SPEXone Level 1A to 1C Processor Release Notes

Raul Laasner, Jeroen Rietjens, Otto Hasekamp, Martijn Smit, Richard van Hees, Jochen Landgraf

Citation: Laasner, R., Aan de Brugh J.M.J., Rietjens J., Hasekamp O.P., Smit, M., Van Hees, R.M., Landgraf, J.: SPEXone Level 1A to 1C Processor Release Notes, SRON Netherlands Institute For Space Research, Utrecht, The Netherlands, SRON-SPEXoneL1-2022-02, issue: 1.0, 2022

document number	:	SRON-SPEXoneL1-2022-10-06
software version	:	2.1
date	:	2022-10-06
status	:	final

Contents

1 Introduction

The SPEXone data processor consists of three parts: the L1A-L1B processor, the L1B-L1C processor, and a calibration key data (CKD) generator. The CKD generator uses dedicated on-ground calibration measurements to acquire the CKD. The L1A-L1B processor uses flight L1A data together with the CKD to generate L1B data. The L1B-L1C data collocates L1B data on a common geolocation grid for a predefined reference height.
The L1A-L1B processor and the CKD generator are algorithmically interlinked to ensure full consistency of the derived CKD and the calibration of the L1A data. Therefore, it has been decided to include both elements in the same software package even though the CKD generator is not required for the processing of the SPEXone flight data.

2 New features

2.1 Binning

In this delivery, the L1A product has been generated using the real flight binning table. This has a noticeable effect on the L1A-L1B processor speed, in particular on noise propagation through the demodulation step. We expect this part of the processor to run much faster compared to the previous release.

2.2 Stray light correction algorithm

The new stray light correction algorithm makes use of multiple kernels. The number of kernels has significantly increased compared to previous deliveries but the application of each kernel is much faster. In this delivery, we test several configurations of applying stray light and how it affects the L1A-L1B processor's speed.

3 Installation and running

3.1 Dependencies

- GNU C++ compiler version 11
- CMake version 3.13.5
- C netcdf library version 4.7.4 (`netcdf_c++4` and `netcdf`)
- C hdf libraries version 1.10.6 (`hdf5_hl` and `hdf5`)
- Linear algebra library (e.g. lapack but preferably Intel MKL)
- FFTW version 3.3.5

3.2 Building

Make a copy of the initial CMake cache file

```
cp initial_cache.cmake.example initial_cache.cmake
```

found in the root source directory and edit it to reflect your environment. Next, create a build directory for configuring and building the SPEXone executable. From the root directory the procedure could look like this:

```
mkdir build && cd build
cmake -C ../initial_cache.cmake -D CMAKE_BUILD_TYPE=release ..
make -j
```

3.3 Running

Run the executable by issuing

```
mpirun -np <N> <spexone> <l1b.yaml>
```

where `<N>` is the number of MPI processes, `<spexone>` is the path to the SPEXone executable, and `<l1b.yaml>` is a configuration file. Three different runs should be perfomed in order to better understand stray light performance:

```
mpirun -np <N> <spexone> L1B_full.yaml
mpirun -np <N> <spexone> L1B_30_kernels.yaml
mpirun -np <N> <spexone> L1B_reduced.yaml
```

4 Files

- `spexone_cal.tar.gz` — source code. Unpack and compile according to instructions in Sec. 3.2.
- `L1A.nc` — simulated L1A product. This will serve as input to all simulations of this delivery.
- `L1B_full.yaml` — configuration file for running the L1A-L1B processor using all 50 stray light kernels.
- `CKD_full.nc` — calibration key data required by the processor when using `L1B_full.yaml`.
- `L1B_full.nc` — L1B product generated by running the processor with `L1B_full.yaml`. This serves as reference.
- `L1B_full.log` — screen output of processor with `L1B_full.yaml`
- `L1B_30_kernels.yaml` — like `L1B_full.yaml` but with 30 stray light kernels.
- `CKD_30_kernels.nc` — used with `L1B_30_kernels.yaml`.
- `L1B_30_kernels.nc` — output of running the processor with `L1B_30_kernels.yaml`.
- `L1B_30_kernels.log` — screen output of processor with `L1B_30_kernels.yaml`
- `L1B_reduced.yaml` — like `L1B_full.yaml` but kernels have a smaller spatial extent.
- `CKD_reduced.nc` — used with `L1B_reduced.yaml`.
- `L1B_reduced.nc` — output of running the processor with `L1B_reduced.yaml`.
- `L1B_reduced.log` — screen output of processor with `L1B_reduced.yaml`
- `gebco_ocssw_v2020.nc` — elevation map required for geolocation.
- `SPX1_CKD_BIN_TBL_20210304T124000_001.nc` — binning table
- `SPX1_TEST_UTC_delivery.nc` — Map of UTC time difference, constructed out of utcpole.dat from NASA.