Characterizing Error in the Verification Procedure of the ICESat-2 ATLAS Instrument's Level-1B Product

Catherine Gosmeyer[1], Megan Bock[2], Jeffrey Lee[3], and Anthony Martino[2]

[1]ADNET Systems Inc.
[2]NASA Goddard Space Flight Center
[3]SGT, Wallops Flight Facility

November 24, 2022

Abstract

The Advanced Topographic Laser Altimeter System (ATLAS) is the sole science instrument on NASA's ICESat-2 mission. It was designed to measure elevation simultaneously along six tracks on the earth's surface with centimeter-level vertical precision, demanding a picosecond-level precision in photon time of flight. To ensure this precision requirement was met, we developed for the Level-1B ATLAS data product a careful verification procedure. To quantify the amount of acceptable error we needed to understand the effects of the various floating-point precisions at all steps of the calculations and the limitations of the programming languages used for the production software (Fortran) and for the verification software (Python). For example, we found in the 64-bit photon time-of-flight calculations that differences even at 13 or 14 decimal places often revealed that an incorrect calibration value had been selected. Without first characterizing the acceptable error, such small errors could be overlooked and could propagate into critical inaccuracies in the science products. We describe our methods and lessons learned in order to inform future remote sensing verification efforts.

CHARACTERIZING ERROR IN THE VERIFICATION PROCEDURE OF ICESAT-2/ATLAS'S LEVEL-1B PRODUCT

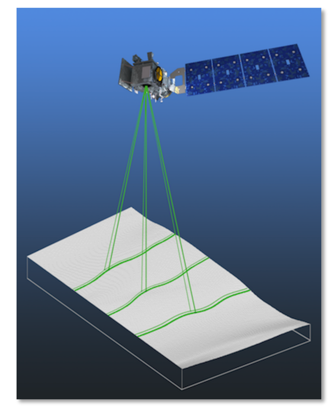
C.M. Gosmeyer, A.J. Martino, M.R. Bock, J.E. Lee
NASA Goddard Space Flight Center
catherine.gosmeyer@nasa.gov

Abstract NG21B-0943
AGU 10 Dec 2019

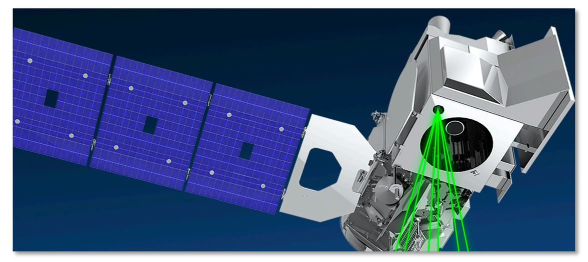
BACKGROUND

- Photon time of flight (TOF) is the primary calculation provided by the ICESat-2/ATLAS's level-1B product (L1B).
- To derive elevations with centimeter-level precision, TOF in L1B must have picosecond-level precision.
- We need characterize acceptable precision in all 25+ contributing calculations and calibrations to estimate how well we can expect the L1B TOFs to match independently calculated TOFs.
- Besides TOF, the L1B HDF5 file stores 1380 calculated and pass-through fields, including component temperatures, laser energies, flag settings, calibration selections, and so on, whose acceptable precision errors need to be characterized and verified as well.
- Goal of verification is to bring L1B code and algorithm description document into agreement by use of an independently written software suite.

ICESAT-2



- NASA's Ice, Cloud, and Elevation Satellite 2.
- Launched 15 Sept 2018.
- Three-year nominal mission.
- Near-polar orbit, with 91-day repeat ground track.
- Specialized data products for sea ice, land ice, inland water, vegetation, ocean, and clouds.



ATLAS

- Advanced Topographic Laser Altimeter System.
- Sole science instrument on ICESat-2.
- 532-nm laser, 10,000 pulses/second, split into six beams.
- Measures times of flight along six tracks on Earth's surface, which are converted to elevations with cm-level vertical precision.

LESSONS LEARNED

- *Issue Tracking*: Share with all parties a verification status table with one row for each field; over 400 of L1B's 1380 HDF5 fields had tracked issues (~20% calculation errors; ~80% metadata errors). Defining this table and committing to upkeep should be first step.
- *Consistency of Precision*: If the final result is declared 64 bits, ensure all intermediate calculations and inputs are also 64 bits. Mixing 32 with 64 accounted for >50% of L1B's calculation errors.
- *Intermediate Steps*: Build into both the verification code and the production code an easy means for reporting intermediate calculations to their full machine precision (no rounding on print).
- *Self-Documenting Code*: All coded calculations should be human-readable and traceable to the primary documents (e.g., variable names that match algorithms'; do *not* invent new nomenclature).
- *Open Software:* Store all software in a shared remote repository.

EVALUATING ACCEPTABLE PRECISION & ERROR

- Exact match to last decimal place not realistic! Need to quantify how close is *good enough* for a L1B value and a verification value to be considered a match. This analysis is critical for TOF because desired reproduceable precision (1e-12 sec) approaches expected machine 64-bit precision (1e-15); see **Figure 1**.

Error to minimize	Action	Justification
Programming language biases	L1B software is in Fortran; write verification software in Python	Precision differences well-documented; known that final floating points can vary somewhat
Measured/reported precision	Default to this precision before the machine precision	Neglecting to first quantify measured/reported precision has resulted in time wasted
Output precision	Pickle (serialized object) the executed verification software instead of writing to a file (e.g., HDF5 format used by L1B)	The verification objects are pickled and all attributes retain the exact form and precision which they had in the calculation; no chance of a formatting mistake; attributes can be directly compared with the appropriate field in the L1B HDF5 file
Algorithm to verification code implementation	As much as practical, variable and functions names match those given in algorithm document	Makes verification code easily traced to the algorithm document
Verification output to L1B output comparison	Write a dictionary for each L1B HDF5 field that maps it to a pickled verification attribute	Makes verification code's attributes easily traced to the L1B HDF5 fields; for added readability same dictionary contains the field's expected precision

- Want to minimize all errors so only remaining errors likely due to differences between the two codes' implementations of algorithms. Above table illustrates this process.
- By evaluating the precision each field *should be*, we were able to catch implementation errors that otherwise may have been overlooked and propagated into the science products.

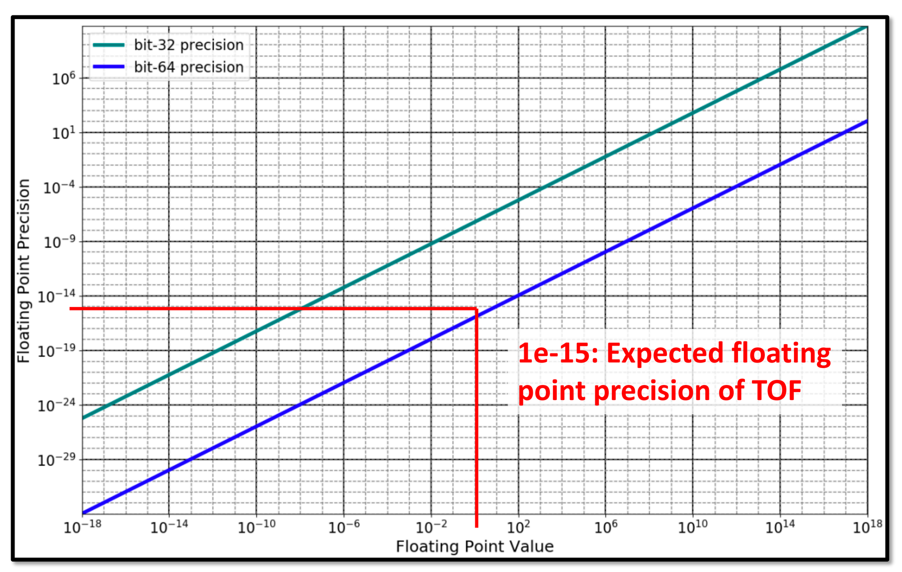


Figure 1 – Theoretical machine precision of both 32 and 64 bit representations for a given floating point value.

VERIFYING PHOTON TIME OF FLIGHT

3 ATLAS settings
5 timing values
6 applied calibrations
17 calculation steps

TOF equation from algorithm doc

$$TOF_{\varepsilon}^{\beta} = RX\_T_{\varepsilon}^{\beta} + TX\_Tfine^{\beta} - T_{center}$$

L1B implementation

TOFs
[0.0033000427777795955, …]

Verification implementation

TOFs
[0.0033000427777798093, …]

Diff two TOF implementations in verification function

```
tof = pickle_load(vfiles.tof)
d = {'ph_tof': [1e-15, (lambda pce : tof.map_pce(pce).TOF.all)],
```

HDF5 field name
Expected precision
Access pickled verification value and generate diffing function

Output diff plot

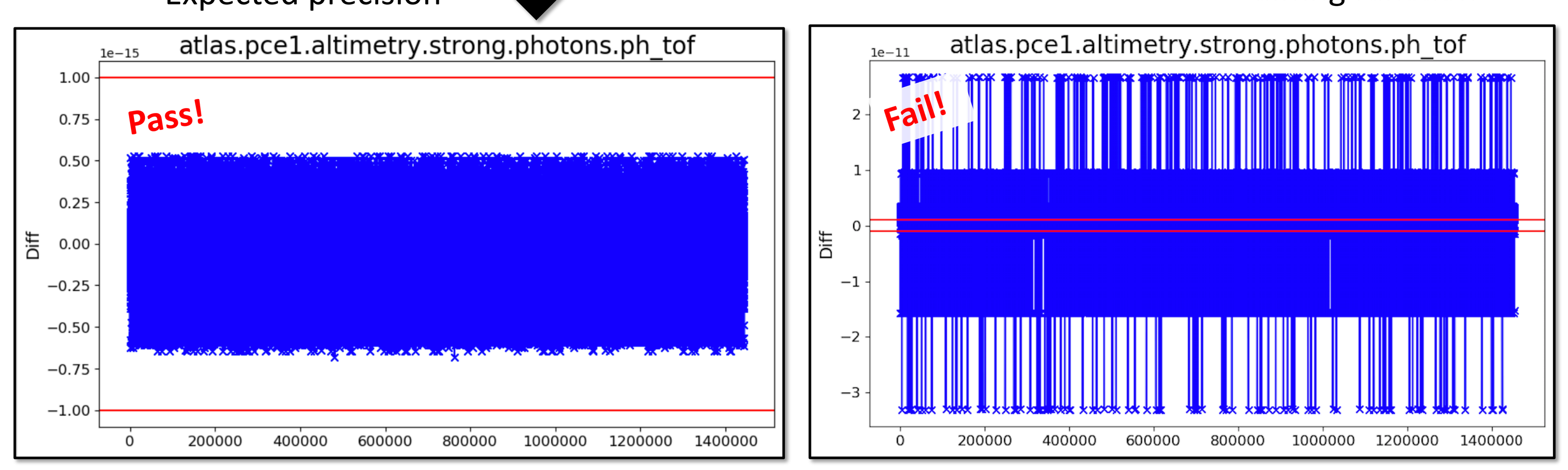


Figure 2 – TOF diffs that fall in expected 1e-15 bounds (left) and TOF diffs where an incorrect calibration value was selected in L1B (right).

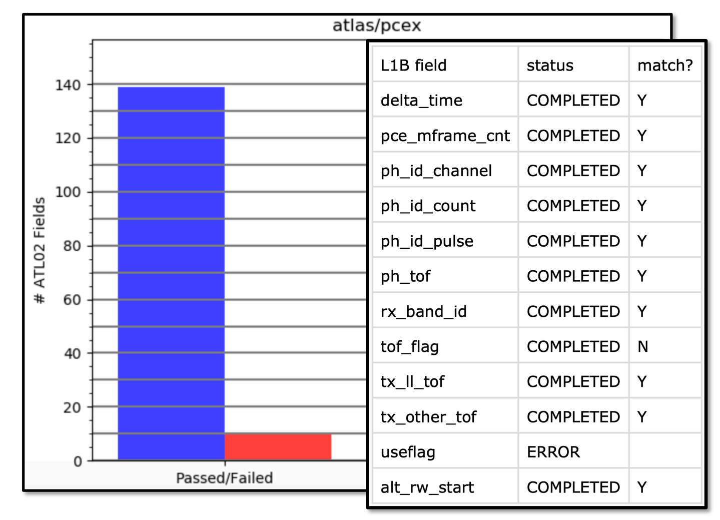


Figure 3 – Summary verification report of whole HDF5 group containing TOFs. Useful as quicklooks when running verification software on a new L1B version.

RESOURCES

1. **Level-1B Algorithm Theoretical Document:** icesat-2.gsfc.nasa.gov/science/data-products
2. **ICESat-2 public website:** icesat-2.gsfc.nasa.gov
3. **ICESat-2 data archive:** nsidc.org/data/icesat-2
4. **Technical video:** youtube.com/watch?v=aYRqkdYJRr0