NISAR User Guide

ASF Engineering

Keywords NISAR, User Guide

1. Introduction

User Guide

Published Nov 26, 2025

1.1. History

NISAR is a joint Earth-observing mission between National Aeronautics and Space Administration (NASA) and the Indian Space Research Organization (ISRO) with the goal to make global measurements of the causes and consequences of land surface changes using advanced radar imaging. This mission concept and the resulting partnership are in response to the National Academy of Science's 2007 survey of Earth observational priorities for the next decade, known as the decadal survey. One of the top priorities identified in this survey was to gain data and insight in three Earth science domains: ecosystems, deformation of Earth's crust and cryospheric sciences.

The Alaska Satellite Facility (ASF) Distributed Active Archive Center (DAAC) has been selected to archive the data from the NASA-ISRO Synthetic Aperture Radar (NISAR) mission. ASF will utilize NASA's Earthdata Cloud, which leverages Amazon Web Services (AWS) for processing, storage and distribution of the data.

The NISAR satellite was launched July 30, 2025.

1.2. NISAR scientific objectives

The NISAR Mission Science Users' Handbook (2019) defined the key scientific objectives as follows:

NISAR addresses the needs of Solid Earth, Ecosystems, and Cryospheric science disciplines, and provides data for many applications. NISAR is an all-weather, global geodetic and polarimetric radar-imaging mission with the following key scientific objectives:

1. *Determine the likelihood of earthquakes, volcanic eruptions, landslides and land subsidence;*
2. *Understand the dynamics of carbon storage and uptake in wooded, agricultural, wetland, and permafrost systems;*
3. *Understand the response of ice sheets to climate change, the interaction of sea ice and climate, and impacts on sea level rise worldwide.*

1.3. About the Jet Propulsion Laboratory

The Jet Propulsion Laboratory (JPL) is located on the outskirts of Pasadena, California. Founded in 1936 by California Institute of Technology (Caltech) researchers, JPL is a research and development lab federally funded by NASA and managed by Caltech.

1.4. About the Alaska Satellite Facility (ASF)

The Alaska Satellite Facility (ASF) operates the NASA archive of synthetic aperture radar (SAR) data from various satellites and aircraft, providing these data and associated services to the public in support of NASA's Earth Science Data and Information System (ESDIS) project.

2. About NISAR

2.1. About the Instrument

NISAR was designed to deliver fast sampling, free and open access, and global coverage at full resolution and with polarimetric diversity. The technical innovation that allows this performance is the scan-on-receive *SweepSAR* design, conceived and refined collaboratively by NASA and the German Space Agency (DLR).

The L-band Synthetic Aperture Radar (L-SAR) instrument is the focus of the NASA-chartered science goals for NISAR, and provides global coverage. The L-SAR is a side-looking, fully polarimetric, interferometric SAR at a wavelength of 24 cm, with a swath width of about 240 km, an along-track resolution of 7 m, and a cross-track resolution of 2 to 8 m (depending on mode).

The S-band SAR (S-SAR) instrument, developed in a collaboration between NASA and the Indian Space Research Organization (ISRO), operates at a wavelength of 9.3 cm and is primarily used for acquisitions over India.

A more detailed description of the NISAR instrument design is in section 4.7 of the NISAR Mission Science Users' Handbook.

2.2. Frequencies and polarizations

NISAR is equipped to acquire data in two slightly different frequencies. The table below summarizes the frequency A and the frequency B settings that define the acquisition modes.

L-band bandwdith [MHz]	Center frequency [MHz]	
	Frequency A	**Frequency B**
77	1257.5	NA
40+5	1239	1293.5
20+5	1229	1293.5
20+20	1229	1286
5+5	1221.5	1236.5

Table 1: NISAR acquisition modes

The polarization refers to the direction that an electromagnetic wave travels. This can be horizontal (left to right), vertical, (up and down), or circular (rotating in a constant plane left or right). The antennas of the SAR system can transmit and receive electromagnetic waves with the various polarizations. These polarimetric properties of the observed surface can reveal the structure, orientation and environmenal conditions of the surface elements.

For the NISAR mission, the following polarizations have been selected. The single polarizations (SH for HH polarized data and SV for VV polarized data), the dual polarizations (DH for HH+HV dual-polarized data and DV for VV+VH dual-polarized data) as well as quad-polal data (HH+HV+VH+VV) have been used in the past. The compact polarization (CP) introduces circular polarization (RH for right-circular HH data and RV for right-circular VV) has not been used routinely with satellite data.

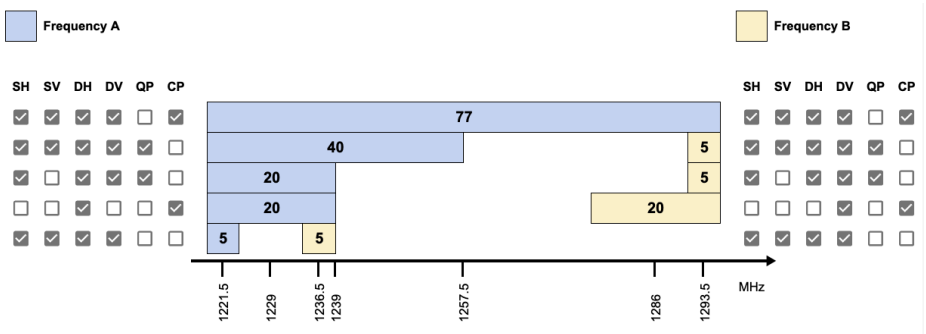

Figure 1: NISAR acquisition modes

For interferometric NISAR product, frequency A only data are used. All other NISAR products can be acquired in a combination of frequency A and frequency B as shown above.

2.3. Algorithm Theoretical Basis Documents

Two algorithm theoretical basis documents (ATBDs) summarize how the data is processed at various processing levels:

- NISAR L1 and L2 Algorithm Theoretical Basis Document (NISAR Algorithm Development Team, 2023)
- NISAR L3 Soil Moisture Data Product: Algorithm Theoretical Basis Document (ATBD) (NISAR Soil Moisture Science Team, 2023)

There are other ATBDs summarizing the findings of the NISAR science team regarding how the requirements for generating particular data products are met:

- Algorithms to Validate NISAR L2 Coseismic, Transient and Secular Displacement Requirements (NISAR Solid Earth Team, 2022)
- NISAR L3 Ice Sheets and Glacier Products (Joughin et al., 2022)
- NISAR L2 Sea Ice Motion Product (Holt, 2022)
- NISAR Ecosystem L2 Products (Siqueira et al., 2018)

2.4. Data Use and Citation Guidance

The NISAR follows NASA's standard Data Use and Citation Guidance.

2.5. Product Specifications

There are links to current versions of the product specification documents on the NISAR Sample Data Product Suite website.

- Level-0B Radar Raw Signal Data and Calibration Product (Hawkins, 2025a)
- Level-1 Range Doppler Single Look Complex (Hawkins, 2025b)
- Level-2 Geocoded Single Look Complex (Fattahi, 2025)
- Level-1 Range Doppler Wrapped Interferogram (Brancato et al., 2025a)
- Level-1 Range Doppler Unwrapped Interferogram (Brancato et al., 2025b)
- Level-2 Geocoded Unwrapped Interferogram (Brancato et al., 2025c)
- Level-2 Geocoded Polarimetric Covariance (Shiroma, 2025)
- Level-1 Range Doppler Pixel Offsets (Brancato et al., 2025d)
- Level-2 Geocoded Pixel Offsets (Brancato et al., 2025e)
- Level-3 Soil Moisture (Huang & Chan, 2025)

2.6. Naming Conventions

The naming schemes for NISAR products are summarized in Appendix B.

3. Processing Levels

- Level 0
- Level 1
- Level 2
- Level 3

3.1. Level 0

NISAR L0B Radar Raw Signal Data (RRSD) is corrected, aligned, and time-ordered radar pulse data derived from RAW products and used to generate higher-level products.

3.2. Level 1

NISAR L1 data products are in the original SAR geometry in range-Doppler coordinates.

The L1 **Range-Doppler Single Look Complex (RSLC)** product contains a focused SAR image providing amplitude and phase information.

Two L1 RSLC products, with the same footprint but acquired on different dates, are used to generate an L1 **Range-Doppler Nearest-Time Interferogram (RIFG)**, a multi-looked interferogram ellipsoid and topographic phase flattened and formed with precise coregistration using geometrical offsets and high-resolution pixel offsets. Only nearest-neighbor pairs are processed; RIFG products with longer temporal baselines will not be automatically generated by the NISAR mission.

The L1 **Range-Doppler Nearest-Time Pixel Offsets (ROFF)** product provides unfiltered and unculled layers of pixel offsets with different resolutions obtained from incoherent cross-correlation.

The L1 **Range-Doppler Nearest-Time Unwrapped Interferogram (RUNW)** is the multi-looked unwrapped interferogram that is ellipsoid- and topography-flattened.

3.3. Level 2

NISAR L2 data products are geocoded L1 products projected to a map coordinate system.

The L2 **Gecoded Single Look Complex (GSLC)** product is map projected using the Medium precision Orbit Ephemeris (MOE) state vectors and a Digital Elevation Model (DEM).

The L2 **Geocoded Nearest-Time Pixel Offsets (GOFF)** product provides unfiltered and unculled layers of pixel offsets in different resolutions obtained from incoherent cross-correlation.

The L2 **Geocoded Nearest-Time Unwrapped Interferogram (GUNW)** is a multi-looked unwrapped interferogram that is ellipsoid- and topography-flattened. Only nearest-neighbor pairs are processed; GUNW products with longer temporal baselines are not automatically generated by the NISAR mission.

The L2 **Geocoded Polarimetric Covariance Matrix (GCOV)** is a multi-looked polarimetric covariance matrix using MOE state vectors and a DEM.

3.4. Level 3

The L3 **Geocoded Soil Moisture (SME2)** product contains the results of several different models that take NISAR backscatter observations as input. They are projected on the 200-meter global EASE Grid 2.0.

4. Data Products

- L0B RRSD
- L1 RSLC
- L2 GSLC
- L1 RIFG
- L1 RUNW
- L2 GUNW
- L2 GCOV
- L1 ROFF
- L2 GOFF
- L3 SME2
- Ancillary files
- Data Format
- Metadata
- Software that supports reading/viewing/using NISAR data

4.1. L0B RRSD

L0B RRSD is the Level 0 product available to the science disciplines from the DAAC. The RRSD product comprises time-sorted unpacked recordings of raw radar echo pulses and related radar instrument telemetry. It is comparable to the Level-0 raw data products delivered by SAR sensors worldwide (e.g., L1.0 by ESA and JAXA). The RRSD data is focused to a Radar Single Look Complex (RSLC) product before use in higher-level processing.

A single RRSD product granule consists of radar echoes acquired in the same imaging mode. An observation constitutes all consecutive radar echoes acquired in the same imaging mode. The radar echoes are organized in sub-groups by imaging frequency (A or B), transmit polarization (H, V, L, or R), and receive polarization (H or V) in that order.

The radar echoes of any given frequency and polarization combination are decoded and aligned to adjust for different sampling window start times and can be accessed as a simple two-dimensional raster image. The RRSD product also contains all the instrument telemetry for the entire data take, including the observation.

4.2. L1 RSLC

The RSLC product is in the zero-Doppler radar geometry convention. The output image is on a grid characterized by constant azimuth time interval and one-way slant range spacing, and all the primary image layers for a multi-polarization or multi-frequency product are generated on a common time-slant range grid. The output grid is also characterized by a fixed starting slant range, azimuth time interval, and slant range spacing values for easy interpolation.

The complex backscatter values of the RSLC are digital numbers (DNs) with secondary layer look-up tables (LUTs) provided to convert to beta-naught, sigma-naught, and gamma-naught radiometry. These radiometric correction LUTs are defined for the ellipsoid, not for the local terrain.

All standard (i.e., not urgent-response) products are processed using the Medium-fidelity Orbit Ephemeris (MOE) product for forward processing and the Precise Orbit Ephemeris (POE) product for reprocessing campaigns.

4.3. L2 GSLC

The GSLC product is a Level 2 product derived from the Level-1 RSLC product by geocoding the input RSLC into a map coordinate system such as a UTM Zone or Polar stereographic projection system.

The geocoding is performed by inverse mapping of the map coordinates with their topographic heights into the radar coordinate system and interpolating the radar signal at the radar location corresponding to the map coordinate. Phase-preserving complex interpolation projects the data onto a uniformly spaced, north-south/east-west aligned geographic grid.

The phase of the GSLC product is flattened for the orbit used in the RSLC processing. The phase flattening removes the topographic phase contribution in the GSLC. Consequently, cross-multiplying two GSLC products will result in an interferometric phase flattened interferogram.

4.4. L1 RIFG

The RIFG product represents the ellipsoid and topography flattened wrapped interferogram generated from two L1 range-Doppler Single Look Complex (RSLC) products in the range-Doppler geometry of the earlier reference acquisition. The WGS84 ellipsoid is used as the reference surface for flat earth correction.

The RIFG product contains a binary raster layer of complex numbers representing the wrapped interferometric phase difference in radians, i.e., the wrapped interferogram. The product also contains a binary raster layer of floating-point numbers representing the normalized interferometric correlation, i.e., the interferometric coherence magnitude.

The wrapped interferogram and the interferometric coherence magnitude are multi-looked to a nominal posting of 30 m on the ground. No ionospheric phase screen correction layers are available with this product.

The interferometric workflow producing RIFG products coregisters a pair of RSLC products using a DEM and the best available orbit ephemeris. This coregistration is refined using incoherent cross-correlation on the pair of coarsely coregistered RSLCs.

The RIFG product includes the slant range and along-track sub-pixel offsets obtained from incoherent cross-correlation and used to generate the complex wrapped interferogram. If an offset product in range-Doppler coordinates (e.g., ROFF) is available for the processed frame, the sub-pixel offset layers included in RIFG are obtained by optimally blending the multiresolution offset layers included in ROFF.

4.5. L1 RUNW

The RUNW product represents the unwrapped, ellipsoid- and topography-flattened, multi-looked interferogram generated from two L1 Range Doppler Single Look Complex (RSLC) products in the earlier reference acquisition range-Doppler geometry. The RUNW product has a nominal posting of 80 meters on the ground, irrespective of the slant range bandwidth.

The product contains raster layers representing single precision floating point unwrapped and normalized interferometric coherence magnitude. The product includes the connected component mask, a mask layer identifying invalid pixels (i.e., pixels affected by geometric distortions), and the slant range and along-track offsets obtained from incoherent cross-correlation. Look-up tables for parallel and perpendicular baseline components are also included.

The RUNW product also contains an ionospheric phase screen layer and a layer quantifying its uncertainty. Where possible, the ionospheric phase screen is estimated from the two spectral bands frequencyA and frequencyB. In the case of mode transitions where the continuity of spectral bands is impacted, a split spectrum ionospheric phase estimate and an estimate of its standard deviation are derived from the main imaging band frequencyA. The estimated ionospheric phase screen is included in the product but not applied to the data layers within RUNW by default.

The RUNW product includes the slant range and along-track sub-pixel offsets obtained from incoherent cross-correlation and used to generate the complex wrapped interferogram. If an offset product in range-Doppler coordinates (e.g., ROFF) is available for the processed frame, the sub-pixel offset layers included in RUNW are obtained by optimally blending the multiresolution offset layers included in ROFF.

4.6. L2 GUNW

The GUNW product is an L2 product derived from the RIFG and RUNW products by geocoding the unwrapped phase and associated data layers (i.e., coherence magnitude, ionospheric phase screen) on a geographical grid at 80 m posting.

Geocoding is performed using the orbit of the reference RSLC product and a DEM to project the data onto a predefined Universal Transverse Mercator (UTM) or Polar stereographic projection system map grid. The geocoding algorithm uses a bilinear interpolation for interpolating data layers with floating-point data types, Sinc interpolation for the complex wrapped interferogram, and nearest-neighbor interpolation for unsigned integer datasets (e.g., connected components mask).

The GUNW products also include the wrapped complex interferogram (multi-looked at 30 m in range-Doppler coordinates and geocoded at 20-m posting), the unwrapped interferometric phase in radians (80-m posting), the normalized interferometric coherence magnitude (20-m and 80-m posting), connected components mask, and sub-pixel offset layers obtained from incoherent cross-correlation. If an offset product in range-Doppler coordinates (e.g., ROFF) is available for the processed frame, the sub-pixel offset layers included in GUNW are obtained by optimally blending the multiresolution offset layers included in ROFF.

The GUNW product also includes an ionospheric phase screen layer from the RUNW product and a layer quantifying its uncertainty. In the case of mode transitions where the continuity of spectral bands is impacted, a split spectrum ionospheric phase estimate is derived from the main imaging band frequencyA. The estimated ionospheric phase screen is included in the product but not applied to the data layers within the GUNW product by default.

The GUNW product also includes geocoded lookup tables for external phase corrections (e.g., solid Earth tides, hydrostatic and wet tropospheric phases). These phase corrections, when available, are not removed from the interferometric data.

4.7. L2 GCOV

The GCOV product is an L2 product derived from the L1 Range Doppler Single Look Complex (RSLC) product, providing terrain-corrected polarimetric covariance projected onto a predefined UTM or polar stereographic projection system grid. RSLC radar samples, organized as a polarimetric scattering vector, are cross-correlated with the scattering vector's conjugate transpose, originating the polarimetric covariance matrix expressed in the same grid as the RSLC range-Doppler grid.

The magnitude of the resulting polarimetric covariance terms is strongly affected by the topography, with areas facing the sensor becoming brighter and areas away from the sensor turning darker in the images, biasing covariance measurements. To reduce the effect of the topography, an area-based radiometric terrain correction (RTC) is applied over the covariance terms, normalizing the backscatter coefficient beta0 to gamma0. The normalized covariance terms are then geocoded onto the output grid using area-based adaptive multi-looking.

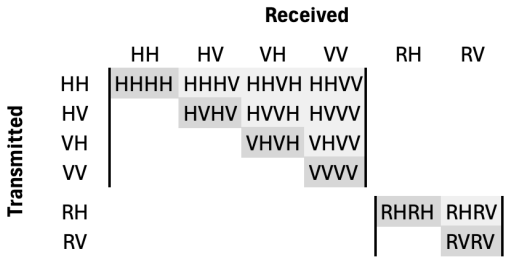

Figure 2: Covariance polarimetric matrix elements

Since the polarimetric covariance matrix is Hermitian, only the upper triangular covariance terms are provided. The diagonal terms of the polarimetric covariance matrix (highlighted in darker gray) are real-valued (HHHH, HVHV, VHVH, and VVVV, or RHRH and RVRV), representing the radar backscatter associated with each polarimetric channel. The off-diagonal terms of the polarimetric covariance matrix (highlighted in lighter gray) are complex-valued (HHHV, HHVH, HHVV, HVVH, and VHVV, or RHRV) and may or may not be present depending on the GCOV processing mode.

4.8. L1 ROFF

The ROFF product contains a collection of dense pixel offset layers obtained from applying incoherent cross-correlation on a pair of coarsely coregistered L1 Range Doppler Single Look Complex (RSLC) products in the range-Doppler geometry of the earlier reference RSLC product.

The pair of RSLCs used to produce ROFF is first coarsely aligned with geometrical coregistration using the best available sensor orbit ephemeris and a DEM. The spacing, the window size, and the search radius used to generate the ROFF offsets layers for L-SAR data are organized by sensor mode and area of observation.

Irrespective of the mode, the pixel offset layers are provided with a nominal posting of 90 m on the ground. It is assumed that pixel offset layers within ROFF share the same spacing and the same starting pixel along slant range and azimuth directions.

Each pixel offset layer is distributed without performing any conventional post-processing operation, i.e., layers might contain offset outliers and are not low-pass filtered to reduce noise in the data.

4.9. L2 GOFF

The GOFF product is an L2 product derived from the ROFF product by geocoding the pixel offset layers and associated data layers (i.e., SNR) on a geographical grid at 80 m posting. Geocoding uses the orbit of the reference RSLC product and a DEM to project the data onto a pre-defined UTM or Polar stereographic projection system map grid. The geocoding algorithm uses a bilinear interpolation for interpolating data layers with floating-point data types.

The GOFF product contains a collection of data layers representing the pixel offset shifts between a pair of coarsely coregistered RSLC granules. The spacing, the window size, and the search radius used to generate the pixel offset layers in the range-Doppler geometry are organized by range bandwidth and area of observation.

Pixel offset layers are distributed without performing any conventional post-processing operation, i.e., layers might contain offset outliers and are not low pass filtered to reduce noise in the data.

4.10. L3 SME2

NISAR backscatter observations will estimate a global high-resolution soil moisture product at a pixel spacing of 200 m. On average, this product will be provided twice every 12 days, and is expected to have a data latency of 72 hours.

Soil moisture estimates will be provided over areas with dense vegetation (VWC greater than 5 kg/m2), but will be flagged as potentially unreliable. No soil moisture retrievals will be provided for areas with urban build-up, permanent snow and ice cover, or permanent inland waterbodies, and these pixels will be flagged appropriately. No soil moisture retrieval will be performed over areas with excessive precipitation, frozen ground, or snow on the ground.

The NISAR soil moisture has an accuracy goal of 0.06 m3/m3 over unflagged areas with vegetation water content below 5 kg/m2. The 5+5 MHz product will be aggregated to 400 meters but will still be posted to the 200-m EASE-Grid 2.0.

4.11. Ancillary files

There are a number of ancillary files that are available since they are used as part of the processing.

Short Name	Description
COP	Coordinated Observation Plan that contains a 7 day plan of observations
LRCLK-UTC	L-SAR Radar clock to UTC Spacecraft Clock Spacecraft Event Time (SCLKSCET) conversion file
FOE	Forecast Orbit Ephemeris
MOE	Medium Orbit Ephemeris
NOE	Near-Realtime Orbit Ephemeris
POE	Precise Orbit Ephemeris
STUF	STUF - 1 week plan of orbit boundaries, convert UTC to Orbit
FRP	Forecast Radar Pointing
NRP	Near-Realtime Radar Pointing
PRP	Precise Radar Pointing
DEM	Digital Elevation Model
ECMWF	European Centre for Medium-Range Weather Forecasts (models)
LOCAL_INC_ANG	Local Incidence Angles
VWC	Vegetation water content
WaterMask	Water Mask

4.12. Data Format

All NISAR standard products are in the Hierarchical Data Format version 5 (HDF5). HDF5 is a general-purpose file format and programming library for storing scientific data. The flexible format allows establishing structure to data and metadata.

While the files are stored in HDF5 format, the internal structure of the NISAR products is compliant with the Climate and Forecast (CF) metadata convention of the Network Common Data Form (netCDF) format. This can cause problems when using some HDF5 drivers to read the data. While ArcGIS Pro introduced support for reading NISAR HDF5 files with version 3.4.0, there is not yet a NISAR-compliant HDF5 reader for GDAL.

Those who use GDAL to read files for use in QGIS or programmatic workflows can replace the '.h5' NISAR product extension to '.nc'. This allows GDAL to recognize the spatial coordinate system and project the file appropriately. Refer to ASF's documentation on workarounds for accessing NISAR HDF5 files.

4.13. Metadata

The complete structure of the metadata is provided in the product specification. Key metadata fields and data dimensions that are embedded in the structure of the respective HDF5 product file are summarized per product in Appendix C.

The metadata cube is a three-dimensional grid designed to simplify the calculation of parameters within a geographic information system (GIS) environment. These grids are defined for all NISAR level 1 (L1) and level 2 (L2) products. Users are advised to use three-dimensional interpolation routines (for Python, use the RegularGridInterpolator function of the open-source software *scipy*) to calculate values for several parameters for useful initial data analysis within a GIS environment. The geolocation error using the grids is estimated to be 1.5 cm. Further analysis is expected to lead to a more robust error estimate.

The example below shows the geolocation grid defined for an L1 RIFG product. Using slant range, zero Doppler time, and height above ellipsoid as the grid axes serve as input to calculate the values of all other parameters.


```
{
  "granule":
"NISAR_L1_PR_RIFG_001_030_A_019_002_2000_SH_20081012T060911_20081012T060925_20081127T06
  "dimension": {
    "geolocationCubeHeight": 20,
    "geolocationCubeLength": 226,
    "geolocationCubeWidth": 68,
    "twoLayersCubeHeight": 2
  },
  "science": {
    "LSAR": {
      "RIFG": {
        "metadata": {
          "geolocationGrid": {
            "coordinateY": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(20,226,68)"
            },
            "coordinateX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(20,226,68)"
            },
            "incidenceAngle": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
```

```
              "data_type": "float64",
              "shape": "(20,226,68)"
            },
            "losUnitVectorX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(20,226,68)"
            },
            "losUnitVectorY": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(20,226,68)"
            },
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(20,226,68)"
            },
            "alongTrackUnitVectorY": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(20,226,68)"
            },
            "elevationAngle": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(20,226,68)"
            },
            "parallelBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(2,226,68)"
            },
            "perpendicularBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(2,226,68)"
            },
```

```
            "slantRange": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(68)"
            },
            "zeroDopplerTime": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeLength)",
              "data_type": "float64",
              "shape": "(226)"
            },
            "groundTrackVelocity": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(20,226,68)"
            },
            "heightAboveEllipsoid": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeHeight)",
              "data_type": "float64",
              "shape": "(20)"
            }
          }
        }
      }
    }
  }
}
```

The L2 GCOV example below takes the x coordinates, y coordinates, and heights above the ellipsoid, spaced 500 m between grid points, as grid axes to calculate the values of all other parameters.

```
{
  "granule":
"NISAR_L2_PR_GCOV_008_029_A_010_0005_NASV_A_20251031T044409_20251031T044425_P05000_N_P_
  "dimension": {
    "radarCubeHeight": 21,
    "radarCubeLength": 686,
    "radarCubeWidth": 699,
  },
  "science": {
    "LSAR": {
      "GCOV": {
        "metadata": {
          "radarGrid": {
            "slantRange": {
              "type": "cube",
              "dimension":
```

```
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(21,686,699)"
            },
            "zeroDopplerAzimuthTime": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(21,686,699)"
            },
            "incidenceAngle": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(21,686,699)"
            },
            "losUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(21,686,699)"
            },
            "losUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(21,686,699)"
            },
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(21,686,699)"
            },
            "alongTrackUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(21,686,699)"
            },
            "elevationAngle": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(21,686,699)"
```

```
            },
            "groundTrackVelocity": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(21,686,699)"
            },
            "xCoordinates": {
              "type": "cube",
              "dimension": "(radarCubeWidth)",
              "data_type": "float64",
              "shape": "(699)"
            },
            "yCoordinates": {
              "type": "cube",
              "dimension": "(radarCubeLength)",
              "data_type": "float64",
              "shape": "(686)"
            },
            "heightAboveEllipsoid": {
              "type": "cube",
              "dimension": "(radarCubeHeight)",
              "data_type": "float64",
              "shape": "(21)"
            }
          }
        }
      }
    }
  }
}
```


The table below summarizes which parameters are available for the users' analysis in the various NISAR L1 and L2 products. The parameters highlighted in gray indicate the input parameters for calculating the values of the remaining parameters.

	LOB	L1_RIFG	L1_ROFF	L1_RSLC	L1_RUNW	L2_GCOV	L2_GOFF	L2_GSLC	L2_GUNW	L3_SME2
coordinateY	☐	☑	☑	☑	☑	☐	☐	☐	☐	☐
coordinateX	☐	☑	☑	☑	☑	☐	☐	☐	☐	☐
incidenceAngle	☐	☑	☑	☑	☑	☑	☑	☑	☑	☐
losUnitVectorX	☐	☑	☑	☑	☑	☑	☑	☑	☑	☐
losUnitVectorY	☐	☑	☑	☑	☑	☑	☑	☑	☑	☐
alongTrackUnitVectorX	☐	☑	☑	☑	☑	☑	☑	☑	☑	☐
alongTrackUnitVectorY	☐	☑	☑	☑	☑	☑	☑	☑	☑	☐
elevationAngle	☐	☑	☑	☑	☑	☑	☑	☑	☑	☐
parallelBaseline	☐	☑	☑	☐	☑	☐	☑	☐	☑	☐
perpendicularBaseline	☐	☑	☑	☐	☑	☐	☑	☐	☑	☐
slantRange	☐	☐	☐	☐	☐	☑	☐	☑	☐	☐
referenceSlantRange	☐	☐	☐	☐	☐	☐	☑	☐	☑	☐
secondarySlantRange	☐	☐	☐	☐	☐	☐	☑	☐	☑	☐
zeroDopplerAzimuthTime	☐	☐	☐	☐	☐	☑	☐	☑	☐	☐
referenceZeroDopplerAzimuthTime	☐	☐	☐	☐	☐	☐	☑	☐	☑	☐
secondaryZeroDopplerAzimuthTime	☐	☐	☐	☐	☐	☐	☑	☐	☑	☐
groundTrackVelocity	☐	☑	☑	☑	☑	☑	☐	☑	☑	☐
hydrostaticTroposphericPhaseScreen	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐
wetTroposphericPhaseScreen	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐
slantRangeSolidEarthTidesPhase	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐
heightAboveEllipsoid	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Figure 3: NISAR cube metadata

4.14. Software that supports reading/viewing/using NISAR data

There are several software packages and tools that can be used to interact with all NISAR products except L0B RRSD data. L0 level data is unprocessed and has very limited viewability.

Instructions for working with NISAR L2 and L3 data formats in ArcGIS, QGIS, and Panoply are provided in ASF's Work with NISAR Sample Data documentation.

The documentation also provides guidance for using GDAL command line tools for these tasks:

- Getting Sub-dataset / Group Paths
- Convert from NISAR H5 to GeoTIFF
- Reproject a Single Subdataset from NISAR H5 Data
- Reproject an Entire NISAR H5 Dataset
- Clip a NISAR H5 Dataset to a Geographic Extent
- Create a Multiband GeoTIFF with a NISAR Dataset

5. Applications

- Amplitude
- Phase
- Ecosystem applications
- Geologic and land hazard monitoring
- Critical infrastructure monitoring
- Maritime and coastal ocean applications
- Hydrology and underground resources
- Cryosphere

5.1. Amplitude

The amplitude is the amount of the signal sent out by the sensor that returns to be measured, and is often referred to as radar backscatter. This part of the NISAR data is best represented in the GCOV products. The individual covariance layers in the GCOV product can be used like Radiometric Terrain Corrected (RTC) products. The pixel values represent the intensity of radar backscatter, with a different layer for each polarization.

The values in the CGOV covariance layers represent the conditions of the scatterers on the earth's surface in that area, and can indicate the degree of surface roughness, structural complexity, and relative moisture content. These values are impacted by changes to the physical structure of the earth's surface, including processes that alter vegetation or built structures.

Because of NISAR's regular acquisition schedule and insensitivity to cloud cover, data is available at regular intervals all over the world. This makes it very valuable for time-series analysis, which can be used to track short- or long-term changes to natural and anthropogenic features.

The L-band sensor can penetrate through moderately complex vegetation and more deeply into the soil than C-band SAR sensors such as Sentinel-1. This provides insight into the conditions under the canopy, and at deeper levels in the soil column, and having observations using both wavelengths (and, in some areas, S-band data in addition) increases the ability to understand processes driving change on the landscape.

5.2. Phase

The phase measures the location of the SAR signal along its wave cycle when it returns to the sensor. This can be used as a proxy for distance, and is the basis of SAR Interferometry (InSAR). By differencing the phase values from one acquisition to another with the same footprint, InSAR allows users to identify and quantify surface deformation that occurred in the time between acquisitions.

InSAR analysis can quantify differences in the line-of-sight of the sensor to the centimeter scale, making it a powerful tool for monitoring deformation over large areas. It can be used in conjunction with GNSS measurements, surveys, and LiDAR measurements to expand these high-accuracy measurements to a much broader scale.

Phase measurements, especially at L-band wavelengths, are sensitive to atmospheric changes from one acquisition to the next, so it is vital to consider how these changes can impact your interpretation of InSAR measurements. NISAR provides a number of resources packaged in with products such as the GUNWs that can be used for atmospheric corrections, and there are also time-series approaches that can help mitigate the impact of atmospheric conditions.

InSAR analysis can be used to detect and quantify surface movements or changes caused by geophysical processes or human activities that result in the signal having to travel a different distance to interact with the scattering surface. This can be due to volcanic or tectonic activity, earth-moving for construction or extraction purposes, and subsidence or uplift due to changes in groundwater or underground reservoirs.

InSAR also generates a coherence measurement, which may be useful for indicating change. If the phase spectra from one acquisition to another differ so much that they can't be compared, that can indicate an area that has undergone such drastic change that it can't be resolved. This indicates the quality of the phase difference calculation, but can also serve as an indicator of change.

5.3. Ecosystem applications

5.3.1. *Agriculture:*

NISAR acquisitions support mapping agricultural regions globally every two weeks. The structures of different crop and land cover types provide a rich variety of responses to the radar illumination in terms of varying polarization and frequency signatures.

Because of the rapid, time-varying nature of crop rotation, growth, and harvest, frequently repeated radar observations can be used to determine both the type of crop and its stage of growth. Information like this is used to predict the health of crops and planned agricultural output as a measure for food security.

5.3.2. *Forestry:*

Forests protect water resources, provide air purification, and reduce erosion. They are key to
sustaining crucial biodiversity and enhancing conditions for agricultural production. Forests also serve a social function as a recreational resource, an important labor market, and living space, and they enhance national defense. Both the amplitude and phase components of NISAR products can be useful for forestry applications.

When forest canopies are disturbed by the full or partial removal of standing trees, or significant fractions of the upper canopy are lost due to forest fire or pest infestations, the changes are reflected in a rich variety of measurable radar signals. NISAR provides a consistent time-series that can be used to track when, where, and by how much forests have been altered.

One of the greatest challenges facing forest managers around the world is maintaining the health and resilience of the forest ecosystems. This requires a coordinated and systematic effort for monitoring forest cover, volume, and productivity, and the development of techniques and policies for improving the stock and sustainable use of woody biomass. NISAR helps by generating large-scale maps needed to more accurately predict forest biomass, fuel loads, fire risk, and fire behavior.

5.3.3. *Fires:*

Fires are crucial to many ecosystems, yet often threaten developed urban areas, protected lands and regional air quality. Categorizing fire danger, detecting fires, identifying burned areas and quantifying the severity of fires is critical for mitigating the impacts of fire.

The structures of different land cover types provide a rich variety of responses to the radar illumination through time-varying polarization signatures. Because of the rapid changes in structure and composition after a disturbance like wildfire, radar backscatter may be useful for determining areas that burned, even when traditional methods do not work well.

5.4. Geologic and land hazard monitoring

5.4.1. *Earthquakes:*

Earthquakes occur suddenly, often with intense ground shaking that causes loss of life and
property. They and their aftershocks can induce landslides, lead to fires, and even bring
neighboring faults closer to rupture. InSAR analysis can be used to measure ground displacement associated with earthquakes as well as continued motion as faults continue to slip aseismically after large earthquakes. Surface disruption can also be measured with radar providing a proxy of where specifically damage occurs to buildings.

5.4.2. *Volcanoes:*

Volcanic eruptions produce a large range of associated hazards. Explosive eruptions eject ballistic rock fragments that can impact the surface up to several kilometers away from the vent. Smaller fragments are carried upward in eruption columns that sometimes reach the stratosphere, forming eruption clouds that pose a serious hazard to aircraft. Volcanic ash fall can collapse buildings, and even minor amounts can impact electrical systems. Volcanic gases contribute to health problems, and also to acid rain that causes corrosion and damage to regional ecosystems. Lava flows inundate property and destroy infrastructure. Volcanic mudflows have the potential to devastate entire cities even far from the source volcano.

Data from NISAR and other radar missions allow us to identify and monitor surface deformation at quiescent and active volcanoes using InSAR. Only InSAR has the capability to monitor deformation at virtually all of the world's potentially active volcanoes on land. InSAR observations allow us to build models of subsurface magma movement preceding, accompanying, and following eruptions.

5.4.3. *Sinkholes:*

Sinkholes form in many ways, all of which lead to collapse of the overburden soil above an underground void. Naturally occurring sinkholes form in areas where the rock can dissolve in water, which could be due to erosion or an underground stream. However, sinkholes also can form more slowly as gentle depressions that indicate where the rock is dissolving below ground.

Regular imaging with NISAR, which will image nearly all land globally, potentially detects movement started days prior to the collapse. InSAR analysis can be used to quickly and efficiently identify slow-collapse sinkholes that can cause damage to structures, protecting lives and property.

5.4.4. *Landslides:*

Landslides and related slope movements can begin and move in a wide variety of ways, depending on the geology, soil and ground-water conditions, vegetation cover,

and weather. Some landslides move at rates of a fraction of an inch per year. Others fail catastrophically and travel at rates of tens of miles per hour. Landslide sizes also vary greatly, from small shallow slumps and slides that are only a couple yards across to large deep-seated slides and flows that can reach lengths of several miles.

NISAR will be able to measure ground surface movement directly through repeat imaging of landslide prone regions and processing the sequence of images using InSAR. One major advantage of InSAR is that it can be used to measure surface deformation directly across large areas (hundreds of miles) with far greater accuracy than is possible with other commonly used remote sensing or ground-based monitoring techniques (e.g., GPS surveying).

5.5. Critical Infrastructure monitoring

5.5.1. *Water Management Infrastructure:*

Levees and dams serve multiple functions besides their primary function of flood prevention. Dams provide hydroelectric power, store and protect the water supply, and provide recreation areas where people relax, and where fish, birds, and game live and breed. Levees do not just keep water from inundating the land, but also channel water to communities and businesses where it is needed.

Remote sensing can improve upon and augment the traditional methods of detecting slope failures, subsidence, and seepage, all of which are indicators of the health of the infrastructure. Instead of determining ground movement from a change in two or more measurements of elevation, it is possible using NISAR to measure surface movement directly without knowledge of absolute surface elevation.

- Coastal erosion and shoreline change
- Transportation
- Facility situational awareness

5.6. Maritime and coastal ocean applications

- Hurricanes and wind storms
- Sea ice monitoring
- Coastal ocean circulation features
- Ocean surface wind speed
- Iceberg and ship detection
- Oil spills
- Flood hazards

5.7. Hydrology and underground reservoirs

5.7.1. *Flood delineation and forecasting:*

Flooding destroys property and infrastructure, and claims tens of thousands of lives every year, first from the flooding itself and then due to food insecurity caused by destroyed crops.

Due to water's high dielectric constant, calm surface water can be easily detected using SAR imagery. This makes SAR backscatter an excellent resource for determining flood inundation extents. The longer wavelength of NISAR's L-band sensor allows for the signal to penetrate through vegetation more effectively than C-band signals, making it more suitable for detecting inundated vegetation and delineating wetlands.

Because peak flooding can last for a short time, it may not be captured by NISAR's 12-day repeat cycle.. To forecast flooding, SAR imagery data must be combined with another more frequent data source, such as precipitation or river discharge data. Once a long enough record of SAR imagery is available (generally 3 years or more), it can be correlated to another data source. This allows potential flood areas to be identified, and can provide guidance for which crops to harvest early to prevent food shortages.

- Coastal inundation
- Soil moisture
- Groundwater withdrawal
- Oil and gas production
- Induced seismicity
- Rapid damage assessment

5.8. Cryosphere

- Glaciology
- Permafrost

6. Getting NISAR Data

Access to NISAR data is currently restricted to a website that stages a NISAR Sample Data Product Suite.

Once NISAR has been launched and data is available, users will be able to discover and access the data using the same platforms NASA has made available for accessing other SAR datasets. All NISAR products will be archived by ASF, and can be discovered through the Vertex Data Search and Earthdata Search map-based web interfaces. In addition, users can leverage the asf_search Python package to programmatically search for and access NISAR data.

7. User Support

- White papers and documents
- Story Maps
- Mission Homepage and Tutorials
- Earthdata Forum
- Frequently Asked Questions

7.1. White papers and documents

7.1.1. *Application white papers:*

- Mountain Glaciers and Snow Hydrology (4/15/2023)
- Earthquake! Tracking Location and Impact from Space (9/2/2022)
- Soil Moisture and Water Resources (2/26/2019)
- Damage Mapping (2/26/2019)
- Changes in Permafrost (2/26/2019)
- Sinkholes and Cavern Collapse (2/26/2017)
- Volcanic Hazards (2/26/2017)
- Landslides (2/26/2017)
- Floods (2/26/2017)
- Induced Seismicity (2/26/2017)
- Hazards in Texas (2/26/2017)
- Coastal Land Loss (2/26/2017)
- Oil Spills (2/26/2017)
- Ice Sheets, Glaciers, and Oceans (2/26/2017)
- Marine Hazards (2/26/2017)
- Sea Ice (2/26/2017)
- Fire Management (2/26/2017)
- Food Security (2/26/2017)
- Forest Resources (2/26/2017)
- Timber and Forest Disturbance (2/26/2017)
- Flood Forecasting (2/26/2017)
- Hazards in Florida (2/26/2017)
- Drought and Groundwater Withdrawal (2/26/2017)
- Oil, Gas, and Water Underground Reservoirs (2/26/2017)
- Levees and Dams (2/26/2017)
- Subsidence (2/26/2017)

7.1.2. *Mission specific documents:*

- NASA-ISRO SAR (NISAR) Mission Science Users' Handbook (8/1/2019)
- NISAR Utilization Plan (9/4/2018)
- NISAR Mission Calibration and Validation Plan (5/14/2018)

7.1.3. *Science white papers:*

- NISAR Brochure (11/10/2021)
- Earth Surface and Interior White Paper (1/1/2016)
- Cryosphere White Paper (1/1/2016)
- Ecosystems White Paper (1/1/2016)
- Water White Paper (1/1/2016)

7.2. Story Maps

NISAR in GIS (February 2025)
Spatial Subsetting for NISAR Data (April 2025)
Floods (2/16/2023)
Wildfires (7/16/2024)
Land Subsidence (3/20/2023)
Oil Slicks (3/20/2023)
Glacial Motion (6/8/2023)

7.3. Mission Homepage and Tutorials

NASA ISRO SAR (NISAR) mission
Work with NISAR Sample Data
ASF NISAR Hub

7.4. Earthdata Forum

The Earthdata Forum is the means of choice for communication on topics involving NASA Earth Science data. Subject matter experts answer questions approved by the moderators. Tags for the following categories are assigned to all topics in the forum: *Discipline*, *DAAC*, *Project*, and *Services/Usage*. These tags can be used to filter the topics, and tags specific to the NISAR mission have been added. Users will want to pay particular attention to *ASF* posts tagged with *Announcements*, *Data Recipes*, and *FAQ*.

Users can also contact ASF directly by emailing uso@asf.alaska.edu.

7.5. Frequently Asked Questions

A set of frequently asked questions (FAQs) will be developed over time and included in the Earthdata Forum. Data recipes will be developed based on FAQs when appropriate.

8. Appendix A - Acronyms

8.1. Appendix A: Acronyms

Acronym	Definition
ADT	Algorithm Development Team
AT	Along Track
AWS	Amazon Web Services
BFPQ	Block adaptive Floating-Point Quantization
Cal/Val	Calibration and Validation (also sometimes cal/val)
CDR	Critical Design Review
CF	Climate and Forecast
CPU	Central Processing Unit
CRSD	Calibration Raw Signal Data
CSV	Comma-separated values
DAAC	Distributed Active Archive Center
DEM	Digital Elevation Model
DN	Digital Number
EAR	Export Administration Regulations
ECMWF	European Centre for Medium-Range Weather Forecasts
ECEF	Earth Centered Earth Fixed
EPSG	European Petroleum Survey Group
ESA	European Space Agency
FM	Frequency Modulation
FOP	Forecast Orbit Ephemeris
FOV	Field of View
GCOV	Geocoded Polarimetric Covariance (also as L2_GCOV)
GCP	Ground Control Point
GDAL	Geospatial Data Abstraction Library
GDS	Ground Data System
GIS	Geographic Information System
GMTED	Global Multi-resolution Terrain Elevation Data
GOFF	Geocoded Pixel Offsets (also as L2_GOFF)
GPU	Graphics Processing Unit
GSLC	Geocoded Single Look Complex (also as L2_GSLC)
GUNW	Geocoded d Unwrapped Interferogram (also as L2_GUNW)
HDF5	Hierarchical Data Format version 5
HK,HKTM	Housekeeping Telemetry
InSAR	Interferometric Synthetic Aperture Radar

Acronym	Definition
ISCE	InSAR Scientific Computing Environment
ISCE3	InSAR Scientific Computing Environment Enhanced Edition (for NISAR)
ISO	International Organization for Standardization
ISRO	Indian Space Research Organisation (British spelling)
L0B	Level-0B (data)
L1	Level-1 (data)
L2	Level-2 (data)
LOS	Line-Of-Sight
LUT	Lookup Table
Mbps	Megabits per second
MHz	Megahertz
MOE	Medium-precision Orbit Ephemeris
NCSA	National Center for Supercomputing Applications
NetCDF4	Network Common Data Form version 4
NISAR	NASA-ISRO Synthetic Aperture Radar
NOE	Near-Realtime Orbit Ephemeris
PDR	Preliminary Design Review
PLM	Product Lifecycle Management
POD	Precision Orbit Determination
POE	Precision Orbit Ephemeris
PRF	Pulse Repetition Frequency
QA	Quality Assurance
REE	Radar Echo Emulator
RFI	Radio Frequency Interference
RIFG	Range-Doppler Interferogram (also as L1_RIFG)
ROFF	Range-Doppler Pixel Offsets (also as L1_ROFF)
RRSD	Radar Raw Signal Data
RRST	Radar Raw Science Telemetry
RSLC	Range-Doppler Single Look Complex (also as L1_RSLC)
RUNW	Range-Doppler UnWrapped Interferogram (also as L1_RUNW)
SAR	Synthetic Aperture Radar
SAS	Science Algorithm Software
SDS	Science Data System
SDT	Science Definition Team
SIS	Software Interface Specification

Acronym	Definition
SLC	Single Look Complex
SNAPHU	Statistical-cost, Network-flow Algorithm for Phase Unwrapping
SRTM	Shuttle Radar Topography Mission
ST	Science Team
TAI	International Atomic Time (Temps Atomique International)
TCF	Terrain Correction Factor
TEC	Total Electron Content
TFdb	Track-frame Database
SWST	Sampling Window Start Time
UR	Urgent Response
UTC	Universal Time Coordinated
UTM	Universal Transverse Mercator
WGS84	World Geodetic System 84
XML	eXtensible Markup Language (xml in code)
YAML	YAML Ain’t Markup Language

9. Appendix B - Naming conventions

9.1. Appendix B: Naming conventions

- NISAR L0B
- NISAR L1 RSLC, L2 GSLC, L2 GCOV, L3 SME2
- NISAR L1 RIFG, L1 RUNW, L2 GUNW, L1 ROFF, L2 GOFF

9.2. NISAR L0B

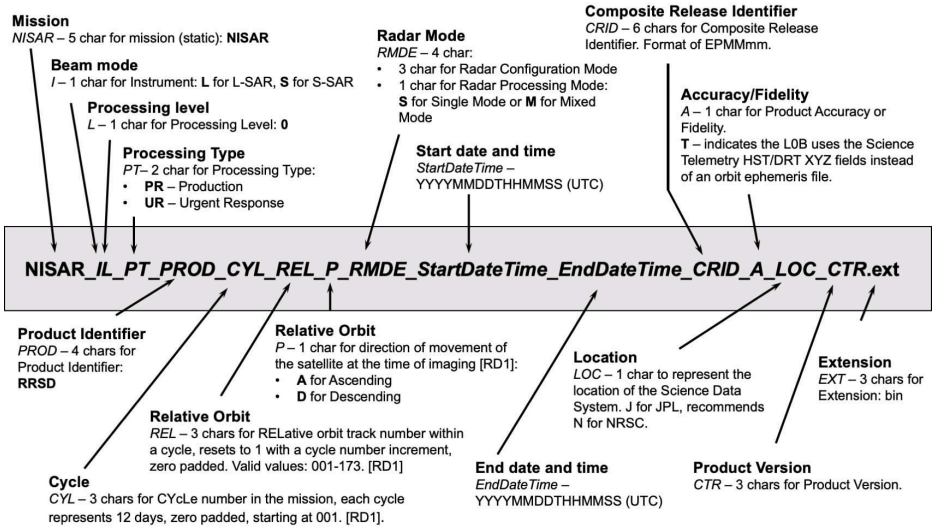

Figure 4: NISAR naming convention

9.3. NISAR L1 RSLC, L2 GSLC, L2 GCOV, L3 SME2

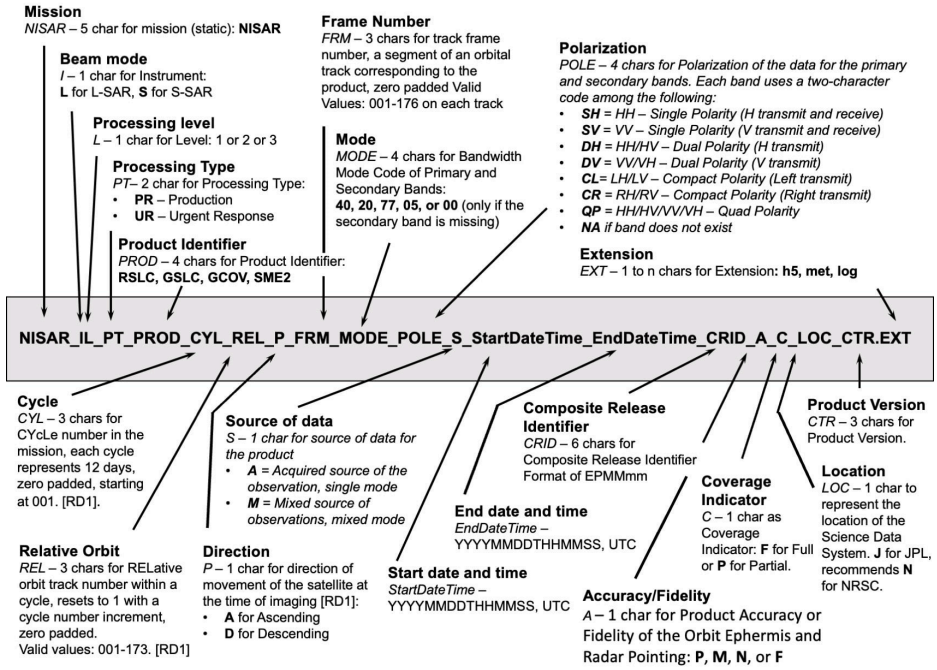

Figure 5: NISAR naming convention

9.4. NISAR L1 RIFG, L1 RUNW, L2 GUNW, L1 ROFF, L2 GOFF

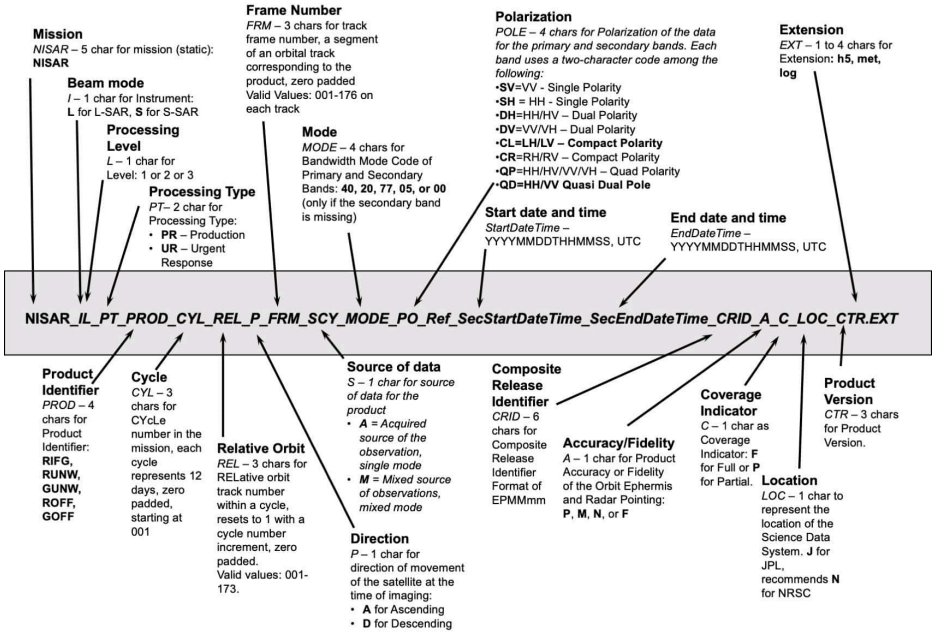

Figure 6: NISAR naming convention

10. Appendix C - NISAR metadata

10.1. Appendix C: NISAR metadata

- L1 RSLC
- L2 GSLC
- L1 RIFG
- L1 RUNW
- L2 GUNW
- L2 GCOV
- L1 ROFF
- L2 GOFF

10.2. L1 RSLC

Metadata


```
{
  "granule": "L1_SingleLookComplex",
  "science": {
    "LSAR": {
      "identification": {
        "absoluteOrbitNumber": "Absolute orbit number",
        "trackNumber": "Track number",
        "frameNumber": "Frame number",
        "missionId": "Mission identifier",
        "processingCenter": "Data processing center",
        "productType": "Product type",
        "granuleId": "Unique granule identification name",
        "productDoi": "Digital Object Identifier (DOI) for the product",
        "productVersion": "Product version which represents the
structure of the product and the science content governed by the
algorithm, input data, and processing parameters",
        "productSpecificationVersion": "Product specification version
which represents the schema of this product",
        "lookDirection": "Look direction, either \"Left\" or \"Right\"",
        "orbitPassDirection": "Orbit direction, either \"Ascending\" or
\"Descending\"",
        "zeroDopplerStartTime": "Azimuth start time (in UTC) of the
product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "zeroDopplerEndTime": "Azimuth stop time (in UTC) of the product
in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "plannedDatatakeId": "List of planned datatakes included in the
product",
        "plannedObservationId": "List of planned observations included
in the product",
        "isUrgentObservation": "Flag indicating if observation is
nominal (\"False\") or urgent (\"True\")",
        "listOfFrequencies": "List of frequency layers available in the
product",
        "listOfObservationModes": "List of observation modes of the
input L0B granules (one mode per L0B)",
        "diagnosticModeFlag": "Indicates if the radar operation mode is
a diagnostic mode (1-2) or DBFed science (0): 0, 1, or 2",
        "productLevel": "Product level. \"L0A\": Unprocessed instrument
data; \"L0B\": Reformatted, unprocessed instrument data; \"L1\":
Processed instrument data in radar coordinates system; and \"L2\":
Processed instrument data in geocoded coordinates system",
        "isGeocoded": "Flag to indicate if the product data is in the
radar geometry (\"False\") or in the map geometry (\"True\")",
        "boundingPolygon": "OGR compatible WKT representing the bounding
polygon of the image. Horizontal coordinates are WGS84 longitude
followed by latitude (both in degrees), and the vertical coordinate is
the height above the WGS84 ellipsoid in meters. The first point
corresponds to the start-time, near-range radar coordinate, and the
perimeter is traversed in counterclockwise order on the map. This means
```

```
the traversal order in radar coordinates differs for left-looking and
right-looking sensors. The polygon includes the four corners of the
radar grid, with equal numbers of points distributed evenly in radar
coordinates along each edge",
        "processingDateTime": "Processing date and time (in UTC) in the
format YYYY-mm-ddTHH:MM:SS",
        "radarBand": "Acquired frequency band, either \"L\" or \"S\"",
        "platformName": "Name of the platform used to collect the remote
sensing data provided in this product",
        "instrumentName": "Name of the instrument used to collect the
remote sensing data provided in this product",
        "processingType": "Processing pipeline used to generate this
granule. \"Nominal\": standard production system; \"Urgent\": time-
sensitive processing in response to urgent response events; \"Custom\":
user-initiated processing outside the nominal production system",
        "isDithered": "\"True\" if the pulse timing was varied
(dithered) during acquisition, \"False\" otherwise",
        "isMixedMode": "\"True\" if this product is a composite of data
collected in multiple radar modes, \"False\" otherwise",
        "isFullFrame": "\"True\" if the product fully covers a NISAR
frame, \"False\" if partial coverage",
        "compositeReleaseId": "Unique version identifier of the science
data production system",
        "isJointObservation": "\"True\" if any portion of this product
was acquired in a joint observation mode (e.g., L-band and S-band
simultaneously), \"False\" otherwise"
      },
      "RSLC": {
        "swaths": {
          "zeroDopplerTime": "Vector of zero Doppler azimuth times
measured relative to a UTC epoch",
          "zeroDopplerTimeSpacing": "Time interval in the along-track
direction for raster layers. This is same as the spacing between
consecutive entries in the zeroDopplerTime array",
          "frequencyA": {
            "listOfPolarizations": "List of processed polarization
layers with frequency A",
            "sceneCenterAlongTrackSpacing": "Nominal along-track spacing
in meters between consecutive lines near mid swath of the RSLC image",
            "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid swath of the RSLC
image",
            "processedRangeBandwidth": "Processed range bandwidth in
hertz",
            "acquiredRangeBandwidth": "Acquisition range bandwidth in
hertz. In case of mode combination, this corresponds to mode with
largest bandwidth",
            "processedAzimuthBandwidth": "Processed azimuth bandwidth in
hertz",
            "nominalAcquisitionPRF": "Nominal PRF of acquisition. In
case of mode combination, this corresponds to mode with least nominal
PRF",
```

```
            "processedCenterFrequency": "Center frequency of the
processed image in hertz",
            "acquiredCenterFrequency": "Center frequency of the
acquisition in hertz. In case of mode combination, this corresponds to
the mode with highest center frequency",
            "slantRangeSpacing": "Slant range spacing of grid. Same as
difference between consecutive samples in slantRange array",
            "slantRange": "Slant range dimension",
            "HH": "Focused RSLC image (HH)",
            "HV": "Focused RSLC image (HV)",
            "VH": "Focused RSLC image (VH)",
            "VV": "Focused RSLC image (VV)",
            "RH": "Focused RSLC image (RH)",
            "RV": "Focused RSLC image (RV)",
            "numberOfSubSwaths": "Number of swaths of continuous
imagery, due to transmit gaps",
            "validSamplesSubSwath1": "First and last valid sample in
each line of 1st subswath",
            "validSamplesSubSwath2": "First and last valid sample in
each line of 2nd subswath",
            "validSamplesSubSwath3": "First and last valid sample in
each line of 3rd subswath",
            "validSamplesSubSwath4": "First and last valid sample in
each line of 4th subswath",
            "validSamplesSubSwath5": "First and last valid sample in
each line of 5th subswath"
          },
          "frequencyB": {
            "listOfPolarizations": "List of processed polarization
layers with frequency B",
            "sceneCenterAlongTrackSpacing": "Nominal along-track spacing
in meters between consecutive lines near mid swath of the RSLC image",
            "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid swath of the RSLC
image",
            "processedRangeBandwidth": "Processed range bandwidth in
hertz",
            "acquiredRangeBandwidth": "Acquisition range bandwidth in
hertz. In case of mode combination, this corresponds to mode with
largest bandwidth",
            "processedAzimuthBandwidth": "Processed azimuth bandwidth in
hertz",
            "nominalAcquisitionPRF": "Nominal PRF of acquisition. In
case of mode combination, this corresponds to mode with least nominal
PRF",
            "processedCenterFrequency": "Center frequency of the
processed image in hertz",
            "acquiredCenterFrequency": "Center frequency of the
acquisition in hertz. In case of mode combination, this corresponds to
the mode with highest center frequency",
            "slantRangeSpacing": "Slant range spacing of grid. Same as
difference between consecutive samples in slantRange array",
```

```
            "slantRange": "Slant range dimension",
            "HH": "Focused RSLC image (HH)",
            "HV": "Focused RSLC image (HV)",
            "VH": "Focused RSLC image (VH)",
            "VV": "Focused RSLC image (VV)",
            "RH": "Focused RSLC image (RH)",
            "RV": "Focused RSLC image (RV)",
            "numberOfSubSwaths": "Number of swaths of continuous
imagery, due to transmit gaps",
            "validSamplesSubSwath1": "First and last valid sample in
each line of 1st subswath",
            "validSamplesSubSwath2": "First and last valid sample in
each line of 2nd subswath",
            "validSamplesSubSwath3": "First and last valid sample in
each line of 3rd subswath",
            "validSamplesSubSwath4": "First and last valid sample in
each line of 4th subswath",
            "validSamplesSubSwath5": "First and last valid sample in
each line of 5th subswath"
          }
        },
        "metadata": {
          "calibrationInformation": {
            "geometry": {
              "zeroDopplerTime": "Vector of zero Doppler azimuth times,
measured relative to a UTC epoch, corresponding to calibration records",
              "slantRange": "Slant range dimension corresponding to
calibration records",
              "beta0": "2D LUT to convert DN to beta 0 assuming as a
function of zero Doppler time and slant range",
              "sigma0": "2D LUT to convert DN to sigma 0 assuming as a
function of zero Doppler time and slant range",
              "gamma0": "2D LUT to convert DN to gamma 0 as a function
of zero Doppler time and slant range"
            },
            "frequencyA": {
              "elevationAntennaPattern": {
                "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to calibration
elevationAntennaPattern records",
                "slantRange": "Slant range dimension corresponding to
calibration elevationAntennaPattern records",
                "HH": "Complex two-way elevation antenna pattern",
                "HV": "Complex two-way elevation antenna pattern",
                "VH": "Complex two-way elevation antenna pattern",
                "VV": "Complex two-way elevation antenna pattern",
                "RH": "Complex two-way elevation antenna pattern",
                "RV": "Complex two-way elevation antenna pattern"
              },
              "noiseEquivalentBackscatter": {
                "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to calibration
```

```
noiseEquivalentBackscatter records",
                "slantRange": "Slant range dimension corresponding to
calibration noiseEquivalentBackscatter records",
                "HH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "HV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RV": "Noise equivalent backscatter in linear scale
(units of DN^2)"
              },
              "commonDelay": "Range delay correction applied to all
polarimetric channels",
              "faradayRotation": "Faraday rotation correction applied in
processing",
              "HH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HH channel",
                "differentialPhase": "Phase correction applied to HH
channel",
                "scaleFactor": "Scale factor applied to HH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HH
channel complex amplitude with respect to elevation angle"
              },
              "HV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HV channel",
                "differentialPhase": "Phase correction applied to HV
channel",
                "scaleFactor": "Scale factor applied to HV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HV
channel complex amplitude with respect to elevation angle"
              },
              "VH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
```

```
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VH channel",
                "differentialPhase": "Phase correction applied to VH
channel",
                "scaleFactor": "Scale factor applied to VH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VH
channel complex amplitude with respect to elevation angle"
              },
              "VV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VV channel",
                "differentialPhase": "Phase correction applied to VV
channel",
                "scaleFactor": "Scale factor applied to VV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VV
channel complex amplitude with respect to elevation angle"
              },
              "RH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RH channel",
                "differentialPhase": "Phase correction applied to RH
channel",
                "scaleFactor": "Scale factor applied to RH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RH
channel complex amplitude with respect to elevation angle"
              },
              "RV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RV channel",
                "differentialPhase": "Phase correction applied to RV
channel",
                "scaleFactor": "Scale factor applied to RV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RV
channel complex amplitude with respect to elevation angle"
              }
```

```
            },
            "frequencyB": {
              "elevationAntennaPattern": {
                "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to calibration
elevationAntennaPattern records",
                "slantRange": "Slant range dimension corresponding to
calibration elevationAntennaPattern records",
                "HH": "Complex two-way elevation antenna pattern",
                "HV": "Complex two-way elevation antenna pattern",
                "VH": "Complex two-way elevation antenna pattern",
                "VV": "Complex two-way elevation antenna pattern",
                "RH": "Complex two-way elevation antenna pattern",
                "RV": "Complex two-way elevation antenna pattern"
              },
              "noiseEquivalentBackscatter": {
                "HH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to calibration
noiseEquivalentBackscatter records",
                "slantRange": "Slant range dimension corresponding to
calibration noiseEquivalentBackscatter records",
                "HV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RV": "Noise equivalent backscatter in linear scale
(units of DN^2)"
              },
              "HH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HH channel",
                "differentialPhase": "Phase correction applied to HH
channel",
                "scaleFactor": "Scale factor applied to HH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HH
channel complex amplitude with respect to elevation angle"
              },
              "HV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
```

```
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HV channel",
                "differentialPhase": "Phase correction applied to HV
channel",
                "scaleFactor": "Scale factor applied to HV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HV
channel complex amplitude with respect to elevation angle"
              },
              "VH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VH channel",
                "differentialPhase": "Phase correction applied to VH
channel",
                "scaleFactor": "Scale factor applied to VH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VH
channel complex amplitude with respect to elevation angle"
              },
              "VV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VV channel",
                "differentialPhase": "Phase correction applied to VV
channel",
                "scaleFactor": "Scale factor applied to VV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VV
channel complex amplitude with respect to elevation angle"
              },
              "RH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RH channel",
                "differentialPhase": "Phase correction applied to RH
channel",
                "scaleFactor": "Scale factor applied to RH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RH
channel complex amplitude with respect to elevation angle"
              },
```

```
            "RV": {
              "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
              "differentialDelay": "Range delay correction applied to
RV channel",
              "differentialPhase": "Phase correction applied to RV
channel",
              "scaleFactor": "Scale factor applied to RV channel
complex amplitude (at antenna boresite)",
              "scaleFactorSlope": "Slope of scale factor applied to RV
channel complex amplitude with respect to elevation angle"
            },
            "commonDelay": "Range delay correction applied to all
polarimetric channels",
            "faradayRotation": "Faraday rotation correction applied in
processing"
          },
          "crosstalk": {
            "slantRange": "Slant range dimension corresponding to
crosstalk records",
            "txHorizontalCrosspol": "Crosstalk in H-transmit channel
expressed as ratio txV / txH",
            "txVerticalCrosspol": "Crosstalk in V-transmit channel
expressed as ratio txH / txV",
            "rxHorizontalCrosspol": "Crosstalk in H-receive channel
expressed as ratio rxV / rxH",
            "rxVerticalCrosspol": "Crosstalk in V-receive channel
expressed as ratio rxH / rxV"
          }
        },
        "processingInformation": {
          "parameters": {
            "azimuthChirpWeighting": "1D array in frequency domain for
azimuth processing. This is used for processing L0B to L1. FFT
length=256 (assumed)",
            "rangeChirpWeighting": "1D array in frequency domain for
range processing. This is used for processing L0B to L1. FFT length=256
(assumed)",
            "referenceTerrainHeight": "Reference terrain height as a
function of time",
            "zeroDopplerTime": "Vector of zero Doppler azimuth times,
measured relative to a UTC epoch, corresponding to processing
information records",
            "slantRange": "Slant range dimension corresponding to
processing information records",
            "frequencyA": {
              "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to processing
information records",
              "slantRange": "Slant range dimension corresponding to
```

```
processing information records",
                "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
              },
              "frequencyB": {
                "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to processing
information records",
                "slantRange": "Slant range dimension corresponding to
processing information records",
                "dopplerCentroid": "2D LUT of Doppler centroid for
frequency B"
              },
              "runConfigurationContents": "Contents of the run
configuration file with parameters used for processing"
            },
            "algorithms": {
              "demInterpolation": "DEM interpolation method",
              "rfiDetection": "Algorithm used for radio frequency
interference (RFI) detection",
              "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation, either \"ST-EVD\" or \"FDNF\" (or
\"disabled\" if no RFI mitigation was applied)",
              "rangeCompression": "Algorithm for focusing the data in
the range direction",
              "elevationAntennaPatternCorrection": "Algorithm for
calibrating the antenna pattern",
              "rangeSpreadingLossCorrection": "Algorithm for calibrating
range fading",
              "dopplerCentroidEstimation": "Algorithm for calculating
Doppler centroid",
              "azimuthPresumming": "Algorithm for regridding and filling
gaps in the raw data in azimuth",
              "azimuthCompression": "Algorithm for focusing the data in
the azimuth direction",
              "softwareVersion": "Software version used for processing"
            },
            "inputs": {
              "l0bGranules": "List of input L0B products used",
              "orbitFiles": "List of input orbit files used",
              "attitudeFiles": "List of input attitude files used",
              "auxcalFiles": "List of input calibration files used",
              "configFiles": "List of input config files used",
              "demSource": "Description of the input digital elevation
model (DEM)"
            }
          },
          "orbit": {
            "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
```

```
            "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
            "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
            "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
          },
          "attitude": {
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
            "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
            "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
            "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
          },
          "geolocationGrid": {
            "epsg": "EPSG code corresponding to coordinate system used
for representing geolocation grid",
            "coordinateY": "Y coordinates in specified EPSG code",
            "coordinateX": "X coordinates in specified EPSG code",
            "incidenceAngle": "Incidence angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the target
height",
            "losUnitVectorX": "East component of the line-of-sight (LOS)
unit vector, defined from the target to the sensor, expressed in the
east-north-up (ENU) coordinate system with its origin at the target
location",
            "losUnitVectorY": "North component of the line-of-sight
(LOS) unit vector, defined from the target to the sensor, expressed in
the east-north-up (ENU) coordinate system with its origin at the target
location",
            "alongTrackUnitVectorX": "East component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "alongTrackUnitVectorY": "North component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "elevationAngle": "Elevation angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the sensor",
            "slantRange": "Slant range values corresponding to the
geolocation grid",
            "zeroDopplerTime": "Vector of zero Doppler azimuth times,
measured relative to a UTC epoch, corresponding to the geolocation
grid",
            "groundTrackVelocity": "Absolute value of the platform
velocity scaled at the target height",
```

```
            "heightAboveEllipsoid": "Height values above WGS84 Ellipsoid
corresponding to the location grid"
          }
        }
      }
    }
  }
}
```


Data


```
{
  "granule": "L1_SingleLookComplex",
  "dimension": {
    "attitudeListLength": 17,
    "calibrationSlantRangeWidth": 17,
    "calibrationTimeLength": 29,
    "chirpFFTFrequency": 256,
    "dopplerCentroidSlantRangeWidth": 11,
    "dopplerCentroidTimeLength": 3,
    "firstLastPair": 2,
    "frequencyASlantRangeWidth": 19,
    "frequencyBSlantRangeWidth": 5,
    "geolocationCubeHeight": 19,
    "geolocationCubeLength": 13,
    "geolocationCubeWidth": 17,
    "orbitListLength": 11,
    "quaternions": 4,
    "tripletxyz": 3,
    "zeroDopplerTimeLength": 17
  },
  "science": {
    "LSAR": {
      "RSLC": {
        "swaths": {
          "zeroDopplerTime": {
            "type": "dimension",
            "dimension": "(zeroDopplerTimeLength)",
            "data_type": "float64",
            "shape": "(17)"
          },
          "frequencyA": {
            "slantRange": {
              "type": "dimension",
              "dimension": "(frequencyASlantRangeWidth)",
              "data_type": "float64",
              "shape": "(19)"
            },
            "HH": {
              "type": "dataset",
              "dimension":
"(zeroDopplerTimeLength,frequencyASlantRangeWidth)",
```

```
            "data_type": "cfloat32",
            "shape": "(17,19)"
          },
          "HV": {
            "type": "dataset",
            "dimension":
"(zeroDopplerTimeLength,frequencyASlantRangeWidth)",
            "data_type": "cfloat32",
            "shape": "(17,19)"
          },
          "VH": {
            "type": "dataset",
            "dimension":
"(zeroDopplerTimeLength,frequencyASlantRangeWidth)",
            "data_type": "cfloat32",
            "shape": "(17,19)"
          },
          "VV": {
            "type": "dataset",
            "dimension":
"(zeroDopplerTimeLength,frequencyASlantRangeWidth)",
            "data_type": "cfloat32",
            "shape": "(17,19)"
          },
          "RH": {
            "type": "dataset",
            "dimension":
"(zeroDopplerTimeLength,frequencyASlantRangeWidth)",
            "data_type": "cfloat32",
            "shape": "(17,19)"
          },
          "RV": {
            "type": "dataset",
            "dimension":
"(zeroDopplerTimeLength,frequencyASlantRangeWidth)",
            "data_type": "cfloat32",
            "shape": "(17,19)"
          },
          "validSamplesSubSwath1": {
            "type": "supplemental",
            "dimension": "(zeroDopplerTimeLength,firstLastPair)",
            "data_type": "uint32",
            "shape": "(17,2)"
          },
          "validSamplesSubSwath2": {
            "type": "supplemental",
            "dimension": "(zeroDopplerTimeLength,firstLastPair)",
            "data_type": "uint32",
            "shape": "(17,2)"
          },
          "validSamplesSubSwath3": {
            "type": "supplemental",
```

```
              "dimension": "(zeroDopplerTimeLength,firstLastPair)",
              "data_type": "uint32",
              "shape": "(17,2)"
            },
            "validSamplesSubSwath4": {
              "type": "supplemental",
              "dimension": "(zeroDopplerTimeLength,firstLastPair)",
              "data_type": "uint32",
              "shape": "(17,2)"
            },
            "validSamplesSubSwath5": {
              "type": "supplemental",
              "dimension": "(zeroDopplerTimeLength,firstLastPair)",
              "data_type": "uint32",
              "shape": "(17,2)"
            }
          },
          "frequencyB": {
            "slantRange": {
              "type": "dimension",
              "dimension": "(frequencyBSlantRangeWidth)",
              "data_type": "float64",
              "shape": "(5)"
            },
            "HH": {
              "type": "dataset",
              "dimension":
"(zeroDopplerTimeLength,frequencyBSlantRangeWidth)",
              "data_type": "cfloat32",
              "shape": "(17,5)"
            },
            "HV": {
              "type": "dataset",
              "dimension":
"(zeroDopplerTimeLength,frequencyBSlantRangeWidth)",
              "data_type": "cfloat32",
              "shape": "(17,5)"
            },
            "VH": {
              "type": "dataset",
              "dimension":
"(zeroDopplerTimeLength,frequencyBSlantRangeWidth)",
              "data_type": "cfloat32",
              "shape": "(17,5)"
            },
            "VV": {
              "type": "dataset",
              "dimension":
"(zeroDopplerTimeLength,frequencyBSlantRangeWidth)",
              "data_type": "cfloat32",
              "shape": "(17,5)"
            },
```

```
            "RH": {
              "type": "dataset",
              "dimension":
"(zeroDopplerTimeLength,frequencyBSlantRangeWidth)",
              "data_type": "cfloat32",
              "shape": "(17,5)"
            },
            "RV": {
              "type": "dataset",
              "dimension":
"(zeroDopplerTimeLength,frequencyBSlantRangeWidth)",
              "data_type": "cfloat32",
              "shape": "(17,5)"
            },
            "validSamplesSubSwath1": {
              "type": "supplemental",
              "dimension": "(zeroDopplerTimeLength,firstLastPair)",
              "data_type": "uint32",
              "shape": "(17,2)"
            },
            "validSamplesSubSwath2": {
              "type": "supplemental",
              "dimension": "(zeroDopplerTimeLength,firstLastPair)",
              "data_type": "uint32",
              "shape": "(17,2)"
            },
            "validSamplesSubSwath3": {
              "type": "supplemental",
              "dimension": "(zeroDopplerTimeLength,firstLastPair)",
              "data_type": "uint32",
              "shape": "(17,2)"
            },
            "validSamplesSubSwath4": {
              "type": "supplemental",
              "dimension": "(zeroDopplerTimeLength,firstLastPair)",
              "data_type": "uint32",
              "shape": "(17,2)"
            },
            "validSamplesSubSwath5": {
              "type": "supplemental",
              "dimension": "(zeroDopplerTimeLength,firstLastPair)",
              "data_type": "uint32",
              "shape": "(17,2)"
            }
          }
        },
        "metadata": {
          "calibrationInformation": {
            "geometry": {
              "zeroDopplerTime": {
                "type": "dimension",
                "dimension": "(calibrationTimeLength)",
```

```
                "data_type": "float64",
                "shape": "(29)"
              },
              "slantRange": {
                "type": "dimension",
                "dimension": "(calibrationSlantRangeWidth)",
                "data_type": "float64",
                "shape": "(17)"
              },
              "beta0": {
                "type": "supplemental",
                "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,17)"
              },
              "sigma0": {
                "type": "supplemental",
                "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,17)"
              },
              "gamma0": {
                "type": "supplemental",
                "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,17)"
              }
            },
            "frequencyA": {
              "elevationAntennaPattern": {
                "zeroDopplerTime": {
                  "type": "dimension",
                  "dimension": "(calibrationTimeLength)",
                  "data_type": "float64",
                  "shape": "(29)"
                },
                "slantRange": {
                  "type": "dimension",
                  "dimension": "(calibrationSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "HH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
```

```
                "HV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "VH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "VV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "RH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "RV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                }
              },
              "noiseEquivalentBackscatter": {
                "zeroDopplerTime": {
                  "type": "dimension",
                  "dimension": "(calibrationTimeLength)",
                  "data_type": "float64",
                  "shape": "(29)"
                },
                "slantRange": {
                  "type": "dimension",
                  "dimension": "(calibrationSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "HH": {
                  "type": "supplemental",
                  "dimension":
```

```
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "HV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "VH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "VV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "RH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "RV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                }
              }
            },
            "frequencyB": {
              "elevationAntennaPattern": {
                "zeroDopplerTime": {
                  "type": "dimension",
                  "dimension": "(calibrationTimeLength)",
                  "data_type": "float64",
                  "shape": "(29)"
                },
                "slantRange": {
                  "type": "dimension",
                  "dimension": "(calibrationSlantRangeWidth)",
```

```
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "HH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "HV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "VH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "VV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "RH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                },
                "RV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(29,17)"
                }
              },
              "noiseEquivalentBackscatter": {
                "HH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
```

```
                  "shape": "(29,17)"
                },
                "zeroDopplerTime": {
                  "type": "dimension",
                  "dimension": "(calibrationTimeLength)",
                  "data_type": "float64",
                  "shape": "(29)"
                },
                "slantRange": {
                  "type": "dimension",
                  "dimension": "(calibrationSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "HV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "VH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "VV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "RH": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                },
                "RV": {
                  "type": "supplemental",
                  "dimension":
"(calibrationTimeLength,calibrationSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(29,17)"
                }
              }
            },
            "crosstalk": {
```

```
      "slantRange": {
        "type": "dimension",
        "dimension": "(calibrationSlantRangeWidth)",
        "data_type": "float64",
        "shape": "(17)"
      },
      "txHorizontalCrosspol": {
        "type": "dimension",
        "dimension": "(calibrationSlantRangeWidth)",
        "data_type": "cfloat32",
        "shape": "(17)"
      },
      "txVerticalCrosspol": {
        "type": "dimension",
        "dimension": "(calibrationSlantRangeWidth)",
        "data_type": "cfloat32",
        "shape": "(17)"
      },
      "rxHorizontalCrosspol": {
        "type": "dimension",
        "dimension": "(calibrationSlantRangeWidth)",
        "data_type": "cfloat32",
        "shape": "(17)"
      },
      "rxVerticalCrosspol": {
        "type": "dimension",
        "dimension": "(calibrationSlantRangeWidth)",
        "data_type": "cfloat32",
        "shape": "(17)"
      }
    }
  },
  "processingInformation": {
    "parameters": {
      "azimuthChirpWeighting": {
        "type": "dimension",
        "dimension": "(chirpFFTFrequency)",
        "data_type": "float32",
        "shape": "(256)"
      },
      "rangeChirpWeighting": {
        "type": "dimension",
        "dimension": "(chirpFFTFrequency)",
        "data_type": "float32",
        "shape": "(256)"
      },
      "referenceTerrainHeight": {
        "type": "dimension",
        "dimension": "(dopplerCentroidTimeLength)",
        "data_type": "float32",
        "shape": "(3)"
      },
```

```
                "zeroDopplerTime": {
                  "type": "dimension",
                  "dimension": "(dopplerCentroidTimeLength)",
                  "data_type": "float64",
                  "shape": "(3)"
                },
                "slantRange": {
                  "type": "dimension",
                  "dimension": "(dopplerCentroidSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(11)"
                },
                "frequencyA": {
                  "zeroDopplerTime": {
                    "type": "dimension",
                    "dimension": "(dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(11)"
                  },
                  "slantRange": {
                    "type": "dimension",
                    "dimension": "(dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(11)"
                  },
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(3,11)"
                  }
                },
                "frequencyB": {
                  "zeroDopplerTime": {
                    "type": "dimension",
                    "dimension": "(dopplerCentroidTimeLength)",
                    "data_type": "float64",
                    "shape": "(3)"
                  },
                  "slantRange": {
                    "type": "dimension",
                    "dimension": "(dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(11)"
                  },
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(3,11)"
```

```
            }
          }
        }
      },
      "orbit": {
        "time": {
          "type": "dimension",
          "dimension": "(orbitListLength)",
          "data_type": "float64",
          "shape": "(11)"
        },
        "position": {
          "type": "supplemental",
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(11,3)"
        },
        "velocity": {
          "type": "supplemental",
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(11,3)"
        }
      },
      "attitude": {
        "time": {
          "type": "dimension",
          "dimension": "(attitudeListLength)",
          "data_type": "float64",
          "shape": "(17)"
        },
        "quaternions": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,quaternions)",
          "data_type": "float64",
          "shape": "(17,4)"
        },
        "eulerAngles": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(17,3)"
        }
      },
      "geolocationGrid": {
        "coordinateY": {
          "type": "cube",
          "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
          "data_type": "float64",
          "shape": "(19,13,17)"
        },
```

```
            "coordinateX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(19,13,17)"
            },
            "incidenceAngle": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(19,13,17)"
            },
            "losUnitVectorX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(19,13,17)"
            },
            "losUnitVectorY": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(19,13,17)"
            },
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(19,13,17)"
            },
            "alongTrackUnitVectorY": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(19,13,17)"
            },
            "elevationAngle": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(19,13,17)"
            },
            "slantRange": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeWidth)",
```

```
            "data_type": "float64",
            "shape": "(17)"
          },
          "zeroDopplerTime": {
            "type": "cube dimension",
            "dimension": "(geolocationCubeWidth)",
            "data_type": "float64",
            "shape": "(17)"
          },
          "heightAboveEllipsoid": {
            "type": "cube dimension",
            "dimension": "(geolocationCubeHeight)",
            "data_type": "float64",
            "shape": "(19)"
          }
        }
      }
    }
  }
}
```

10.3. L2 GSLC

Metadata


```
{
  "granule": "L2_GeocodedSlc",
  "science": {
    "LSAR": {
      "identification": {
        "absoluteOrbitNumber": "Absolute orbit number",
        "trackNumber": "Track number",
        "frameNumber": "Frame number",
        "missionId": "Mission identifier",
        "processingCenter": "Data processing center",
        "productType": "Product type",
        "granuleId": "Unique granule identification name",
        "productDoi": "Digital Object Identifier (DOI) for the product",
        "productVersion": "Product version which represents the
structure of the product and the science content governed by the
algorithm, input data, and processing parameters",
        "productSpecificationVersion": "Product specification version
which represents the schema of this product",
        "lookDirection": "Look direction, either \"Left\" or \"Right\"",
        "orbitPassDirection": "Orbit direction, either \"Ascending\" or
\"Descending\"",
        "zeroDopplerStartTime": "Azimuth start time (in UTC) of the
product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "zeroDopplerEndTime": "Azimuth stop time (in UTC) of the product
in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "plannedDatatakeId": "List of planned datatakes included in the
product",
        "plannedObservationId": "List of planned observations included
in the product",
        "isUrgentObservation": "Flag indicating if observation is
nominal (\"False\") or urgent (\"True\")",
        "listOfFrequencies": "List of frequency layers available in the
product",
        "listOfObservationModes": "List of observation modes of the L0B
granules used to generate the input RSLC (one mode per L0B)",
        "diagnosticModeFlag": "Indicates if the radar operation mode is
a diagnostic mode (1-2) or DBFed science (0): 0, 1, or 2",
        "productLevel": "Product level. \"L0A\": Unprocessed instrument
data; \"L0B\": Reformatted, unprocessed instrument data; \"L1\":
Processed instrument data in radar coordinates system; and \"L2\":
Processed instrument data in geocoded coordinates system",
        "isGeocoded": "Flag to indicate if the product data is in the
radar geometry (\"False\") or in the map geometry (\"True\")",
        "boundingPolygon": "OGR compatible WKT representing the bounding
polygon of the image. Horizontal coordinates are WGS84 longitude
followed by latitude (both in degrees), and the vertical coordinate is
the height above the WGS84 ellipsoid in meters. The first point
corresponds to the start-time, near-range radar coordinate, and the
perimeter is traversed in counterclockwise order on the map. This means
```

```
the traversal order in radar coordinates differs for left-looking and
right-looking sensors. The polygon includes the four corners of the
radar grid, with equal numbers of points distributed evenly in radar
coordinates along each edge",
        "processingDateTime": "Processing date and time (in UTC) in the
format YYYY-mm-ddTHH:MM:SS",
        "radarBand": "Acquired frequency band, either \"L\" or \"S\"",
        "platformName": "Name of the platform used to collect the remote
sensing data provided in this product",
        "instrumentName": "Name of the instrument used to collect the
remote sensing data provided in this product",
        "processingType": "Processing pipeline used to generate this
granule. \"Nominal\": standard production system; \"Urgent\": time-
sensitive processing in response to urgent response events; \"Custom\":
user-initiated processing outside the nominal production system",
        "isDithered": "\"True\" if the pulse timing was varied
(dithered) during acquisition, \"False\" otherwise",
        "isMixedMode": "\"True\" if this product is a composite of data
collected in multiple radar modes, \"False\" otherwise",
        "isFullFrame": "\"True\" if the product fully covers a NISAR
frame, \"False\" if partial coverage",
        "compositeReleaseId": "Unique version identifier of the science
data production system",
        "isJointObservation": "\"True\" if any portion of this product
was acquired in a joint observation mode (e.g., L-band and S-band
simultaneously), \"False\" otherwise"
      },
      "GSLC": {
        "identification": {
          "staticLayersDataAccess": "Location of the static layers
product associated with this product (URL or DOI)"
        },
        "grids": {
          "frequencyA": {
            "listOfPolarizations": "List of processed polarization
layers with frequency A",
            "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
            "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
            "rangeBandwidth": "Processed range bandwidth in hertz",
            "azimuthBandwidth": "Processed azimuth bandwidth in hertz",
            "centerFrequency": "Center frequency of the processed image
in hertz",
            "slantRangeSpacing": "Slant range spacing of grid. Same as
difference between consecutive samples in slantRange array",
            "zeroDopplerTimeSpacing": "Time interval in the along-track
direction for raster layers. This is same as the spacing between
consecutive entries in the zeroDopplerTime array",
            "projection": "Product map grid projection: EPSG code, with
additional projection information as HDF5 Attributes",
            "xCoordinates": "X coordinates in specified projection",
```

```
            "yCoordinates": "Y coordinates in specified projection",
            "mask": "Mask indicating the subswath number representing
valid GSLC samples. Each GSLC pixel is assumed valid if all the pixels
in the interpolation window are fully focused in the input RSLC. A value
of 0 indicates that at least one RSLC pixel in the interpolation window
is partially focused or invalid. Pixels outside of the radar acquisition
extent are filled with the value 255",
            "HH": "Focused SLC image (HH)",
            "HV": "Focused SLC image (HV)",
            "VH": "Focused SLC image (VH)",
            "VV": "Focused SLC image (VV)",
            "RH": "Focused SLC image (RH)",
            "RV": "Focused SLC image (RV)",
            "numberOfSubSwaths": "Number of swaths of continuous
imagery, due to transmit gaps"
          },
          "frequencyB": {
            "listOfPolarizations": "List of processed polarization
layers with frequency B",
            "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
            "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
            "rangeBandwidth": "Processed range bandwidth in hertz",
            "azimuthBandwidth": "Processed azimuth bandwidth in hertz",
            "centerFrequency": "Center frequency of the processed image
in hertz",
            "slantRangeSpacing": "Slant range spacing of grid. Same as
difference between consecutive samples in slantRange array",
            "zeroDopplerTimeSpacing": "Time interval in the along-track
direction for raster layers. This is same as the spacing between
consecutive entries in the zeroDopplerTime array",
            "projection": "Product map grid projection: EPSG code, with
additional projection information as HDF5 Attributes",
            "xCoordinates": "X coordinates in specified projection",
            "yCoordinates": "Y coordinates in specified projection",
            "mask": "Mask indicating the subswath number representing
valid GSLC samples. Each GSLC pixel is assumed valid if all the pixels
in the interpolation window are fully focused in the input RSLC. A value
of 0 indicates that at least one RSLC pixel in the interpolation window
is partially focused or invalid. Pixels outside of the radar acquisition
extent are filled with the value 255",
            "HH": "Focused SLC image (HH)",
            "HV": "Focused SLC image (HV)",
            "VH": "Focused SLC image (VH)",
            "VV": "Focused SLC image (VV)",
            "RH": "Focused SLC image (RH)",
            "RV": "Focused SLC image (RV)",
            "numberOfSubSwaths": "Number of swaths of continuous
imagery, due to transmit gaps"
          }
        },
```

```
        "metadata": {
          "calibrationInformation": {
            "geometry": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "beta0": "2D LUT to convert DN to beta 0 assuming as a
function of geographical location",
              "sigma0": "2D LUT to convert DN to sigma 0 assuming as a
function of geographical location",
              "gamma0": "2D LUT to convert DN to gamma 0 as a function
of geographical location"
            },
            "frequencyA": {
              "elevationAntennaPattern": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinates": "X coordinates in specified projection",
                "HH": "Complex two-way elevation antenna pattern",
                "HV": "Complex two-way elevation antenna pattern",
                "VH": "Complex two-way elevation antenna pattern",
                "VV": "Complex two-way elevation antenna pattern",
                "RH": "Complex two-way elevation antenna pattern",
                "RV": "Complex two-way elevation antenna pattern"
              },
              "noiseEquivalentBackscatter": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinates": "X coordinates in specified projection",
                "HH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "HV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RV": "Noise equivalent backscatter in linear scale
(units of DN^2)"
              },
              "commonDelay": "Range delay correction applied to all
polarimetric channels",
              "faradayRotation": "Faraday rotation correction applied in
processing",
              "HH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
```

```
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HH channel",
                "differentialPhase": "Phase correction applied to HH
channel",
                "scaleFactor": "Scale factor applied to HH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HH
channel complex amplitude with respect to elevation angle"
              },
              "HV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HV channel",
                "differentialPhase": "Phase correction applied to HV
channel",
                "scaleFactor": "Scale factor applied to HV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HV
channel complex amplitude with respect to elevation angle"
              },
              "VH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VH channel",
                "differentialPhase": "Phase correction applied to VH
channel",
                "scaleFactor": "Scale factor applied to VH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VH
channel complex amplitude with respect to elevation angle"
              },
              "VV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VV channel",
                "differentialPhase": "Phase correction applied to VV
channel",
                "scaleFactor": "Scale factor applied to VV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VV
channel complex amplitude with respect to elevation angle"
```

```
            },
            "RH": {
              "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
              "differentialDelay": "Range delay correction applied to
RH channel",
              "differentialPhase": "Phase correction applied to RH
channel",
              "scaleFactor": "Scale factor applied to RH channel
complex amplitude (at antenna boresite)",
              "scaleFactorSlope": "Slope of scale factor applied to RH
channel complex amplitude with respect to elevation angle"
            },
            "RV": {
              "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
              "differentialDelay": "Range delay correction applied to
RV channel",
              "differentialPhase": "Phase correction applied to RV
channel",
              "scaleFactor": "Scale factor applied to RV channel
complex amplitude (at antenna boresite)",
              "scaleFactorSlope": "Slope of scale factor applied to RV
channel complex amplitude with respect to elevation angle"
            }
          },
          "frequencyB": {
            "elevationAntennaPattern": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "HH": "Complex two-way elevation antenna pattern",
              "HV": "Complex two-way elevation antenna pattern",
              "VH": "Complex two-way elevation antenna pattern",
              "VV": "Complex two-way elevation antenna pattern",
              "RH": "Complex two-way elevation antenna pattern",
              "RV": "Complex two-way elevation antenna pattern"
            },
            "noiseEquivalentBackscatter": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "HH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
              "HV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
```

```
                "VH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RV": "Noise equivalent backscatter in linear scale
(units of DN^2)"
              },
              "HH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HH channel",
                "differentialPhase": "Phase correction applied to HH
channel",
                "scaleFactor": "Scale factor applied to HH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HH
channel complex amplitude with respect to elevation angle"
              },
              "HV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HV channel",
                "differentialPhase": "Phase correction applied to HV
channel",
                "scaleFactor": "Scale factor applied to HV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HV
channel complex amplitude with respect to elevation angle"
              },
              "VH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VH channel",
                "differentialPhase": "Phase correction applied to VH
channel",
                "scaleFactor": "Scale factor applied to VH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VH
channel complex amplitude with respect to elevation angle"
              },
              "VV": {
```

```
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VV channel",
                "differentialPhase": "Phase correction applied to VV
channel",
                "scaleFactor": "Scale factor applied to VV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VV
channel complex amplitude with respect to elevation angle"
              },
              "RH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RH channel",
                "differentialPhase": "Phase correction applied to RH
channel",
                "scaleFactor": "Scale factor applied to RH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RH
channel complex amplitude with respect to elevation angle"
              },
              "RV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RV channel",
                "differentialPhase": "Phase correction applied to RV
channel",
                "scaleFactor": "Scale factor applied to RV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RV
channel complex amplitude with respect to elevation angle"
              },
              "commonDelay": "Range delay correction applied to all
polarimetric channels",
              "faradayRotation": "Faraday rotation correction applied in
processing"
            },
            "crosstalk": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "txHorizontalCrosspol": "Crosstalk in H-transmit channel
```

```
expressed as ratio txV / txH",
              "txVerticalCrosspol": "Crosstalk in V-transmit channel
expressed as ratio txH / txV",
              "rxHorizontalCrosspol": "Crosstalk in H-receive channel
expressed as ratio rxV / rxH",
              "rxVerticalCrosspol": "Crosstalk in V-receive channel
expressed as ratio rxH / rxV"
            }
          },
          "processingInformation": {
            "parameters": {
              "azimuthChirpWeighting": "1D array in frequency domain for
azimuth processing. This is used for processing L0B to L1. FFT
length=256 (assumed)",
              "rangeChirpWeighting": "1D array in frequency domain for
range processing. This is used for processing L0B to L1. FFT length=256
(assumed)",
              "dryTroposphericGeolocationCorrectionApplied": "Flag to
indicate if the dry tropospheric correction was applied during the
generation of the input RSLC to improve geolocation",
              "wetTroposphericGeolocationCorrectionApplied": "Flag to
indicate if the wet tropospheric correction was applied during the
generation of the input RSLC to improve geolocation",
              "rangeIonosphericGeolocationCorrectionApplied": "Flag to
indicate if the range ionospheric correction has been applied to improve
geolocation",
              "azimuthIonosphericGeolocationCorrectionApplied": "Flag to
indicate if the azimuth ionospheric correction has been applied to
improve geolocation",
              "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the input RSLC",
              "ellipsoidalFlatteningApplied": "Flag to indicate if the
GSLC phase has been flattened with respect to a zero height ellipsoid",
              "topographicFlatteningApplied": "Flag to indicate if the
GSLC phase has been flattened with respect to topographic height using a
DEM",
              "referenceTerrainHeight": "Reference terrain height as a
function of map coordinates",
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "frequencyA": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinates": "X coordinates in specified projection",
                "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
              },
              "frequencyB": {
```

```
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinates": "X coordinates in specified projection",
                "dopplerCentroid": "2D LUT of Doppler centroid for
frequency B"
              },
              "runConfigurationContents": "Contents of the run
configuration file with parameters used for processing"
            },
            "timingCorrections": {
              "frequencyA": {
                "azimuthIonosphere": "Azimuth ionospheric timing
correction, derived from Total Electron Content (TEC) data",
                "slantRangeIonosphere": "Slant range ionospheric timing
correction, derived from Total Electron Content (TEC) data",
                "slantRangeSolidEarthTides": "2D lookup table of line-
of-sight solid Earth tides timing correction",
                "slantRange": "Slant range dimension corresponding to
the timing correction lookup tables",
                "slantRangeSpacing": "Slant range spacing of the timing
correction lookup tables",
                "zeroDopplerTime": "Zero Doppler time dimension since
UTC epoch corresponding to the timing correction lookup tables",
                "zeroDopplerTimeSpacing": "Time interval in the along-
track direction of the timing correction lookup tables"
              },
              "frequencyB": {
                "azimuthIonosphere": "Azimuth ionospheric timing
correction, derived from Total Electron Content (TEC) data",
                "slantRangeIonosphere": "Slant range ionospheric timing
correction, derived from Total Electron Content (TEC) data",
                "slantRangeSolidEarthTides": "2D lookup table of line-
of-sight solid Earth tides timing correction",
                "slantRange": "Slant range dimension corresponding to
the timing correction lookup tables",
                "slantRangeSpacing": "Slant range spacing of the timing
correction lookup tables",
                "zeroDopplerTime": "Zero Doppler time dimension since
UTC epoch corresponding to the timing correction lookup tables",
                "zeroDopplerTimeSpacing": "Time interval in the along-
track direction of the timing correction lookup tables"
              }
            },
            "algorithms": {
              "softwareVersion": "Software version used for processing",
              "demInterpolation": "DEM interpolation method",
              "geocoding": "Geocoding algorithm"
            },
            "inputs": {
              "l1SlcGranules": "List of input L1 products used",
              "orbitFiles": "List of input orbit files used",
```

```
              "tecFiles": "List of input total electron content (TEC)
files used",
              "configFiles": "List of input config files used",
              "demSource": "Description of the input digital elevation
model (DEM)"
            }
          },
          "ceosAnalysisReadyData": {
            "boundingBox": "OGR compatible WKT representing the bounding
box of the product (the union of the geocoded grid for available
frequency A and frequency B image datasets). Horizontal coordinates are
X and Y directions of the geocoded grid coordinate system, and the
vertical coordinate is the height above the WGS84 ellipsoid in meters.
The first point corresponds to the top left edge of the bounding box.
The remaining points are traversed in counterclockwise order on the map.
The fifth point closes the bounding box by repeating the first point",
            "ceosAnalysisReadyDataProductType": "CEOS Analysis Ready
Data (CARD) product type",
            "ceosAnalysisReadyDataDocumentIdentifier": "CEOS Analysis
Ready Data (CARD) document identifier",
            "geometricAccuracy": {
              "bias": {
                "y": "An estimate of the localization error bias in the
Y/northing direction",
                "x": "An estimate of the localization error bias in the
X/easting direction"
              },
              "standardDeviation": {
                "y": "An estimate of the localization error standard
deviation in the Y/northing direction",
                "x": "An estimate of the localization error standard
deviation in the X/easting direction"
              }
            }
          },
          "orbit": {
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
            "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
            "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
            "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris",
            "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\""
          },
          "attitude": {
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
```

```
            "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
            "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
            "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
          },
          "radarGrid": {
            "zeroDopplerAzimuthTime": "Zero Doppler azimuth time in
seconds since UTC epoch",
            "slantRange": "Slant range in meters",
            "incidenceAngle": "Incidence angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the target
height",
            "losUnitVectorX": "East component of the line-of-sight (LOS)
unit vector, defined from the target to the sensor, expressed in the
east-north-up (ENU) coordinate system with its origin at the target
location",
            "losUnitVectorY": "North component of the line-of-sight
(LOS) unit vector, defined from the target to the sensor, expressed in
the east-north-up (ENU) coordinate system with its origin at the target
location",
            "alongTrackUnitVectorX": "East component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "alongTrackUnitVectorY": "North component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "elevationAngle": "Elevation angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the sensor",
            "groundTrackVelocity": "Absolute value of the platform
velocity scaled at the target height",
            "projection": "Product map grid projection: EPSG code, with
additional projection information as HDF5 Attributes",
            "xCoordinates": "X coordinates in specified projection",
            "yCoordinates": "Y coordinates in specified projection",
            "heightAboveEllipsoid": "Height values above WGS84 Ellipsoid
corresponding to the radar grid"
          }
        }
      }
    }
  }
}
```

Data

```
{
  "granule": "L2_GeocodedSlc",
  "dimension": {
```

```
    "attitudeListLength": 11,
    "calibrationLength": 17,
    "calibrationWidth": 29,
    "chirpFFTFrequency": 256,
    "dopplerCentroidLength": 29,
    "dopplerCentroidWidth": 3,
    "frequencyALength": 17,
    "frequencyAWidth": 3,
    "frequencyBLength": 17,
    "frequencyBWidth": 19,
    "orbitListLength": 3,
    "quaternions": 4,
    "radarCubeHeight": 17,
    "radarCubeLength": 5,
    "radarCubeWidth": 23,
    "timingCorrectionLength": 3,
    "timingCorrectionWidth": 11,
    "tripletxyz": 3
  },
  "science": {
    "LSAR": {
      "GSLC": {
        "grids": {
          "frequencyA": {
            "xCoordinates": {
              "type": "dimension",
              "dimension": "(frequencyAWidth)",
              "data_type": "float64",
              "shape": "(3)"
            },
            "yCoordinates": {
              "type": "dimension",
              "dimension": "(frequencyALength)",
              "data_type": "float64",
              "shape": "(17)"
            },
            "mask": {
              "type": "supplemental",
              "dimension": "(frequencyALength,frequencyAWidth)",
              "data_type": "ubyte",
              "shape": "(17,3)"
            },
            "HH": {
              "type": "dataset",
              "dimension": "(frequencyALength,frequencyAWidth)",
              "data_type": "cfloat32",
              "shape": "(17,3)"
            },
            "HV": {
              "type": "dataset",
              "dimension": "(frequencyALength,frequencyAWidth)",
              "data_type": "cfloat32",
```

```
        "shape": "(17,3)"
      },
      "VH": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "cfloat32",
        "shape": "(17,3)"
      },
      "VV": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "cfloat32",
        "shape": "(17,3)"
      },
      "RH": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "cfloat32",
        "shape": "(17,3)"
      },
      "RV": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "cfloat32",
        "shape": "(17,3)"
      }
    },
    "frequencyB": {
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(frequencyBWidth)",
        "data_type": "float64",
        "shape": "(19)"
      },
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(frequencyBLength)",
        "data_type": "float64",
        "shape": "(17)"
      },
      "mask": {
        "type": "supplemental",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "ubyte",
        "shape": "(17,19)"
      },
      "HH": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(17,19)"
      },
```

```
      "HV": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(17,19)"
      },
      "VH": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(17,19)"
      },
      "VV": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(17,19)"
      },
      "RH": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(17,19)"
      },
      "RV": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(17,19)"
      }
    }
  },
  "metadata": {
    "calibrationInformation": {
      "geometry": {
        "yCoordinates": {
          "type": "dimension",
          "dimension": "(calibrationLength)",
          "data_type": "float64",
          "shape": "(17)"
        },
        "xCoordinates": {
          "type": "dimension",
          "dimension": "(calibrationWidth)",
          "data_type": "float64",
          "shape": "(29)"
        },
        "beta0": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "float32",
          "shape": "(17,29)"
```

```
      },
      "sigma0": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "gamma0": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      }
    },
    "frequencyA": {
      "elevationAntennaPattern": {
        "yCoordinates": {
          "type": "dimension",
          "dimension": "(calibrationLength)",
          "data_type": "float64",
          "shape": "(17)"
        },
        "xCoordinates": {
          "type": "dimension",
          "dimension": "(calibrationWidth)",
          "data_type": "float64",
          "shape": "(29)"
        },
        "HH": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "cfloat32",
          "shape": "(17,29)"
        },
        "HV": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "cfloat32",
          "shape": "(17,29)"
        },
        "VH": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "cfloat32",
          "shape": "(17,29)"
        },
        "VV": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "cfloat32",
          "shape": "(17,29)"
        },
```

```
      "RH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "cfloat32",
        "shape": "(17,29)"
      },
      "RV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "cfloat32",
        "shape": "(17,29)"
      }
    },
    "noiseEquivalentBackscatter": {
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(calibrationLength)",
        "data_type": "float64",
        "shape": "(17)"
      },
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(calibrationWidth)",
        "data_type": "float64",
        "shape": "(29)"
      },
      "HH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "HV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "VH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "VV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "RH": {
        "type": "supplemental",
```

```
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "RV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      }
    }
  },
  "frequencyB": {
    "elevationAntennaPattern": {
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(calibrationLength)",
        "data_type": "float64",
        "shape": "(17)"
      },
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(calibrationWidth)",
        "data_type": "float64",
        "shape": "(29)"
      },
      "HH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "cfloat32",
        "shape": "(17,29)"
      },
      "HV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "cfloat32",
        "shape": "(17,29)"
      },
      "VH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "cfloat32",
        "shape": "(17,29)"
      },
      "VV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "cfloat32",
        "shape": "(17,29)"
      },
      "RH": {
        "type": "supplemental",
```

```
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "RV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    }
  },
  "noiseEquivalentBackscatter": {
    "yCoordinates": {
      "type": "dimension",
      "dimension": "(calibrationLength)",
      "data_type": "float64",
      "shape": "(17)"
    },
    "xCoordinates": {
      "type": "dimension",
      "dimension": "(calibrationWidth)",
      "data_type": "float64",
      "shape": "(29)"
    },
    "HH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "float32",
      "shape": "(17,29)"
    },
    "HV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "float32",
      "shape": "(17,29)"
    },
    "VH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "float32",
      "shape": "(17,29)"
    },
    "VV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "float32",
      "shape": "(17,29)"
    },
    "RH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "float32",
```

```
            "shape": "(17,29)"
          },
          "RV": {
            "type": "supplemental",
            "dimension": "(calibrationLength,calibrationWidth)",
            "data_type": "float32",
            "shape": "(17,29)"
          }
        }
      },
      "crosstalk": {
        "yCoordinates": {
          "type": "dimension",
          "dimension": "(calibrationLength)",
          "data_type": "float64",
          "shape": "(17)"
        },
        "xCoordinates": {
          "type": "dimension",
          "dimension": "(calibrationWidth)",
          "data_type": "float64",
          "shape": "(29)"
        },
        "txHorizontalCrosspol": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "cfloat32",
          "shape": "(17,29)"
        },
        "txVerticalCrosspol": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "cfloat32",
          "shape": "(17,29)"
        },
        "rxHorizontalCrosspol": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "cfloat32",
          "shape": "(17,29)"
        },
        "rxVerticalCrosspol": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "cfloat32",
          "shape": "(17,29)"
        }
      }
    },
    "processingInformation": {
      "parameters": {
        "azimuthChirpWeighting": {
```

```
                  "type": "dimension",
                  "dimension": "(chirpFFTFrequency)",
                  "data_type": "float32",
                  "shape": "(256)"
                },
                "rangeChirpWeighting": {
                  "type": "dimension",
                  "dimension": "(chirpFFTFrequency)",
                  "data_type": "float32",
                  "shape": "(256)"
                },
                "referenceTerrainHeight": {
                  "type": "supplemental",
                  "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                  "data_type": "float32",
                  "shape": "(29,3)"
                },
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(dopplerCentroidLength)",
                  "data_type": "float64",
                  "shape": "(29)"
                },
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(dopplerCentroidWidth)",
                  "data_type": "float64",
                  "shape": "(3)"
                },
                "frequencyA": {
                  "yCoordinates": {
                    "type": "dimension",
                    "dimension": "(dopplerCentroidLength)",
                    "data_type": "float64",
                    "shape": "(29)"
                  },
                  "xCoordinates": {
                    "type": "dimension",
                    "dimension": "(dopplerCentroidWidth)",
                    "data_type": "float64",
                    "shape": "(3)"
                  },
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                    "data_type": "float64",
                    "shape": "(29,3)"
                  }
                },
                "frequencyB": {
```

```
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(dopplerCentroidLength)",
                  "data_type": "float64",
                  "shape": "(29)"
                },
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(dopplerCentroidWidth)",
                  "data_type": "float64",
                  "shape": "(3)"
                },
                "dopplerCentroid": {
                  "type": "supplemental",
                  "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                  "data_type": "float64",
                  "shape": "(29,3)"
                }
              }
            },
            "timingCorrections": {
              "frequencyA": {
                "azimuthIonosphere": {
                  "type": "supplemental",
                  "dimension":
"(timingCorrectionLength,timingCorrectionWidth)",
                  "data_type": "float64",
                  "shape": "(3,11)"
                },
                "slantRangeIonosphere": {
                  "type": "supplemental",
                  "dimension":
"(timingCorrectionLength,timingCorrectionWidth)",
                  "data_type": "float64",
                  "shape": "(3,11)"
                },
                "slantRangeSolidEarthTides": {
                  "type": "supplemental",
                  "dimension":
"(timingCorrectionLength,timingCorrectionWidth)",
                  "data_type": "float64",
                  "shape": "(3,11)"
                },
                "slantRange": {
                  "type": "dimension",
                  "dimension": "(timingCorrectionWidth)",
                  "data_type": "float64",
                  "shape": "(11)"
                },
                "zeroDopplerTime": {
                  "type": "dimension",
```

```
                  "dimension": "(timingCorrectionLength)",
                  "data_type": "float64",
                  "shape": "(3)"
                }
              },
              "frequencyB": {
                "azimuthIonosphere": {
                  "type": "supplemental",
                  "dimension":
"(timingCorrectionLength,timingCorrectionWidth)",
                  "data_type": "float64",
                  "shape": "(3,11)"
                },
                "slantRangeIonosphere": {
                  "type": "supplemental",
                  "dimension":
"(timingCorrectionLength,timingCorrectionWidth)",
                  "data_type": "float64",
                  "shape": "(3,11)"
                },
                "slantRangeSolidEarthTides": {
                  "type": "supplemental",
                  "dimension":
"(timingCorrectionLength,timingCorrectionWidth)",
                  "data_type": "float64",
                  "shape": "(3,11)"
                },
                "slantRange": {
                  "type": "dimension",
                  "dimension": "(timingCorrectionWidth)",
                  "data_type": "float64",
                  "shape": "(11)"
                },
                "zeroDopplerTime": {
                  "type": "dimension",
                  "dimension": "(timingCorrectionLength)",
                  "data_type": "float64",
                  "shape": "(3)"
                }
              }
            }
          },
          "orbit": {
            "time": {
              "type": "dimension",
              "dimension": "(orbitListLength)",
              "data_type": "float64",
              "shape": "(3)"
            },
            "position": {
              "type": "supplemental",
              "dimension": "(orbitListLength,tripletxyz)",
```

```
              "data_type": "float64",
              "shape": "(3,3)"
            },
            "velocity": {
              "type": "supplemental",
              "dimension": "(orbitListLength,tripletxyz)",
              "data_type": "float64",
              "shape": "(3,3)"
            }
          },
          "attitude": {
            "time": {
              "type": "dimension",
              "dimension": "(attitudeListLength)",
              "data_type": "float64",
              "shape": "(11)"
            },
            "quaternions": {
              "type": "supplemental",
              "dimension": "(attitudeListLength,quaternions)",
              "data_type": "float64",
              "shape": "(11,4)"
            },
            "eulerAngles": {
              "type": "supplemental",
              "dimension": "(attitudeListLength,tripletxyz)",
              "data_type": "float64",
              "shape": "(11,3)"
            }
          },
          "radarGrid": {
            "zeroDopplerAzimuthTime": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(17,5,23)"
            },
            "slantRange": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(17,5,23)"
            },
            "incidenceAngle": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(17,5,23)"
            },
```

```
            "losUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(17,5,23)"
            },
            "losUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(17,5,23)"
            },
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(17,5,23)"
            },
            "alongTrackUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(17,5,23)"
            },
            "elevationAngle": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(17,5,23)"
            },
            "xCoordinates": {
              "type": "cube dimension",
              "dimension": "(radarCubeWidth)",
              "data_type": "float64",
              "shape": "(23)"
            },
            "yCoordinates": {
              "type": "cube dimension",
              "dimension": "(radarCubeLength)",
              "data_type": "float64",
              "shape": "(5)"
            },
            "heightAboveEllipsoid": {
              "type": "cube dimension",
              "dimension": "(radarCubeHeight)",
              "data_type": "float64",
              "shape": "(17)"
```

```
            }
          }
        }
      }
    }
  }
}
```

10.4. L1 RIFG

Metadata


```
{
  "granule": "L1_WrappedInterferogram",
  "science": {
    "LSAR": {
      "identification": {
        "referenceAbsoluteOrbitNumber": "Absolute orbit number for the
reference RSLC",
        "secondaryAbsoluteOrbitNumber": "Absolute orbit number for the
secondary RSLC",
        "referenceIsJointObservation": "\"True\" if any portion of the
reference RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "secondaryIsJointObservation": "\"True\" if any portion of the
secondary RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "trackNumber": "Track number",
        "frameNumber": "Frame number",
        "missionId": "Mission identifier",
        "processingCenter": "Data processing center",
        "productType": "Product type",
        "granuleId": "Unique granule identification name",
        "productDoi": "Digital Object Identifier (DOI) for the product",
        "productVersion": "Product version which represents the
structure of the product and the science content governed by the
algorithm, input data, and processing parameters",
        "productSpecificationVersion": "Product specification version
which represents the schema of this product",
        "lookDirection": "Look direction, either \"Left\" or \"Right\"",
        "orbitPassDirection": "Orbit direction, either \"Ascending\" or
\"Descending\"",
        "referenceZeroDopplerStartTime": "Azimuth start time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referenceZeroDopplerEndTime": "Azimuth stop time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerStartTime": "Azimuth start time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerEndTime": "Azimuth stop time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referencePlannedDatatakeId": "List of planned datatakes
included in the reference RSLC",
        "secondaryPlannedDatatakeId": "List of planned datatakes
included in the secondary RSLC",
        "referencePlannedObservationId": "List of planned observations
included in the reference RSLC ",
        "secondaryPlannedObservationId": "List of planned observations
included in the secondary RSLC ",
        "referenceListOfObservationModes": "List of observation modes of
the L0B granules used to generate the reference RSLC (one mode per
L0B)",
```

```
        "secondaryListOfObservationModes": "List of observation modes of
the L0B granules used to generate the secondary RSLC (one mode per
L0B)",
        "isUrgentObservation": "Flag indicating if observation is
nominal (\"False\") or urgent (\"True\")",
        "listOfFrequencies": "List of frequency layers available in the
product",
        "diagnosticModeFlag": "Indicates if the radar operation mode is
a diagnostic mode (1-2) or DBFed science (0): 0, 1, or 2",
        "productLevel": "Product level. \"L0A\": Unprocessed instrument
data; \"L0B\": Reformatted, unprocessed instrument data; \"L1\":
Processed instrument data in radar coordinates system; and \"L2\":
Processed instrument data in geocoded coordinates system",
        "isGeocoded": "Flag to indicate if the product data is in the
radar geometry (\"False\") or in the map geometry (\"True\")",
        "boundingPolygon": "OGR compatible WKT representing the bounding
polygon of the image. Horizontal coordinates are WGS84 longitude
followed by latitude (both in degrees), and the vertical coordinate is
the height above the WGS84 ellipsoid in meters. The first point
corresponds to the start-time, near-range radar coordinate, and the
perimeter is traversed in counterclockwise order on the map. This means
the traversal order in radar coordinates differs for left-looking and
right-looking sensors. The polygon includes the four corners of the
radar grid, with equal numbers of points distributed evenly in radar
coordinates along each edge",
        "processingDateTime": "Processing date and time (in UTC) in the
format YYYY-mm-ddTHH:MM:SS",
        "radarBand": "Acquired frequency band, either \"L\" or \"S\"",
        "platformName": "Name of the platform used to collect the remote
sensing data provided in this product",
        "instrumentName": "Name of the instrument used to collect the
remote sensing data provided in this product",
        "processingType": "Processing pipeline used to generate this
granule. \"Nominal\": standard production system; \"Urgent\": time-
sensitive processing in response to urgent response events; \"Custom\":
user-initiated processing outside the nominal production system",
        "isDithered": "\"True\" if the pulse timing was varied
(dithered) during acquisition, \"False\" otherwise",
        "isMixedMode": "\"True\" if this product is generated from
reference and secondary RSLCs with different range bandwidths, \"False\"
otherwise",
        "isFullFrame": "\"True\" if the product fully covers a NISAR
frame, \"False\" if partial coverage",
        "compositeReleaseId": "Unique version identifier of the science
data production system"
      },
      "RIFG": {
        "swaths": {
          "frequencyA": {
            "listOfPolarizations": "List of processed polarization
layers with frequency A",
            "centerFrequency": "Center frequency of the processed image
```

```
in hertz",
            "interferogram": {
              "slantRangeSpacing": "Slant range spacing of grid. Same as
difference between consecutive samples in slantRange array",
              "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for raster layers. This is same as the spacing between
consecutive entries in the zeroDopplerTime array",
              "sceneCenterAlongTrackSpacing": "Nominal along-track
spacing in meters between consecutive lines near mid-swath of the
product images",
              "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid-swath of the
product images",
              "slantRange": "Slant range vector",
              "zeroDopplerTime": "Vector of zero Doppler azimuth times
measured relative to a UTC epoch",
              "digitalElevationModel": "Digital Elevation Model (DEM) in
radar coordinates. This dataset is generated by referencing the input
DEM elevations to the WGS84 ellipsoid and projecting them onto a slant
range/azimuth grid",
              "mask": "Mask indicating the subswaths of valid samples in
the reference RSLC and geometrically-coregistered secondary RSLC. Each
pixel value is a two-digit number: the least significant digit
represents the subswath number of that pixel in the secondary RSLC, and
the most significant digit represents the subswath number of that pixel
in the reference RSLC. A value of 0 in either digit indicates an invalid
sample in the corresponding RSLC",
              "HH": {
                "wrappedInterferogram": "Interferogram between HH
layers",
                "coherenceMagnitude": "Coherence magnitude between HH
layers"
              },
              "VV": {
                "wrappedInterferogram": "Interferogram between VV
layers",
                "coherenceMagnitude": "Coherence magnitude between VV
layers"
              }
            },
            "pixelOffsets": {
              "sceneCenterAlongTrackSpacing": "Nominal along-track
spacing in meters between consecutive lines near mid-swath of the
product images",
              "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid-swath of the
product images",
              "slantRangeSpacing": "Slant range spacing of the offset
grid",
              "zeroDopplerTimeSpacing": "Along-track spacing of the
offset grid",
              "mask": "Mask indicating the subswaths of valid samples in
```

```
the reference RSLC and geometrically-coregistered secondary RSLC. Each
pixel value is a two-digit number: the least significant digit
represents the subswath number of that pixel in the secondary RSLC, and
the most significant digit represents the subswath number of that pixel
in the reference RSLC. A value of 0 in either digit indicates an invalid
sample in the corresponding RSLC",
              "digitalElevationModel": "Digital Elevation Model (DEM) in
radar coordinates. This dataset is generated by referencing the input
DEM elevations to the WGS84 ellipsoid and projecting them onto a slant
range/azimuth grid",
              "HH": {
                "slantRangeOffset": "Slant range offset",
                "alongTrackOffset": "Along-track offset",
                "correlationSurfacePeak": "Normalized correlation
surface peak"
              },
              "VV": {
                "slantRangeOffset": "Slant range offset",
                "alongTrackOffset": "Along-track offset",
                "correlationSurfacePeak": "Normalized correlation
surface peak"
              },
              "slantRange": "Slant range vector",
              "zeroDopplerTime": "Vector of zero Doppler azimuth times
measured relative to a UTC epoch"
            }
          }
        },
        "metadata": {
          "processingInformation": {
            "parameters": {
              "runConfigurationContents": "Contents of the run
configuration file with parameters used for processing",
              "reference": {
                "isMixedMode": "\"True\" if reference RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the reference RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the reference RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "referenceTerrainHeight": "Reference terrain height as a
function of time for reference RSLC",
                "frequencyA": {
                  "slantRangeStart": "Slant range start distance for the
reference RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the reference RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the reference RSLC",
```

```
                "slantRangeSpacing": "Slant range spacing of reference
RSLC",
                "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for reference RSLC raster layers",
                "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the reference RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                "rangeBandwidth": "Processed slant range bandwidth for
reference RSLC",
                "azimuthBandwidth": "Processed azimuth bandwidth for
reference RSLC",
                "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
              }
            },
            "secondary": {
              "isMixedMode": "\"True\" if secondary RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
              "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the secondary RSLC",
              "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the reference RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
              "referenceTerrainHeight": "Reference terrain height as a
function of time for secondary RSLC",
              "frequencyA": {
                "slantRangeStart": "Slant range start distance for the
secondary RSLC",
                "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the secondary RSLC",
                "numberOfAzimuthLines": "Number of azimuth lines
within the secondary RSLC",
                "slantRangeSpacing": "Slant range spacing of secondary
RSLC",
                "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for secondary RSLC raster layers",
                "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the secondary RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                "rangeBandwidth": "Processed slant range bandwidth for
secondary RSLC",
                "azimuthBandwidth": "Processed azimuth bandwidth for
secondary RSLC",
                "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
              }
```

```
            },
            "common": {
              "frequencyA": {
                "dopplerCentroid": "2D LUT of common Doppler centroid
between reference and secondary RSLCs",
                "dopplerBandwidth": "Common Doppler bandwidth between
reference and secondary RSLCs"
              }
            },
            "interferogram": {
              "frequencyA": {
                "rangeBandwidth": "Processed slant range bandwidth for
frequency A interferometric layers",
                "azimuthBandwidth": "Processed azimuth bandwidth for
frequency A interferometric layers",
                "numberOfRangeLooks": "Number of looks applied in the
slant range direction to form the wrapped interferogram",
                "numberOfAzimuthLooks": "Number of looks applied in
the along-track direction to form the wrapped interferogram",
                "commonBandRangeFilterApplied": "Flag to indicate if
common band range filter has been applied",
                "commonBandAzimuthFilterApplied": "Flag to indicate if
common band azimuth filter has been applied",
                "ellipsoidalFlatteningApplied": "Flag to indicate if
the interferometric phase has been flattened with respect to a zero
height ellipsoid",
                "topographicFlatteningApplied": "Flag to indicate if
the interferometric phase has been flattened with respect to topographic
height using a DEM"
              }
            },
            "pixelOffsets": {
              "frequencyA": {
                "isOffsetsBlendingApplied": "Flag to indicate if pixel
offsets are the results of blending multi-resolution layers of pixel
offsets",
                "alongTrackWindowSize": "Along-track cross-correlation
window size in pixels",
                "slantRangeWindowSize": "Slant range cross-correlation
window size in pixels",
                "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                "slantRangeSearchWindowSize": "Slant range cross-
correlation search window size in pixels",
                "alongTrackSkipWindowSize": "Along-track cross-
correlation skip window size in pixels",
                "slantRangeSkipWindowSize": "Slant range cross-
correlation skip window size in pixels",
                "correlationSurfaceOversampling": "Oversampling factor
of the cross-correlation surface"
              }
            }
```

```
            },
            "algorithms": {
              "softwareVersion": "Software version used for processing",
              "coregistration": {
                "coregistrationMethod": "RSLC coregistration method",
                "geometryCoregistration": "Geometry coregistration
algorithm",
                "crossCorrelation": "Cross-correlation algorithm for
sub-pixel offsets computation",
                "resampling": "Secondary RSLC resampling algorithm",
                "crossCorrelationOutliers": "Outliers identification
algorithm",
                "crossCorrelationFilling": "Outliers data filling
algorithm for cross-correlation offsets",
                "crossCorrelationFilterKernel": "Filtering algorithm for
cross-correlation offsets"
              },
              "interferogramFormation": {
                "multilooking": "Multilooking algorithm",
                "wrappedInterferogramFiltering": "Algorithm used to
filter the wrapped interferogram prior to phase unwrapping",
                "flatteningMethod": "Algorithm used to flatten the
wrapped interferogram"
              }
            },
            "inputs": {
              "l1ReferenceSlcGranules": "List of input reference L1 RSLC
products used",
              "l1SecondarySlcGranules": "List of input secondary L1 RSLC
products used",
              "configFiles": "List of input config files used",
              "demSource": "Description of the input digital elevation
model (DEM)",
              "waterMaskSource": "Description of the input water mask",
              "orbitFiles": "List of input orbit files used"
            }
          },
          "orbit": {
            "temporalBaseline": "Time interval between reference and
secondary RSLCs",
            "reference": {
              "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
              "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
              "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
              "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
```

```
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
            },
            "secondary": {
              "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
              "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
              "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
              "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
            }
          },
          "attitude": {
            "reference": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            },
            "secondary": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            }
          },
          "geolocationGrid": {
            "epsg": "EPSG code corresponding to the coordinate system
used for representing the geolocation grid",
            "coordinateY": "Y coordinates in specified EPSG code",
            "coordinateX": "X coordinates in specified EPSG code",
            "incidenceAngle": "Incidence angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the target
height",
            "losUnitVectorX": "East component of the line-of-sight (LOS)
unit vector, defined from the target to the sensor, expressed in the
east-north-up (ENU) coordinate system with its origin at the target
location",
            "losUnitVectorY": "North component of the line-of-sight
```

```
(LOS) unit vector, defined from the target to the sensor, expressed in
the east-north-up (ENU) coordinate system with its origin at the target
location",
            "alongTrackUnitVectorX": "East component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "alongTrackUnitVectorY": "North component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "elevationAngle": "Elevation angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the sensor",
            "parallelBaseline": "Parallel component of the InSAR
baseline",
            "perpendicularBaseline": "Perpendicular component of the
InSAR baseline",
            "slantRange": "Slant range values corresponding to the
geolocation grid",
            "zeroDopplerTime": "Vector of zero Doppler azimuth times,
measured relative to a UTC epoch, corresponding to the geolocation
grid",
            "groundTrackVelocity": "Absolute value of the platform
velocity scaled at the target height",
            "heightAboveEllipsoid": "Height values above WGS84 Ellipsoid
corresponding to the location grid"
          }
        }
      }
    }
  }
}
```

Data

```
{
  "granule": "L1_WrappedInterferogram",
  "dimension": {
    "attitudeListLength": 5,
    "dopplerCentroidSlantRangeWidth": 17,
    "dopplerCentroidTimeLength": 13,
    "geolocationCubeHeight": 17,
    "geolocationCubeLength": 19,
    "geolocationCubeWidth": 5,
    "interferogramSlantRangeWidth": 3,
    "interferogramZeroDopplerTimeLength": 17,
    "offsetSlantRangeWidth": 11,
    "offsetZeroDopplerTimeLength": 13,
    "orbitListLength": 19,
    "quaternions": 4,
    "tripletxyz": 3,
    "twoLayersCubeHeight": 17
```

```
  },
  "science": {
    "LSAR": {
      "RIFG": {
        "swaths": {
          "frequencyA": {
            "interferogram": {
              "slantRange": {
                "type": "dimension",
                "dimension": "(interferogramSlantRangeWidth)",
                "data_type": "float64",
                "shape": "(3)"
              },
              "zeroDopplerTime": {
                "type": "dimension",
                "dimension": "(interferogramZeroDopplerTimeLength)",
                "data_type": "float64",
                "shape": "(17)"
              },
              "digitalElevationModel": {
                "type": "supplemental",
                "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(17,3)"
              },
              "mask": {
                "type": "supplemental",
                "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                "data_type": "ubyte",
                "shape": "(17,3)"
              },
              "HH": {
                "wrappedInterferogram": {
                  "type": "dataset",
                  "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(17,3)"
                },
                "coherenceMagnitude": {
                  "type": "supplemental",
                  "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,3)"
                }
              },
              "VV": {
                "wrappedInterferogram": {
                  "type": "dataset",
```

```
                  "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                  "data_type": "cfloat32",
                  "shape": "(17,3)"
                },
                "coherenceMagnitude": {
                  "type": "supplemental",
                  "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,3)"
                }
              }
            },
            "pixelOffsets": {
              "mask": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(13,11)"
              },
              "digitalElevationModel": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(13,11)"
              },
              "HH": {
                "slantRangeOffset": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(13,11)"
                },
                "alongTrackOffset": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(13,11)"
                },
                "correlationSurfacePeak": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(13,11)"
                }
              },
```

```
            "VV": {
              "slantRangeOffset": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(13,11)"
              },
              "alongTrackOffset": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(13,11)"
              },
              "correlationSurfacePeak": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(13,11)"
              }
            },
            "slantRange": {
              "type": "dimension",
              "dimension": "(offsetSlantRangeWidth)",
              "data_type": "float64",
              "shape": "(11)"
            },
            "zeroDopplerTime": {
              "type": "dimension",
              "dimension": "(offsetZeroDopplerTimeLength)",
              "data_type": "float64",
              "shape": "(13)"
            }
          }
        }
      },
      "metadata": {
        "processingInformation": {
          "parameters": {
            "reference": {
              "referenceTerrainHeight": {
                "type": "dimension",
                "dimension": "(dopplerCentroidTimeLength)",
                "data_type": "float32",
                "shape": "(13)"
              },
              "frequencyA": {
                "dopplerCentroid": {
                  "type": "supplemental",
                  "dimension":
```

```
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(13,17)"
                  }
                }
              },
              "secondary": {
                "referenceTerrainHeight": {
                  "type": "dimension",
                  "dimension": "(dopplerCentroidTimeLength)",
                  "data_type": "float32",
                  "shape": "(13)"
                },
                "frequencyA": {
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(13,17)"
                  }
                }
              },
              "common": {
                "frequencyA": {
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(13,17)"
                  }
                }
              }
            }
          },
          "orbit": {
            "reference": {
              "time": {
                "type": "dimension",
                "dimension": "(orbitListLength)",
                "data_type": "float64",
                "shape": "(19)"
              },
              "position": {
                "type": "supplemental",
                "dimension": "(orbitListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(19,3)"
              },
              "velocity": {
                "type": "supplemental",
```

```
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(19,3)"
        }
      },
      "secondary": {
        "time": {
          "type": "dimension",
          "dimension": "(orbitListLength)",
          "data_type": "float64",
          "shape": "(19)"
        },
        "position": {
          "type": "supplemental",
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(19,3)"
        },
        "velocity": {
          "type": "supplemental",
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(19,3)"
        }
      }
    },
    "attitude": {
      "reference": {
        "time": {
          "type": "dimension",
          "dimension": "(attitudeListLength)",
          "data_type": "float64",
          "shape": "(5)"
        },
        "quaternions": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,quaternions)",
          "data_type": "float64",
          "shape": "(5,4)"
        },
        "eulerAngles": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(5,3)"
        }
      },
      "secondary": {
        "time": {
          "type": "dimension",
          "dimension": "(attitudeListLength)",
          "data_type": "float64",
```

```
              "shape": "(5)"
            },
            "quaternions": {
              "type": "supplemental",
              "dimension": "(attitudeListLength,quaternions)",
              "data_type": "float64",
              "shape": "(5,4)"
            },
            "eulerAngles": {
              "type": "supplemental",
              "dimension": "(attitudeListLength,tripletxyz)",
              "data_type": "float64",
              "shape": "(5,3)"
            }
          }
        },
        "geolocationGrid": {
          "coordinateY": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float64",
            "shape": "(17,19,5)"
          },
          "coordinateX": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float64",
            "shape": "(17,19,5)"
          },
          "incidenceAngle": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float64",
            "shape": "(17,19,5)"
          },
          "losUnitVectorX": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float32",
            "shape": "(17,19,5)"
          },
          "losUnitVectorY": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float32",
            "shape": "(17,19,5)"
          },
```

```
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(17,19,5)"
            },
            "alongTrackUnitVectorY": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(17,19,5)"
            },
            "elevationAngle": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(17,19,5)"
            },
            "parallelBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(17,19,5)"
            },
            "perpendicularBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(17,19,5)"
            },
            "slantRange": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(5)"
            },
            "zeroDopplerTime": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeLength)",
              "data_type": "float64",
              "shape": "(19)"
            },
            "heightAboveEllipsoid": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeHeight)",
              "data_type": "float64",
              "shape": "(17)"
```

```
            }
          }
        }
      }
    }
  }
}
```

10.5. L1 RUNW

Metadata


```
{
  "granule": "L1_UnwrappedInterferogram",
  "science": {
    "LSAR": {
      "identification": {
        "referenceAbsoluteOrbitNumber": "Absolute orbit number for the
reference RSLC",
        "secondaryAbsoluteOrbitNumber": "Absolute orbit number for the
secondary RSLC",
        "referenceIsJointObservation": "\"True\" if any portion of the
reference RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "secondaryIsJointObservation": "\"True\" if any portion of the
secondary RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "trackNumber": "Track number",
        "frameNumber": "Frame number",
        "missionId": "Mission identifier",
        "processingCenter": "Data processing center",
        "productType": "Product type",
        "granuleId": "Unique granule identification name",
        "productDoi": "Digital Object Identifier (DOI) for the product",
        "productVersion": "Product version which represents the
structure of the product and the science content governed by the
algorithm, input data, and processing parameters",
        "productSpecificationVersion": "Product specification version
which represents the schema of this product",
        "lookDirection": "Look direction, either \"Left\" or \"Right\"",
        "orbitPassDirection": "Orbit direction, either \"Ascending\" or
\"Descending\"",
        "referenceZeroDopplerStartTime": "Azimuth start time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referenceZeroDopplerEndTime": "Azimuth stop time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerStartTime": "Azimuth start time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerEndTime": "Azimuth stop time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referencePlannedDatatakeId": "List of planned datatakes
included in the reference RSLC",
        "secondaryPlannedDatatakeId": "List of planned datatakes
included in the secondary RSLC",
        "referencePlannedObservationId": "List of planned observations
included in the reference RSLC ",
        "secondaryPlannedObservationId": "List of planned observations
included in the secondary RSLC ",
        "referenceListOfObservationModes": "List of observation modes of
the L0B granules used to generate the reference RSLC (one mode per
L0B)",
```

```
        "secondaryListOfObservationModes": "List of observation modes of
the L0B granules used to generate the secondary RSLC (one mode per
L0B)",
        "isUrgentObservation": "Flag indicating if observation is
nominal (\"False\") or urgent (\"True\")",
        "listOfFrequencies": "List of frequency layers available in the
product",
        "diagnosticModeFlag": "Indicates if the radar operation mode is
a diagnostic mode (1-2) or DBFed science (0): 0, 1, or 2",
        "productLevel": "Product level. \"L0A\": Unprocessed instrument
data; \"L0B\": Reformatted, unprocessed instrument data; \"L1\":
Processed instrument data in radar coordinates system; and \"L2\":
Processed instrument data in geocoded coordinates system",
        "isGeocoded": "Flag to indicate if the product data is in the
radar geometry (\"False\") or in the map geometry (\"True\")",
        "boundingPolygon": "OGR compatible WKT representing the bounding
polygon of the image. Horizontal coordinates are WGS84 longitude
followed by latitude (both in degrees), and the vertical coordinate is
the height above the WGS84 ellipsoid in meters. The first point
corresponds to the start-time, near-range radar coordinate, and the
perimeter is traversed in counterclockwise order on the map. This means
the traversal order in radar coordinates differs for left-looking and
right-looking sensors. The polygon includes the four corners of the
radar grid, with equal numbers of points distributed evenly in radar
coordinates along each edge",
        "processingDateTime": "Processing date and time (in UTC) in the
format YYYY-mm-ddTHH:MM:SS",
        "radarBand": "Acquired frequency band, either \"L\" or \"S\"",
        "platformName": "Name of the platform used to collect the remote
sensing data provided in this product",
        "instrumentName": "Name of the instrument used to collect the
remote sensing data provided in this product",
        "processingType": "Processing pipeline used to generate this
granule. \"Nominal\": standard production system; \"Urgent\": time-
sensitive processing in response to urgent response events; \"Custom\":
user-initiated processing outside the nominal production system",
        "isDithered": "\"True\" if the pulse timing was varied
(dithered) during acquisition, \"False\" otherwise",
        "isMixedMode": "\"True\" if this product is generated from
reference and secondary RSLCs with different range bandwidths, \"False\"
otherwise",
        "isFullFrame": "\"True\" if the product fully covers a NISAR
frame, \"False\" if partial coverage",
        "compositeReleaseId": "Unique version identifier of the science
data production system"
      },
      "RUNW": {
        "swaths": {
          "frequencyA": {
            "listOfPolarizations": "List of processed polarization
layers with frequency A",
            "centerFrequency": "Center frequency of the processed image
```

```
in hertz",
            "interferogram": {
              "sceneCenterAlongTrackSpacing": "Nominal along-track
spacing in meters between consecutive lines near mid-swath of the
product images",
              "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid-swath of the
product images",
              "slantRangeSpacing": "Slant range spacing of grid. Same as
difference between consecutive samples in slantRange array",
              "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for raster layers. This is same as the spacing between
consecutive entries in the zeroDopplerTime array",
              "slantRange": "Slant range vector",
              "zeroDopplerTime": "Vector of zero Doppler azimuth times
measured relative to a UTC epoch",
              "mask": "Mask indicating the subswaths of valid samples in
the reference RSLC and geometrically-coregistered secondary RSLC. Each
pixel value is a two-digit number: the least significant digit
represents the subswath number of that pixel in the secondary RSLC, and
the most significant digit represents the subswath number of that pixel
in the reference RSLC. A value of 0 in either digit indicates an invalid
sample in the corresponding RSLC",
              "digitalElevationModel": "Digital Elevation Model (DEM) in
radar coordinates. This dataset is generated by referencing the input
DEM elevations to the WGS84 ellipsoid and projecting them onto a slant
range/azimuth grid",
              "HH": {
                "unwrappedPhase": "Unwrapped interferogram between HH
layers",
                "coherenceMagnitude": "Coherence magnitude between HH
layers",
                "connectedComponents": "Connected components for HH
layer",
                "ionospherePhaseScreen": "Ionosphere phase screen",
                "ionospherePhaseScreenUncertainty": "Uncertainty of the
ionosphere phase screen"
              },
              "VV": {
                "unwrappedPhase": "Unwrapped interferogram between VV
layers",
                "coherenceMagnitude": "Coherence magnitude between VV
layers",
                "connectedComponents": "Connected components for VV
layer",
                "ionospherePhaseScreen": "Ionosphere phase screen",
                "ionospherePhaseScreenUncertainty": "Uncertainty of the
ionosphere phase screen"
              }
            },
            "pixelOffsets": {
              "sceneCenterAlongTrackSpacing": "Nominal along-track
```

```
spacing in meters between consecutive lines near mid-swath of the
product images",
              "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid-swath of the
product images",
              "slantRangeSpacing": "Slant range spacing of the offset
grid",
              "zeroDopplerTimeSpacing": "Along-track spacing of the
offset grid",
              "mask": "Mask indicating the subswaths of valid samples in
the reference RSLC and geometrically-coregistered secondary RSLC. Each
pixel value is a two-digit number: the least significant digit
represents the subswath number of that pixel in the secondary RSLC, and
the most significant digit represents the subswath number of that pixel
in the reference RSLC. A value of 0 in either digit indicates an invalid
sample in the corresponding RSLC",
              "digitalElevationModel": "Digital Elevation Model (DEM) in
radar coordinates. This dataset is generated by referencing the input
DEM elevations to the WGS84 ellipsoid and projecting them onto a slant
range/azimuth grid",
              "HH": {
                "slantRangeOffset": "Slant range offset",
                "alongTrackOffset": "Along-track offset",
                "correlationSurfacePeak": "Normalized correlation
surface peak"
              },
              "VV": {
                "slantRangeOffset": "Slant range offset",
                "alongTrackOffset": "Along-track offset",
                "correlationSurfacePeak": "Normalized correlation
surface peak"
              },
              "slantRange": "Slant range vector",
              "zeroDopplerTime": "Vector of zero Doppler azimuth times
measured relative to a UTC epoch"
            }
          }
        },
        "metadata": {
          "processingInformation": {
            "parameters": {
              "runConfigurationContents": "Contents of the run
configuration file with parameters used for processing",
              "reference": {
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the reference RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the reference RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "isMixedMode": "\"True\" if reference RSLC is a
composite of data collected in multiple radar modes, \"False\"
```

```
otherwise",
                "referenceTerrainHeight": "Reference terrain height as a
function of time for reference RSLC",
                "frequencyA": {
                  "slantRangeStart": "Slant range start distance for the
reference RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the reference RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the reference RSLC",
                  "slantRangeSpacing": "Slant range spacing of reference
RSLC",
                  "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the reference RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                  "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                  "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for reference RSLC raster layers",
                  "rangeBandwidth": "Processed slant range bandwidth for
reference RSLC",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
reference RSLC",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
                }
              },
              "secondary": {
                "referenceTerrainHeight": "Reference terrain height as a
function of time for secondary RSLC",
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the secondary RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the secondary RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "isMixedMode": "\"True\" if secondary RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
                "frequencyA": {
                  "slantRangeStart": "Slant range start distance for the
secondary RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the secondary RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the secondary RSLC",
                  "slantRangeSpacing": "Slant range spacing of secondary
RSLC",
                  "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the secondary RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                  "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
```

```
the secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                  "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for secondary RSLC raster layers",
                  "rangeBandwidth": "Processed slant range bandwidth for
secondary RSLC",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
secondary RSLC",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
                }
              },
              "common": {
                "frequencyA": {
                  "dopplerCentroid": "2D LUT of common Doppler centroid
between reference and secondary RSLCs",
                  "dopplerBandwidth": "Common Doppler bandwidth between
reference and secondary RSLCs"
                }
              },
              "interferogram": {
                "frequencyA": {
                  "rangeBandwidth": "Processed slant range bandwidth for
frequency A interferometric layers",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
frequency A interferometric layers",
                  "numberOfRangeLooks": "Number of looks applied in the
slant range direction to form the wrapped interferogram",
                  "numberOfAzimuthLooks": "Number of looks applied in
the along-track direction to form the wrapped interferogram",
                  "commonBandRangeFilterApplied": "Flag to indicate if
common band range filter has been applied",
                  "commonBandAzimuthFilterApplied": "Flag to indicate if
common band azimuth filter has been applied",
                  "ellipsoidalFlatteningApplied": "Flag to indicate if
the interferometric phase has been flattened with respect to a zero
height ellipsoid",
                  "topographicFlatteningApplied": "Flag to indicate if
the interferometric phase has been flattened with respect to topographic
height using a DEM"
                }
              },
              "ionosphere": {
                "lowBandBandwidth": "Slant range bandwidth of the low
sub-band image",
                "highBandBandwidth": "Slant range bandwidth of the high
sub-band image"
              },
              "pixelOffsets": {
                "frequencyA": {
                  "alongTrackWindowSize": "Along-track cross-correlation
window size in pixels",
                  "slantRangeWindowSize": "Slant range cross-correlation
```

```
window size in pixels",
                  "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                  "slantRangeSearchWindowSize": "Slant range cross-
correlation search window size in pixels",
                  "alongTrackSkipWindowSize": "Along-track cross-
correlation skip window size in pixels",
                  "slantRangeSkipWindowSize": "Slant range cross-
correlation skip window size in pixels",
                  "correlationSurfaceOversampling": "Oversampling factor
of the cross-correlation surface",
                  "isOffsetsBlendingApplied": "Flag to indicate if pixel
offsets are the results of blending multi-resolution layers of pixel
offsets"
                }
              }
            },
            "algorithms": {
              "softwareVersion": "Software version used for processing",
              "coregistration": {
                "coregistrationMethod": "RSLC coregistration method",
                "geometryCoregistration": "Geometry coregistration
algorithm",
                "crossCorrelation": "Cross-correlation algorithm for
sub-pixel offsets computation",
                "resampling": "Secondary RSLC resampling algorithm",
                "crossCorrelationOutliers": "Outliers identification
algorithm",
                "crossCorrelationFilling": "Outliers data filling
algorithm for cross-correlation offsets",
                "crossCorrelationFilterKernel": "Filtering algorithm for
cross-correlation offsets"
              },
              "interferogramFormation": {
                "multilooking": "Multilooking algorithm",
                "wrappedInterferogramFiltering": "Algorithm used to
filter the wrapped interferogram prior to phase unwrapping",
                "flatteningMethod": "Algorithm used to flatten the
wrapped interferogram"
              },
              "unwrapping": {
                "unwrappingAlgorithm": "Algorithm used for phase
unwrapping",
                "unwrappingInitializer": "Algorithm used to initialize
phase unwrapping",
                "costMode": "Cost mode algorithm for phase unwrapping",
                "preprocessing": {
                  "wrappedPhaseOutliers": "Algorithm identifying
outliers in the wrapped interferogram",
                  "wrappedPhaseFilling": "Outliers data filling
algorithm for phase unwrapping preprocessing"
                }
```

```
            },
            "ionosphereEstimation": {
              "ionosphereAlgorithm": "Algorithm used to estimate
ionosphere phase screen",
              "ionosphereOutliers": "Algorithm identifying outliers in
unfiltered ionosphere phase screen",
              "ionosphereFilling": "Outliers data filling algorithm
for ionosphere phase estimation",
              "ionosphereFiltering": "Filtering algorithm for
ionosphere phase screen computation",
              "unwrappingErrorCorrection": "Algorithm correcting
unwrapping errors in sub-band unwrapped interferograms"
            }
          },
          "inputs": {
            "l1ReferenceSlcGranules": "List of input reference L1 RSLC
products used",
            "l1SecondarySlcGranules": "List of input secondary L1 RSLC
products used",
            "configFiles": "List of input config files used",
            "demSource": "Description of the input digital elevation
model (DEM)",
            "waterMaskSource": "Description of the input water mask",
            "orbitFiles": "List of input orbit files used"
          }
        },
        "orbit": {
          "temporalBaseline": "Time interval between reference and
secondary RSLCs",
          "reference": {
            "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
            "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
            "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
            "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
          },
          "secondary": {
            "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
            "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
            "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
```

```
              "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
            }
          },
          "attitude": {
            "reference": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            },
            "secondary": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            }
          },
          "geolocationGrid": {
            "epsg": "EPSG code corresponding to the coordinate system
used for representing the geolocation grid",
            "coordinateY": "Y coordinates in specified EPSG code",
            "coordinateX": "X coordinates in specified EPSG code",
            "incidenceAngle": "Incidence angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the target
height",
            "losUnitVectorX": "East component of the line-of-sight (LOS)
unit vector, defined from the target to the sensor, expressed in the
east-north-up (ENU) coordinate system with its origin at the target
location",
            "losUnitVectorY": "North component of the line-of-sight
(LOS) unit vector, defined from the target to the sensor, expressed in
the east-north-up (ENU) coordinate system with its origin at the target
location",
            "alongTrackUnitVectorX": "East component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "alongTrackUnitVectorY": "North component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
```

```
            "elevationAngle": "Elevation angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the sensor",
            "parallelBaseline": "Parallel component of the InSAR
baseline",
            "perpendicularBaseline": "Perpendicular component of the
InSAR baseline",
            "slantRange": "Slant range values corresponding to the
geolocation grid",
            "zeroDopplerTime": "Vector of zero Doppler azimuth times,
measured relative to a UTC epoch, corresponding to the geolocation
grid",
            "groundTrackVelocity": "Absolute value of the platform
velocity scaled at the target height",
            "heightAboveEllipsoid": "Height values above WGS84 Ellipsoid
corresponding to the location grid"
          }
        }
      }
    }
  }
}
```

Data

```
{
  "granule": "L1_UnwrappedInterferogram",
  "dimension": {
    "attitudeListLength": 19,
    "dopplerCentroidSlantRangeWidth": 11,
    "dopplerCentroidTimeLength": 3,
    "geolocationCubeHeight": 19,
    "geolocationCubeLength": 29,
    "geolocationCubeWidth": 17,
    "interferogramSlantRangeWidth": 13,
    "interferogramZeroDopplerTimeLength": 17,
    "offsetSlantRangeWidth": 3,
    "offsetZeroDopplerTimeLength": 29,
    "orbitListLength": 17,
    "quaternions": 4,
    "tripletxyz": 3,
    "twoLayersCubeHeight": 11
  },
  "science": {
    "LSAR": {
      "RUNW": {
        "swaths": {
          "frequencyA": {
            "interferogram": {
              "slantRange": {
                "type": "dimension",
                "dimension": "(interferogramSlantRangeWidth)",
                "data_type": "float64",
```

```
            "shape": "(13)"
          },
          "zeroDopplerTime": {
            "type": "dimension",
            "dimension": "(interferogramZeroDopplerTimeLength)",
            "data_type": "float64",
            "shape": "(17)"
          },
          "mask": {
            "type": "supplemental",
            "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
            "data_type": "float32",
            "shape": "(17,13)"
          },
          "digitalElevationModel": {
            "type": "supplemental",
            "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
            "data_type": "float32",
            "shape": "(17,13)"
          },
          "HH": {
            "unwrappedPhase": {
              "type": "dataset",
              "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
              "data_type": "float32",
              "shape": "(17,13)"
            },
            "coherenceMagnitude": {
              "type": "supplemental",
              "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
              "data_type": "float32",
              "shape": "(17,13)"
            },
            "connectedComponents": {
              "type": "supplemental",
              "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
              "data_type": "uint16",
              "shape": "(17,13)"
            },
            "ionospherePhaseScreen": {
              "type": "supplemental",
              "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
              "data_type": "float32",
              "shape": "(17,13)"
            },
            "ionospherePhaseScreenUncertainty": {
```

```
                "type": "supplemental",
                "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(17,13)"
              }
            },
            "VV": {
              "unwrappedPhase": {
                "type": "dataset",
                "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(17,13)"
              },
              "coherenceMagnitude": {
                "type": "supplemental",
                "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(17,13)"
              },
              "connectedComponents": {
                "type": "supplemental",
                "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                "data_type": "uint16",
                "shape": "(17,13)"
              },
              "ionospherePhaseScreen": {
                "type": "supplemental",
                "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(17,13)"
              },
              "ionospherePhaseScreenUncertainty": {
                "type": "supplemental",
                "dimension":
"(interferogramZeroDopplerTimeLength,interferogramSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(17,13)"
              }
            }
          },
          "pixelOffsets": {
            "mask": {
              "type": "supplemental",
              "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
              "data_type": "float32",
              "shape": "(29,3)"
```

```
            },
            "digitalElevationModel": {
              "type": "supplemental",
              "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
              "data_type": "float32",
              "shape": "(29,3)"
            },
            "HH": {
              "slantRangeOffset": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,3)"
              },
              "alongTrackOffset": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,3)"
              },
              "correlationSurfacePeak": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,3)"
              }
            },
            "VV": {
              "slantRangeOffset": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,3)"
              },
              "alongTrackOffset": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,3)"
              },
              "correlationSurfacePeak": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(29,3)"
```

```
            }
          },
          "slantRange": {
            "type": "dimension",
            "dimension": "(offsetSlantRangeWidth)",
            "data_type": "float64",
            "shape": "(3)"
          },
          "zeroDopplerTime": {
            "type": "dimension",
            "dimension": "(offsetZeroDopplerTimeLength)",
            "data_type": "float64",
            "shape": "(29)"
          }
        }
      }
    },
    "metadata": {
      "processingInformation": {
        "parameters": {
          "reference": {
            "referenceTerrainHeight": {
              "type": "dimension",
              "dimension": "(dopplerCentroidTimeLength)",
              "data_type": "float32",
              "shape": "(3)"
            },
            "frequencyA": {
              "dopplerCentroid": {
                "type": "supplemental",
                "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                "data_type": "float64",
                "shape": "(3,11)"
              }
            }
          },
          "secondary": {
            "referenceTerrainHeight": {
              "type": "dimension",
              "dimension": "(dopplerCentroidTimeLength)",
              "data_type": "float32",
              "shape": "(3)"
            },
            "frequencyA": {
              "dopplerCentroid": {
                "type": "supplemental",
                "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                "data_type": "float64",
                "shape": "(3,11)"
              }
```

```
                }
              },
              "common": {
                "frequencyA": {
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(3,11)"
                  }
                }
              }
            }
          },
          "orbit": {
            "reference": {
              "time": {
                "type": "dimension",
                "dimension": "(orbitListLength)",
                "data_type": "float64",
                "shape": "(17)"
              },
              "position": {
                "type": "supplemental",
                "dimension": "(orbitListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(17,3)"
              },
              "velocity": {
                "type": "supplemental",
                "dimension": "(orbitListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(17,3)"
              }
            },
            "secondary": {
              "time": {
                "type": "dimension",
                "dimension": "(orbitListLength)",
                "data_type": "float64",
                "shape": "(17)"
              },
              "position": {
                "type": "supplemental",
                "dimension": "(orbitListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(17,3)"
              },
              "velocity": {
                "type": "supplemental",
                "dimension": "(orbitListLength,tripletxyz)",
```

```
            "data_type": "float64",
            "shape": "(17,3)"
          }
        }
      },
      "attitude": {
        "reference": {
          "time": {
            "type": "dimension",
            "dimension": "(attitudeListLength)",
            "data_type": "float64",
            "shape": "(19)"
          },
          "quaternions": {
            "type": "supplemental",
            "dimension": "(attitudeListLength,quaternions)",
            "data_type": "float64",
            "shape": "(19,4)"
          },
          "eulerAngles": {
            "type": "supplemental",
            "dimension": "(attitudeListLength,tripletxyz)",
            "data_type": "float64",
            "shape": "(19,3)"
          }
        },
        "secondary": {
          "time": {
            "type": "dimension",
            "dimension": "(attitudeListLength)",
            "data_type": "float64",
            "shape": "(19)"
          },
          "quaternions": {
            "type": "supplemental",
            "dimension": "(attitudeListLength,quaternions)",
            "data_type": "float64",
            "shape": "(19,4)"
          },
          "eulerAngles": {
            "type": "supplemental",
            "dimension": "(attitudeListLength,tripletxyz)",
            "data_type": "float64",
            "shape": "(19,3)"
          }
        }
      },
      "geolocationGrid": {
        "coordinateY": {
          "type": "cube",
          "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
```

```
            "data_type": "float64",
            "shape": "(19,29,17)"
          },
          "coordinateX": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float64",
            "shape": "(19,29,17)"
          },
          "incidenceAngle": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float32",
            "shape": "(19,29,17)"
          },
          "losUnitVectorX": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float32",
            "shape": "(19,29,17)"
          },
          "losUnitVectorY": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float32",
            "shape": "(19,29,17)"
          },
          "alongTrackUnitVectorX": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float32",
            "shape": "(19,29,17)"
          },
          "alongTrackUnitVectorY": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float32",
            "shape": "(19,29,17)"
          },
          "elevationAngle": {
            "type": "cube",
            "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
            "data_type": "float32",
            "shape": "(19,29,17)"
          },
```

```
            "parallelBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(11,29,17)"
            },
            "perpendicularBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(11,29,17)"
            },
            "slantRange": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(17)"
            },
            "zeroDopplerTime": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeLength)",
              "data_type": "float64",
              "shape": "(29)"
            },
            "heightAboveEllipsoid": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeHeight)",
              "data_type": "float64",
              "shape": "(19)"
            }
          }
        }
      }
    }
  }
}
```

10.6. L2 GUNW

Metadata


```
{
  "granule": "L2_GeocodedUnwrapped",
  "science": {
    "LSAR": {
      "identification": {
        "referenceAbsoluteOrbitNumber": "Absolute orbit number for the
reference RSLC",
        "secondaryAbsoluteOrbitNumber": "Absolute orbit number for the
secondary RSLC",
        "referenceIsJointObservation": "\"True\" if any portion of the
reference RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "secondaryIsJointObservation": "\"True\" if any portion of the
secondary RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "trackNumber": "Track number",
        "frameNumber": "Frame number",
        "missionId": "Mission identifier",
        "processingCenter": "Data processing center",
        "productType": "Product type",
        "granuleId": "Unique granule identification name",
        "productDoi": "Digital Object Identifier (DOI) for the product",
        "productVersion": "Product version which represents the
structure of the product and the science content governed by the
algorithm, input data, and processing parameters",
        "productSpecificationVersion": "Product specification version
which represents the schema of this product",
        "lookDirection": "Look direction, either \"Left\" or \"Right\"",
        "orbitPassDirection": "Orbit direction, either \"Ascending\" or
\"Descending\"",
        "referenceZeroDopplerStartTime": "Azimuth start time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referenceZeroDopplerEndTime": "Azimuth stop time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerStartTime": "Azimuth start time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerEndTime": "Azimuth stop time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referencePlannedDatatakeId": "List of planned datatakes
included in the reference RSLC",
        "secondaryPlannedDatatakeId": "List of planned datatakes
included in the secondary RSLC",
        "referencePlannedObservationId": "List of planned observations
included in the reference RSLC ",
        "secondaryPlannedObservationId": "List of planned observations
included in the secondary RSLC ",
        "referenceListOfObservationModes": "List of observation modes of
the L0B granules used to generate the reference RSLC (one mode per
L0B)",
```

```
        "secondaryListOfObservationModes": "List of observation modes of
the L0B granules used to generate the secondary RSLC (one mode per
L0B)",
        "isUrgentObservation": "Flag indicating if observation is
nominal (\"False\") or urgent (\"True\")",
        "listOfFrequencies": "List of frequency layers available in the
product",
        "diagnosticModeFlag": "Indicates if the radar operation mode is
a diagnostic mode (1-2) or DBFed science (0): 0, 1, or 2",
        "productLevel": "Product level. \"L0A\": Unprocessed instrument
data; \"L0B\": Reformatted, unprocessed instrument data; \"L1\":
Processed instrument data in radar coordinates system; and \"L2\":
Processed instrument data in geocoded coordinates system",
        "isGeocoded": "Flag to indicate if the product data is in the
radar geometry (\"False\") or in the map geometry (\"True\")",
        "boundingPolygon": "OGR compatible WKT representing the bounding
polygon of the image. Horizontal coordinates are WGS84 longitude
followed by latitude (both in degrees), and the vertical coordinate is
the height above the WGS84 ellipsoid in meters. The first point
corresponds to the start-time, near-range radar coordinate, and the
perimeter is traversed in counterclockwise order on the map. This means
the traversal order in radar coordinates differs for left-looking and
right-looking sensors. The polygon includes the four corners of the
radar grid, with equal numbers of points distributed evenly in radar
coordinates along each edge",
        "processingDateTime": "Processing date and time (in UTC) in the
format YYYY-mm-ddTHH:MM:SS",
        "radarBand": "Acquired frequency band, either \"L\" or \"S\"",
        "platformName": "Name of the platform used to collect the remote
sensing data provided in this product",
        "instrumentName": "Name of the instrument used to collect the
remote sensing data provided in this product",
        "processingType": "Processing pipeline used to generate this
granule. \"Nominal\": standard production system; \"Urgent\": time-
sensitive processing in response to urgent response events; \"Custom\":
user-initiated processing outside the nominal production system",
        "isDithered": "\"True\" if the pulse timing was varied
(dithered) during acquisition, \"False\" otherwise",
        "isMixedMode": "\"True\" if this product is generated from
reference and secondary RSLCs with different range bandwidths, \"False\"
otherwise",
        "compositeReleaseId": "Unique version identifier of the science
data production system",
        "isFullFrame": "\"True\" if the product fully covers a NISAR
frame, \"False\" if partial coverage"
      },
      "GUNW": {
        "identification": {
          "staticLayersDataAccess": "Location of the static layers
product associated with this product (URL or DOI)"
        },
        "grids": {
```

```
          "frequencyA": {
            "listOfPolarizations": "List of processed polarization
layers with frequency A",
            "centerFrequency": "Center frequency of the processed image
in hertz",
            "pixelOffsets": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
              "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
              "xCoordinates": "X coordinates in specified projection",
              "yCoordinates": "Y coordinates in specified projection",
              "HH": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "slantRangeOffset": "Slant range offset",
                "alongTrackOffset": "Along-track offset",
                "correlationSurfacePeak": "Normalized cross-correlation
surface peak",
                "xCoordinates": "X coordinates in specified projection",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
                "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines"
              },
              "VV": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "slantRangeOffset": "Slant range offset",
                "alongTrackOffset": "Along-track offset",
                "correlationSurfacePeak": "Normalized correlation
surface peak",
                "xCoordinates": "X coordinates in specified projection",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
                "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines"
              }
            },
            "wrappedInterferogram": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
              "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
              "xCoordinates": "X coordinates in specified projection",
              "yCoordinates": "Y coordinates in specified projection",
```

```
            "HH": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
              "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
              "xCoordinates": "X coordinates in specified projection",
              "yCoordinates": "Y coordinates in specified projection",
              "wrappedInterferogram": "Complex wrapped interferogram
between HH layers",
              "coherenceMagnitude": "Coherence magnitude between HH
layers"
            },
            "VV": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
              "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
              "xCoordinates": "X coordinates in specified projection",
              "yCoordinates": "Y coordinates in specified projection",
              "wrappedInterferogram": "Complex wrapped interferogram
between VV layers",
              "coherenceMagnitude": "Coherence magnitude between VV
layers"
            },
            "mask": "Combination of water mask and a mask of subswaths
of valid samples in the reference RSLC and geometrically-coregistered
secondary RSLC. Each pixel value is a three-digit number: the most
significant digit represents the water flag of that pixel the reference
RSLC, where 1 is water and 0 is non-water; the second digit represents
the subswath number of that pixel in the reference RSLC; the least
significant digit represents the subswath number of that pixel in the
secondary RSLC. A value of 0 in either subswath digit indicates an
invalid sample in the corresponding RSLC"
          },
          "unwrappedInterferogram": {
            "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
            "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
            "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
            "xCoordinates": "X coordinates in specified projection",
            "yCoordinates": "Y coordinates in specified projection",
            "mask": "Combination of water mask and a mask of subswaths
of valid samples in the reference RSLC and geometrically-coregistered
secondary RSLC. Each pixel value is a three-digit number: the most
significant digit represents the water flag of that pixel the reference
RSLC, where 1 is water and 0 is non-water; the second digit represents
```

```
the subswath number of that pixel in the reference RSLC; the least
significant digit represents the subswath number of that pixel in the
secondary RSLC. A value of 0 in either subswath digit indicates an
invalid sample in the corresponding RSLC",
              "HH": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
                "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
                "xCoordinates": "X coordinates in specified projection",
                "yCoordinates": "Y coordinates in specified projection",
                "unwrappedPhase": "Unwrapped interferogram between HH
layers",
                "connectedComponents": "Connected components for HH
layer",
                "coherenceMagnitude": "Coherence magnitude between HH
layers",
                "ionospherePhaseScreen": "Ionosphere phase screen",
                "ionospherePhaseScreenUncertainty": "Uncertainty of the
ionosphere phase screen"
              },
              "VV": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
                "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
                "xCoordinates": "X coordinates in specified projection",
                "yCoordinates": "Y coordinates in specified projection",
                "unwrappedPhase": "Unwrapped interferogram between VV
layers",
                "connectedComponents": "Connected components for VV
layer",
                "coherenceMagnitude": "Coherence magnitude between VV
layers",
                "ionospherePhaseScreen": "Ionosphere phase screen",
                "ionospherePhaseScreenUncertainty": "Uncertainty of the
ionosphere phase screen"
              }
            },
            "pixelOffset": {
              "mask": "Combination of water mask and a mask of subswaths
of valid samples in the reference RSLC and geometrically-coregistered
secondary RSLC. Each pixel value is a three-digit number: the most
significant digit represents the water flag of that pixel the reference
RSLC, where 1 is water and 0 is non-water; the second digit represents
the subswath number of that pixel in the reference RSLC; the least
significant digit represents the subswath number of that pixel in the
secondary RSLC. A value of 0 in either subswath digit indicates an
```

```
invalid sample in the corresponding RSLC"
            }
          }
        },
        "metadata": {
          "processingInformation": {
            "parameters": {
              "runConfigurationContents": "Contents of the run
configuration file with parameters used for processing",
              "reference": {
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the reference RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the reference RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "isMixedMode": "\"True\" if reference RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
                "referenceTerrainHeight": "Reference terrain height as a
function of map coordinates for reference RSLC",
                "frequencyA": {
                  "slantRangeStart": "Slant range start distance for the
reference RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the reference RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the reference RSLC",
                  "slantRangeSpacing": "Slant range spacing of reference
RSLC",
                  "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for reference RSLC raster layers",
                  "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the reference RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                  "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                  "rangeBandwidth": "Processed slant range bandwidth for
reference RSLC",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
reference RSLC",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
                }
              },
              "secondary": {
                "referenceTerrainHeight": "Reference terrain height as a
function of map coordinates for secondary RSLC",
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the secondary RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
```

```
interference (RFI) mitigation in the secondary RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "isMixedMode": "\"True\" if secondary RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
                "frequencyA": {
                  "slantRangeStart": "Slant range start distance for the
secondary RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the secondary RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the secondary RSLC",
                  "slantRangeSpacing": "Slant range spacing of secondary
RSLC",
                  "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for secondary RSLC raster layers",
                  "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the secondary RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                  "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                  "rangeBandwidth": "Processed slant range bandwidth for
secondary RSLC",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
secondary RSLC",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
                }
              },
              "common": {
                "frequencyA": {
                  "dopplerCentroid": "2D LUT of common Doppler centroid
between reference and secondary RSLCs",
                  "dopplerBandwidth": "Common Doppler bandwidth between
reference and secondary RSLCs"
                }
              },
              "wrappedInterferogram": {
                "frequencyA": {
                  "rangeBandwidth": "Processed slant range bandwidth for
frequency A interferometric layers",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
frequency A interferometric layers",
                  "commonBandRangeFilterApplied": "Flag to indicate if
common band range filter has been applied",
                  "commonBandAzimuthFilterApplied": "Flag to indicate if
common band azimuth filter has been applied",
                  "numberOfRangeLooks": "Number of looks applied in the
slant range direction to form the wrapped interferogram",
                  "numberOfAzimuthLooks": "Number of looks applied in
the along-track direction to form the wrapped interferogram",
                  "ellipsoidalFlatteningApplied": "Flag to indicate if
```

```
the interferometric phase has been flattened with respect to a zero
height ellipsoid",
                  "topographicFlatteningApplied": "Flag to indicate if
the interferometric phase has been flattened with respect to topographic
height using a DEM"
                }
              },
              "unwrappedInterferogram": {
                "frequencyA": {
                  "rangeBandwidth": "Processed slant range bandwidth for
frequency A interferometric layers",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
frequency A interferometric layers",
                  "commonBandRangeFilterApplied": "Flag to indicate if
common band range filter has been applied",
                  "commonBandAzimuthFilterApplied": "Flag to indicate if
common band azimuth filter has been applied",
                  "numberOfRangeLooks": "Number of looks applied in the
slant range direction to form the unwrapped interferogram",
                  "numberOfAzimuthLooks": "Number of looks applied in
the along-track direction to form the unwrapped interferogram",
                  "ellipsoidalFlatteningApplied": "Flag to indicate if
the interferometric phase has been flattened with respect to a zero
height ellipsoid",
                  "topographicFlatteningApplied": "Flag to indicate if
the interferometric phase has been flattened with respect to topographic
height using a DEM"
                }
              },
              "ionosphere": {
                "lowBandBandwidth": "Slant range bandwidth of the low
sub-band image",
                "highBandBandwidth": "Slant range bandwidth of the high
sub-band image"
              },
              "geocoding": {
                "rangeIonosphericCorrectionApplied": "Flag to indicate
if the range ionospheric correction is applied to improve geolocation",
                "azimuthIonosphericCorrectionApplied": "Flag to indicate
if the azimuth ionospheric correction is applied to improve
geolocation",
                "hydrostaticTroposphericCorrectionApplied": "Flag to
indicate if the hydrostatic tropospheric correction is applied to
improve geolocation",
                "wetTroposphericCorrectionApplied": "Flag to indicate if
the wet tropospheric correction is applied to improve geolocation"
              },
              "pixelOffsets": {
                "frequencyA": {
                  "alongTrackWindowSize": "Along-track cross-correlation
window size in pixels",
                  "slantRangeWindowSize": "Slant range cross-correlation
```

```
window size in pixels",
                  "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                  "slantRangeSearchWindowSize": "Slant range cross-
correlation search window size in pixels",
                  "alongTrackSkipWindowSize": "Along-track cross-
correlation skip window size in pixels",
                  "slantRangeSkipWindowSize": "Slant range cross-
correlation skip window size in pixels",
                  "correlationSurfaceOversampling": "Oversampling factor
of the cross-correlation surface",
                  "isOffsetsBlendingApplied": "Flag to indicate if pixel
offsets are the results of blending multi-resolution layers of pixel
offsets"
                }
              }
            },
            "algorithms": {
              "softwareVersion": "Software version used for processing",
              "coregistration": {
                "coregistrationMethod": "RSLC coregistration method",
                "geometryCoregistration": "Geometry coregistration
algorithm",
                "crossCorrelation": "Cross-correlation algorithm for
sub-pixel offsets computation",
                "resampling": "Secondary RSLC resampling algorithm",
                "crossCorrelationOutliers": "Outliers identification
algorithm",
                "crossCorrelationFilling": "Outliers data filling
algorithm for cross-correlation offsets",
                "crossCorrelationFilterKernel": "Filtering algorithm for
cross-correlation offsets"
              },
              "interferogramFormation": {
                "multilooking": "Multilooking algorithm",
                "wrappedInterferogramFiltering": "Algorithm used to
filter the wrapped interferogram prior to phase unwrapping",
                "flatteningMethod": "Algorithm used to flatten the
wrapped interferogram"
              },
              "unwrapping": {
                "unwrappingAlgorithm": "Algorithm used for phase
unwrapping",
                "unwrappingInitializer": "Algorithm used to initialize
phase unwrapping",
                "costMode": "Cost mode algorithm for phase unwrapping",
                "preprocessing": {
                  "wrappedPhaseOutliers": "Algorithm identifying
outliers in the wrapped interferogram",
                  "wrappedPhaseFilling": "Outliers data filling
algorithm for phase unwrapping preprocessing"
                }
```

```
            },
            "ionosphereEstimation": {
              "ionosphereAlgorithm": "Algorithm used to estimate
ionosphere phase screen",
              "ionosphereOutliers": "Algorithm identifying outliers in
unfiltered ionosphere phase screen",
              "ionosphereFilling": "Outliers data filling algorithm
for ionosphere phase estimation",
              "ionosphereFiltering": "Filtering algorithm for
ionosphere phase screen computation",
              "unwrappingErrorCorrection": "Algorithm correcting
unwrapping errors in sub-band unwrapped interferograms"
            },
            "geocoding": {
              "demInterpolation": "DEM interpolation algorithm",
              "floatingGeocodingInterpolation": "Geocoding
interpolation algorithm for floating point datasets",
              "integerGeocodingInterpolation": "Geocoding
interpolation algorithm for integer datasets",
              "complexGeocodingInterpolation": "Geocoding
interpolation algorithm for complex-valued datasets"
            }
          },
          "inputs": {
            "l1ReferenceSlcGranules": "List of input reference L1 RSLC
products used",
            "l1SecondarySlcGranules": "List of input secondary L1 RSLC
products used",
            "orbitFiles": "List of input orbit files used",
            "configFiles": "List of input config files used",
            "demSource": "Description of the input digital elevation
model (DEM)",
            "waterMaskSource": "Description of the input water mask"
          }
        },
        "orbit": {
          "temporalBaseline": "Time interval between reference and
secondary RSLCs",
          "reference": {
            "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
            "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
            "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
            "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
          },
```

```
            "secondary": {
              "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
              "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
              "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
              "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
            }
          },
          "attitude": {
            "reference": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            },
            "secondary": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            }
          },
          "radarGrid": {
            "referenceSlantRange": "Slant range of the reference RSLC in
meters",
            "hydrostaticTroposphericPhaseScreen": "Hydrostatic component
of the troposphere phase screen",
            "wetTroposphericPhaseScreen": "Wet component of the
troposphere phase screen",
            "slantRangeSolidEarthTidesPhase": "Solid Earth tides phase
along slant range direction",
            "referenceZeroDopplerAzimuthTime": "Zero Doppler azimuth
time in seconds since UTC epoch of the reference RSLC image",
            "incidenceAngle": "Incidence angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the target
height",
            "losUnitVectorX": "East component of the line-of-sight (LOS)
```

```
unit vector, defined from the target to the sensor, expressed in the
east-north-up (ENU) coordinate system with its origin at the target
location",
            "losUnitVectorY": "North component of the line-of-sight
(LOS) unit vector, defined from the target to the sensor, expressed in
the east-north-up (ENU) coordinate system with its origin at the target
location",
            "alongTrackUnitVectorX": "East component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "alongTrackUnitVectorY": "North component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "elevationAngle": "Elevation angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the sensor",
            "groundTrackVelocity": "Absolute value of the platform
velocity scaled at the target height",
            "secondaryZeroDopplerAzimuthTime": "Zero Doppler azimuth
time in seconds since UTC epoch of the secondary RSLC image",
            "secondarySlantRange": "Slant range of the secondary RSLC in
meters",
            "parallelBaseline": "Parallel component of the InSAR
baseline",
            "perpendicularBaseline": "Perpendicular component of the
InSAR baseline",
            "xCoordinates": "X coordinates in specified projection",
            "yCoordinates": "Y coordinates in specified projection",
            "heightAboveEllipsoid": "Height values above WGS84 Ellipsoid
corresponding to the radar grid",
            "projection": "Product map grid projection: EPSG code, with
additional projection information as HDF5 Attributes"
          }
        }
      }
    }
  }
}
```


Data


```
{
  "granule": "L2_GeocodedUnwrapped",
  "dimension": {
    "attitudeListLength": 11,
    "dopplerCentroidLength": 19,
    "dopplerCentroidWidth": 23,
    "offsetLength": 19,
    "offsetWidth": 17,
    "orbitListLength": 3,
    "quaternions": 4,
```

```
    "radarCubeHeight": 5,
    "radarCubeLength": 11,
    "radarCubeWidth": 11,
    "tripletxyz": 3,
    "twoLayersCubeHeight": 23,
    "unwrappedInterferogramLength": 19,
    "unwrappedInterferogramWidth": 23,
    "wrappedInterferogramLength": 19,
    "wrappedInterferogramWidth": 17
  },
  "science": {
    "LSAR": {
      "GUNW": {
        "grids": {
          "frequencyA": {
            "pixelOffsets": {
              "xCoordinates": {
                "type": "dimension",
                "dimension": "(unwrappedInterferogramWidth)",
                "data_type": "float64",
                "shape": "(23)"
              },
              "yCoordinates": {
                "type": "dimension",
                "dimension": "(unwrappedInterferogramLength)",
                "data_type": "float64",
                "shape": "(19)"
              },
              "HH": {
                "slantRangeOffset": {
                  "type": "supplemental",
                  "dimension": "(offsetLength,offsetWidth)",
                  "data_type": "float32",
                  "shape": "(19,17)"
                },
                "alongTrackOffset": {
                  "type": "supplemental",
                  "dimension": "(offsetLength,offsetWidth)",
                  "data_type": "float32",
                  "shape": "(19,17)"
                },
                "correlationSurfacePeak": {
                  "type": "supplemental",
                  "dimension": "(offsetLength,offsetWidth)",
                  "data_type": "float32",
                  "shape": "(19,17)"
                },
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(offsetWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
```

```
      },
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(offsetLength)",
        "data_type": "float64",
        "shape": "(19)"
      }
    },
    "VV": {
      "slantRangeOffset": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "alongTrackOffset": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "correlationSurfacePeak": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(offsetWidth)",
        "data_type": "float64",
        "shape": "(17)"
      },
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(offsetLength)",
        "data_type": "float64",
        "shape": "(19)"
      }
    }
  },
  "wrappedInterferogram": {
    "xCoordinates": {
      "type": "dimension",
      "dimension": "(unwrappedInterferogramWidth)",
      "data_type": "float64",
      "shape": "(23)"
    },
    "yCoordinates": {
      "type": "dimension",
      "dimension": "(unwrappedInterferogramLength)",
      "data_type": "float64",
```

```
                "shape": "(19)"
              },
              "HH": {
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(wrappedInterferogramWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(wrappedInterferogramLength)",
                  "data_type": "float64",
                  "shape": "(19)"
                },
                "wrappedInterferogram": {
                  "type": "dataset",
                  "dimension":
"(wrappedInterferogramLength,wrappedInterferogramWidth)",
                  "data_type": "cfloat32",
                  "shape": "(19,17)"
                },
                "coherenceMagnitude": {
                  "type": "supplemental",
                  "dimension":
"(wrappedInterferogramLength,wrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,17)"
                }
              },
              "VV": {
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(wrappedInterferogramWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(wrappedInterferogramLength)",
                  "data_type": "float64",
                  "shape": "(19)"
                },
                "wrappedInterferogram": {
                  "type": "dataset",
                  "dimension":
"(wrappedInterferogramLength,wrappedInterferogramWidth)",
                  "data_type": "cfloat32",
                  "shape": "(19,17)"
                },
                "coherenceMagnitude": {
                  "type": "supplemental",
```

```
                  "dimension":
"(wrappedInterferogramLength,wrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,17)"
                }
              },
              "mask": {
                "type": "supplemental",
                "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                "data_type": "ubyte",
                "shape": "(19,23)"
              }
            },
            "unwrappedInterferogram": {
              "xCoordinates": {
                "type": "dimension",
                "dimension": "(unwrappedInterferogramWidth)",
                "data_type": "float64",
                "shape": "(23)"
              },
              "yCoordinates": {
                "type": "dimension",
                "dimension": "(unwrappedInterferogramLength)",
                "data_type": "float64",
                "shape": "(19)"
              },
              "mask": {
                "type": "supplemental",
                "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                "data_type": "ubyte",
                "shape": "(19,23)"
              },
              "HH": {
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(unwrappedInterferogramWidth)",
                  "data_type": "float64",
                  "shape": "(23)"
                },
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(unwrappedInterferogramLength)",
                  "data_type": "float64",
                  "shape": "(19)"
                },
                "unwrappedPhase": {
                  "type": "dataset",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "float32",
```

```
                  "shape": "(19,23)"
                },
                "connectedComponents": {
                  "type": "supplemental",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "uint16",
                  "shape": "(19,23)"
                },
                "coherenceMagnitude": {
                  "type": "supplemental",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                },
                "ionospherePhaseScreen": {
                  "type": "supplemental",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                },
                "ionospherePhaseScreenUncertainty": {
                  "type": "supplemental",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                }
              },
              "VV": {
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(unwrappedInterferogramWidth)",
                  "data_type": "float64",
                  "shape": "(23)"
                },
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(unwrappedInterferogramLength)",
                  "data_type": "float64",
                  "shape": "(19)"
                },
                "unwrappedPhase": {
                  "type": "dataset",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                },
                "connectedComponents": {
```

```
                  "type": "supplemental",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "uint16",
                  "shape": "(19,23)"
                },
                "coherenceMagnitude": {
                  "type": "supplemental",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                },
                "ionospherePhaseScreen": {
                  "type": "supplemental",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                },
                "ionospherePhaseScreenUncertainty": {
                  "type": "supplemental",
                  "dimension":
"(unwrappedInterferogramLength,unwrappedInterferogramWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                }
              }
            },
            "pixelOffset": {
              "mask": {
                "type": "supplemental",
                "dimension": "(offsetLength,offsetWidth)",
                "data_type": "ubyte",
                "shape": "(19,17)"
              }
            }
          }
        },
        "metadata": {
          "processingInformation": {
            "parameters": {
              "reference": {
                "referenceTerrainHeight": {
                  "type": "supplemental",
                  "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                },
                "frequencyA": {
                  "dopplerCentroid": {
```

```
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                    "data_type": "float64",
                    "shape": "(19,23)"
                  }
                }
              },
              "secondary": {
                "referenceTerrainHeight": {
                  "type": "supplemental",
                  "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                },
                "frequencyA": {
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                    "data_type": "float64",
                    "shape": "(19,23)"
                  }
                }
              },
              "common": {
                "frequencyA": {
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                    "data_type": "float64",
                    "shape": "(19,23)"
                  }
                }
              }
            }
          },
          "orbit": {
            "reference": {
              "time": {
                "type": "dimension",
                "dimension": "(orbitListLength)",
                "data_type": "float64",
                "shape": "(3)"
              },
              "position": {
                "type": "supplemental",
                "dimension": "(orbitListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(3,3)"
```

```
      },
      "velocity": {
        "type": "supplemental",
        "dimension": "(orbitListLength,tripletxyz)",
        "data_type": "float64",
        "shape": "(3,3)"
      }
    },
    "secondary": {
      "time": {
        "type": "dimension",
        "dimension": "(orbitListLength)",
        "data_type": "float64",
        "shape": "(3)"
      },
      "position": {
        "type": "supplemental",
        "dimension": "(orbitListLength,tripletxyz)",
        "data_type": "float64",
        "shape": "(3,3)"
      },
      "velocity": {
        "type": "supplemental",
        "dimension": "(orbitListLength,tripletxyz)",
        "data_type": "float64",
        "shape": "(3,3)"
      }
    }
  },
  "attitude": {
    "reference": {
      "time": {
        "type": "dimension",
        "dimension": "(attitudeListLength)",
        "data_type": "float64",
        "shape": "(11)"
      },
      "quaternions": {
        "type": "supplemental",
        "dimension": "(attitudeListLength,quaternions)",
        "data_type": "float64",
        "shape": "(11,4)"
      },
      "eulerAngles": {
        "type": "supplemental",
        "dimension": "(attitudeListLength,tripletxyz)",
        "data_type": "float64",
        "shape": "(11,3)"
      }
    },
    "secondary": {
      "time": {
```

```
                "type": "dimension",
                "dimension": "(attitudeListLength)",
                "data_type": "float64",
                "shape": "(11)"
              },
              "quaternions": {
                "type": "supplemental",
                "dimension": "(attitudeListLength,quaternions)",
                "data_type": "float64",
                "shape": "(11,4)"
              },
              "eulerAngles": {
                "type": "supplemental",
                "dimension": "(attitudeListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(11,3)"
              }
            }
          },
          "radarGrid": {
            "referenceSlantRange": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "hydrostaticTroposphericPhaseScreen": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "wetTroposphericPhaseScreen": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "slantRangeSolidEarthTidesPhase": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "referenceZeroDopplerAzimuthTime": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
```

```
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "incidenceAngle": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "losUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "losUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "alongTrackUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "elevationAngle": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "groundTrackVelocity": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
```

```
            "secondaryZeroDopplerAzimuthTime": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "secondarySlantRange": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "parallelBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(23,11,11)"
            },
            "perpendicularBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(23,11,11)"
            },
            "xCoordinates": {
              "type": "cube dimension",
              "dimension": "(radarCubeWidth)",
              "data_type": "float64",
              "shape": "(11)"
            },
            "yCoordinates": {
              "type": "cube dimension",
              "dimension": "(radarCubeLength)",
              "data_type": "float64",
              "shape": "(11)"
            },
            "heightAboveEllipsoid": {
              "type": "cube dimension",
              "dimension": "(radarCubeHeight)",
              "data_type": "float64",
              "shape": "(5)"
            }
          }
        }
      }
    }
  }
}
```

10.7. L2 GCOV

Metadata


```
{
  "granule": "L2_GeocodedCovariance",
  "science": {
    "LSAR": {
      "identification": {
        "absoluteOrbitNumber": "Absolute orbit number",
        "trackNumber": "Track number",
        "frameNumber": "Frame number",
        "missionId": "Mission identifier",
        "processingCenter": "Data processing center",
        "productType": "Product type",
        "granuleId": "Unique granule identification name",
        "productDoi": "Digital Object Identifier (DOI) for the product",
        "productVersion": "Product version which represents the
structure of the product and the science content governed by the
algorithm, input data, and processing parameters",
        "productSpecificationVersion": "Product specification version
which represents the schema of this product",
        "lookDirection": "Look direction, either \"Left\" or \"Right\"",
        "orbitPassDirection": "Orbit direction, either \"Ascending\" or
\"Descending\"",
        "zeroDopplerStartTime": "Azimuth start time (in UTC) of the
product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "zeroDopplerEndTime": "Azimuth stop time (in UTC) of the product
in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "plannedDatatakeId": "List of planned datatakes included in the
product",
        "plannedObservationId": "List of planned observations included
in the product",
        "isUrgentObservation": "Flag indicating if observation is
nominal (\"False\") or urgent (\"True\")",
        "listOfFrequencies": "List of frequency layers available in the
product",
        "listOfObservationModes": "List of observation modes of the L0B
granules used to generate the input RSLC (one mode per L0B)",
        "diagnosticModeFlag": "Indicates if the radar operation mode is
a diagnostic mode (1-2) or DBFed science (0): 0, 1, or 2",
        "productLevel": "Product level. \"L0A\": Unprocessed instrument
data; \"L0B\": Reformatted, unprocessed instrument data; \"L1\":
Processed instrument data in radar coordinates system; and \"L2\":
Processed instrument data in geocoded coordinates system",
        "isGeocoded": "Flag to indicate if the product data is in the
radar geometry (\"False\") or in the map geometry (\"True\")",
        "boundingPolygon": "OGR compatible WKT representing the bounding
polygon of the image. Horizontal coordinates are WGS84 longitude
followed by latitude (both in degrees), and the vertical coordinate is
the height above the WGS84 ellipsoid in meters. The first point
corresponds to the start-time, near-range radar coordinate, and the
perimeter is traversed in counterclockwise order on the map. This means
```

```
the traversal order in radar coordinates differs for left-looking and
right-looking sensors. The polygon includes the four corners of the
radar grid, with equal numbers of points distributed evenly in radar
coordinates along each edge",
        "processingDateTime": "Processing date and time (in UTC) in the
format YYYY-mm-ddTHH:MM:SS",
        "radarBand": "Acquired frequency band, either \"L\" or \"S\"",
        "platformName": "Name of the platform used to collect the remote
sensing data provided in this product",
        "instrumentName": "Name of the instrument used to collect the
remote sensing data provided in this product",
        "processingType": "Processing pipeline used to generate this
granule. \"Nominal\": standard production system; \"Urgent\": time-
sensitive processing in response to urgent response events; \"Custom\":
user-initiated processing outside the nominal production system",
        "isDithered": "\"True\" if the pulse timing was varied
(dithered) during acquisition, \"False\" otherwise",
        "isMixedMode": "\"True\" if this product is a composite of data
collected in multiple radar modes, \"False\" otherwise",
        "isFullFrame": "\"True\" if the product fully covers a NISAR
frame, \"False\" if partial coverage",
        "compositeReleaseId": "Unique version identifier of the science
data production system",
        "isJointObservation": "\"True\" if any portion of this product
was acquired in a joint observation mode (e.g., L-band and S-band
simultaneously), \"False\" otherwise"
      },
      "GCOV": {
        "identification": {
          "staticLayersDataAccess": "Location of the static layers
product associated with this product (URL or DOI)"
        },
        "grids": {
          "frequencyA": {
            "listOfPolarizations": "List of processed polarization
layers with frequency A",
            "listOfCovarianceTerms": "List of processed covariance
terms",
            "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
            "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
            "numberOfLooks": "Number of averaged radar-grid pixels for
covariance estimation",
            "projection": "Product map grid projection: EPSG code, with
additional projection information as HDF5 Attributes",
            "xCoordinates": "X coordinates in specified projection",
            "yCoordinates": "Y coordinates in specified projection",
            "rtcGammaToSigmaFactor": "Radiometric terrain correction
(RTC) scaling factor to normalize backscatter coefficients from gamma0
to sigma0, accounting for local terrain",
            "mask": "Mask indicating the subswath number associated with
```

```
valid GCOV samples. A GCOV sample is only considered valid if it is
generated from fully-focused radar samples. If at least one radar sample
in the averaging set is partially focused or invalid, the corresponding
mask pixel will contain the value 0. GCOV pixels outside of the radar
acquisition extent are filled with the value 255",
            "HHHH": "Covariance between HH and HH",
            "HHHV": "Covariance between HH and HV",
            "HHVH": "Covariance between HH and VH",
            "HHVV": "Covariance between HH and VV",
            "HVHV": "Covariance between HV and HV",
            "HVVH": "Covariance between HV and VH",
            "HVVV": "Covariance between HH and VV",
            "VHVH": "Covariance between VH and VH",
            "VHVV": "Covariance between VH and VV",
            "VVVV": "Covariance between VV and VV",
            "RHRH": "Covariance between RH and RH",
            "RHRV": "Covariance between RH and RV",
            "RVRV": "Covariance between RV and RV",
            "numberOfSubSwaths": "Number of swaths of continuous
imagery, due to transmit gaps"
          },
          "frequencyB": {
            "listOfPolarizations": "List of processed polarization
layers with frequency B",
            "listOfCovarianceTerms": "List of processed covariance
terms",
            "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
            "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
            "numberOfLooks": "Number of averaged radar-grid pixels for
covariance estimation",
            "projection": "Product map grid projection: EPSG code, with
additional projection information as HDF5 Attributes",
            "xCoordinates": "X coordinates in specified projection",
            "yCoordinates": "Y coordinates in specified projection",
            "rtcGammaToSigmaFactor": "Radiometric terrain correction
(RTC) scaling factor to normalize backscatter coefficients from gamma0
to sigma0, accounting for local terrain",
            "mask": "Mask indicating the subswath number associated with
valid GCOV samples. A GCOV sample is only considered valid if it is
generated from fully-focused radar samples. If at least one radar sample
in the averaging set is partially focused or invalid, the corresponding
mask pixel will contain the value 0. GCOV pixels outside of the radar
acquisition extent are filled with the value 255",
            "HHHH": "Covariance between HH and HH",
            "HHHV": "Covariance between HH and HV",
            "HHVH": "Covariance between HH and VH",
            "HHVV": "Covariance between HH and VV",
            "HVHV": "Covariance between HV and HV",
            "HVVH": "Covariance between HV and VH",
            "HVVV": "Covariance between HH and VV",
```

```
            "VHVH": "Covariance between VH and VH",
            "VHVV": "Covariance between VH and VV",
            "VVVV": "Covariance between VV and VV",
            "RHRH": "Covariance between RH and RH",
            "RHRV": "Covariance between RH and RV",
            "RVRV": "Covariance between RV and RV",
            "numberOfSubSwaths": "Number of swaths of continuous
imagery, due to transmit gaps"
          }
        },
        "metadata": {
          "calibrationInformation": {
            "frequencyA": {
              "elevationAntennaPattern": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinates": "X coordinates in specified projection",
                "HH": "Complex two-way elevation antenna pattern",
                "HV": "Complex two-way elevation antenna pattern",
                "VH": "Complex two-way elevation antenna pattern",
                "VV": "Complex two-way elevation antenna pattern",
                "RH": "Complex two-way elevation antenna pattern",
                "RV": "Complex two-way elevation antenna pattern"
              },
              "noiseEquivalentBackscatter": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinates": "X coordinates in specified projection",
                "HH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "HV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RV": "Noise equivalent backscatter in linear scale
(units of DN^2)"
              },
              "commonDelay": "Range delay correction applied to all
polarimetric channels",
              "faradayRotation": "Faraday rotation correction applied in
processing",
              "HH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
```

```
                "differentialDelay": "Range delay correction applied to
HH channel",
                "differentialPhase": "Phase correction applied to HH
channel",
                "scaleFactor": "Scale factor applied to HH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HH
channel complex amplitude with respect to elevation angle"
              },
              "HV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HV channel",
                "differentialPhase": "Phase correction applied to HV
channel",
                "scaleFactor": "Scale factor applied to HV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HV
channel complex amplitude with respect to elevation angle"
              },
              "VH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VH channel",
                "differentialPhase": "Phase correction applied to VH
channel",
                "scaleFactor": "Scale factor applied to VH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VH
channel complex amplitude with respect to elevation angle"
              },
              "VV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VV channel",
                "differentialPhase": "Phase correction applied to VV
channel",
                "scaleFactor": "Scale factor applied to VV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VV
channel complex amplitude with respect to elevation angle"
              },
              "RH": {
```

```
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RH channel",
                "differentialPhase": "Phase correction applied to RH
channel",
                "scaleFactor": "Scale factor applied to RH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RH
channel complex amplitude with respect to elevation angle"
              },
              "RV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RV channel",
                "differentialPhase": "Phase correction applied to RV
channel",
                "scaleFactor": "Scale factor applied to RV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RV
channel complex amplitude with respect to elevation angle"
              }
            },
            "frequencyB": {
              "elevationAntennaPattern": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinates": "X coordinates in specified projection",
                "HH": "Complex two-way elevation antenna pattern",
                "HV": "Complex two-way elevation antenna pattern",
                "VH": "Complex two-way elevation antenna pattern",
                "VV": "Complex two-way elevation antenna pattern",
                "RH": "Complex two-way elevation antenna pattern",
                "RV": "Complex two-way elevation antenna pattern"
              },
              "noiseEquivalentBackscatter": {
                "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                "yCoordinates": "Y coordinates in specified projection",
                "xCoordinates": "X coordinates in specified projection",
                "HH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "HV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "VH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
```

```
                "VV": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RH": "Noise equivalent backscatter in linear scale
(units of DN^2)",
                "RV": "Noise equivalent backscatter in linear scale
(units of DN^2)"
              },
              "HH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HH channel",
                "differentialPhase": "Phase correction applied to HH
channel",
                "scaleFactor": "Scale factor applied to HH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HH
channel complex amplitude with respect to elevation angle"
              },
              "HV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
HV channel",
                "differentialPhase": "Phase correction applied to HV
channel",
                "scaleFactor": "Scale factor applied to HV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to HV
channel complex amplitude with respect to elevation angle"
              },
              "VH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VH channel",
                "differentialPhase": "Phase correction applied to VH
channel",
                "scaleFactor": "Scale factor applied to VH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VH
channel complex amplitude with respect to elevation angle"
              },
              "VV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
```

```
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
VV channel",
                "differentialPhase": "Phase correction applied to VV
channel",
                "scaleFactor": "Scale factor applied to VV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to VV
channel complex amplitude with respect to elevation angle"
              },
              "RH": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RH channel",
                "differentialPhase": "Phase correction applied to RH
channel",
                "scaleFactor": "Scale factor applied to RH channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RH
channel complex amplitude with respect to elevation angle"
              },
              "RV": {
                "rfiLikelihood": "Severity of radio frequency
interference (RFI) contamination in the data. Value is in the interval
[0,1], where 0: lowest severity, and 1: highest severity (or NaN if RFI
detection was skipped)",
                "differentialDelay": "Range delay correction applied to
RV channel",
                "differentialPhase": "Phase correction applied to RV
channel",
                "scaleFactor": "Scale factor applied to RV channel
complex amplitude (at antenna boresite)",
                "scaleFactorSlope": "Slope of scale factor applied to RV
channel complex amplitude with respect to elevation angle"
              },
              "commonDelay": "Range delay correction applied to all
polarimetric channels",
              "faradayRotation": "Faraday rotation correction applied in
processing"
            },
            "crosstalk": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "txHorizontalCrosspol": "Crosstalk in H-transmit channel
expressed as ratio txV / txH",
              "txVerticalCrosspol": "Crosstalk in V-transmit channel
```

```
expressed as ratio txH / txV",
              "rxHorizontalCrosspol": "Crosstalk in H-receive channel
expressed as ratio rxV / rxH",
              "rxVerticalCrosspol": "Crosstalk in V-receive channel
expressed as ratio rxH / rxV"
            }
          },
          "sourceData": {
            "productDoi": "Digital Object Identifier (DOI) of the source
product",
            "productVersion": "Product version of the source data",
            "lookDirection": "Look direction, either \"Left\" or
\"Right\"",
            "productLevel": "Source data product level. Product level.
\"L0A\": Unprocessed instrument data; \"L0B\": Reformatted, unprocessed
instrument data; \"L1\": Processed instrument data in radar coordinates
system; and \"L2\": Processed instrument data in geocoded coordinates
system",
            "productGeometry": "Source data product geometry, either
\"Slant range\" or \"Ground range\"",
            "processingDateTime": "Source data processing date and time
(in UTC) in the format YYYY-mm-ddTHH:MM:SS",
            "processingCenter": "Source data processing center",
            "productType": "Product type of the source data",
            "granuleId": "Unique granule identification name of the
source data",
            "processingInformation": {
              "parameters": {
                "runConfigurationContents": "Contents of the run
configuration file associated with the processing of the source data",
                "referenceTerrainHeight": "Reference terrain height
corresponding to the source data processing information records",
                "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to source data
processing information records",
                "slantRange": "Slant range dimension corresponding to
source data processing information records",
                "frequencyA": {
                  "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to source data
processing information records",
                  "slantRange": "Slant range dimension corresponding to
the source data processing information records",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A corresponding to the source data processing information
records"
                },
                "frequencyB": {
                  "zeroDopplerTime": "Vector of zero Doppler azimuth
times, measured relative to a UTC epoch, corresponding to source data
processing information records",
                  "slantRange": "Slant range dimension corresponding to
```

```
source data processing information records",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency B corresponding to the source data processing information
records"
                }
              },
              "algorithms": {
                "rfiDetection": "Algorithm used for radio frequency
interference (RFI) detection corresponding to the source data processing
information records",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation corresponding to the source data
processing information records, either \"ST-EVD\" or \"FDNF\" (or
\"disabled\" if no RFI mitigation was applied)",
                "rangeCompression": "Algorithm for focusing the data in
the range direction corresponding to the source data processing
information records",
                "elevationAntennaPatternCorrection": "Algorithm for
calibrating the antenna pattern corresponding to the source data
processing information records",
                "rangeSpreadingLossCorrection": "Algorithm for
calibrating range fading corresponding to the source data processing
information records",
                "dopplerCentroidEstimation": "Algorithm for calculating
Doppler centroid corresponding to the source data processing information
records",
                "azimuthPresumming": "Algorithm for regridding and
filling gaps in the raw data in azimuth corresponding to the source data
processing information records",
                "azimuthCompression": "Algorithm for focusing the data
in the azimuth direction corresponding to the source data processing
information records",
                "softwareVersion": "Software version used for processing
the source data"
              },
            },
            "swaths": {
              "zeroDopplerStartTime": "Azimuth start time (in UTC) of
the source data product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
              "zeroDopplerEndTime": "Azimuth stop time (in UTC) of the
source data product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
              "zeroDopplerTimeSpacing": "Time interval in the along-
track direction of the source data in seconds",
              "numberOfAzimuthLines": "Number of azimuth lines within
the source data product",
              "frequencyA": {
                "listOfPolarizations": "List of polarization layers with
frequency A within the source data product",
                "sceneCenterAlongTrackResolution": "Nominal along-track
resolution in meters of the source data at scene center",
                "rangeResolution": "Nominal range resolution in meters
of the source data",
```

```
                "sceneCenterAlongTrackSpacing": "Nominal along-track
spacing in meters between consecutive lines near mid swath of the source
data",
                "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid swath of the
source data",
                "acquiredRangeBandwidth": "Source data acquisition range
bandwidth in hertz. In case of mode combination, this corresponds to
mode with largest bandwidth",
                "processedRangeBandwidth": "Source data processed range
bandwidth in hertz",
                "processedAzimuthBandwidth": "Source data processed
azimuth bandwidth in hertz",
                "centerFrequency": "Center frequency of the processed
source data image in hertz",
                "slantRangeStart": "Source data slant range start
distance",
                "slantRangeSpacing": "Slant range spacing of the source
data in meters",
                "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the source data",
                "nearRangeIncidenceAngle": "Near-range incidence angle
in degrees at the central azimuth line. The incidence angle is defined
as the angle between the LOS vector and the normal to the ellipsoid at
the target height",
                "farRangeIncidenceAngle": "Far-range incidence angle in
degrees at the central azimuth line. The incidence angle is defined as
the angle between the LOS vector and the normal to the ellipsoid at the
target height"
              },
              "frequencyB": {
                "listOfPolarizations": "List of polarization layers with
frequency B within the source data product",
                "sceneCenterAlongTrackResolution": "Nominal along-track
resolution in meters of the source data at scene center",
                "rangeResolution": "Nominal range resolution in meters
of the source data",
                "sceneCenterAlongTrackSpacing": "Nominal along-track
spacing in meters between consecutive lines near mid swath of the source
data",
                "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid swath of the
source data",
                "acquiredRangeBandwidth": "Source data acquisition range
bandwidth in hertz. In case of mode combination, this corresponds to
mode with largest bandwidth",
                "processedRangeBandwidth": "Source data processed range
bandwidth in hertz",
                "processedAzimuthBandwidth": "Source data processed
azimuth bandwidth in hertz",
                "centerFrequency": "Center frequency of the processed
source data image in hertz",
```

```
                "slantRangeStart": "Source data slant range start
distance",
                "slantRangeSpacing": "Slant range spacing of the source
data in meters",
                "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the source data",
                "nearRangeIncidenceAngle": "Near-range incidence angle
in degrees at the central azimuth line. The incidence angle is defined
as the angle between the LOS vector and the normal to the ellipsoid at
the target height",
                "farRangeIncidenceAngle": "Far-range incidence angle in
degrees at the central azimuth line. The incidence angle is defined as
the angle between the LOS vector and the normal to the ellipsoid at the
target height"
              }
            }
          },
          "processingInformation": {
            "parameters": {
              "noiseCorrectionApplied": "Flag to indicate if noise
correction has been applied",
              "preprocessingMultilookingApplied": "Flag to indicate if a
preprocessing multilooking has been applied",
              "polarizationOrientationCorrectionApplied": "Flag to
indicate if the polarization orientation correction has been applied",
              "faradayRotationApplied": "Flag to indicate if the Faraday
rotation correction has been applied",
              "radiometricTerrainCorrectionApplied": "Flag to indicate
if the radiometric terrain correction has been applied",
              "dryTroposphericGeolocationCorrectionApplied": "Flag to
indicate if the dry tropospheric correction was applied during the
generation of the input RSLC to improve geolocation",
              "wetTroposphericGeolocationCorrectionApplied": "Flag to
indicate if the wet tropospheric correction was applied during the
generation of the input RSLC to improve geolocation",
              "rangeIonosphericGeolocationCorrectionApplied": "Flag to
indicate if the range ionospheric correction has been applied",
              "azimuthIonosphericGeolocationCorrectionApplied": "Flag to
indicate if the azimuth ionospheric correction has been applied",
              "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the input RSLC",
              "postProcessingFilteringApplied": "Flag to indicate if the
post-processing filtering has been applied",
              "isFullCovariance": "Flag to indicate if the product is
full-covariance",
              "validSamplesSubSwathMaskingApplied": "Flag to indicate if
the valid samples subswath masking has been applied",
              "shadowMaskingApplied": "Flag to indicate if the shadow
masking has been applied",
              "polarimetricSymmetrizationApplied": "Flag to indicate if
the polarimetric symmetrization has been applied",
```

```
            "preprocessing": {
              "frequencyA": {
                "numberOfRangeLooks": "Number of range looks applied
to the RSLC before geocoding",
                "numberOfAzimuthLooks": "Number of azimuth looks
applied to the RSLC before geocoding"
              },
              "frequencyB": {
                "numberOfRangeLooks": "Number of range looks applied
to the RSLC before geocoding",
                "numberOfAzimuthLooks": "Number of azimuth looks
applied to the RSLC before geocoding"
              }
            },
            "rtc": {
              "inputBackscatterNormalizationConvention": "Backscatter
normalization convention of the source data",
              "outputBackscatterNormalizationConvention": "Backscatter
normalization convention of the primary data associated with this
product",
              "outputBackscatterExpressionConvention": "Backscatter
expression convention",
              "memoryMode": "Radiometric terrain correction (RTC)
memory mode",
              "minRtcAreaNormalizationFactorInDB": "Radiometric
terrain correction (RTC) minimum area normalization factor value in dB
computed as `10 * log10(area_out / area_in)` where `area_in` and
`area_out` are the reference surfaces associated with the source data
(input) and GCOV terms (output) backscatter conventions, respectively",
              "geogridUpsampling": "Radiometric terrain correction
(RTC) geogrid upsampling"
            },
            "geocoding": {
              "memoryMode": "Geocoding memory mode",
              "geogridUpsampling": "Geocoding geogrid upsampling",
              "minBlockSize": "Minimum block size in megabytes (MB)
per thread",
              "maxBlockSize": "Maximum block size in megabytes (MB)
per thread",
              "isSourceDataUpsampled": "Flag to indicate if the source
data is upsampled for geocoding"
            },
            "geo2rdr": {
              "convergenceThreshold": "Slant range convergence
threshold for geo2rdr transformation",
              "maxNumberOfIterations": "Maximum number of iterations
for geo2rdr transformation",
              "deltaRange": "Step size for computing numerical
gradient of Doppler in meters for geo2rdr transformation"
            },
            "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
```

```
            "yCoordinates": "Y coordinates in specified projection",
            "xCoordinates": "X coordinates in specified projection",
            "referenceTerrainHeight": "Reference terrain height as a
function of map coordinates",
            "frequencyA": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
            },
            "frequencyB": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "dopplerCentroid": "2D LUT of Doppler centroid for
frequency B"
            },
            "runConfigurationContents": "Contents of the run
configuration file with parameters used for processing"
          },
          "algorithms": {
            "demInterpolation": "DEM interpolation method",
            "geocoding": "Geocoding algorithm",
            "radiometricTerrainCorrection": "Radiometric terrain
correction (RTC) algorithm",
            "polarimetricSymmetrization": "Polarimetric symmetrization
algorithm",
            "radiometricTerrainCorrectionAlgorithmReference":
"Reference to the radiometric terrain correction (RTC) algorithm applied
(if applicable)",
            "geocodingAlgorithmReference": "Reference to the geocoding
algorithm applied (if applicable)",
            "softwareVersion": "Software version used for processing"
          },
          "timingCorrections": {
            "frequencyA": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
              "xCoordinates": "X coordinates in specified projection",
              "azimuthIonosphere": "Azimuth ionospheric timing
correction, derived from Total Electron Content (TEC) data",
              "slantRangeIonosphere": "Slant range ionospheric timing
correction, derived from Total Electron Content (TEC) data"
            },
            "frequencyB": {
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinates": "Y coordinates in specified projection",
```

```
                "xCoordinates": "X coordinates in specified projection",
                "azimuthIonosphere": "Azimuth ionospheric timing
correction, derived from Total Electron Content (TEC) data",
                "slantRangeIonosphere": "Slant range ionospheric timing
correction, derived from Total Electron Content (TEC) data"
              }
            },
            "inputs": {
              "l1SlcGranules": "List of input L1 RSLC products used",
              "orbitFiles": "List of input orbit files used",
              "tecFiles": "List of input total electron content (TEC)
files used",
              "configFiles": "List of input config files used",
              "demSource": "Description of the input digital elevation
model (DEM)"
            }
          },
          "ceosAnalysisReadyData": {
            "boundingBox": "OGR compatible WKT representing the bounding
box of the product (the union of the geocoded grid for available
frequency A and frequency B image datasets). Horizontal coordinates are
X and Y directions of the geocoded grid coordinate system, and the
vertical coordinate is the height above the WGS84 ellipsoid in meters.
The first point corresponds to the top left edge of the bounding box.
The remaining points are traversed in counterclockwise order on the map.
The fifth point closes the bounding box by repeating the first point",
            "outputBackscatterDecibelConversionFormula": "Formula to
convert provided backscatter to decibel (dB)",
            "ceosAnalysisReadyDataProductType": "CEOS Analysis Ready
Data (CARD) product type",
            "ceosAnalysisReadyDataDocumentIdentifier": "CEOS Analysis
Ready Data (CARD) document identifier",
            "geometricAccuracy": {
              "bias": {
                "y": "An estimate of the localization error bias in the
Y/northing direction",
                "x": "An estimate of the localization error bias in the
X/easting direction"
              },
              "standardDeviation": {
                "y": "An estimate of the localization error standard
deviation in the Y/northing direction",
                "x": "An estimate of the localization error standard
deviation in the X/easting direction"
              }
            }
          },
          "orbit": {
            "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
```

```
            "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
            "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
            "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
          },
          "attitude": {
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
            "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
            "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
            "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
          },
          "radarGrid": {
            "slantRange": "Slant range in meters",
            "zeroDopplerAzimuthTime": "Zero Doppler azimuth time in
seconds since UTC epoch",
            "incidenceAngle": "Incidence angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the target
height",
            "losUnitVectorX": "East component of the line-of-sight (LOS)
unit vector, defined from the target to the sensor, expressed in the
east-north-up (ENU) coordinate system with its origin at the target
location",
            "losUnitVectorY": "North component of the line-of-sight
(LOS) unit vector, defined from the target to the sensor, expressed in
the east-north-up (ENU) coordinate system with its origin at the target
location",
            "alongTrackUnitVectorX": "East component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "alongTrackUnitVectorY": "North component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "elevationAngle": "Elevation angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the sensor",
            "groundTrackVelocity": "Absolute value of the platform
velocity scaled at the target height",
            "projection": "Product map grid projection: EPSG code, with
additional projection information as HDF5 Attributes",
            "xCoordinates": "X coordinates in specified projection",
            "yCoordinates": "Y coordinates in specified projection",
            "heightAboveEllipsoid": "Height values above WGS84 Ellipsoid
corresponding to the radar grid"
```

```
          }
        }
      }
    }
  }
}
```

Data


```
{
  "granule": "L2_GeocodedCovariance",
  "dimension": {
    "attitudeListLength": 5,
    "calibrationLength": 17,
    "calibrationWidth": 29,
    "frequencyALength": 19,
    "frequencyAWidth": 17,
    "frequencyBLength": 19,
    "frequencyBWidth": 17,
    "orbitListLength": 19,
    "quaternions": 4,
    "radarCubeHeight": 13,
    "radarCubeLength": 17,
    "radarCubeWidth": 3,
    "sourceDataDopplerCentroidHeight": 7,
    "sourceDataDopplerCentroidSlantRangeWidth": 17,
    "sourceDataDopplerCentroidTimeLength": 5,
    "tripletxyz": 3
  },
  "science": {
    "LSAR": {
      "GCOV": {
        "grids": {
          "frequencyA": {
            "numberOfLooks": {
              "type": "supplemental",
              "dimension": "(frequencyALength,frequencyAWidth)",
              "data_type": "float32",
              "shape": "(19,17)"
            },
            "xCoordinates": {
              "type": "dimension",
              "dimension": "(frequencyAWidth)",
              "data_type": "float64",
              "shape": "(17)"
            },
            "yCoordinates": {
              "type": "dimension",
              "dimension": "(frequencyALength)",
              "data_type": "float64",
              "shape": "(19)"
            },
```

```
"rtcGammaToSigmaFactor": {
  "type": "supplemental",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "float32",
  "shape": "(19,17)"
},
"mask": {
  "type": "supplemental",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "ubyte",
  "shape": "(19,17)"
},
"HHHH": {
  "type": "dataset",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "float32",
  "shape": "(19,17)"
},
"HHHV": {
  "type": "dataset",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "cfloat32",
  "shape": "(19,17)"
},
"HHVH": {
  "type": "dataset",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "cfloat32",
  "shape": "(19,17)"
},
"HHVV": {
  "type": "dataset",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "cfloat32",
  "shape": "(19,17)"
},
"HVHV": {
  "type": "dataset",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "float32",
  "shape": "(19,17)"
},
"HVVH": {
  "type": "dataset",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "cfloat32",
  "shape": "(19,17)"
},
"HVVV": {
  "type": "dataset",
  "dimension": "(frequencyALength,frequencyAWidth)",
  "data_type": "cfloat32",
```

```
        "shape": "(19,17)"
      },
      "VHVH": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "VHVV": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "cfloat32",
        "shape": "(19,17)"
      },
      "VVVV": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "RHRH": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "RHRV": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "cfloat32",
        "shape": "(19,17)"
      },
      "RVRV": {
        "type": "dataset",
        "dimension": "(frequencyALength,frequencyAWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      }
    },
    "frequencyB": {
      "numberOfLooks": {
        "type": "supplemental",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(frequencyBWidth)",
        "data_type": "float64",
        "shape": "(17)"
      },
```

```
"yCoordinates": {
  "type": "dimension",
  "dimension": "(frequencyBLength)",
  "data_type": "float64",
  "shape": "(19)"
},
"rtcGammaToSigmaFactor": {
  "type": "supplemental",
  "dimension": "(frequencyBLength,frequencyBWidth)",
  "data_type": "float32",
  "shape": "(19,17)"
},
"mask": {
  "type": "supplemental",
  "dimension": "(frequencyBLength,frequencyBWidth)",
  "data_type": "ubyte",
  "shape": "(19,17)"
},
"HHHH": {
  "type": "dataset",
  "dimension": "(frequencyBLength,frequencyBWidth)",
  "data_type": "float32",
  "shape": "(19,17)"
},
"HHHV": {
  "type": "dataset",
  "dimension": "(frequencyBLength,frequencyBWidth)",
  "data_type": "cfloat32",
  "shape": "(19,17)"
},
"HHVH": {
  "type": "dataset",
  "dimension": "(frequencyBLength,frequencyBWidth)",
  "data_type": "cfloat32",
  "shape": "(19,17)"
},
"HHVV": {
  "type": "dataset",
  "dimension": "(frequencyBLength,frequencyBWidth)",
  "data_type": "cfloat32",
  "shape": "(19,17)"
},
"HVHV": {
  "type": "dataset",
  "dimension": "(frequencyBLength,frequencyBWidth)",
  "data_type": "float32",
  "shape": "(19,17)"
},
"HVVH": {
  "type": "dataset",
  "dimension": "(frequencyBLength,frequencyBWidth)",
  "data_type": "cfloat32",
```

```
        "shape": "(19,17)"
      },
      "HVVV": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(19,17)"
      },
      "VHVH": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "VHVV": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(19,17)"
      },
      "VVVV": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "RHRH": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "RHRV": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "cfloat32",
        "shape": "(19,17)"
      },
      "RVRV": {
        "type": "dataset",
        "dimension": "(frequencyBLength,frequencyBWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      }
    }
  },
  "metadata": {
    "calibrationInformation": {
      "frequencyA": {
        "elevationAntennaPattern": {
          "yCoordinates": {
            "type": "dimension",
```

```
      "dimension": "(calibrationLength)",
      "data_type": "float64",
      "shape": "(17)"
    },
    "xCoordinates": {
      "type": "dimension",
      "dimension": "(calibrationWidth)",
      "data_type": "float64",
      "shape": "(29)"
    },
    "HH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "HV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "VH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "VV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "RH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "RV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    }
  },
  "noiseEquivalentBackscatter": {
    "yCoordinates": {
      "type": "dimension",
      "dimension": "(calibrationLength)",
      "data_type": "float64",
```

```
        "shape": "(17)"
      },
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(calibrationWidth)",
        "data_type": "float64",
        "shape": "(29)"
      },
      "HH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "HV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "VH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "VV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "RH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "RV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      }
    }
  },
  "frequencyB": {
    "elevationAntennaPattern": {
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(calibrationLength)",
        "data_type": "float64",
```

```
      "shape": "(17)"
    },
    "xCoordinates": {
      "type": "dimension",
      "dimension": "(calibrationWidth)",
      "data_type": "float64",
      "shape": "(29)"
    },
    "HH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "HV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "VH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "VV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "RH": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    },
    "RV": {
      "type": "supplemental",
      "dimension": "(calibrationLength,calibrationWidth)",
      "data_type": "cfloat32",
      "shape": "(17,29)"
    }
  },
  "noiseEquivalentBackscatter": {
    "yCoordinates": {
      "type": "dimension",
      "dimension": "(calibrationLength)",
      "data_type": "float64",
      "shape": "(17)"
    },
```

```
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(calibrationWidth)",
        "data_type": "float64",
        "shape": "(29)"
      },
      "HH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "HV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "VH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "VV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "RH": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      },
      "RV": {
        "type": "supplemental",
        "dimension": "(calibrationLength,calibrationWidth)",
        "data_type": "float32",
        "shape": "(17,29)"
      }
    }
  },
  "crosstalk": {
    "yCoordinates": {
      "type": "dimension",
      "dimension": "(calibrationLength)",
      "data_type": "float64",
      "shape": "(17)"
    },
    "xCoordinates": {
```

```
                "type": "dimension",
                "dimension": "(calibrationWidth)",
                "data_type": "float64",
                "shape": "(29)"
              },
              "txHorizontalCrosspol": {
                "type": "supplemental",
                "dimension": "(calibrationLength,calibrationWidth)",
                "data_type": "cfloat32",
                "shape": "(17,29)"
              },
              "txVerticalCrosspol": {
                "type": "supplemental",
                "dimension": "(calibrationLength,calibrationWidth)",
                "data_type": "cfloat32",
                "shape": "(17,29)"
              },
              "rxHorizontalCrosspol": {
                "type": "supplemental",
                "dimension": "(calibrationLength,calibrationWidth)",
                "data_type": "cfloat32",
                "shape": "(17,29)"
              },
              "rxVerticalCrosspol": {
                "type": "supplemental",
                "dimension": "(calibrationLength,calibrationWidth)",
                "data_type": "cfloat32",
                "shape": "(17,29)"
              }
            }
          },
          "sourceData": {
            "processingInformation": {
              "parameters": {
                "referenceTerrainHeight": {
                  "type": "dimension",
                  "dimension": "(sourceDataDopplerCentroidHeight)",
                  "data_type": "float32",
                  "shape": "(7)"
                },
                "zeroDopplerTime": {
                  "type": "dimension",
                  "dimension": "(sourceDataDopplerCentroidTimeLength)",
                  "data_type": "float64",
                  "shape": "(5)"
                },
                "slantRange": {
                  "type": "dimension",
                  "dimension":
"(sourceDataDopplerCentroidSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
```

```
                },
                "frequencyA": {
                  "zeroDopplerTime": {
                    "type": "dimension",
                    "dimension":
"(sourceDataDopplerCentroidTimeLength)",
                    "data_type": "float64",
                    "shape": "(5)"
                  },
                  "slantRange": {
                    "type": "dimension",
                    "dimension":
"(sourceDataDopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(17)"
                  },
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(sourceDataDopplerCentroidTimeLength,sourceDataDopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(5,17)"
                  }
                },
                "frequencyB": {
                  "zeroDopplerTime": {
                    "type": "dimension",
                    "dimension":
"(sourceDataDopplerCentroidTimeLength)",
                    "data_type": "float64",
                    "shape": "(5)"
                  },
                  "slantRange": {
                    "type": "dimension",
                    "dimension":
"(sourceDataDopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(17)"
                  },
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(sourceDataDopplerCentroidTimeLength,sourceDataDopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(5,17)"
                  }
                }
              }
            }
          },
          "processingInformation": {
            "parameters": {
```

```
              "yCoordinates": {
                "type": "dimension",
                "dimension": "(sourceDataDopplerCentroidTimeLength)",
                "data_type": "float64",
                "shape": "(5)"
              },
              "xCoordinates": {
                "type": "dimension",
                "dimension":
"(sourceDataDopplerCentroidSlantRangeWidth)",
                "data_type": "float64",
                "shape": "(17)"
              },
              "referenceTerrainHeight": {
                "type": "supplemental",
                "dimension":
"(sourceDataDopplerCentroidTimeLength,sourceDataDopplerCentroidSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(5,17)"
              },
              "frequencyA": {
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(sourceDataDopplerCentroidTimeLength)",
                  "data_type": "float64",
                  "shape": "(5)"
                },
                "xCoordinates": {
                  "type": "dimension",
                  "dimension":
"(sourceDataDopplerCentroidSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "dopplerCentroid": {
                  "type": "supplemental",
                  "dimension":
"(sourceDataDopplerCentroidTimeLength,sourceDataDopplerCentroidSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(5,17)"
                }
              },
              "frequencyB": {
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(sourceDataDopplerCentroidTimeLength)",
                  "data_type": "float64",
                  "shape": "(5)"
                },
                "xCoordinates": {
                  "type": "dimension",
                  "dimension":
```

```
"(sourceDataDopplerCentroidSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "dopplerCentroid": {
                  "type": "supplemental",
                  "dimension":
"(sourceDataDopplerCentroidTimeLength,sourceDataDopplerCentroidSlantRangeWidth)",
                  "data_type": "float64",
                  "shape": "(5,17)"
                }
              }
            },
            "timingCorrections": {
              "frequencyA": {
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(calibrationLength)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(calibrationWidth)",
                  "data_type": "float64",
                  "shape": "(29)"
                },
                "azimuthIonosphere": {
                  "type": "supplemental",
                  "dimension": "(calibrationLength,calibrationWidth)",
                  "data_type": "float64",
                  "shape": "(17,29)"
                },
                "slantRangeIonosphere": {
                  "type": "supplemental",
                  "dimension": "(calibrationLength,calibrationWidth)",
                  "data_type": "float64",
                  "shape": "(17,29)"
                }
              },
              "frequencyB": {
                "yCoordinates": {
                  "type": "dimension",
                  "dimension": "(calibrationLength)",
                  "data_type": "float64",
                  "shape": "(17)"
                },
                "xCoordinates": {
                  "type": "dimension",
                  "dimension": "(calibrationWidth)",
                  "data_type": "float64",
                  "shape": "(29)"
```

```
        },
        "azimuthIonosphere": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "float64",
          "shape": "(17,29)"
        },
        "slantRangeIonosphere": {
          "type": "supplemental",
          "dimension": "(calibrationLength,calibrationWidth)",
          "data_type": "float64",
          "shape": "(17,29)"
        }
      }
    }
  },
  "orbit": {
    "time": {
      "type": "dimension",
      "dimension": "(orbitListLength)",
      "data_type": "float64",
      "shape": "(19)"
    },
    "position": {
      "type": "supplemental",
      "dimension": "(orbitListLength,tripletxyz)",
      "data_type": "float64",
      "shape": "(19,3)"
    },
    "velocity": {
      "type": "supplemental",
      "dimension": "(orbitListLength,tripletxyz)",
      "data_type": "float64",
      "shape": "(19,3)"
    }
  },
  "attitude": {
    "time": {
      "type": "dimension",
      "dimension": "(attitudeListLength)",
      "data_type": "float64",
      "shape": "(5)"
    },
    "quaternions": {
      "type": "supplemental",
      "dimension": "(attitudeListLength,quaternions)",
      "data_type": "float64",
      "shape": "(5,4)"
    },
    "eulerAngles": {
      "type": "supplemental",
      "dimension": "(attitudeListLength,tripletxyz)",
```

```
              "data_type": "float64",
              "shape": "(5,3)"
            }
          },
          "radarGrid": {
            "slantRange": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(13,17,3)"
            },
            "zeroDopplerAzimuthTime": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(13,17,3)"
            },
            "incidenceAngle": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(13,17,3)"
            },
            "losUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(13,17,3)"
            },
            "losUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(13,17,3)"
            },
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(13,17,3)"
            },
            "alongTrackUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
```

```
            "shape": "(13,17,3)"
          },
          "elevationAngle": {
            "type": "cube",
            "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
            "data_type": "float32",
            "shape": "(13,17,3)"
          },
          "xCoordinates": {
            "type": "cube dimension",
            "dimension": "(radarCubeWidth)",
            "data_type": "float64",
            "shape": "(3)"
          },
          "yCoordinates": {
            "type": "cube dimension",
            "dimension": "(radarCubeLength)",
            "data_type": "float64",
            "shape": "(17)"
          },
          "heightAboveEllipsoid": {
            "type": "cube dimension",
            "dimension": "(radarCubeHeight)",
            "data_type": "float64",
            "shape": "(13)"
          }
        }
      }
    }
  }
}
```

10.8. L1 ROFF

Metadata


```
{
  "granule": "L1_RadarPixelOffsets",
  "science": {
    "LSAR": {
      "identification": {
        "referenceAbsoluteOrbitNumber": "Absolute orbit number for the
reference RSLC",
        "secondaryAbsoluteOrbitNumber": "Absolute orbit number for the
secondary RSLC",
        "referenceIsJointObservation": "\"True\" if any portion of the
reference RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "secondaryIsJointObservation": "\"True\" if any portion of the
secondary RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "trackNumber": "Track number",
        "frameNumber": "Frame number",
        "missionId": "Mission identifier",
        "processingCenter": "Data processing center",
        "productType": "Product type",
        "granuleId": "Unique granule identification name",
        "productDoi": "Digital Object Identifier (DOI) for the product",
        "productVersion": "Product version which represents the
structure of the product and the science content governed by the
algorithm, input data, and processing parameters",
        "productSpecificationVersion": "Product specification version
which represents the schema of this product",
        "lookDirection": "Look direction, either \"Left\" or \"Right\"",
        "orbitPassDirection": "Orbit direction, either \"Ascending\" or
\"Descending\"",
        "referenceZeroDopplerStartTime": "Azimuth start time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referenceZeroDopplerEndTime": "Azimuth stop time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerStartTime": "Azimuth start time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerEndTime": "Azimuth stop time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referencePlannedDatatakeId": "List of planned datatakes
included in the reference RSLC",
        "secondaryPlannedDatatakeId": "List of planned datatakes
included in the secondary RSLC",
        "referencePlannedObservationId": "List of planned observations
included in the reference RSLC ",
        "secondaryPlannedObservationId": "List of planned observations
included in the secondary RSLC ",
        "referenceListOfObservationModes": "List of observation modes of
the L0B granules used to generate the reference RSLC (one mode per
L0B)",
```

```
        "secondaryListOfObservationModes": "List of observation modes of
the L0B granules used to generate the secondary RSLC (one mode per
L0B)",
        "isUrgentObservation": "Flag indicating if observation is
nominal (\"False\") or urgent (\"True\")",
        "listOfFrequencies": "List of frequency layers available in the
product",
        "diagnosticModeFlag": "Indicates if the radar operation mode is
a diagnostic mode (1-2) or DBFed science (0): 0, 1, or 2",
        "productLevel": "Product level. \"L0A\": Unprocessed instrument
data; \"L0B\": Reformatted, unprocessed instrument data; \"L1\":
Processed instrument data in radar coordinates system; and \"L2\":
Processed instrument data in geocoded coordinates system",
        "isGeocoded": "Flag to indicate if the product data is in the
radar geometry (\"False\") or in the map geometry (\"True\")",
        "boundingPolygon": "OGR compatible WKT representing the bounding
polygon of the image. Horizontal coordinates are WGS84 longitude
followed by latitude (both in degrees), and the vertical coordinate is
the height above the WGS84 ellipsoid in meters. The first point
corresponds to the start-time, near-range radar coordinate, and the
perimeter is traversed in counterclockwise order on the map. This means
the traversal order in radar coordinates differs for left-looking and
right-looking sensors. The polygon includes the four corners of the
radar grid, with equal numbers of points distributed evenly in radar
coordinates along each edge",
        "processingDateTime": "Processing date and time (in UTC) in the
format YYYY-mm-ddTHH:MM:SS",
        "radarBand": "Acquired frequency band, either \"L\" or \"S\"",
        "platformName": "Name of the platform used to collect the remote
sensing data provided in this product",
        "instrumentName": "Name of the instrument used to collect the
remote sensing data provided in this product",
        "processingType": "Processing pipeline used to generate this
granule. \"Nominal\": standard production system; \"Urgent\": time-
sensitive processing in response to urgent response events; \"Custom\":
user-initiated processing outside the nominal production system",
        "isDithered": "\"True\" if the pulse timing was varied
(dithered) during acquisition, \"False\" otherwise",
        "isMixedMode": "\"True\" if this product is generated from
reference and secondary RSLCs with different range bandwidths, \"False\"
otherwise",
        "isFullFrame": "\"True\" if the product fully covers a NISAR
frame, \"False\" if partial coverage",
        "compositeReleaseId": "Unique version identifier of the science
data production system"
      },
      "ROFF": {
        "swaths": {
          "frequencyA": {
            "listOfPolarizations": "List of processed polarization
layers with frequency A",
            "centerFrequency": "Center frequency of the processed image
```

```
in hertz",
            "listOfLayers": "List of pixel offsets layers",
            "pixelOffsets": {
              "sceneCenterAlongTrackSpacing": "Nominal along-track
spacing in meters between consecutive lines near mid-swath of the
product images",
              "sceneCenterGroundRangeSpacing": "Nominal ground range
spacing in meters between consecutive pixels near mid-swath of the
product images",
              "slantRangeSpacing": "Slant range spacing of the offset
grid",
              "zeroDopplerTimeSpacing": "Along-track spacing of the
offset grid",
              "mask": "Mask indicating the subswaths of valid samples in
the reference RSLC and geometrically-coregistered secondary RSLC. Each
pixel value is a two-digit number: the least significant digit
represents the subswath number of that pixel in the secondary RSLC, and
the most significant digit represents the subswath number of that pixel
in the reference RSLC. A value of 0 in either digit indicates an invalid
sample in the corresponding RSLC",
              "digitalElevationModel": "Digital Elevation Model (DEM) in
radar coordinates. This dataset is generated by referencing the input
DEM elevations to the WGS84 ellipsoid and projecting them onto a slant
range/azimuth grid",
              "HH": {
                "layer1": {
                  "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "snr": "Pixel offsets signal-to-noise ratio",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                  "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix"
                },
                "layer2": {
                  "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "snr": "Pixel offsets signal-to-noise ratio",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
```

```
                "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix"
              },
              "layer3": {
                "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                "snr": "Pixel offsets signal-to-noise ratio",
                "correlationSurfacePeak": "Normalized correlation
surface peak",
                "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix"
              }
            },
            "VV": {
              "layer1": {
                "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                "snr": "Pixel offsets signal-to-noise ratio",
                "correlationSurfacePeak": "Normalized correlation
surface peak",
                "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix"
              },
              "layer2": {
                "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                "snr": "Pixel offsets signal-to-noise ratio",
                "correlationSurfacePeak": "Normalized correlation
surface peak",
                "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                "alongTrackOffsetVariance": "Along-track pixel offset
variance",
                "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix"
              },
              "layer3": {
                "slantRangeOffset": "Raw (unculled, unfiltered) slant
```

```
range pixel offsets",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "snr": "Pixel offsets signal-to-noise ratio",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                  "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix"
                }
              },
              "slantRange": "Slant range vector",
              "zeroDopplerTime": "Vector of zero Doppler azimuth times
measured relative to a UTC epoch"
            }
          }
        },
        "metadata": {
          "processingInformation": {
            "parameters": {
              "runConfigurationContents": "Contents of the run
configuration file with parameters used for processing",
              "reference": {
                "referenceTerrainHeight": "Reference terrain height as a
function of time for reference RSLC",
                "isMixedMode": "\"True\" if reference RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the reference RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the reference RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "frequencyA": {
                  "slantRangeStart": "Slant range start distance for the
reference RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the reference RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the reference RSLC",
                  "slantRangeSpacing": "Slant range spacing of reference
RSLC",
                  "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the reference RSLC product in the format YYYY-mm-
ddTHH:MM:SS.ssssssssss",
                  "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                  "zeroDopplerTimeSpacing": "Time interval in the along-
```

```
track direction for reference RSLC raster layers",
                  "rangeBandwidth": "Processed slant range bandwidth for
reference RSLC",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
reference RSLC",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
                }
              },
              "secondary": {
                "referenceTerrainHeight": "Reference terrain height as a
function of time for secondary RSLC",
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the secondary RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the secondary RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "isMixedMode": "\"True\" if secondary RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
                "frequencyA": {
                  "slantRangeStart": "Slant range start distance for the
secondary RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the secondary RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the secondary RSLC",
                  "slantRangeSpacing": "Slant range spacing of secondary
RSLC",
                  "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the secondary RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                  "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                  "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for secondary RSLC raster layers",
                  "rangeBandwidth": "Processed slant range bandwidth for
secondary RSLC",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
secondary RSLC",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
                }
              },
              "common": {
                "frequencyA": {
                  "dopplerCentroid": "2D LUT of common Doppler centroid
between reference and secondary RSLCs",
                  "dopplerBandwidth": "Common Doppler bandwidth between
reference and secondary RSLCs"
                }
```

```
            },
            "pixelOffsets": {
              "frequencyA": {
                "rangeBandwidth": "Processed slant range bandwidth for
frequency A pixel offsets layers",
                "azimuthBandwidth": "Processed azimuth bandwidth for
frequency A pixel offsets layers",
                "correlationSurfaceOversampling": "Oversampling factor
of the cross-correlation surface",
                "margin": "Margin in pixels around reference RSLC
edges excluded during cross-correlation computation",
                "slantRangeStartPixel": "Reference RSLC start pixel in
slant range",
                "alongTrackStartPixel": "Reference RSLC start pixel in
along-track",
                "slantRangeSkipWindowSize": "Slant range cross-
correlation skip window size in pixels",
                "alongTrackSkipWindowSize": "Along-track cross-
correlation skip window size in pixels",
                "layer1": {
                  "alongTrackWindowSize": "Along-track cross-
correlation window size in pixels",
                  "slantRangeWindowSize": "Slant range cross-
correlation window size in pixels",
                  "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                  "slantRangeSearchWindowSize": "Slant range cross-
correlation search window size in pixels"
                },
                "layer2": {
                  "alongTrackWindowSize": "Along-track cross-
correlation window size in pixels",
                  "slantRangeWindowSize": "Slant range cross-
correlation window size in pixels",
                  "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                  "slantRangeSearchWindowSize": "Slant range cross-
correlation search window size in pixels"
                },
                "layer3": {
                  "alongTrackWindowSize": "Along-track cross-
correlation window size in pixels",
                  "slantRangeWindowSize": "Slant range cross-
correlation window size in pixels",
                  "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                  "slantRangeSearchWindowSize": "Slant range cross-
correlation search window size in pixels"
                }
              }
            }
          },
```

```
            "algorithms": {
              "softwareVersion": "Software version used for processing",
              "coregistration": {
                "coregistrationMethod": "RSLC coregistration method",
                "geometryCoregistration": "Geometry coregistration
algorithm",
                "resampling": "Secondary RSLC resampling algorithm"
              },
              "crossCorrelation": {
                "layer1": {
                  "crossCorrelationAlgorithm": "Cross-correlation
algorithm for layer 1"
                },
                "layer2": {
                  "crossCorrelationAlgorithm": "Cross-correlation
algorithm for layer 2"
                },
                "layer3": {
                  "crossCorrelationAlgorithm": "Cross-correlation
algorithm for layer 3"
                }
              }
            },
            "inputs": {
              "l1ReferenceSlcGranules": "List of input reference L1 RSLC
products used",
              "l1SecondarySlcGranules": "List of input secondary L1 RSLC
products used",
              "configFiles": "List of input config files used",
              "demSource": "Description of the input digital elevation
model (DEM)",
              "waterMaskSource": "Description of the input water mask",
              "orbitFiles": "List of input orbit files used"
            }
          },
          "orbit": {
            "temporalBaseline": "Time interval between reference and
secondary RSLCs",
            "reference": {
              "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
              "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
              "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
              "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
            },
```

```
            "secondary": {
              "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
              "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
              "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
              "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
            }
          },
          "attitude": {
            "reference": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            },
            "secondary": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            }
          },
          "geolocationGrid": {
            "epsg": "EPSG code corresponding to the coordinate system
used for representing the geolocation grid",
            "coordinateY": "Y coordinates in specified EPSG code",
            "coordinateX": "X coordinates in specified EPSG code",
            "incidenceAngle": "Incidence angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the target
height",
            "losUnitVectorX": "East component of the line-of-sight (LOS)
unit vector, defined from the target to the sensor, expressed in the
east-north-up (ENU) coordinate system with its origin at the target
location",
            "losUnitVectorY": "North component of the line-of-sight
(LOS) unit vector, defined from the target to the sensor, expressed in
the east-north-up (ENU) coordinate system with its origin at the target
```

```
location",
            "alongTrackUnitVectorX": "East component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "alongTrackUnitVectorY": "North component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "elevationAngle": "Elevation angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the sensor",
            "parallelBaseline": "Parallel component of the InSAR
baseline",
            "perpendicularBaseline": "Perpendicular component of the
InSAR baseline",
            "slantRange": "Slant range values corresponding to the
geolocation grid",
            "zeroDopplerTime": "Vector of zero Doppler azimuth times,
measured relative to a UTC epoch, corresponding to the geolocation
grid",
            "groundTrackVelocity": "Absolute value of the platform
velocity scaled at the target height",
            "heightAboveEllipsoid": "Height values above WGS84 Ellipsoid
corresponding to the location grid"
          }
        }
      }
    }
  }
}
```


Data


```
{
  "granule": "L1_RadarPixelOffsets",
  "dimension": {
    "attitudeListLength": 17,
    "dopplerCentroidSlantRangeWidth": 3,
    "dopplerCentroidTimeLength": 19,
    "geolocationCubeHeight": 11,
    "geolocationCubeLength": 17,
    "geolocationCubeWidth": 17,
    "offsetSlantRangeWidth": 19,
    "offsetZeroDopplerTimeLength": 17,
    "orbitListLength": 5,
    "quaternions": 4,
    "tripletxyz": 3,
    "twoLayersCubeHeight": 3
  },
  "science": {
    "LSAR": {
      "ROFF": {
```

```
        "swaths": {
          "frequencyA": {
            "pixelOffsets": {
              "mask": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "ubyte",
                "shape": "(17,19)"
              },
              "digitalElevationModel": {
                "type": "supplemental",
                "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                "data_type": "float32",
                "shape": "(17,19)"
              },
              "HH": {
                "layer1": {
                  "slantRangeOffset": {
                    "type": "dataset",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "alongTrackOffset": {
                    "type": "dataset",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "snr": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "correlationSurfacePeak": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "slantRangeOffsetVariance": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
```

```
                    "shape": "(17,19)"
                  },
                  "alongTrackOffsetVariance": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "crossOffsetVariance": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  }
                },
                "layer2": {
                  "slantRangeOffset": {
                    "type": "dataset",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "alongTrackOffset": {
                    "type": "dataset",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "snr": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "correlationSurfacePeak": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "slantRangeOffsetVariance": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
```

```
                    },
                    "alongTrackOffsetVariance": {
                      "type": "supplemental",
                      "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                      "data_type": "float32",
                      "shape": "(17,19)"
                    },
                    "crossOffsetVariance": {
                      "type": "supplemental",
                      "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                      "data_type": "float32",
                      "shape": "(17,19)"
                    }
                  },
                  "layer3": {
                    "slantRangeOffset": {
                      "type": "dataset",
                      "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                      "data_type": "float32",
                      "shape": "(17,19)"
                    },
                    "alongTrackOffset": {
                      "type": "dataset",
                      "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                      "data_type": "float32",
                      "shape": "(17,19)"
                    },
                    "snr": {
                      "type": "supplemental",
                      "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                      "data_type": "float32",
                      "shape": "(17,19)"
                    },
                    "correlationSurfacePeak": {
                      "type": "supplemental",
                      "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                      "data_type": "float32",
                      "shape": "(17,19)"
                    },
                    "slantRangeOffsetVariance": {
                      "type": "supplemental",
                      "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                      "data_type": "float32",
                      "shape": "(17,19)"
                    },
```

```
                "alongTrackOffsetVariance": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "crossOffsetVariance": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                }
              }
            },
            "VV": {
              "layer1": {
                "slantRangeOffset": {
                  "type": "dataset",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "alongTrackOffset": {
                  "type": "dataset",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "snr": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "correlationSurfacePeak": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "slantRangeOffsetVariance": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
```

```
                  },
                  "alongTrackOffsetVariance": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "crossOffsetVariance": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  }
                },
                "layer2": {
                  "slantRangeOffset": {
                    "type": "dataset",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "alongTrackOffset": {
                    "type": "dataset",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "snr": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "correlationSurfacePeak": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "slantRangeOffsetVariance": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
```

```
                "alongTrackOffsetVariance": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "crossOffsetVariance": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                }
              },
              "layer3": {
                "slantRangeOffset": {
                  "type": "dataset",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "alongTrackOffset": {
                  "type": "dataset",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "snr": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "correlationSurfacePeak": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "slantRangeOffsetVariance": {
                  "type": "supplemental",
                  "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                  "data_type": "float32",
                  "shape": "(17,19)"
                },
                "alongTrackOffsetVariance": {
```

```
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  },
                  "crossOffsetVariance": {
                    "type": "supplemental",
                    "dimension":
"(offsetZeroDopplerTimeLength,offsetSlantRangeWidth)",
                    "data_type": "float32",
                    "shape": "(17,19)"
                  }
                }
              },
              "slantRange": {
                "type": "dimension",
                "dimension": "(offsetSlantRangeWidth)",
                "data_type": "float64",
                "shape": "(19)"
              },
              "zeroDopplerTime": {
                "type": "dimension",
                "dimension": "(offsetZeroDopplerTimeLength)",
                "data_type": "float64",
                "shape": "(17)"
              }
            }
          }
        },
        "metadata": {
          "processingInformation": {
            "parameters": {
              "reference": {
                "referenceTerrainHeight": {
                  "type": "dimension",
                  "dimension": "(dopplerCentroidTimeLength)",
                  "data_type": "float32",
                  "shape": "(19)"
                },
                "frequencyA": {
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                    "data_type": "float64",
                    "shape": "(19,3)"
                  }
                }
              },
              "secondary": {
                "referenceTerrainHeight": {
```

```
                    "type": "dimension",
                    "dimension": "(dopplerCentroidTimeLength)",
                    "data_type": "float32",
                    "shape": "(19)"
                  },
                  "frequencyA": {
                    "dopplerCentroid": {
                      "type": "supplemental",
                      "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                      "data_type": "float64",
                      "shape": "(19,3)"
                    }
                  }
                },
                "common": {
                  "frequencyA": {
                    "dopplerCentroid": {
                      "type": "supplemental",
                      "dimension":
"(dopplerCentroidTimeLength,dopplerCentroidSlantRangeWidth)",
                      "data_type": "float64",
                      "shape": "(19,3)"
                    }
                  }
                }
              }
            },
            "orbit": {
              "reference": {
                "time": {
                  "type": "dimension",
                  "dimension": "(orbitListLength)",
                  "data_type": "float64",
                  "shape": "(5)"
                },
                "position": {
                  "type": "supplemental",
                  "dimension": "(orbitListLength,tripletxyz)",
                  "data_type": "float64",
                  "shape": "(5,3)"
                },
                "velocity": {
                  "type": "supplemental",
                  "dimension": "(orbitListLength,tripletxyz)",
                  "data_type": "float64",
                  "shape": "(5,3)"
                }
              },
              "secondary": {
                "time": {
                  "type": "dimension",
```

```
          "dimension": "(orbitListLength)",
          "data_type": "float64",
          "shape": "(5)"
        },
        "position": {
          "type": "supplemental",
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(5,3)"
        },
        "velocity": {
          "type": "supplemental",
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(5,3)"
        }
      }
    },
    "attitude": {
      "reference": {
        "time": {
          "type": "dimension",
          "dimension": "(attitudeListLength)",
          "data_type": "float64",
          "shape": "(17)"
        },
        "quaternions": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,quaternions)",
          "data_type": "float64",
          "shape": "(17,4)"
        },
        "eulerAngles": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(17,3)"
        }
      },
      "secondary": {
        "time": {
          "type": "dimension",
          "dimension": "(attitudeListLength)",
          "data_type": "float64",
          "shape": "(17)"
        },
        "quaternions": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,quaternions)",
          "data_type": "float64",
          "shape": "(17,4)"
        },
```

```
              "eulerAngles": {
                "type": "supplemental",
                "dimension": "(attitudeListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(17,3)"
              }
            }
          },
          "geolocationGrid": {
            "coordinateY": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(11,17,17)"
            },
            "coordinateX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(11,17,17)"
            },
            "incidenceAngle": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(11,17,17)"
            },
            "losUnitVectorX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(11,17,17)"
            },
            "losUnitVectorY": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(11,17,17)"
            },
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(11,17,17)"
            },
            "alongTrackUnitVectorY": {
```

```
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(11,17,17)"
            },
            "elevationAngle": {
              "type": "cube",
              "dimension":
"(geolocationCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(11,17,17)"
            },
            "parallelBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(3,17,17)"
            },
            "perpendicularBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,geolocationCubeLength,geolocationCubeWidth)",
              "data_type": "float32",
              "shape": "(3,17,17)"
            },
            "slantRange": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeWidth)",
              "data_type": "float64",
              "shape": "(17)"
            },
            "zeroDopplerTime": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeLength)",
              "data_type": "float64",
              "shape": "(17)"
            },
            "heightAboveEllipsoid": {
              "type": "cube dimension",
              "dimension": "(geolocationCubeHeight)",
              "data_type": "float64",
              "shape": "(11)"
            }
          }
        }
      }
    }
  }
}
```

10.9. L2 GOFF

Metadata


```
{
  "granule": "L2_GeocodedPixelOffsets",
  "science": {
    "LSAR": {
      "identification": {
        "referenceAbsoluteOrbitNumber": "Absolute orbit number for the
reference RSLC",
        "secondaryAbsoluteOrbitNumber": "Absolute orbit number for the
secondary RSLC",
        "referenceIsJointObservation": "\"True\" if any portion of the
reference RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "secondaryIsJointObservation": "\"True\" if any portion of the
secondary RSLC was acquired in a joint observation mode (e.g., L-band
and S-band simultaneously), \"False\" otherwise",
        "trackNumber": "Track number",
        "frameNumber": "Frame number",
        "missionId": "Mission identifier",
        "processingCenter": "Data processing center",
        "productType": "Product type",
        "granuleId": "Unique granule identification name",
        "productDoi": "Digital Object Identifier (DOI) for the product",
        "productVersion": "Product version which represents the
structure of the product and the science content governed by the
algorithm, input data, and processing parameters",
        "productSpecificationVersion": "Product specification version
which represents the schema of this product",
        "lookDirection": "Look direction, either \"Left\" or \"Right\"",
        "orbitPassDirection": "Orbit direction, either \"Ascending\" or
\"Descending\"",
        "referenceZeroDopplerStartTime": "Azimuth start time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referenceZeroDopplerEndTime": "Azimuth stop time (in UTC) of
reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerStartTime": "Azimuth start time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "secondaryZeroDopplerEndTime": "Azimuth stop time (in UTC) of
secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
        "referencePlannedDatatakeId": "List of planned datatakes
included in the reference RSLC",
        "secondaryPlannedDatatakeId": "List of planned datatakes
included in the secondary RSLC",
        "referencePlannedObservationId": "List of planned observations
included in the reference RSLC ",
        "secondaryPlannedObservationId": "List of planned observations
included in the secondary RSLC ",
        "referenceListOfObservationModes": "List of observation modes of
the L0B granules used to generate the reference RSLC (one mode per
L0B)",
```

```
        "secondaryListOfObservationModes": "List of observation modes of
the L0B granules used to generate the secondary RSLC (one mode per
L0B)",
        "isUrgentObservation": "Flag indicating if observation is
nominal (\"False\") or urgent (\"True\")",
        "listOfFrequencies": "List of frequency layers available in the
product",
        "diagnosticModeFlag": "Indicates if the radar operation mode is
a diagnostic mode (1-2) or DBFed science (0): 0, 1, or 2",
        "productLevel": "Product level. \"L0A\": Unprocessed instrument
data; \"L0B\": Reformatted, unprocessed instrument data; \"L1\":
Processed instrument data in radar coordinates system; and \"L2\":
Processed instrument data in geocoded coordinates system",
        "isGeocoded": "Flag to indicate if the product data is in the
radar geometry (\"False\") or in the map geometry (\"True\")",
        "boundingPolygon": "OGR compatible WKT representing the bounding
polygon of the image. Horizontal coordinates are WGS84 longitude
followed by latitude (both in degrees), and the vertical coordinate is
the height above the WGS84 ellipsoid in meters. The first point
corresponds to the start-time, near-range radar coordinate, and the
perimeter is traversed in counterclockwise order on the map. This means
the traversal order in radar coordinates differs for left-looking and
right-looking sensors. The polygon includes the four corners of the
radar grid, with equal numbers of points distributed evenly in radar
coordinates along each edge",
        "processingDateTime": "Processing date and time (in UTC) in the
format YYYY-mm-ddTHH:MM:SS",
        "radarBand": "Acquired frequency band, either \"L\" or \"S\"",
        "platformName": "Name of the platform used to collect the remote
sensing data provided in this product",
        "instrumentName": "Name of the instrument used to collect the
remote sensing data provided in this product",
        "processingType": "Processing pipeline used to generate this
granule. \"Nominal\": standard production system; \"Urgent\": time-
sensitive processing in response to urgent response events; \"Custom\":
user-initiated processing outside the nominal production system",
        "isDithered": "\"True\" if the pulse timing was varied
(dithered) during acquisition, \"False\" otherwise",
        "isMixedMode": "\"True\" if this product is generated from
reference and secondary RSLCs with different range bandwidths, \"False\"
otherwise",
        "compositeReleaseId": "Unique version identifier of the science
data production system",
        "isFullFrame": "\"True\" if the product fully covers a NISAR
frame, \"False\" if partial coverage"
      },
      "GOFF": {
        "identification": {
          "staticLayersDataAccess": "Location of the static layers
product associated with this product (URL or DOI)"
        },
        "grids": {
```

```
          "frequencyA": {
            "listOfPolarizations": "List of processed polarization
layers with frequency A",
            "centerFrequency": "Center frequency of the processed image
in hertz",
            "listOfLayers": "List of pixel offset layers",
            "pixelOffsets": {
              "mask": "Combination of water mask and a mask of subswaths
of valid samples in the reference RSLC and geometrically-coregistered
secondary RSLC. Each pixel value is a three-digit number: the most
significant digit represents the water flag of that pixel the reference
RSLC, where 1 is water and 0 is non-water; the second digit represents
the subswath number of that pixel in the reference RSLC; the least
significant digit represents the subswath number of that pixel in the
secondary RSLC. A value of 0 in either subswath digit indicates an
invalid sample in the corresponding RSLC",
              "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
              "yCoordinateSpacing": "Nominal spacing in meters between
consecutive lines",
              "xCoordinateSpacing": "Nominal spacing in meters between
consecutive pixels",
              "HH": {
                "layer1": {
                  "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                  "yCoordinateSpacing": "Nominal spacing in meters
between consecutive lines",
                  "xCoordinateSpacing": "Nominal spacing in meters
between consecutive pixels",
                  "xCoordinates": "X coordinates in specified
projection",
                  "yCoordinates": "Y coordinates in specified
projection",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                  "snr": "Pixel offsets signal-to-noise ratio"
                },
                "layer2": {
                  "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                  "yCoordinateSpacing": "Nominal spacing in meters
```

```
between consecutive lines",
                  "xCoordinateSpacing": "Nominal spacing in meters
between consecutive pixels",
                  "xCoordinates": "X coordinates in specified
projection",
                  "yCoordinates": "Y coordinates in specified
projection",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                  "snr": "Pixel offsets signal-to-noise ratio"
                },
                "layer3": {
                  "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                  "yCoordinateSpacing": "Nominal spacing in meters
between consecutive lines",
                  "xCoordinateSpacing": "Nominal spacing in meters
between consecutive pixels",
                  "xCoordinates": "X coordinates in specified
projection",
                  "yCoordinates": "Y coordinates in specified
projection",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                  "snr": "Pixel offsets signal-to-noise ratio"
                }
              },
              "VV": {
                "layer1": {
                  "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                  "yCoordinateSpacing": "Nominal spacing in meters
```

```
between consecutive lines",
                  "xCoordinateSpacing": "Nominal spacing in meters
between consecutive pixels",
                  "xCoordinates": "X coordinates in specified
projection",
                  "yCoordinates": "Y coordinates in specified
projection",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                  "snr": "Pixel offsets signal to noise ratio"
                },
                "layer2": {
                  "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                  "yCoordinateSpacing": "Nominal spacing in meters
between consecutive lines",
                  "xCoordinateSpacing": "Nominal spacing in meters
between consecutive pixels",
                  "xCoordinates": "X coordinates in specified
projection",
                  "yCoordinates": "Y coordinates in specified
projection",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                  "snr": "Pixel offsets signal-to-noise ratio"
                },
                "layer3": {
                  "projection": "Product map grid projection: EPSG code,
with additional projection information as HDF5 Attributes",
                  "yCoordinateSpacing": "Nominal spacing in meters
between consecutive lines",
                  "xCoordinateSpacing": "Nominal spacing in meters
```

```
between consecutive pixels",
                  "xCoordinates": "X coordinates in specified
projection",
                  "yCoordinates": "Y coordinates in specified
projection",
                  "alongTrackOffset": "Raw (unculled, unfiltered) along-
track pixel offsets",
                  "slantRangeOffset": "Raw (unculled, unfiltered) slant
range pixel offsets",
                  "correlationSurfacePeak": "Normalized correlation
surface peak",
                  "crossOffsetVariance": "Off-diagonal term of the pixel
offsets covariance matrix",
                  "slantRangeOffsetVariance": "Slant range pixel offsets
variance",
                  "alongTrackOffsetVariance": "Along-track pixel offsets
variance",
                  "snr": "Pixel offsets signal-to-noise ratio"
                }
              },
              "xCoordinates": "X coordinates in specified projection",
              "yCoordinates": "Y coordinates in specified projection"
            }
          }
        },
        "metadata": {
          "processingInformation": {
            "parameters": {
              "runConfigurationContents": "Contents of the run
configuration file with parameters used for processing",
              "reference": {
                "referenceTerrainHeight": "Reference terrain height as a
function of map coordinates for reference RSLC",
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the reference RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the reference RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "isMixedMode": "\"True\" if reference RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
                "frequencyA": {
                  "slantRangeStart": "Slant range start distance for the
reference RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the reference RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the reference RSLC",
                  "slantRangeSpacing": "Slant range spacing of reference
RSLC",
                  "zeroDopplerTimeSpacing": "Time interval in the along-
```

```
track direction for reference RSLC raster layers",
                  "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the reference RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                  "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the reference RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                  "rangeBandwidth": "Processed slant range bandwidth for
reference RSLC",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
reference RSLC",
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A"
                }
              },
              "secondary": {
                "rfiMitigationApplied": "Flag to indicate if radio
frequency interference (RFI) mitigation was applied during the
generation of the secondary RSLC",
                "rfiMitigation": "Algorithm used for radio frequency
interference (RFI) mitigation in the secondary RSLC, either \"ST-EVD\"
or \"FDNF\" (or \"disabled\" if no RFI mitigation was applied)",
                "isMixedMode": "\"True\" if secondary RSLC is a
composite of data collected in multiple radar modes, \"False\"
otherwise",
                "referenceTerrainHeight": "Reference terrain height as a
function of map coordinates for secondary RSLC",
                "frequencyA": {
                  "dopplerCentroid": "2D LUT of Doppler centroid for
frequency A",
                  "slantRangeStart": "Slant range start distance for the
secondary RSLC",
                  "numberOfRangeSamples": "Number of slant range samples
for each azimuth line within the secondary RSLC",
                  "numberOfAzimuthLines": "Number of azimuth lines
within the secondary RSLC",
                  "slantRangeSpacing": "Slant range spacing of secondary
RSLC",
                  "zeroDopplerTimeSpacing": "Time interval in the along-
track direction for secondary RSLC raster layers",
                  "zeroDopplerStartTime": "Azimuth start time (in UTC)
of the secondary RSLC product in the format YYYY-mm-
ddTHH:MM:SS.sssssssss",
                  "zeroDopplerEndTime": "Azimuth stop time (in UTC) of
the secondary RSLC product in the format YYYY-mm-ddTHH:MM:SS.sssssssss",
                  "rangeBandwidth": "Processed slant range bandwidth for
secondary RSLC",
                  "azimuthBandwidth": "Processed azimuth bandwidth for
secondary RSLC"
                }
              },
              "common": {
                "frequencyA": {
```

```
                "dopplerCentroid": "2D LUT of common Doppler centroid
between reference and secondary RSLCs",
                "dopplerBandwidth": "Common Doppler bandwidth between
reference and secondary RSLCs"
              }
            },
            "pixelOffsets": {
              "frequencyA": {
                "rangeBandwidth": "Processed slant range bandwidth for
frequency A pixel offsets layers",
                "azimuthBandwidth": "Processed azimuth bandwidth for
frequency A pixel offsets layers",
                "correlationSurfaceOversampling": "Oversampling factor
of the cross-correlation surface",
                "margin": "Margin in pixels around reference RSLC
edges excluded during cross-correlation computation",
                "slantRangeStartPixel": "Reference RSLC start pixel in
slant range",
                "alongTrackStartPixel": "Reference RSLC start pixel in
along-track",
                "slantRangeSkipWindowSize": "Slant range cross-
correlation skip window size in pixels",
                "alongTrackSkipWindowSize": "Along-track cross-
correlation skip window size in pixels",
                "layer1": {
                  "alongTrackWindowSize": "Along-track cross-
correlation window size in pixels",
                  "slantRangeWindowSize": "Slant range cross-
correlation window size in pixels",
                  "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                  "slantRangeSearchWindowSize": "Slant range cross-
correlation search window size in pixels"
                },
                "layer2": {
                  "alongTrackWindowSize": "Along-track cross-
correlation window size in pixels",
                  "slantRangeWindowSize": "Slant range cross-
correlation window size in pixels",
                  "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                  "slantRangeSearchWindowSize": "Slant range cross-
correlation search window size in pixels"
                },
                "layer3": {
                  "alongTrackWindowSize": "Along-track cross-
correlation window size in pixels",
                  "slantRangeWindowSize": "Slant range cross-
correlation window size in pixels",
                  "alongTrackSearchWindowSize": "Along-track cross-
correlation search window size in pixels",
                  "slantRangeSearchWindowSize": "Slant range cross-
```

```
correlation search window size in pixels"
                  }
                }
              },
              "geocoding": {
                "rangeIonosphericCorrectionApplied": "Flag to indicate
if the range ionospheric correction is applied to improve geolocation",
                "azimuthIonosphericCorrectionApplied": "Flag to indicate
if the azimuth ionospheric correction is applied to improve
geolocation",
                "hydrostaticTroposphericCorrectionApplied": "Flag to
indicate if the hydrostatic tropospheric correction is applied to
improve geolocation",
                "wetTroposphericCorrectionApplied": "Flag to indicate if
the wet tropospheric correction is applied to improve geolocation"
              }
            },
            "algorithms": {
              "softwareVersion": "Software version used for processing",
              "geocoding": {
                "complexGeocodingInterpolation": "Geocoding
interpolation algorithm for complex-valued datasets",
                "integerGeocodingInterpolation": "Geocoding
interpolation algorithm for integer datasets",
                "floatingGeocodingInterpolation": "Geocoding
interpolation algorithm for floating point datasets",
                "demInterpolation": "DEM interpolation algorithm"
              },
              "coregistration": {
                "coregistrationMethod": "RSLC coregistration method",
                "geometryCoregistration": "Geometry coregistration
algorithm",
                "resampling": "Secondary RSLC resampling algorithm"
              },
              "crossCorrelation": {
                "layer1": {
                  "crossCorrelationAlgorithm": "Cross-correlation
algorithm for layer 1"
                },
                "layer2": {
                  "crossCorrelationAlgorithm": "Cross-correlation
algorithm for layer 2"
                },
                "layer3": {
                  "crossCorrelationAlgorithm": "Cross-correlation
algorithm for layer 3"
                }
              }
            },
            "inputs": {
              "l1ReferenceSlcGranules": "List of input reference L1 RSLC
products used",
```

```
            "l1SecondarySlcGranules": "List of input secondary L1 RSLC
products used",
            "orbitFiles": "List of input orbit files used",
            "configFiles": "List of input config files used",
            "demSource": "Description of the input digital elevation
model (DEM)",
            "waterMaskSource": "Description of the input water mask"
          }
        },
        "orbit": {
          "temporalBaseline": "Time interval between reference and
secondary RSLCs",
          "reference": {
            "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
            "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
            "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
            "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
          },
          "secondary": {
            "interpMethod": "Orbit interpolation method, either
\"Hermite\" or \"Legendre\"",
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to position and velocity records",
            "position": "Position vector record. This record contains
the platform position data with respect to WGS84 G1762 reference frame",
            "velocity": "Velocity vector record. This record contains
the platform velocity data with respect to WGS84 G1762 reference frame",
            "orbitType": "Orbit product type, either \"FOE\", \"NOE\",
\"MOE\", \"POE\", or \"Custom\", where \"FOE\" stands for Forecast Orbit
Ephemeris, \"NOE\" is Near real-time Orbit Ephemeris, \"MOE\" is Medium
precision Orbit Ephemeris, and \"POE\" is Precise Orbit Ephemeris"
          }
        },
        "attitude": {
          "reference": {
            "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
            "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
            "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
            "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
          },
```

```
            "secondary": {
              "time": "Time vector record. This record contains the time
since UTC epoch corresponding to attitude and quaternion records",
              "quaternions": "Attitude quaternions (q0, q1, q2, q3)",
              "eulerAngles": "Attitude Euler angles (roll, pitch, yaw)",
              "attitudeType": "Attitude type, either \"FRP\", \"NRP\",
\"PRP, or \"Custom\", where \"FRP\" stands for Forecast Radar Pointing,
\"NRP\" is Near Real-time Pointing, and \"PRP\" is Precise Radar
Pointing"
            }
          },
          "radarGrid": {
            "referenceSlantRange": "Slant range of the reference RSLC in
meters",
            "referenceZeroDopplerAzimuthTime": "Zero Doppler azimuth
time in seconds since UTC epoch of the reference RSLC image",
            "xCoordinates": "X coordinates in specified projection",
            "yCoordinates": "Y coordinates in specified projection",
            "incidenceAngle": "Incidence angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the target
height",
            "losUnitVectorX": "East component of the line-of-sight (LOS)
unit vector, defined from the target to the sensor, expressed in the
east-north-up (ENU) coordinate system with its origin at the target
location",
            "losUnitVectorY": "North component of the line-of-sight
(LOS) unit vector, defined from the target to the sensor, expressed in
the east-north-up (ENU) coordinate system with its origin at the target
location",
            "alongTrackUnitVectorX": "East component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "alongTrackUnitVectorY": "North component of the along-track
unit vector at the target location, expressed in the east-north-up (ENU)
coordinate system and projected onto the horizontal plane (i.e.,
excluding the up component)",
            "elevationAngle": "Elevation angle is defined as the angle
between the LOS vector and the normal to the ellipsoid at the sensor",
            "groundTrackVelocity": "Absolute value of the platform
velocity scaled at the target height",
            "secondaryZeroDopplerAzimuthTime": "Zero Doppler azimuth
time in seconds since UTC epoch of the secondary RSLC image",
            "secondarySlantRange": "Slant range of the secondary RSLC in
meters",
            "parallelBaseline": "Parallel component of the InSAR
baseline",
            "perpendicularBaseline": "Perpendicular component of the
InSAR baseline",
            "projection": "Product map grid projection: EPSG code, with
additional projection information as HDF5 Attributes",
            "heightAboveEllipsoid": "Height values above WGS84 Ellipsoid
```

```
corresponding to the radar grid"
          }
        }
      }
    }
  }
}
```

Data

```
{
  "granule": "L2_GeocodedPixelOffsets",
  "dimension": {
    "attitudeListLength": 11,
    "dopplerCentroidLength": 19,
    "dopplerCentroidWidth": 23,
    "offsetLength": 19,
    "offsetWidth": 17,
    "orbitListLength": 3,
    "quaternions": 4,
    "radarCubeHeight": 5,
    "radarCubeLength": 11,
    "radarCubeWidth": 11,
    "tripletxyz": 3,
    "twoLayersCubeHeight": 23
  },
  "science": {
    "LSAR": {
      "GOFF": {
        "grids": {
          "frequencyA": {
            "pixelOffsets": {
              "mask": {
                "type": "supplemental",
                "dimension": "(offsetLength,offsetWidth)",
                "data_type": "ubyte",
                "shape": "(19,17)"
              },
              "xCoordinates": {
                "type": "dimension",
                "dimension": "(offsetWidth)",
                "data_type": "float64",
                "shape": "(17)"
              },
              "yCoordinates": {
                "type": "dimension",
                "dimension": "(offsetLength)",
                "data_type": "float64",
                "shape": "(19)"
              },
              "HH": {
                "layer1": {
```

```
    "xCoordinates": {
      "type": "dimension",
      "dimension": "(offsetWidth)",
      "data_type": "float64",
      "shape": "(17)"
    },
    "yCoordinates": {
      "type": "dimension",
      "dimension": "(offsetLength)",
      "data_type": "float64",
      "shape": "(19)"
    },
    "alongTrackOffset": {
      "type": "dataset",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "slantRangeOffset": {
      "type": "dataset",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "correlationSurfacePeak": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "crossOffsetVariance": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "slantRangeOffsetVariance": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "alongTrackOffsetVariance": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "snr": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
```

```
        "shape": "(19,17)"
      }
    },
    "layer2": {
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(offsetWidth)",
        "data_type": "float64",
        "shape": "(17)"
      },
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(offsetLength)",
        "data_type": "float64",
        "shape": "(19)"
      },
      "alongTrackOffset": {
        "type": "dataset",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "slantRangeOffset": {
        "type": "dataset",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "correlationSurfacePeak": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "crossOffsetVariance": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "slantRangeOffsetVariance": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "alongTrackOffsetVariance": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
```

```
      "snr": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      }
    },
    "layer3": {
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(offsetWidth)",
        "data_type": "float64",
        "shape": "(17)"
      },
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(offsetLength)",
        "data_type": "float64",
        "shape": "(19)"
      },
      "alongTrackOffset": {
        "type": "dataset",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "slantRangeOffset": {
        "type": "dataset",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "correlationSurfacePeak": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "crossOffsetVariance": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "slantRangeOffsetVariance": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "alongTrackOffsetVariance": {
        "type": "supplemental",
```

```
          "dimension": "(offsetLength,offsetWidth)",
          "data_type": "float32",
          "shape": "(19,17)"
        },
        "snr": {
          "type": "supplemental",
          "dimension": "(offsetLength,offsetWidth)",
          "data_type": "float32",
          "shape": "(19,17)"
        }
      }
    },
    "VV": {
      "layer1": {
        "xCoordinates": {
          "type": "dimension",
          "dimension": "(offsetWidth)",
          "data_type": "float64",
          "shape": "(17)"
        },
        "yCoordinates": {
          "type": "dimension",
          "dimension": "(offsetLength)",
          "data_type": "float64",
          "shape": "(19)"
        },
        "alongTrackOffset": {
          "type": "dataset",
          "dimension": "(offsetLength,offsetWidth)",
          "data_type": "float32",
          "shape": "(19,17)"
        },
        "slantRangeOffset": {
          "type": "dataset",
          "dimension": "(offsetLength,offsetWidth)",
          "data_type": "float32",
          "shape": "(19,17)"
        },
        "correlationSurfacePeak": {
          "type": "supplemental",
          "dimension": "(offsetLength,offsetWidth)",
          "data_type": "float32",
          "shape": "(19,17)"
        },
        "crossOffsetVariance": {
          "type": "supplemental",
          "dimension": "(offsetLength,offsetWidth)",
          "data_type": "float32",
          "shape": "(19,17)"
        },
        "slantRangeOffsetVariance": {
          "type": "supplemental",
```

```
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "alongTrackOffsetVariance": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "snr": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    }
  },
  "layer2": {
    "xCoordinates": {
      "type": "dimension",
      "dimension": "(offsetWidth)",
      "data_type": "float64",
      "shape": "(17)"
    },
    "yCoordinates": {
      "type": "dimension",
      "dimension": "(offsetLength)",
      "data_type": "float64",
      "shape": "(19)"
    },
    "alongTrackOffset": {
      "type": "dataset",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "slantRangeOffset": {
      "type": "dataset",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "correlationSurfacePeak": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
      "shape": "(19,17)"
    },
    "crossOffsetVariance": {
      "type": "supplemental",
      "dimension": "(offsetLength,offsetWidth)",
      "data_type": "float32",
```

```
        "shape": "(19,17)"
      },
      "slantRangeOffsetVariance": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "alongTrackOffsetVariance": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "snr": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      }
    },
    "layer3": {
      "xCoordinates": {
        "type": "dimension",
        "dimension": "(offsetWidth)",
        "data_type": "float64",
        "shape": "(17)"
      },
      "yCoordinates": {
        "type": "dimension",
        "dimension": "(offsetLength)",
        "data_type": "float64",
        "shape": "(19)"
      },
      "alongTrackOffset": {
        "type": "dataset",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "slantRangeOffset": {
        "type": "dataset",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
      "correlationSurfacePeak": {
        "type": "supplemental",
        "dimension": "(offsetLength,offsetWidth)",
        "data_type": "float32",
        "shape": "(19,17)"
      },
```

```
                    "crossOffsetVariance": {
                      "type": "supplemental",
                      "dimension": "(offsetLength,offsetWidth)",
                      "data_type": "float32",
                      "shape": "(19,17)"
                    },
                    "slantRangeOffsetVariance": {
                      "type": "supplemental",
                      "dimension": "(offsetLength,offsetWidth)",
                      "data_type": "float32",
                      "shape": "(19,17)"
                    },
                    "alongTrackOffsetVariance": {
                      "type": "supplemental",
                      "dimension": "(offsetLength,offsetWidth)",
                      "data_type": "float32",
                      "shape": "(19,17)"
                    },
                    "snr": {
                      "type": "supplemental",
                      "dimension": "(offsetLength,offsetWidth)",
                      "data_type": "float32",
                      "shape": "(19,17)"
                    }
                  }
                }
              }
            }
          },
          "metadata": {
            "processingInformation": {
              "parameters": {
                "reference": {
                  "referenceTerrainHeight": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                    "data_type": "float32",
                    "shape": "(19,23)"
                  },
                  "frequencyA": {
                    "dopplerCentroid": {
                      "type": "supplemental",
                      "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                      "data_type": "float64",
                      "shape": "(19,23)"
                    }
                  }
                },
                "secondary": {
                  "referenceTerrainHeight": {
```

```
                  "type": "supplemental",
                  "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                  "data_type": "float32",
                  "shape": "(19,23)"
                },
                "frequencyA": {
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                    "data_type": "float64",
                    "shape": "(19,23)"
                  }
                }
              },
              "common": {
                "frequencyA": {
                  "dopplerCentroid": {
                    "type": "supplemental",
                    "dimension":
"(dopplerCentroidLength,dopplerCentroidWidth)",
                    "data_type": "float64",
                    "shape": "(19,23)"
                  }
                }
              }
            }
          },
          "orbit": {
            "reference": {
              "time": {
                "type": "dimension",
                "dimension": "(orbitListLength)",
                "data_type": "float64",
                "shape": "(3)"
              },
              "position": {
                "type": "supplemental",
                "dimension": "(orbitListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(3,3)"
              },
              "velocity": {
                "type": "supplemental",
                "dimension": "(orbitListLength,tripletxyz)",
                "data_type": "float64",
                "shape": "(3,3)"
              }
            },
            "secondary": {
              "time": {
```

```
          "type": "dimension",
          "dimension": "(orbitListLength)",
          "data_type": "float64",
          "shape": "(3)"
        },
        "position": {
          "type": "supplemental",
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(3,3)"
        },
        "velocity": {
          "type": "supplemental",
          "dimension": "(orbitListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(3,3)"
        }
      }
    },
    "attitude": {
      "reference": {
        "time": {
          "type": "dimension",
          "dimension": "(attitudeListLength)",
          "data_type": "float64",
          "shape": "(11)"
        },
        "quaternions": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,quaternions)",
          "data_type": "float64",
          "shape": "(11,4)"
        },
        "eulerAngles": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,tripletxyz)",
          "data_type": "float64",
          "shape": "(11,3)"
        }
      },
      "secondary": {
        "time": {
          "type": "dimension",
          "dimension": "(attitudeListLength)",
          "data_type": "float64",
          "shape": "(11)"
        },
        "quaternions": {
          "type": "supplemental",
          "dimension": "(attitudeListLength,quaternions)",
          "data_type": "float64",
          "shape": "(11,4)"
```

```
            },
            "eulerAngles": {
              "type": "supplemental",
              "dimension": "(attitudeListLength,tripletxyz)",
              "data_type": "float64",
              "shape": "(11,3)"
            }
          }
        },
        "radarGrid": {
          "referenceSlantRange": {
            "type": "cube",
            "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
            "data_type": "float64",
            "shape": "(5,11,11)"
          },
          "referenceZeroDopplerAzimuthTime": {
            "type": "cube",
            "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
            "data_type": "float64",
            "shape": "(5,11,11)"
          },
          "xCoordinates": {
            "type": "cube dimension",
            "dimension": "(radarCubeWidth)",
            "data_type": "float64",
            "shape": "(11)"
          },
          "yCoordinates": {
            "type": "cube dimension",
            "dimension": "(radarCubeLength)",
            "data_type": "float64",
            "shape": "(11)"
          },
          "incidenceAngle": {
            "type": "cube",
            "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
            "data_type": "float32",
            "shape": "(5,11,11)"
          },
          "losUnitVectorX": {
            "type": "cube",
            "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
            "data_type": "float32",
            "shape": "(5,11,11)"
          },
          "losUnitVectorY": {
            "type": "cube",
```

```
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "alongTrackUnitVectorX": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "alongTrackUnitVectorY": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "elevationAngle": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float32",
              "shape": "(5,11,11)"
            },
            "secondaryZeroDopplerAzimuthTime": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "secondarySlantRange": {
              "type": "cube",
              "dimension":
"(radarCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(5,11,11)"
            },
            "parallelBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
              "shape": "(23,11,11)"
            },
            "perpendicularBaseline": {
              "type": "cube",
              "dimension":
"(twoLayersCubeHeight,radarCubeLength,radarCubeWidth)",
              "data_type": "float64",
```

```
            "shape": "(23,11,11)"
          },
          "heightAboveEllipsoid": {
            "type": "cube dimension",
            "dimension": "(radarCubeHeight)",
            "data_type": "float64",
            "shape": "(5)"
          }
        }
      }
    }
  }
}
```


References

Brancato, V., Jung, J., & Huang, X. (2025d). *Level-1 Range Doppler Pixel Offsets* [Technical report].

Brancato, V., Jung, J., & Huang, X. (2025b). *Level-1 Range Doppler Unwrapped Interferogram* [Technical report].

Brancato, V., Jung, J., & Huang, X. (2025a). *Level-1 Range Doppler Wrapped Interferogram* [Technical report].

Brancato, V., Jung, J., & Huang, X. (2025e). *Level-2 Geocoded Pixel Offsets* [Technical report].

Brancato, V., Jung, J., Huang, X., & Fattahi, H. (2025c). *Level-2 Geocoded Unwrapped Interferogram* [Technical report].

Fattahi, H. (2025). *Level-2 Geocoded Single Look Complex* [Technical report].

Hawkins, B. (2025a). *Level-0B Radar Raw Signal Data and Calibration Product* [Technical report].

Hawkins, B. (2025b). *Level-1 Range Doppler Single Look Complex* [Technical report].

Holt, B. (2022). *NISAR L2 Sea Ice Motion Product* (Techreport No. Algorithm Theoretical Basis Document (Revision 4). https://gitlab.com/nisar-science-algorithms/cryosphere/sea-ice/rgps_calval/-/blob/main/docs/SeaIceMotion_ATBD_V4.docx

Huang, X., & Chan, S. (2025). *Level-3 Soil Moisture* [Technical report].

Joughin, I., Rignot, E., Forster, R., Gardner, A., & Scheuchl, B. (2022). *NISAR L3 Ice Sheets and Glacier Products* (Techreport No. Algorithm Theoretical Basis Document (Revision 2.1). https://gitlab.com/nisar-science-algorithms/cryosphere/land-ice/atbd/documentation/-/blob/main/ATBD_IceSheetVelocity-Version2.1.pdf

NISAR Algorithm Development Team. (2023). *NISAR L1 and L2 Algorithm Theoretical Basis Document* (Techreport No. D-95677 (Revision A). https://nisar.asf.earthdatacloud.nasa.gov/NISAR-SAMPLE-DATA/DOCS/NISAR_D-95677_NASA_L1_L2_ATBD_20231112_R3.4_w-sigs.pdf

NISAR Soil Moisture Science Team. (2023). *NISAR L3 Soil Moisture Data Product: Algorithm Theoretical Basis Document (ATBD)* (Techreport No. D-107679 (Revision 3.3). https://nisar.asf.earthdatacloud.nasa.gov/NISAR-SAMPLE-DATA/DOCS/NISAR_D-107679_L3SM_ATBD_R3.3_20230428_w-sigs.pdf

NISAR Solid Earth Team. (2022). *Algorithms to Validate NISAR L2 Coseismic, Transient and Secular Displacement Requirements* (Techreport No. Algorithm Theoretical Basis Document (Revision A). https://gitlab.com/nisar-science-algorithms/solid-earth/ATBD/-/blob/main/docs/SES_ATBD.pdf

Shiroma, G. (2025). *Level-2 Geocoded Polarimetric Covariance* [Technical report].

Siqueira, P., Chapman, B., Kelldorfer, J., Saatchi, S., & Simard, M. (2018). *NISAR Ecosystem L2 Products* (Techreport No. Algorithm Theoretical Basis Document). https://gitlab.com/nisar-science-algorithms/ecosystems/documentation/-/blob/main/NISAR_Ecosystems_Combined_ATBD.pdf