



# SWIM L2S Product

Algorithm Theoretical Basis Document

Ifremer Wind & Wave Operational Center



# Modifications

Ver.	Rev.	Date	Object
1	0	03/07/15	Document creation: <ul style="list-style-type: none"> <li>• ATBD basis for the V1 prototype</li> <li>• Theoretical content relative to the directional properties of CFOSAT radar signal</li> <li>• Theoretical content relative to the range bunching effect</li> <li>• Methodology of the ribbon partitioning</li> </ul>
1	1	11/12/15	<ul style="list-style-type: none"> <li>• Move into a separate document the theoretical content relative to the directional properties and to the range bunching</li> <li>• Reorganize the document in order to clearly distinguish:               <ul style="list-style-type: none"> <li>◦ the current implementation of the prototype: all is now described in “mathematical description” sections</li> <li>◦ the open points (+ modified ones and new ones)</li> </ul> </li> <li>• Initiate a product chapter</li> <li>• add parameter <math>\phi_r</math> in section 8.3.6</li> </ul>
2	0	14/11/16	<ul style="list-style-type: none"> <li>• Add section 4 for ancillary data: land mask, sea-ice concentration and bathymetry</li> <li>• Major modifications of fluctuation spectrum computation (section 5 now): add masking and modify resampling, detrending and spectral transform</li> <li>• Update SWIM chronogram parameters (Table 8) with new SWIM PRF and Nimp values</li> </ul>
3	0	20/11/17	<ul style="list-style-type: none"> <li>• Add missing mathematical descriptions in section 5.3</li> <li>• Modify sections 4, 5, 6, 7 according to the current status of L2S prototype</li> <li>• Add an example of “L2S expert” output in section 9</li> </ul>

# Table of Contents

Open points.....	7
List of TBC and TBD parameters.....	8
1 Introduction.....	9
1.1 Introduction.....	9
1.2 Document overview.....	9
1.3 Acronyms and definitions.....	9
1.3.1 Acronyms.....	9
1.3.2 Definitions.....	10
1.3.3 Mathematical notations.....	10
1.4 Documentation.....	12
1.4.1 Reference documents.....	12
2 General overview of mission and instrument.....	13
2.1 CFOSAT mission.....	13
2.2 SWIM instrument.....	13
2.2.1 Instrument overview.....	13
2.2.2 Physical measurements.....	15
3 General overview of SWIM L2S processing.....	18
3.1 Description.....	18
3.2 Main block diagram.....	19
4 Step 1: Ancillary data processing.....	20
4.1 Land mask.....	20
4.2 Sea-ice concentration.....	22
4.3 Bathymetry.....	24
4.4 Model wind.....	25
5 Step 2: Fluctuation spectrum computation.....	26
5.1 Theoretical description.....	26
5.1.1 Bright target removal.....	26
5.1.2 Regular ground range resampling.....	26
5.1.3 Detrending.....	29
5.1.4 Spectral transform.....	30
5.1.5 Post-processing.....	33
5.2 Block diagram and processing parameters.....	33
5.3 Mathematical description.....	34
5.3.1 regular_ground_range_locations.....	34
5.3.2 sinc_resample.....	35
5.3.3 gaussian_lowpass.....	37
5.3.4 segment_positions.....	37
5.3.5 spectral_transform.....	38
6 Step 3: Modulation spectrum computation.....	40
6.1 Theoretical description.....	40
6.1.1 Method 0 (simulation based).....	40
6.1.2 Method 1 (not yet implemented nor tested).....	40
6.2 Block diagram.....	41
6.3 Mathematical description.....	41
7 Step 4: Wave spectrum computation.....	42
7.1 Theoretical description.....	42
7.1.1 Estimate MTF - method 0 (“theoretical” tilt MTF).....	42

7.1.2	Estimate MTF - method 1 (not implemented).....	43
7.2	Block diagram.....	43
7.3	Mathematical description.....	43
7.3.1	Estimate MTF - method 0 (“theoretical” tilt MTF).....	43
8	Step 5: Ribbon partitioning.....	45
8.1	Theoretical description.....	45
8.2	Block diagram.....	47
8.3	Mathematical description.....	48
8.3.1	k-resample.....	48
8.3.2	smooth spectrum.....	49
8.3.3	estimate noise level.....	50
8.3.4	apply watershed.....	51
8.3.5	merge partitions.....	52
8.3.6	discard partitions.....	54
8.3.7	compute partitions parameters.....	55
9	SWIM L2S product.....	57

## Index of Figures

Figure 1: SWIM illumination pattern.....	15
Figure 2: Example of the three different scientific results from SWIM backscatter coefficient processing.....	16
Figure 3: Main SWIM operating modes.....	17
Figure 4: Main block diagram.....	19
Figure 5: Exemple of SWIM land mask.....	22
Figure 6: Example of sea-ice concentration onto SWIM.....	23
Figure 7: Example of original sea-ice concentration.....	23
Figure 8: Example of bathymetry onto SWIM.....	24
Figure 9: Example of original bathymetry.....	25
Figure 10: Example of sigma0 resampling at near range.....	28
Figure 11: Example of sigma0 resampling at far range.....	29
Figure 12: Example of sigma0 trend.....	30
Figure 13: Example of fluctuation spectrum.....	32
Figure 14: Block diagram of fluctuation spectrum computation.....	34
Figure 15: Block diagram of modulation spectrum computation.....	41
Figure 16: Block diagram of wave spectrum computatino.....	43
Figure 17: Block diagram of ribbon partitioning.....	48

## Index of Tables

Table 1: TBC parameters.....	8
Table 2: TBD parameters.....	8
Table 3: Acronyms.....	10
Table 4: Definitions.....	10
Table 5: Mathematical notations.....	11
Table 6: Reference documents.....	12
Table 7: SWIM main parameters.....	14
Table 8: SWIM chronogram parameters.....	17
Table 9: Nominal values of spacings, coverage and resolutions for each beam in normal mode.....	27
Table 10: Processing parameters of fluctuation spectrum computation.....	34

# Open points

*Formatting note: open points may be accessed directly from the table below. They are mainly ideas to discuss, clarify and implement for the future versions of the prototype. Indicated with orange color, they are described (more or less) in the “theoretical description” sections. Regarding the methods implemented in the current prototype, they are in normal text and are described in the “theoretical description” and “mathematical description” sections.*

## Table of Open Points

Open Point 1	More ancillary data.....	20
Open Point 2	Bright target removal.....	26
Open Point 3	Long swell weak signatures (spectral domain).....	33
Open Point 4	NRCS heterogeneity (remove low freq).....	33
Open Point 5	Speckle LUT.....	40
Open Point 6	IR from instrument.....	41
Open Point 7	Directional spread.....	41
Open Point 8	MTF.....	43
Open Point 9	Partitioning parameters.....	46
Open Point 10	Long swell weak signatures (partitioning).....	46
Open Point 11	Propagation ambiguity removal.....	46
Open Point 12	Merge beam.....	47

## List of TBC and TBD parameters

Section	Description

*Table 1: TBC parameters*

Section	Description

*Table 2: TBD parameters*



# 1 Introduction

## 1.1 Introduction

This document is intended to specify the physical and mathematical description of the algorithms to be used in the generation of SWIM L2S products at IWWOC center.

## 1.2 Document overview

The structure of the document is as follows:

- Chapter 1 gives information about acronyms, definitions, mathematical notations and reference documents.
- Chapter 2 presents the CFOSAT mission context and the SWIM instrument.
- Chapter 3 contains the general description of L2S processing.
- Chapters 4 to 8 describe in detail the different steps of the L2S processing.
- Chapter 9 gives information about L2S product.

## 1.3 Acronyms and definitions

### 1.3.1 Acronyms

<b>Acronym</b>	<b>Signification</b>
CERSAT	Centre ERS d'Archivage et de Traitement (Ifremer)
CFOSAT	China France Oceanography SATellite
CNES	Centre National d'Études Spatiales
CNSA	Chinese National Space Administration
CWWIC	CNES Waves & Wind Instrument Center
FROGS	FRench Oceanographic Ground Segment
Ifremer	Institut Francais de Recherche pour l'Exploitation de la Mer
IRF	Impulse Response Function
IWWOC	Ifremer Wind & Wave Operational Center
LUT	LookUp Table
MTF	Modulation Transfer Function
NRCS	Normalized Radar Cross-Section
NRT	Near Real-Time
PRF	Pulse Repetition Frequency
SCAT	SCATterometer
SWIM	Surface Waves Investigation and Monitoring

<b>Acronym</b>	<b>Signification</b>
TBC	To Be Confirmed
TBD	To Be Defined

Table 3: Acronyms

### 1.3.2 Definitions

<b>Expression</b>	<b>Signification</b>

Table 4: Definitions

### 1.3.3 Mathematical notations

<b>Notation</b>	<b>Definition</b>	<b>Units</b>
$c_{seaice}$	Sea-ice concentration	%
$\delta r$	Slant range resolution	m
$\delta x$	Ground range resolution	m
$\delta x_i$	Effective ground range resolution	m
$\Delta k$	Wavenumber spacing	rad/m
$\Delta \phi$	Azimuth spacing	rad
$\Delta r$	Slant range spacing	m
$\Delta \sigma_0$	NRCS fluctuations	<i>unitless</i>
$\Delta x$	Ground range spacing	m
$e_{bathy}$	Bathymetry elevation	m
$hs$	Significant wave height	m
$k$	Wavenumber	rad/m
$L$	Partition label	<i>unitless</i>
$\lambda$	Wavelength	m
$L_{dis}$	Number of range gates used for on-board average	<i>unitless</i>
$L_x$	Ground range coverage	m
$L_y$	Footprint azimuthal ground length (at 3dB)	m
$N_{imp}$	number of pulses used for on-board average	<i>unitless</i>
$N_k$	Number of wavenumbers	<i>unitless</i>
$N_r$	Number of pixels in slant range dimension	<i>unitless</i>
$N_x$	Number of pixels in ground range dimension	<i>unitless</i>

<b>Notation</b>	<b>Definition</b>	<b>Units</b>
$\varphi$	Azimuth angle	rad
$\sigma_0$	NRCS (Normalized Radar Cross Section)	<i>unitless</i>
$\sigma_{0t}$	NRCS trend	<i>unitless</i>
$\sigma_N$	Wave spectrum noise level	m <sup>2</sup>
$S_f$	Fluctuation spectrum	m
$S_{ir}$	Impulse response spectrum	<i>unitless</i>
$S_s$	Wave slope spectrum	m <sup>2</sup>
$S_{sp}$	Speckle spectrum	m
$S_m$	Modulation spectrum	m
$T$	MTF (Modulation Transfer Function)	m <sup>-1</sup>
$\theta$	Incidence angle	rad
$\theta_c$	Incidence angle at middle swath	rad
$U$	Wind speed	m/s

*Table 5: Mathematical notations*

## 1.4 Documentation

### 1.4.1 Reference documents

Notation	Reference
[RD1]	Algorithm Theoretical Baseline Document - Traitement des signaux SWIM - Niveau L1A T. Grelier CF-GSFR-SP-802-CNES
[RD2]	Algorithm Theoretical Baseline Document - Traitement des signaux SWIM - Niveau L1B des produits "vagues" D. Hauser, L. Delaye and N. Lamquin CF-GSFR-SP-803-CNES
[RD3]	Algorithm Theoretical Baseline Document - SWIM signal processing - Preliminary specification of L2A non nadir level L. Delaye, N. Lamquin and D. Hauser CF-GSFR-SP-804-CNES
[RD4]	Portilla J., F. J. Ocampo-Torres and J. Monbaliu 2009: Spectral Partitioning and identification of Wind Sea and Swell Journal of Atmospheric and Oceanic Technology, 26, 107-122
[RD5]	Study on range bunching effect and directional properties of SWIM signal IWWOC
[RD6]	Arctic & Antarctic sea ice concentration and Arctic sea ice drift estimated from special sensor microwave data User's manual, version 2.1, February 2007 Robert Ezraty, Fanny Girard-Ardhuin, Jean-François Piollé

*Table 6: Reference documents*

## 2 General overview of mission and instrument

### 2.1 CFOSAT mission

The CFOSAT program is carried out through cooperation between French and Chinese Space Agencies (CNES and CNSA respectively). CFOSAT aims at characterizing the ocean surfaces to better model and predict the ocean states and improve the knowledge in ocean / atmosphere exchanges. The CFOSAT products will help for marine and weather forecast and for climate monitoring.

The CFOSAT satellite will embark two payloads: SCAT, a wind scatterometer, and SWIM, a wave scatterometer to allow a joint characterization of ocean surface winds and waves.

The SWIM instrument delivered by CNES is dedicated to the measurement of the directional wave spectrum (height, direction and periodicity of waves). It will contribute to improve knowledge in the following fields:

- directional wave spectrum
- modeling and prediction of ocean surface wind and waves
- physical processes associated with surface wave evolution
- wind wave interactions
- interactions between surface waves and atmosphere or ocean
- wave evolution in coastal region
- polar ice sheet
- sea ice and iceberg movement monitoring
- land surfaces (surface soil moisture, soil roughness...)

As parts of the French ground segment (FROGS managed by CNES), CNES and Ifremer will deliver SWIM products:

- NRT instrumental and geophysical products (L0 to L2) will be processed at CWWIC (CNES). The data must be provided to users (meteorology agencies mainly) in NRT, i.e. in less than three hours from the acquisition.
- delayed time geophysical products (L2 to L4) will be processed at IWWOC (Ifremer).

### 2.2 SWIM instrument

#### 2.2.1 Instrument overview

SWIM is a Ku-band (13.575 GHz) real aperture radar that illuminates the surface

sequentially with 6 incidence angles: 0°, 2°, 4°, 6°, 8° and 10° with an antenna aperture of approximately 2°. In order to acquire data in all azimuth orientations, the antenna is rotating at a speed rate of 5.6 rpm.

SWIM aims at measuring the modulation of the backscattering coefficient of the sea surface which is linked to long waves. Actually, long waves (i.e. wavelength greater than a few decimeters) create slopes on the sea surface which will modify the backscattering compared to flat surfaces. The microwave backscatter from the sea occurs by means of quasi-specular reflections from wave facets oriented normal to the radar's line of sight. The modulation signal shall be mostly dependent on waves (i.e. long slopes) and not on surface wind (i.e. small roughness) for the incidence angles around 8° (see electromagnetic models). Radar measurements in Ku-band from TRMM and ENVISAT show that the backscattering coefficient of the surface is independent from small roughness (and, as consequence, from wind) around 8° to 10° of incidence angles (see Figure 2b). At these incidence angles, only the long slopes modulate the signal. Besides, only those surface waves whose phase fronts are "matched" to the satellite line of sight will survive the lateral averaging on azimuth. Therefore, a conical scan around nadir angle allows detecting the direction of the waves.

Table 7 summarizes the main parameters of SWIM. The six beams are illuminated sequentially with different PRF values for each incidence angle.

Frequency	13.575 GHz
Useful bandwidth	320 MHz
Useful pulse duration	50 $\mu$ s
Peak power	120 W
Central elevation angles (on board)	0° - 2.3° - 3.7° - 5.55° - 7.4° - 9.25°
PRF	from 2 kHz to 6.3 kHz
Antenna rotation speed	5.6 rpm
Antenna diameter	90 cm
Antenna 3 dB aperture at nadir and 2°	1.6°
Antenna 3 dB aperture at 4°, 6°, 8° and 10°	> 1.75°
Polarization	Linear polarization and rotating

*Table 7: SWIM main parameters*

In a similar way as the Poseidon altimeter, the nadir beam allows to measure the distance between the satellite and the sea surface, and measures the backscattering coefficient,  $\sigma_0$ , and its evolution in the swath. That allows estimating the significant wave height (SWH) and the wind speed (WS). The nadir processing is similar to a conventional Poseidon altimeter with searching and locking loops, but digital techniques are used in this case for the range compression.

The nadir beam allows the general synchronization of the instrument since it gives its relative altitude. This beam is needed for the synchronization and sequential piloting of the other beams ( $2^\circ$  to  $10^\circ$ ).

The five non nadir beams allow measuring the backscattering coefficient and creating a  $\sigma_0$  profile as function of the incident angle. The  $6^\circ$ ,  $8^\circ$  and  $10^\circ$  beams are called the “spectrum beams” and they are used to obtain the wave modulation spectrum (which is directly linked to the directional wave spectrum)

Figure 1 exhibits the geometry of the SWIM acquisition.

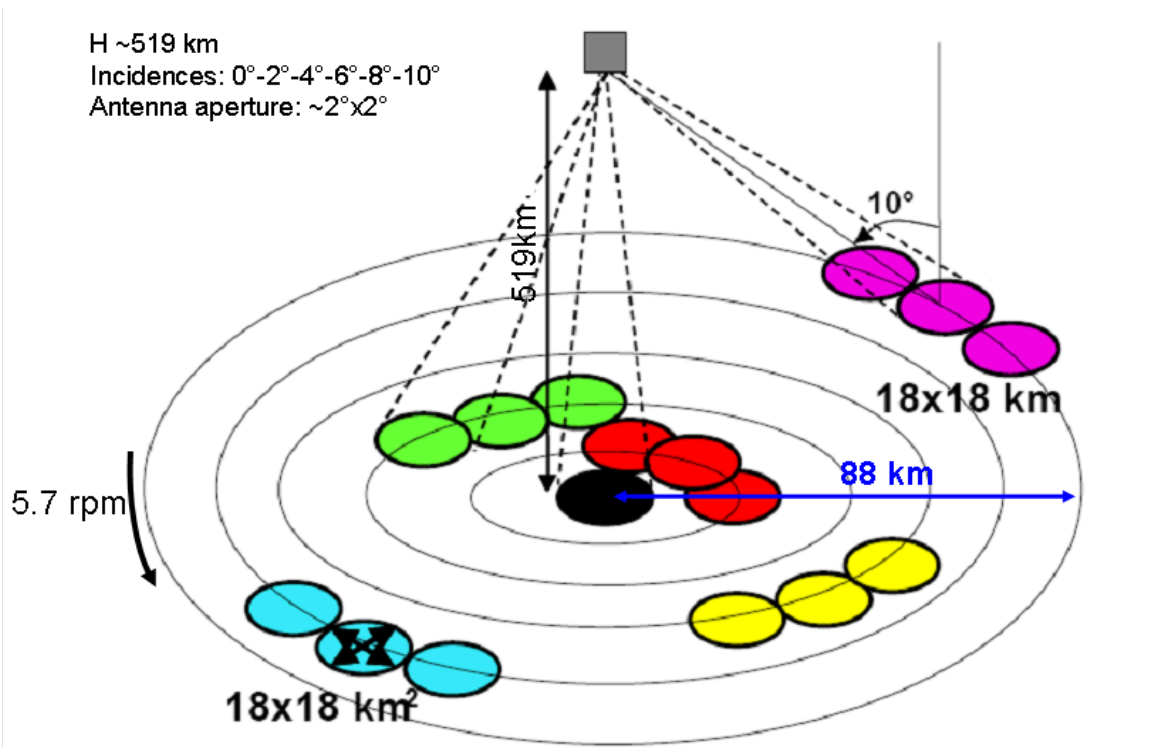
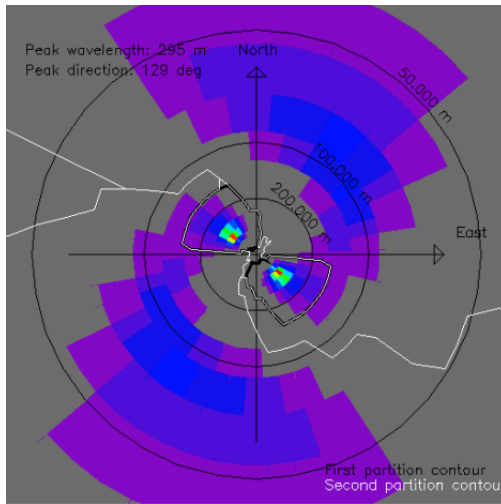


Figure 1: SWIM illumination pattern

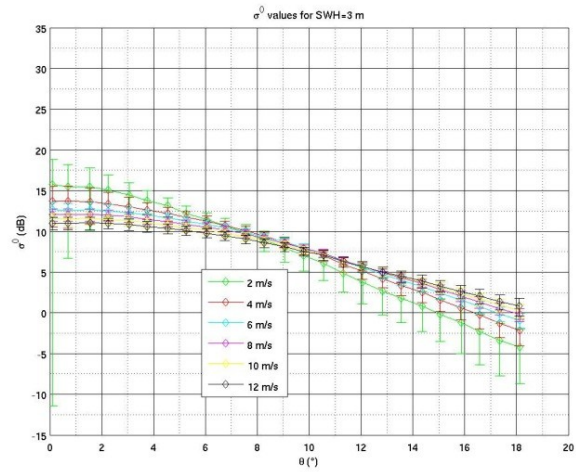
### 2.2.2 Physical measurements

The SWIM products contain three main kinds of variables (see Figure 2):

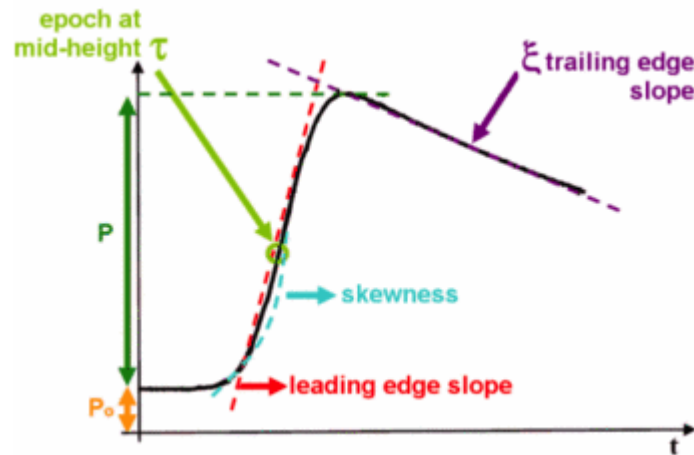
- The backscattering profile  $\sigma_0$  from  $0^\circ$  to  $10^\circ$
- The wave spectrum from beams  $6^\circ$  to  $10^\circ$
- SWH and WS from the nadir beam



(a) 2D wave spectrum



(b) Backscattering coefficient profiles



(c) Nadir waveform for SWH and WS estimation

Figure 2: Example of the three different scientific results from SWIM backscatter coefficient processing

Different operational modes have been defined for SWIM (see Figure 3). The scientific data are obtained in TRACKING mode and are processed by the CWWIC. In TRACKING mode, the instrument illuminates successively the incident angles specified by the macro-cycle as defined in Table 8.

The measurement of  $N_{imp}$  impulsions at a given incidence is a cycle. The time duration spent to cover all the beams is named macro-cycle. These macro-cycles can be:

- $\{0^\circ, 2^\circ, 4^\circ, 6^\circ, 8^\circ, 10^\circ\}$  (nominal macro-cycle),
- $\{0^\circ, 6^\circ, 8^\circ, 10^\circ\}$ ,
- $\{0^\circ, 8^\circ, 10^\circ\}$ ,
- $\{0^\circ, 10^\circ\}$  or  $\{0^\circ, 10^\circ, 10^\circ\}$  or  $\{0^\circ, 10^\circ, 10^\circ, 10^\circ\}$  (TBC),
- $\{0^\circ, 8^\circ\}$  or  $\{0^\circ, 8^\circ, 8^\circ\}$  or  $\{0^\circ, 8^\circ, 8^\circ, 8^\circ\}$  (TBC).



The order of the beams must always appear in the products, from L1A product onwards. Although some of the extracted parameters are only produced for the spectral beams the order is reminded by indexing parameters by the order of acquisition within the macro-cycle. Thus, each beam-depending parameter is suffixed by “\_0”, “\_1”,... “\_n\_beam” where “n\_beam” is the amount of beams (-1 if starts at 0) which cannot exceed 6 and the first cycle is always a nadir cycle. If a parameter is only defined for the spectral beams then these numbers are kept as a reference. In the case of the nominal macro-cycle these parameters will be suffixed by “\_3”, “\_4”, and “\_5”. There is only one type of macro-cycle per product and no confusion is possible.

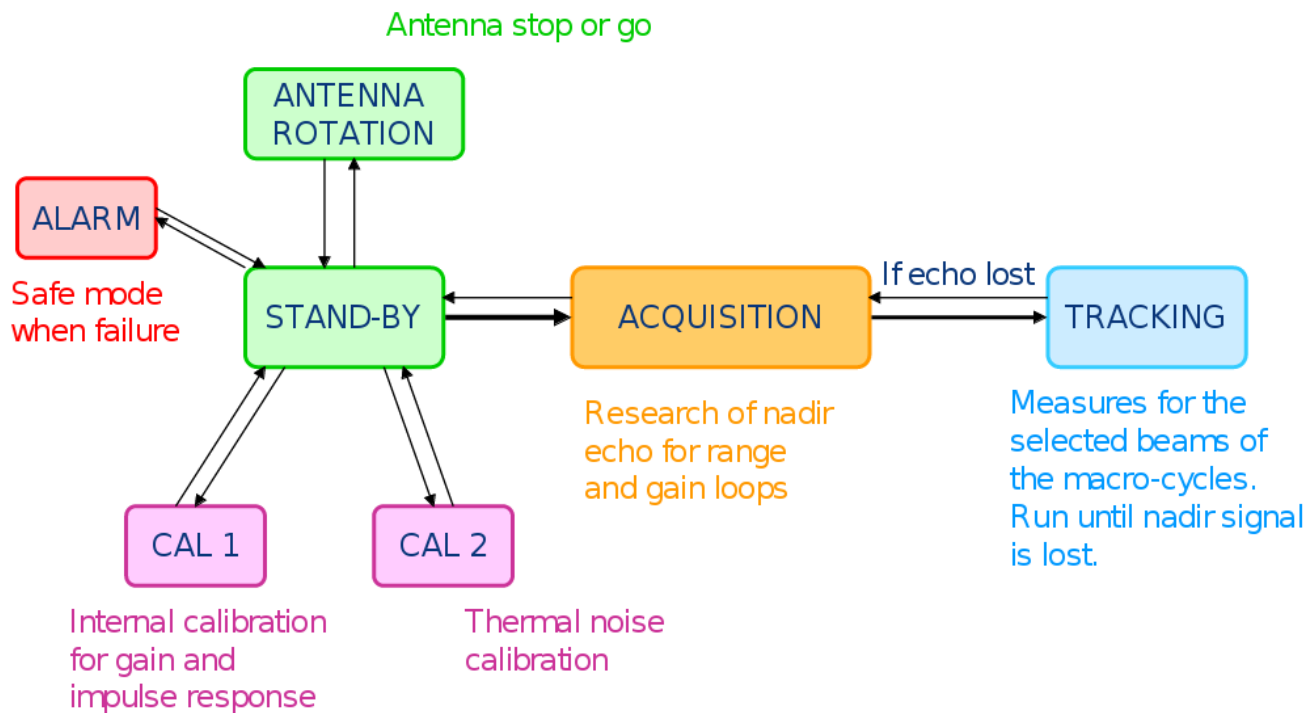


Figure 3: Main SWIM operating modes

	0°	2°	4°	6°	8°	10°
Ambiguity rank	18	18	18	18	18	18
Min PRF (Hz)	5093	5079	5079	5065	5037	5023
Max PRF (Hz)	5427	5427	5411	5395	5379	5348
N <sub>imp</sub>	264	97	97	156	186	204
Max integration time length (ms)	51.8	19.1	19.1	30.8	36.9	40.6
Min cycle length (ms)	52.0	21.2	21.3	32.3	37.9	41.5
Max cycle length (ms)	52.4	22.6	22.6	34.4	40.5	44.2
On-board averaging L <sub>dis</sub>	1	4	4	2	3	3

Table 8: SWIM chronogram parameters

## 3 General overview of SWIM L2S processing

### 3.1 Description

The main objective of SWIM L2 products is to provide directional wave spectra measures as well as the integrated parameters of the associated wave systems. The SWIM L2 level is common to CWWIC and IWWOC centers:

- SWIM L2 product processed at CWWIC in NRT (see [RD2] and [RD3])
- SWIM L2S product processed at IWWOC in delayed time (this document) and serving as input for IWWOC L3 (statistics) and L4 (propagated waves and storm sources) products.

The particular motivations of the L2S processing are to:

- take advantage of the use of various ancillary data which are in NRT either not available (e.g. observations) or not the best estimates (e.g. model forecast). Ancillary data are particularly useful in order to increase the accuracy of the wave retrieval algorithm. Delayed time offers freedom to use suitable ancillary data, for example: ice mask product, best estimates of wind or wave model, wind from SCAT (on-board CFOSAT) or even propagated waves and storm sources from IWWOC L4 product in a feedback loop.
- handle complex situations such as coastal areas and heterogeneous seas. These complex situations are by nature of particular scientific interest (e.g. wind wave interactions, wave evolution in coastal region). It is also important to optimize the number and the representativeness of the wave observations with regard to IWWOC L3 and L4 products.
- build a flexible processing chain in order to facilitate algorithmic modifications and reprocessing. It is a key point for suggesting and testing alternative and original methodologies in the wave inversion scheme.

### 3.2 Main block diagram

Note: block diagram will be updated in future version.

Figure 4 illustrates the main block diagram of SWIM L2S processing.

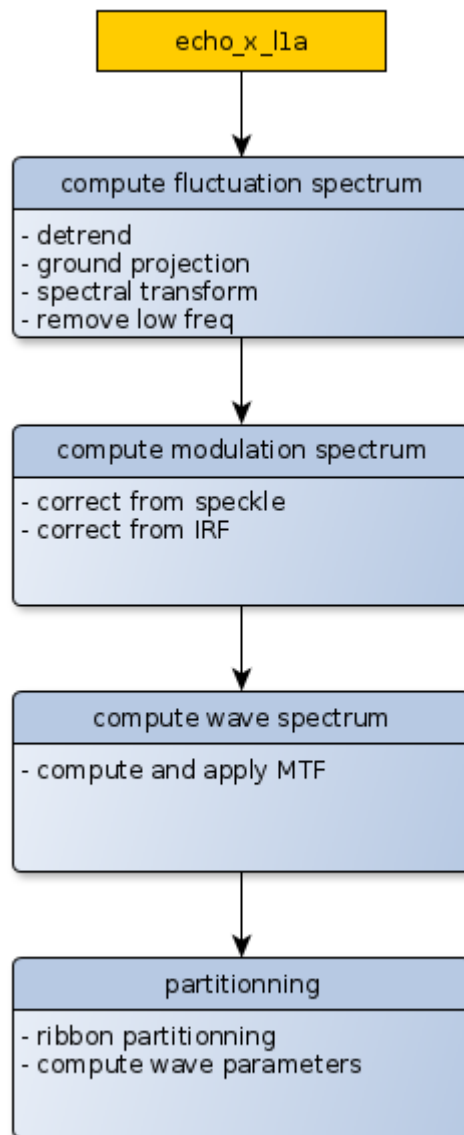


Figure 4: Main block diagram

Main input is the echo waveforms from CWWIC L1A product (echo\_x\_l1a).

Main output is the identified wave systems with their parameters.

## 4 Step 1: Ancillary data processing

This step is dedicated to the processing of various ancillary data for later use in L2s processing. The main processing task is to obtain or resample these ancillary data on SWIM geometry. This task is explained and the results are illustrated hereinafter for each kind of ancillary data.

The main inputs of this step are SWIM L1A longitudes and latitudes (i.e. among variables `lon_l1a_*` and `lat_l1a_*` according to the corresponding incidence angle). However, ancillary data are not necessarily resampled for every L1A points in range dimension. For now, the granularity of resampled ancillary data is the following:

- land: 1 point every 10 points in slant range dimension is used for computing the land mask.
- sea-ice / bathymetry / model wind: these data are resampled for each segment used in spectral estimation (see section 5.1.4).

Some ancillary data such as land or bathymetry data are considered fixed in time and will be used independently from L1A acquisitions. A new release of these data may eventually replace an old one from time to time. For other ancillary data (e.g. 3-hourly or daily data), closest data in time from L1A acquisitions has to be used. The way these data are collected is left outside the scope of this document.

### Open Point 1 More ancillary data

Currently, ancillary data mostly focus on masking capability. Other ancillary data will be progressively added. Among them:

- SCAT wind: to do along with SCAT activities.
- fireworks (IWWOC L4): to do along with L4 implementation and tests.
- wave model: wave model is mentioned in some open points mostly for use as a first guess (long swell weak signatures and propagation ambiguity removal). Really necessary for L2S processing compared to wind + fireworks inputs ? Otherwise, interesting from a user perspective to have it in L2S output ?

### 4.1 Land mask

Land mask is computed from OpenStreetMap land polygons data, available for download

at <http://openstreetmapdata.com/>. OpenStreetMap data are copyright OpenStreetMap contributors and available under the Open Database License (ODbL).

Given these land polygons, the definition of the land mask  $land(\phi, r_{sub})$  is as follows:

$$land(\phi, r_{sub}) = \begin{cases} 1 & \text{if } lon(\phi, r_{sub})/lat(\phi, r_{sub}) \text{ is inside a land polygon,} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

where  $r_{sub}$  indicates that slant range dimension is subsampled, currently 1 point every 10 points.

Regarding the high number of both SWIM geographical points (even subsampled) and high resolution OpenStreetMap land polygons, the above formula would require a huge number of point-in-polygon tests. A naive approach would make the processing very time-consuming. Without entering into too many implementation details, the processing is organized as follows:

- first, a rasterized version of the land polygons is used. This rasterized version has been computed over a regular lat/lon grid at medium resolution (0.025 degrees). It allows to tell quickly if SWIM geographical points are inside geographical bins containing either only sea or only land or a mix of sea and land. Only the points in the last case (i.e in coastal areas) need further tests with a point-in-polygon approach.
- then, a fast implementation of the point-in-polygon test is used and the number of these point-in-polygon tests to perform is reduced thanks to simple bounding box intersection tests. Smaller are the bounding boxes, less point-in-polygon tests are needed. As a consequence, SWIM geographical points are grouped by geographical areas (SWIM orbit cutting) and the split version of OpenStreetMap land polygons is used (more local and smaller polygons).

Figure 5 provides an example of SWIM land mask at 10° incidence angle for multiple cycles: blue color indicates SWIM positions at sea while chocolate color indicates SWIM positions on land. The original land polygons used for land mask computation are filled with gray color.

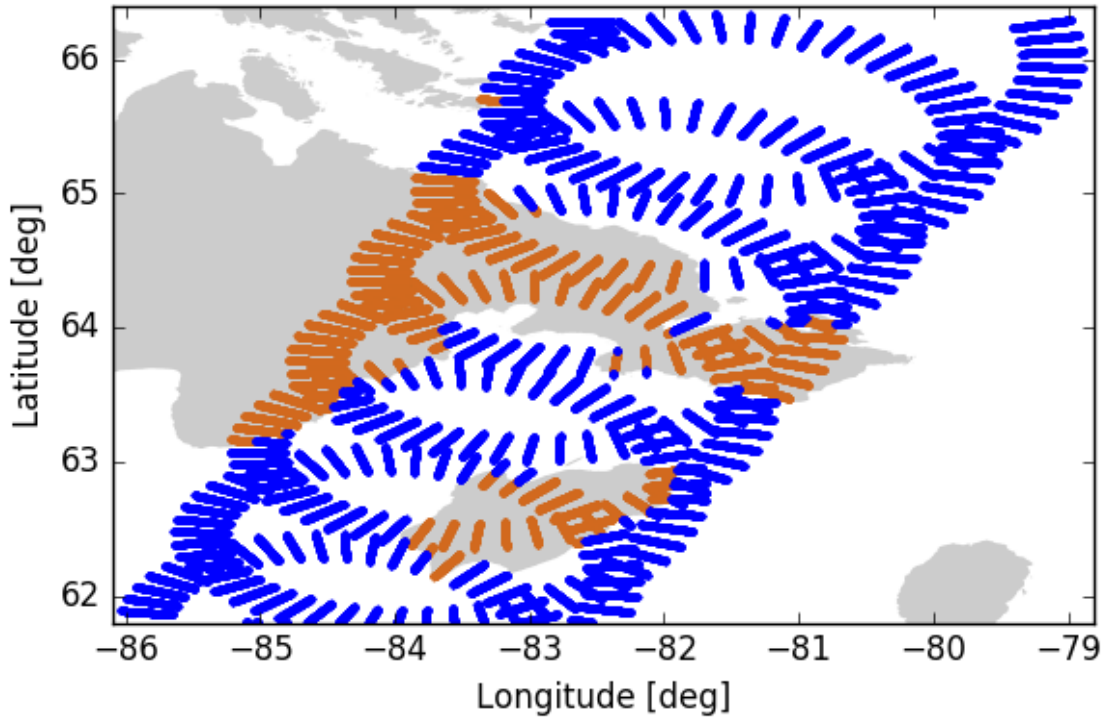


Figure 5: Exemple of SWIM land mask

## 4.2 Sea-ice concentration

Sea-ice concentration is obtained from CERSAT SSM/I 12.5km sea-ice concentration products ([RD6]), available at [http://products.cersat.fr/details/?id=CER\\_PSI\\_ARC\\_1D\\_012\\_PSI\\_SS](http://products.cersat.fr/details/?id=CER_PSI_ARC_1D_012_PSI_SS). These data are available in separate products for both polar regions: arctic and antarctic. Moreover, the sea-ice concentration is provided over a specific polar stereographic projection grid (according to the polar region).

The mapping of sea-ice concentration onto SWIM acquisition denoted by  $c_{seaice}(\phi, seg)$  is achieved independently for each polar region and at the scale of segments later used in spectral transform (see section 5.1.4). The first step is to convert SWIM  $lon(\phi, seg)$  and  $lat(\phi, seg)$  at high enough latitudes into projection coordinates in the sea-ice product projection. Then a bilinear interpolation of the sea-ice concentration is performed at SWIM points. Regarding the quality flags (see [RD6] for mode details), sea-ice concentration values are set to 0 before interpolation where the following flags have been raised: land pixel, bad pixel or open ocean. And values are left untouched elsewhere: no flags or possible land contamination.

In the same configuration than Figure 5, Figure 6 and Figure 7 illustrates respectively sea-ice concentration  $c_{seaice}(\phi, seg)$  and the one from the original product. Values equal to 0 are not displayed.

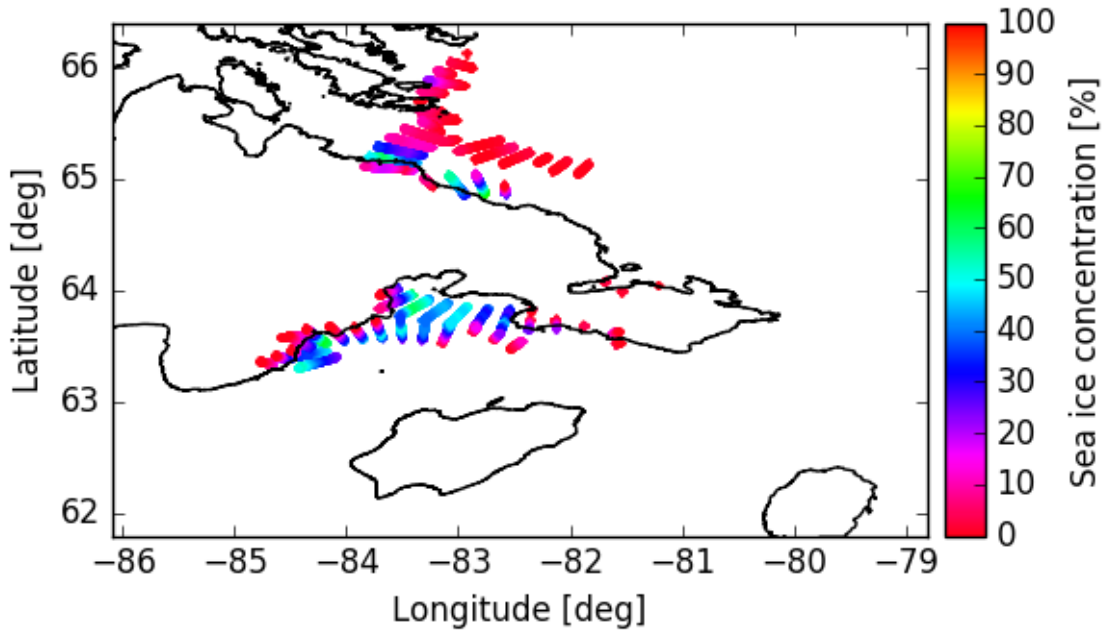


Figure 6: Example of sea-ice concentration onto SWIM

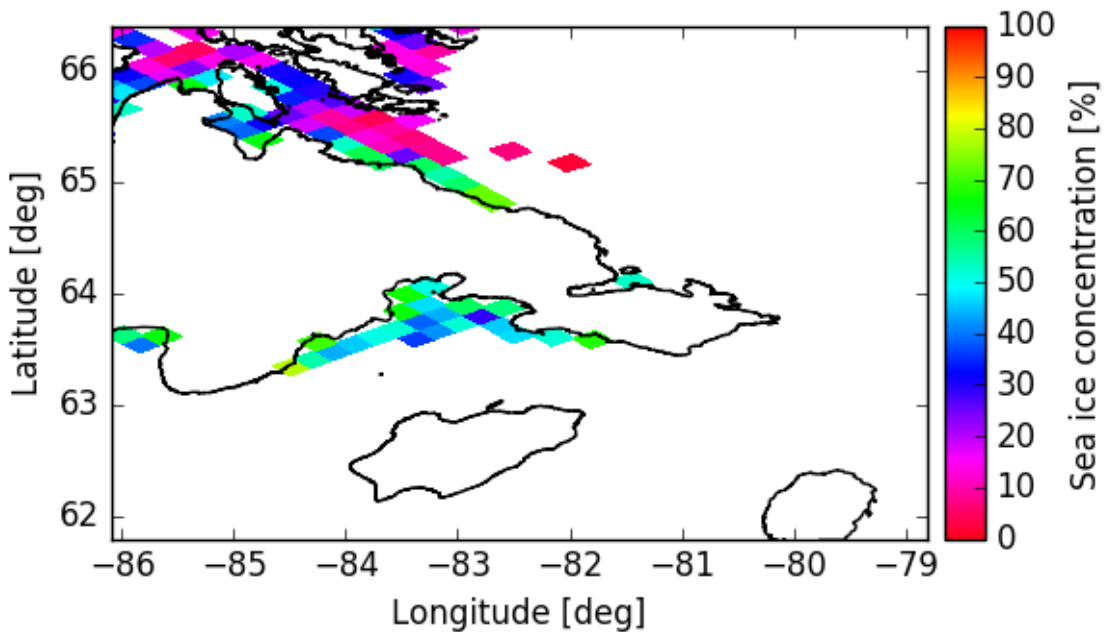


Figure 7: Example of original sea-ice concentration

### 4.3 Bathymetry

Bathymetry is obtained from the General Bathymetric Chart of the Oceans (GEBCO) data, available at <http://www.gebco.net/>. The most recent GEBCO grid is used, namely the GEBCO\_2014 grid providing a 1/120 degrees global lat/lon grid of elevations.

Bathymetry elevations are mapped onto SWIM acquisition by bilinear interpolation at the scale of segments later used in spectral transform (see section 5.1.4). The result of this mapping is denoted by  $e_{bathy}(\phi, seg)$ . Due to the very large-size of the GEBCO grid, the only special feature of the processing is to cut SWIM orbit into smaller parts in order to read only the relevant parts of the GEBCO grid.

In the same configuration than Figure 5, Figure 8 and Figure 9 illustrates respectively bathymetry elevation  $e_{bathy}(\phi, seg)$  and the one from the original product.

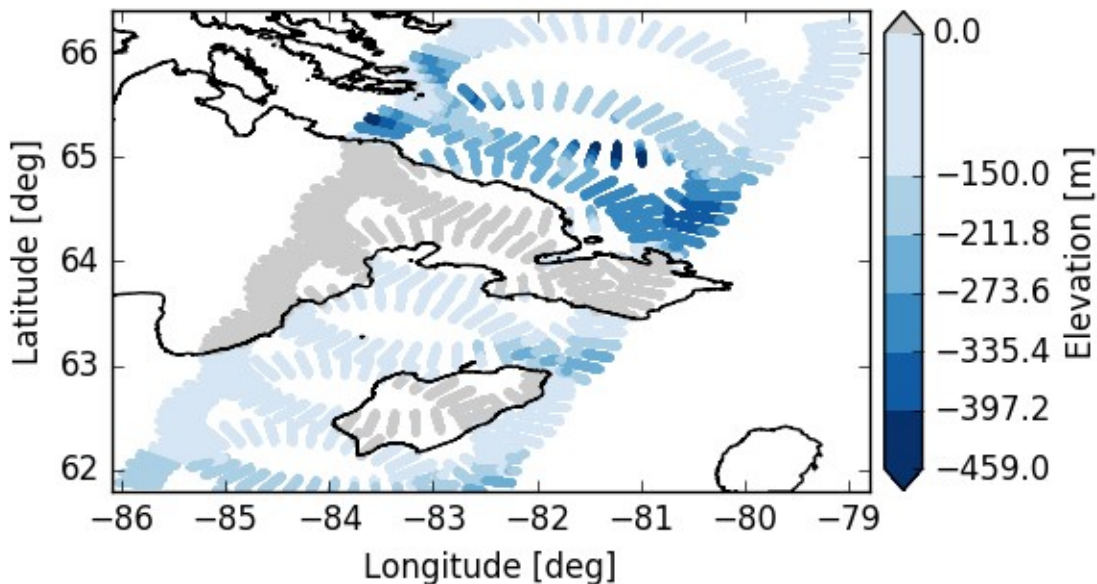
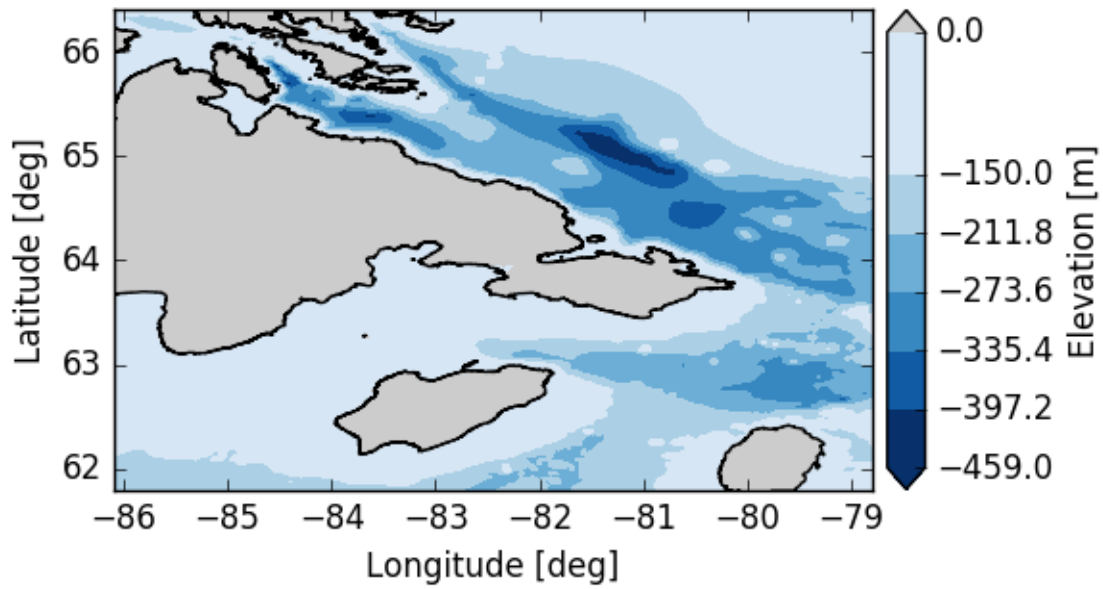


Figure 8: Example of bathymetry onto SWIM





*Figure 9: Example of original bathymetry*

#### **4.4 Model wind**

Currently, model wind is directly obtained from CWWIC AUX METEO files:  $u_{10}(\phi)$  and  $v_{10}(\phi)$ . In the future, model wind will be obtained from interpolation of ECMWF best analysis (at segment scale like sea-ice and bathymetry).

## 5 Step 2: Fluctuation spectrum computation

### 5.1 Theoretical description

This step is dedicated to the computation of the fluctuation spectrum  $S_f$  defined as the power spectral density of the NRCS fluctuations  $\Delta\sigma_0$ . These fluctuations, mainly related to the waves, are defined with respect to an NRCS trend  $\sigma_{0t}$  mainly related to the wind:

$$\Delta\sigma_0 = \frac{\sigma_0 - \sigma_{0t}}{\sigma_{0t}} \quad (2)$$

Then, with  $F$  denoting the Fourier transform and  $L_x$  the ground range coverage,  $S_f$  is defined as:

$$S_f = \lim_{L_x \rightarrow \infty} \frac{1}{2\pi L_x} |F(\Delta\sigma_0)|^2 \quad (3)$$

#### 5.1.1 Bright target removal

##### Open Point 2 Bright target removal

Bright targets (e.g. ships or rain cell impacts) may cause many issues: extreme values to resample, a bias in the NRCS trend and undesired spectral energy at most of the wavelengths. Therefore, these bright targets have to be filtered out in spatial domain as soon as possible. We need a detection method (CFAR like ?) and a strategy to replace the detected high  $\sigma_0$  values (mean or low-pass around the bright pixels ?).

#### 5.1.2 Regular ground range resampling

At this stage, the signal is still in radar geometry, i.e. sampled with a constant slant range  $\Delta r$ . Given  $\Delta r$ , the ground range spacing  $\Delta x$  varies with incidence angle  $\theta$  along the range dimension according to:

$$\Delta x(r) = \frac{\Delta r}{\sin(\theta(r))} \quad (4)$$

Given the low incidence angles of SWIM, the ground range spacing can not be approximated to the one at middle swath (see Table 9 for an overview of spacings, coverage and resolutions for each beam). Therefore, before applying any spectral transform, the signal needs to be resampled from a constant slant range spacing  $\Delta r$  to

a constant ground range spacing  $\Delta x$ . Moreover, this resampling is achieved as soon as possible during the processing because it is in general far easier to deal with regularly spaced signals. Only the bright target removal is done before the resampling in order not to have extreme values to resample.

	6° beam	8° beam	10° beam
Number of points $N_r$	2772	2640	3216
Slant range spacing $\Delta r$	0.749m	1.124m	1.124m
Ground range spacing at $-1^\circ$ and $+1^\circ$	8.59m $\rightarrow$ 6.14m	9.22m $\rightarrow$ 7.18m	7.18m $\rightarrow$ 5.89m
Ground range coverage	20194m	21515m	20896m
Slant range resolution	0.937m	1.405m	1.405m
Ground range resolution at $-1^\circ$ and $+1^\circ$	10.75m $\rightarrow$ 7.68m	11.52m $\rightarrow$ 8.98m	8.98m $\rightarrow$ 7.36m

*Table 9: Nominal values of spacings, coverage and resolutions for each beam in normal mode*

The requested ground range spacing  $\Delta x$  is a configurable processing parameter which will define the number of points  $N_x$  in the range dimension after resampling. All the variables defined along the range dimension need to be resampled, so far it includes  $\sigma_0(\phi, r)$  and  $land(\phi, r_{sub})$ . These variables are resampled by interpolation, the first step of resampling is then to identify locations of interpolation inside the signal in slant range geometry. This step is performed by function *regular\_ground\_range\_locations* detailed in section 5.3.1. This function takes as inputs the ground range values  $x(\phi, r)$  in slant range geometry (derived from `ground_range_*` variables in L1A files) and parameter  $\Delta x$ . Outputs are  $N_x$  and resampling locations  $l_r(\phi, x)$  defined as floating indices of the signal in slant range geometry.

$\sigma_0(\phi, r)$  is resampled with a sinc interpolation scheme. With omission of the cycle dimension  $\phi$ ,  $\sigma_0(r)$  is written at any slant range  $r$  as:

$$\sigma_0(r) = \sum_{i_r} \sigma_0(i_r \Delta r) \text{sinc}\left(\frac{r - i_r \Delta r}{\Delta r}\right) \quad (5)$$

corresponding to Whittaker-Shannon interpolation formula.

The sinc interpolation scheme is performed by function *sinc\_resample* (detailed in section 5.3.2) according to  $\sigma_0(\phi, r)$  and resampling locations  $l_r(\phi, x)$ . Equation (5) is used in practice with a windowed sinc kernel of finite length. This kernel length  $L_{rsp}$  is

a processing parameter. Moreover, the sinc kernel depends on the relative position of the resampling points from the input points (i.e.  $(r-i_r\Delta r)/\Delta r$  in equation 5). In order to deal with a limited number of kernels, input intervals are quantized over  $Q_{rsp}$  values,  $Q_{rsp}$  being also a processing parameter. And finally, in case of downsampling this function takes care of using a larger sinc kernel which is equivalent to a low-pass filter and avoids aliasing effects.

Figure 10 and Figure 11 show the results of this resampling for a portion of  $\sigma_0$  at respectively near and far range. Blue dots are original  $\sigma_0$  sampled at about 7.2m in near range and about 5.9m in far range (SWIM 10° incidence angle beam). Red dots are resampled  $\sigma_0$  at  $\Delta x=5m$ . Blue line, used for display purpose, was computed with a resampling at 1m. Here  $L_{rsp}=16$  and  $Q_{rsp}=64$ .

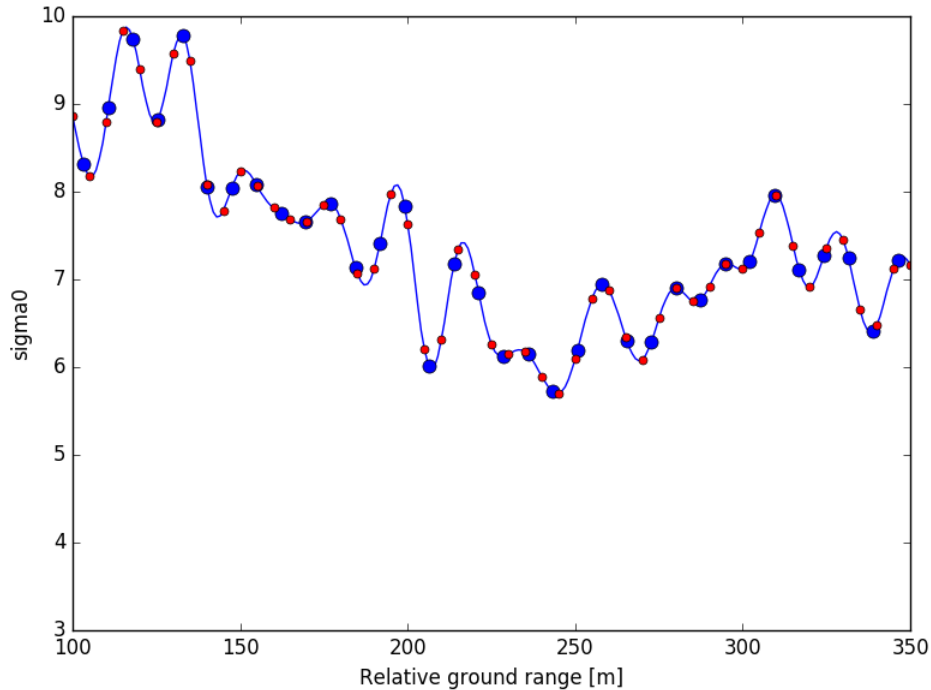


Figure 10: Example of  $\sigma_0$  resampling at near range

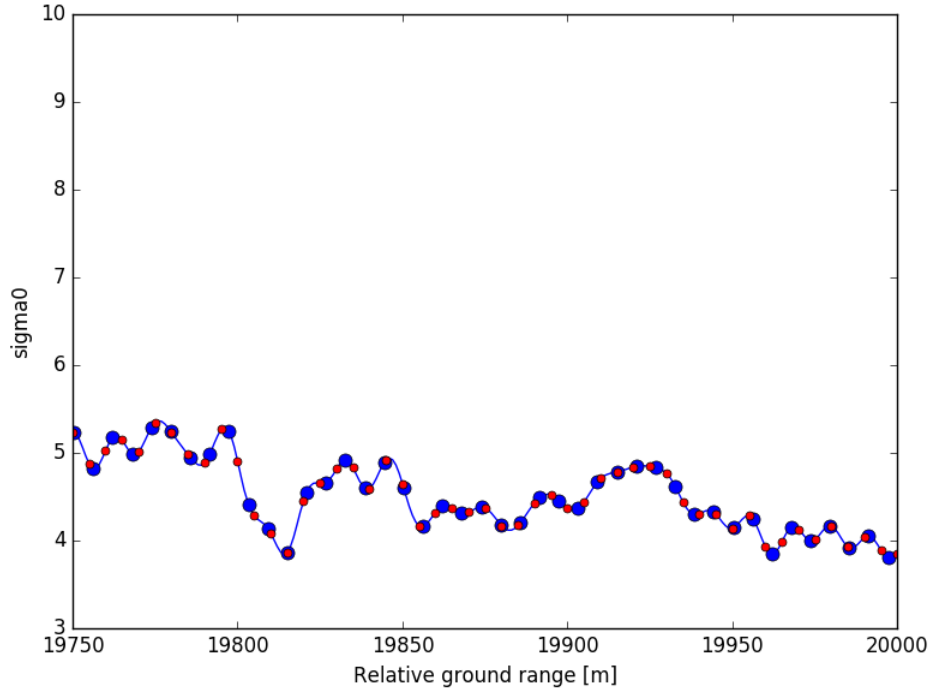


Figure 11: Example of  $\sigma_{0t}$  resampling at far range

The resampling of  $land(\phi, r_{sub})$  into  $land(\phi, x)$  is performed from the same resampling locations  $l_r(\phi, x)$  but with a simple nearest interpolation scheme.

### 5.1.3 Detrending

Detrending refers to the estimation of the trend  $\sigma_{0t}$  and to the computation of the fluctuations  $\Delta\sigma_0$ . In order to not make any assumptions on the scene homogeneity, the trend  $\sigma_{0t}$  is estimated with a gaussian low-pass filtering:

$$\sigma_{0t}(\phi, i_x \Delta x) = \frac{1}{\sqrt{2\pi} w_x} \sum_n \sigma_0(\phi, (i_x - n) \Delta x) e^{-\frac{(n \Delta x)^2}{2w_x^2}} \quad (6)$$

where  $w_x$  is a processing parameter controlling the gaussian width. This filtering is achieved by the function *gaussian\_lowpass* detailed in section 5.3.3. Figure 12 illustrates such a trend with  $w_x = 500m$ .

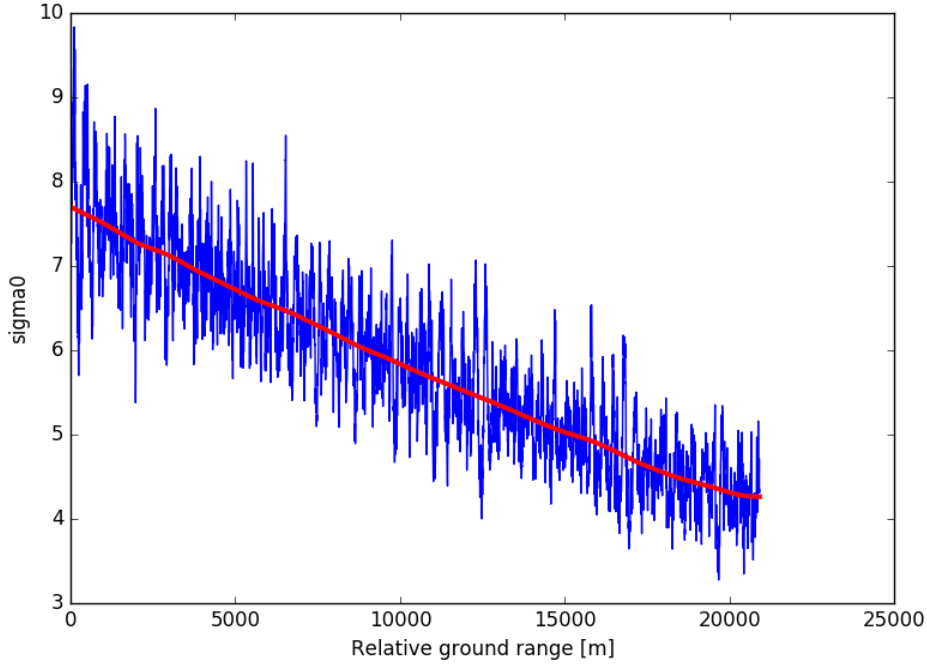


Figure 12: Example of  $\sigma_0$  trend

NRCS fluctuations  $\Delta\sigma_0(\phi, \chi)$  are then:

$$\Delta\sigma_0(\phi, \chi) = \frac{\sigma_0(\phi, \chi)}{\sigma_{0t}(\phi, \chi)} - 1 \quad (7)$$

#### 5.1.4 Spectral transform

Spectral transform is applied to the NRCS fluctuations in ground range geometry according to equation (3). Instead of using a single discrete Fourier transform for each cycle, the Welch's method is used:

- for a given cycle, fluctuations are split up into overlapping segments: the overlapping segments are windowed and periodograms are estimated by computing the discrete Fourier transform,
- the individual periodograms are then averaged.

Welch's method allows to reduce noise in the estimated power spectra in exchange of reducing the frequency resolution because of the use of segments smaller than the full signal. And since fluctuations are cut into smaller segments at each cycle, it is also an opportunity to compute a power spectral density even for partially masked cycles.

The number  $N_{seg}$  and starting/ending positions  $seg_0 / seg_1$  of the segments are identical for each cycle and are computed according to the number of points in range

dimension  $N_x$  and 2 processing parameters: the periodogram length  $L_{per}$  and the periodogram overlap  $O_{per}$ . It is achieved by the function *segment\_positions* detailed in section 5.3.4. Starting/ending positions of segments also allow to define useful variables at the middle of each segment, such as incidence angle  $\theta(\phi, seg)$ , latitude  $lat(\phi, seg)$  and longitude  $lon(\phi, seg)$ .

For each segment, the periodogram (power spectral density) of the NRCS fluctuations is computed from the classical FFT algorithm. NRCS fluctuations are multiplied by a Hann window before the FFT. And normalizations are applied in order to take into account the windowing, the FFT implementation and the ground coverage of the segments. More details can be found in the description of the function *spectral\_transform* in section 5.3.5. Outputs of this function are  $S_f(\phi, k, seg)$  the fluctuation spectra per cycles and per segments,  $N_k$  the number of wavenumbers and  $\Delta k$  the wavenumber spacing.

Figure 13 shows in red an example of a fluctuation spectrum computed with the method described here and after averaging over the segment dimension. Spectrum in blue has been computed for comparison with a single FFT over the range dimension. The following settings have been used for this example:  $\Delta x=5\text{m}$  (resulting to  $N_x=4182$ ),  $L_{per}=512$  and  $O_{per}=0.5$ . It results to a number of averaged periodograms  $N_{seg}$  equal to 15.

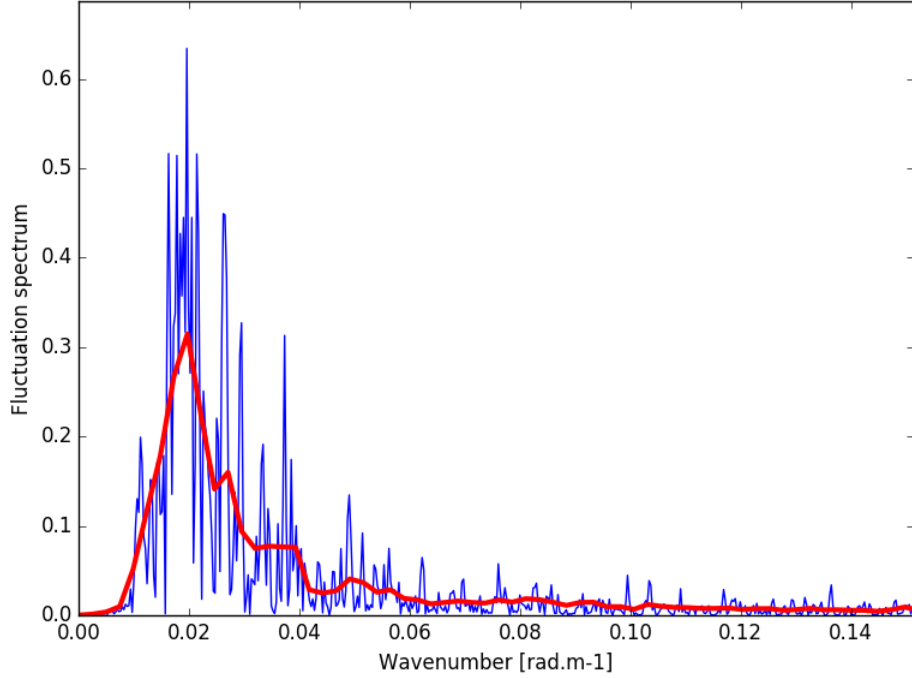


Figure 13: Example of fluctuation spectrum

This step does not average fluctuation spectra over segment dimension, it is done later in the processing. However, a flag is needed in order to know which segments (ie periodograms) can be used for averaging. This flag is first defined from the following (flag initialized with zeros):

- $flag_{seg}(\phi, seg) += 2$  if cycle is not available according to L1A flag\_availability.
- $flag_{seg}[\phi, i_{seg}] += 4$  if  $\max(land[\phi, seg_0[i_{seg}]:seg_1[i_{seg}]]) = 1$
- $flag_{seg}[\phi, i_{seg}] += 8$  if  $c_{seaice}[\phi, i_{seg}] > T_{seaice}$
- $flag_{seg}[\phi, i_{seg}] += 16$  if  $e_{bathy}[\phi, i_{seg}] > T_{bathy}$

$T_{seaice}$  and  $T_{bathy}$  are two processing parameters. Note that this flag may evolve in the future by taken into account more L1A flags or  $\sigma_0$  properties.

Candidates for averaging are where  $flag_{seg}$  is still 0. By defining  $A_{seg}(\phi)$  the total number of candidates for each cycle, indication of future averaging is then:

- $flag_{seg}[\phi, i_{seg}] = 1$  if  $flag_{seg}[\phi, i_{seg}] = 0$  and  $A_{seg}(\phi) \geq T_{per}$

$T_{per}$  is a processing parameter.



### Open Point 3 Long swell weak signatures (spectral domain)

Some weak signals like the modulations of long waves could require a special processing in order to detect them. This special processing could be triggered by a first guess from a wave model or from propagated wave observations (fireworks with potentially SAR, SWIM, buoys and seismograph observations).

With respect to the spectral domain, we would like to ensure that the wavenumbers are suitable to highlight the energy at long wavelengths. It means using longer periodograms (but increasing noise in spectrum). This open point may be summarized as: can we use a constant periodogram length or do we need to adapt it from case to case (based on ancillary information) ?

## 5.1.5 Post-processing

### Open Point 4 NRCS heterogeneity (remove low freq)

Low frequency signatures (e.g. atmospheric signatures) may produce spectral energy at wavelengths close to the long swell wavelengths. The idea for filtering out these signatures is to consider that, starting at the first wavenumber  $> 0$ , as long as the energy decreases it can be removed because it is not wave energy. In SAR, it is classically applied in 2D (cartesian coordinates) with a 4-connectivity for energy comparison. To be tested either in 1D (for each azimuth) or in 2D (ribbon).

## 5.2 Block diagram and processing parameters

Figure 14 illustrates the block diagram of this processing step and Table 10 summarizes all the processing parameters with their default values.

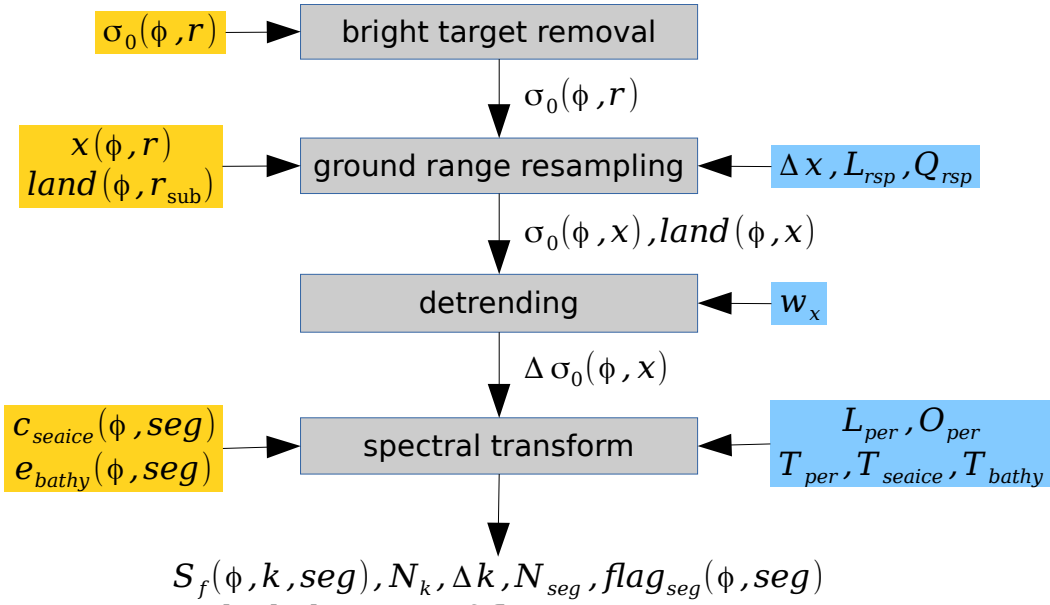


Figure 14: Block diagram of fluctuation spectrum computation

Notation	Definition	Default
$\Delta x$	Ground range spacing (resampling)	10m
$L_{rsp}$	Sinc kernel length (resampling)	32
$Q_{rsp}$	Interval quantization (resampling)	64
$w_x$	Gaussian standard deviation for low-pass (trend)	750m
$L_{per}$	Periodogram length (spectral transform)	256
$O_{per}$	Periodogram overlap (spectral transform)	0.5
$T_{per}$	Threshold on number of periodograms	5
$T_{seaice}$	Threshold on sea-ice concentration	10%
$T_{bathy}$	Threshold on bathymetry elevation	-150m

Table 10: Processing parameters of fluctuation spectrum computation

## 5.3 Mathematical description

### 5.3.1 regular\_ground\_range\_locations

**Objective:** Find the locations of regular ground range positions in regular slant range signal.

**Inputs**

$x(\phi, i_r, \Delta r)$  : ground range in slant range geometry (i.e. irregular ground range).

<b>Parameters</b>
$\Delta x$ : ground range spacing
<b>Outputs</b>
$N_x$ : number of regular ground range points
$l_r(\phi, i_x \Delta x)$ : regular ground range locations inside regular slant range signal (floating indices)

By convention,  $x$  is 0 at the beginning of each footprint:  $x[\phi, i_r=0]=0$ .

As a consequence, last value of  $x$  in range dimension is the ground range coverage for each footprint:  $x[\phi, i_r=N_r-1]$ .

Then, it follows:

- $L_x = \min(x[\phi, i_r=N_r-1] \forall \phi)$  the resampling ground range coverage,
- $N_x = \text{floor}(\frac{L_x}{\Delta x}) + 1$  the number of resampling points according to  $\Delta x$ .

Finally, for  $i_x \in 0..N_x-1$ :

$$l_r[\phi, i_x] = i_r + \frac{i_x \Delta x - x_r[\phi, i_r]}{x_r[\phi, i_r+1] - x_r[\phi, i_r]} \quad \text{with } x_r[\phi, i_r] \leq i_x \Delta x < x_r[\phi, i_r+1]$$

$l_r(\phi, i_x \Delta x)$  is in fact a linear interpolation of function  $i_r = f(x(\phi, i_r \Delta r))$  at points  $i_x \Delta x$ .

### 5.3.2 sinc\_resample

<b>Objective:</b> Resample signal with a sinc interpolation scheme
<b>Inputs</b>
$\sigma_0(\phi, i_r \Delta r)$ : NRCS in regular slant range
$x(\phi, i_r \Delta r)$ : ground range in regular slant range
$l_r(\phi, i_x \Delta x)$ : regular ground range locations inside regular slant range signal
<b>Parameters</b>
$\Delta x$ : ground range spacing

$L_{rsp}$ : resampling sinc kernel length
$Q_{rsp}$ : resampling interval quantization
<b>Outputs</b>
$\sigma_0(\phi, i_x \Delta x)$ : resampled NRCS in regular ground range

Locations  $l_r(\phi, i_x \Delta x)$  can be separated into integer and fractional parts:

- $l_{rint}(\phi, i_x \Delta x) = \text{floor}(l_r(\phi, i_x \Delta x))$
- $l_{frac}(\phi, i_x \Delta x) = l_r(\phi, i_x \Delta x) - l_{rint}(\phi, i_x \Delta x)$

Then, a resampling factor is defined as:

$$F_{rsp}[\phi, i_x] = \max\left(\frac{\Delta x}{x[\phi, l_{rint}[\phi, i_x] + 1] - x[\phi, l_{rint}[\phi, i_x]]}, 1\right).$$

This factor greater than 1 for downsampling will make sinc interpolation behave like a low pass filter and then avoids aliasing.

For given  $\phi$  and  $i_x$ , the sinc kernel is defined from:

- $x_K[i_k] = i_k - \frac{L_{rsp}}{2} + 1$  for  $i_k \in 0..L_{rsp} - 1$
- a window:  $w_k[i_k] = 0.54 + 0.46 \cos\left(\frac{2\pi(x_K[i_k] - l_{frac}[\phi, i_x])}{L_{rsp}}\right)$  for  $i_k \in 0..L_{rsp} - 1$
- and finally the kernel:  $K[i_k] = w_k[i_k] \text{sinc}\left(\frac{x_K[i_k] - l_{frac}[\phi, i_x]}{F_{rsp}[\phi, i_x]}\right)$  for  $i_k \in 0..L_{rsp} - 1$

If  $F_{rsp}$  is constant, sinc kernel  $K$  only depends on  $l_{frac}$ . Parameter  $Q_{rsp}$  tells to round  $l_{frac}$  to  $1/Q_{rsp}$  which decreases the number of kernels to generate.

Finally, sinc interpolation is written:

$$\sigma_0[\phi, i_x] = \frac{1}{\sum_{i_k=0}^{L_{rsp}} K[i_k]} \sum_{i_k=0}^{L_{rsp}} K[i_k] \sigma_0[\phi, l_{rint}[\phi, i_x] + x_K[i_k]]$$

(Note:  $\sigma_0$  on the right hand side is input  $\sigma_0$  defined on a regular slant range)

### 5.3.3 gaussian\_lowpass

<b>Objective:</b> Computes the NRCS trend with a gaussian low-pass filtering
<b>Inputs</b>
$\sigma_0(\phi, x)$ : NRCS
$\Delta x$ : ground range spacing
<b>Parameters</b>
$w_x$ : gaussian standard deviation in meters
<b>Outputs</b>
$\sigma_{0t}(\phi, x)$ : NRCS trend

The gaussian standard deviation in pixels is  $w = w_x / \Delta x$ .

The gaussian kernel half size is defined as  $N_g = \text{floor}(4 * w + 0.5)$ .

Then the gaussian kernel is  $G[i_g] = e^{\frac{-i_g^2}{2w^2}}$  for  $i_g \in [-N_g, N_g]$ .

Finally the low-pass filtering at a given position  $i_x$  in range is written as:

$$\sigma_{0t}[\phi, i_x] = \frac{1}{\sum_{i_g=-N_g}^{i_g=N_g} G[i_g]} \sum_{i_g=-N_g}^{i_g=N_g} G[i_g] \sigma_0[\phi, i_x + i_g]$$

### 5.3.4 segment\_positions

<b>Objective:</b> Compute segment positions for periodograms
<b>Inputs</b>
$N_x$ : number of pixels in ground range dimension
<b>Parameters</b>
$L_{per}$ : periodogram length
$O_{per}$ : periodogram overlap
<b>Outputs</b>
$N_{seg}$ : number of segments

$seg_0$ : starting positions of segments
$seg_1$ : ending positions of segments

The number of segments (or periodograms) is computed as:

$$N_{seg} = \text{round}((N_x / L_{per} - O_{per}) / (1 - O_{per}))$$

Then starting / ending positions (ie indices in range dimension) of segments are given by:

$$seg_0[i_{seg}] = \text{round}(i_{seg} \frac{N_x - L_{per}}{N_{seg} - 1}) \text{ for } i_{seg} \in 0..N_{seg} - 1$$

$$seg_1[i_{seg}] = seg_0[i_{seg}] + L_{per} - 1 \text{ for } i_{seg} \in 0..N_{seg} - 1$$

### 5.3.5 spectral\_transform

<b>Objective:</b> Compute the fluctuation spectrum from the NRCS fluctuations using the Welch's method
<b>Inputs</b>
$\Delta\sigma_0(\phi, x)$ : NRCS fluctuations
$\Delta x$ : ground range spacing
$N_{seg}$ : number of segments
$seg_0$ : starting positions of segments
$seg_1$ : ending positions of segments
<b>Parameters</b>
$L_{per}$ : periodogram length
<b>Outputs</b>
$S_f(\phi, k, seg)$ : fluctuation spectrum
$N_k$ : number of wavenumbers
$\Delta k$ : wavenumber spacing

*Note: all equations are written here considering that  $L_{per}$  is even.*

In order to simplify notations,  $\phi$  is omitted since spectral transform is only applied to range dimension. And NRCS fluctuations for a particular segment  $i_{seg}$  are denoted  $\Delta\sigma_{0iseq}$  (ie.  $\Delta\sigma_{0iseq}=\Delta\sigma_0[seg_0[i_{seg}]:seg_1[i_{seg}]]$ ).

The window function  $w(\chi)$  (Hann window) is defined as:

$$w[i]=\frac{1}{2}-\frac{1}{2}\cos\left(\frac{2\pi i}{L_{per}}\right) \text{ for } i\in 0..L_{per}-1$$

Then for each segment, the discrete Fourier transform is defined as:

$$DFT(w\Delta\sigma_{0iseq})[i_k]=\sum_{n=0}^{L_{per}-1} w[n]\Delta\sigma_{0iseq}[n]e^{-j2\pi i_k n/L_{per}} \text{ for } i_k\in 0..N_k-1$$

$$\text{with } N_k=\frac{L_{per}}{2}+1, \Delta k=\frac{2\pi}{L_{per}\Delta x} \text{ and } k=i_k\Delta k \text{ for } i_k\in 0..N_k-1$$

The DFT is defined here only for positive wavenumbers because the fluctuations are real. Moreover, this definition is merely a convention, it depends on FFT implementation.

In order to preserve the energy from the spatial domain to the spectral domain, we define the following normalization coefficients :

- $C_w=1/\overline{w^2}$
- $C_{real}[i_k]=\begin{cases} 1 & \text{if } i_k=0 \\ 2 & \text{if } i_k>0 \end{cases}$
- $C_{DFT}=\frac{\Delta x}{L_{per}}$  (with the definition of DFT given here)

Finally, equation (3) in discrete form and per segment is written:

$$S_f[i_k, i_{seg}]=\frac{C_{DFT}C_{real}[i_k]C_w}{2\pi}|DFT(w\Delta\sigma_{0iseq})|^2 \text{ for } i_k\in[0, N_k-1] \text{ and } i_{seg}\in[0, N_{seg}-1]$$

## 6 Step 3: Modulation spectrum computation

### 6.1 Theoretical description

This step is dedicated to the computation of the modulation spectrum  $S_m$  corresponding to the fluctuation spectrum  $S_f$  corrected from impulse response (IR) and from speckle noise.

#### 6.1.1 Method 0 (simulation based)

Based on simulation analyses, the following formula are used:

- $S_f(\phi, k, seg) = S_{ir}(\phi, k, seg)(S_m(\phi, k, seg) + S_{sp}(\phi, k, seg))$
- $S_{ir}(\phi, k, seg) = \text{tri}\left(\frac{k \delta r}{2 \pi L_{dis} \sin(\theta(\phi, seg))}\right)^2$ , IR spectrum
- $S_{sp}(\phi, k, seg) = \frac{2 \Delta r}{2 \pi L_{dis} N_{imp} \sin(\theta(\phi, seg))}$ , speckle spectrum

With:

- $\text{tri}(t) = \max(1 - |t|, 0)$  the triangle function
- $\delta r / \Delta r$  the slant range resolution / spacing
- $L_{dis}$  the number of range gates averaged on-board
- $N_{imp}$  the number of pulses averaged on-board
- $\theta(\phi, seg)$  the incidence angle for each segment used in spectral transform

Notes:

- thermal noise is ignored
- adding  $S_{sp}$  to  $S_m$  is an approximation

#### 6.1.2 Method 1 (not yet implemented nor tested)

##### Open Point 5 Speckle LUT

In order to use an empirical characterization of the speckle given the geometry and the sea state, tools must be prepared to create such a LUT during the cal/val phase. Preliminary idea for learning the speckle LUT is to use fluctuation spectra without any waves signatures or/and to use cross-spectra in speckle mode.



## Open Point 6 IR from instrument

Investigation and implementation of the estimation of IR from the calibration channels of the instrument.

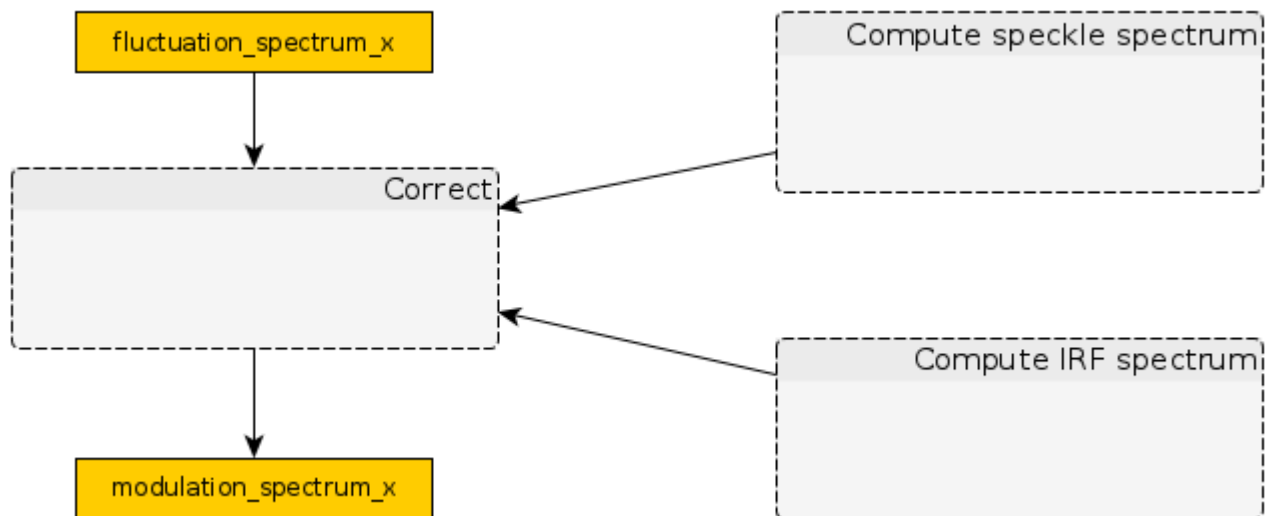
## Open Point 7 Directional spread

As described in [RD5], the SWIM instrument imaging process has a directional impulse response that will induce a larger directional spread (of about 5 to 10 degrees) than the wave spectrum. A corrective method will be proposed either to deconvolve this directional impulse response from the SWIM 2D spectra or to correct a posteriori the directional spread parameter of each wave partitions. The deconvolution could take the form of a simple division by the Fourier transform of the impulse response in the Fourier domain.

## 6.2 Block diagram

*Note: block diagram will be updated in future version.*

Figure 15 illustrates the block diagram of this processing step.



*Figure 15: Block diagram of modulation spectrum computation*

## 6.3 Mathematical description

*Note: will be updated in a future version with method 1 progress.*

## 7 Step 4: Wave spectrum computation

### 7.1 Theoretical description

This step is dedicated to the computation of the modulation transfer function (MTF) characterizing the wave imaging mechanisms. Given the MTF denoted as  $T$ , the wave spectrum  $S_s$  (wave slope) is obtained from the modulation spectrum  $S_m$  with:

$$S_m(\phi, k) = T(\phi, k) \times S_s(\phi, k) \quad (8)$$

The different wave imaging mechanisms are the:

- wave slope tilt: the tilt of short wave mss by long wave is modulating the radar backscatter. The modulation transfer function between the slope of long waves and the modulation of observed backscatter is referred as tilt MTF.
- range bunching: as described in [RD5], for small incidence angles a surface tilted towards the radar will appear brighter (independent of its  $\sigma_0$  properties) and foreshortened, because the reflected energy has to be mapped into a smaller slant range interval on the image.

#### 7.1.1 Estimate MTF - method 0 (“theoretical” tilt MTF)

In this method, the range bunching is not taken into account and a theoretical formulation of the tilt MTF  $T_t$  is used:

$$T(\phi, k) = T_t(\phi) = \frac{\sqrt{2\pi}}{L_y(\phi)} \alpha^2$$

with  $L_y$  the footprint azimuthal ground length and  $\alpha$  the sensitivity coefficient.

With the assumption of a Gaussian distribution of slopes:

$$\alpha(\theta) = \cot(\theta) - 4 \tan(\theta) + \frac{2 \tan(\theta)}{mss \times \cos(\theta)^2}$$

And, given an auxiliary wind  $U$ ,  $mss$  is estimated from the empirical relationship of Jackson et al. (1985):

$$mss = a_{mss} U + b_{mss} \quad \text{where } a_{mss} = 0.0028 \quad \text{and } b_{mss} = 0.009$$

Equation (8) is applied per segment according to incidence  $\theta(\phi, seg)$  and wind  $u_{10}(\phi) / v_{10}(\phi)$ . Then the resulting wave slope spectra are averaged over all the segments for each  $\phi$ .

## 7.1.2 Estimate MTF - method 1 (not implemented)

### Open Point 8 MTF

The empirical characterization of MTF will be studied from known wave spectra (buoys/model) and observations (TRMM/GPM/Kuros/SWIM\_calval).

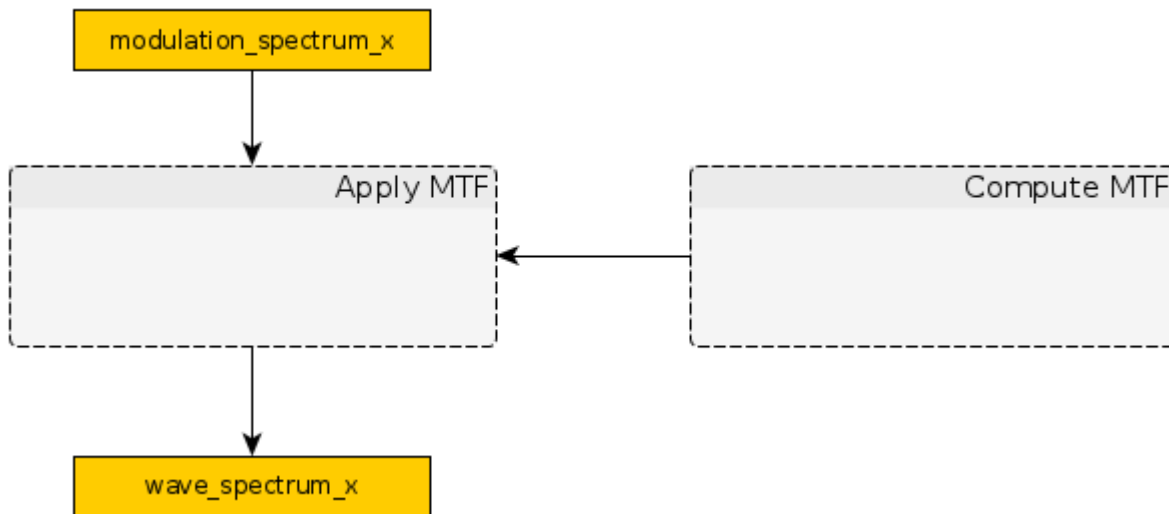
This study must allow to define the best strategy for the MTF computation, from a simple one using a LUT and ancillary data to a more complex one if the MTF is not bijective regarding the ancillary information.

Additional modulation of the radar backscatter will be induced by the range bunching phenomena that amplifies the tilt modulation by a factor dependent on the incidence angle but also the wave steepness. The incidence angle dependency between consecutive beams at 8 and 10° incidence (covering in total the range of incidence 7 to 11°) could be used to estimate the wave steepness and therefore to estimate the range bunching factor to compensate for. Additional information on the wave amplitude could be derived from the estimated steepness if the wavelength can be estimated by a Fourier or similar analysis of the SWIM waveform.

## 7.2 Block diagram

*Note: block diagram will be updated in future version.*

Figure 16 illustrates the block diagram of this processing step.



*Figure 16: Block diagram of wave spectrum computation*

## 7.3 Mathematical description

### 7.3.1 Estimate MTF - method 0 (“theoretical” tilt MTF)

**Objective:** Estimate the modulation transfer function

<b>Inputs</b>
$\theta_c$ : incidence angle at middle swath
$L_y(\phi)$ : footprint azimuthal ground length for each azimuth
$U(\phi)$ : wind for each azimuth (aux meteo)
<b>Outputs</b>
$T(\phi)$ : modulation transfer function

$mss(\phi) = a_{mss} U(\phi) + b_{mss}$  with  $a_{mss} = 0.0028$  and  $b_{mss} = 0.009$  (Jackson et al. 1985)

$$\alpha(\phi) = \cot(\theta_c) - 4 \tan(\theta_c) + \frac{2 \tan(\theta_c)}{mss(\phi) \cos(\theta_c)^2}$$

$$T(\phi) = \frac{\sqrt{2\pi}}{L_y(\phi)} \alpha(\phi)^2$$

## 8 Step 5: Ribbon partitioning

### 8.1 Theoretical description

Ocean wave spectrum partitioning consists of identifying individual wind sea or swell systems produced by different meteorological events and computing integral parameters for each one of these wave systems. It is an important step for interpretation since these integral parameters are more meaningful than the ones integrated over the whole spectrum. In the IWWOC context, individual wave systems are also the starting points for wave propagation and thus are critical inputs for the generation of L4 products.

Partitioning is usually applied to either 1D or 2D spectra. In the case of SWIM and for a given incidence angle, each  $\sigma_0$  profile measured at an azimuth angle  $\phi$  gives a 1D wave spectrum. By concatenation of these 1D wave spectra over  $\phi$ , we obtain a polar 2D spectrum covering more than  $360^\circ$  in  $\phi$  due to the multiple rotations of the SWIM antenna while the satellite is moving forward. In this way, the resulting spectrum respects the natural geometry of the sensor and corresponds to continuous waves observations. Individual wave systems should be well depicted in such a spectrum, even in complex cases such as heterogeneous seas or coastal areas. This section is dedicated to the partitioning of this  $\phi$ -concatenated spectrum, called ribbon partitioning because of the particular shape of the SWIM swath along the antenna rotations.

Regarding the partitioning methodology, there are not obvious reasons to consider the ribbon partitioning differently than a classical 2D spectrum partitioning. As a consequence, the methodology follows here the classical watershed approach (see [RD4] for an overview of different methodologies). In this approach, the spectrum is treated as an image and the watershed algorithm is used for segmentation. With the watershed algorithm, the wave spectrum is regarded as an inverted height map from which the limits of the adjacent catchment basins are found by faking a flood.

The major issue with the watershed approach is an over-segmentation due to the random variability or noise in observed wave spectra. An over-segmentation could lead to false partitions, either insignificant partitions or split partitions from a same wave system. It is therefore necessary to reduce the noise in the wave spectrum before applying the watershed algorithm. This is achieved with two processes:

- resampling the wave spectrum in the wavenumber dimension with far less points and with a logarithmic scale.
- smoothing the spectrum with a Gaussian kernel.


The resampling and smoothing will be performed with a resampling factor and Gaussian kernel size optimized to reduce the spectral noise while still maintaining the capability to distinguish between two close spectral swell peaks at least in the limit of the spectral and directional resolution of the SWIM instrument.

However, the noise reduction alone can not guarantee a perfect segmentation. In order to minimize the number of insignificant partitions, the watershed algorithm is applied only in the most energetic parts of the spectrum given a threshold and in a wavenumber range corresponding to the waves imaged by SWIM. Then, some extra rules are defined to deal with the remaining false partitions:

- merging the adjacent partitions when the valley between the two peaks is not enough pronounced. For each peak, the difference of energy with the valley is compared to a given threshold.
- discarding the partitions when their energy is small compared to the energy of the piece of ribbon around the partition.

As pointed out by Portilla in [RD4], this kind of strategy for merging partitions could “rely on rather arbitrary parameters that need to be adjusted from situation to situation”. In order to minimize this issue, the merging thresholds are relative to a noise level estimated from the spectra. Moreover, this noise level is also used with the threshold for removing some parts of the spectrum from the watershed. Notice that this approach is different from the one suggested by Portilla who prefers to use an iterative smoothing until the number of partitions detected is less than a prescribed number. This prescribed number of partitions is clearly meaningful but can be overconstraining and remains difficult to set, even more in the ribbon case.

### **Open Point 9 Partitioning parameters**

The strategy with the many thresholds for watershed/merge/discard needs to be heavily tested and potentially adapted, one of the key point being the estimation of noise from the spectrum. 

### **Open Point 10 Long swell weak signatures (partitioning)**

Some weak signals like the modulations of long waves could require a special processing in order to detect them. This special processing could be triggered by a first guess from a wave model or from propagated wave observations (fireworks with potentially SAR, SWIM, buoys and seismograph observations).

With respect to the partitioning:

- we have to take care that the noise reduction (k-resample and smoothing) before the partitioning does not “erase” the weak energy at long wavelengths
- we could adjust the partitioning parameters to be at least more tolerant for the foreground energy (part of spectrum sent to the watershed) and for the discard part. The direct use of the first guess (i.e. not just as a trigger) could also allow to adapt the parameters locally for some wavenumbers.

### **Open Point 11 Propagation ambiguity removal**

*Note: this open point has been moved here because the ambiguity removal is usually*

applied on the partitions (therefore after the partitioning).

If possible, a strategy will be defined to remove the propagation ambiguity from:

- a study of the use of cross-spectra with different mode/macrocycle (speckle mode, nominal and short macrocycle).
- the opportunity to use ancillary data like SAR fireworks for long waves and wave model for shorter waves.
- the possible upwind/crosswind modulation amplitude asymmetry.
- the possible change of dominant wavelength for swell propagating in the same direction but seen in the upwind and downwind direction. This spatial change in the dominant wavelength is caused by the wave dispersion relation where longer waves traveling faster are seen ahead of shorter waves from the same swell system. The rate of change depends linearly on the distance from the storm source, the further away the smaller the rate of change.

### Open Point 12 Merge beam

There are two scenarios (not necessarily exclusive) for taking advantage of the different beams:

- the wave systems are obtained separately for each beam but with the help of the other beams at some parts of the processing (e.g. MTF or propagation ambiguity removal).
- the wave systems from all beams are merged and/or used to reconstruct classical 2D spectra. This could lead to a new step (chapter) after the partitioning.

## 8.2 Block diagram

Figure 17 represents the block diagram of the ribbon partitioning step with the main functions in grey, the inputs from previous steps in yellow, the parameters in blue and the outputs (either new or updated variables) with no color.

This step is applied for each “wave” incidence angle (i.e. in  $\{6^\circ, 8^\circ, 10^\circ\}$ ). As a consequence, each variable is assumed to correspond to a given incidence angle, not denoted in the notations. Moreover, in a case of a non nominal macrocycle (such as  $(0^\circ, 10^\circ, 10^\circ)$ ), it is also assumed that the variables have been grouped by incidence angle instead of by cycles.

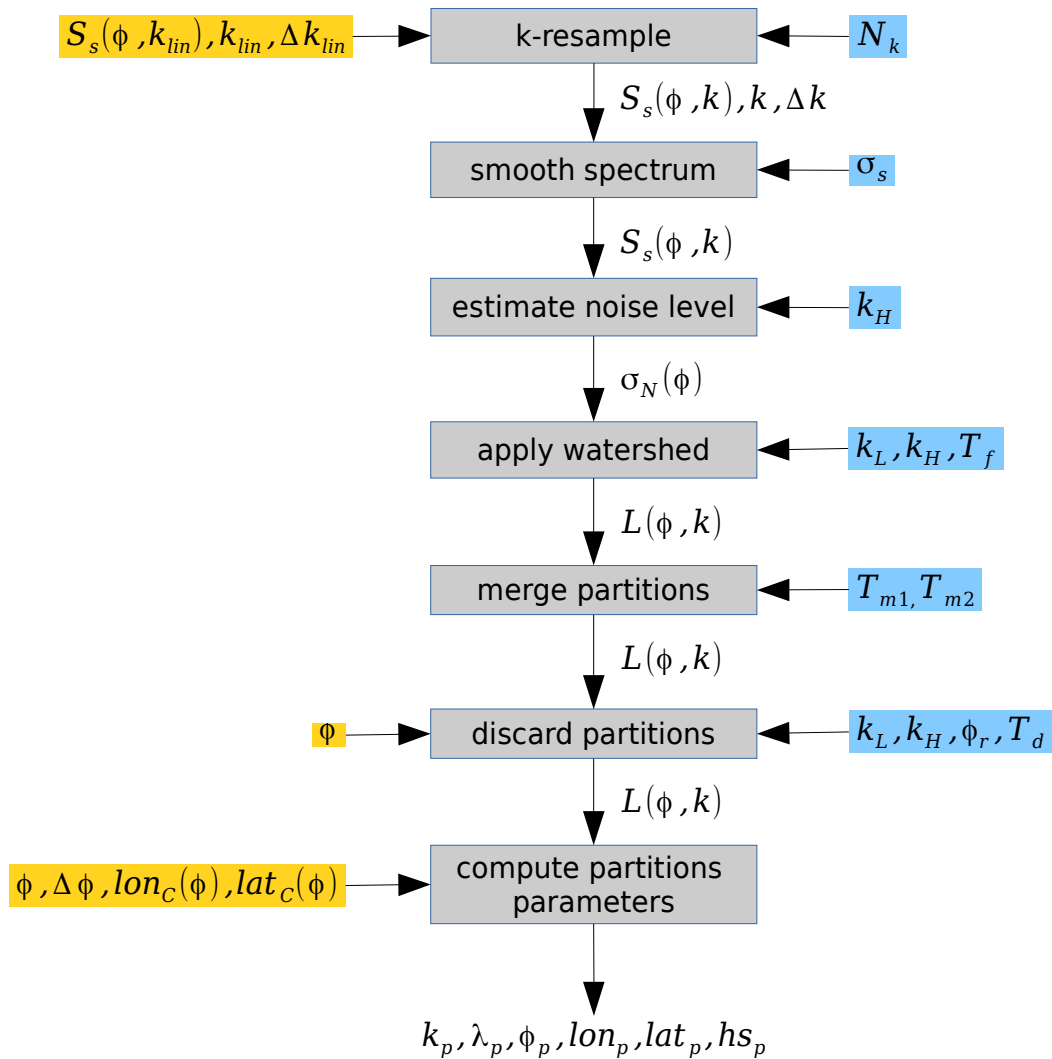


Figure 17: Block diagram of ribbon partitioning

## 8.3 Mathematical description

### 8.3.1 k-resample

<b>Objective:</b> resample the wave spectrum in the wavenumber dimension, mainly for noise reduction
<b>Inputs</b>
$S_s(\phi, k_{lin})$ : wave slope spectrum
$k_{lin}$ : linearly scaled wavenumbers from the spectral transform
$\Delta k_{lin}$ : wavenumber spacing from the spectral transform
<b>Parameters</b>



$N_k=50$ : number of $k$ bins for the resampling
<b>Outputs</b>
$S_s(\phi, k)$ : resampled wave slope spectrum
$k$ : logarithmic scaled wavenumbers
$\Delta k$ : wavenumber spacings

The first step is to compute wavenumbers with a logarithmic scale:

$$k_{\log}[i] = (\max(k_{lin}) - \min(k_{lin})) \frac{\exp(\frac{i}{10}) - 1}{\exp(\frac{N_k - 1}{10}) - 1} + \min(k_{lin}) \text{ for } i \in 0..N_k - 1$$

Then the resampling is done for each  $i$  in  $0..N_k - 1$  :

- find all indices  $j$  where:

- if  $i=0$  ,  $k_{lin}[j] < \frac{(k_{\log}[i] + k_{\log}[i+1])}{2}$

- if  $i=N_k - 1$  ,  $\frac{(k_{\log}[i-1] + k_{\log}[i])}{2} \leq k_{lin}[j]$

- otherwise,  $\frac{(k_{\log}[i-1] + k_{\log}[i])}{2} \leq k_{lin}[j] < \frac{(k_{\log}[i] + k_{\log}[i+1])}{2}$

and define  $N_j$  as the number of indices  $j$

- compute  $k[i] = \frac{\min_j(k_{lin}[j]) + \max_j(k_{lin}[j])}{2}$
- compute  $\Delta k[i] = N_j \Delta k_{lin}$
- compute  $S_s(\phi, k[i]) = \frac{1}{N_j} \sum_j S_s(\phi, k_{lin}[j])$

### 8.3.2 smooth spectrum

<b>Objective:</b> smooth the wave spectrum with a Gaussian kernel for noise reduction
<b>Inputs</b>
$S_s(\phi, k)$ : wave slope spectrum

<b>Parameters</b>
$\sigma_s=1$ : standard deviation of the smoothing Gaussian kernel (in pixels)
<b>Outputs</b>
$S_s(\phi, k)$ : updated wave slope spectrum in its smoothed version

Smoothing is achieved with the convolution of the spectrum and a Gaussian kernel. The convolution can be written as a sequence of two convolutions with a 1D Gaussian kernel  $G$ , i.e. in discrete form:

$$S_s[i, j] = \frac{1}{\sum_k G[k] \sum_l G[l]} \sum_k \sum_l S_s[i-k, j-l] G[k] G[l]$$

with  $G[k] = \exp\left(\frac{-k^2}{2\sigma_s^2}\right)$  for  $k \in -N_G \dots N_G$ , the kernel of length  $2N_G - 1$

By default,  $N_G$  depends on  $\sigma_s$  with  $N_G = \lfloor 4\sigma_s + 0.5 \rfloor$ .


### 8.3.3 estimate noise level

<b>Objective:</b> estimate the noise level in the wave spectrum
<b>Inputs</b>
$S_s(\phi, k)$ : wave slope spectrum
$k$ : wavenumbers
<b>Parameters</b>
$k_H = \frac{2\pi}{30}$ : high wavenumber threshold
<b>Outputs</b>
$\sigma_N(\phi)$ : noise level

Noise level is estimated from the spectrum mean energy at short wavelengths where waves are not imaged by SWIM. The parameter  $k_H$  defines the lowest wavenumber from which the noise is estimated:

$$\sigma_N(\phi) = \overline{S_s(\phi, k \geq k_H)} \text{ for each } \phi$$

### 8.3.4 apply watershed

<b>Objective:</b> apply the watershed algorithm as a first draft of the partitioning
<b>Inputs</b>
$S_s(\phi, k)$ : wave slope spectrum
$k$ : wavenumbers
$\sigma_N(\phi)$ : noise level
<b>Parameters</b>
$k_L = \frac{2\pi}{1000}$ : low wavenumber threshold
$k_H = \frac{2\pi}{30}$ : high wavenumber threshold
$T_f = 1.5$ : foreground energy threshold, relative to $\sigma_N(\phi)$ 
<b>Outputs</b>
$L(\phi, k)$ : labeled matrix

Wave systems are only searched in a specific range of wavenumbers and where the spectrum energy is high enough. This lead to the definition of the foreground matrix  $F(\phi, k)$  :

- $F(\phi, k) = True$  where  $k_L < k < k_H$  and  $S_s(\phi, k) > T_f \sigma_N(\phi)$
- $F(\phi, k) = False$  otherwise (background part)

Then, energy peaks are searched into the foreground part of  $S_s(\phi, k)$  , i.e. where  $F(\phi, k) = True$  . Peaks are defined here as the local maxima in a region of 3 by 3 pixels. It consists of finding all  $(i, j)$  pairs of indices such as:

$$F[i, j] = True \text{ and } S_s[i, j] = \max_{k, l} (S_s[i-k, j-l]) \text{ for } k, l \in -1..1$$

For  $N_p$  peaks labeled with  $i$  in  $1..N_p$  and found at  $(\phi_i, k_i)$  , let's define the peak matrix  $P(\phi, k)$  :

- $P(\phi_i, k_i) = i$
- $P(\phi, k) = 0$  otherwise

Finally, the watershed segmentation algorithm is applied on the opposite spectrum

$-S_s(\phi, k)$  since the algorithm delimits basins of energy. The peak matrix  $P(\phi, k)$  tells the algorithm where are the markers (or seeds) for the basins. And the foreground matrix  $F(\phi, k)$  makes the algorithm only segment the foreground part of the spectrum.

$$L(\phi, k) = \text{watershed}(\text{image} = -S_s(\phi, k), \text{markers} = P(\phi, k), \text{mask} = F(\phi, k))$$

The watershed output  $L(\phi, k)$  defines a label for each pixel. With  $N_L$  partitions:

- $L(\phi, k) = i$  for pixels of the partition with label  $i \in 1..N_L$
- $L(\phi, k) = 0$  for background

#### Notes on watershed function:

This function implements a watershed algorithm that apportions pixels into marked basins: see [http://en.wikipedia.org/wiki/Watershed\\_%28image\\_processing%29](http://en.wikipedia.org/wiki/Watershed_%28image_processing%29) or <http://cmm.ensmp.fr/~beucher/wtshed.html>.

The algorithm uses a priority queue to hold the pixels with the metric for the priority queue being pixel value, then the time of entry into the queue - this settles ties in favor of the closest marker.

Some ideas taken from Soille, "Automated Basin Delineation from Digital Elevation Models Using Mathematical Morphology", Signal Processing 20 (1990) 171-182

The most important insight in the paper is that entry time onto the queue solves two problems: a pixel should be assigned to the neighbor with the largest gradient or, if there is no gradient, pixels on a plateau should be split between markers on opposite sides.

### 8.3.5 merge partitions

<b>Objective:</b> merge the partitions when the valley between them is not enough pronounced
<b>Inputs</b>
$S_s(\phi, k)$ : wave slope spectrum
$L(\phi, k)$ : labeled matrix
$\sigma_N(\phi)$ : noise level
<b>Parameters</b>
$T_{m1} = 1$ : merge threshold 1, relative to $\sigma_N(\phi)$
$T_{m2} = 2$ : merge threshold 2, relative to $\sigma_N(\phi)$
<b>Outputs</b>

$L(\phi, k)$  : updated labeled matrix

For a given partition with label  $p$ , let's define:

- the indices of the pixels in the partition: the pairs  $(i_p, j_p)$  such as  $L[i_p[k], j_p[k]] = p$  for  $k \in 0..N_p - 1$  with  $N_p$  the number of pixels in the partition.
- the indices of the peak (local maximum) in the partition: the pair  $(i_p^P, j_p^P)$  such as  $S_s[i_p^P, j_p^P] = \max(S_s[i_p[k], j_p[k]])$  with  $k \in 0..N_p - 1$ .

And for two partitions with labels  $p$  and  $q$ :

- the indices of the pixels of the path between partition  $p$  and  $q$ : the pairs  $(i_{pq}, j_{pq})$  of the  $N_{pq}$  pixels intersecting a straight line in the pixel space between the peaks of partitions  $p$  and  $q$ :  $(i_p^P, j_p^P)$  and  $(i_q^P, j_q^P)$ .
- the indices of the valley of the path between partition  $p$  and  $q$ : the pair  $(i_{pq}^V, j_{pq}^V)$  such as  $S_s[i_{pq}^V, j_{pq}^V] = \min(S_s[i_{pq}[k], j_{pq}[k]])$  with  $k \in 0..N_{pq} - 1$ .

Then, two partitions with labels  $p$  and  $q$  are said connected if:

- at least one pixel in partition  $p$  is connected to at least one pixel in partition  $q$ . The pixel connectivity is here in a 4-connected way, i.e. connected horizontally and vertically:

$$\exists(k, l): (|i_p[k] - i_q[l]| = 1 \text{ and } j_p[k] = j_q[l]) \text{ or } (i_p[k] = i_q[l] \text{ and } |j_p[k] - j_q[l]| = 1)$$

- and all the pixels in the path between partitions  $p$  and  $q$  belong to the partitions  $p$  or  $q$ :

$$\forall k \in 0..N_{pq} - 1: L[i_{pq}[k], j_{pq}[k]] = p \text{ or } L[i_{pq}[k], j_{pq}[k]] = q$$

From the previous definitions, a connectivity matrix  $C$  of shape  $[N_L, N_L]$  is then defined in order to know which partitions are connected and to allocate some weights according to their connection. For two partitions with labels  $p$  and  $q$ ,  $C$  is defined as follows:

- if  $p = q$  then  $C[p-1, q-1] = C[q-1, p-1] = +\infty$
- if partition  $p$  is not connected to partition  $q$  then  $C[p-1, q-1] = C[q-1, p-1] = +\infty$
- otherwise, the connectivity weights are set to the difference of energy between the two peaks and the valley:

- $C[p-1, q-1] = S_s[i_p^P, j_p^P] - S_s[i_{pq}^V, j_{pq}^V]$
- $C[q-1, p-1] = S_s[i_q^P, j_q^P] - S_s[i_{pq}^V, j_{pq}^V]$

Given  $C$ , two partitions with labels  $p$  and  $q$  are mergeable if the connectivity weights are inferior to some thresholds:

- $\min(C[p-1, q-1], C[q-1, p-1]) \leq T_{m1} \sigma_N[i_{pq}^V]$
- $\max(C[p-1, q-1], C[q-1, p-1]) \leq T_{m2} \sigma_N[i_{pq}^V]$

with  $\sigma_N[i_{pq}^V]$  the estimated noise level at the corresponding azimuth  $\phi[i_{pq}^V]$  of the valley between  $p$  and  $q$ .

From an algorithmic point of view, the merge process is iterative until  $\min(C) = +\infty$ :

- find  $p$  and  $q$  such as  $\min(C[p-1, q-1] + C[q-1, p-1]) = \min(C + C^T)$ .
- if  $p$  and  $q$  are not mergeable, set  $C[p-1, q-1] = C[q-1, p-1] = +\infty$  and continue.
- otherwise, merge the partition of lowest peak energy inside the partition of highest peak energy then update the labels  $L$  and the connectivity  $C$ . If  $p$  (resp.  $q$ ) is the partition of lowest (resp. highest) peak energy:
  - $L[i_p[k], j_p[k]] = q$  for  $k \in 0..N_p - 1$
  - $C[p-1, k] = C[k, p-1] = +\infty$  for  $k \in 0..N_L - 1$
  - recompute  $C[q-1, k]$  and  $C[k, q-1]$  for  $k \in 0..N_L - 1$

After the merge process,  $L$  is relabeled in  $1..N_L$  with  $N_L$  the new number of partitions.

### 8.3.6 discard partitions

<b>Objective:</b> discard the partitions of low energy
<b>Inputs</b>
$S_s(\phi, k)$ : wave slope spectrum
$k$ : wavenumbers
$\phi$ : azimuths
$L(\phi, k)$ : labeled matrix
<b>Parameters</b>

$k_L = \frac{2\pi}{1000}$ : low wavenumber threshold
$k_H = \frac{2\pi}{30}$ : high wavenumber threshold
$\phi_r = \pi$ : range of $\phi$ to consider around each partition
$T_d = 2.5$ : discard threshold (per cent)
<b>Outputs</b>
$L(\phi, k)$ : updated labeled matrix

N.B.: the indices  $(i_p, j_p)$  of the pixels in the partition  $p$  are defined in section 8.3.5.

For a given partition with label  $p$ , let's define:

- the energy of partition  $p$  :  $E_p = \sum_{k \in 0..N_p-1} S_s[i_p[k], j_p[k]]$

- the indices  $(i_p^C, j_p^C)$  of the weighted centroid of partition  $p$  :

$$i_p^C = \frac{1}{E_p} \sum_{k \in 0..N_p-1} i_p[k] S_s[i_p[k], j_p[k]] \quad \text{and} \quad j_p^C = \frac{1}{E_p} \sum_{k \in 0..N_p-1} j_p[k] S_s[i_p[k], j_p[k]]$$

- the total energy around the partition  $p$  :  $E_p^T = \sum_{\phi, k} S_s(|\phi - \phi[i_p^C]| \leq \frac{\phi_r}{2}, k_L < k < k_H)$ , the energy over  $\phi_r$  in azimuth around the weighted centroid and in the wavenumber range  $]k_L, k_H[$

Then, the discard process consists of removing the partitions where the ratio between  $E_p$  and  $E_p^T$  is under a given threshold:

- if  $\frac{E_p}{E_p^T} \leq \frac{T_d}{100}$  then  $L[i_p[k], j_p[k]] = 0$  for  $k \in 0..N_p-1$

After the discard process,  $L$  is relabeled in  $1..N_L$  with  $N_L$  the new number of partitions.

### 8.3.7 compute partitions parameters

<b>Objective:</b> compute the integral parameters for each partition
<b>Inputs</b>
$S_s(\phi, k)$ : wave slope spectrum

$k$ : wavenumbers
$\Delta k$ : wavenumber spacings
$\phi$ : azimuths
$\Delta \phi$ : azimuth spacings
$L(\phi, k)$ : labeled matrix
$lon_C(\phi)$ : longitude at the footprint center for each azimuth
$lat_C(\phi)$ : latitude at the footprint center for each azimuth
<b>Outputs</b>
$k_p$ : dominant wavenumber for each partition
$\lambda_p$ : dominant wavelength for each partition
$\phi_p$ : dominant wave direction for each partition
$lon_p$ : associated longitude for each partition
$lat_p$ : associated latitude for each partition
$hs_p$ : significant wave height for each partition

N.B.: the indices  $(i_p, j_p)$  of the pixels in the partition  $p$  are defined in section 8.3.5 and the indices  $(i_p^C, j_p^C)$  of the weighted centroid of partition  $p$  are defined in section 8.3.6.

For a given partition with label  $p$ , let's define:

- $k_p = k(j_p^C)$  the cubic interpolation of  $k$  at  $j_p^C$ , as the dominant wavenumber of  $p$
- $\lambda_p = \frac{2\pi}{k_p}$ , as the dominant wavelength of  $p$
- $\phi_p = \phi(i_p^C)$  the linear interpolation of  $\phi$  at  $i_p^C$ , as the dominant wave direction of  $p$
- $lon_p = lon_C(i_p^C)$  the cubic interpolation of  $lon_C$  at  $i_p^C$ , as the longitude of  $p$
- $lat_p = lat_C(i_p^C)$  the cubic interpolation of  $lat_C$  at  $i_p^C$ , as the latitude of  $p$
- $hs_p = 4\sqrt{E}$  with  $E = \sum_{(i_p, j_p)} \frac{S_s[i_p, j_p] \Delta k[j_p] \Delta \phi[i_p]}{k[j_p]}$ , as the significant wave height of  $p$ .



N.B.: with the wave height spectrum  $S_h = \frac{S_s}{k^2}$ ,  $E$  is written  $\sum S_h k \Delta k \Delta \phi$ .

## 9 SWIM L2S product

The two main outputs of SWIM L2S processing are:

- the wave spectrum in the ribbon space, i.e. one  $\phi$ -concatenated spectrum for the whole acquisition.
- the wave partitions located in time and space (given the location inside the ribbon) and the associated integrated wave parameters such as dominant wave direction, dominant wavelength and significant wave height.

These outputs are produced for each “wave” incidence angle present in the macrocycle. For example, with:

- a nominal macrocycle ( $\{0^\circ, 2^\circ, 4^\circ, 6^\circ, 8^\circ, 10^\circ\}$ ), there are 3 wave spectra and 3 sets of partitions: for  $6^\circ$ ,  $8^\circ$  and  $10^\circ$ .
- a short macrocycle (such as  $\{0^\circ, 8^\circ, 8^\circ\}$ ), there is only one wave spectrum and one set of partitions: for  $8^\circ$ .

In terms of product, multiple NetCDF files (NetCDF-4 version) are produced:

- a “L2S Expert” NetCDF file for each incidence angle (i.e. from 1 to 3 files given the macrocycle) containing the wave spectrum, the matrix of partition labels and the partition parameters, as well as many L2S intermediate processing variables.
- 1 “L2S Simplified” NetCDF file containing only the main results of L2S processing (subset of “L2S Expert” files). This output is not yet defined.

The following gives an overview of current L2S Expert contents:

```
netcdf CF0_T04801_SWI_L2S06__F_20160823T000000_20160823T014500 {
```

```
dimensions:
```

```
    time = 15739 ;
```

```
    range = 2014 ;
```

```
    segment = 15 ;
```

```
    klin = 129 ;
```

```
    k = 60 ;
```

```
    partition = 991 ;
```

```
variables:
```

```
    double time(time) ;
```

```
        time:_FillValue = 9.96920996838687e+36 ;
```

```

time:long_name = "time" ;
time:standard_name = "time" ;
time:units = "seconds since 2009-01-01T00:00:00Z" ;
float lat(time) ;
lat:_FillValue = 9.96921e+36f ;
lat:long_name = "latitude at mid range" ;
lat:standard_name = "latitude" ;
lat:units = "degrees_north" ;
float lon(time) ;
lon:_FillValue = 9.96921e+36f ;
lon:long_name = "longitude at mid range" ;
lon:standard_name = "latitude" ;
lon:units = "degrees_east" ;
float incidence(time) ;
incidence:_FillValue = 9.96921e+36f ;
incidence:long_name = "incidence angle at mid range" ;
incidence:units = "degree" ;
float near_lat(time) ;
near_lat:_FillValue = 9.96921e+36f ;
near_lat:long_name = "latitude at near range" ;
near_lat:units = "degrees_north" ;
float near_lon(time) ;
near_lon:_FillValue = 9.96921e+36f ;
near_lon:long_name = "longitude at near range" ;
near_lon:units = "degrees_east" ;
float near_incidence(time) ;
near_incidence:_FillValue = 9.96921e+36f ;
near_incidence:long_name = "incidence angle at near range" ;
near_incidence:units = "degree" ;
float far_lat(time) ;
far_lat:_FillValue = 9.96921e+36f ;

```

```

    far_lat:long_name = "latitude at far range" ;
    far_lat:units = "degrees_north" ;
float far_lon(time) ;
    far_lon:_FillValue = 9.96921e+36f ;
    far_lon:long_name = "longitude at far range" ;
    far_lon:units = "degrees_east" ;
float far_incidence(time) ;
    far_incidence:_FillValue = 9.96921e+36f ;
    far_incidence:long_name = "incidence angle at far range" ;
    far_incidence:units = "degree" ;
byte flag_availability(time) ;
    flag_availability:_FillValue = -127b ;
    flag_availability:long_name = "flag on l0 data validity" ;
    flag_availability:flag_values = 0b, 1b, 2b, 3b ;
    flag_availability:flag_meanings = "error valid warning no_data" ;
float phi(time) ;
    phi:_FillValue = 9.96921e+36f ;
    phi:long_name = "azimuth angle in local orbital reference frame, clockwise
from satellite velocity vector" ;
    phi:units = "degree" ;
float phi_geo(time) ;
    phi_geo:_FillValue = 9.96921e+36f ;
    phi_geo:long_name = "geographical azimuth angle, clockwise from north" ;
    phi_geo:units = "degree" ;
float ly(time) ;
    ly:_FillValue = 9.96921e+36f ;
    ly:long_name = "ground length of azimuth beam footprint (at 3dB)" ;
    ly:units = "m" ;
byte sigma0_flag(time, range) ;
    sigma0_flag:_FillValue = -127b ;
    sigma0_flag:long_name = "sigma0 flag" ;

```

```

sigma0_flag:flag_masks = 1b, 2b ;
sigma0_flag:flag_meanings = "l0_no_data land" ;
float sigma0(time, range) ;
sigma0:_FillValue = 9.96921e+36f ;
sigma0:least_significant_digit = 3LL ;
sigma0:long_name = "sigma0" ;
sigma0:units = "dB" ;
float sigma0_trend(time, range) ;
sigma0_trend:_FillValue = 9.96921e+36f ;
sigma0_trend:least_significant_digit = 3LL ;
sigma0_trend:long_name = "sigma0 trend" ;
sigma0_trend:units = "dB" ;
float sigma0_mean(time) ;
sigma0_mean:_FillValue = 9.96921e+36f ;
sigma0_mean:long_name = "sigma0 mean" ;
sigma0_mean:units = "dB" ;
sigma0_mean:comment = "computed from linear sigma0" ;
float sigma0_nv(time) ;
sigma0_nv:_FillValue = 9.96921e+36f ;
sigma0_nv:long_name = "sigma0 normalized variance" ;
sigma0_nv:comment = "computed from linear sigma0 and after multiplicative
detrend" ;
float sigma0_skewness(time) ;
sigma0_skewness:_FillValue = 9.96921e+36f ;
sigma0_skewness:long_name = "sigma0 skewness" ;
sigma0_skewness:comment = "computed from linear sigma0 and after
multiplicative detrend" ;
float sigma0_kurtosis(time) ;
sigma0_kurtosis:_FillValue = 9.96921e+36f ;
sigma0_kurtosis:long_name = "sigma0 kurtosis" ;
sigma0_kurtosis:comment = "computed from linear sigma0 and after
multiplicative detrend" ;

```

```

int seg_start(segment) ;
    seg_start:long_name = "start position in range dimension of overlapping
segments used for spectral estimation" ;
int seg_stop(segment) ;
    seg_stop:long_name = "stop position in range dimension of overlapping
segments used for spectral estimation" ;
float seg_lat(time, segment) ;
    seg_lat:_FillValue = 9.96921e+36f ;
    seg_lat:long_name = "latitude at mid segment" ;
    seg_lat:units = "degrees_north" ;
float seg_lon(time, segment) ;
    seg_lon:_FillValue = 9.96921e+36f ;
    seg_lon:long_name = "longitude at mid segment" ;
    seg_lon:units = "degrees_east" ;
float seg_incidence(time, segment) ;
    seg_incidence:_FillValue = 9.96921e+36f ;
    seg_incidence:long_name = "incidence angle at mid segment" ;
    seg_incidence:units = "degree" ;
float seg_sigma0_mean(time, segment) ;
    seg_sigma0_mean:_FillValue = 9.96921e+36f ;
    seg_sigma0_mean:long_name = "sigma0 mean for each segment" ;
    seg_sigma0_mean:comment = "computed from linear sigma0" ;
float seg_sigma0_nv(time, segment) ;
    seg_sigma0_nv:_FillValue = 9.96921e+36f ;
    seg_sigma0_nv:long_name = "sigma0 normalized variance for each segment" ;
    seg_sigma0_nv:comment = "computed from linear sigma0 and after
multiplicative detrend" ;
float seg_sigma0_skewness(time, segment) ;
    seg_sigma0_skewness:_FillValue = 9.96921e+36f ;
    seg_sigma0_skewness:long_name = "sigma0 skewness for each segment" ;
    seg_sigma0_skewness:comment = "computed from linear sigma0 and after
multiplicative detrend" ;

```

```

float seg_sigma0_kurtosis(time, segment) ;
    seg_sigma0_kurtosis:_FillValue = 9.96921e+36f ;
    seg_sigma0_kurtosis:long_name = "sigma0 kurtosis for each segment" ;
    seg_sigma0_kurtosis:comment = "computed from linear sigma0 and after
multiplicative detrend" ;
float seg_sea_ice_concentration(time, segment) ;
    seg_sea_ice_concentration:_FillValue = 9.96921e+36f ;
    seg_sea_ice_concentration:long_name = "sea ice concentration at mid
segment" ;
    seg_sea_ice_concentration:units = "percent" ;
float seg_bathymetry(time, segment) ;
    seg_bathymetry:_FillValue = 9.96921e+36f ;
    seg_bathymetry:long_name = "bathymetry at mid segment" ;
    seg_bathymetry:units = "m" ;
float seg_model_u10(time, segment) ;
    seg_model_u10:_FillValue = 9.96921e+36f ;
    seg_model_u10:long_name = "model eastward wind at mid segment" ;
    seg_model_u10:units = "m s-1" ;
float seg_model_v10(time, segment) ;
    seg_model_v10:_FillValue = 9.96921e+36f ;
    seg_model_v10:long_name = "model northward wind at mid segment" ;
    seg_model_v10:units = "m s-1" ;
byte seg_flag(time, segment) ;
    seg_flag:_FillValue = -127b ;
    seg_flag:long_name = "segment flag" ;
    seg_flag:flag_masks = 1b, 2b, 4b, 8b, 16b ;
    seg_flag:flag_meanings = "used_in_spectral_estimation l0_no_data land
low_bathy sea_ice" ;
float fluctuation_spectra(time, klin, segment) ;
    fluctuation_spectra:_FillValue = 9.96921e+36f ;
    fluctuation_spectra:least_significant_digit = 4LL ;
    fluctuation_spectra:long_name = "fluctuation spectra" ;

```

```

float klin(klin) ;
    klin:_FillValue = 9.96921e+36f ;
    klin:long_name = "wavenumber" ;
    klin:units = "rad m-1" ;
float modulation_spectra(time, klin, segment) ;
    modulation_spectra:_FillValue = 9.96921e+36f ;
    modulation_spectra:least_significant_digit = 4LL ;
    modulation_spectra:long_name = "modulation spectra" ;
float mtf(time, segment) ;
    mtf:_FillValue = 9.96921e+36f ;
    mtf:long_name = "modulation transfer function" ;
float wave_spectra(time, k) ;
    wave_spectra:_FillValue = 9.96921e+36f ;
    wave_spectra:least_significant_digit = 4LL ;
    wave_spectra:long_name = "wave slope spectra" ;
float k(k) ;
    k:_FillValue = 9.96921e+36f ;
    k:long_name = "wavenumber" ;
    k:units = "rad m-1" ;
float dk(k) ;
    dk:_FillValue = 9.96921e+36f ;
    dk:long_name = "wavenumber bin width" ;
    dk:units = "rad m-1" ;
int partition_label(time, k) ;
    partition_label:_FillValue = -2147483647 ;
    partition_label:long_name = "partition label" ;
    partition_label:comment = "0 indicates no partition, otherwise goes from 1
to n where n is equal to partition dimension length" ;
float partition_hs(partition) ;
    partition_hs:_FillValue = 9.96921e+36f ;
    partition_hs:long_name = "partition Hs" ;

```



```

    partition_hs:units = "m" ;
float partition_wavelength(partition) ;
    partition_wavelength:_FillValue = 9.96921e+36f ;
    partition_wavelength:long_name = "partition wavelength" ;
    partition_wavelength:units = "m" ;
float partition_direction(partition) ;
    partition_direction:_FillValue = 9.96921e+36f ;
    partition_direction:long_name = "partition direction, clockwise from
north" ;
    partition_direction:units = "degree" ;
    partition_direction:comment = "propagation ambiguity not removed, the
sense of pdir is by default the same as phi_geo" ;
double partition_time(partition) ;
    partition_time:_FillValue = 9.96920996838687e+36 ;
    partition_time:long_name = "partition time" ;
    partition_time:units = "seconds since 2009-01-01T00:00:00Z" ;
float partition_lat(partition) ;
    partition_lat:_FillValue = 9.96921e+36f ;
    partition_lat:long_name = "partition latitude" ;
    partition_lat:units = "degrees_north" ;
float partition_lon(partition) ;
    partition_lon:_FillValue = 9.96921e+36f ;
    partition_lon:long_name = "partition longitude" ;
    partition_lon:units = "degrees_east" ;

```