Organisation of data in standardized HDF5 Brillouin spectra containers

Pierre Bouvet

November 6, 2024

1 Introduction

The evolution of Brillouin Light Scattering (BLS) applications has lead to an increasing number of publications and research on this domain, resulting in a vast collection of data obtained with different techniques and setups by different laboratories. The evolution of this domain has reached the point where to allow it to grow outside of dedicated laboratories, a common approach to storing the data, treating them and interpreting them is needed. In an effort to facilitate this endeavour, we here propose a new structure for data organization based on the HDF5 file format.

This document is here to explicitely define the structure of these files.

2 Raw data

One of the first challenges is to allow all the users of BLS to adopt the HDF5 file format. To do so two things will be needed: a robust structure, and an interface to translate data to this structure seamlessly, robustly and reliably.

2.1 Concept

When acquiring raw spectra, we can reduce the need of the file format to two essential components:

- The informations on the data
- The data themselves

As it is possible to store information in the form of attributes inside the HDF5 file format, we hereby propose to store raw files in a HBF5 file structure as follows:

```
HBH5 file
├─ Attributes (type: HBF5 attributes)
├─ Data1 (type: Dataset)
├─ Data2 (type: Dataset)
└─ ...
```

2.2 File structure

The limiting factor in this idea will mainly come from the name given to the dataset. Two approaches can be adopted here:

- All the data (including the data treated) will be stored in the 'Data' dataset
- We separate the treated data form the 'Data' dataset

Here we propose to use the second solution to distinguish the physical measures we will obtain from the values that will be derived from them. This also allows us to add in the dataset dedicated to the raw data, the calibration curves and impulse responses we might need for the treatment. As such, the structure of the HBF5 file should look like:

```
HBH5 file
├─ Attributes (type: HBF5 attributes)
├─ Raw Data (type: Dataset)
│  ├─ Raw spectrum
│  ├─ Calibration spectrum
│  ├─ Impulse response
│  ├─ Independent coordinates
│  └─ ...
└─ Treated Data (type: Dataset)
```

In this software, we will only focus on the generation of HDF5 data, using callable functions which will be integrable in more developped softwares.

3 List of attributes

In an effort to standardize the names of the attributes inside the header of the file, we propose to rely on a fixed set of names, separated in 3 main categories:

- Measure: all the parameters that are measure-dependent and that are not strictly linked to the tool used to measure the BLS spectra
- Spectrometer: all the parameters that are instrument-dependent. This includes all the choices made at the moment of developping the spectrometer. This separation is particularly interesting for manufacturers of standard tools as they can directly refer the model of their device here and adapt their treatments to this single parameter.
- File properties: all the file-related parameters that are only linked to the numerical storage of the data.

This list is prone to evolve and we encourage everyone interested in this project to share with us the parameters they feel are needed for their particular application in order to add it to the standard. These parameters are for now the ones listed below:

Category	Parameter name	Parameter Unit or Format	Parameter description	Example	V
MEASURE	Sample	text [NA]	The sample being measured	Water	0.1
MEASURE	Date	Year-Month-Day-Hour-Minute-Second	The time when the sample was acquired	24-12-25-00-00-00	0.1
MEASURE	Exposure	float [s]	The exposure on the sample	0.01	0.1
MEASURE	Dimension	integer [NA]	The dimensions of the measure	3	0.1
MEASURE	Sampling_matrix	(n_x, n_y, n_z) [NA]	The number of samples per axis	(100,100,10)	0.1
MEASURE	Sampling_step	(d_x, d_y, d_z) [μm]	The step size for each axis	(10,10,100)	0.1
SPECTROMETER	Type	text [NA]	The type of spectrometer being used	6-pass TFP	0.1
SPECTROMETER	Model	text [NA]	The model of spectrometer being used	JRS-TFP2	0.1
SPECTROMETER	Wavelength	float [nm]	The wavelength at which the measures were made	780	0.1
SPECTROMETER	Confocality	float [AU]	The size of the confocal pinhole given in Airy Units	0.97	0.1
SPECTROMETER	NA_illumination	float [NA]	The effective numerical aperture of the illumination	0.1	0.1
SPECTROMETER	NA_detection	float [NA]	The numerical aperture of the detection lens	0.45	0.1
SPECTROMETER	Detector_model	text [NA]	The manufacturer and model of the detector	Hamamatsu Orca C11440	0.1
SPECTROMETER	Detector_type	text [NA]	The type of detector	CMOS	0.1
SPECTROMETER	Filtering_module	text [NA]	The type of filter being used	Rubidium Cell 15cm	0.1

Category	Parameter name	Parameter Unit or Format	Parameter description	Example	V
SPECTROMETER	Illumination_power	float [mW]	The power of the laser being used to illuminate the sample	25	0.1
SPECTROMETER	Illumination_type	text [NA]	The type of illumination being used	CW	0.1
SPECTROMETER	Laser_model	text [NA]	The manufacturer and model of the laser	Cobolt Flamenco	0.1
SPECTROMETER	Laser_drift	float [MHz/h]	The frequency drift of the laser	100	0.1
SPECTROMETER	Phonons_measured	text [NA]	The phonons probed by the experiment	longitudinal and transverse	0.1
SPECTROMETER	Polarization_probed_analyzed	text [NA]	The polarization probed and analyzed by the experiment	longitudinal and transverse	0.1
SPECTROMETER	Scan_amplitude	float [GHz]	The frequency amplitude of single-channel spectrometers (mainly TFP)	15.02	0.1
SPECTROMETER	Scanning_strategy	text [NA]	The way the scans are performed	point-scan	0.1
SPECTROMETER	Scattering_angles	float [degree]	The geometry of the measure	180	0.1
SPECTROMETER	Spectral_resolution	float [MHz]	The spectral resolution of the spectrometer	15	0.1
SPECTROMETER	Spectrometer_resolution	float [MHz]	The spectrometer resolution	100	0.1
SPECTROMETER	x-Mechanical_resolution	float [μm]	The mechanical resolution along the x axis	1	0.1
SPECTROMETER	y-Mechanical_resolution	float [μm]	The mechanical resolution along the y axis	1	0.1
SPECTROMETER	z-Mechanical_resolution	float [μm]	The mechanical resolution along the z axis	1	0.1

Category	Parameter name	Parameter Unit or Format	Parameter description	Example	V
SPECTROMETER	x-Optical_res olution	float [μm]	The mechanical resolution along the x axis	10	0.1
SPECTROMETER	y-Optical_res olution	float [μm]	The mechanical resolution along the y axis	10	0.1
SPECTROMETER	z-Optical_res olution	float [μm]	The mechanical resolution along the z axis	100	0.1

All these parameters are indicated as attributes following this nomenclature: "Category.Parameter_name" (for example: "MEASURE.Sample").

4 Organization of the dimensions of the raw data

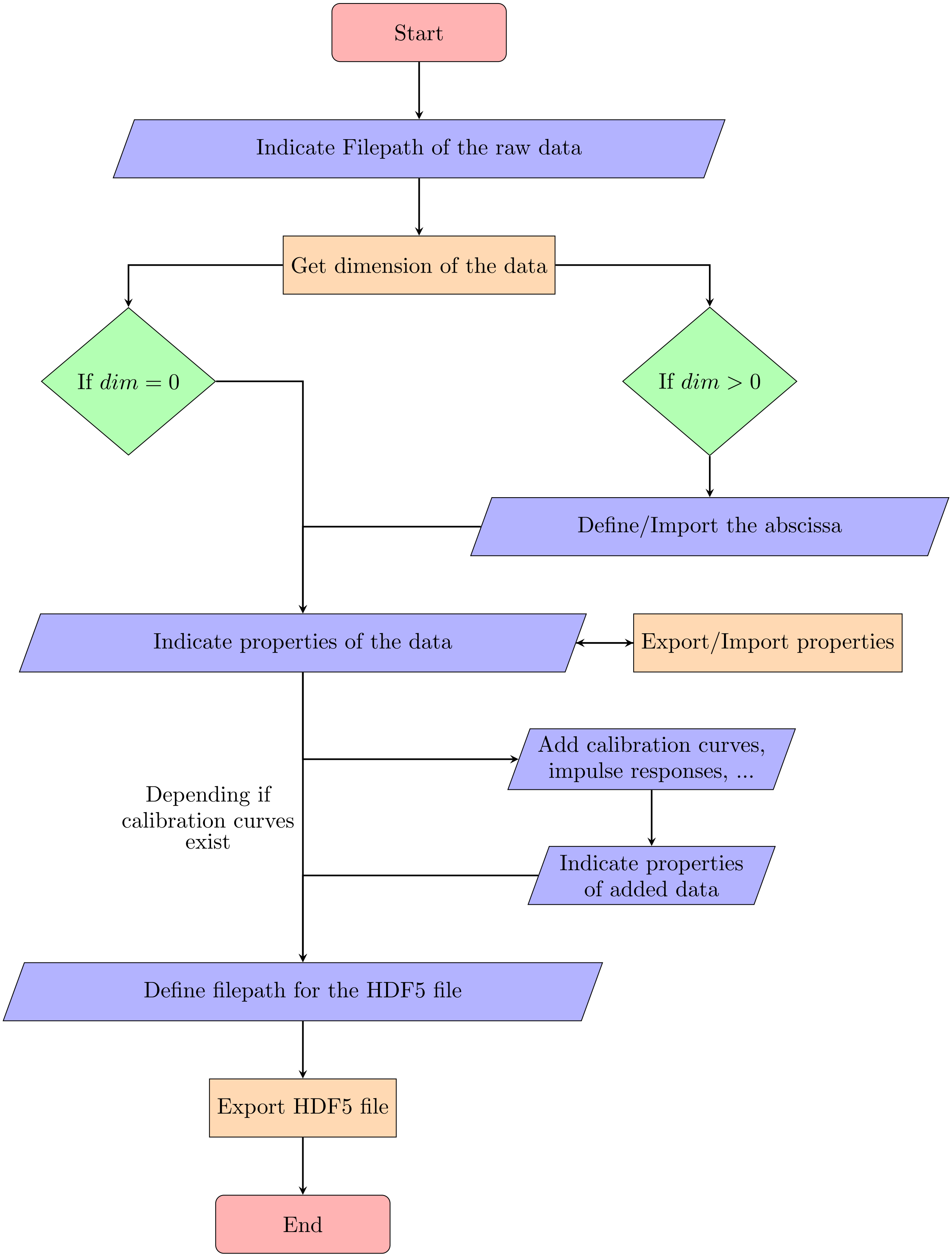
In order to simplify the reading of the raw data by softwares, we propose to set their dimensions as follows:

- Nth dimension: spectral channels
- (N-1)th dimension: (if reported) z dependence
- (N-2)th dimension: (if reported) y dependence
- (N-3)th dimension: (if reported) x dependence
- (N-4)th dimension: (if reported) radial angular dependence
- (N-4)th dimension: (if reported) azimutal angular dependence
- (N-5)th dimension: (if reported) temperature dependence
- (N-6)th dimension: (if reported) concentration dependence
- (N-7)th dimension and further: user-specific dimensions

5 HDF5_Brillouin_creator class structure

5.1 Schematical idea

To facilitate the creation of HDF5 files, we have developped a Python class to create HDF5 file containers. This program works as follows:


Start
Indicate Filepath of the raw data
Get dimension of the data
If $dim = 0$
If $dim > 0$
Define/Import the abscissa
Indicate properties of the data
Export/Import properties
Add calibration curves, impulse responses, ...
Depending if calibration curves exist
Indicate properties of added data
Define filepath for the HDF5 file
Export HDF5 file
End

5.2 Function definitions

Function Name	Description	Arguments	Returns
self	Initiates the class	None	Nothing
open_data	Opens a raw data file and returns the filepath	None	str (the filepath)
define_abscissa	Defines a new abscissa axis based on minimal values, maximal values and number of values	min, max, nb_samples	np.array (the abscissa array)
import_abscissa	Opens an abscissa file containing the abscissa points and returns the associated array	None	np.array (the abscissa array)
properties_data	creates a dictionnary with the given properties. All the properties are the attributes defined in section 3 of this document	**kwargs (the attributes with the key word as they will be written in the HDF5 file, for example "MEASURE.Sample")	dictionnary (the dictionnary with every attribute)
import_properties_data	creates a dictionnary with the given properties from a csv file.	filepath_csv	dictionnary (the dictionnary with every attribute)
export_properties_data	Asks for a csv name, and creates and saves a csv file with the properties listed in a dictionnary.	dictionnary	filepath (the filepath of the created csv file)
open_calibration	Asks for a calibration curve filepath and returns the calibration curve.	None	np.array
open_IR	Asks for an Impulse response curve filepath and returns the curve.	None	np.array
save_hdf5_as	Asks for the filepath where to save the hdf5 file and saves it.	None	None