                wvert_ft:standard_name = "upward_air_velocity" ;
        double height_to_pblh_ratio_ft(time, trajectory_type, trajectory_time) ;
                height_to_pblh_ratio_ft:_FillValue = -9999. ;
                height_to_pblh_ratio_ft:long_name = "Free-troposphere (at initialization) airmass height-
to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl)" ;
                height_to_pblh_ratio_ft:units = "1" ;
        double latitude_ens_mean(time, trajectory_type, trajectory_time) ;
                latitude_ens_mean:_FillValue = -9999. ;
                latitude_ens_mean:long_name = "Airmass latitude ensemble mean" ;
                latitude_ens_mean:units = "degree_N" ;
                latitude_ens_mean:standard_name = "grid_latitude" ;
        double latitude_ens_std(time, trajectory_type, trajectory_time) ;
                latitude_ens_std:_FillValue = -9999. ;
                latitude_ens_std:long_name = "Airmass latitude ensemble std" ;
                latitude_ens_std:units = "degree_N" ;
                latitude_ens_std:standard_name = "grid_latitude" ;
        double latitude_ens_min(time, trajectory_type, trajectory_time) ;
                latitude_ens_min:_FillValue = -9999. ;
                latitude_ens_min:long_name = "Airmass latitude ensemble min" ;
                latitude_ens_min:units = "degree_N" ;
                latitude_ens_min:standard_name = "grid_latitude" ;
        double latitude_ens_max(time, trajectory_type, trajectory_time) ;
                latitude_ens_max:_FillValue = -9999. ;
                latitude_ens_max:long_name = "Airmass latitude ensemble max" ;
                latitude_ens_max:units = "degree_N" ;
                latitude_ens_max:standard_name = "grid_latitude" ;
        double longitude_ens_mean(time, trajectory_type, trajectory_time) ;
                longitude_ens_mean:_FillValue = -9999. ;
                longitude_ens_mean:long_name = "Airmass longitude ensemble mean" ;
                longitude_ens_mean:units = "degree_E" ;
                longitude_ens_mean:standard_name = "grid_longitude" ;
        double longitude_ens_std(time, trajectory_type, trajectory_time) ;
                longitude_ens_std:_FillValue = -9999. ;
                longitude_ens_std:long_name = "Airmass longitude ensemble std" ;
```

longitude_ens_std:units = "degree_E" ;
longitude_ens_std:standard_name = "grid_longitude" ;
double longitude_ens_min(time, trajectory_type, trajectory_time) ;
longitude_ens_min:_FillValue = -9999. ;
longitude_ens_min:long_name = "Airmass longitude ensemble min" ;
longitude_ens_min:units = "degree_E" ;
longitude_ens_min:standard_name = "grid_longitude" ;
double longitude_ens_max(time, trajectory_type, trajectory_time) ;
longitude_ens_max:_FillValue = -9999. ;
longitude_ens_max:long_name = "Airmass longitude ensemble max" ;
longitude_ens_max:units = "degree_E" ;
longitude_ens_max:standard_name = "grid_longitude" ;
double height_ens_mean(time, trajectory_type, trajectory_time) ;
height_ens_mean:_FillValue = -9999. ;
height_ens_mean:long_name = "Airmass height above ground level ensemble mean" ;
height_ens_mean:units = "m" ;
height_ens_mean:standard_name = "height" ;
double height_ens_std(time, trajectory_type, trajectory_time) ;
height_ens_std:_FillValue = -9999. ;
height_ens_std:long_name = "Airmass height above ground level ensemble std" ;
height_ens_std:units = "m" ;
height_ens_std:standard_name = "height" ;
double height_ens_min(time, trajectory_type, trajectory_time) ;
height_ens_min:_FillValue = -9999. ;
height_ens_min:long_name = "Airmass height above ground level ensemble min" ;
height_ens_min:units = "m" ;
height_ens_min:standard_name = "height" ;
double height_ens_max(time, trajectory_type, trajectory_time) ;
height_ens_max:_FillValue = -9999. ;
height_ens_max:long_name = "Airmass height above ground level ensemble max" ;
height_ens_max:units = "m" ;
height_ens_max:standard_name = "height" ;
double pres_ens_mean(time, trajectory_type, trajectory_time) ;
pres_ens_mean:_FillValue = -9999. ;
pres_ens_mean:long_name = "Airmass pressure ensemble mean" ;
pres_ens_mean:units = "hPa" ;
pres_ens_mean:standard_name = "air_pressure" ;
double pres_ens_std(time, trajectory_type, trajectory_time) ;
pres_ens_std:_FillValue = -9999. ;
pres_ens_std:long_name = "Airmass pressure ensemble std" ;
pres_ens_std:units = "hPa" ;
pres_ens_std:standard_name = "air_pressure" ;
double pres_ens_min(time, trajectory_type, trajectory_time) ;
pres_ens_min:_FillValue = -9999. ;
pres_ens_min:long_name = "Airmass pressure ensemble min" ;
pres_ens_min:units = "hPa" ;

```
        pres_ens_min:standard_name = "air_pressure" ;
double pres_ens_max(time, trajectory_type, trajectory_time) ;
        pres_ens_max:_FillValue = -9999. ;
        pres_ens_max:long_name = "Airmass pressure ensemble max" ;
        pres_ens_max:units = "hPa" ;
        pres_ens_max:standard_name = "air_pressure" ;
double theta_ens_mean(time, trajectory_type, trajectory_time) ;
        theta_ens_mean:_FillValue = -9999. ;
        theta_ens_mean:long_name = "Airmass potential temperature ensemble mean" ;
        theta_ens_mean:units = "K" ;
        theta_ens_mean:standard_name = "air_potential_temperature" ;
double theta_ens_std(time, trajectory_type, trajectory_time) ;
        theta_ens_std:_FillValue = -9999. ;
        theta_ens_std:long_name = "Airmass potential temperature ensemble std" ;
        theta_ens_std:units = "K" ;
        theta_ens_std:standard_name = "air_potential_temperature" ;
double theta_ens_min(time, trajectory_type, trajectory_time) ;
        theta_ens_min:_FillValue = -9999. ;
        theta_ens_min:long_name = "Airmass potential temperature ensemble min" ;
        theta_ens_min:units = "K" ;
        theta_ens_min:standard_name = "air_potential_temperature" ;
double theta_ens_max(time, trajectory_type, trajectory_time) ;
        theta_ens_max:_FillValue = -9999. ;
        theta_ens_max:long_name = "Airmass potential temperature ensemble max" ;
        theta_ens_max:units = "K" ;
        theta_ens_max:standard_name = "air_potential_temperature" ;
double temp_ens_mean(time, trajectory_type, trajectory_time) ;
        temp_ens_mean:_FillValue = -9999. ;
        temp_ens_mean:long_name = "Airmass temperature ensemble mean" ;
        temp_ens_mean:units = "degC" ;
        temp_ens_mean:standard_name = "air_temperature" ;
double temp_ens_std(time, trajectory_type, trajectory_time) ;
        temp_ens_std:_FillValue = -9999. ;
        temp_ens_std:long_name = "Airmass temperature ensemble std" ;
        temp_ens_std:units = "degC" ;
        temp_ens_std:standard_name = "air_temperature" ;
double temp_ens_min(time, trajectory_type, trajectory_time) ;
        temp_ens_min:_FillValue = -9999. ;
        temp_ens_min:long_name = "Airmass temperature ensemble min" ;
        temp_ens_min:units = "degC" ;
        temp_ens_min:standard_name = "air_temperature" ;
double temp_ens_max(time, trajectory_type, trajectory_time) ;
        temp_ens_max:_FillValue = -9999. ;
        temp_ens_max:long_name = "Airmass temperature ensemble max" ;
        temp_ens_max:units = "degC" ;
        temp_ens_max:standard_name = "air_temperature" ;
```

```
        double pblh_ens_mean(time, trajectory_type, trajectory_time) ;
                pblh_ens_mean:_FillValue = -9999. ;
                pblh_ens_mean:long_name = "Planetary boundary layer height in airmass column
ensemble mean" ;
                pblh_ens_mean:units = "m" ;
                pblh_ens_mean:standard_name = "atmosphere_boundary_layer_thickness" ;
        double pblh_ens_std(time, trajectory_type, trajectory_time) ;
                pblh_ens_std:_FillValue = -9999. ;
                pblh_ens_std:long_name = "Planetary boundary layer height in airmass column ensemble
std" ;
                pblh_ens_std:units = "m" ;
                pblh_ens_std:standard_name = "atmosphere_boundary_layer_thickness" ;
        double pblh_ens_min(time, trajectory_type, trajectory_time) ;
                pblh_ens_min:_FillValue = -9999. ;
                pblh_ens_min:long_name = "Planetary boundary layer height in airmass column
ensemble min" ;
                pblh_ens_min:units = "m" ;
                pblh_ens_min:standard_name = "atmosphere_boundary_layer_thickness" ;
        double pblh_ens_max(time, trajectory_type, trajectory_time) ;
                pblh_ens_max:_FillValue = -9999. ;
                pblh_ens_max:long_name = "Planetary boundary layer height in airmass column
ensemble max" ;
                pblh_ens_max:units = "m" ;
                pblh_ens_max:standard_name = "atmosphere_boundary_layer_thickness" ;
        double rh_ens_mean(time, trajectory_type, trajectory_time) ;
                rh_ens_mean:_FillValue = -9999. ;
                rh_ens_mean:long_name = "Airmass relative humidity ensemble mean" ;
                rh_ens_mean:units = "%" ;
                rh_ens_mean:standard_name = "relative_humidity" ;
        double rh_ens_std(time, trajectory_type, trajectory_time) ;
                rh_ens_std:_FillValue = -9999. ;
                rh_ens_std:long_name = "Airmass relative humidity ensemble std" ;
                rh_ens_std:units = "%" ;
                rh_ens_std:standard_name = "relative_humidity" ;
        double rh_ens_min(time, trajectory_type, trajectory_time) ;
                rh_ens_min:_FillValue = -9999. ;
                rh_ens_min:long_name = "Airmass relative humidity ensemble min" ;
                rh_ens_min:units = "%" ;
                rh_ens_min:standard_name = "relative_humidity" ;
        double rh_ens_max(time, trajectory_type, trajectory_time) ;
                rh_ens_max:_FillValue = -9999. ;
                rh_ens_max:long_name = "Airmass relative humidity ensemble max" ;
                rh_ens_max:units = "%" ;
                rh_ens_max:standard_name = "relative_humidity" ;
        double surf_altitude_ens_mean(time, trajectory_type, trajectory_time) ;
                surf_altitude_ens_mean:_FillValue = -9999. ;
```

surf_altitude_ens_mean:long_name = "Surface altitude above mean sea level in airmass column ensemble mean" ;
surf_altitude_ens_mean:units = "m" ;
surf_altitude_ens_mean:standard_name = "surface_altitude" ;
double surf_altitude_ens_std(time, trajectory_type, trajectory_time) ;
surf_altitude_ens_std:_FillValue = -9999. ;
surf_altitude_ens_std:long_name = "Surface altitude above mean sea level in airmass column ensemble std" ;
surf_altitude_ens_std:units = "m" ;
surf_altitude_ens_std:standard_name = "surface_altitude" ;
double surf_altitude_ens_min(time, trajectory_type, trajectory_time) ;
surf_altitude_ens_min:_FillValue = -9999. ;
surf_altitude_ens_min:long_name = "Surface altitude above mean sea level in airmass column ensemble min" ;
surf_altitude_ens_min:units = "m" ;
surf_altitude_ens_min:standard_name = "surface_altitude" ;
double surf_altitude_ens_max(time, trajectory_type, trajectory_time) ;
surf_altitude_ens_max:_FillValue = -9999. ;
surf_altitude_ens_max:long_name = "Surface altitude above mean sea level in airmass column ensemble max" ;
surf_altitude_ens_max:units = "m" ;
surf_altitude_ens_max:standard_name = "surface_altitude" ;
double qv_ens_mean(time, trajectory_type, trajectory_time) ;
qv_ens_mean:_FillValue = -9999. ;
qv_ens_mean:long_name = "Airmass specific humidity ensemble mean" ;
qv_ens_mean:units = "g/kg" ;
qv_ens_mean:standard_name = "specific_humidity" ;
double qv_ens_std(time, trajectory_type, trajectory_time) ;
qv_ens_std:_FillValue = -9999. ;
qv_ens_std:long_name = "Airmass specific humidity ensemble std" ;
qv_ens_std:units = "g/kg" ;
qv_ens_std:standard_name = "specific_humidity" ;
double qv_ens_min(time, trajectory_type, trajectory_time) ;
qv_ens_min:_FillValue = -9999. ;
qv_ens_min:long_name = "Airmass specific humidity ensemble min" ;
qv_ens_min:units = "g/kg" ;
qv_ens_min:standard_name = "specific_humidity" ;
double qv_ens_max(time, trajectory_type, trajectory_time) ;
qv_ens_max:_FillValue = -9999. ;
qv_ens_max:long_name = "Airmass specific humidity ensemble max" ;
qv_ens_max:units = "g/kg" ;
qv_ens_max:standard_name = "specific_humidity" ;
double rhi_ens_mean(time, trajectory_type, trajectory_time) ;
rhi_ens_mean:_FillValue = -9999. ;
rhi_ens_mean:long_name = "Airmass relative humidity with respect to ice ensemble mean" ;

rhi_ens_mean:units = "%" ;
double rhi_ens_std(time, trajectory_type, trajectory_time) ;
rhi_ens_std:_FillValue = -9999. ;
rhi_ens_std:long_name = "Airmass relative humidity with respect to ice ensemble std" ;
rhi_ens_std:units = "%" ;
double rhi_ens_min(time, trajectory_type, trajectory_time) ;
rhi_ens_min:_FillValue = -9999. ;
rhi_ens_min:long_name = "Airmass relative humidity with respect to ice ensemble min" ;
rhi_ens_min:units = "%" ;
double rhi_ens_max(time, trajectory_type, trajectory_time) ;
rhi_ens_max:_FillValue = -9999. ;
rhi_ens_max:long_name = "Airmass relative humidity with respect to ice ensemble max"
;
rhi_ens_max:units = "%" ;
double theta_e_ens_mean(time, trajectory_type, trajectory_time) ;
theta_e_ens_mean:_FillValue = -9999. ;
theta_e_ens_mean:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble mean" ;
theta_e_ens_mean:units = "K" ;
theta_e_ens_mean:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ens_std(time, trajectory_type, trajectory_time) ;
theta_e_ens_std:_FillValue = -9999. ;
theta_e_ens_std:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble std" ;
theta_e_ens_std:units = "K" ;
theta_e_ens_std:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ens_min(time, trajectory_type, trajectory_time) ;
theta_e_ens_min:_FillValue = -9999. ;
theta_e_ens_min:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble min" ;
theta_e_ens_min:units = "K" ;
theta_e_ens_min:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ens_max(time, trajectory_type, trajectory_time) ;
theta_e_ens_max:_FillValue = -9999. ;
theta_e_ens_max:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble max" ;
theta_e_ens_max:units = "K" ;
theta_e_ens_max:standard_name = "air_equivalent_potential_temperature" ;
double theta_v_ens_mean(time, trajectory_type, trajectory_time) ;
theta_v_ens_mean:_FillValue = -9999. ;
theta_v_ens_mean:long_name = "Airmass virtual potential temperature ensemble mean" ;
theta_v_ens_mean:units = "K" ;
double theta_v_ens_std(time, trajectory_type, trajectory_time) ;
theta_v_ens_std:_FillValue = -9999. ;
theta_v_ens_std:long_name = "Airmass virtual potential temperature ensemble std" ;
theta_v_ens_std:units = "K" ;

```
        double theta_v_ens_min(time, trajectory_type, trajectory_time) ;
                theta_v_ens_min:_FillValue = -9999. ;
                theta_v_ens_min:long_name = "Airmass virtual potential temperature ensemble min" ;
                theta_v_ens_min:units = "K" ;
        double theta_v_ens_max(time, trajectory_type, trajectory_time) ;
                theta_v_ens_max:_FillValue = -9999. ;
                theta_v_ens_max:long_name = "Airmass virtual potential temperature ensemble max" ;
                theta_v_ens_max:units = "K" ;
        double altitude_ens_mean(time, trajectory_type, trajectory_time) ;
                altitude_ens_mean:_FillValue = -9999. ;
                altitude_ens_mean:long_name = "Airmass altitude above mean sea level ensemble mean"
;
                altitude_ens_mean:units = "m" ;
                altitude_ens_mean:standard_name = "height_above_mean_sea_level" ;
        double altitude_ens_std(time, trajectory_type, trajectory_time) ;
                altitude_ens_std:_FillValue = -9999. ;
                altitude_ens_std:long_name = "Airmass altitude above mean sea level ensemble std" ;
                altitude_ens_std:units = "m" ;
                altitude_ens_std:standard_name = "height_above_mean_sea_level" ;
        double altitude_ens_min(time, trajectory_type, trajectory_time) ;
                altitude_ens_min:_FillValue = -9999. ;
                altitude_ens_min:long_name = "Airmass altitude above mean sea level ensemble min" ;
                altitude_ens_min:units = "m" ;
                altitude_ens_min:standard_name = "height_above_mean_sea_level" ;
        double altitude_ens_max(time, trajectory_type, trajectory_time) ;
                altitude_ens_max:_FillValue = -9999. ;
                altitude_ens_max:long_name = "Airmass altitude above mean sea level ensemble max" ;
                altitude_ens_max:units = "m" ;
                altitude_ens_max:standard_name = "height_above_mean_sea_level" ;
        double wvert_ens_mean(time, trajectory_type, trajectory_time) ;
                wvert_ens_mean:_FillValue = -9999. ;
                wvert_ens_mean:long_name = "Mean hourly airmass ascent rate ensemble mean" ;
                wvert_ens_mean:units = "m/s" ;
                wvert_ens_mean:standard_name = "upward_air_velocity" ;
        double wvert_ens_std(time, trajectory_type, trajectory_time) ;
                wvert_ens_std:_FillValue = -9999. ;
                wvert_ens_std:long_name = "Mean hourly airmass ascent rate ensemble std" ;
                wvert_ens_std:units = "m/s" ;
                wvert_ens_std:standard_name = "upward_air_velocity" ;
        double wvert_ens_min(time, trajectory_type, trajectory_time) ;
                wvert_ens_min:_FillValue = -9999. ;
                wvert_ens_min:long_name = "Mean hourly airmass ascent rate ensemble min" ;
                wvert_ens_min:units = "m/s" ;
                wvert_ens_min:standard_name = "upward_air_velocity" ;
        double wvert_ens_max(time, trajectory_type, trajectory_time) ;
                wvert_ens_max:_FillValue = -9999. ;
```

wvert_ens_max:long_name = "Mean hourly airmass ascent rate ensemble max" ;
wvert_ens_max:units = "m/s" ;
wvert_ens_max:standard_name = "upward_air_velocity" ;
double height_to_pblh_ratio_ens_mean(time, trajectory_type, trajectory_time) ;
height_to_pblh_ratio_ens_mean:_FillValue = -9999. ;
height_to_pblh_ratio_ens_mean:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL) ensemble mean" ;
height_to_pblh_ratio_ens_mean:units = "1" ;
double height_to_pblh_ratio_ens_std(time, trajectory_type, trajectory_time) ;
height_to_pblh_ratio_ens_std:_FillValue = -9999. ;
height_to_pblh_ratio_ens_std:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL) ensemble std" ;
height_to_pblh_ratio_ens_std:units = "1" ;
double height_to_pblh_ratio_ens_min(time, trajectory_type, trajectory_time) ;
height_to_pblh_ratio_ens_min:_FillValue = -9999. ;
height_to_pblh_ratio_ens_min:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL) ensemble min" ;
height_to_pblh_ratio_ens_min:units = "1" ;
double height_to_pblh_ratio_ens_max(time, trajectory_type, trajectory_time) ;
height_to_pblh_ratio_ens_max:_FillValue = -9999. ;
height_to_pblh_ratio_ens_max:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above PBL; <1 - airmass within PBL) ensemble max" ;
height_to_pblh_ratio_ens_max:units = "1" ;
double latitude_ft_ens_mean(time, trajectory_type, trajectory_time) ;
latitude_ft_ens_mean:_FillValue = -9999. ;
latitude_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass latitude ensemble mean" ;
latitude_ft_ens_mean:units = "degree_N" ;
latitude_ft_ens_mean:standard_name = "grid_latitude" ;
double latitude_ft_ens_std(time, trajectory_type, trajectory_time) ;
latitude_ft_ens_std:_FillValue = -9999. ;
latitude_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass latitude ensemble std" ;
latitude_ft_ens_std:units = "degree_N" ;
latitude_ft_ens_std:standard_name = "grid_latitude" ;
double latitude_ft_ens_min(time, trajectory_type, trajectory_time) ;
latitude_ft_ens_min:_FillValue = -9999. ;
latitude_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass latitude ensemble min" ;
latitude_ft_ens_min:units = "degree_N" ;
latitude_ft_ens_min:standard_name = "grid_latitude" ;
double latitude_ft_ens_max(time, trajectory_type, trajectory_time) ;
latitude_ft_ens_max:_FillValue = -9999. ;
latitude_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass latitude ensemble max" ;
latitude_ft_ens_max:units = "degree_N" ;

latitude_ft_ens_max:standard_name = "grid_latitude" ;
double longitude_ft_ens_mean(time, trajectory_type, trajectory_time) ;
longitude_ft_ens_mean:_FillValue = -9999. ;
longitude_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass longitude ensemble mean" ;
longitude_ft_ens_mean:units = "degree_E" ;
longitude_ft_ens_mean:standard_name = "grid_longitude" ;
double longitude_ft_ens_std(time, trajectory_type, trajectory_time) ;
longitude_ft_ens_std:_FillValue = -9999. ;
longitude_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass longitude ensemble std" ;
longitude_ft_ens_std:units = "degree_E" ;
longitude_ft_ens_std:standard_name = "grid_longitude" ;
double longitude_ft_ens_min(time, trajectory_type, trajectory_time) ;
longitude_ft_ens_min:_FillValue = -9999. ;
longitude_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass longitude ensemble min" ;
longitude_ft_ens_min:units = "degree_E" ;
longitude_ft_ens_min:standard_name = "grid_longitude" ;
double longitude_ft_ens_max(time, trajectory_type, trajectory_time) ;
longitude_ft_ens_max:_FillValue = -9999. ;
longitude_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass longitude ensemble max" ;
longitude_ft_ens_max:units = "degree_E" ;
longitude_ft_ens_max:standard_name = "grid_longitude" ;
double height_ft_ens_mean(time, trajectory_type, trajectory_time) ;
height_ft_ens_mean:_FillValue = -9999. ;
height_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass height above ground level ensemble mean" ;
height_ft_ens_mean:units = "m" ;
height_ft_ens_mean:standard_name = "height" ;
double height_ft_ens_std(time, trajectory_type, trajectory_time) ;
height_ft_ens_std:_FillValue = -9999. ;
height_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass height above ground level ensemble std" ;
height_ft_ens_std:units = "m" ;
height_ft_ens_std:standard_name = "height" ;
double height_ft_ens_min(time, trajectory_type, trajectory_time) ;
height_ft_ens_min:_FillValue = -9999. ;
height_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass height above ground level ensemble min" ;
height_ft_ens_min:units = "m" ;
height_ft_ens_min:standard_name = "height" ;
double height_ft_ens_max(time, trajectory_type, trajectory_time) ;
height_ft_ens_max:_FillValue = -9999. ;

height_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass height above ground level ensemble max" ;
height_ft_ens_max:units = "m" ;
height_ft_ens_max:standard_name = "height" ;
double pres_ft_ens_mean(time, trajectory_type, trajectory_time) ;
pres_ft_ens_mean:_FillValue = -9999. ;
pres_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass pressure ensemble mean" ;
pres_ft_ens_mean:units = "hPa" ;
pres_ft_ens_mean:standard_name = "air_pressure" ;
double pres_ft_ens_std(time, trajectory_type, trajectory_time) ;
pres_ft_ens_std:_FillValue = -9999. ;
pres_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass pressure ensemble std" ;
pres_ft_ens_std:units = "hPa" ;
pres_ft_ens_std:standard_name = "air_pressure" ;
double pres_ft_ens_min(time, trajectory_type, trajectory_time) ;
pres_ft_ens_min:_FillValue = -9999. ;
pres_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass pressure ensemble min" ;
pres_ft_ens_min:units = "hPa" ;
pres_ft_ens_min:standard_name = "air_pressure" ;
double pres_ft_ens_max(time, trajectory_type, trajectory_time) ;
pres_ft_ens_max:_FillValue = -9999. ;
pres_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass pressure ensemble max" ;
pres_ft_ens_max:units = "hPa" ;
pres_ft_ens_max:standard_name = "air_pressure" ;
double theta_ft_ens_mean(time, trajectory_type, trajectory_time) ;
theta_ft_ens_mean:_FillValue = -9999. ;
theta_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass potential temperature ensemble mean" ;
theta_ft_ens_mean:units = "K" ;
theta_ft_ens_mean:standard_name = "air_potential_temperature" ;
double theta_ft_ens_std(time, trajectory_type, trajectory_time) ;
theta_ft_ens_std:_FillValue = -9999. ;
theta_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass potential temperature ensemble std" ;
theta_ft_ens_std:units = "K" ;
theta_ft_ens_std:standard_name = "air_potential_temperature" ;
double theta_ft_ens_min(time, trajectory_type, trajectory_time) ;
theta_ft_ens_min:_FillValue = -9999. ;
theta_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass potential temperature ensemble min" ;
theta_ft_ens_min:units = "K" ;
theta_ft_ens_min:standard_name = "air_potential_temperature" ;

double theta_ft_ens_max(time, trajectory_type, trajectory_time) ;
theta_ft_ens_max:_FillValue = -9999. ;
theta_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass potential temperature ensemble max" ;
theta_ft_ens_max:units = "K" ;
theta_ft_ens_max:standard_name = "air_potential_temperature" ;
double temp_ft_ens_mean(time, trajectory_type, trajectory_time) ;
temp_ft_ens_mean:_FillValue = -9999. ;
temp_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass temperature ensemble mean" ;
temp_ft_ens_mean:units = "degC" ;
temp_ft_ens_mean:standard_name = "air_temperature" ;
double temp_ft_ens_std(time, trajectory_type, trajectory_time) ;
temp_ft_ens_std:_FillValue = -9999. ;
temp_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass temperature ensemble std" ;
temp_ft_ens_std:units = "degC" ;
temp_ft_ens_std:standard_name = "air_temperature" ;
double temp_ft_ens_min(time, trajectory_type, trajectory_time) ;
temp_ft_ens_min:_FillValue = -9999. ;
temp_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass temperature ensemble min" ;
temp_ft_ens_min:units = "degC" ;
temp_ft_ens_min:standard_name = "air_temperature" ;
double temp_ft_ens_max(time, trajectory_type, trajectory_time) ;
temp_ft_ens_max:_FillValue = -9999. ;
temp_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass temperature ensemble max" ;
temp_ft_ens_max:units = "degC" ;
temp_ft_ens_max:standard_name = "air_temperature" ;
double pblh_ft_ens_mean(time, trajectory_type, trajectory_time) ;
pblh_ft_ens_mean:_FillValue = -9999. ;
pblh_ft_ens_mean:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column ensemble mean" ;
pblh_ft_ens_mean:units = "m" ;
pblh_ft_ens_mean:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ft_ens_std(time, trajectory_type, trajectory_time) ;
pblh_ft_ens_std:_FillValue = -9999. ;
pblh_ft_ens_std:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column ensemble std" ;
pblh_ft_ens_std:units = "m" ;
pblh_ft_ens_std:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ft_ens_min(time, trajectory_type, trajectory_time) ;
pblh_ft_ens_min:_FillValue = -9999. ;
pblh_ft_ens_min:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column ensemble min" ;

pblh_ft_ens_min:units = "m" ;
pblh_ft_ens_min:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ft_ens_max(time, trajectory_type, trajectory_time) ;
pblh_ft_ens_max:_FillValue = -9999. ;
pblh_ft_ens_max:long_name = "Free-troposphere (at initialization) planetary boundary layer height in airmass column ensemble max" ;
pblh_ft_ens_max:units = "m" ;
pblh_ft_ens_max:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh_ft_ens_mean(time, trajectory_type, trajectory_time) ;
rh_ft_ens_mean:_FillValue = -9999. ;
rh_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass relative humidity ensemble mean" ;
rh_ft_ens_mean:units = "%" ;
rh_ft_ens_mean:standard_name = "relative_humidity" ;
double rh_ft_ens_std(time, trajectory_type, trajectory_time) ;
rh_ft_ens_std:_FillValue = -9999. ;
rh_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass relative humidity ensemble std" ;
rh_ft_ens_std:units = "%" ;
rh_ft_ens_std:standard_name = "relative_humidity" ;
double rh_ft_ens_min(time, trajectory_type, trajectory_time) ;
rh_ft_ens_min:_FillValue = -9999. ;
rh_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass relative humidity ensemble min" ;
rh_ft_ens_min:units = "%" ;
rh_ft_ens_min:standard_name = "relative_humidity" ;
double rh_ft_ens_max(time, trajectory_type, trajectory_time) ;
rh_ft_ens_max:_FillValue = -9999. ;
rh_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass relative humidity ensemble max" ;
rh_ft_ens_max:units = "%" ;
rh_ft_ens_max:standard_name = "relative_humidity" ;
double surf_altitude_ft_ens_mean(time, trajectory_type, trajectory_time) ;
surf_altitude_ft_ens_mean:_FillValue = -9999. ;
surf_altitude_ft_ens_mean:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column ensemble mean" ;
surf_altitude_ft_ens_mean:units = "m" ;
surf_altitude_ft_ens_mean:standard_name = "surface_altitude" ;
double surf_altitude_ft_ens_std(time, trajectory_type, trajectory_time) ;
surf_altitude_ft_ens_std:_FillValue = -9999. ;
surf_altitude_ft_ens_std:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column ensemble std" ;
surf_altitude_ft_ens_std:units = "m" ;
surf_altitude_ft_ens_std:standard_name = "surface_altitude" ;
double surf_altitude_ft_ens_min(time, trajectory_type, trajectory_time) ;
surf_altitude_ft_ens_min:_FillValue = -9999. ;

surf_altitude_ft_ens_min:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column ensemble min" ;
surf_altitude_ft_ens_min:units = "m" ;
surf_altitude_ft_ens_min:standard_name = "surface_altitude" ;
double surf_altitude_ft_ens_max(time, trajectory_type, trajectory_time) ;
surf_altitude_ft_ens_max:_FillValue = -9999. ;
surf_altitude_ft_ens_max:long_name = "Free-troposphere (at initialization) surface altitude above mean sea level in airmass column ensemble max" ;
surf_altitude_ft_ens_max:units = "m" ;
surf_altitude_ft_ens_max:standard_name = "surface_altitude" ;
double qv_ft_ens_mean(time, trajectory_type, trajectory_time) ;
qv_ft_ens_mean:_FillValue = -9999. ;
qv_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass specific humidity ensemble mean" ;
qv_ft_ens_mean:units = "g/kg" ;
qv_ft_ens_mean:standard_name = "specific_humidity" ;
double qv_ft_ens_std(time, trajectory_type, trajectory_time) ;
qv_ft_ens_std:_FillValue = -9999. ;
qv_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass specific humidity ensemble std" ;
qv_ft_ens_std:units = "g/kg" ;
qv_ft_ens_std:standard_name = "specific_humidity" ;
double qv_ft_ens_min(time, trajectory_type, trajectory_time) ;
qv_ft_ens_min:_FillValue = -9999. ;
qv_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass specific humidity ensemble min" ;
qv_ft_ens_min:units = "g/kg" ;
qv_ft_ens_min:standard_name = "specific_humidity" ;
double qv_ft_ens_max(time, trajectory_type, trajectory_time) ;
qv_ft_ens_max:_FillValue = -9999. ;
qv_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass specific humidity ensemble max" ;
qv_ft_ens_max:units = "g/kg" ;
qv_ft_ens_max:standard_name = "specific_humidity" ;
double rhi_ft_ens_mean(time, trajectory_type, trajectory_time) ;
rhi_ft_ens_mean:_FillValue = -9999. ;
rhi_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice ensemble mean" ;
rhi_ft_ens_mean:units = "%" ;
double rhi_ft_ens_std(time, trajectory_type, trajectory_time) ;
rhi_ft_ens_std:_FillValue = -9999. ;
rhi_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice ensemble std" ;
rhi_ft_ens_std:units = "%" ;
double rhi_ft_ens_min(time, trajectory_type, trajectory_time) ;
rhi_ft_ens_min:_FillValue = -9999. ;

rhi_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice ensemble min" ;
rhi_ft_ens_min:units = "%" ;
double rhi_ft_ens_max(time, trajectory_type, trajectory_time) ;
rhi_ft_ens_max:_FillValue = -9999. ;
rhi_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass relative humidity with respect to ice ensemble max" ;
rhi_ft_ens_max:units = "%" ;
double theta_e_ft_ens_mean(time, trajectory_type, trajectory_time) ;
theta_e_ft_ens_mean:_FillValue = -9999. ;
theta_e_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble mean" ;
theta_e_ft_ens_mean:units = "K" ;
theta_e_ft_ens_mean:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ft_ens_std(time, trajectory_type, trajectory_time) ;
theta_e_ft_ens_std:_FillValue = -9999. ;
theta_e_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble std" ;
theta_e_ft_ens_std:units = "K" ;
theta_e_ft_ens_std:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ft_ens_min(time, trajectory_type, trajectory_time) ;
theta_e_ft_ens_min:_FillValue = -9999. ;
theta_e_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble min" ;
theta_e_ft_ens_min:units = "K" ;
theta_e_ft_ens_min:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_ft_ens_max(time, trajectory_type, trajectory_time) ;
theta_e_ft_ens_max:_FillValue = -9999. ;
theta_e_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble max" ;
theta_e_ft_ens_max:units = "K" ;
theta_e_ft_ens_max:standard_name = "air_equivalent_potential_temperature" ;
double theta_v_ft_ens_mean(time, trajectory_type, trajectory_time) ;
theta_v_ft_ens_mean:_FillValue = -9999. ;
theta_v_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature ensemble mean" ;
theta_v_ft_ens_mean:units = "K" ;
double theta_v_ft_ens_std(time, trajectory_type, trajectory_time) ;
theta_v_ft_ens_std:_FillValue = -9999. ;
theta_v_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature ensemble std" ;
theta_v_ft_ens_std:units = "K" ;
double theta_v_ft_ens_min(time, trajectory_type, trajectory_time) ;
theta_v_ft_ens_min:_FillValue = -9999. ;
theta_v_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature ensemble min" ;

theta_v_ft_ens_min:units = "K" ;
double theta_v_ft_ens_max(time, trajectory_type, trajectory_time) ;
theta_v_ft_ens_max:_FillValue = -9999. ;
theta_v_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass virtual potential temperature ensemble max" ;
theta_v_ft_ens_max:units = "K" ;
double altitude_ft_ens_mean(time, trajectory_type, trajectory_time) ;
altitude_ft_ens_mean:_FillValue = -9999. ;
altitude_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level ensemble mean" ;
altitude_ft_ens_mean:units = "m" ;
altitude_ft_ens_mean:standard_name = "height_above_mean_sea_level" ;
double altitude_ft_ens_std(time, trajectory_type, trajectory_time) ;
altitude_ft_ens_std:_FillValue = -9999. ;
altitude_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level ensemble std" ;
altitude_ft_ens_std:units = "m" ;
altitude_ft_ens_std:standard_name = "height_above_mean_sea_level" ;
double altitude_ft_ens_min(time, trajectory_type, trajectory_time) ;
altitude_ft_ens_min:_FillValue = -9999. ;
altitude_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level ensemble min" ;
altitude_ft_ens_min:units = "m" ;
altitude_ft_ens_min:standard_name = "height_above_mean_sea_level" ;
double altitude_ft_ens_max(time, trajectory_type, trajectory_time) ;
altitude_ft_ens_max:_FillValue = -9999. ;
altitude_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass altitude above mean sea level ensemble max" ;
altitude_ft_ens_max:units = "m" ;
altitude_ft_ens_max:standard_name = "height_above_mean_sea_level" ;
double wvert_ft_ens_mean(time, trajectory_type, trajectory_time) ;
wvert_ft_ens_mean:_FillValue = -9999. ;
wvert_ft_ens_mean:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate ensemble mean" ;
wvert_ft_ens_mean:units = "m/s" ;
wvert_ft_ens_mean:standard_name = "upward_air_velocity" ;
double wvert_ft_ens_std(time, trajectory_type, trajectory_time) ;
wvert_ft_ens_std:_FillValue = -9999. ;
wvert_ft_ens_std:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate ensemble std" ;
wvert_ft_ens_std:units = "m/s" ;
wvert_ft_ens_std:standard_name = "upward_air_velocity" ;
double wvert_ft_ens_min(time, trajectory_type, trajectory_time) ;
wvert_ft_ens_min:_FillValue = -9999. ;
wvert_ft_ens_min:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate ensemble min" ;

wvert_ft_ens_min:units = "m/s" ;
wvert_ft_ens_min:standard_name = "upward_air_velocity" ;
double wvert_ft_ens_max(time, trajectory_type, trajectory_time) ;
wvert_ft_ens_max:_FillValue = -9999. ;
wvert_ft_ens_max:long_name = "Free-troposphere (at initialization) mean hourly airmass ascent rate ensemble max" ;
wvert_ft_ens_max:units = "m/s" ;
wvert_ft_ens_max:standard_name = "upward_air_velocity" ;
double height_to_pblh_ratio_ft_ens_mean(time, trajectory_type, trajectory_time) ;
height_to_pblh_ratio_ft_ens_mean:_FillValue = -9999. ;
height_to_pblh_ratio_ft_ens_mean:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble mean" ;
height_to_pblh_ratio_ft_ens_mean:units = "1" ;
double height_to_pblh_ratio_ft_ens_std(time, trajectory_type, trajectory_time) ;
height_to_pblh_ratio_ft_ens_std:_FillValue = -9999. ;
height_to_pblh_ratio_ft_ens_std:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble std" ;
height_to_pblh_ratio_ft_ens_std:units = "1" ;
double height_to_pblh_ratio_ft_ens_min(time, trajectory_type, trajectory_time) ;
height_to_pblh_ratio_ft_ens_min:_FillValue = -9999. ;
height_to_pblh_ratio_ft_ens_min:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble min" ;
height_to_pblh_ratio_ft_ens_min:units = "1" ;
double height_to_pblh_ratio_ft_ens_max(time, trajectory_type, trajectory_time) ;
height_to_pblh_ratio_ft_ens_max:_FillValue = -9999. ;
height_to_pblh_ratio_ft_ens_max:long_name = "Free-troposphere (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble max" ;
height_to_pblh_ratio_ft_ens_max:units = "1" ;
double sgs_orography_angle(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_angle:_FillValue = -9999. ;
sgs_orography_angle:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_angle:units = "radian" ;
double sgs_orography_angle_ens_mean(time, trajectory_type, trajectory_time) ;
sgs_orography_angle_ens_mean:_FillValue = -9999. ;
sgs_orography_angle_ens_mean:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_angle_ens_mean:units = "radian" ;
double sgs_orography_angle_ft(time, trajectory_type, trajectory_time) ;
sgs_orography_angle_ft:_FillValue = -9999. ;
sgs_orography_angle_ft:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere trajectory)" ;
sgs_orography_angle_ft:units = "radian" ;

double sgs_orography_angle_ft_ens_mean(time, trajectory_type, trajectory_time) ;
sgs_orography_angle_ft_ens_mean:_FillValue = -9999. ;
sgs_orography_angle_ft_ens_mean:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere ensemble mean trajectory)" ;
sgs_orography_angle_ft_ens_mean:units = "radian" ;
double sgs_orography_anisotropy(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_anisotropy:_FillValue = -9999. ;
sgs_orography_anisotropy:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_anisotropy:units = "1" ;
double sgs_orography_anisotropy_ens_mean(time, trajectory_type, trajectory_time) ;
sgs_orography_anisotropy_ens_mean:_FillValue = -9999. ;
sgs_orography_anisotropy_ens_mean:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_anisotropy_ens_mean:units = "1" ;
double sgs_orography_anisotropy_ft(time, trajectory_type, trajectory_time) ;
sgs_orography_anisotropy_ft:_FillValue = -9999. ;
sgs_orography_anisotropy_ft:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere trajectory)" ;
sgs_orography_anisotropy_ft:units = "1" ;
double sgs_orography_anisotropy_ft_ens_mean(time, trajectory_type, trajectory_time) ;
sgs_orography_anisotropy_ft_ens_mean:_FillValue = -9999. ;
sgs_orography_anisotropy_ft_ens_mean:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere ensemble mean trajectory)" ;
sgs_orography_anisotropy_ft_ens_mean:units = "1" ;
double sgs_orography_std(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_std:_FillValue = -9999. ;
sgs_orography_std:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_std:units = "m" ;
double sgs_orography_std_ens_mean(time, trajectory_type, trajectory_time) ;
sgs_orography_std_ens_mean:_FillValue = -9999. ;
sgs_orography_std_ens_mean:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_std_ens_mean:units = "m" ;
double sgs_orography_std_ft(time, trajectory_type, trajectory_time) ;
sgs_orography_std_ft:_FillValue = -9999. ;

sgs_orography_std_ft:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere trajectory)" ;
sgs_orography_std_ft:units = "m" ;
double sgs_orography_std_ft_ens_mean(time, trajectory_type, trajectory_time) ;
sgs_orography_std_ft_ens_mean:_FillValue = -9999. ;
sgs_orography_std_ft_ens_mean:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere ensemble mean trajectory)" ;
sgs_orography_std_ft_ens_mean:units = "m" ;
double sgs_orography_slope(time, trajectory_type, trajectory_time, vert_layer) ;
sgs_orography_slope:_FillValue = -9999. ;
sgs_orography_slope:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_slope:units = "1" ;
double sgs_orography_slope_ens_mean(time, trajectory_type, trajectory_time) ;
sgs_orography_slope_ens_mean:_FillValue = -9999. ;
sgs_orography_slope_ens_mean:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_slope_ens_mean:units = "1" ;
double sgs_orography_slope_ft(time, trajectory_type, trajectory_time) ;
sgs_orography_slope_ft:_FillValue = -9999. ;
sgs_orography_slope_ft:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere trajectory)" ;
sgs_orography_slope_ft:units = "1" ;
double sgs_orography_slope_ft_ens_mean(time, trajectory_type, trajectory_time) ;
sgs_orography_slope_ft_ens_mean:_FillValue = -9999. ;
sgs_orography_slope_ft_ens_mean:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along free-troposphere ensemble mean trajectory)" ;
sgs_orography_slope_ft_ens_mean:units = "1" ;
double land_sea_mask(time, trajectory_type, trajectory_time, vert_layer) ;
land_sea_mask:_FillValue = -9999. ;
land_sea_mask:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land)" ;
land_sea_mask:units = "1" ;
double land_sea_mask_ens_mean(time, trajectory_type, trajectory_time) ;
land_sea_mask_ens_mean:_FillValue = -9999. ;
land_sea_mask_ens_mean:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along ensemble mean trajectory)" ;
land_sea_mask_ens_mean:units = "1" ;
double land_sea_mask_ft(time, trajectory_type, trajectory_time) ;
land_sea_mask_ft:_FillValue = -9999. ;

land_sea_mask_ft:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along free-troposphere trajectory)" ;
land_sea_mask_ft:units = "1" ;
double land_sea_mask_ft_ens_mean(time, trajectory_type, trajectory_time) ;
land_sea_mask_ft_ens_mean:_FillValue = -9999. ;
land_sea_mask_ft_ens_mean:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along free-troposphere ensemble mean trajectory)" ;
land_sea_mask_ft_ens_mean:units = "1" ;
float high_vegetation_type(time, trajectory_type, trajectory_time, vert_layer) ;
high_vegetation_type:_FillValue = -9999.f ;
high_vegetation_type:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation)" ;
high_vegetation_type:units = "1" ;
float high_vegetation_type_ens_mean(time, trajectory_type, trajectory_time) ;
high_vegetation_type_ens_mean:_FillValue = -9999.f ;
high_vegetation_type_ens_mean:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along ensemble mean trajectory)" ;
high_vegetation_type_ens_mean:units = "1" ;
float high_vegetation_type_ft(time, trajectory_type, trajectory_time) ;
high_vegetation_type_ft:_FillValue = -9999.f ;
high_vegetation_type_ft:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along free-troposphere trajectory)" ;
high_vegetation_type_ft:units = "1" ;
float high_vegetation_type_ft_ens_mean(time, trajectory_type, trajectory_time) ;
high_vegetation_type_ft_ens_mean:_FillValue = -9999.f ;
high_vegetation_type_ft_ens_mean:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along free-troposphere ensemble mean trajectory)" ;
high_vegetation_type_ft_ens_mean:units = "1" ;
float low_vegetation_type(time, trajectory_type, trajectory_time, vert_layer) ;
low_vegetation_type:_FillValue = -9999.f ;
low_vegetation_type:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation)" ;
low_vegetation_type:units = "1" ;
float low_vegetation_type_ens_mean(time, trajectory_type, trajectory_time) ;
low_vegetation_type_ens_mean:_FillValue = -9999.f ;

low_vegetation_type_ens_mean:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along ensemble mean trajectory)" ;
low_vegetation_type_ens_mean:units = "1" ;
float low_vegetation_type_ft(time, trajectory_type, trajectory_time) ;
low_vegetation_type_ft:_FillValue = -9999.f ;
low_vegetation_type_ft:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along free-troposphere trajectory)" ;
low_vegetation_type_ft:units = "1" ;
float low_vegetation_type_ft_ens_mean(time, trajectory_type, trajectory_time) ;
low_vegetation_type_ft_ens_mean:_FillValue = -9999.f ;
low_vegetation_type_ft_ens_mean:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along free-troposphere ensemble mean trajectory)" ;
low_vegetation_type_ft_ens_mean:units = "1" ;
double high_vegetation_cover(time, trajectory_type, trajectory_time, vert_layer) ;
high_vegetation_cover:_FillValue = -9999. ;
high_vegetation_cover:long_name = "High vegetation cover fraction in airmass column" ;
high_vegetation_cover:units = "1" ;
double high_vegetation_cover_ens_mean(time, trajectory_type, trajectory_time) ;
high_vegetation_cover_ens_mean:_FillValue = -9999. ;
high_vegetation_cover_ens_mean:long_name = "High vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
high_vegetation_cover_ens_mean:units = "1" ;
double high_vegetation_cover_ft(time, trajectory_type, trajectory_time) ;
high_vegetation_cover_ft:_FillValue = -9999. ;
high_vegetation_cover_ft:long_name = "High vegetation cover fraction in airmass column (along free-troposphere trajectory)" ;
high_vegetation_cover_ft:units = "1" ;
double high_vegetation_cover_ft_ens_mean(time, trajectory_type, trajectory_time) ;
high_vegetation_cover_ft_ens_mean:_FillValue = -9999. ;
high_vegetation_cover_ft_ens_mean:long_name = "High vegetation cover fraction in airmass column (along free-troposphere ensemble mean trajectory)" ;
high_vegetation_cover_ft_ens_mean:units = "1" ;
double low_vegetation_cover(time, trajectory_type, trajectory_time, vert_layer) ;
low_vegetation_cover:_FillValue = -9999. ;
low_vegetation_cover:long_name = "Low vegetation cover fraction in airmass column" ;
low_vegetation_cover:units = "1" ;
double low_vegetation_cover_ens_mean(time, trajectory_type, trajectory_time) ;
low_vegetation_cover_ens_mean:_FillValue = -9999. ;

low_vegetation_cover_ens_mean:long_name = "Low vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
low_vegetation_cover_ens_mean:units = "1" ;
double low_vegetation_cover_ft(time, trajectory_type, trajectory_time) ;
low_vegetation_cover_ft:_FillValue = -9999. ;
low_vegetation_cover_ft:long_name = "Low vegetation cover fraction in airmass column (along free-troposphere trajectory)" ;
low_vegetation_cover_ft:units = "1" ;
double low_vegetation_cover_ft_ens_mean(time, trajectory_type, trajectory_time) ;
low_vegetation_cover_ft_ens_mean:_FillValue = -9999. ;
low_vegetation_cover_ft_ens_mean:long_name = "Low vegetation cover fraction in airmass column (along free-troposphere ensemble mean trajectory)" ;
low_vegetation_cover_ft_ens_mean:units = "1" ;
float soil_type(time, trajectory_type, trajectory_time, vert_layer) ;
soil_type:_FillValue = -9999.f ;
soil_type:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point)" ;
soil_type:units = "1" ;
soil_type:standard_name = "soil_type" ;
float soil_type_ens_mean(time, trajectory_type, trajectory_time) ;
soil_type_ens_mean:_FillValue = -9999.f ;
soil_type_ens_mean:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along ensemble mean trajectory)" ;
soil_type_ens_mean:units = "1" ;
soil_type_ens_mean:standard_name = "soil_type" ;
float soil_type_ft(time, trajectory_type, trajectory_time) ;
soil_type_ft:_FillValue = -9999.f ;
soil_type_ft:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along free-troposphere trajectory)" ;
soil_type_ft:units = "1" ;
soil_type_ft:standard_name = "soil_type" ;
float soil_type_ft_ens_mean(time, trajectory_type, trajectory_time) ;
soil_type_ft_ens_mean:_FillValue = -9999.f ;
soil_type_ft_ens_mean:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along free-troposphere ensemble mean trajectory)" ;
soil_type_ft_ens_mean:units = "1" ;
soil_type_ft_ens_mean:standard_name = "soil_type" ;
double sea_ice_cover(time, trajectory_type, trajectory_time, vert_layer) ;
sea_ice_cover:_FillValue = -9999. ;
sea_ice_cover:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column" ;
sea_ice_cover:units = "1" ;

sea_ice_cover:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ens_mean(time, trajectory_type, trajectory_time) ;
sea_ice_cover_ens_mean:_FillValue = -9999. ;
sea_ice_cover_ens_mean:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along ensemble mean trajectory)" ;
sea_ice_cover_ens_mean:units = "1" ;
sea_ice_cover_ens_mean:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ft(time, trajectory_type, trajectory_time) ;
sea_ice_cover_ft:_FillValue = -9999. ;
sea_ice_cover_ft:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along free-troposphere trajectory)" ;
sea_ice_cover_ft:units = "1" ;
sea_ice_cover_ft:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ft_ens_mean(time, trajectory_type, trajectory_time) ;
sea_ice_cover_ft_ens_mean:_FillValue = -9999. ;
sea_ice_cover_ft_ens_mean:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along free-troposphere ensemble mean trajectory)" ;
sea_ice_cover_ft_ens_mean:units = "1" ;
sea_ice_cover_ft_ens_mean:standard_name = "sea_ice_area_fraction" ;
int layer(layer) ;
layer:long_name = "Cloud layer number" ;
layer:units = "1" ;
float range(range) ;
range:long_name = "Height above ground level" ;
range:units = "m" ;
range:standard_name = "height" ;
float mwr_lwp_1h_mean_arscl(time) ;
mwr_lwp_1h_mean_arscl:_FillValue = -9999.f ;
mwr_lwp_1h_mean_arscl:long_name = "Liquid water path best-estimate from microwave radiometer, 1-hour mean (averaging window starting at the corresponding timestamp)" ;
mwr_lwp_1h_mean_arscl:units = "g/m^2" ;
float mwr_lwp_1h_std_arscl(time) ;
mwr_lwp_1h_std_arscl:_FillValue = -9999.f ;
mwr_lwp_1h_std_arscl:long_name = "Liquid water path best-estimate from microwave radiometer, 1-hour std (averaging window starting at the corresponding timestamp)" ;
mwr_lwp_1h_std_arscl:units = "g/m^2" ;
float lowest_cth_1h_mean_asrcl(time) ;
lowest_cth_1h_mean_asrcl:_FillValue = -9999.f ;
lowest_cth_1h_mean_asrcl:long_name = "KAZR top height of lowest significant detection layer, before clutter removal, 1-hour mean (averaging window starting at the corresponding timestamp)" ;
lowest_cth_1h_mean_asrcl:units = "m" ;
lowest_cth_1h_mean_asrcl:valid_range = 0.f, 25000.f ;
float lowest_cth_1h_std_asrcl(time) ;
lowest_cth_1h_std_asrcl:_FillValue = -9999.f ;

lowest_cth_1h_std_asrcl:long_name = "KAZR top height of lowest significant detection layer, before clutter removal, 1-hour std (averaging window starting at the corresponding timestamp)" ;
lowest_cth_1h_std_asrcl:units = "m" ;
lowest_cth_1h_std_asrcl:valid_range = 0.f, 25000.f ;
float lowest_cbh_1h_mean_asrcl(time) ;
lowest_cbh_1h_mean_asrcl:_FillValue = -9999.f ;
lowest_cbh_1h_mean_asrcl:long_name = "Cloud base best estimate, based on ceilometer and micropulse lidar, 1-hour mean (averaging window starting at the corresponding timestamp)" ;
lowest_cbh_1h_mean_asrcl:units = "m" ;
lowest_cbh_1h_mean_asrcl:valid_range = 0.f, 25000.f ;
lowest_cbh_1h_mean_asrcl:comment = "-2. Possible clear sky (No MPL observations available, Ceilometer obscured, but no cloud detected), -1. Clear sky, >= 0. Valid cloud base height" ;
float lowest_cbh_1h_std_asrcl(time) ;
lowest_cbh_1h_std_asrcl:_FillValue = -9999.f ;
lowest_cbh_1h_std_asrcl:long_name = "Cloud base best estimate, based on ceilometer and micropulse lidar, 1-hour std (averaging window starting at the corresponding timestamp)" ;
lowest_cbh_1h_std_asrcl:units = "m" ;
lowest_cbh_1h_std_asrcl:valid_range = 0.f, 25000.f ;
lowest_cbh_1h_std_asrcl:comment = "-2. Possible clear sky (No MPL observations available, Ceilometer obscured, but no cloud detected), -1. Clear sky, >= 0. Valid cloud base height" ;
float hydrometeor_layer_base_1h_mean_arscl(time, layer) ;
hydrometeor_layer_base_1h_mean_arscl:_FillValue = -9999.f ;
hydrometeor_layer_base_1h_mean_arscl:long_name = "Base height of hydrometeor layers for up to 10 layers, based on combined radar and micropulse lidar observations, 1-hour mean (averaging window starting at the corresponding timestamp)" ;
hydrometeor_layer_base_1h_mean_arscl:units = "m" ;
hydrometeor_layer_base_1h_mean_arscl:valid_range = 0.f, 25000.f ;
float hydrometeor_layer_base_1h_std_arscl(time, layer) ;
hydrometeor_layer_base_1h_std_arscl:_FillValue = -9999.f ;
hydrometeor_layer_base_1h_std_arscl:long_name = "Base height of hydrometeor layers for up to 10 layers, based on combined radar and micropulse lidar observations, 1-hour std (averaging window starting at the corresponding timestamp)" ;
hydrometeor_layer_base_1h_std_arscl:units = "m" ;
hydrometeor_layer_base_1h_std_arscl:valid_range = 0.f, 25000.f ;
float hydrometeor_layer_top_1h_mean_arscl(time, layer) ;
hydrometeor_layer_top_1h_mean_arscl:_FillValue = -9999.f ;
hydrometeor_layer_top_1h_mean_arscl:long_name = "Top height of hydrometeor layers for up to 10 layers, based on combined radar and micropulse lidar observations, 1-hour mean (averaging window starting at the corresponding timestamp)" ;
hydrometeor_layer_top_1h_mean_arscl:units = "m" ;
hydrometeor_layer_top_1h_mean_arscl:valid_range = 0.f, 25000.f ;
float hydrometeor_layer_top_1h_std_arscl(time, layer) ;
hydrometeor_layer_top_1h_std_arscl:_FillValue = -9999.f ;
hydrometeor_layer_top_1h_std_arscl:long_name = "Top height of hydrometeor layers for up to 10 layers, based on combined radar and micropulse lidar observations, 1-hour std (averaging window starting at the corresponding timestamp)" ;

```
                hydrometeor_layer_top_1h_std_arscl:units = "m" ;
                hydrometeor_layer_top_1h_std_arscl:valid_range = 0.f, 25000.f ;
        float lowest_liq_cld_base(time) ;
                lowest_liq_cld_base:_FillValue = -9999.f ;
                lowest_liq_cld_base:long_name = "Lowest liquid-bearing cloud deck base height " ;
                lowest_liq_cld_base:units = "m" ;
        float lowest_liq_cld_base_sd(time) ;
                lowest_liq_cld_base_sd:_FillValue = -9999.f ;
                lowest_liq_cld_base_sd:long_name = "Lowest liquid-bearing cloud deck base height
standard deviation" ;
                lowest_liq_cld_base_sd:units = "m" ;
        float lowest_liq_cld_top(time) ;
                lowest_liq_cld_top:_FillValue = -9999.f ;
                lowest_liq_cld_top:long_name = "Lowest liquid-bearing cloud deck top height " ;
                lowest_liq_cld_top:units = "m" ;
        float lowest_liq_cld_top_sd(time) ;
                lowest_liq_cld_top_sd:_FillValue = -9999.f ;
                lowest_liq_cld_top_sd:long_name = "Lowest liquid-bearing cloud deck top height
standard deviation" ;
                lowest_liq_cld_top_sd:units = "m" ;
        double liq_cld_fraction(time) ;
                liq_cld_fraction:_FillValue = -9999. ;
                liq_cld_fraction:long_name = "Liquid-bearing cloud occurrence fraction during 1-hour
window" ;
                liq_cld_fraction:units = "1" ;
        double hyd_fraction(time, range) ;
                hyd_fraction:_FillValue = -9999. ;
                hyd_fraction:long_name = "Hydrometeor occurrence fraction during 1-hour window
based on KAZR and MPL data (requiring that data from at least one of those instruments exist)" ;
                hyd_fraction:units = "1" ;
        double time(time) ;
                time:long_name = "Time in seconds since volume start" ;
                time:units = "seconds since 2023-08-20 00:00:00 0:00" ;
        double time_offset(time) ;
                time_offset:_FillValue = -9999. ;
                time_offset:long_name = "Time offset from base_time" ;
                time_offset:units = "seconds since 2023-08-20 00:00:00 0:00" ;
        int base_time ;
                base_time:_FillValue = -9999 ;
                base_time:long_name = "Base time in Epoch" ;
                base_time:units = "seconds since 1970-01-01 0:00:00 0:00" ;
        double lat ;
                lat:_FillValue = -9999. ;
                lat:long_name = "latitude" ;
                lat:units = "degree_N" ;
        double lon ;
```

lon:_FillValue = -9999. ;
lon:long_name = "longitude" ;
lon:units = "degree_E" ;
double alt ;
alt:_FillValue = -9999. ;
alt:long_name = "Altitude above mean sea level" ;
alt:units = "m" ;

// global attributes:
:mode = "arscl" ;
:ensemble_size = "99" ;
:ensemble_size_FT = "9" ;
:ensemble_dx_from_center = "[-7.5 0.   7.5] km" ;
:ensemble_dy_from_center = "[-7.5 0.   7.5] km" ;
:ensemble_dz_from_center = "[0] m" ;
:used_era5_resolution = "0.25 degrees" ;
:total_trajectory_hours = "[-120, 120]" ;
:number_of_starting_center_points_per_time_step = "[11 11 11 11 11 11 11 11 11]" ;
:title = "Back and forward trajectories for primary cloud decks based on ARSCL" ;
:summary = "The ARMTRAJ VAP provides trajectory datasets initialized at ARM deployment coordinates and configured using ARM datasets. The trajectory datasets support aerosol, cloud, and planetary boundary layer research. Trajectory calculations use the HYSPLIT model informed by the ERA5 reanalysis dataset at its highest spatial resolution (~31 km). For each sample in each of the datasets, HYSPLIT also runs at multiple starting locations surrounding ARM deployments, enabling an ensemble of runs from which the mean and variability (estimated uncertainty) of each sample\'s trajectory coordinates, thermodynamic properties, or other fields are reported." ;
:keywords = "ARM, ground-based measurements, ground-based remote sensing, clouds, aerosol, PBL, trajectories, Lagrangian analysis" ;
:doi = "10.5439/2309849" ;
:location_description = "Eastern Pacific Cloud Aerosol Precipitation Experiment (EPCAPE), Scripps Pier, La Jolla, CA" ;
:dod_version = "v1.0" ;
:command_line = "python armtraj_run.py" ;
:datastream = "epcarmtrajarsclM1.c1" ;
:data_level = "c1" ;
:facility_id = "M1" ;
:site_id = "epc" ;
:platform_id = "armtrajarscl" ;
:history = "created by user isilber1 on machine dev-proc2 at 27-Sep-2024,15:26:49" ;
}

Appendix E

ARMTRAJ-AAF Output Data

```
netcdf sgparmtrajaafU2.c1.20240518.000000 {
dimensions:
        time = UNLIMITED ; // (5 currently)
        trajectory_time = 121 ;
        trajectory_type = 1 ;
        vert_layer = 11 ;
        time_aaf = 26952 ;
variables:
        int trajectory_time(trajectory_time) ;
                trajectory_time:long_name = "Trajectory offset from start time (0 h)" ;
                trajectory_time:units = "h" ;
        int trajectory_type(trajectory_type) ;
                trajectory_type:long_name = "-1 - back trajectories, 1 - fwd trajectories" ;
                trajectory_type:units = "1" ;
        double date(time, trajectory_time) ;
                date:_FillValue = -9999. ;
                date:long_name = "Trajectory date per trajectory type and time step" ;
                date:units = "seconds since 1970-01-01" ;
        double latitude(time, trajectory_time, vert_layer) ;
                latitude:_FillValue = -9999. ;
                latitude:long_name = "Airmass latitude" ;
                latitude:units = "degree_N" ;
                latitude:standard_name = "grid_latitude" ;
        double longitude(time, trajectory_time, vert_layer) ;
                longitude:_FillValue = -9999. ;
                longitude:long_name = "Airmass longitude" ;
                longitude:units = "degree_E" ;
                longitude:standard_name = "grid_longitude" ;
        double height(time, trajectory_time, vert_layer) ;
                height:_FillValue = -9999. ;
                height:long_name = "Airmass height above ground level" ;
                height:units = "m" ;
                height:standard_name = "height" ;
        double pres(time, trajectory_time, vert_layer) ;
```

pres:_FillValue = -9999. ;
pres:long_name = "Airmass pressure" ;
pres:units = "hPa" ;
pres:standard_name = "air_pressure" ;
double theta(time, trajectory_time, vert_layer) ;
theta:_FillValue = -9999. ;
theta:long_name = "Airmass potential temperature" ;
theta:units = "K" ;
theta:standard_name = "air_potential_temperature" ;
double temp(time, trajectory_time, vert_layer) ;
temp:_FillValue = -9999. ;
temp:long_name = "Airmass temperature" ;
temp:units = "degC" ;
temp:standard_name = "air_temperature" ;
double pblh(time, trajectory_time, vert_layer) ;
pblh:_FillValue = -9999. ;
pblh:long_name = "Planetary boundary layer height in airmass column" ;
pblh:units = "m" ;
pblh:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh(time, trajectory_time, vert_layer) ;
rh:_FillValue = -9999. ;
rh:long_name = "Airmass relative humidity" ;
rh:units = "%" ;
rh:standard_name = "relative_humidity" ;
double surf_altitude(time, trajectory_time, vert_layer) ;
surf_altitude:_FillValue = -9999. ;
surf_altitude:long_name = "Surface altitude above mean sea level in airmass column" ;
surf_altitude:units = "m" ;
surf_altitude:standard_name = "surface_altitude" ;
double qv(time, trajectory_time, vert_layer) ;
qv:_FillValue = -9999. ;
qv:long_name = "Airmass specific humidity" ;
qv:units = "g/kg" ;
qv:standard_name = "specific_humidity" ;
double rhi(time, trajectory_time, vert_layer) ;
rhi:_FillValue = -9999. ;
rhi:long_name = "Airmass relative humidity with respect to ice" ;
rhi:units = "%" ;
double theta_e(time, trajectory_time, vert_layer) ;
theta_e:_FillValue = -9999. ;
theta_e:long_name = "Airmass equivalent potential temperature (excluding condensate from the calculation)" ;
theta_e:units = "K" ;
theta_e:standard_name = "air_equivalent_potential_temperature" ;
double theta_v(time, trajectory_time, vert_layer) ;
theta_v:_FillValue = -9999. ;

```
        theta_v:long_name = "Airmass virtual potential temperature" ;
        theta_v:units = "K" ;
    double altitude(time, trajectory_time, vert_layer) ;
        altitude:_FillValue = -9999. ;
        altitude:long_name = "Airmass altitude above mean sea level" ;
        altitude:units = "m" ;
        altitude:standard_name = "height_above_mean_sea_level" ;
    double wvert(time, trajectory_time, vert_layer) ;
        wvert:_FillValue = -9999. ;
        wvert:long_name = "Mean hourly airmass ascent rate" ;
        wvert:units = "m/s" ;
        wvert:standard_name = "upward_air_velocity" ;
    double height_to_pblh_ratio(time, trajectory_time, vert_layer) ;
        height_to_pblh_ratio:_FillValue = -9999. ;
        height_to_pblh_ratio:long_name = "Airmass height-to-PBLH ratio (>1 - airmass above
PBL; <1 - airmass within PBL)" ;
        height_to_pblh_ratio:units = "1" ;
    double latitude_sfc(time, trajectory_time) ;
        latitude_sfc:_FillValue = -9999. ;
        latitude_sfc:long_name = "Surface (at initialization) airmass latitude" ;
        latitude_sfc:units = "degree_N" ;
        latitude_sfc:standard_name = "grid_latitude" ;
    double longitude_sfc(time, trajectory_time) ;
        longitude_sfc:_FillValue = -9999. ;
        longitude_sfc:long_name = "Surface (at initialization) airmass longitude" ;
        longitude_sfc:units = "degree_E" ;
        longitude_sfc:standard_name = "grid_longitude" ;
    double height_sfc(time, trajectory_time) ;
        height_sfc:_FillValue = -9999. ;
        height_sfc:long_name = "Surface (at initialization) airmass height above ground level" ;
        height_sfc:units = "m" ;
        height_sfc:standard_name = "height" ;
    double pres_sfc(time, trajectory_time) ;
        pres_sfc:_FillValue = -9999. ;
        pres_sfc:long_name = "Surface (at initialization) airmass pressure" ;
        pres_sfc:units = "hPa" ;
        pres_sfc:standard_name = "air_pressure" ;
    double theta_sfc(time, trajectory_time) ;
        theta_sfc:_FillValue = -9999. ;
        theta_sfc:long_name = "Surface (at initialization) airmass potential temperature" ;
        theta_sfc:units = "K" ;
        theta_sfc:standard_name = "air_potential_temperature" ;
    double temp_sfc(time, trajectory_time) ;
        temp_sfc:_FillValue = -9999. ;
        temp_sfc:long_name = "Surface (at initialization) airmass temperature" ;
        temp_sfc:units = "degC" ;
```

temp_sfc:standard_name = "air_temperature" ;
double pblh_sfc(time, trajectory_time) ;
pblh_sfc:_FillValue = -9999. ;
pblh_sfc:long_name = "Surface (at initialization) planetary boundary layer height in airmass column" ;
pblh_sfc:units = "m" ;
pblh_sfc:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh_sfc(time, trajectory_time) ;
rh_sfc:_FillValue = -9999. ;
rh_sfc:long_name = "Surface (at initialization) airmass relative humidity" ;
rh_sfc:units = "%" ;
rh_sfc:standard_name = "relative_humidity" ;
double surf_altitude_sfc(time, trajectory_time) ;
surf_altitude_sfc:_FillValue = -9999. ;
surf_altitude_sfc:long_name = "Surface (at initialization) surface altitude above mean sea level in airmass column" ;
surf_altitude_sfc:units = "m" ;
surf_altitude_sfc:standard_name = "surface_altitude" ;
double qv_sfc(time, trajectory_time) ;
qv_sfc:_FillValue = -9999. ;
qv_sfc:long_name = "Surface (at initialization) airmass specific humidity" ;
qv_sfc:units = "g/kg" ;
qv_sfc:standard_name = "specific_humidity" ;
double rhi_sfc(time, trajectory_time) ;
rhi_sfc:_FillValue = -9999. ;
rhi_sfc:long_name = "Surface (at initialization) airmass relative humidity with respect to ice" ;
rhi_sfc:units = "%" ;
double theta_e_sfc(time, trajectory_time) ;
theta_e_sfc:_FillValue = -9999. ;
theta_e_sfc:long_name = "Surface (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation)" ;
theta_e_sfc:units = "K" ;
theta_e_sfc:standard_name = "air_equivalent_potential_temperature" ;
double theta_v_sfc(time, trajectory_time) ;
theta_v_sfc:_FillValue = -9999. ;
theta_v_sfc:long_name = "Surface (at initialization) airmass virtual potential temperature" ;
theta_v_sfc:units = "K" ;
double altitude_sfc(time, trajectory_time) ;
altitude_sfc:_FillValue = -9999. ;
altitude_sfc:long_name = "Surface (at initialization) airmass altitude above mean sea level" ;
altitude_sfc:units = "m" ;
altitude_sfc:standard_name = "height_above_mean_sea_level" ;
double wvert_sfc(time, trajectory_time) ;

wvert_sfc:_FillValue = -9999. ;
wvert_sfc:long_name = "Surface (at initialization) mean hourly airmass ascent rate" ;
wvert_sfc:units = "m/s" ;
wvert_sfc:standard_name = "upward_air_velocity" ;
double height_to_pblh_ratio_sfc(time, trajectory_time) ;
height_to_pblh_ratio_sfc:_FillValue = -9999. ;
height_to_pblh_ratio_sfc:long_name = "Surface (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl)" ;
height_to_pblh_ratio_sfc:units = "1" ;
double latitude_ens_mean(time, trajectory_time, vert_layer) ;
latitude_ens_mean:_FillValue = -9999. ;
latitude_ens_mean:long_name = "Airmass latitude ensemble mean" ;
latitude_ens_mean:units = "degree_N" ;
latitude_ens_mean:standard_name = "grid_latitude" ;
double latitude_ens_std(time, trajectory_time, vert_layer) ;
latitude_ens_std:_FillValue = -9999. ;
latitude_ens_std:long_name = "Airmass latitude ensemble std" ;
latitude_ens_std:units = "degree_N" ;
latitude_ens_std:standard_name = "grid_latitude" ;
double latitude_ens_min(time, trajectory_time, vert_layer) ;
latitude_ens_min:_FillValue = -9999. ;
latitude_ens_min:long_name = "Airmass latitude ensemble min" ;
latitude_ens_min:units = "degree_N" ;
latitude_ens_min:standard_name = "grid_latitude" ;
double latitude_ens_max(time, trajectory_time, vert_layer) ;
latitude_ens_max:_FillValue = -9999. ;
latitude_ens_max:long_name = "Airmass latitude ensemble max" ;
latitude_ens_max:units = "degree_N" ;
latitude_ens_max:standard_name = "grid_latitude" ;
double longitude_ens_mean(time, trajectory_time, vert_layer) ;
longitude_ens_mean:_FillValue = -9999. ;
longitude_ens_mean:long_name = "Airmass longitude ensemble mean" ;
longitude_ens_mean:units = "degree_E" ;
longitude_ens_mean:standard_name = "grid_longitude" ;
double longitude_ens_std(time, trajectory_time, vert_layer) ;
longitude_ens_std:_FillValue = -9999. ;
longitude_ens_std:long_name = "Airmass longitude ensemble std" ;
longitude_ens_std:units = "degree_E" ;
longitude_ens_std:standard_name = "grid_longitude" ;
double longitude_ens_min(time, trajectory_time, vert_layer) ;
longitude_ens_min:_FillValue = -9999. ;
longitude_ens_min:long_name = "Airmass longitude ensemble min" ;
longitude_ens_min:units = "degree_E" ;
longitude_ens_min:standard_name = "grid_longitude" ;
double longitude_ens_max(time, trajectory_time, vert_layer) ;
longitude_ens_max:_FillValue = -9999. ;

longitude_ens_max:long_name = "Airmass longitude ensemble max" ;
longitude_ens_max:units = "degree_E" ;
longitude_ens_max:standard_name = "grid_longitude" ;
double height_ens_mean(time, trajectory_time, vert_layer) ;
height_ens_mean:_FillValue = -9999. ;
height_ens_mean:long_name = "Airmass height above ground level ensemble mean" ;
height_ens_mean:units = "m" ;
height_ens_mean:standard_name = "height" ;
double height_ens_std(time, trajectory_time, vert_layer) ;
height_ens_std:_FillValue = -9999. ;
height_ens_std:long_name = "Airmass height above ground level ensemble std" ;
height_ens_std:units = "m" ;
height_ens_std:standard_name = "height" ;
double height_ens_min(time, trajectory_time, vert_layer) ;
height_ens_min:_FillValue = -9999. ;
height_ens_min:long_name = "Airmass height above ground level ensemble min" ;
height_ens_min:units = "m" ;
height_ens_min:standard_name = "height" ;
double height_ens_max(time, trajectory_time, vert_layer) ;
height_ens_max:_FillValue = -9999. ;
height_ens_max:long_name = "Airmass height above ground level ensemble max" ;
height_ens_max:units = "m" ;
height_ens_max:standard_name = "height" ;
double pres_ens_mean(time, trajectory_time, vert_layer) ;
pres_ens_mean:_FillValue = -9999. ;
pres_ens_mean:long_name = "Airmass pressure ensemble mean" ;
pres_ens_mean:units = "hPa" ;
pres_ens_mean:standard_name = "air_pressure" ;
double pres_ens_std(time, trajectory_time, vert_layer) ;
pres_ens_std:_FillValue = -9999. ;
pres_ens_std:long_name = "Airmass pressure ensemble std" ;
pres_ens_std:units = "hPa" ;
pres_ens_std:standard_name = "air_pressure" ;
double pres_ens_min(time, trajectory_time, vert_layer) ;
pres_ens_min:_FillValue = -9999. ;
pres_ens_min:long_name = "Airmass pressure ensemble min" ;
pres_ens_min:units = "hPa" ;
pres_ens_min:standard_name = "air_pressure" ;
double pres_ens_max(time, trajectory_time, vert_layer) ;
pres_ens_max:_FillValue = -9999. ;
pres_ens_max:long_name = "Airmass pressure ensemble max" ;
pres_ens_max:units = "hPa" ;
pres_ens_max:standard_name = "air_pressure" ;
double theta_ens_mean(time, trajectory_time, vert_layer) ;
theta_ens_mean:_FillValue = -9999. ;
theta_ens_mean:long_name = "Airmass potential temperature ensemble mean" ;

theta_ens_mean:units = "K" ;
theta_ens_mean:standard_name = "air_potential_temperature" ;
double theta_ens_std(time, trajectory_time, vert_layer) ;
theta_ens_std:_FillValue = -9999. ;
theta_ens_std:long_name = "Airmass potential temperature ensemble std" ;
theta_ens_std:units = "K" ;
theta_ens_std:standard_name = "air_potential_temperature" ;
double theta_ens_min(time, trajectory_time, vert_layer) ;
theta_ens_min:_FillValue = -9999. ;
theta_ens_min:long_name = "Airmass potential temperature ensemble min" ;
theta_ens_min:units = "K" ;
theta_ens_min:standard_name = "air_potential_temperature" ;
double theta_ens_max(time, trajectory_time, vert_layer) ;
theta_ens_max:_FillValue = -9999. ;
theta_ens_max:long_name = "Airmass potential temperature ensemble max" ;
theta_ens_max:units = "K" ;
theta_ens_max:standard_name = "air_potential_temperature" ;
double temp_ens_mean(time, trajectory_time, vert_layer) ;
temp_ens_mean:_FillValue = -9999. ;
temp_ens_mean:long_name = "Airmass temperature ensemble mean" ;
temp_ens_mean:units = "degC" ;
temp_ens_mean:standard_name = "air_temperature" ;
double temp_ens_std(time, trajectory_time, vert_layer) ;
temp_ens_std:_FillValue = -9999. ;
temp_ens_std:long_name = "Airmass temperature ensemble std" ;
temp_ens_std:units = "degC" ;
temp_ens_std:standard_name = "air_temperature" ;
double temp_ens_min(time, trajectory_time, vert_layer) ;
temp_ens_min:_FillValue = -9999. ;
temp_ens_min:long_name = "Airmass temperature ensemble min" ;
temp_ens_min:units = "degC" ;
temp_ens_min:standard_name = "air_temperature" ;
double temp_ens_max(time, trajectory_time, vert_layer) ;
temp_ens_max:_FillValue = -9999. ;
temp_ens_max:long_name = "Airmass temperature ensemble max" ;
temp_ens_max:units = "degC" ;
temp_ens_max:standard_name = "air_temperature" ;
double pblh_ens_mean(time, trajectory_time, vert_layer) ;
pblh_ens_mean:_FillValue = -9999. ;
pblh_ens_mean:long_name = "Planetary boundary layer height in airmass column ensemble mean" ;
pblh_ens_mean:units = "m" ;
pblh_ens_mean:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ens_std(time, trajectory_time, vert_layer) ;
pblh_ens_std:_FillValue = -9999. ;

pblh_ens_std:long_name = "Planetary boundary layer height in airmass column ensemble std" ;
pblh_ens_std:units = "m" ;
pblh_ens_std:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ens_min(time, trajectory_time, vert_layer) ;
pblh_ens_min:_FillValue = -9999. ;
pblh_ens_min:long_name = "Planetary boundary layer height in airmass column ensemble min" ;
pblh_ens_min:units = "m" ;
pblh_ens_min:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_ens_max(time, trajectory_time, vert_layer) ;
pblh_ens_max:_FillValue = -9999. ;
pblh_ens_max:long_name = "Planetary boundary layer height in airmass column ensemble max" ;
pblh_ens_max:units = "m" ;
pblh_ens_max:standard_name = "atmosphere_boundary_layer_thickness" ;
double rh_ens_mean(time, trajectory_time, vert_layer) ;
rh_ens_mean:_FillValue = -9999. ;
rh_ens_mean:long_name = "Airmass relative humidity ensemble mean" ;
rh_ens_mean:units = "%" ;
rh_ens_mean:standard_name = "relative_humidity" ;
double rh_ens_std(time, trajectory_time, vert_layer) ;
rh_ens_std:_FillValue = -9999. ;
rh_ens_std:long_name = "Airmass relative humidity ensemble std" ;
rh_ens_std:units = "%" ;
rh_ens_std:standard_name = "relative_humidity" ;
double rh_ens_min(time, trajectory_time, vert_layer) ;
rh_ens_min:_FillValue = -9999. ;
rh_ens_min:long_name = "Airmass relative humidity ensemble min" ;
rh_ens_min:units = "%" ;
rh_ens_min:standard_name = "relative_humidity" ;
double rh_ens_max(time, trajectory_time, vert_layer) ;
rh_ens_max:_FillValue = -9999. ;
rh_ens_max:long_name = "Airmass relative humidity ensemble max" ;
rh_ens_max:units = "%" ;
rh_ens_max:standard_name = "relative_humidity" ;
double surf_altitude_ens_mean(time, trajectory_time, vert_layer) ;
surf_altitude_ens_mean:_FillValue = -9999. ;
surf_altitude_ens_mean:long_name = "Surface altitude above mean sea level in airmass column ensemble mean" ;
surf_altitude_ens_mean:units = "m" ;
surf_altitude_ens_mean:standard_name = "surface_altitude" ;
double surf_altitude_ens_std(time, trajectory_time, vert_layer) ;
surf_altitude_ens_std:_FillValue = -9999. ;
surf_altitude_ens_std:long_name = "Surface altitude above mean sea level in airmass column ensemble std" ;

surf_altitude_ens_std:units = "m" ;
surf_altitude_ens_std:standard_name = "surface_altitude" ;
double surf_altitude_ens_min(time, trajectory_time, vert_layer) ;
surf_altitude_ens_min:_FillValue = -9999. ;
surf_altitude_ens_min:long_name = "Surface altitude above mean sea level in airmass column ensemble min" ;
surf_altitude_ens_min:units = "m" ;
surf_altitude_ens_min:standard_name = "surface_altitude" ;
double surf_altitude_ens_max(time, trajectory_time, vert_layer) ;
surf_altitude_ens_max:_FillValue = -9999. ;
surf_altitude_ens_max:long_name = "Surface altitude above mean sea level in airmass column ensemble max" ;
surf_altitude_ens_max:units = "m" ;
surf_altitude_ens_max:standard_name = "surface_altitude" ;
double qv_ens_mean(time, trajectory_time, vert_layer) ;
qv_ens_mean:_FillValue = -9999. ;
qv_ens_mean:long_name = "Airmass specific humidity ensemble mean" ;
qv_ens_mean:units = "g/kg" ;
qv_ens_mean:standard_name = "specific_humidity" ;
double qv_ens_std(time, trajectory_time, vert_layer) ;
qv_ens_std:_FillValue = -9999. ;
qv_ens_std:long_name = "Airmass specific humidity ensemble std" ;
qv_ens_std:units = "g/kg" ;
qv_ens_std:standard_name = "specific_humidity" ;
double qv_ens_min(time, trajectory_time, vert_layer) ;
qv_ens_min:_FillValue = -9999. ;
qv_ens_min:long_name = "Airmass specific humidity ensemble min" ;
qv_ens_min:units = "g/kg" ;
qv_ens_min:standard_name = "specific_humidity" ;
double qv_ens_max(time, trajectory_time, vert_layer) ;
qv_ens_max:_FillValue = -9999. ;
qv_ens_max:long_name = "Airmass specific humidity ensemble max" ;
qv_ens_max:units = "g/kg" ;
qv_ens_max:standard_name = "specific_humidity" ;
double rhi_ens_mean(time, trajectory_time, vert_layer) ;
rhi_ens_mean:_FillValue = -9999. ;
rhi_ens_mean:long_name = "Airmass relative humidity with respect to ice ensemble mean" ;
rhi_ens_mean:units = "%" ;
double rhi_ens_std(time, trajectory_time, vert_layer) ;
rhi_ens_std:_FillValue = -9999. ;
rhi_ens_std:long_name = "Airmass relative humidity with respect to ice ensemble std" ;
rhi_ens_std:units = "%" ;
double rhi_ens_min(time, trajectory_time, vert_layer) ;
rhi_ens_min:_FillValue = -9999. ;
rhi_ens_min:long_name = "Airmass relative humidity with respect to ice ensemble min" ;

```
                rhi_ens_min:units = "%" ;
        double rhi_ens_max(time, trajectory_time, vert_layer) ;
                rhi_ens_max:_FillValue = -9999. ;
                rhi_ens_max:long_name = "Airmass relative humidity with respect to ice ensemble max"
;
                rhi_ens_max:units = "%" ;
        double theta_e_ens_mean(time, trajectory_time, vert_layer) ;
                theta_e_ens_mean:_FillValue = -9999. ;
                theta_e_ens_mean:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble mean" ;
                theta_e_ens_mean:units = "K" ;
                theta_e_ens_mean:standard_name = "air_equivalent_potential_temperature" ;
        double theta_e_ens_std(time, trajectory_time, vert_layer) ;
                theta_e_ens_std:_FillValue = -9999. ;
                theta_e_ens_std:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble std" ;
                theta_e_ens_std:units = "K" ;
                theta_e_ens_std:standard_name = "air_equivalent_potential_temperature" ;
        double theta_e_ens_min(time, trajectory_time, vert_layer) ;
                theta_e_ens_min:_FillValue = -9999. ;
                theta_e_ens_min:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble min" ;
                theta_e_ens_min:units = "K" ;
                theta_e_ens_min:standard_name = "air_equivalent_potential_temperature" ;
        double theta_e_ens_max(time, trajectory_time, vert_layer) ;
                theta_e_ens_max:_FillValue = -9999. ;
                theta_e_ens_max:long_name = "Airmass equivalent potential temperature (excluding
condensate from the calculation) ensemble max" ;
                theta_e_ens_max:units = "K" ;
                theta_e_ens_max:standard_name = "air_equivalent_potential_temperature" ;
        double theta_v_ens_mean(time, trajectory_time, vert_layer) ;
                theta_v_ens_mean:_FillValue = -9999. ;
                theta_v_ens_mean:long_name = "Airmass virtual potential temperature ensemble mean" ;
                theta_v_ens_mean:units = "K" ;
        double theta_v_ens_std(time, trajectory_time, vert_layer) ;
                theta_v_ens_std:_FillValue = -9999. ;
                theta_v_ens_std:long_name = "Airmass virtual potential temperature ensemble std" ;
                theta_v_ens_std:units = "K" ;
        double theta_v_ens_min(time, trajectory_time, vert_layer) ;
                theta_v_ens_min:_FillValue = -9999. ;
                theta_v_ens_min:long_name = "Airmass virtual potential temperature ensemble min" ;
                theta_v_ens_min:units = "K" ;
        double theta_v_ens_max(time, trajectory_time, vert_layer) ;
                theta_v_ens_max:_FillValue = -9999. ;
                theta_v_ens_max:long_name = "Airmass virtual potential temperature ensemble max" ;
                theta_v_ens_max:units = "K" ;
```

```
        double altitude_ens_mean(time, trajectory_time, vert_layer) ;
                altitude_ens_mean:_FillValue = -9999. ;
                altitude_ens_mean:long_name = "Airmass altitude above mean sea level ensemble mean"
;
                altitude_ens_mean:units = "m" ;
                altitude_ens_mean:standard_name = "height_above_mean_sea_level" ;
        double altitude_ens_std(time, trajectory_time, vert_layer) ;
                altitude_ens_std:_FillValue = -9999. ;
                altitude_ens_std:long_name = "Airmass altitude above mean sea level ensemble std" ;
                altitude_ens_std:units = "m" ;
                altitude_ens_std:standard_name = "height_above_mean_sea_level" ;
        double altitude_ens_min(time, trajectory_time, vert_layer) ;
                altitude_ens_min:_FillValue = -9999. ;
                altitude_ens_min:long_name = "Airmass altitude above mean sea level ensemble min" ;
                altitude_ens_min:units = "m" ;
                altitude_ens_min:standard_name = "height_above_mean_sea_level" ;
        double altitude_ens_max(time, trajectory_time, vert_layer) ;
                altitude_ens_max:_FillValue = -9999. ;
                altitude_ens_max:long_name = "Airmass altitude above mean sea level ensemble max" ;
                altitude_ens_max:units = "m" ;
                altitude_ens_max:standard_name = "height_above_mean_sea_level" ;
        double wvert_ens_mean(time, trajectory_time, vert_layer) ;
                wvert_ens_mean:_FillValue = -9999. ;
                wvert_ens_mean:long_name = "Mean hourly airmass ascent rate ensemble mean" ;
                wvert_ens_mean:units = "m/s" ;
                wvert_ens_mean:standard_name = "upward_air_velocity" ;
        double wvert_ens_std(time, trajectory_time, vert_layer) ;
                wvert_ens_std:_FillValue = -9999. ;
                wvert_ens_std:long_name = "Mean hourly airmass ascent rate ensemble std" ;
                wvert_ens_std:units = "m/s" ;
                wvert_ens_std:standard_name = "upward_air_velocity" ;
        double wvert_ens_min(time, trajectory_time, vert_layer) ;
                wvert_ens_min:_FillValue = -9999. ;
                wvert_ens_min:long_name = "Mean hourly airmass ascent rate ensemble min" ;
                wvert_ens_min:units = "m/s" ;
                wvert_ens_min:standard_name = "upward_air_velocity" ;
        double wvert_ens_max(time, trajectory_time, vert_layer) ;
                wvert_ens_max:_FillValue = -9999. ;
                wvert_ens_max:long_name = "Mean hourly airmass ascent rate ensemble max" ;
                wvert_ens_max:units = "m/s" ;
                wvert_ens_max:standard_name = "upward_air_velocity" ;
        double height_to_pblh_ratio_ens_mean(time, trajectory_time, vert_layer) ;
                height_to_pblh_ratio_ens_mean:_FillValue = -9999. ;
                height_to_pblh_ratio_ens_mean:long_name = "Airmass height-to-PBLH ratio (>1 -
airmass above PBL; <1 - airmass within PBL) ensemble mean" ;
                height_to_pblh_ratio_ens_mean:units = "1" ;
```

```
        double height_to_pblh_ratio_ens_std(time, trajectory_time, vert_layer) ;
                height_to_pblh_ratio_ens_std:_FillValue = -9999. ;
                height_to_pblh_ratio_ens_std:long_name = "Airmass height-to-PBLH ratio (>1 - airmass
above PBL; <1 - airmass within PBL) ensemble std" ;
                height_to_pblh_ratio_ens_std:units = "1" ;
        double height_to_pblh_ratio_ens_min(time, trajectory_time, vert_layer) ;
                height_to_pblh_ratio_ens_min:_FillValue = -9999. ;
                height_to_pblh_ratio_ens_min:long_name = "Airmass height-to-PBLH ratio (>1 -
airmass above PBL; <1 - airmass within PBL) ensemble min" ;
                height_to_pblh_ratio_ens_min:units = "1" ;
        double height_to_pblh_ratio_ens_max(time, trajectory_time, vert_layer) ;
                height_to_pblh_ratio_ens_max:_FillValue = -9999. ;
                height_to_pblh_ratio_ens_max:long_name = "Airmass height-to-PBLH ratio (>1 -
airmass above PBL; <1 - airmass within PBL) ensemble max" ;
                height_to_pblh_ratio_ens_max:units = "1" ;
        double latitude_sfc_ens_mean(time, trajectory_time) ;
                latitude_sfc_ens_mean:_FillValue = -9999. ;
                latitude_sfc_ens_mean:long_name = "Surface (at initialization) airmass latitude ensemble
mean" ;
                latitude_sfc_ens_mean:units = "degree_N" ;
                latitude_sfc_ens_mean:standard_name = "grid_latitude" ;
        double latitude_sfc_ens_std(time, trajectory_time) ;
                latitude_sfc_ens_std:_FillValue = -9999. ;
                latitude_sfc_ens_std:long_name = "Surface (at initialization) airmass latitude ensemble
std" ;
                latitude_sfc_ens_std:units = "degree_N" ;
                latitude_sfc_ens_std:standard_name = "grid_latitude" ;
        double latitude_sfc_ens_min(time, trajectory_time) ;
                latitude_sfc_ens_min:_FillValue = -9999. ;
                latitude_sfc_ens_min:long_name = "Surface (at initialization) airmass latitude ensemble
min" ;
                latitude_sfc_ens_min:units = "degree_N" ;
                latitude_sfc_ens_min:standard_name = "grid_latitude" ;
        double latitude_sfc_ens_max(time, trajectory_time) ;
                latitude_sfc_ens_max:_FillValue = -9999. ;
                latitude_sfc_ens_max:long_name = "Surface (at initialization) airmass latitude ensemble
max" ;
                latitude_sfc_ens_max:units = "degree_N" ;
                latitude_sfc_ens_max:standard_name = "grid_latitude" ;
        double longitude_sfc_ens_mean(time, trajectory_time) ;
                longitude_sfc_ens_mean:_FillValue = -9999. ;
                longitude_sfc_ens_mean:long_name = "Surface (at initialization) airmass longitude
ensemble mean" ;
                longitude_sfc_ens_mean:units = "degree_E" ;
                longitude_sfc_ens_mean:standard_name = "grid_longitude" ;
        double longitude_sfc_ens_std(time, trajectory_time) ;
```

longitude_sfc_ens_std:_FillValue = -9999. ;
longitude_sfc_ens_std:long_name = "Surface (at initialization) airmass longitude ensemble std" ;
longitude_sfc_ens_std:units = "degree_E" ;
longitude_sfc_ens_std:standard_name = "grid_longitude" ;
double longitude_sfc_ens_min(time, trajectory_time) ;
longitude_sfc_ens_min:_FillValue = -9999. ;
longitude_sfc_ens_min:long_name = "Surface (at initialization) airmass longitude ensemble min" ;
longitude_sfc_ens_min:units = "degree_E" ;
longitude_sfc_ens_min:standard_name = "grid_longitude" ;
double longitude_sfc_ens_max(time, trajectory_time) ;
longitude_sfc_ens_max:_FillValue = -9999. ;
longitude_sfc_ens_max:long_name = "Surface (at initialization) airmass longitude ensemble max" ;
longitude_sfc_ens_max:units = "degree_E" ;
longitude_sfc_ens_max:standard_name = "grid_longitude" ;
double height_sfc_ens_mean(time, trajectory_time) ;
height_sfc_ens_mean:_FillValue = -9999. ;
height_sfc_ens_mean:long_name = "Surface (at initialization) airmass height above ground level ensemble mean" ;
height_sfc_ens_mean:units = "m" ;
height_sfc_ens_mean:standard_name = "height" ;
double height_sfc_ens_std(time, trajectory_time) ;
height_sfc_ens_std:_FillValue = -9999. ;
height_sfc_ens_std:long_name = "Surface (at initialization) airmass height above ground level ensemble std" ;
height_sfc_ens_std:units = "m" ;
height_sfc_ens_std:standard_name = "height" ;
double height_sfc_ens_min(time, trajectory_time) ;
height_sfc_ens_min:_FillValue = -9999. ;
height_sfc_ens_min:long_name = "Surface (at initialization) airmass height above ground level ensemble min" ;
height_sfc_ens_min:units = "m" ;
height_sfc_ens_min:standard_name = "height" ;
double height_sfc_ens_max(time, trajectory_time) ;
height_sfc_ens_max:_FillValue = -9999. ;
height_sfc_ens_max:long_name = "Surface (at initialization) airmass height above ground level ensemble max" ;
height_sfc_ens_max:units = "m" ;
height_sfc_ens_max:standard_name = "height" ;
double pres_sfc_ens_mean(time, trajectory_time) ;
pres_sfc_ens_mean:_FillValue = -9999. ;
pres_sfc_ens_mean:long_name = "Surface (at initialization) airmass pressure ensemble mean" ;
pres_sfc_ens_mean:units = "hPa" ;

pres_sfc_ens_mean:standard_name = "air_pressure" ;
double pres_sfc_ens_std(time, trajectory_time) ;
pres_sfc_ens_std:_FillValue = -9999. ;
pres_sfc_ens_std:long_name = "Surface (at initialization) airmass pressure ensemble std" ;
pres_sfc_ens_std:units = "hPa" ;
pres_sfc_ens_std:standard_name = "air_pressure" ;
double pres_sfc_ens_min(time, trajectory_time) ;
pres_sfc_ens_min:_FillValue = -9999. ;
pres_sfc_ens_min:long_name = "Surface (at initialization) airmass pressure ensemble min" ;
pres_sfc_ens_min:units = "hPa" ;
pres_sfc_ens_min:standard_name = "air_pressure" ;
double pres_sfc_ens_max(time, trajectory_time) ;
pres_sfc_ens_max:_FillValue = -9999. ;
pres_sfc_ens_max:long_name = "Surface (at initialization) airmass pressure ensemble max" ;
pres_sfc_ens_max:units = "hPa" ;
pres_sfc_ens_max:standard_name = "air_pressure" ;
double theta_sfc_ens_mean(time, trajectory_time) ;
theta_sfc_ens_mean:_FillValue = -9999. ;
theta_sfc_ens_mean:long_name = "Surface (at initialization) airmass potential temperature ensemble mean" ;
theta_sfc_ens_mean:units = "K" ;
theta_sfc_ens_mean:standard_name = "air_potential_temperature" ;
double theta_sfc_ens_std(time, trajectory_time) ;
theta_sfc_ens_std:_FillValue = -9999. ;
theta_sfc_ens_std:long_name = "Surface (at initialization) airmass potential temperature ensemble std" ;
theta_sfc_ens_std:units = "K" ;
theta_sfc_ens_std:standard_name = "air_potential_temperature" ;
double theta_sfc_ens_min(time, trajectory_time) ;
theta_sfc_ens_min:_FillValue = -9999. ;
theta_sfc_ens_min:long_name = "Surface (at initialization) airmass potential temperature ensemble min" ;
theta_sfc_ens_min:units = "K" ;
theta_sfc_ens_min:standard_name = "air_potential_temperature" ;
double theta_sfc_ens_max(time, trajectory_time) ;
theta_sfc_ens_max:_FillValue = -9999. ;
theta_sfc_ens_max:long_name = "Surface (at initialization) airmass potential temperature ensemble max" ;
theta_sfc_ens_max:units = "K" ;
theta_sfc_ens_max:standard_name = "air_potential_temperature" ;
double temp_sfc_ens_mean(time, trajectory_time) ;
temp_sfc_ens_mean:_FillValue = -9999. ;

temp_sfc_ens_mean:long_name = "Surface (at initialization) airmass temperature ensemble mean" ;
temp_sfc_ens_mean:units = "degC" ;
temp_sfc_ens_mean:standard_name = "air_temperature" ;
double temp_sfc_ens_std(time, trajectory_time) ;
temp_sfc_ens_std:_FillValue = -9999. ;
temp_sfc_ens_std:long_name = "Surface (at initialization) airmass temperature ensemble std" ;
temp_sfc_ens_std:units = "degC" ;
temp_sfc_ens_std:standard_name = "air_temperature" ;
double temp_sfc_ens_min(time, trajectory_time) ;
temp_sfc_ens_min:_FillValue = -9999. ;
temp_sfc_ens_min:long_name = "Surface (at initialization) airmass temperature ensemble min" ;
temp_sfc_ens_min:units = "degC" ;
temp_sfc_ens_min:standard_name = "air_temperature" ;
double temp_sfc_ens_max(time, trajectory_time) ;
temp_sfc_ens_max:_FillValue = -9999. ;
temp_sfc_ens_max:long_name = "Surface (at initialization) airmass temperature ensemble max" ;
temp_sfc_ens_max:units = "degC" ;
temp_sfc_ens_max:standard_name = "air_temperature" ;
double pblh_sfc_ens_mean(time, trajectory_time) ;
pblh_sfc_ens_mean:_FillValue = -9999. ;
pblh_sfc_ens_mean:long_name = "Surface (at initialization) planetary boundary layer height in airmass column ensemble mean" ;
pblh_sfc_ens_mean:units = "m" ;
pblh_sfc_ens_mean:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_sfc_ens_std(time, trajectory_time) ;
pblh_sfc_ens_std:_FillValue = -9999. ;
pblh_sfc_ens_std:long_name = "Surface (at initialization) planetary boundary layer height in airmass column ensemble std" ;
pblh_sfc_ens_std:units = "m" ;
pblh_sfc_ens_std:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_sfc_ens_min(time, trajectory_time) ;
pblh_sfc_ens_min:_FillValue = -9999. ;
pblh_sfc_ens_min:long_name = "Surface (at initialization) planetary boundary layer height in airmass column ensemble min" ;
pblh_sfc_ens_min:units = "m" ;
pblh_sfc_ens_min:standard_name = "atmosphere_boundary_layer_thickness" ;
double pblh_sfc_ens_max(time, trajectory_time) ;
pblh_sfc_ens_max:_FillValue = -9999. ;
pblh_sfc_ens_max:long_name = "Surface (at initialization) planetary boundary layer height in airmass column ensemble max" ;
pblh_sfc_ens_max:units = "m" ;
pblh_sfc_ens_max:standard_name = "atmosphere_boundary_layer_thickness" ;

double rh_sfc_ens_mean(time, trajectory_time) ;
rh_sfc_ens_mean:_FillValue = -9999. ;
rh_sfc_ens_mean:long_name = "Surface (at initialization) airmass relative humidity ensemble mean" ;
rh_sfc_ens_mean:units = "%" ;
rh_sfc_ens_mean:standard_name = "relative_humidity" ;
double rh_sfc_ens_std(time, trajectory_time) ;
rh_sfc_ens_std:_FillValue = -9999. ;
rh_sfc_ens_std:long_name = "Surface (at initialization) airmass relative humidity ensemble std" ;
rh_sfc_ens_std:units = "%" ;
rh_sfc_ens_std:standard_name = "relative_humidity" ;
double rh_sfc_ens_min(time, trajectory_time) ;
rh_sfc_ens_min:_FillValue = -9999. ;
rh_sfc_ens_min:long_name = "Surface (at initialization) airmass relative humidity ensemble min" ;
rh_sfc_ens_min:units = "%" ;
rh_sfc_ens_min:standard_name = "relative_humidity" ;
double rh_sfc_ens_max(time, trajectory_time) ;
rh_sfc_ens_max:_FillValue = -9999. ;
rh_sfc_ens_max:long_name = "Surface (at initialization) airmass relative humidity ensemble max" ;
rh_sfc_ens_max:units = "%" ;
rh_sfc_ens_max:standard_name = "relative_humidity" ;
double surf_altitude_sfc_ens_mean(time, trajectory_time) ;
surf_altitude_sfc_ens_mean:_FillValue = -9999. ;
surf_altitude_sfc_ens_mean:long_name = "Surface (at initialization) surface altitude above mean sea level in airmass column ensemble mean" ;
surf_altitude_sfc_ens_mean:units = "m" ;
surf_altitude_sfc_ens_mean:standard_name = "surface_altitude" ;
double surf_altitude_sfc_ens_std(time, trajectory_time) ;
surf_altitude_sfc_ens_std:_FillValue = -9999. ;
surf_altitude_sfc_ens_std:long_name = "Surface (at initialization) surface altitude above mean sea level in airmass column ensemble std" ;
surf_altitude_sfc_ens_std:units = "m" ;
surf_altitude_sfc_ens_std:standard_name = "surface_altitude" ;
double surf_altitude_sfc_ens_min(time, trajectory_time) ;
surf_altitude_sfc_ens_min:_FillValue = -9999. ;
surf_altitude_sfc_ens_min:long_name = "Surface (at initialization) surface altitude above mean sea level in airmass column ensemble min" ;
surf_altitude_sfc_ens_min:units = "m" ;
surf_altitude_sfc_ens_min:standard_name = "surface_altitude" ;
double surf_altitude_sfc_ens_max(time, trajectory_time) ;
surf_altitude_sfc_ens_max:_FillValue = -9999. ;
surf_altitude_sfc_ens_max:long_name = "Surface (at initialization) surface altitude above mean sea level in airmass column ensemble max" ;

surf_altitude_sfc_ens_max:units = "m" ;
surf_altitude_sfc_ens_max:standard_name = "surface_altitude" ;
double qv_sfc_ens_mean(time, trajectory_time) ;
qv_sfc_ens_mean:_FillValue = -9999. ;
qv_sfc_ens_mean:long_name = "Surface (at initialization) airmass specific humidity ensemble mean" ;
qv_sfc_ens_mean:units = "g/kg" ;
qv_sfc_ens_mean:standard_name = "specific_humidity" ;
double qv_sfc_ens_std(time, trajectory_time) ;
qv_sfc_ens_std:_FillValue = -9999. ;
qv_sfc_ens_std:long_name = "Surface (at initialization) airmass specific humidity ensemble std" ;
qv_sfc_ens_std:units = "g/kg" ;
qv_sfc_ens_std:standard_name = "specific_humidity" ;
double qv_sfc_ens_min(time, trajectory_time) ;
qv_sfc_ens_min:_FillValue = -9999. ;
qv_sfc_ens_min:long_name = "Surface (at initialization) airmass specific humidity ensemble min" ;
qv_sfc_ens_min:units = "g/kg" ;
qv_sfc_ens_min:standard_name = "specific_humidity" ;
double qv_sfc_ens_max(time, trajectory_time) ;
qv_sfc_ens_max:_FillValue = -9999. ;
qv_sfc_ens_max:long_name = "Surface (at initialization) airmass specific humidity ensemble max" ;
qv_sfc_ens_max:units = "g/kg" ;
qv_sfc_ens_max:standard_name = "specific_humidity" ;
double rhi_sfc_ens_mean(time, trajectory_time) ;
rhi_sfc_ens_mean:_FillValue = -9999. ;
rhi_sfc_ens_mean:long_name = "Surface (at initialization) airmass relative humidity with respect to ice ensemble mean" ;
rhi_sfc_ens_mean:units = "%" ;
double rhi_sfc_ens_std(time, trajectory_time) ;
rhi_sfc_ens_std:_FillValue = -9999. ;
rhi_sfc_ens_std:long_name = "Surface (at initialization) airmass relative humidity with respect to ice ensemble std" ;
rhi_sfc_ens_std:units = "%" ;
double rhi_sfc_ens_min(time, trajectory_time) ;
rhi_sfc_ens_min:_FillValue = -9999. ;
rhi_sfc_ens_min:long_name = "Surface (at initialization) airmass relative humidity with respect to ice ensemble min" ;
rhi_sfc_ens_min:units = "%" ;
double rhi_sfc_ens_max(time, trajectory_time) ;
rhi_sfc_ens_max:_FillValue = -9999. ;
rhi_sfc_ens_max:long_name = "Surface (at initialization) airmass relative humidity with respect to ice ensemble max" ;
rhi_sfc_ens_max:units = "%" ;

double theta_e_sfc_ens_mean(time, trajectory_time) ;
theta_e_sfc_ens_mean:_FillValue = -9999. ;
theta_e_sfc_ens_mean:long_name = "Surface (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble mean" ;
theta_e_sfc_ens_mean:units = "K" ;
theta_e_sfc_ens_mean:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_sfc_ens_std(time, trajectory_time) ;
theta_e_sfc_ens_std:_FillValue = -9999. ;
theta_e_sfc_ens_std:long_name = "Surface (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble std" ;
theta_e_sfc_ens_std:units = "K" ;
theta_e_sfc_ens_std:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_sfc_ens_min(time, trajectory_time) ;
theta_e_sfc_ens_min:_FillValue = -9999. ;
theta_e_sfc_ens_min:long_name = "Surface (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble min" ;
theta_e_sfc_ens_min:units = "K" ;
theta_e_sfc_ens_min:standard_name = "air_equivalent_potential_temperature" ;
double theta_e_sfc_ens_max(time, trajectory_time) ;
theta_e_sfc_ens_max:_FillValue = -9999. ;
theta_e_sfc_ens_max:long_name = "Surface (at initialization) airmass equivalent potential temperature (excluding condensate from the calculation) ensemble max" ;
theta_e_sfc_ens_max:units = "K" ;
theta_e_sfc_ens_max:standard_name = "air_equivalent_potential_temperature" ;
double theta_v_sfc_ens_mean(time, trajectory_time) ;
theta_v_sfc_ens_mean:_FillValue = -9999. ;
theta_v_sfc_ens_mean:long_name = "Surface (at initialization) airmass virtual potential temperature ensemble mean" ;
theta_v_sfc_ens_mean:units = "K" ;
double theta_v_sfc_ens_std(time, trajectory_time) ;
theta_v_sfc_ens_std:_FillValue = -9999. ;
theta_v_sfc_ens_std:long_name = "Surface (at initialization) airmass virtual potential temperature ensemble std" ;
theta_v_sfc_ens_std:units = "K" ;
double theta_v_sfc_ens_min(time, trajectory_time) ;
theta_v_sfc_ens_min:_FillValue = -9999. ;
theta_v_sfc_ens_min:long_name = "Surface (at initialization) airmass virtual potential temperature ensemble min" ;
theta_v_sfc_ens_min:units = "K" ;
double theta_v_sfc_ens_max(time, trajectory_time) ;
theta_v_sfc_ens_max:_FillValue = -9999. ;
theta_v_sfc_ens_max:long_name = "Surface (at initialization) airmass virtual potential temperature ensemble max" ;
theta_v_sfc_ens_max:units = "K" ;
double altitude_sfc_ens_mean(time, trajectory_time) ;
altitude_sfc_ens_mean:_FillValue = -9999. ;

altitude_sfc_ens_mean:long_name = "Surface (at initialization) airmass altitude above mean sea level ensemble mean" ;
altitude_sfc_ens_mean:units = "m" ;
altitude_sfc_ens_mean:standard_name = "height_above_mean_sea_level" ;
double altitude_sfc_ens_std(time, trajectory_time) ;
altitude_sfc_ens_std:_FillValue = -9999. ;
altitude_sfc_ens_std:long_name = "Surface (at initialization) airmass altitude above mean sea level ensemble std" ;
altitude_sfc_ens_std:units = "m" ;
altitude_sfc_ens_std:standard_name = "height_above_mean_sea_level" ;
double altitude_sfc_ens_min(time, trajectory_time) ;
altitude_sfc_ens_min:_FillValue = -9999. ;
altitude_sfc_ens_min:long_name = "Surface (at initialization) airmass altitude above mean sea level ensemble min" ;
altitude_sfc_ens_min:units = "m" ;
altitude_sfc_ens_min:standard_name = "height_above_mean_sea_level" ;
double altitude_sfc_ens_max(time, trajectory_time) ;
altitude_sfc_ens_max:_FillValue = -9999. ;
altitude_sfc_ens_max:long_name = "Surface (at initialization) airmass altitude above mean sea level ensemble max" ;
altitude_sfc_ens_max:units = "m" ;
altitude_sfc_ens_max:standard_name = "height_above_mean_sea_level" ;
double wvert_sfc_ens_mean(time, trajectory_time) ;
wvert_sfc_ens_mean:_FillValue = -9999. ;
wvert_sfc_ens_mean:long_name = "Surface (at initialization) mean hourly airmass ascent rate ensemble mean" ;
wvert_sfc_ens_mean:units = "m/s" ;
wvert_sfc_ens_mean:standard_name = "upward_air_velocity" ;
double wvert_sfc_ens_std(time, trajectory_time) ;
wvert_sfc_ens_std:_FillValue = -9999. ;
wvert_sfc_ens_std:long_name = "Surface (at initialization) mean hourly airmass ascent rate ensemble std" ;
wvert_sfc_ens_std:units = "m/s" ;
wvert_sfc_ens_std:standard_name = "upward_air_velocity" ;
double wvert_sfc_ens_min(time, trajectory_time) ;
wvert_sfc_ens_min:_FillValue = -9999. ;
wvert_sfc_ens_min:long_name = "Surface (at initialization) mean hourly airmass ascent rate ensemble min" ;
wvert_sfc_ens_min:units = "m/s" ;
wvert_sfc_ens_min:standard_name = "upward_air_velocity" ;
double wvert_sfc_ens_max(time, trajectory_time) ;
wvert_sfc_ens_max:_FillValue = -9999. ;
wvert_sfc_ens_max:long_name = "Surface (at initialization) mean hourly airmass ascent rate ensemble max" ;
wvert_sfc_ens_max:units = "m/s" ;
wvert_sfc_ens_max:standard_name = "upward_air_velocity" ;

double height_to_pblh_ratio_sfc_ens_mean(time, trajectory_time) ;
height_to_pblh_ratio_sfc_ens_mean:_FillValue = -9999. ;
height_to_pblh_ratio_sfc_ens_mean:long_name = "Surface (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble mean" ;
height_to_pblh_ratio_sfc_ens_mean:units = "1" ;
double height_to_pblh_ratio_sfc_ens_std(time, trajectory_time) ;
height_to_pblh_ratio_sfc_ens_std:_FillValue = -9999. ;
height_to_pblh_ratio_sfc_ens_std:long_name = "Surface (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble std" ;
height_to_pblh_ratio_sfc_ens_std:units = "1" ;
double height_to_pblh_ratio_sfc_ens_min(time, trajectory_time) ;
height_to_pblh_ratio_sfc_ens_min:_FillValue = -9999. ;
height_to_pblh_ratio_sfc_ens_min:long_name = "Surface (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble min" ;
height_to_pblh_ratio_sfc_ens_min:units = "1" ;
double height_to_pblh_ratio_sfc_ens_max(time, trajectory_time) ;
height_to_pblh_ratio_sfc_ens_max:_FillValue = -9999. ;
height_to_pblh_ratio_sfc_ens_max:long_name = "Surface (at initialization) airmass height-to-pblh ratio (>1 - airmass above pbl; <1 - airmass within pbl) ensemble max" ;
height_to_pblh_ratio_sfc_ens_max:units = "1" ;
int vert_layer(vert_layer) ;
vert_layer:long_name = "Vertical layer index (bottom-up)" ;
vert_layer:units = "1" ;
double sgs_orography_angle(time, trajectory_time, vert_layer) ;
sgs_orography_angle:_FillValue = -9999. ;
sgs_orography_angle:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_angle:units = "radian" ;
double sgs_orography_angle_ens_mean(time, trajectory_time, vert_layer) ;
sgs_orography_angle_ens_mean:_FillValue = -9999. ;
sgs_orography_angle_ens_mean:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_angle_ens_mean:units = "radian" ;
double sgs_orography_angle_sfc(time, trajectory_time) ;
sgs_orography_angle_sfc:_FillValue = -9999. ;
sgs_orography_angle_sfc:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along surface trajectory)" ;
sgs_orography_angle_sfc:units = "radian" ;
double sgs_orography_angle_sfc_ens_mean(time, trajectory_time) ;
sgs_orography_angle_sfc_ens_mean:_FillValue = -9999. ;
sgs_orography_angle_sfc_ens_mean:long_name = "Terrain orientation in the horizontal plane (from a bird\'s-eye view) relative to an eastwards axis in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along surface ensemble mean trajectory)" ;

sgs_orography_angle_sfc_ens_mean:units = "radian" ;
double sgs_orography_anisotropy(time, trajectory_time, vert_layer) ;
sgs_orography_anisotropy:_FillValue = -9999. ;
sgs_orography_anisotropy:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_anisotropy:units = "1" ;
double sgs_orography_anisotropy_ens_mean(time, trajectory_time, vert_layer) ;
sgs_orography_anisotropy_ens_mean:_FillValue = -9999. ;
sgs_orography_anisotropy_ens_mean:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_anisotropy_ens_mean:units = "1" ;
double sgs_orography_anisotropy_sfc(time, trajectory_time) ;
sgs_orography_anisotropy_sfc:_FillValue = -9999. ;
sgs_orography_anisotropy_sfc:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along surface trajectory)" ;
sgs_orography_anisotropy_sfc:units = "1" ;
double sgs_orography_anisotropy_sfc_ens_mean(time, trajectory_time) ;
sgs_orography_anisotropy_sfc_ens_mean:_FillValue = -9999. ;
sgs_orography_anisotropy_sfc_ens_mean:long_name = "Terrain distortion from a circle in the horizontal plane (from a bird\'s-eye view) in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along surface ensemble mean trajectory)" ;
sgs_orography_anisotropy_sfc_ens_mean:units = "1" ;
double sgs_orography_std(time, trajectory_time, vert_layer) ;
sgs_orography_std:_FillValue = -9999. ;
sgs_orography_std:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_std:units = "m" ;
double sgs_orography_std_ens_mean(time, trajectory_time, vert_layer) ;
sgs_orography_std_ens_mean:_FillValue = -9999. ;
sgs_orography_std_ens_mean:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_std_ens_mean:units = "m" ;
double sgs_orography_std_sfc(time, trajectory_time) ;
sgs_orography_std_sfc:_FillValue = -9999. ;
sgs_orography_std_sfc:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along surface trajectory)" ;
sgs_orography_std_sfc:units = "m" ;
double sgs_orography_std_sfc_ens_mean(time, trajectory_time) ;
sgs_orography_std_sfc_ens_mean:_FillValue = -9999. ;

sgs_orography_std_sfc_ens_mean:long_name = "Standard deviation of orography within a grid box in airmass column; calculated using a minimum orographic feature horizontal scale of 5 km (along surface ensemble mean trajectory)" ;
sgs_orography_std_sfc_ens_mean:units = "m" ;
double sgs_orography_slope(time, trajectory_time, vert_layer) ;
sgs_orography_slope:_FillValue = -9999. ;
sgs_orography_slope:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km" ;
sgs_orography_slope:units = "1" ;
double sgs_orography_slope_ens_mean(time, trajectory_time, vert_layer) ;
sgs_orography_slope_ens_mean:_FillValue = -9999. ;
sgs_orography_slope_ens_mean:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along ensemble mean trajectory)" ;
sgs_orography_slope_ens_mean:units = "1" ;
double sgs_orography_slope_sfc(time, trajectory_time) ;
sgs_orography_slope_sfc:_FillValue = -9999. ;
sgs_orography_slope_sfc:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along surface trajectory)" ;
sgs_orography_slope_sfc:units = "1" ;
double sgs_orography_slope_sfc_ens_mean(time, trajectory_time) ;
sgs_orography_slope_sfc_ens_mean:_FillValue = -9999. ;
sgs_orography_slope_sfc_ens_mean:long_name = "Slope of orography within a grid box in airmass column (e.g., 0 - flat, 0.5 - 45 degree slope); calculated using a minimum orographic feature horizontal scale of 5 km (along surface ensemble mean trajectory)" ;
sgs_orography_slope_sfc_ens_mean:units = "1" ;
double land_sea_mask(time, trajectory_time, vert_layer) ;
land_sea_mask:_FillValue = -9999. ;
land_sea_mask:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land)" ;
land_sea_mask:units = "1" ;
double land_sea_mask_ens_mean(time, trajectory_time, vert_layer) ;
land_sea_mask_ens_mean:_FillValue = -9999. ;
land_sea_mask_ens_mean:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along ensemble mean trajectory)" ;
land_sea_mask_ens_mean:units = "1" ;
double land_sea_mask_sfc(time, trajectory_time) ;
land_sea_mask_sfc:_FillValue = -9999. ;
land_sea_mask_sfc:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along surface trajectory)" ;
land_sea_mask_sfc:units = "1" ;
double land_sea_mask_sfc_ens_mean(time, trajectory_time) ;
land_sea_mask_sfc_ens_mean:_FillValue = -9999. ;

land_sea_mask_sfc_ens_mean:long_name = "Land-sea mask in airmass column (area fraction per ERA5 ~31 km grid-cell; 0 - open water, 1 - all land) (along surface ensemble mean trajectory)" ;

land_sea_mask_sfc_ens_mean:units = "1" ;

float high_vegetation_type(time, trajectory_time, vert_layer) ;

high_vegetation_type:_FillValue = -9999.f ;

high_vegetation_type:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation)" ;

high_vegetation_type:units = "1" ;

float high_vegetation_type_ens_mean(time, trajectory_time, vert_layer) ;

high_vegetation_type_ens_mean:_FillValue = -9999.f ;

high_vegetation_type_ens_mean:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along ensemble mean trajectory)" ;

high_vegetation_type_ens_mean:units = "1" ;

float high_vegetation_type_sfc(time, trajectory_time) ;

high_vegetation_type_sfc:_FillValue = -9999.f ;

high_vegetation_type_sfc:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along surface trajectory)" ;

high_vegetation_type_sfc:units = "1" ;

float high_vegetation_type_sfc_ens_mean(time, trajectory_time) ;

high_vegetation_type_sfc_ens_mean:_FillValue = -9999.f ;

high_vegetation_type_sfc_ens_mean:long_name = "High vegetation type in airmass column out of 6 values: 3 = Evergreen needleleaf trees, 4 = Deciduous needleleaf trees, 5 = Deciduous broadleaf trees, 6 = Evergreen broadleaf trees, 18 = Mixed forest/woodland, 19 = Interrupted forest. (0 indicates lack of high vegetation) (along surface ensemble mean trajectory)" ;

high_vegetation_type_sfc_ens_mean:units = "1" ;

float low_vegetation_type(time, trajectory_time, vert_layer) ;

low_vegetation_type:_FillValue = -9999.f ;

low_vegetation_type:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation)" ;

low_vegetation_type:units = "1" ;

float low_vegetation_type_ens_mean(time, trajectory_time, vert_layer) ;

low_vegetation_type_ens_mean:_FillValue = -9999.f ;

low_vegetation_type_ens_mean:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along ensemble mean trajectory)" ;

low_vegetation_type_ens_mean:units = "1" ;

float low_vegetation_type_sfc(time, trajectory_time) ;
low_vegetation_type_sfc:_FillValue = -9999.f ;
low_vegetation_type_sfc:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along surface trajectory)" ;
low_vegetation_type_sfc:units = "1" ;
float low_vegetation_type_sfc_ens_mean(time, trajectory_time) ;
low_vegetation_type_sfc_ens_mean:_FillValue = -9999.f ;
low_vegetation_type_sfc_ens_mean:long_name = "Low vegetation type in airmass column out of 10 values: 1 = Crops, Mixed farming, 2 = Grass, 7 = Tall grass, 9 = Tundra, 10 = Irrigated crops, 11 = Semidesert, 13 = Bogs and marshes, 16 = Evergreen shrubs, 17 = Deciduous shrubs, 20 = Water and land mixtures. (0 indicates lack of low vegetation) (along surface ensemble mean trajectory)" ;
low_vegetation_type_sfc_ens_mean:units = "1" ;
double high_vegetation_cover(time, trajectory_time, vert_layer) ;
high_vegetation_cover:_FillValue = -9999. ;
high_vegetation_cover:long_name = "High vegetation cover fraction in airmass column" ;
high_vegetation_cover:units = "1" ;
double high_vegetation_cover_ens_mean(time, trajectory_time, vert_layer) ;
high_vegetation_cover_ens_mean:_FillValue = -9999. ;
high_vegetation_cover_ens_mean:long_name = "High vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
high_vegetation_cover_ens_mean:units = "1" ;
double high_vegetation_cover_sfc(time, trajectory_time) ;
high_vegetation_cover_sfc:_FillValue = -9999. ;
high_vegetation_cover_sfc:long_name = "High vegetation cover fraction in airmass column (along surface trajectory)" ;
high_vegetation_cover_sfc:units = "1" ;
double high_vegetation_cover_sfc_ens_mean(time, trajectory_time) ;
high_vegetation_cover_sfc_ens_mean:_FillValue = -9999. ;
high_vegetation_cover_sfc_ens_mean:long_name = "High vegetation cover fraction in airmass column (along surface ensemble mean trajectory)" ;
high_vegetation_cover_sfc_ens_mean:units = "1" ;
double low_vegetation_cover(time, trajectory_time, vert_layer) ;
low_vegetation_cover:_FillValue = -9999. ;
low_vegetation_cover:long_name = "Low vegetation cover fraction in airmass column" ;
low_vegetation_cover:units = "1" ;
double low_vegetation_cover_ens_mean(time, trajectory_time, vert_layer) ;
low_vegetation_cover_ens_mean:_FillValue = -9999. ;
low_vegetation_cover_ens_mean:long_name = "Low vegetation cover fraction in airmass column (along ensemble mean trajectory)" ;
low_vegetation_cover_ens_mean:units = "1" ;
double low_vegetation_cover_sfc(time, trajectory_time) ;
low_vegetation_cover_sfc:_FillValue = -9999. ;

low_vegetation_cover_sfc:long_name = "Low vegetation cover fraction in airmass column (along surface trajectory)" ;
low_vegetation_cover_sfc:units = "1" ;
double low_vegetation_cover_sfc_ens_mean(time, trajectory_time) ;
low_vegetation_cover_sfc_ens_mean:_FillValue = -9999. ;
low_vegetation_cover_sfc_ens_mean:long_name = "Low vegetation cover fraction in airmass column (along surface ensemble mean trajectory)" ;
low_vegetation_cover_sfc_ens_mean:units = "1" ;
float soil_type(time, trajectory_time, vert_layer) ;
soil_type:_FillValue = -9999.f ;
soil_type:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point)" ;
soil_type:units = "1" ;
soil_type:standard_name = "soil_type" ;
float soil_type_ens_mean(time, trajectory_time, vert_layer) ;
soil_type_ens_mean:_FillValue = -9999.f ;
soil_type_ens_mean:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along ensemble mean trajectory)" ;
soil_type_ens_mean:units = "1" ;
soil_type_ens_mean:standard_name = "soil_type" ;
float soil_type_sfc(time, trajectory_time) ;
soil_type_sfc:_FillValue = -9999.f ;
soil_type_sfc:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along surface trajectory)" ;
soil_type_sfc:units = "1" ;
soil_type_sfc:standard_name = "soil_type" ;
float soil_type_sfc_ens_mean(time, trajectory_time) ;
soil_type_sfc_ens_mean:_FillValue = -9999.f ;
soil_type_sfc_ens_mean:long_name = "Soil type in airmass column out of 7 values: 1 = Coarse, 2 = Medium, 3 = Medium fine, 4 = Fine, 5 = Very fine, 6: Organic, 7: Tropical organic. (0 indicates non-land point) (along surface ensemble mean trajectory)" ;
soil_type_sfc_ens_mean:units = "1" ;
soil_type_sfc_ens_mean:standard_name = "soil_type" ;
double sea_ice_cover(time, trajectory_time, vert_layer) ;
sea_ice_cover:_FillValue = -9999. ;
sea_ice_cover:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column" ;
sea_ice_cover:units = "1" ;
sea_ice_cover:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_ens_mean(time, trajectory_time, vert_layer) ;
sea_ice_cover_ens_mean:_FillValue = -9999. ;
sea_ice_cover_ens_mean:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along ensemble mean trajectory)" ;

sea_ice_cover_ens_mean:units = "1" ;
sea_ice_cover_ens_mean:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_sfc(time, trajectory_time) ;
sea_ice_cover_sfc:_FillValue = -9999. ;
sea_ice_cover_sfc:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along surface trajectory)" ;
sea_ice_cover_sfc:units = "1" ;
sea_ice_cover_sfc:standard_name = "sea_ice_area_fraction" ;
double sea_ice_cover_sfc_ens_mean(time, trajectory_time) ;
sea_ice_cover_sfc_ens_mean:_FillValue = -9999. ;
sea_ice_cover_sfc_ens_mean:long_name = "Sea-ice cover fraction (based on monthly means) in airmass column (along surface ensemble mean trajectory)" ;
sea_ice_cover_sfc_ens_mean:units = "1" ;
sea_ice_cover_sfc_ens_mean:standard_name = "sea_ice_area_fraction" ;
double lat_mean_aafnav(time) ;
lat_mean_aafnav:_FillValue = -9999. ;
lat_mean_aafnav:long_name = "North latitude, 1-hour mean (averaging window starting at the corresponding timestamp)" ;
lat_mean_aafnav:units = "degree_N" ;
lat_mean_aafnav:valid_min = -90. ;
lat_mean_aafnav:valid_max = 90. ;
lat_mean_aafnav:standard_name = "latitude" ;
double lat_std_aafnav(time) ;
lat_std_aafnav:_FillValue = -9999. ;
lat_std_aafnav:long_name = "North latitude, 1-hour std (averaging window starting at the corresponding timestamp)" ;
lat_std_aafnav:units = "degree_N" ;
lat_std_aafnav:valid_min = -90. ;
lat_std_aafnav:valid_max = 90. ;
lat_std_aafnav:standard_name = "latitude" ;
double lat_min_aafnav(time) ;
lat_min_aafnav:_FillValue = -9999. ;
lat_min_aafnav:long_name = "North latitude, 1-hour min (averaging window starting at the corresponding timestamp)" ;
lat_min_aafnav:units = "degree_N" ;
lat_min_aafnav:valid_min = -90. ;
lat_min_aafnav:valid_max = 90. ;
lat_min_aafnav:standard_name = "latitude" ;
double lat_max_aafnav(time) ;
lat_max_aafnav:_FillValue = -9999. ;
lat_max_aafnav:long_name = "North latitude, 1-hour max (averaging window starting at the corresponding timestamp)" ;
lat_max_aafnav:units = "degree_N" ;
lat_max_aafnav:valid_min = -90. ;
lat_max_aafnav:valid_max = 90. ;
lat_max_aafnav:standard_name = "latitude" ;

double lon_mean_aafnav(time) ;
lon_mean_aafnav:_FillValue = -9999. ;
lon_mean_aafnav:long_name = "East longitude, 1-hour mean (averaging window starting at the corresponding timestamp)" ;
lon_mean_aafnav:units = "degree_E" ;
lon_mean_aafnav:valid_min = -180. ;
lon_mean_aafnav:valid_max = 180. ;
lon_mean_aafnav:standard_name = "longitude" ;
double lon_std_aafnav(time) ;
lon_std_aafnav:_FillValue = -9999. ;
lon_std_aafnav:long_name = "East longitude, 1-hour std (averaging window starting at the corresponding timestamp)" ;
lon_std_aafnav:units = "degree_E" ;
lon_std_aafnav:valid_min = -180. ;
lon_std_aafnav:valid_max = 180. ;
lon_std_aafnav:standard_name = "longitude" ;
double lon_min_aafnav(time) ;
lon_min_aafnav:_FillValue = -9999. ;
lon_min_aafnav:long_name = "East longitude, 1-hour min (averaging window starting at the corresponding timestamp)" ;
lon_min_aafnav:units = "degree_E" ;
lon_min_aafnav:valid_min = -180. ;
lon_min_aafnav:valid_max = 180. ;
lon_min_aafnav:standard_name = "longitude" ;
double lon_max_aafnav(time) ;
lon_max_aafnav:_FillValue = -9999. ;
lon_max_aafnav:long_name = "East longitude, 1-hour max (averaging window starting at the corresponding timestamp)" ;
lon_max_aafnav:units = "degree_E" ;
lon_max_aafnav:valid_min = -180. ;
lon_max_aafnav:valid_max = 180. ;
lon_max_aafnav:standard_name = "longitude" ;
double alt_mean_aafnav(time) ;
alt_mean_aafnav:_FillValue = -9999. ;
alt_mean_aafnav:long_name = "Altitude above mean sea level, 1-hour mean (averaging window starting at the corresponding timestamp)" ;
alt_mean_aafnav:units = "m" ;
alt_mean_aafnav:standard_name = "altitude" ;
double alt_std_aafnav(time) ;
alt_std_aafnav:_FillValue = -9999. ;
alt_std_aafnav:long_name = "Altitude above mean sea level, 1-hour std (averaging window starting at the corresponding timestamp)" ;
alt_std_aafnav:units = "m" ;
alt_std_aafnav:standard_name = "altitude" ;
double alt_min_aafnav(time) ;
alt_min_aafnav:_FillValue = -9999. ;

alt_min_aafnav:long_name = "Altitude above mean sea level, 1-hour min (averaging window starting at the corresponding timestamp)" ;
alt_min_aafnav:units = "m" ;
alt_min_aafnav:standard_name = "altitude" ;
double alt_max_aafnav(time) ;
alt_max_aafnav:_FillValue = -9999. ;
alt_max_aafnav:long_name = "Altitude above mean sea level, 1-hour max (averaging window starting at the corresponding timestamp)" ;
alt_max_aafnav:units = "m" ;
alt_max_aafnav:standard_name = "altitude" ;
double aafnav_01_alt_percentile(time) ;
aafnav_01_alt_percentile:_FillValue = -9999. ;
aafnav_01_alt_percentile:long_name = "AAF 1st percentile altitude during 1-hour window (used in initialization)" ;
aafnav_01_alt_percentile:units = "m" ;
double aafnav_99_alt_percentile(time) ;
aafnav_99_alt_percentile:_FillValue = -9999. ;
aafnav_99_alt_percentile:long_name = "AAF 99th percentile altitude during 1-hour window (used in initialization)" ;
aafnav_99_alt_percentile:units = "m" ;
int lowest_aaf_alt_below_era5_sfc_flag(time) ;
lowest_aaf_alt_below_era5_sfc_flag:_FillValue = -9999 ;
lowest_aaf_alt_below_era5_sfc_flag:long_name = "AAF minimum altitude lower than interpolated ERA5 sfc altitude during 1-hour window (setting surface altitude as lowest HYSPLIT run altitude)" ;
lowest_aaf_alt_below_era5_sfc_flag:units = "1" ;
double effective_aaf_ens_dx(time) ;
effective_aaf_ens_dx:_FillValue = -9999. ;
effective_aaf_ens_dx:long_name = "Effective dx used in AAF trajectory ensemble (max[3.75, σ], where σ is the facility longitude standard deviation (translated to kilometers) during the 1-hour window used in initialization)" ;
effective_aaf_ens_dx:units = "km" ;
double effective_aaf_ens_dy(time) ;
effective_aaf_ens_dy:_FillValue = -9999. ;
effective_aaf_ens_dy:long_name = "Effective dy used in AAF trajectory ensemble (max[3.75, σ], where σ is the facility latitude standard deviation (translated to kilometers) during the 1-hour window used in initialization)" ;
effective_aaf_ens_dy:units = "km" ;
int time_aaf(time_aaf) ;
time_aaf:long_name = "(AAF) Seconds since 2024-05-18 00:00:00 0:00" ;
time_aaf:units = "Time offset from midnight" ;
double lat_aaf(time_aaf) ;
lat_aaf:_FillValue = -9999. ;
lat_aaf:long_name = "(AAF) North latitude" ;
lat_aaf:units = "degree_N" ;
double lon_aaf(time_aaf) ;

```
        lon_aaf:_FillValue = -9999. ;
        lon_aaf:long_name = "(AAF) East longitude" ;
        lon_aaf:units = "degree_E" ;
    double alt_aaf(time_aaf) ;
        alt_aaf:_FillValue = -9999. ;
        alt_aaf:long_name = "(AAF) Altitude above mean sea level" ;
        alt_aaf:units = "m" ;
    double time(time) ;
        time:long_name = "Time in seconds since volume start" ;
        time:units = "seconds since 2024-05-18 00:00:00 0:00" ;
    double time_offset(time) ;
        time_offset:_FillValue = -9999. ;
        time_offset:long_name = "Time offset from base_time" ;
        time_offset:units = "seconds since 2024-05-18 00:00:00 0:00" ;
    int base_time ;
        base_time:_FillValue = -9999 ;
        base_time:long_name = "Base time in Epoch" ;
        base_time:units = "seconds since 1970-01-01 0:00:00 0:00" ;
    double lat(time) ;
        lat:_FillValue = -9999. ;
        lat:long_name = "latitude" ;
        lat:units = "degree_N" ;
    double lon(time) ;
        lon:_FillValue = -9999. ;
        lon:long_name = "longitude" ;
        lon:units = "degree_E" ;
    double alt(time) ;
        alt:_FillValue = -9999. ;
        alt:long_name = "Altitude above mean sea level" ;
        alt:units = "m" ;

// global attributes:
        :mode = "aaf" ;
        :ensemble_size = "25" ;
        :ensemble_size_sfc = "25" ;
        :default_ensemble_dx_from_center = "[-7.5  -3.75  0.    3.75  7.5 ] km" ;
        :default_ensemble_dy_from_center = "[-7.5  -3.75  0.    3.75  7.5 ] km" ;
        :ensemble_dz_from_center = "[0] m" ;
        :used_era5_resolution = "0.25 degrees" ;
        :total_trajectory_hours = "[-120]" ;
        :number_of_starting_center_points_per_time_step = "[11 11 11 11 11]" ;
        :title = "Back trajectories for AAF (crewed and uncrewed) flights" ;
        :summary = "The ARMTRAJ VAP provides trajectory datasets initialized at ARM
deployment coordinates and configured using ARM datasets. The trajectory datasets support aerosol,
cloud, and planetary boundary layer research. Trajectory calculations use the HYSPLIT model informed
by the ERA5 reanalysis dataset at its highest spatial resolution (~31 km). For each sample in each of the
```

datasets, HYSPLIT also runs at multiple starting locations surrounding ARM deployments, enabling an ensemble of runs from which the mean and variability (estimated uncertainty) of each sample\'s trajectory coordinates, thermodynamic properties, or other fields are reported." ;

:keywords = "ARM, airborne measurements, ARM Aerial Facility, AAF, clouds, aerosol, PBL, trajectories, Lagrangian analysis" ;

:doi = "10.5439/2473260" ;

:location_description = "Southern Great Plains (SGP), Lamont, Oklahoma" ;

:dod_version = "v1.0" ;

:command_line = "python armtraj_run.py" ;

:datastream = "sgparmtrajaafU2.c1" ;

:data_level = "c1" ;

:facility_id = "U2" ;

:site_id = "sgp" ;

:platform_id = "armtrajaaf" ;

:history = "created by user isilber1 on machine dev-proc2 at 24-Oct-2024,16:05:56" ;

}

www.arm.gov

U.S. DEPARTMENT OF
ENERGY

Office of Science