APT-HDF5 File Specification

Summary

This is an open and inter-operable file format for storing the data acquired in an APT experiment. This format allows for data to be transferred seamlessly between labs and instrument models, independent of the equipment in use.

This specification lays out the requirements for storing data from Atom Probe Tomography (APT) experiments into a common, instrument-agnostic format for consumption by data processing and analysis applications. The latest version can be found online [Specification]

Each file is to be an HDF5 file (a commonly used binary file container format with compression), following the format laid out below. The files must verify as valid against the latest revision of the TC provided verification tool [TCVerification].

The purpose of this specification is to utilise existing file storage technologies (HDF) that provide high performance primitives for data storage, removing any ambiguities about data storage and retrieval. This specification thus provides a basic container that allows for a single atom probe experiment to be described, as collected by an APT system.

Why do we need this?

The specification is meant to support and facilitate current and future data retrieval and data sharing with a publicly and openly documented efficient file format and open-source read/write routines that are instrument neutral and permanently available. The specified HDF5 format can be supported by a hashing algorithm that creates a hashing value for uniquely identifying a dataset and ensure its immutability.

Overview

The file container has 4 regions (groups) which contain fields, the fields contain data. The different regions are: ExperimentContext, ToolEnvironment, ToolStateAndSettings and ExperimentResults. These regions and the fields they contain are defined below. Each field has a name and a formatted data entry. Each data entry has a specified shape and data type defining what data can be stored in each field. If applicable, the physical units of the measurement (all SI) are also indicated. The fields are plain English descriptions of what they contain, such as ExperimentStartDateUTC = the start time of the experiment in coordinated universal time (UTC). The different data types used are explained below.

Changelog

This specification is a living document. The changelogs indicate any additions, deletions or modifications. This version is the first draft for community feedback.

Notes

- Additional fields are strongly discouraged, and the writer should not assume that downstream consumers of the file will use this information.
    - Additional fields must not change the output of code run on this data, with respect to any output generated without this information, i.e. any additional

fields outside this specification can only be used for context/metadata purposes to enhance the final result.

- Fields not marked as optional are required to be present.
- All units in this file are to be specified in SI units
- Keywords such as "must" "required" "should", etc are as per RFC-2119 [RFC2119].

Data Types

The data types used in this document are specified below:

- *string* types : Must be valid UTF-8 format strings
- *real* types : For storing floating point data - , these must be IEEE-754 floating point variables, either 32 or 64 bits width ("Single"/"double" precision). NaN and inf values of any kind are not permitted unless explicitly specified.
- *integer* : an integral data type, of either 32 or 64 bits in length, unsigned unless otherwise specified.
- *array n x m*: where content of array not specified, this is an array of type *real*, with an HDF container size of shape n by m. The specific entry in the below for a given type should specify the array direction (corresponding dimension for n and m).

Coordinate systems

- Several coordinate systems are defined by this document. Each coordinate system must be right handed and Cartesian coordinates with base vectors defining X,Y,Z directions and covering 3D Euclidean space, unless otherwise specified.
    - *Laboratory space*. This is the space that is set by the chassis of the instrument. The Z direction must be reasonably parallel to gravity (+ve defined to be gravity vector pointing towards lowest ground), but can be defined to be a direction that is nominally parallel to gravity during an experiment. The origin must be placed within a bounding box of the chassis.
    - *Tip space* : The space occupied by a tip in the neutral position when ready for analysis. Z+ should be located in the direction of the needle (apex is Z+ from needle centreline).
        - If the specimen moves relative to the laboratory frame, and the electrode does not, or if no electrode is present, then the space should be defined such that when the tip is moved physically, it also moves through tip space.
        - If the electrode moves relative to the laboratory frame, then the space should be defined such that, in tip space, the electrode's position does not change.
        - The transformation between laboratory space and Tip space must be describable by a fixed 3x3 transformation matrix.
    - *Detector space:* This is a 2D space only, and contains X+ and Y+ directions. X+ and Y+ should indicate primary directions on the detector, and should be, for an equivalent straight-flight-path configuration, such that X+ and Y+ is matched to that of tip space.

    - *Reconstruction space* : The space occupied by a correctly reconstructed dataset. X+ and Y+ should correspond to X+ and Y+ in the detector space. Z+ should be such that the needle centre line is normal to the detector position. The origin shall be placed at the tip apex.
    - *Laser space*: The coordinate frame describing the impinging wavefront on the sample. Z+ is the vector of the propagating wavefront. X+ is the orthogonal component of the tip direction within the tip-laser plane. The origin shall be placed at the best estimate for tip apex position at the start of the experiment.
- Spatial transformations are always active transformations; i.e. the location and direction of a vector in one coordinate system is expressed by the following transformation matrix, T
    - $P_{transformed} = TP_{original}$

Regions

“Regions” help to structure data items that belong together, (termed “groups” in HDF5). Regions contain either data or metadata, e.g. arrays of temperature data (real), or strings , such as for a specimen name.

ExperimentContext

This section contains metadata regarding the samples to be analysed, any counter electrode, and the governing standard

- ExperimentType : *String*
    - Must contain “AtomProbeTomographyExperiment”
- SampleDescription : *String*
    - Free text, may contain any description up to 5MB in size. Information must be human readable.
        - Example : “Stainless steel, heat treated at 300C for 2 hrs, as received condition”.
- SampleName : *String*
    - Free text, must be < 200 characters, should be human readable.
- SampleUniqueIdentifier : *String*
    - Must contain a unique identifier that can be used for sample tracking purposes. Maximum 100 charachters
- ApertureType: *String*
    - Specifies the type of aperture/counter-electrode that is in use. Currently this may be:
        - “none” - No aperture is present in the experiment
        - “conical” - a conical aperture with a circular hole
        - “feedthrough” - an aperture where the sample protrudes through a circular hole
        - “custom” - A user modified aperture, which is otherwise non-standard
- ApertureUniqueIdentifier: *String*

- A unique identifier that can be used for aperture. Maximum 100 chars. Empty if ApertureType is “none”.
- Version : *String*
  - Standards version for this HDF5 file. Must be of the form YYYY-MM[.X], where Y stands for Year (digit), M for month (digit) and X (digit) is an optional incremental revision number. Currently “2020-06”.
  - Valid range YYYY>=2020, MM <=12, X>=1

ToolEnvironment

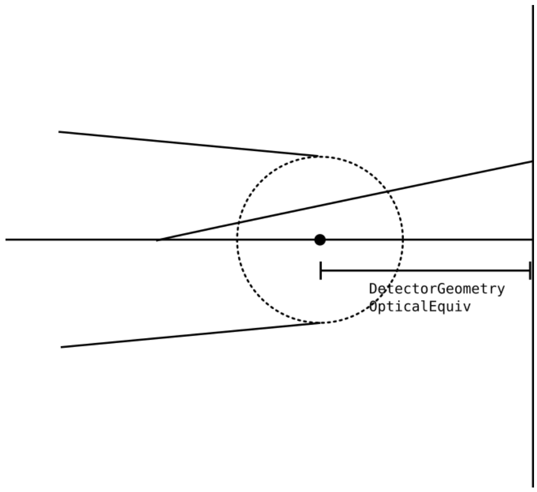
This section contains metadata regarding the environment within which the system operates

- ExperimentStartDateUTC: *String*
  - ISO8601 formatted string with minimum of second resolution . Must refer to UTC time that experiment started. Experiment is to be considered started when pulsing is initiated to the sample or electrode.
  - Range: Dates must not be in the future.
- ExperimentStartDateLocal : *String*
  - ISO8601 formatted string with minimum of second resolution . Must refer to local time, and the same instant in time as ExperimentStartDateUTC
  - Range: Date must not be in the future with respect to local time, excepting for local clock changes, e.g. daylight savings.
- ExperimentEndDateUTC : *String*
  - ISO8601 formatted string with minimum of second resolution. Must refer to UTC time that experiment was completed. Experiment is to be considered when sample pulsing stops, and should not wait for further user input.
  - Range: Dates must not be in the future.
- ExperimentEndDateLocal : *String*
  - ISO8601 formatted string with minimum of second resolution . Must refer to local time, and the same instant in time as ExperimentEndDateUTC
  - Range: Date must not be in the future with respect to local time, excepting for local clock changes, e.g. daylight savings.

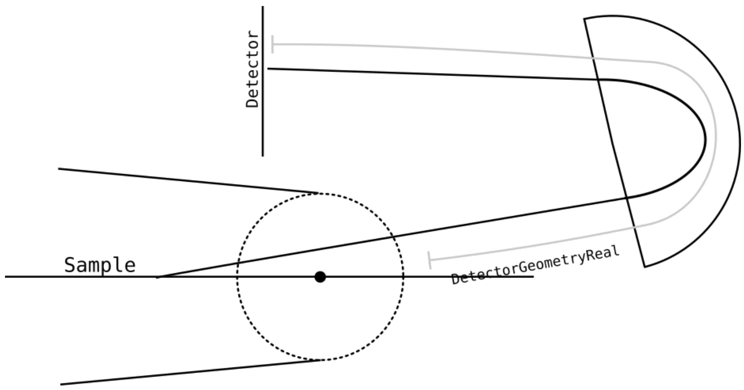
ToolStateAndSettings

This section contains metadata regarding the acquiring instrument

- FlightPathSpatial : *Real,* (m)
  - Virtual imaging distance between sample and detector, from centre of tip sphere to detector centre, for an ion leaving in the forwards direction from a tip aligned at a neutral rotation. Units are in m. The virtual imaging distance is defined as the distance that would be required to achieve a stereographic projection as if the tip was being imaged in a free-flight configuration.
  - Valid Range : > 0



- DetectorGeometryOpticalEquiv: *Real, array 1x2,* (m)
    - The effective size of the detector, at the virtual imaging distance, such that the optical path can, for reconstruction purposes, be approximated by a fixed detector
    - Valid range > 0
- FlightPathTiming: *Real,* (m)
    - The distance that an ion that leaves the tip normally will traverse, on average, before striking the detector, along the electrostatic trajectory followed by an ion that leaves the central position of the tip
    - Valid range > 0



- DetectorType: *String*
    - The detector technology in use, maximum 500 chars. The currently permitted values are:
        - "DelayLine" - a wire-meander detector
        - "WedgeAndStrip" - a wedge based charge discriminating detector
        - "Camera" - an optically driven detector system
        - "Other" + "/" description (e.g. other/future detector types), eg "FutureDetector/A detector that operates as described in the paper by Author et al, 2603, Journal of the Future
- DetectorReadout: *String*
    - The detector operating method used to convert the detector potentials into hit timings, current values allowed are:
        - "Threshold"
        - "Discriminator"

        - “Time-resolved”
        - “Unspecified”
- DetectorResolution: *Real, 1xn array* (m)
    - The minimum resolvable spacing on the detector. A single value may be used.
    - Valid range > 0
- DetectorSize: *Real, 1x2 array,* (m)
    - The minimum size of the rectangular bounding box that can contain the complete active detector area. The distances should wherever possible be given in the laboratory “X+” (1^{st} entry) and “Z+” frame (2^{nd} entry)
    - Valid range for each entry > 0
- InstrumentIdentifier: *String*
    - A unique identifier that indicates the manufacturer, model and serial information. This must not change for any given machine configuration at any time.
        - Example : “CompanyID.MyAPT 123”
        - A company should use a persistent identifier, that will survive product or company renamings.
            - Each identifier must be unique per machine, including any modifications that materially alter the reconstruction of the data.
- LaserIncidence: *Real,* 1x3 array
    - A vector in tip space that describes the direction in which the laser impinges on the sample, i.e. the mean vector parallel to the laser propagation direction. Vector must be normalised, such that ||v|| =1, within numerical precision.
- LaserWavelength: *Real,* (m)
    - The wavelength of the laser beam
    - Valid range > 0
- ReflectronInfo: *String*
    - Indicates the presence and type of reflectron device. Valid values are “Linear”, “Spherical” and “None”
- LabToTipSpace: *Real 4x4 array*
    - rotation and translation matrix (3x3 upper left triangular component is rotation, Det(matrix) = 1). Remainder is the translation component.
    - matrix indicates the rotation necessary to transform that should take the dataset from the laboratory space to the tip space.
    - Translational components of matrix in (m).
- TipToLaserSpace : *Real, 3x3 array*
    - Rotation matrix that converts from tip space to the nominal impingement vector of the laser.

ExperimentResults

This section contains acquired data from the undertaken experiment

- PulseNumber: *Integer, array, 1xn (64 bit integer)*
    - Absolute number of pulses starting from the beginning of the experiment, one per ion event
- DetectorHitPositions: *Real 2xn array* (m)
    - Contains the X-Y hit positions, as measured by the detector, without any corrections
    - The (0,0) position must correspond to the centre of the detector.
    - Valid range: DetectorSize/2>||X,Y||
- LaserEnergy: *array, 1xn*, (J)
    - The estimated energy of the incoming laser pulse packet onto the sample
    - This specifies the best estimate of the total energy of the laser pulse in free-space immediately prior to hitting the sample.
    - Empty if laser is not used.
- LaserPosition: *Real array, 2xn or empty*, (m)
    - The estimated X-Y position, as viewed in the laser frame, tip apex to the right (+ve X), of the centre of mass of the laser spot.
    - This field may have zero entries if the laser is not used in the experiment
- PulseFrequency : *Real array, 2xn* (Hz)
    - This is the frequency of the high voltage or laser pulser.
        - first entry : first pulse number where the spacing between this and all subsequent pulses are considered to be at the selected frequency. Each first entry must be strictly increasing in value
        - The second entry contains the frequency value
- StandingVoltage : *array, 1xn,* (V)
    - The standing voltage applied to the sample, relative to system [component electrical ground] ground.
- PulseFraction: *Real, 1xn array* (-)
    - If a standing voltage is applied, this gives nominal pulse fraction (as a function of standing voltage), Otherwise this field should not be present, or should contain zero only. Values in the range range[0,100].
    - Must have same dimensions as PulseNumber
- ReflectronVoltage: *Real, 1xn array* (V)
    - Must be present if ReflectronInfo is not “None” The maximum voltage applied to the reflectron, relative to system ground.
- StagePosition: *Real 3xn array : 3xn,* (m)
    - The X-Y-Z coordinates of the stage in tip space. Z+ must correspond to the nominal tip apex direction. X+,Y+ should correspond to X+,Y+
- TimeOfFlight: *Real 1xn, array,* (s)
    - The as-measured time of flight, as measured by the detector and free of any post-corrections.
    - Must have same dimensions as PulseNumber

    - May include primary corrections from instrument, but must exclude any per-experiment user input.
- TimeOfFlightCorrectionModel: *String*
    - This provides information that allows the time of flight correction information to be generated, such as a filename or experiment string which can allow for the linking back to a dataset that can be used to regenerate the TOF corrections applied in the Time-Of-Flight above.
- TipTemperature: *Real, 1xn, array,* (K)
    - The best estimate, at experiment time, for the temperature at the sample base (furthest point along sample apex and holding assembly that is removable from the sample stage).
- TipTempereatureModel: String
    - Short descriptive string, identifying model for estimate of the apex temperature, e.g. “Conduction/radiation”, “calibrated” or “extrapolated”, maximum 200 characters

References

[RFC2119] https://tools.ietf.org/html/rfc2119
[Specification]
https://docs.google.com/document/d/e/2PACX-1vRxcJ_xF_jiNS77CoeZdQdDXD8l2BebL-DoOBkDrAsGTrkArdjLHEMCXAifBieeS8pTO9jJ9xnstKxs/pub
[TCVerification] https://gitlab.com/apt_software_toolbox/apt-hdf5