Surface Water and Ocean Topography Project

Algorithm Theoretical Basis Document
Long Name: Level 1B KaRIn Low Rate Interferogram Science Algorithm Software
Short Name: L1B_LR_INTF

Initial Release

Document Custodians:

Bryan Stiles JPL Algorithm Engineer	Date	Pierre Dubois CNES Algorithm Engineer	Date

Approved by:

Curtis Chen JPL Algorithm System Engineer	Date	Alejandro Bohe CNES Algorithm System Engineer	Date

Concurred by:

Lee Fu JPL Project Scientist	Date	Rosemary Morrow CNES Oceans Science Lead	Date

See Sect 1.4 for how to cite this document

National Aeronautics and Space Administration
Jet Propulsion Laboratory
California Institute of Technology

cnes
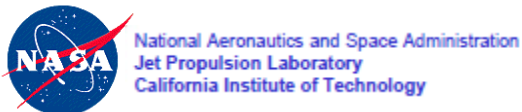
Contributing Authors

Name	Affiliation
Bryan Stiles	Jet Propulsion Laboratory
Pierre Dubois	Collecte Localisation Satellites

Science Team Reviewers

Name	Affiliation
J. T. Farrar	Woods Hole Oceanographic Institution
G. Dibarboure	Centre National d'Etudes Spatiales
B. Chapron	Ifremer, Plouzane, France
F. Nouguier	Ifremer, Plouzane, France
D. Vandemark	University of New Hampshire

Change Log

VERSION	DATE	SECTIONS CHANGED	REASON FOR CHANGE
Initial Release	2023-07-24	ALL	Initial Release Approved for public release (URS317367/CL#23-3293)

Table of Contents

Table of Figures

Table of Tables

List of TBC Items

Page	Section

List of TBD Items

Page	Section
34	3.4.5, Description of accuracy of power and phase standard deviation
38	3.6.5, Study to verify no systematic algorithm errors in sigma0 estimation
38	3.7.4, Algorithm to compute standard deviation of SNR

1 Introduction

1.1 Purpose

The purpose of this Algorithm Theoretical Basis Document (ATBD) is to describe the physical and mathematical basis for the science data processing algorithms that are used to generate the SWOT Level 1B Ka-band Radar Interferometer (KaRIn) low rate interferogram science data product (L1B_LR_INTF). Low rate (LR) data is obtained globally but used primarily to determine topography over the ocean where high rate data is unavailable due to downlink constraints. This data is "low rate" because synthetic aperture radar (SAR) image formation, interferogram formation, and multi-look averaging to decrease noise at the expense of resolution are performed prior to downlink by the KaRIn onboard processor (OBP), resulting in a lower downlink rate than would be obtained if raw time-sampled data were downlinked.

The algorithms described in this document are applied in the L1B_LR_INTF science algorithm software (SAS). The L1B_LR_INTF SAS performs Level 1 processing of the low rate telemetry data from the SWOT KaRIn system. This document describes the Level 1 processing steps and their corresponding algorithm theoretical basis.

1.2 Scope

The scope of this document is to:
1. Identify the list of primary functions that compose the processing steps within the L1B_LR_INTF SAS and their flow. These functions are broken down by the primary functional steps involved in the processing.
2. Describe the purpose of each of the functions.
3. Describe the input data to each function.
4. Describe the output data from each function.
5. Describe the mathematical basis of the algorithm in each function.
6. Describe the expected accuracy and/or limitations of the algorithm in each function.
7. Provide the relevant references for the algorithms described in this document.

This document is not intended to cover onboard algorithms or the algorithms used to determine onboard parameters.

1.3 Document Organization

Section 1 provides the purpose and scope of this document.

Section 2 provides the background and context of the algorithms described in this document and the functional flow of the primary functions (e.g., block diagram).

Section 3 provides the algorithm description for each of the functions shown in the block diagram, including input data, output data, mathematical basis, and expected accuracy.

Section 4 provides references for the algorithms described in this document.

Appendix A provides a listing of the acronyms used in this document.

Appendix B provides details on the integration performed by SimulatePhaseBias

Appendix C provides mathematical derivations and technical details.

1.4 Citing This Document

Please cite this document as follows:

> JPL D-105501, "SWOT Algorithm Theoretical Basis Document: Level 1B KaRIn Low Rate Interferogram (L1B_LR_INTF) Science Algorithm Software," Jet Propulsion Laboratory Internal Document, 2023.

2 Overview

2.1 Background and Context

The Surface Water and Ocean Topography (SWOT) mission is a partnership between two communities, physical oceanography and hydrology, to share high vertical accuracy topography data produced by the payload, whose principal instrument is the Ka-band Radar Interferometer (KaRIn). The details of SWOT mission objectives and requirements can be found in the SWOT Science Requirements Document [1].

This document describes the Level 1 processing steps that are used to generate the L1B_LR_INTF science data product from downlinked low rate telemetry after L0 processing (initial processing to unpack the KaRIn telemetry from the downlink format). The downlinked telemetry is the output of the OBP [2]. In addition to downlinked telemetry, L1B_LR_INTF processing also requires several other inputs, including calibration parameters, spacecraft ephemeris and attitude, and geophysical models.

The OBP operates on raw, time-sampled radar echo data to generate three-dimensional maps of the power received by each antenna and the complex interferogram made by comparing the echoes of the two antennas. The three dimensions are along-track distance, cross-track distance, and Doppler beam. The *unnormalized interferogram* is formed by multiplying each complex SAR pixel value (voltage) from one receiving antenna by the complex conjugate of the corresponding SAR pixel from the other antenna. The *normalized interferogram* is formed by dividing the unnormalized interferogram by the geometric mean of the power from both antennas. After spatial averaging, the magnitude of the normalized interferogram is an estimate of the *correlation coefficient*, and the phase is the *interferometric phase* used by the downstream processing to determine the sea surface height. Below, we use the term "interferogram" without a modifier to refer to the unnormalized interferogram. The term "interferogram" may also be used in contexts where normalization is irrelevant, such as when referring to the interferogram phase, which is independent of normalization.

The LR interferogram sampling is largely determined by the OBP. For each Doppler beam formed by the OBP, the sample locations in downlinked telemetry and in the L1B_LR_INTF product [3] are the intersections of the response functions for each beam with a reference surface used during processing. This results in a spatial posting of approximately 250 m in both the cross-track and along-track directions for a given beam. Each beam has its own sampling grid. The width of the filters in the onboard processing used to produce these pixels is approximately 1 km by 1 km with approximate half-power widths (resolution) of 500 m by 500 m. The ground processing to create the L1B_LR_INTF product does not alter the resolution of the OBP output.

There are nine different Doppler beams that are formed from OBP synthetic-aperture processing of cotemporaneous data. The data acquired from each beam is a contiguous, slightly irregular 2-D image. Data from different beams with the same along-track index are cotemporaneous, but not spatially aligned. For this reason, it is more correct to think of the along track dimension as time instead of spatial distance.

The beams are numbered 1–9. The beam orientations from the spacecraft are offset in angle, so the beam intersections with the Earth's surface are offset in space (primarily in the along-track direction). Nominally, Beam 5 is the beam most closely aligned with the peak of the antenna gain in the azimuth direction. The spatial samples from Beams 1 and 9 are

approximately 600-m removed, in opposite directions, in the along-track direction from the corresponding sample from Beam 5. Beam 1 is the aftmost beam, and samples of successively numbered beams are located further forward along the direction of spacecraft motion. This is true for both the left and right half swaths, independent of the yaw state of the spacecraft.

Figure 1 illustrates the 2-D interferograms for each beam, showing only Beams 1, 5 and 9 for simplicity. The offsets of the 2-D interferograms for each beam in the cross-track direction are exaggerated for visual clarity (the illustration is not to scale). Samples from different beams that were acquired at the same time are offset spatially; equivalently, different beams pass over the same point on the Earth surface at different times.

Figure **2** illustrates at the pixel level the sample locations from different beams for a given instant in time relative to the resolution of each sample. The numbers denote beam numbers; the locations of the numbers in the figure indicate the center of the sample for that beam.
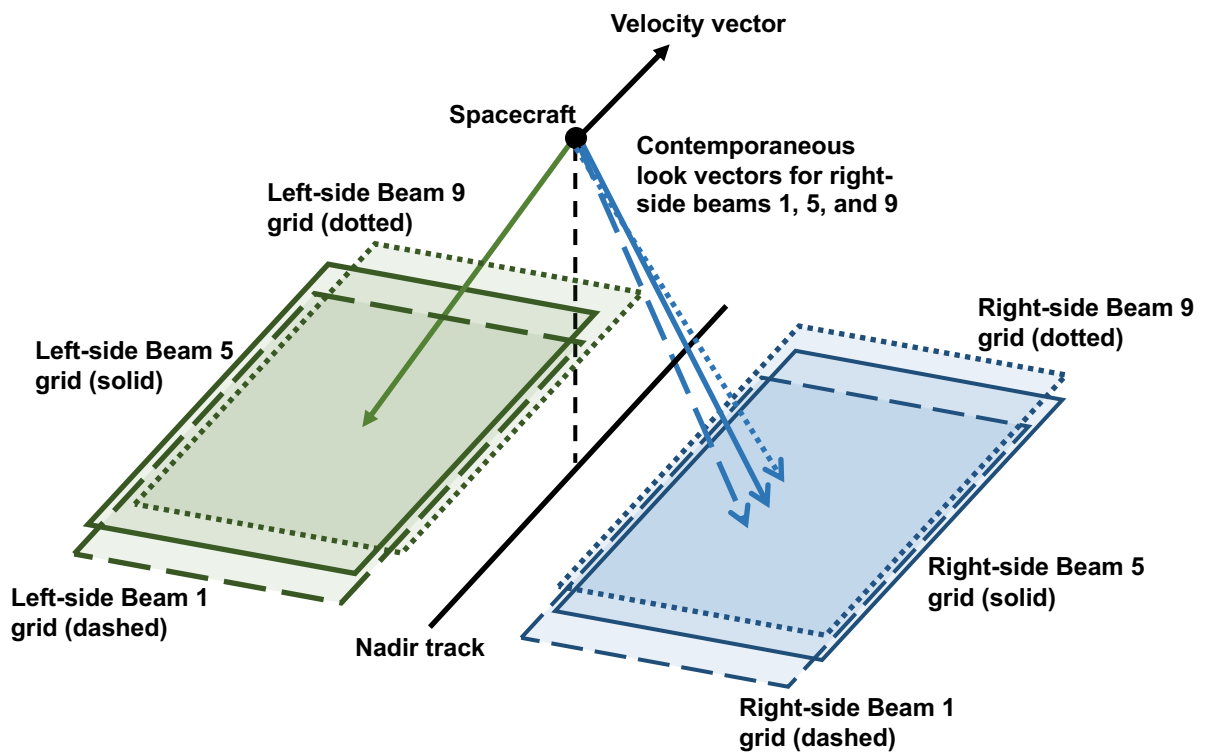
Figure 1 Illustration of the interferogram image geometry for each beam and each side, showing only Beams 1, 5 and 9 for simplicity. The offsets of the 2-D interferograms for each beam in the cross-track direction are exaggerated for visual clarity (the illustration is not to scale). Samples from different beams that were acquired at the same time are offset spatially; equivalently, different beams pass over the same point on the Earth's surface at different times.
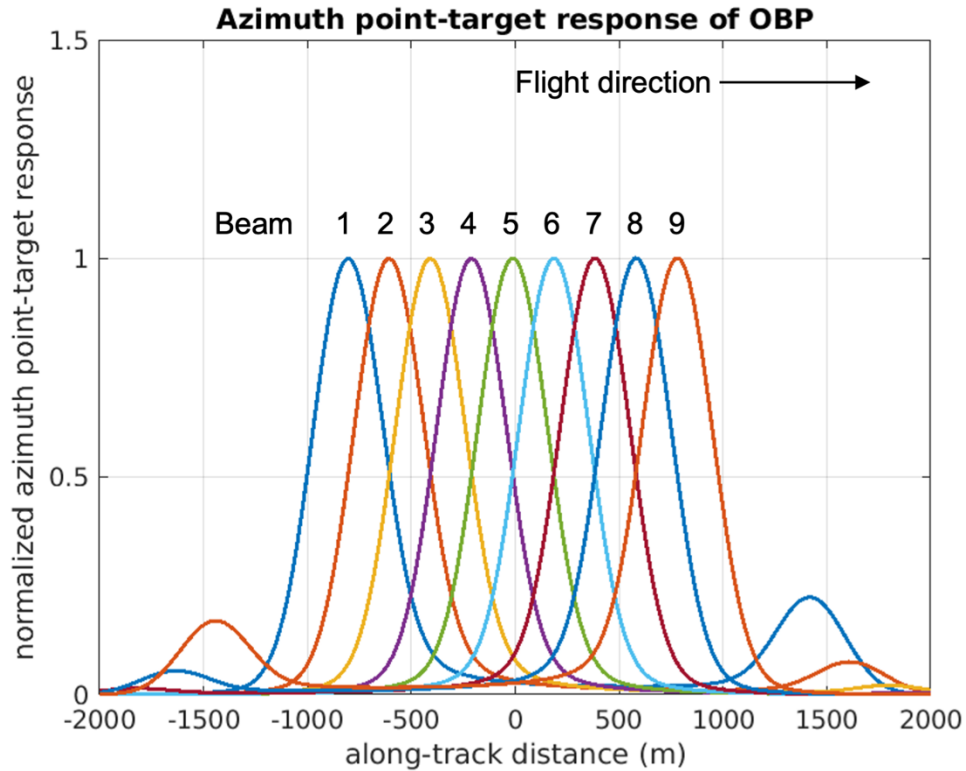
Figure 2 Spatial arrangement and along-track point-target responses of the 9 beams. The numbers above the curves denote beam numbers; the locations of the numbers in the figure indicate the center of the sample for that beam. The beams were all collected at the same moment in time (same data record in telemetry, same sensor line). The specific shape varies somewhat based upon instrument viewing geometry, antenna pattern, and wave spectra.

SWOT height reconstruction equations are derived based upon the difference between the two path lengths from a point target on the ground to each of the two receiving antennas. The antennas are aligned so that this difference in path length maps in a useful manner to surface height. When the two received signals are correlated after SAR image formation, the differential phase between the signals from the two antennas can be used to estimate the path length difference, which, along with the range and the Doppler of the target, can be used to solve for the location of the target in 3-D space.

The main function of the L1B_LR_INTF processor is to correct systematic biases in the interferometric phase that, if left uncorrected, would produce errors in the measured sea surface height (SSH). The term "systematic bias" does not imply that these quantities are constant over all space and time but rather that they are deterministic. In fact, the correction applied to the phase varies with beam, cross track index, along track index, spacecraft attitude, and with variations in parameter values used by the flight software and the OBP. The primary contributor to the systematic phase bias is the fact that the OBP SAR processing results in relatively coarse along-track resolution compared to the sensitivity of the interferometric phase to along-track variations in the geometry.

For SAR imagery at sufficiently fine along-track resolution, an individual pixel is sufficiently constrained in range and Doppler so that one can approximate the height

reconstruction as if all the echo energy from the pixel came from a single point on the surface. For the coarse-resolution SAR pixels constructed by the OBP, however, the single-point height reconstruction approximation yields maximal errors of several tenths of a radian in phase (0.5-2 meters in height) as depicted in Figure 3.
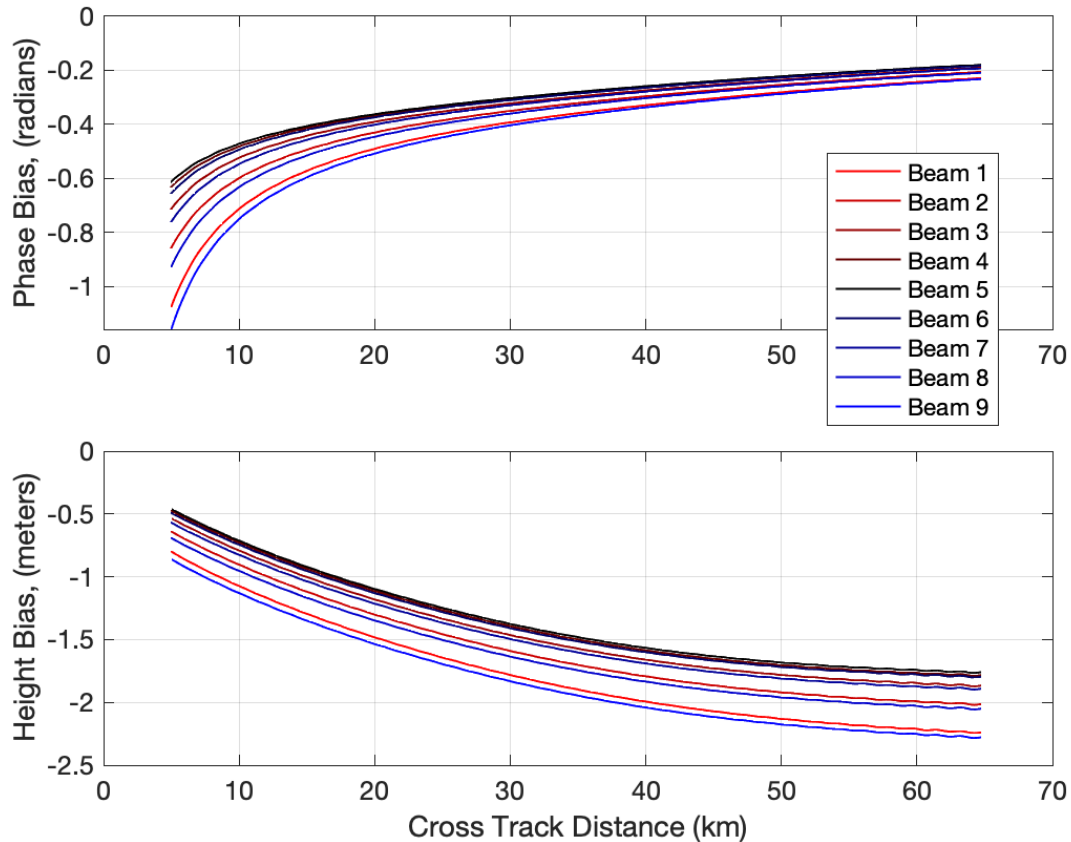
Figure 3 Systematic Phase Bias and Resultant Height Error if Not Corrected. The top panel shows the phase bias in radians for all nine beams, where the lowermost red and blue curves correspond to beams 1 and 9 respectively, and the uppermost black curve corresponds to beam 5. The bottom panel shows the resultant height error that would occur if the phase bias were not corrected. The values depicted are from a simulation of the mean sea surface with no waves as viewed from the right side of the KaRIn swath with ideal spacecraft attitude and antenna pointing.

The cause of the phase bias is illustrated by Figure 4 which shows the variation in phase over a region that is approximately 10-km wide in both cross track and along track. The x-axis is the cross-track distance with the spacecraft nadir point at zero. The y-axis is the along-track distance, also with nadir at zero. The boundaries of the colored region are iso-range and iso-Doppler contours, which would result in a shape that approaches rectangular for small spatial extents, especially in the far-range part of the swath, but would diverge substantially from rectangular for large extents in the near-range (near nadir) part of the swath. Individual pixels would exhibit similar behavior, though a large region is shown here to better illustrate the effect, which would be hard to see visually for a single 500-m resolution pixel. During onboard processing the phase of each pixel is *flattened* by removing the phase at the mean range, zero

azimuth point in the pixel. For the large "pixel" case shown in Figure 4 after flattening, the average phase of the 10-km by 10-km region is 11 radians, because the illustrated pixel is not centered at zero azimuth (e.g. it is not beam 5) and the low phase and high phase portions of the pixel are not equal in area. The white 'o' symbol indicates the location at which the flattening is performed in the OBP. Here the phase is 0. The white '+' symbol indicates the weighted center (cf. reference location) of the pixel. Here the phase is 11 radians, the same as the average phase for the region. For the SWOT geometry, increasing interferometric phase (difference in range between the two antennas) aligns closely with cross track distance for all azimuths, but range does not align so closely. Figure 4 is a simplified case. To accurately compute the phase bias, it is necessary to include other complicating factors such variations in phase due to antenna patterns from the two antennas and the precise spatial weighting applied by the SAR pixel formation process.
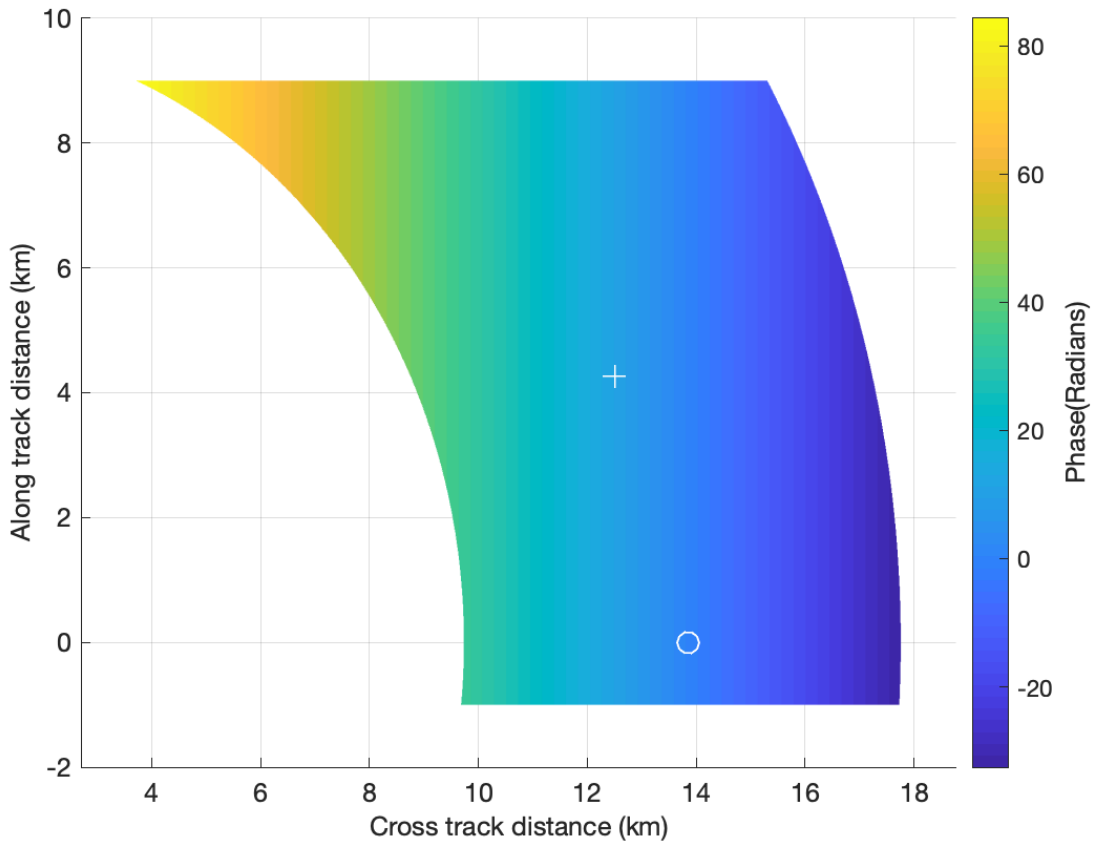
Figure 4 Phase Variation Within a 10-km by 10-km range/azimuth Pixel. The x-axis is the cross-track distance with the spacecraft nadir point at zero. The y-axis is the along-track distance, also with nadir at zero. The boundaries of the colored region are iso-range and iso-Doppler contours, which result in a shape that approaches rectangular for small spatial extents, especially in the far-range part of the swath, but diverges substantially from rectangular for large extents in the near-range (near nadir) part of the swath. Individual 500-m resolution pixels in KaRIn telemetry exhibit similar behavior, though a larger pixel is shown here to better illustrate the effect, which would be hard to see visually for a single

500-m resolution pixel.

In the L1B_LR_INTF processor we numerically integrate simulated interferometric responses over a reference surface to determine the interferometric phase and the equivalently weighted location on the reference surface that we would expect for each SWOT measurement pixel. The simulated location and phase can then be used in the downstream processing. If the measured phase is precisely the same as the simulated phase, then the measured pixel is on the reference surface at the simulated (reference) location assuming no 2π radian ambiguity. Otherwise the difference between the simulated and measured phase is used to accurately determine the vector between the reference and measured locations.

In addition to removing systematic phase biases, the L1B_LR_INTF processor also computes and calibrates other quantities that are used to perform refinements to SSH in downstream processing.

2.2 Functional Flow

Figure 5 shows the flow diagram of the L1B_LR_INTF processor. Table 1 provides a high-level description of each of the processing functions that are used to generate the L1B_LR_INTF product.

The SWOT KaRIn instrument collects data from both the left and right side half-swaths. Some of the functions described below compute quantities for both sides, while others are applied independently to each side.
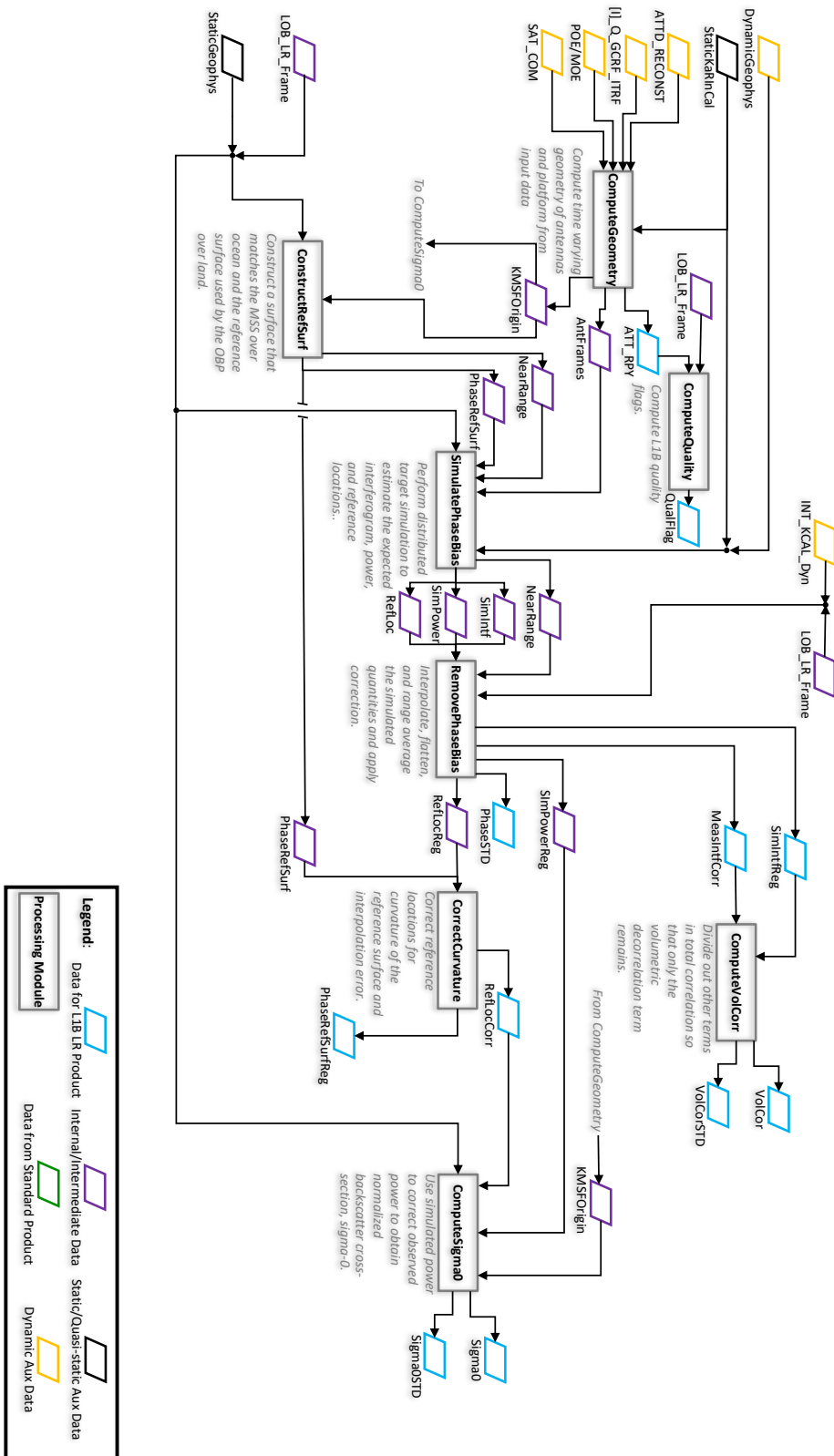
Figure 5 Flow Diagram. High level functions are rectangles. Parallelograms are color-codes

sets of inputs, outputs, auxiliary files and internal data structures. A brief description of each function is included below each rectangle.

Table 1. High level description of the functions used to generate the L1B_LR_INTF product

Function Name	Description
ComputeGeometry	Compute time varying geometry of antennas and platform from input data
ConstructRefSurf	Construct a reference surface using the MSS over ocean and the reference surface used by the OBP over land. The constructed reference surface may differ from the MSS where convenient (e.g. smoother or coarser resolution).
SimulatePhaseBias	Perform distributed target simulation to estimate the expected interferometric phase, angular correlation, power, and reference location for each pixel.
RemovePhaseBias	Interpolate, flatten, and range average the simulated interferogram. Subtract the simulated phase from the observed phase. Also interpolate and range average the simulated power and reference locations.
CorrectCurvature	Correct reference (expected) pixel locations for curvature of the reference surface and other approximation errors so that the reference locations are on the reference surface.
ComputeSigma0	Use simulated power to correct observed power to obtain normalized backscatter cross-section, sigma0.
ComputeVolCorr	Divide out other terms in total correlation so that only the volumetric decorrelation term remains.
ComputeQuality	Compute L1B quality flags

3 Algorithm Descriptions

3.1 ComputeGeometry

3.1.1 Purpose

Given spacecraft ephemeris and attitude and known static antenna orientation with respect to the spacecraft, this function computes antenna positions and attitude with respect to an Earth-fixed coordinate system at the time of each KaRIn LR observation. The inputs to this algorithm are in several different coordinate frames. One of these is a reference frame that is fixed to the KaRIn instrument called the KaRIn Metering Structure Frame (KMSF), illustrated in Figure 6. This frame is defined with the origin near the middle of the interferometric baseline, with the two antennas along the $+y$ and $-y$ axes. The $+z$ axis of this frame is controlled by the spacecraft to point approximately toward nadir, so the x axis is approximately parallel or antiparallel to the Earth-fixed spacecraft velocity vector. However, the spacecraft periodically performs 180˚ yaw flips (for thermal management reasons, several times per year) such that sometimes the $+x$ axis is in the direction of the velocity vector (i.e., satellite flying forward), and sometimes the $-x$ axis is in the direction of the velocity vector (i.e., satellite flying backward). Which of the $+y$ and $-y$ antennas is to the left or right of the spacecraft along-track direction therefore depends on the yaw state of the spacecraft. As elsewhere in this document, "left" and "right" are defined as if standing on the Earth surface and facing the direction of the spacecraft velocity vector.

19
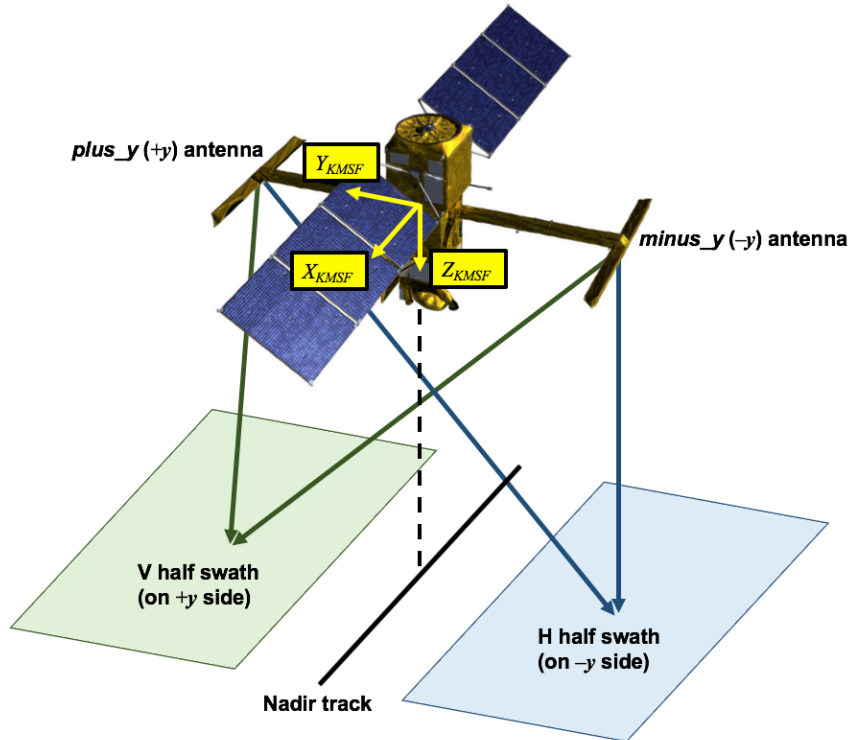
Figure 6 Illustration of the KMSF frame and the polarizations (V and H) of the two KaRIn half swaths. The velocity direction can be along $+X_{KMSF}$ or $-X_{KMSF}$ depending on the yaw state of the spacecraft. See definition of KMSF frame in text above.

In addition to KMSF some inputs (spacecraft attitude) are provided in the geocentric celestial reference frame (GCRF), an Earth-centered inertial coordinate system. The precise definition of this frame can be found in **[4]**.

The outputs of this routine are represented in the international terrestrial reference frame (ITRF). In this Earth-centered, Earth-fixed (ECEF) frame, the $+z$ axis of the ECEF frame goes through the north pole, and the $+x$ axis goes through both the equator (zero latitude) and the prime meridian (zero longitude). This routine also outputs rotations from ITRF to the frames which were used to measure the antenna patterns so that antenna gain can be computed. These rotations are compositions of rotations from ITRF to KMSF and rotations from KMSF to the antenna measurement frames. The precise rotation from KMSF to each antenna measurement frame is a calibration input to this routine, which is determined during cal/val with prelaunch measurements of the antenna mounting being used as an initial estimate. There are four different antenna measurement frames ($+y$ and $-y$ antennas for each of the left and right half swaths).

3.1.2   Input Data

Description	Source
Lever arm estimates (LeverArm) from Cal/Val in spacecraft coordinate system, KMSF, and antenna phase center (AntPhaseCent) orientation estimates as a	StaticKaRInCal

function of polarization and antenna, (e.g. rotation matrices from each antenna pattern measurement frame to KMSF).	
Ground reconstructed spacecraft attitude in GCRF.	ATTD_RECONST
Time varying rotations from GCRF to ITRF.	[I]_Q_GCRF_ITRF
Position and velocity of spacecraft center of mass in Earth Fixed (ITRF) coordinates	POE/MOE
Position of spacecraft center of mass relative to KMSF.	SAT_COM

3.1.3 Output Data

Description
Antenna position and pointing in Earth fixed (ITRF) coordinate system for each of the four antenna phase centers (two physical antennas and two polarizations with different phase centers for each). (AntFrame)
Position and velocity of origin of KMSF frame in ITRF coordinates (KMSFOrigin)
Spacecraft attitude with respect to desired pointing (ATT_RPY) expressed as roll, pitch, and yaw.

The data produced by this routine is used by SimulatePhaseBias and RemovePhaseBias.

3.1.4 Mathematical Statement

The algorithm proceeds as follows:

Step 1: Legacy CNES algorithms are utilized to compute at each desired time, t, the position $r_0(t)$ and velocity $V(t)$ of the origin of the spacecraft coordinate system (KMSF) in the Earth-fixed reference frame (ITRF) from the precision orbit ephemeris (POE) or medium-accuracy orbit ephemeris (MOE), the satellite center of mass (SAT_COM) in KMSF, and rotational information necessary to transform the vector between the KMSF origin and the spacecraft center of mass to ITRF coordinates. The POE and MOE provide high-accuracy estimates of the position and velocity of the spacecraft center of mass in the ITRF coordinate system. They are computed using a combination of spacecraft force models and measurements from the onboard Global Positioning System (GPS) and Doppler Orbitography and Radiopositioning Integrated by Satellite (DORIS) tracking systems [5]. The POE and MOE are interpolated to the time of interest using the well-established Everett interpolation method [6]. The location of the spacecraft center of mass in the KMSF is modelled based upon knowledge of the solar array orientation, and propellant consumption [7].

Step 2: Legacy CNES algorithms are used to compute the rotation matrix $M_0(t)$ from KMSF to ITRF at time t from ATTD_RECONST and [I]_Q_GCRF_ITRF inputs. Spacecraft attitude in the ITRF frame are generated using a combination of observations of orientation of the (body-fixed) KMSF coordinate system with respect to the geocentric celestial reference frame (GCRF) [8] and computations of the orientation of the ITRF with respect to the GCRF. The GCRF is the Earth-centered inertial coordinate system as defined in [4]. The GCRF to KMSF orientation is based upon observations from the onboard star trackers and gyros, and provided at 64 Hz. The GCRF to ITRF orientation is generated during the precise orbit determination of the spacecraft based upon Earth orientation parameters produced by the International Earth Rotation Service. These orientations are provided as quaternions and are interpolated to the

time of interest using Spherical Linear Interpolation. Rotation matrix notation is used here for ease of understanding.

Step 3: The location of the antennas *r(t,p,a)* are computed by the following equation at each time *t* and for each polarization $p \in \{VV, HH\}$ and each antenna $a \in \{$ +KMSF-y-axis, -KMSF-y-axis$\}$.

$$r(t, p, a) = r_0(t) + M_0(t)r_c(p, a)$$

Where $r_c(p, a)$ is the calibrated lever arm for polarization, *p*, and antenna *a*. The lever arm is the location of each antenna phase center in KMSF coordinates. An assumption is made here that the phase center of each antenna and polarization is constant over time in KMSF coordinates.

Step 4: The rotation matrix *M(t,p,a)* from each antenna pattern measurement frame to ITRF is given by $M(t, p, a) = M_0(t)M_c(p, a)$, where *M$_c$(p,a)* is the calibrated rotation matrix from the antenna pattern measurement frame to KMSF. In this manner, the pointing of each antenna and polarization is composed of a constant rotation computed during instrument calibration and the reconstructed time-varying spacecraft attitude.

Step 5: Compute attitude relative to a local frame. The local frame is defined to be a TCN (track, cross-track, nadir) frame. The *z*-axis of the TCN frame is the vector, *N*, normal to the ellipsoid and through the KMSF origin; the +*z*-axis points downward. The *y* axis is parallel to the cross product $N \times V$ where *V* is the spacecraft velocity vector in the ITRF frame. The *x*-axis is the vector nearly parallel to *V* that completes the righthand coordinate system. The rotation is parameterized by three Euler angles: roll, pitch, and yaw in that order. The yaw is the angle of right-handed rotation of the TCN +x axis about the nadir direction (TCN -z axis). If the KMSF +x axis is aligned with the horizontal projection of the Earth-relative spacecraft velocity vector (not yaw-flipped), the yaw will be zero. If the KMSF -x axis is aligned with the horizontal projection of the Earth-relative spacecraft velocity vector (yaw-flipped), the yaw will be 180˚. Both orientations will occur frequently during nominal operations. The *velocity heading*, as reported in the L1B product, is the angle with respect to true north of the nadir track direction such that if the spacecraft were flying due east, the velocity heading would be 90˚. The heading of the KMSF +x axis relative to true north is consequently the sum of the velocity heading and the yaw (modulo 360˚). The pitch is defined such that a positive pitch moves the KMSF +x axis up. The roll is defined such that a positive roll moves the KMSF +y axis down. Note that when the yaw is near 180˚, the sense of pitch and roll may be counterintuitive to users who are accustomed to airborne platforms since the spacecraft would be flying "tail first." Ideally, when not yaw-flipped, the spacecraft would have zero attitude, and the KMSF frame would be aligned with the TCN frame, but the two frames will in general not coincide exactly because attitude control of the spacecraft is not perfect and because the antennas are not perfectly aligned with KMSF.

3.1.5 Accuracy

The algorithm should impart no significant errors. The magnitude of the errors in the

antenna pointing angle will be solely due to errors in spacecraft attitude knowledge and antenna phase center calibration. The position errors of each antenna will be solely due to errors in inputs to algorithm (e.g. error in calibrated lever arm, spacecraft ephemeris knowledge error, knowledge error in the location of the center of mass, and lever arm dynamics). The algorithm assumes that antenna phase centers do not move in KMSF.

3.2 ConstructRefSurf

3.2.1 Purpose

Given a mean sea surface and a reference topography table over land, construct the *reference surface* for use in SimulatePhaseBias. The *reference surface* is the surface for which the measured phase after correction for biases would be zero. After phase bias correction, a nonzero measurement phase indicates that the observed surface height differs from the height of the reference surface. Downstream processing determines the magnitude and sign of that difference. This function produces a swath-aligned representation of the reference surface, which is assumed to be the mean sea surface (MSS) over the ocean. The swath-aligned grid is much smaller in spatial extent than the MSS but large enough to cover the SWOT observation being processed.

3.2.2 Input Data

Description	Source
Mean Sea Surface (MSS), Surface Type Map (SurfType)	StaticGeophys
Topography table (TopoOBP) and near range (NearRange) used in Onboard Processor	L0B_LR_Frame
Position and velocity of origin of KMSF frame in ITRF coordinates (KMSFOrigin)	ComputeGeometry

3.2.3 Output Data

Description
Reference surface (PhaseRefSurf) used in SimulatePhaseBias
Near range table (NearRange), simply pass through

3.2.4 Mathematical Statement

The algorithm constructs a reference surface whose heights, $H(s,i,j)$, are relative to the ellipsoid. The index s denotes either the right or the left swath, i is the along-track index, and j is the cross-track index. Cross-track distances are defined to be negative to the left of the nadir track and positive to the right. Along-track distances are defined to increase with time or with the flight direction of the spacecraft. The two sides have separate maps to avoid creating discontinuities at nadir over land.

Along-track spacing. The first along-track position is chosen to be a distance $d_0 = 5$ km

before the spacecraft location during acquisition of the first record in the input L0B_LR_Frame granule. The last along-track location is approximately d_0 beyond the spacecraft location of the last record in the input L1B_LR_Frame. The along-track spacing is slightly irregular because it corresponds to spacecraft nadir points that are regularly spaced in time at every 0.003426 s which results in approximately 22-m separation between consecutive along track positions. Although, the pulse repetition interval (PRI) varies in the orbits, a regular sampling in time was chosen to simplify interpolation of the reference surface.

Cross-track spacing. For the left swath, cross track distances are sampled from -80 km to 0 km spaced every 50 m. For the right swath, cross track distances are sampled from 0 km to 80 km spaced every 50 m.

Step 1: Determine latitude and longitude for each reference surface pixel for each along track-index, i, and cross-track index, j. The latitude and longitude of the zero cross-track-distance (nadir) $j=0$ pixels are the spacecraft latitude and longitudes for the ith regularly sampled ephemeris time, t_i. Stated differently, they are the sub-spacecraft(nadir positions) for each time t_i, $r_o(t_i)$. For other cross-track locations, we utilize a spherical approximation to the ellipsoid in the neighborhood of $r_o(t_i)$. The pixel location $r(s, i, j)$ are regularly sampled points on the great circle of the approximating sphere that is orthogonal to the plane containing the surface normal vector at $r_o(t_i)$ and the spacecraft velocity vector $V(t_i)$.

Step 2: Determine surface type for each pixel. The surface type is looked up by latitude and longitude from a table in the StaticGeophys input data. The surface type for each pixel is derived from a global gridded surface classification map. The product description document [3] should be referred to for the currently adopted surface classification map. Surface types include open ocean, land, continental water, aquatic vegetation, continental ice or snow, floating ice, and salted basin. Only a binary choice of ocean or land is used in this routine. Open ocean and floating ice are treated as ocean. All other categories are treated as land.

Step 3: Determine heights for pixels over ocean and floating ice. Height over ocean at each pixel is derived from bilinear interpolation of a global gridded mean sea surface height map. The product description document [3] should be referred to for the currently adopted mean sea surface height map.

Step 4: Determine heights for pixels over other surface types. Heights over other surface types are the heights in the topography table used by the OBP. The OBP heights are assumed to be chosen to best match the surface heights of water bodies of highest priority for LR processing. Over land we will not be able to get accurate heights over all bodies of water because inaccurate flattening done in the on board processor will lead to averaging phases that can vary significantly. Because of an approximation in the OBP if the height of two bodies of water observed at the same time (same sensor line; same telemetry record) are different by more than 10-m or so, one of them may be improperly flattened. The OBP reference surface will be

selected to optimize only one of the water bodies. We use the same reference surface as the OBP over land so the data the OBP did properly flatten is treated properly by the ground processing. Data that was not properly flattened by the OBP is typically not recoverable anyway.

3.2.5 Accuracy

This routine imparts no significant error to SSH. Errors related to how the reference surface is used in SimulatePhaseBias are reported in the section describing that routine.

3.3 SimulatePhaseBias

3.3.1 Purpose

As explained in Section 2, the OBP produces interferograms that contain a systematic phase bias that has to be compensated in ground processing. This bias is due to the misalignment of iso-range contours and iso-interferometric-phase contours and due to approximations in the OBP itself. The SimulatePhaseBias routine computes this phase bias by means of a distributed-target simulation.

3.3.2 Input Data

Description	Source
Reference surface (PhaseRefSurf) and range to first pixel for each simulated telemetry line (NearRange), which needs to be consistent with but need not be identical to first pixel range in downlinked, depacketized (L0B) telemetry	ConstructRefSurf
Antenna position and pointing (AntFrames) in Earth fixed, ITRF, coordinate system.	ComputeGeometry
Antenna patterns (AntPat)	StaticKaRInCal
Doppler compensation coefficients (DopCoeff) used by OBP needed to compute azimuthal point target response	L0B_LR_Frame
Interferometric and single channel range point target response (RangePTR)	Param_L1B_LR_INTF
Geophysical model function table that relates wind speed and incidence angle to normalized backscatter cross-section, sigma0 (WindGMF)	StaticGeophys
Media delays from dry troposphere, wet troposphere, and ionosphere (MediaDelay)	DynamicGeophys

3.3.3 Output Data

Description
Simulated complex interferogram (SimIntf) which includes interferometric phase and coherence due to reference surface and viewing geometry for each sensor line L, range index k, and beam m.
Simulated SAR power (SimPower) from both channels and for each sensor line L, range index k, and beam m.

| Reference locations (RefLoc) for each for each sensor line *L*, range index *k*, and beam *m*. |
| Near range table (NearRange), simply pass through |

3.3.4 Mathematical Statement

The algorithm described below is for a single side. How it varies with left or right side is specified in Table 2 below. Given a reference surface (S), this routine simulates the complex interferogram by integrating the radar response over S for each sensor line L, range bin number k, and beam m. The reference surface S, is divided into pixels indexed by i and j in the along and cross track directions, respectively (Figure 7). The sensor line number L is approximately proportional to the along track index, i. The spacing of L is 2.5-km which is 100 times coarser than the 25-m spacing of i. (Index i, should not be confused with $\iota = \sqrt{-1}$.) Range bin index k increases with distance from nadir and is linearly proportional to slant range which is the distance from a pixel on the surface to the spacecraft. Cross track index j, on the other hand, is proportional to ground range, the arc length on a local spherical approximation to the ellipsoid from the sub-satellite point. Index n represent further division of pixels in the cross track direction introduced for proper sampling of the surface as explained in Appendix B.
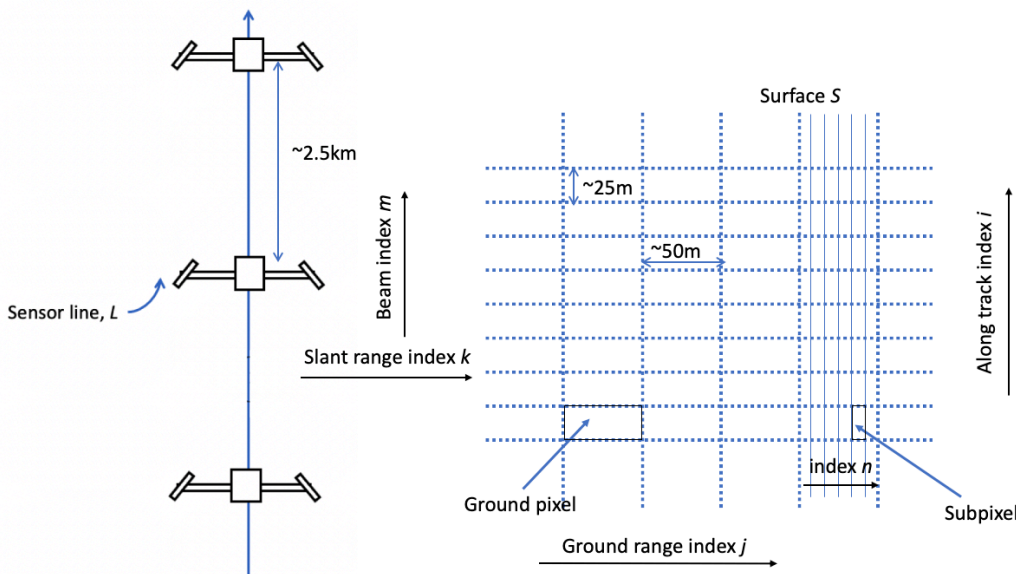
Figure 7 A pictorial of the surface over which the SimulatePhaseBias routine integrates to obtain the interferometric phase. The grid of the simulated surface is given by ground range index, j, and along track index, i, in which a DEM pixel is denoted by (i,j). Subpixels are ground range subdivisions of a pixel with index n. The number of subdivisions varies with incidence angle and thus with ground range. The directions of the other important indices beam index, m, and slant range index, k, are also shown. The illustration on the right shows the positions of the instrument for each telemetry record (sensor line) that is simulated.

During onboard processing, nine different beams are formed from the range compressed KaRIn SAR data using azimuth/Doppler compression. Phase biases are different for the different

beams due to differences in the point target response (PTR), which is the response of the onboard azimuth and range compression processing and interferogram formation to energy scattered from a single point on the surface. For each side of the swath, there are three different PTRs that are relevant to phase bias simulation: the two single channel PTRs and the interferometric PTR. The PTRs are the responses due to range compression, geometric decorrelation correction, azimuth compression, and/or interferogram formation only. They do not include the effects of antenna gain, surface roughness, attenuation due to propagation, or differences in the path from the point target to each antenna. The two complex-valued single channel PTR functions $R_1(\vec{x}, L, k, m)$ and $R_2(\vec{x}, L, k, m)$ characterize the voltage gain and phase for a signal scattered from a single point target at location $\vec{x}$ for each range bin, k, and beam, m, after range and azimuth compression at the measurement time corresponding to sensor line L. The subscripts denote the receiving antenna (channel). Channel 1 receives on the transmitting antenna. Channel 2 receives on the non-transmitting antenna. The distribution of power into range and Doppler (beam) bins is characterized by the single channel real-valued power PTRs: $\|R_1(\vec{x}, L, k, m)\|^2$ and $\|R_2(\vec{x}, L, k, m)\|^2$. The interferometric PTR is represented by

$$R_{intf}(\vec{x}, L, k, m) = R_1(\vec{x}, L, k, m)R_2^*(\vec{x}, L, k, m)$$

where "*" superscript indicates the complex conjugation operation. It characterizes the response of the range and Doppler compression in both channels followed by interferogram formation. Both power and interferogram PTRs are decomposed into range and azimuth PTRs by the following equations.

$$R_1(\vec{x}, L, k, m) = \sqrt{R^{az}(\vec{x}, L, k, m)}R_1^r(r - r_k)$$

$$R_2(\vec{x}, L, k, m) = \sqrt{R^{az}(\vec{x}, L, k, m)}R_2^r(r - r_k)$$

$$R_{intf}(\vec{x}, L, k, m) = R^{az}(\vec{x}, L, k, m)R_{intf}^r(r - r_k)$$

The azimuth PTR, R^{az}, depends on k and L because the Doppler centroid used by the OBP varies with range and time. Range PTRs are precomputed using a simulation of the OBP and then oversampled. The range PTR depends only on the difference between slant range to the point target, r, and the range bin r_k. R^{az} is a real-valued quantity and is proportional to power leading to the square root in the single channel PTR equations.

This decomposition is correct for nondispersive antennas, that is for antennas for which the gain and phase contribution from the antenna does not vary with frequency. For dispersive (frequency varying) antenna patterns, the antenna pattern complex gains cannot in general be decoupled from the PTR. The best estimates of the SWOT antenna pattern include some dispersion, but the frequency variation can be modeled reasonably well as a linear phase ramp with frequency. The slope of this phase ramp varies with look direction, and can be modeled as a delay that varies with antenna elevation and azimuth angle. Antenna dispersion is handled by a combination of effective nondispersive antenna patterns and delays common to both antennas

that vary with azimuth and elevation. The algorithm described below is for a nondispersive antenna, but validation for dispersive antenna patterns is provided in the accuracy section.

The various range PTRs are determined by passing a single ideal transmitted pulse through an oversampled simulation of the OBP range compression and interferogram processing. The resulting power and interferometric PTR are tabulated as a function of $r - r_k$ and used as a look up table during the phase bias computation. The azimuth PTR is determined by mimicking the pulse weighting used in Doppler compression by the OBP as follows:

$$R^{az}(\vec{x}, L, k, m) = \sum_{p=1}^{9} \big(W_d(p)\big)^2 + \sum_{p_1=1}^{9} \sum_{p_2=p_1+1}^{9} 2W_d(p_1)W_d(p_2) \cos\big(\mu(\vec{x}, L, k, m)\big)$$

$$\mu(\vec{x}, L, k, m) = 2\pi(p_1 - p_2)\left(\frac{B_f(m-1)}{9} - 4 + d_f(L, k)\right) + \frac{4\pi}{\lambda}(r(t_1, \vec{x}) - r(t_2, \vec{x}))$$

which is equivalent to

$$R^{az}(\vec{x}, L, k, m) = \left| \sum_{p=1}^{9} W_d(p) e^{2\iota\pi\left[p\left(\frac{B_f(m-1)}{9} - 4 + d_f(L,k)\right) + \frac{4\pi}{\lambda}(r(t(p),\vec{x}))\right]} \right|^2$$

Here p, p_1 and p_2 are indices for nine radar pulses centered on the measurement time of sensor line L; $t(p)$, $t_1 = t(p_1)$, and $t_2 = t(p_2)$ are the measurements times that correspond to those indices; W_d is the windowing function applied to the pulses in the OBP, $B_f = 0.8$ is the fraction of the Doppler Bandwidth processed in the OBP, $d_f(L, k) \in (-0.5, 0.5]$ is the Doppler centroid of the radar echo expressed as a fraction of the PRF, λ is the operating wavelength, and $r(t, \vec{x})$ is the distance from the midpoint between the two antennas to the point target location $\vec{x}$ at time t.

Given the reference surface, S, the complex-valued normalized interferogram can be expressed by integrating the product of the interferometric PTR and other terms due to antenna gain, surface roughness, attenuation due to propagation, and differences in the path from the point target to each antenna [2]. In the integral equation that follows, all ranges and antenna gains are implicitly a function of time and thus a function of L. Antenna patterns are also a function of polarization (see Table 2). Polarization is co-pol only and is only used to isolate signal returns from different sides. We assume cross-pol contributions due to backscatter and the antenna are zero.

$$\Phi(L, k, m) = \frac{P_t \int_S A(r) G_{11}(\vec{x}) G_{12}^*(\vec{x}) \sigma_0(\vec{x}) R_{intf}^r(r - r_k) R^{az}(\vec{x}, L, k, m) e^{\iota\frac{2\pi}{\lambda}(r_1 - r_2)} d\vec{x}}{\sqrt{\dfrac{P_{r1}(L, k, m) P_{r2}(L, k, m)}{X_{cal1} X_{cal2}}}}$$

where

P_t = transmit power

X_{cal1} and X_{cal2} are static power calibration terms for each channel

A = signal attenuation due to propagation

G_1 = the one-way transmitting antenna complex voltage gain

G_2 = the one-way nontransmitting antenna complex voltage gain

$G_{11}(\vec{x}) = G_1(\vec{x})G_1(\vec{x})$, and $G_{12}(\vec{x}) = G_1(\vec{x})G_2(\vec{x})$.

σ_0 = the normalized radar cross-section at the surface (NRCS or sigma0)

$\iota = \sqrt{-1}$

r_1 = slant range from the transmitting antenna including media delays

r_2 = slant range from the nontransmitting antenna including media delays

P_{r1} = received power on transmitting antenna (defined below)

P_{r2} = received power on nontransmitting antenna (defined below)

This integration is carried out numerically by placing point targets at the locations in the reference surface map produced by ConstructRefSurf. The numerical implementation is expressed by the following sum

$$\Phi(L, \text{k}, m) = \frac{P_t \sum_{i=i_0(L)}^{i_1(L)} \sum_{j=j_0(k)}^{j_1(k)} A_{ij}^f A(r_{ij}) G_{11}(\vec{x}_{ij}) G_{12}^*(\vec{x}_{ij}) \sigma_0(\vec{x}_{ij}) R_{intf}(\vec{x}_{ij}, L, k, m) e^{-\iota \phi_{ij}}}{\sqrt{\dfrac{P_{r1}(L, k, m) P_{r2}(L, k, m)}{X_{cal1} X_{cal2}}}}$$

where i, are the indices for the reference surface lines, j is the cross track spacing of the reference surface, A_{ij}^f is the facet area, and $\phi_{ij} = \dfrac{2\pi(r_{1_{ij}} - r_{2_{ij}})}{\lambda}$. The facet area is the area of a rectangular region (facet) on the reference surface that contains a single point target. The facets are contiguous so that their combined area approximates the portion of the reference surface containing the KaRIn swath. Facets are much smaller than a KaRIn pixel in range (indexed by k) and Doppler (indexed by m).

To ensure proper sampling when performing the double sum in the phase equation above, we need to divide each ground pixel into n subpixels, as indicated in Fig. 7, such that the slant range difference from one subpixel to the next is less than the sampling resolution which is ~0.5m. For pixels with 50m size in the cross-track direction, the number of divisions needed for each ground pixel ranges between 1 in the near range and 80 in the far range. This sub-division of each ground pixel adds a significant additional computational requirements and can be avoided by recognizing that within each pixel, the only fast varying term is R_{intf}, while all other terms can be modeled as linearly varying or constants. This allows us to capture the integral of R_{intf} between pixels in a "PTR integration table" and avoid the subdivision of pixels. These PTR integration tables speed up the process by a factor of 70-80 at the cost of an additional submillimetric errors in SSH. More details on the computation and use of these tables are given in Appendix B.

Antenna gains are looked up from antenna pattern measurement tables. The tables are organized by elevation and azimuth angle in the antenna measurement frames. For each sensor line, L, the summation limits $[i_0(L), i_1(L)]$ and $[j_0(k), j_1(k)]$ are chosen to be large enough to cover the main lobe of the KaRIn pixel and sufficient side lobe structure in order to meet science requirements for algorithm error. Summation limits that comprise 50-m in slant range centered at the slant range r_k and 5 km centered at the peak of the antenna beam footprint in the azimuth direction are sufficiently large. Point targets from the opposite swath have been shown to not contribute significantly to the summation. The time-varying coordinate transformations computed in ComputeAntennaGeometry are used to map each point target to its location in the antenna measurement frames. The sigma0 used here is an analytically determined estimate which should not be confused with the KaRIn sigma0 measurement. The sigma0 estimate is computed as a function of incidence angle and ocean surface wind speed using an empirically-derived model function in the WindGMF table. Incidence angles are determined using the KMSF origin positions and the reference locations. A constant "typical" ocean wind speed, which is a configurable parameter, is used in the calculation. Transmit power and receiver gain are computed during instrument calibration and are expected to be constant over the short time scales that might impact the phase bias calculation. Dynamic variation of transmit power and receiver gain on longer time scales are computed in the Dynamic KaRIn Calibration processor and used to refine the measured sigma0 calculation in ComputeSigma0. Any receiver gain or transmit power variation in frequency will be accommodated by modifying the common group delay table used in the dispersive antenna phase correction.

For a dispersive antenna, we compute and use the effective gains G_r, G_{11}, and G_{12} by averaging the complex gain over the transmit bandwidth after subtracting a best-fit frequency-dependent phase ramp. These effective gains are thus frequency-independent, and can be treated as the gains of an approximately equivalent non-dispersive antenna. The phase ramps are then converted to the equivalent group delay, τ. The phase ramp and group delay are both functions of antenna azimuth and elevation angle. We then account for this group delay by converting it to the equivalent range, $\frac{c\tau}{2}$, and adding this range to r_k.

The phase of $\Phi(L, k, m)$ can have a sign error compared to the OBP in some circumstances, the table below describes when a sign change is needed and thus the phase of $\Phi^*(L, k, m)$ is used instead. It also lists the polarization.

Table 2 Sign of phase as a function of halfswath side and orientation of spacecraft in flight

Do we need to apply complex conjugation to $\Phi(L, k, m)$?	Antenna Polarization	Side of half-swath with respect spacecraft heading	Is the spacecraft yaw flipped (i.e., flying backwards)?	Is the default (KMSF +y-axis) antenna used for transmit?
No	HH	Left	No	Yes
No	VV	Right	No	Yes

No	VV	Left	Yes	Yes
No	HH	Right	Yes	Yes
Yes	HH	Left	No	No
Yes	VV	Right	No	No
Yes	VV	Left	Yes	No
Yes	HH	Right	Yes	No

Similar summations are used to simulate the received powers for each channel.

$$P_{r1}(L, \text{k}, m) = P_1 \sum_{i=i_0(L)}^{i_{1(L)}} \sum_{j=j_0(k)}^{j_1(k)} A_m(\vec{x}_{ij}) A_{ij}^f A(r_{ij}) G_{11}(\vec{x}_{ij}) G_{11}^*(\vec{x}_{ij}) \sigma_0(\vec{x}_{ij}) \left\| R_1(\vec{x}_{ij}, L, k, m) \right\|^2$$

$$P_{r2}(L, \text{k}, m) = P_2 \sum_{i=i_0(L)}^{i_{1(L)}} \sum_{j=j_0(k)}^{j_1(k)} A_m(\vec{x}_{ij}) A_{ij}^f A(r_{ij}) G_{12}(\vec{x}_{ij}) G_{12}^*(\vec{x}_{ij}) \sigma_0(\vec{x}_{ij}) \left\| R_2(\vec{x}_{ij}, L, k, m) \right\|^2$$

Here $A_m(\vec{x}_{ij})$ is the media attenuation term, and the constant coefficients for each channel are $P_1 = P_t G_r X_{cal1}$ and $P_2 = P_t G_r X_{cal2}$.

In addition to computing powers for each channel, the radiometric calibration quantities $X_1(L,k,m)$ and $X_2(L,k,m)$ are computed similarly by simply omitting the σ_0 terms in the summations. These quantities are the ratios between received power and σ_0.

The reference locations, $\vec{x}_{ref}$ are computed by a normalized weighted summation of the point target locations

$$\vec{x}_{ref}(L, \text{k}, m) = \frac{\sum_{i=i_0(L)}^{i_{1(L)}} \sum_{j=j_0(k)}^{j_1(k)} W_{ijm} \vec{x}_{ij}}{\sum_{i=i_0(L)}^{i_{1(L)}} \sum_{j=j_0(k)}^{j_1(k)} W_{ijm}}$$

such that

$$W_{ijm} = A_m(\vec{x}_{ij}) A_{ij}^f \left\| G_{11}(\vec{x}_{ij}) \right\| \left\| \hat{G}_2(\vec{x}_{ij}) \right\| \left\| \sigma_0(\vec{x}_{ij}) \right\| \left\| R_1(\vec{x}_{ij}, L, k, m) \right\| \left\| R_2(\vec{x}_{ij}, L, k, m) \right\|$$

The vector $\vec{x}_{ref}(L, \mathrm{k}, m)$ is not precisely on the reference surface due to the curvature of the surface, but the CorrectCurvature routine discussed below fixes this problem so that it does not introduce errors in the downstream processing. The reference locations and the phase of the simulated interferogram Φ are used by the downstream processing to determine SSH. The magnitude of Φ is used to compute volumetric correlation in ComputeVolCor. The simulated received powers are used to compute sigma0 in ComputeSigma0.

3.3.5 Accuracy

The errors in this routine are due to approximations made that reduce the computational complexity. These approximations include 1) the use of a table look-up approximation that removes slowly varying terms from the inner loops of the computation (see Appendix B for details), 2) the relatively coarse 2.5-km sampling in along track, 3) the approximation that the range PTR is separable from the azimuth PTR, 4) the assumption that the complex-valued antenna patterns can be correctly represented by a nondispersive (frequency-invariant) antenna pattern and a group delay pattern, 5) the numeric integration over the ground surface that treats the surfaces as a discretized set of targets with finite integration extent, and 6) the reference surface used for the bias correction is not the true surface. Note that errors in the knowledge of parameters such as the spacecraft position, velocity, and attitude; media delays; antenna pattern; lever arms; and range PTR are not treated as errors in the phase bias algorithm per se; such errors will affect the eventual height errors, but they are captured separately.

The science requirements [1] that drive the accuracy requirement in the L1B processing is that the along track sea surface height error spectra *E(f)* is bounded by.

$$E(f) \leq 2 + 0.00125 f^{-2}\ [cm^2/(cycles/km)], \lambda \in [15,1000]\ km$$

E(f) is defined to be the function of the spatial frequency *f* (i.e., *f=1/wavelength=1/λ*) (the same as the term "wavenumber" used in some oceanographic literature) such that the expected SSH error variance in the wavelength interval [λ_{min}, λ_{max}] is given by the integral of *E(f)* from $f = 1/\lambda_{max}$ to $1/\lambda_{min}$. Here λ refers to spatial (ocean) wavelength rather than electromagnetic (radar pulse) wavelength.

To meet this requirement the SWOT error budget bounds the ground processing algorithm error portion of the error spectrum by $E_a(f) =\ 2.2 \times 10^{-6} f^{-2.5}\ [cm^2/(cycles/km)]$, for λ>47.6 km and $E_a(f) = 0.0342\ [cm^2/(cycles/km)]$ for $\lambda \leq 47.6$ km.

There is no requirement on the algorithm accuracy of single beam heights or phase so long as the beam-combined heights meet the requirements. The accuracy of the L1B algorithms are therefore evaluated at a higher level by validating the ultimate height errors in end-to-ed simulations. For plots demonstrating the accuracy of KaRIn beam-combined heights and thus demonstrating the accuracy of this routine please see [9].

3.4 RemovePhaseBias

3.4.1 Purpose

RemovePhaseBias modifies the simulated interferogram to account for effects after interferogram formation in the OBP by phase flattening and range averaging the simulated interferogram exactly as was done in the OBP for the measured interferogram. It also interpolates in azimuth to align the 2.5-km azimuth sampling of the simulated phase bias with the 250-m posting of the measured interferogram output by the OBP. Once the simulated phase bias is properly coregistered with the measured interferogram the phase bias correction is applied.

3.4.2 Input Data

Description	Source
Simulated interferogram (SimIntf), simulated SAR power (SimPower) from both channels, reference locations (RefLoc) for each simulated interferogram pixel, near range used by phase bias simulation (NearRange)	SimulatePhaseBias
Various OBP parameters needed to properly mimic OBP operation (OBPParam)	INT_KCAL_Dyn
Measured interferogram from telemetry (MeasIntf)	L0B_LR_Frame

3.4.3 Output Data

Description
Measured normalized interferogram with phase bias correction applied (MeasIntfCorr)
Simulated power coregistered with measurements (SimPowerReg)
Simulated normalized interferogram coregistered with measurements (SimIntfReg)
Reference Locations coregistered with measurements (RefLocReg)
Standard deviation of measured phase analytically determined from measured coherence and known instrument noise characteristics (PhaseSTD)

3.4.4 Mathematical Statement

Step 1: *Flatten the simulated unnormalized interferogram.* A flattening phase $\beta(L, k, m)$ is added to the phase of the interferogram for each sensor line L, range index k, and beam, m. The flattening phase is computed in the same manner used by the OBP, using the law of cosines [2].

$$\beta(L, k, m) = c_f \sin\left(\cos^{-1}\left(\frac{(h_p + d)^2 + r_k^2 - (h_g + d)^2}{2(h_p + d)r_k - b} \right) \right)$$

where c_f is the flattening coefficient used by the OBP, which incorporates the OBP's estimate of the wavelength and the distance between the antennas, h_p is the OBP's value for the platform height above the ellipsoid, h_g is the surface height above the ellipsoid from the OBP's topography table, and d is the OBP's estimate of the radius of curvature of Earth. The values of h_p and h_g used by the OBP are downlinked in telemetry. While the fixed-point

arithmetic of the OBP is not duplicated exactly in this function, the associated quantization effects are approximated in the function through the use of the term b in the equation above. The value of b is half the value of the denominator's least-significant bit to account for the truncation performed in the OBP integer math; neglecting this term would give a height bias of approximately 2 mm.

Step 2: *Range average the simulated data*. The simulated unnormalized interferogram, powers, and reference locations are filtered in the range dimension to obtain 500-m resolution, 250-m posting, cross track ground range bins aligned with the OBP downlinked power and interferogram measurements. The filter used is a continuous Parzen window centered at the center of each ground range bin. For more details see [2]. The window used here is the same as that used by the OBP except that it has been normalized.

Step 3: *Interpolate the simulated data*. The simulated interferogram, powers, and reference locations are interpolated to the actual OBP sample locations in the along track direction (time) using a 16-point sinc interpolator with a Hanning window.

Step 4: *Normalize the measured interferogram.* The measured interferogram is normalized by dividing by the square root of the product of the measured powers. After normalization, the magnitude of the interferogram is the measured correlation.

Step 5: *Normalize the simulated interferogram.* The simulated interferogram is normalized by dividing by the square root of the product of the simulated powers. After normalization, the magnitude of the interferogram is the product of the angular and geometric correlations.

Step 6 *Correct the measured interferogram phase*. The phase bias is removed by multiplying the normalized measured interferogram by the complex conjugate of the simulated interferogram *normalized to unit magnitude*. The phase-bias-corrected measured, normalized interferogram is given by:

$$\Phi_{meas}^{corr}(i_{250m}, j_{250m}, m) = \frac{\Phi_{meas}^{uncorr}(i_{250m}, j_{250m}, m)\Phi_{sim}^{*}(i_{250m}, j_{250m}, m)}{\|\Phi_{sim}(i_{250m}, j_{250m}, m)\|}$$

The along track, i_{250m}, and cross track, j_{250m}, indices denote the ~250-m grid at which the downlinked telemetry is posted. The simulated interferogram, $\Phi_{sim}(i_{250m}, j_{250m}, m)$ is the result of applying steps 1 to 3 to $\Phi(L, k, m)$ from SimulatePhaseBias. The measured uncorrected normalized interferogram, $\Phi_{meas}^{uncorr}(i_{250m}, j_{250m}, m)$, is the result of applying step 4 to the interferogram in downlinked telemetry.

Step 7 *Compute expected standard deviation of phase.* The expected standard deviation of the phase, $\Delta\phi$, is computed analytically using the Cramer-Rao bound equation 24 in [2] reproduced here.

$$\Delta\phi(i_{250m}, j_{250m}, m) = \sqrt{\frac{1 - [C_{tot}(i_{250m}, j_{250m}, m)]^2}{2N_L(j_{250m})[C_{tot}(i_{250m}, j_{250m}, m)]^2}}$$

The total (or measured correlation), C_{tot}, is the magnitude of the normalized, measured interferogram. The effective number of looks, N_L , also referred to as the number of independent looks, is computed by

$$N_L(j_{250m}) = \frac{\Delta_{ct} \, \Delta_{at} \, \sin(\alpha(j_{250m}))}{\Delta_r \Delta_{az}}$$

$$\Delta_r = \frac{c}{2B}$$

$$\Delta_{az} = \frac{9v}{f_p}$$

Δ_{ct}=500-m is the nominal cross track resolution of the OBP outputs. Δ_{at}=500-m is the nominal along track resolution of the OBP outputs. Δ_r=0.75-m is the intrinsic slant range resolution of the SAR processor portion of the OBP before averaging to produce the outputs. Δ_{az}=14.25-m is the typical distance along the ground that SWOT moves between two independent 9-pulse measurement cycles. The sine of $\alpha(j_{250m})$, the angle between nadir and the look vector to the cross track pixel, j_{250m}, is used to covert from slant range resolution to cross track distance. The intrinsic range resolution values are computed from the speed of light, c, and the chirp bandwidth, B.

Step 8 Rescale the measured powers. Powers are rescaled to remove bit shifts that were applied during downlink to optimize the dynamic range of the data. The bit shifts and thus the rescaling factor are a function of cross track pixel. At this point, the scaling due to the unnormalized Parzen window used for range averaging in the OBP is also removed. These bit shifts do not affect the correlation in the OBP since they are done before the interferogram is formed in the OBP so are common to the interferogram and power values.

3.4.5 Accuracy

Because of the coupling between this routine and SimulatePhaseBias, it is semantically ambiguous whether one should assign errors to this routine or that one. For simplicity, errors related to both routines are discussed in the Accuracy section for the SimulatePhaseBias routine.

There are no requirements for the accuracy of power and phase standard deviation calculations. A description of the accuracy of those computations is TBD.

3.5 CorrectCurvature

3.5.1 Purpose

The reference locations computed by SimulatePhaseBias and then interpolated to the OBP pixel spacing in RemovePhaseBias are not on the reference surface due to the curvature of the reference surface. The curvature leads to a vertical bias of 1-2 mm. A 2-D cut illustrating the

curvature effect for a spherical reference surface is provided in Figure 8. The vertical component of this error would cause height errors of the same magnitude in the downstream processing. However, height errors are relatively insensitive to horizontal errors in the reference. The CorrectCurvature module projects the reference locations vertically so that they are always on the reference surface. In Figure 8, the y-coordinate is height above the ellipsoid and the x-coordinate is the length along an arc on the ellipsoid (e.g. cross track).
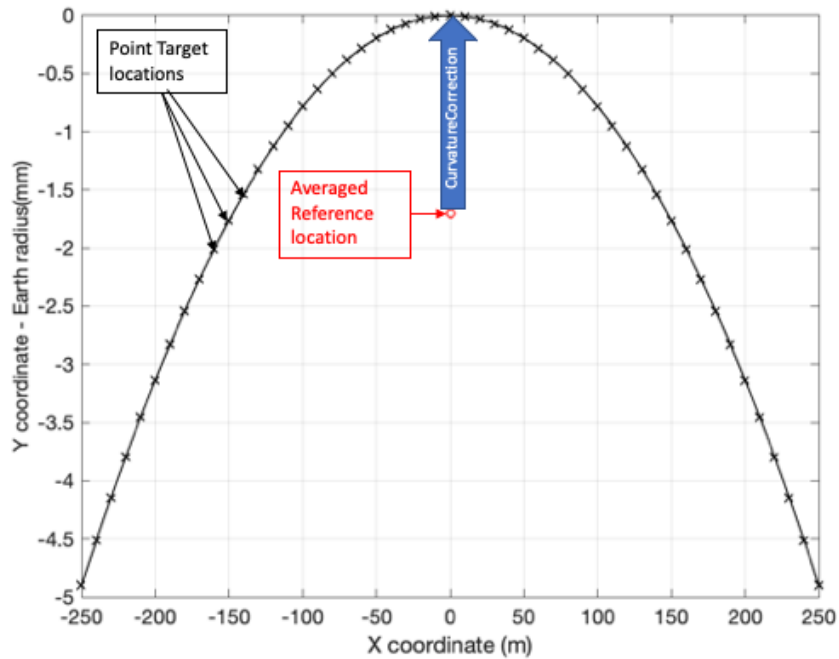
Figure 8 Illustration of surface curvature effect on reference location. Only two dimensions are shown. For a convex surface (e.g. the spheroid Earth) averaging the locations of points on the surface (black x's) results in an average location below that surface. We correct for this bias by translating the integrated reference locations upward to the reference surface. The ~1.6 mm correction shown is typical for a 500-m resolution pixel on the Earth's surface.

3.5.2 Input Data

Description	Source
Reference Surface (PhaseRefSurf)	ConstructRefSurf
Reference locations (RefLocReg) for each measurement beam, along track, and cross track pixel	RemovePhaseBias

3.5.3 Output Data

Description
Reference locations corrected for curvature (RefLocCorr)
Reference surface registered to OBP pixels (PhaseRefSurfReg)

3.5.4 Mathematical Statement

The height above the ellipsoid, longitude and geodetic latitude of each reference location is computed in the standard manner from its cartesian coordinates in ITRF. The corrected height

above the ellipsoid is interpolated from the reference surface table computed in ConstructRefSurf. For every reference location, we solve for the location in the PhaseRefSurf swath-aligned grid using the longitude and geodetic latitude. Next we bilinearly interpolate the PhaseRefSurf to the reference locations. Because of the fine sampling of the reference surface table bilinear interpolation is sufficiently accurate. Finally the corrected height, longitude, and geodetic latitude, are transformed in the standard manner to get the cartesian coordinates of the corrected reference location. The curvature correction only changes the reference location heights without modifying the latitudes and longitudes.

3.5.5 Accuracy

A 1-m reference location height error causes a 1-m error in SSH. Errors in the method used to interpolate the reference surface (i.e. the vertical errors of the reference locations) are less than 0.01-mm, resulting in less than 0.01-mm errors in SSH. The sensitivity of SSH to horizontal error in the reference surface is less than 10^{-5} meters/meter, resulting in less than 0.001 mm of error in SSH. These errors are negligibly small.

3.6 ComputeSigma0

3.6.1 Purpose

This module computes the normalized backscatter cross-section, sigma0, a geophysical quantity that is related to the roughness of the ocean surface, and the wind velocity. Much of the work needed to do this computation is performed upstream in SimulatePhaseBias, which simulates received powers for an assumed model sigma0. This routine computes sigma0 from the ratio of the actual power to the simulated power.

3.6.2 Input Data

Description	Source
Simulated powers registered to OBP pixels (SimPowerReg)	RemovePhaseBias
Geophysical model function table that relates winds to normalized backscatter cross-section, sigma0 (WindGMF)	StaticGeophys
Measured SAR power from both channels (MeasPower)	L0B_LR_Frame
Reference locations corrected for curvature (RefLocCorr)	CorrectCurvature
Position of origin of KMSF frame in ITRF coordinates (KMSFOrigin) $r_0(t)$	ComputeGeometry

3.6.3 Output Data

Description
Normalized backscatter cross section (Sigma0) for each beam, along track, and cross track pixel

> Standard deviation of Sigma0 (Sigma0STD) for each beam, along track, and cross track pixel

3.6.4 Mathematical Statement

The algorithm is applied separately for each half swath.

For each beam, m, cross track, j_{250m}, and along track position, i_{250m}, sigma0 is computed in three steps. (For simplicity of notation in what follows, we replace i_{250m} with i and j_{250m} with j but these indices should not be confused with the finer along track and cross track sampling employed in ComputeRefSurf and SimulatePhaseBias.)

Step 1 Subtract the empirically estimated thermal noise power bias from the measured power. Thermal noise power $P_n(i, j, m)$ is calculated in a manner that is calculated in the Dynamic KaRIn Calibration processor, scaled empirically, and then subtracted from received power in each channel to obtain noise subtracted power $P_{ns1}(i, j, m)$ and $P_{ns2}(i, j, m)$. Subtraction is performed on linear powers (not in dB).

Step 2: Divide the measured power by X. For each channel sigma0 is obtained by dividing the noise-subtracted power by the appropriate X, e.g. $\sigma_{01}(i, j, m) = \frac{P_{ns1}(i,j,m)}{X_1(i,j,m)}$. A single sigma0 value, the arithmetic mean (linear space not dB) of the sigma0 values from both channels, is reported in the data product. Due to random errors inherent in noise subtraction, the linear scale sigma0 values may be negative and the data product is specified to allow this. When averaging high resolution sigma0 values down to coarser resolution, as is performed in the downstream processing, removing negative values or forcing them to zero would bias the result.

Step 3: Compute the expected standard deviation of sigma0. The standard deviation, σ_σ, of sigma0 is computed from the measured signal-to-ratio (SNR) and the number of independent looks, N_L.

$$\sigma_\sigma(i, j, m) = |\sigma_0(i, j, m)| \sqrt{\frac{1 + \frac{2}{|SNR(i,j,m)|} + \frac{1}{SNR(i,j,m)^2}}{N_L}}$$

This equation is equivalent to equation 48 derived in [10] because the number of effective looks is the same as the time bandwidth product and many more effective looks are used to determine the noise power than the signal power. The absolute values around sigma0 and SNR are included to avoid getting negative values in very radar dark areas due to variance in thermal noise power. SNR is the arithmetic mean of the noise subtracted powers in both channels divided by the noise power estimate. It is a linear quantity (not in dB).

3.6.5 Accuracy

The standard deviation reports an estimate of the accuracy of sigma0 given the known instrument noise characteristics. A study to verify that there are no systematic algorithm errors

in the sigma0 estimation large enough to impact SSH accuracy is TBD. Errors in sigma0 lead to errors in wind speed. Errors in wind speed, in turn, impact the accuracy of sea state height bias estimates and thus affect the accuracy of SSH.

3.7 ComputeVolCorr

3.7.1 Purpose

This routine computes the volumetric correlation coefficient [11] which is used to determine significant wave height in the level 2 processing.

3.7.2 Input Data

Description	Source
Measured normalized interferogram with phase bias correction applied (MeasIntfCorr), Simulated normalized interferogram (SimIntfReg) registered to measurements	RemovePhaseBias

3.7.3 Output Data

Description
Volumetric Correlation (VolCor) for each beam, along track, and cross track pixel
Standard deviation of Volumetric Correlation (VolCorSTD) for each beam, along track, and cross track pixel

3.7.4 Mathematical Statement

The measured correlation or *total correlation*, C_{tot}, is the magnitude of the measured complex normalized interferogram MeasIntfCorr. C_{tot} is the product of several different correlation terms: angular correlation, $C_{ang}(i,j,m)$, noise correlation, C_{noise}, geometric correlation, C_{geom}, and volumetric correlation, C_{vol}. This routine computes the other three terms and then obtains

$$C_{vol}(i,j,m) = \frac{C_{tot}(i,j,m)}{C_{ang}(i,j,m)C_{noise}(i,j,m)C_{geom}(i,j,m)}.$$

The product of the angular and geometric correlation terms is the magnitude of the simulated normalized interferogram. The noise correlation term is computed by

$$C_{noise}(i,j,m) = \frac{1}{1+1/SNR(i,j,m)}.$$

The standard deviation, σ_{vol}, of the volumetric correlation is

$$\sigma_{vol}(i,j,m) = \frac{1 - C_{tot}(i,j,m)^2}{\sqrt{2N_L}C_{ang}(i,j,m)C_{noise}(i,j,m)C_{geom}(i,j,m) + \frac{C_{vol}(i,j,m)\sigma_{SNR}(i,j,m)}{|SNR(i,j,k)|(1+|SNR(i,j,k)|)}}$$

The derivation for this equation can be found in Appendix C.

The algorithm to compute the standard deviation, $\sigma_{SNR}(i,j,m)$, of SNR is TBD. For now we assume it is a constant 1 dB (0.26) value.

3.7.5 Accuracy

Errors in volumetric correlation lead to errors in SWH. The current baseline algorithm does not use the KaRIn SWH measurement in computing Sea State Bias, so errors in this routine do not impact meeting the science requirements. For more information about sources of decorrelation see [12].

3.8 ComputeQuality

3.8.1 Purpose

To assign a quality indicator for each beam, along track pixel, and cross track pixel of the product.

3.8.2 Input Data

Description	Source
Measured SAR power from both channels 500-m resolution, 250-m posting (MeasPower), Mitigation SAR power from both channels 250-m resolution, 250-m posting (MitiPower), Mitigation SAR power standard deviation from both channels 250-m resolution, 250-m posting (MitiSTD)	L0B_LR_Frame
Spacecraft attitude with respect to desired pointing (ATT_RPY) expressed as roll, pitch, and yaw.	ComputeGeometry

3.8.3 Output Data

Description
Quality flag for each beam, along track index, and cross track index (QualFlag)

3.8.4 Mathematical Statement

The quality flag is a 32-bit flag for which any bit being set to 1 implies that pixel is either suspect, degraded, or bad (invalid). The exact specification of bits is described in the Product Description Document.

3.8.5 Accuracy

N/A

4   References

[1]   JPL D-61923, "SWOT Science Requirements Document," Jet Propulsion Laboratory Internal Document, 2018.

[2]   JPL D-79130, "KaRIn: Ka-band Radar Interferometer On-Board Processor (OBP) Algorithm Theoretical Basis Document (ATBD)," Jet Propulsion Laboratory Internal Document, 2021.

[3]   JPL D-56405, "SWOT Level 1B KaRIn Low Rate Interferogram Product Description Document," Jet Propulsion Laboratory Internal Document, 2023.

[4]   "IERS Conventions (2010), Gerard Petit and Brian Luzum (eds), IERS Technical Note 35," Verlag des Bundesamts fur Kartographie und Geodasie, Frankfurt am Main, 2010, 179 pp., ISBN 3-89888-989-6.

[5]   S. Houry, "SWOT Precise and Medium-accuracy Orbit Ephemeris data product," SWOT-IS-CDM-0658-CNES, CNES, Toulouse, 2021.

[6]   J. D. Everett, "On Interpolation Formulae," *Quarterly J.,* vol. 32, pp. 306-313, 1900.

[7]   S. Houry, "SWOT Satellite Center of Mass Product Description Document," CNES, SWOT-IS-CDM-1073-CNES, Toulouse, 2021.

[8]   B. Raffier and S. Delavault, "SWOT Reconstructed Attitude Product Description Document," CNES, SWOT-IS-CDM-0684-CNES, Toulouse, 2022.

[9]   JPL D-105502, "SWOT Level 2 Low Rate Sea Surface Height Algorithm Theoretical Basis Document," Jet Propulsion Laboratory Internal Document, 2023.

[10] R. E. Fischer, "Standard Deviation of Scatterometer Measurements from Space," *IEEE Transactions on Geoscience Electronics,* vol. 10, no. 2, pp. 106-113, 1972, doi:10.1109/TGE.1972.271276.

[11] E. Peral and D. Esteban-Fernandez, "SWOT Mission Performance and Error Budget," *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium,* pp. 8625-8628, 2018m doi:10.1109/IGARSS.2018.8517385.

[12] JPL D-79084, "SWOT Mission Performance and Error Budget," Jet Propulsion Laboratory Internal Document, 2017.

[13] M. S. Seymour and I. G. Cumming, "Maximum Likelihood Estimation for SAR Interferometry," *Proceedings of IGARSS '94 - 1994 IEEE International Geoscience and Remote Sensing Symposium,* vol. 4, pp. 2272-2275, 1994, doi:10.1109/IGARSS.1994.399711.

Appendix A. **Acronyms**

AD	Applicable Document
AMR	Advanced Microwave Radiometer
API	Application Interface
ATBD	Algorithm Theoretical Basis Document
CNES	Centre National d'Études Spatiales
JPL	Jet Propulsion Laboratory
NASA	National Aeronautics and Space Administration
OBP	On-Board Processor
PGE	Product Generation Executable
RD	Reference Document
SAS	Science Algorithm Software
SDS	Science Data System
SWOT	Surface Water Ocean Topography
TBC	To Be Confirmed
TBD	To Be Determined

Appendix B.   SimulatePhaseBias Speed Up

The SimulatePhaseBias process computes the following sum which is given in section 3.3.4 where all the variables are defined (R^r below means the same thing as R_{intf}^r in section 3.3.4)

$$\Phi(L, \mathrm{k}, m) = \frac{P_t \sum_{i=i_0(L)}^{i_1(L)} \sum_{j=j_0(k)}^{j_1(k)} A_{ij}^f A(r_{ij}) G_{11}(\vec{x}_{ij}) G_{12}^*(\vec{x}_{ij}) \sigma_0(\vec{x}_{ij}) R^r(\vec{x}_{ij},L,k) R^{az}(\vec{x}_{ij},L,k,m) e^{-\iota \phi_{ij}}}{\sqrt{\frac{P_{r1}(L,k,m) P_{r2}(L,k,m)}{X_{cal1} X_{cal2}}}}$$ B.1

For a given *L, k, m,* the sum is defined over the pixels on a surface *S* (Fig. 7 in section 3.3.4), *i* and *j* are the indices of each pixel in the along and cross track, respectively. To ensure proper sampling when performing this double sum, we need to divide each ground pixel into subpixels such that the slant range difference from one subpixel to the next is less than the sampling resolution. Based on this criterion, the number of subpixels needed for proper sampling is given by the following equation and is illustrated in Figure 9.

$$J = 1 + \mathrm{Int}\left(f_{os} \frac{R_{gdem,res} \times \sin(\theta)}{\delta r_{spacing}}\right)$$ B.2

where

Int = nearest integer

f_{os} = an over sampling factor, which is nominally equal to 10

$R_{gdem,res}$ = the range spacing in the reference surface, which is about 50m

θ = the look incident angle

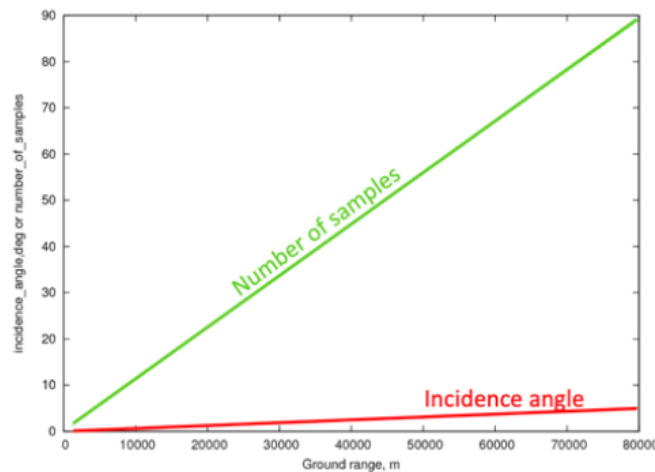$\delta r_{spacing}$ = range spacing, which is ~0.5m

Figure 9 SWOT incident look angle as a function of ground range (red line) and the corresponding number of subpixel (*J*) in (B.2) (green line).

This implies that (B.1) needs to be replaced by the following equation where an additional sum over n is introduced, which adds significant additional computational requirements.

$$\Phi(L,k,m) =$$
$$\frac{P_t \sum_{i=i_0(L)}^{i_1(L)} \sum_{j=j_0(k)}^{j_1(k)} \sum_{n=1}^{J(j)} A_{ijn}^f A(r_{ijn}) G_{11}(\vec{x}_{ijn}) G_{12}^*(\vec{x}_{ijn}) \sigma_0(\vec{x}_{ijn}) R^r(\vec{x}_{ij},L,k) R^{az}(\vec{x}_{ijn},L,k,m) e^{-i\phi_{ijn}}}{\sqrt{\frac{P_{r1}(L,k,m) P_{r2}(L,k,m)}{X_{cal1} X_{cal2}}}} \qquad \text{(B.3)}$$

Substantial saving in the computation of the above sum can be achieved by recognizing that the only fast varying term inside the sum is R_{intf}^r, while all other terms can be modeled either as constants or linearly varying within each pixel. In particular, within a pixel, the product

$$A(r_{ijn}) G_{11}(\vec{x}_{ijn}) G_{12}^*(\vec{x}_{ijn}) \sigma_0(\vec{x}_{ijn})$$

in (B.3) is usually treated as a constant and can be pulled of the sum over n. In addition, the pixel area $A_{ijn}^f = \frac{A_{ij}^f}{J(j)}$ and can be pulled out of the sum over n. Therefore, (B.3) can be written as

$$\Phi(L,k,m) = \frac{P_t \sum_{i=i_0(L)}^{i_1(L)} \sum_{j=j_0(k)}^{j_1(k)} A(r_{ij}) G_{11}(\vec{x}_{ij}) G_{12}^*(\vec{x}_{ij}) \sigma_0(\vec{x}_{ij}) \frac{A_{ij}^f}{J(j)} F(L,k,i,j,m)}{\sqrt{\frac{P_{r1}(L,k,m) P_{r2}(L,k,m)}{X_{cal1} X_{cal2}}}} \qquad \text{(B.4-a)}$$

where

$$F(L,k,i,j,m) = \sum_{n=1}^{J(j)} R^r(n)(\vec{x}_{ijn},L,k) R^{az}(\vec{x}_{ijn},L,k,m) e^{-i\phi_{ijn}} \qquad \text{(B.4-b)}$$

To simplify the notations, we drop the implicit dependence on L,k,i,j,m in (B.4-b), so it reduces to

$$F = \sum_{n=1}^{J} R_n^r R_n^{az} e^{-i\phi_n} \qquad \text{(B.4-c)}$$

Within each pixel, the terms R_n^{az} and ϕ_n can be treated as linear functions of n. The term R_n^r is tabularized as a function of one index $q = r - r_k$, where r is the slant range to a subpixel and r_k is the range bin. For given $L,k,i,j,n,$ there is a unique q. We define

$$q_0 = q(L,k,i,j,n=0)$$

and

$$q_0^+ = q(L,k,i,j+1,n=0)$$

Within each pixel, the index q relates linearly to n; therefore, we can approximate R^{az} and ϕ_n as linear functions of q as follows

$$R_n^{az} = R_0^{az} + (q - q_0)\dot{R}^{az} \qquad \text{(B.5)}$$

where

$$\dot{R}^{az} = \frac{R^{az}\left(\vec{x}_{i,j+1,n=0}, L, k, m\right) - R^{az}\left(\vec{x}_{ij,n=0}, L, k, m\right)}{(q_1 - q_0)}$$

Similarly, we have

$$\phi_n = \phi_0 + (q - q_0)\dot{\phi} \tag{B.6}$$

where

$$\dot{\phi} = \frac{\phi_{i,j+1,n=0} - \phi_{ij,n=0}}{(q_1 - q_0)}$$

From (B.6), we can write

$$e^{-\iota\phi_n} = e^{-\iota(\phi_0 - q_0\dot{\phi})} \sum_{v=0}^{N}\left[c_v\left(q\dot{\phi}\right)^v - \iota s_v\left(q\dot{\phi}\right)^v\right] \tag{B.7}$$

where c_v and s_v are, respectively, the Taylor series expansion coefficients of cosine and sine for small angles up to order N. Making use of (B.5) and (B.7) and rearranging terms, (B.4.c) can be written as

$$F = \sum_{v=0}^{N}\alpha_v\left(\sum_{q=q_0}^{q_0^+} q^v R_q^r\right) = \sum_{v=0}^{N}\alpha_v T(q_0, q_0^+, v) \tag{B.8}$$

In (B.8), α_v is a constant within a pixel and is determined from the various terms in (B.6) and (B.7). $T(q_0, q_0^+, v)$ is given by $\left(\sum_{q=q_0}^{q_0^+} q^v R_q^r\right)$ and only depends on the indices q_0, q_0^+, v. It is independent of the spacecraft relative geometry to the surface or the surface properties, and, therefore, can be computed a priori for any given PTR table. Normally, the Taylor series expansion in (B.7) needs to go up to order $N=6$ in order to achieve the required level of accuracy. For a reference surface with a range resolution of ~50m, the use of this PTR integration table speeds up the process by a factor of 70-80.

Appendix C.   Derivation of Standard Deviation of Volumetric Correlation

The total decorrelation is as follows with i, ,j, and b indices omitted for simplicity

$$C_{\text{tot}} = C_{\text{noise}} \cdot C_{\text{geom}} \cdot C_{\text{ang}} \cdot C_{\text{vol}}$$

One can thus estimate the volumetric decorrelation by

$$\hat{C}_{\text{vol}} = \frac{\hat{C}_{\text{tot}}}{\hat{C}_{\text{noise}} \cdot C_{\text{geom}} \cdot C_{\text{ang}}}$$

Given the estimation of the noise decorrelation,

$$\frac{1}{\hat{C}_{\text{noise}}} = 1 + \frac{1}{\widehat{\text{SNR}}} = 1 + K$$

we have

$$\hat{C}_{\text{vol}} = \frac{\hat{C}_{\text{tot}} + K\hat{C}_{\text{tot}}}{C_{\text{geom}} \cdot C_{\text{ang}}}$$

The square of the mean is

$$\langle \hat{C}_{\text{vol}} \rangle = \frac{\langle \hat{C}_{\text{tot}} \rangle + \langle K\hat{C}_{\text{tot}} \rangle}{C_{\text{geom}} \cdot C_{\text{ang}}}$$

$$\langle \hat{C}_{\text{vol}} \rangle^2 = \frac{\langle \hat{C}_{\text{tot}} \rangle^2 + \langle K\hat{C}_{\text{tot}} \rangle^2 + 2\langle \hat{C}_{\text{tot}} \rangle \langle K\hat{C}_{\text{tot}} \rangle}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2}$$

The second moment is

$$\langle \hat{C}_{\text{vol}}^2 \rangle = \frac{\langle \hat{C}_{\text{tot}}^2 \rangle + \langle K^2 \hat{C}_{\text{tot}}^2 \rangle + 2\langle K\hat{C}_{\text{tot}}^2 \rangle}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2}$$

The variance of the volumetric decorrelation estimation is

$$\sigma_{\text{vol}}^2 = \frac{\langle \hat{C}_{\text{tot}}^2 \rangle + \langle K^2 \hat{C}_{\text{tot}}^2 \rangle + 2\langle K\hat{C}_{\text{tot}}^2 \rangle}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2} - \langle \hat{C}_{\text{vol}} \rangle^2$$

$$\sigma_{\text{vol}}^2 = \frac{\langle \hat{C}_{\text{tot}}^2 \rangle (1 + 2\langle K \rangle + \langle K^2 \rangle) + 2\,\text{cov}(K, \hat{C}_{\text{tot}}^2) + \text{cov}(K^2, \hat{C}_{\text{tot}}^2)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2} - \langle \hat{C}_{\text{vol}} \rangle^2$$

$$\sigma_{\text{vol}}^2 = \frac{\sigma_{\text{tot}}^2(1 + 2\langle K \rangle + \langle K^2 \rangle)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2} + \frac{\langle \hat{C}_{\text{tot}} \rangle^2(1 + 2\langle K \rangle + \langle K^2 \rangle)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2} + \frac{2\,\text{cov}(K, \hat{C}_{tot}^2) + \text{cov}(K^2, \hat{C}_{tot}^2)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2}$$
$$- \langle \hat{C}_{\text{vol}} \rangle^2$$

From $\langle \hat{C}_{\text{vol}} \rangle = \frac{\langle \hat{C}_{\text{tot}} \rangle + \langle K \hat{C}_{\text{tot}} \rangle}{C_{\text{geom}} \cdot C_{\text{ang}}} = \frac{\langle \hat{C}_{\text{tot}} \rangle(1 + \langle K \rangle) + \text{cov}(K, \hat{C}_{tot})}{C_{\text{geom}} \cdot C_{\text{ang}}}$

$$\langle \hat{C}_{\text{tot}} \rangle^2 = \frac{\left(\langle \hat{C}_{\text{vol}} \rangle \cdot C_{\text{geom}} \cdot C_{\text{ang}} - \text{cov}(K, \hat{C}_{\text{tot}})\right)^2}{(1 + \langle K \rangle)^2}$$

The centered variance is

$$\sigma_{\text{vol}}^2 = \frac{\sigma_{\text{tot}}^2(1 + 2\langle K \rangle + \langle K^2 \rangle)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2} + \langle \hat{C}_{\text{vol}} \rangle^2 \left(\frac{1 + 2\langle K \rangle + \langle K^2 \rangle}{(1 + \langle K \rangle)^2} - 1\right)$$
$$+ \text{cov}(K, \hat{C}_{\text{tot}})\left(\text{cov}(K, \hat{C}_{\text{tot}}) - 2\langle \hat{C}_{\text{vol}} \rangle\right)\left(\frac{1 + 2\langle K \rangle + \langle K^2 \rangle}{(1 + \langle K \rangle)^2}\right)$$
$$+ \frac{2\,\text{cov}(K, \hat{C}_{\text{tot}}^2) + \text{cov}(K^2, \hat{C}_{\text{tot}}^2)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2}$$

By definition

$$\sigma_K^2 \stackrel{\text{def}}{=} \langle K^2 \rangle - \langle K \rangle^2$$

Then

$$\sigma_{\text{vol}}^2 = \frac{\sigma_{\text{tot}}^2(1 + 2\langle K \rangle + \sigma_K^2 + \langle K \rangle^2)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2} + \frac{\langle \hat{C}_{\text{vol}} \rangle^2 \sigma_K^2}{(1 + \langle K \rangle)^2}$$
$$+ \text{cov}(K, \hat{C}_{\text{tot}})\left(\text{cov}(K, \hat{C}_{\text{tot}}) - 2\langle \hat{C}_{\text{vol}} \rangle\right)\left(\frac{(1 + 2\langle K \rangle + \sigma_K^2 + \langle K \rangle^2)}{(1 + \langle K \rangle)^2}\right)$$
$$+ \frac{2\,\text{cov}(K, \hat{C}_{\text{tot}}^2) + \text{cov}(K^2, \hat{C}_{\text{tot}}^2)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2}$$

According to the Cramer Rao bound from [13]

$$\sigma_{\text{tot}}^2 = \frac{\left(1 - \hat{C}_{\text{tot}}^2\right)^2}{2N_L}$$

Thus, the standard deviation is

$$\sigma_{\text{vol}}^2 = \frac{\left(1 - \hat{C}_{\text{tot}}^2\right)^2}{2N_L C_{\text{geom}}^2 \cdot C_{\text{ang}}^2}(1 + \langle K \rangle)^2$$
$$+ \sigma_K^2 \left(\frac{\langle \hat{C}_{vol} \rangle^2}{(1 + \langle K \rangle)^2} + \frac{\left(1 - \hat{C}_{\text{tot}}^2\right)^2}{2N_L C_{\text{geom}}^2 \cdot C_{\text{ang}}^2} + \frac{\text{cov}(K, \hat{C}_{\text{tot}})\left(\text{cov}(K, \hat{C}_{\text{tot}}) - 2\langle \hat{C}_{\text{vol}} \rangle\right)}{(1 + \langle K \rangle)^2}\right)$$
$$+ \text{cov}(K, \hat{C}_{\text{tot}})\left(\text{cov}(K, \hat{C}_{\text{tot}}) - 2\langle \hat{C}_{\text{vol}} \rangle\right) + \frac{2\,\text{cov}(K, \hat{C}_{\text{tot}}^2) + \text{cov}(K^2, \hat{C}_{\text{tot}}^2)}{C_{\text{geom}}^2 \cdot C_{\text{ang}}^2}$$

Note: N_L is the *effective* number of looks also known as the number of independent. Other common uses of number of looks (outside this document) are not equivalent.

The mean and variance of the inverse of the estimated SNR are approximated with first order Taylor expansion to

$$\langle K \rangle \simeq \frac{1}{\langle \widehat{SNR} \rangle}$$

$$\sigma_K^2 = \frac{\sigma_{\widehat{SNR}}^2}{\langle \widehat{SNR} \rangle^4}$$

Finally

$$\sigma_{vol}^2 \simeq \frac{\left(1 - \hat{C}_{tot}^2\right)^2}{2N_L C_{geom.}^2 C_{ang}^2} \left(\frac{1 + \langle \widehat{SNR} \rangle}{\langle \widehat{SNR} \rangle}\right)^2$$

$$+ \frac{\sigma_{\widehat{SNR}}^2}{\langle \widehat{SNR} \rangle^4} \left(\left(\langle \hat{C}_{vol} \rangle^2 + cov(K, \hat{C}_{tot}) \left(cov(K, \hat{C}_{tot}) - 2\langle \hat{C}_{vol} \rangle\right)\right) \left(\frac{\langle \widehat{SNR} \rangle}{1 + \langle \widehat{SNR} \rangle}\right)^2 \right.$$

$$\left. + \frac{\left(1 - \hat{C}_{tot}^2\right)^2}{2N_L C_{geom.}^2 C_{ang}^2} \right) + cov(K, \hat{C}_{tot}) \left(cov(K, \hat{C}_{tot}) - 2\langle \hat{C}_{vol} \rangle\right)$$

$$+ \frac{2\, cov(K, \hat{C}_{tot}^2) + cov(K^2, \hat{C}_{tot}^2)}{C_{geom.}^2 C_{ang}^2}$$

The estimation of σ_{vol}^2 will use $\widehat{SNR}$ as an estimate of $\langle \widehat{SNR} \rangle$ and therefore $\hat{C}_{noise}$ as an estimate of $\frac{\langle \widehat{SNR} \rangle}{1 + \langle \widehat{SNR} \rangle}$. In practice, the knowledge of $cov(K, \hat{C}_{tot})$, $cov(K, \hat{C}_{tot}^2)$, $cov(K^2, \hat{C}_{tot}^2)$ is not trivial. The hypothesis of negligible contribution of these terms is made for the final expression of the estimated variance. The validity of the hypothesis is tested with simulations results; the model shows some reasonable differences with the simulation at external beam, but substantial difference for the central beam, the lower the SNR, the better the model.

The estimator for the standard deviation of the volumetric coherence is

$$\hat{\sigma}_{vol}^2 \simeq \frac{\left(1 - \hat{C}_{tot}^2\right)^2}{2N_L C_{geom.}^2 C_{ang}^2 \hat{C}_{noise}^2} + \sigma_{\widehat{SNR}}^2 \left[\frac{\hat{C}_{vol}^2}{\widehat{SNR}^2 \left(1 + \widehat{SNR}\right)^2} + \frac{\left(1 - \hat{C}_{tot}^2\right)^2}{2N_L C_{geom.}^2 C_{ang}^2 \widehat{SNR}^4} \right]$$
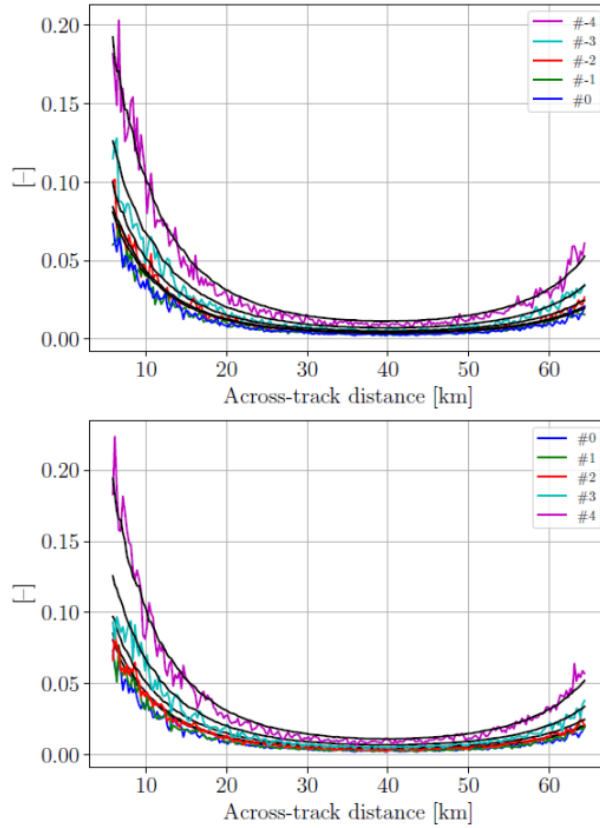
Figure 10 Standard deviation of volumetric coherence from end-to-end simulated data (colored lines) versus estimated standard deviation from the model for a constant SWH simulation. Different colors are different beams. Number 0 is the center beam (denoted as beam 5 in other figures); numbers 4 and -4 are the outer two beams. Colored lines are the standard deviation of measured volumetric correlations over all along track samples. Black lines are the analytically determined volumetric correlation uncertainty as reported in the product.