Fiducial Reference Measurements for Satellite Ocean Colour Phase-2

Protocols for uncertainty budget calculation of FRMOCnet OCR and practical guide for OCR measurement end-to-end uncertainty budget calculation (FRM4SOC2-D10)

Title	Protocols for uncertainty budget calculation of FRMOCnet OCR and practical guide for OCR measurement end-to-end uncertainty budget calculation
Document reference	FRM4SOC2-D10
Project	EUMETSAT – FRM4SOC Phase-2
Contract	EUMETSAT Contract No. EUM/CO/21/460002539/JIG
Deliverable	D-10
Version	v.2.4
Date issued	21.04.2023

Prepared By		Approved by	
Name:	Agnieszka Bialek Ashley Ramsay Pieter De Vis	Name:	
Organisation:	NPL	Organisation:	EUMETSAT
Position:	Senior Research Scientist Research Scientist	Position:	
Date:	21.04.2023	Date:	
Signature:		Signature:	Juan Ignacio Gossn Digitally signed by Juan Ignacio Gossn Date: 2023.05.15 14:57:45 +02'00'

Funded by the European Union

European Commission

Copernicus Europe's eyes on Earth

EUMETSAT

Document Control Table

Title	Protocols for uncertainty budget calculation of FRMOCnet OCR and practical guide for OCR measurement end-to-end uncertainty budget calculation
Document reference	FRM4SOC2-D10
Project	EUMETSAT – FRM4SOC Phase-2
Contract	EUMETSAT Contract No. EUM/CO/21/460002539/JIG
Deliverable	D-10
Version	v.2.3
Date Issued	06/03/2023

Document Change Record

Index	Issue	Revision	Date	Brief description	Issued by
1	0	1	22/02/2022	First draft	A. Bialek A. Ramsay P. De Vis
2	2	2	30/11/2022	Second draft	A. Bialek
3	2	3	06/03/2023	Released to the consortium and EUMETSAT	A.Bialek
4	2	4	21/04/2023	Final version	A.Bialek

Distribution List

Company/Organisation	Name	Format	No. of Copies
UT	Riho Vendt Karin Pai	Electronic file – pdf and original (WORD) file	1 pdf 1 WORD file
ACRI-ST	Christphe Lerebourg Sébastien Clerc	Electronic file – pdf and original (WORD) file	1 pdf 1 WORD file
Brockmann Consult	Carsten Brockmann Helge Dzierzon	Electronic file – pdf and original (WORD) file	1 pdf 1 WORD file
NPL	Agnieszka Bialek	Electronic file – pdf and original (WORD) file	1 pdf 1 WORD file
PML	Gavin Tilstone	Electronic file – pdf and original (WORD) file	1 pdf 1 WORD file
RBINS	Kevin Ruddick	Electronic file – pdf and original (WORD) file	1 pdf 1 WORD file
EUMETSAT	Juan Ignacio Gossn Ewa Kwiatkowska	Electronic file – pdf and original (WORD) file	1 pdf 1 WORD file

Acronyms and Abbreviations

Acronym	Description
AAOT	Acqua Alta Oceanographic Tower
AERONET-OC	The Ocean Color component of the Aerosol Robotic Network
AMT	Atlantic Meridional Transect
API	Application Program Interface
ARC	Assessment of In Situ Radiometric Capabilities for Coastal Water Remote Sensing Applications
BRDF	Bidirectional reflectance distribution function
Cal	Calibration
CCPR	Consultative Committee for Photometry and Radiometry
CEOS	Committee on Earth Observation Satellites
Char	Characterization
CIPM	Comité International des Poids et Mesures (International Committee for Weights and Measures)
CIMP MRA	CIPM Mutual Recognition Arrangement
HyperCP	HyperInSPACE Community Processor
EO	Earth Observation
ESA	European Space Agency
EUMETSAT	European Organisation for the Exploitation of Meteorological Satellites
FOV	Field of view
FRM	Fiducial Reference Measurements
FRMOCnet	Copernicus FRM-certified OC instrument network
FRM4SOC	Fiducial Reference Measurements for Satellite Ocean Colour
FWHM	Full Width at Half Maximum
GEO	Group on Earth Observations
GUM	Guide to the expression of Uncertainty in Measurement
ILAC	International Laboratory Accreditation Cooperation
IOCCG	International Ocean-Colour Coordinating Group
LPU	The Law of Propagation of Uncertainties
LSF	Line-Spread Function
LUT	Look Up Table
MC	Monte Carlo
MERIS	Medium Resolution Imaging Spectrometer
MVT	MERIS Validation Team
NASA	National Aeronautics and Space Administration
NERC	Natural Environment Research Council
NMI	National Metrology Institute
NPL	National Physical Laboratory
OC	Ocean Colour
OCDB	Ocean Colour Database
OCR	Ocean Colour Radiometer
PDF	Probability Distribution Function
PML	Plymouth Marine Laboratory
Punpy	Propagating Uncertainties with PYthon
QA	Quality Assurance
QA4EO	Quality Assurance framework for Earth Observation
QC	Quality Control
QTH	Quartz tungsten halogen
ROI	Return On Investment
RSP	Remote Sensing and Products Division
RD	Reference Document
S3	Sentinel-3
S3VT-OC	Sentinel-3 Validation Team – Ocean Colour group
SeaWiFS	Sea-Viewing Wide Field-of-View Sensor
SIRREX	SeaWiFS Intercalibration Round Robin Experiments
SI	International System of Units
SOW	Statement of Work
SST	Sea Surface Temperature
TO	Tartu Observatory, University of Tartu
TR	Technical Report
TSM	Total suspended material
UT	University of Tartu
VAL	Validation
VIM	International Vocabulary of Metrology

Contents

Applicable documents

ID	Description
[AD-0]	Statement of Work for FRM4SOC phase2, EUM/RSP/SOW/19/1131157
[AD-1]	FRM4SOC2-D7 Complete characterisation and calibration results for FRMOCnet OCR models and re-characterisation routine: an update (available at: FRM4SOC2SharePoint/Documents/Deliverable/D7)
[AD -2]	FRM4SOC2-D12 Harmonised cal/char lab guidelines, including lab protocols for FRMOCnet OCR models (available at: FRM4SOC2SharePoint/Documents/Deliverable/D12)
[AD-3]	FRM4SOC-D18 Community processor Architecture Design and User Manual document (ADUM) (available at: FRM4SOC2SharePoint/Documents/Deliverable/D18)
[AD-4]	FRM4SOC-D26 FRM OCR instrument calibration and Field Inter-Comparison Experiment report (available at: FRM4SOC2SharePoint/Documents/Deliverable/D126).

1 Scope

The current document is the Protocols for uncertainty budget calculation of FRMOCnet OCR and practical guide for OCR measurement end-to-end uncertainty budget calculation as required by the terms of the Invitation To Tender (ITT) No. 20/220036 "Copernicus – Fiducial Reference Measurements for Satellite Ocean Colour (FRM4SOC phase-2) issued by EUMETSAT. This document forms deliverable D-10 v2 of the FRM4SOC phase-2 project.

The main objective of this task is to develop and finalise the uncertainty budget evaluation started in FRM4SOC phase-1. The foreseen updates are discussed in more detail below, but primarily this will be an extension to include remote sensing reflectance (R_{rs}) as measurand, as well as including a more detailed study on some uncertainty inputs components like the sea surface reflectance factor (ρ). Two instrument models are addressed during the FRM4SOC-2 study: TriOS and HyperOCR.

The outputs of this activity were disseminated to the community at the project workshop in December 2022, as well as in a short practical guide for laboratories to provide them with clear approach to preparing uncertainty evaluations for their measurements. Further, the protocol is also a key input to the uncertainty evaluation in the community processor development activity. To ensure that the community processor can be applied by many users the uncertainty evaluation is split into two branches. The default branch is for users who do not have their instruments fully characterised, in this branch none of the instrument related errors are corrected for, but the uncertainty is estimated for each effect based of instrument class-based characterisation provided in [AD-2]. The second branch (so called FRM branch) is designed for the users who have individual radiometer characterisation records; thus, the correction can be applied based on instrument number, this branch is still until development, at present, and will be completed for v3 of this document.

2 Compatibility

Table 2-1. Compatibility

No.	Requirement
1.	**SOW-REQ-32**: Protocols for uncertainty budget calculation shall be finalized and include additional uncertainty components to compose the full uncertainty budget for the instrument and the measurement. Starting point for this activity shall be phase 1 protocols. The protocols shall also incorporate the outcome of measurement protocols from Task 3 and instrument characterizations from Task 4. An optimal calibration protocol shall be established that will enable to provide robust absolute calibration coefficients
2.	**SOW-REQ-32**: The protocols shall provide uncertainty budget calculation for in situ data processing leading to remote sensing reflectance and fully normalised water-leaving radiance
3.	**SOW-REQ-34** The deliverable shall include a short practical guide for OCR measurement end-to-end uncertainty budget calculation, allowing to reproduce the same routines in other laboratories different than the pilot one, selected by the Contractor
4.	**SOW-REQ-35** The protocols shall be delivered to EUMETSAT as a first version v.1 of D-10, and shall be made accessible by broader community for further review and acquiescence (when agreed by the review board experts)

Funded by the European Union

3 Introduction

FRM4SOC phase-1 endeavoured to build a method for calculating uncertainty budgets for *in situ* Ocean Colour measurements (Banks *et al.*, 2020; Białek *et al.*, 2020). FRM4SOC phase-2 now aims to apply these findings broadly within the OC community, in part, through adding uncertainty budget calculations to a 'community processor', the HyperCP. By packaging complete uncertainty budgets within the processor's output, much progress can be made towards making *in situ* OC data FRM compliant.

FRM4SOC-1 began to quantify instrument related uncertainty for the TriOS radiometers used *in situ*. These were propagated with absolute calibration uncertainty to generate complete standard uncertainties for downwelling irradiance (E_d) and water leaving radiance (L_w). The project successfully demonstrated how to conduct an unbroken chain of SI traceable calibrations and comparisons for *in situ* Ocean Colour measurements. FRM4SOC-1 described the state-of-the-art of OC uncertainty evaluation, particularly the need for data to be corrected for stray-light, non-linearity, temperature dependence, and cosine response in the case of downwelling irradiance. Some uncertainty components were not evaluated in the frame of FRM4SOC-1 project, thus the information about their estimates was taken from literature. This was the case for polarisation uncertainty and the sea surface reflectance factor (ρ). The uncertainty budget calculated in FRM4SOC-1 focused on instrument related effects and has shown the difference in uncertainty for the case when the instrument correction was applied versus the case without any corrections. The instruments from the University of Tartu, which had their characterisation done, were used. The uncertainty budget was evaluated separately from the normal data processing chain, using the data provided by the instrument owner. In the frame of the FRM4SOC-2 project the following changes are applied.

1. The uncertainty evaluation becomes a part of the data processing in the HyperCP.
2. A class-based approach of uncertainty evaluation was developed for users who do not have full instrument characterisation. This is called the default branch, where instrument effects are not corrected for.
3. The FRM branch is being created for the users with full instruments characterisation data, where several instrument related effects are corrected.
4. Updated uncertainty evaluation for the sea surface reflectance factor (ρ) including a comparison of different models.

In this project the work done in FRM4SOC-1 is being expanded upon, with the aim to share uncertainty analysis with the OC community, to aid in the generation of validation data that is FRM compliant. A 'community processor' with a full uncertainty budget output is being created alongside a guide to the correct application of uncertainties in OC measurement.

The remaining sections of this report describe at first the methodology, followed by the processing chain for above water radiometry. Next, the steps in the evaluation of uncertainty for default branch addressed, with the last sections containing the results of a few case studies and the summary. At present, the default branch of the HyperCP with integrated uncertainty analysis is complete. The user who downloads the HyperCP and processes their data correctly will automatically obtain uncertainty estimates for each *in situ* measurement. An important caveat to recognise is that the effect of some errors might still not be addressed by this version of the HyperCP. Users may still need to consider additional effects that could influence the measurements. For instance, although the processor includes several optional autonomous filtering stages, it is not possible to account for site or cruise specific conditions within a general approach. For example, structural shading is not addressed.

The *in situ* data used to prepare this report were collected during FICE1 and FICE2. FICE1 data were obtained from a HyperOCR system from Plymouth Marine Laboratory (PML) provided by Gavin Tilstone. FICE2 included TriOS systems from the University of Tartu (UT), Martin Ligi and PML providing for HyperOCR. All instruments characterisation files used were provided by Ilmar Ansko and Viktor Vabson from UT. HyperInSpace was made available in agreement with Dirk Aurin from NASA with new TRIOS processing added by Alexis Deru from Acri ST.

4 Methodology

The Guide to the expression of Uncertainty in Measurement (GUM) (JCGM100:2008, 2008) outlines a framework of general rules for expressing and evaluating uncertainty. Uncertainty is defined in International vocabulary of metrology guide (VIM) as: “a non-negative parameter characterising the dispersion of the quantity values being attributed to a measurand, based on the information used”. (JCGM200:2012, 2012).

4.1 Law of Propagation of Uncertainties

Measurement in science is rarely as simple as one direct measurement. In cases where the measurand is more complex the GUM outlines the use of a measurement function, describing the mathematical relationship between the known input quantities and the measurement itself. Generally, this takes the form:

$$y = f(x_1, \dots, x_N), \qquad \text{Eq. 1}$$

where y is the measurand, x_i are the input quantities and i = {1, 2, ..., N}.

Uncertainty analysis is performed by considering each input quantity to the measurement function in turn, this is represented in Figure 1. Each input quantity may be influenced by one or more error effects described by an uncertainty distribution. The Law of Propagation of Uncertainties (LPU) (JCGM100:2008, 2008) is used to combine these separate uncertainty distributions into the combined uncertainty of the measurand. The LPU is stated by the equation:

$$u^2(y) = cS(x)c^T, \qquad \text{Eq. 2}$$

where c is a vector composed of the sensitivity coefficients $\left(\frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_N}\right)$, $S(x)$ is the uncertainty matrix (also called covariance matrix or variance-covariance matrix) of dimension $N \times N$ containing on its diagonal the squares of the standard uncertainties associated with estimates of the components of the N - dimensional vector quantity, and in its off-diagonal positions the covariances associated with pairs of input quantities i.e.

$$\boldsymbol{S}(x) = \begin{bmatrix} u^2(x_1) & u(x_1, x_2) & \cdots \\ u(x_2, x_1) & u^2(x_2) & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \qquad \text{Eq. 3}$$

and u^2 (y) represents the combined variance, where u(y) is the combined standard uncertainty, which is the positive square root of the combined variance.

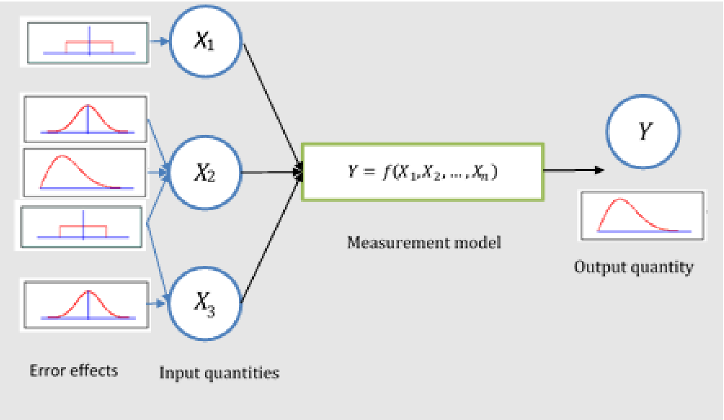


Figure 1. Representation of the process for uncertainty propagation

4.2 Monte Carlo method

The LPU (sometimes called ‘an analytical approach’) is a standard tool for propagating uncertainties, however, its complexity can become a drawback of the approach. The analytical method can become difficult to apply on complex functions with many correlated input parameters where the calculation of sensitivity coefficients is not straightforward. A common approach, the so-called sum of squares, used to combine uncertainties is derived from the LPU. However, this is only valid for specific measurement functions, particularly ones that include only either multiplicative or additive operations; for combinations of these mathematical operations the simple sum of squares does not provide correct uncertainty values. A way of circumventing these drawbacks is by using the Monte Carlo method (MC) instead of LPU, which uses a stochastic process to propagate uncertainty numerically.

The methods samples probability distribution functions (PDFs) generated for each input mean and standard uncertainty. These PDFs are generated using knowledge from the inputs themselves, their expected distribution, and their error-correlation, between input errors and pixels/bands. A joint PDF is created where each draw has the appropriate error-correlation for each dimension along each of the input quantities and the correct error-correlation between the different input quantities. This allows the output uncertainties to encompass as much of the physical reality of the measurements as possible. The MC will then apply these generated distributions as inputs to the measurement function. For a given number of draws M, a sample r, is evaluated for all input uncertainties N, such that $y_r = f(x_r), r = \{1, \ldots, N\}$, where f(x) represents the measurement function and x_r contains $x_{1,r}, \ldots, x_{N,r}$ (JCGM101:2008, 2008). The resulting output distribution allows an output uncertainty to be derived using well defined statistical methods.

To run a numerical method a software is needed, we propose to use Propagating Uncertainties with Python (punpy) package developed at NPL as part of the CoMet Toolkit (https://www.comet-toolkit.org/) which includes methods for propagating uncertainty using MC. The user needs to define measurement functions and uncertainty associated with each input. The tool allows for the propagation of uncertainty in a variety of applications. According to the GUM, the number of draws M expected to deliver a 95% coverage interval is of order 10^6 (JCGM 101:2008, 2008). This represents computational complexity which scales with the complexity of the method, as punpy will run the method for each draw. To avoid repeating a computationally intensive operation O (10^6) punpy includes a functionality which linearises the PDFs into vector and matrix notation to avoid such repetition. Another advantage to MC is that it does not require the uncertainty to be propagated in stages. The number of inputs or their relationship (additive/multiplicative) does not affect the outcome as sensitivity coefficients do not need to be found. This reduces the complexity of uncertainty propagation meaning that there are fewer points of contact between the code - which generates the uncertainty budget - and the processor it is embedded in. The running of punpy is discussed in more detail in a section 5.1.

4.3 Applying GUM Measurement Function Based Analysis

Section 4.1 introduced the measurement function, which describes how the measurand is determined from its input quantities. To illustrate the different error contributions in the measurement equations, a schematic which represents the sensor measurement function, called an Uncertainty Tree Diagram is utilised. The Uncertainty Tree Diagrams which were used in phase-1 of FRM4SOC project are retained for reference and presented in the Appendix. The updated uncertainty tree diagrams for above-water radiometric measurements are now created separately for the default and FRM branch, as can be found in the relevant sections of this document. Here we present the uncertainty tree diagram for the remote sensing reflectance shown in Figure 2, for a sake of readability the wavelength dependence is not included in the equations and not all uncertainty components are traced back to their origins, the details about individual radiometric inputs consisting radiance and irradiance measurands are presented in details in Figure 6 and Figure 7. These may serve both as examples of how measurement function-based analysis can be done but also as detailed schematics of how uncertainty is combined through the process of calculating OC products. This uncertainty can be traced back through to its impact on the measurand by the sensitivity coefficients of each branch. Finally, the effects which cause each respective input error are connected to the end of each branch.

Note that we should also consider the extent to which the measurement function describes the true physical state of the instrument – this is accounted for by including the term +0 at the end of the measurement function. This explicitly represents effects, expected to have zero mean, that are not captured by the measurement function (i.e. there is an uncertainty associated with this quantity being zero).

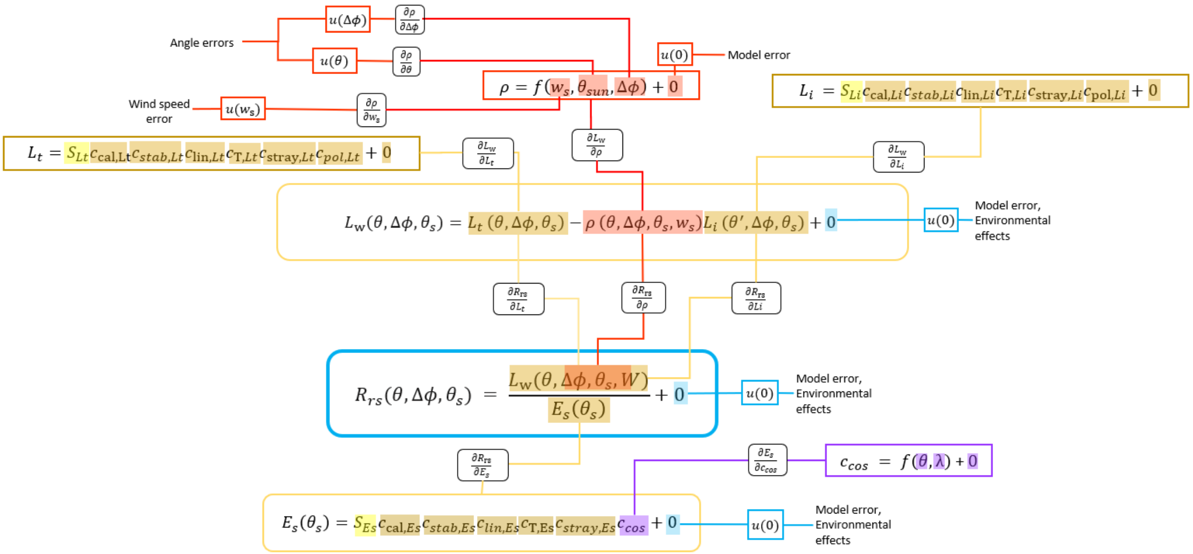

Angle errors
$u(\Delta\phi)$
$\frac{\partial\rho}{\partial\Delta\phi}$
$u(\theta)$
$\frac{\partial\rho}{\partial\theta}$
$u(0)$
Model error
$\rho = f(w_s, \theta_{sun}, \Delta\phi) + 0$
$L_i = S_{Li} c_{\mathrm{cal},Li} c_{stab,Li} c_{\mathrm{lin},Li} c_{\mathrm{T},Li} c_{\mathrm{stray},Li} c_{\mathrm{pol},Li} + 0$
Wind speed error
$u(w_s)$
$\frac{\partial\rho}{\partial w_s}$
$L_t = S_{Lt} c_{\mathrm{cal,Lt}} c_{stab,Lt} c_{\mathrm{lin},Lt} c_{\mathrm{T},Lt} c_{\mathrm{stray},Lt} c_{pol,Lt} + 0$
$\frac{\partial L_w}{\partial L_t}$
$\frac{\partial L_w}{\partial\rho}$
$\frac{\partial L_w}{\partial L_i}$
$L_{\mathrm{w}}(\theta, \Delta\phi, \theta_s) = L_t\,(\theta, \Delta\phi, \theta_s) - \rho\,(\theta, \Delta\phi, \theta_s, w_s) L_i\,(\theta', \Delta\phi, \theta_s) + 0$
$u(0)$
Model error, Environmental effects
$\frac{\partial R_{rs}}{\partial L_t}$
$\frac{\partial R_{rs}}{\partial\rho}$
$\frac{\partial R_{rs}}{\partial Li}$
$R_{rs}(\theta, \Delta\phi, \theta_s) = \frac{L_{\mathrm{w}}(\theta, \Delta\phi, \theta_s, W)}{E_s(\theta_s)} + 0$
$u(0)$
Model error, Environmental effects
$\frac{\partial R_{rs}}{\partial E_s}$
$\frac{\partial E_s}{\partial c_{cos}}$
$c_{cos} = f(\theta, \lambda) + 0$
$E_s(\theta_s) = S_{Es} c_{\mathrm{cal},Es} c_{stab,Es} c_{lin,Es} c_{\mathrm{T,Es}} c_{stray,Es} c_{cos} + 0$
$u(0)$
Model error, Environmental effects

Figure 2 – Combined Uncertainty tree showing Remote Sensing Reflectance (R_{rs}).

4.3.1 Effects table reporting

With the use of the uncertainty tree diagram each of the effects associated with the measurement function can be identified by tracing the branches to their end. The "Effects Table" is a method of reporting and quantifying these effects in a standard way agreed by the FIDUCEO project. The FIDUCEO project laid out a framework for applying GUM-based methods to satellite sensor fundamental climate data records (FCDRs) to provide per pixel uncertainties (Mittaz *et al.*, 2019).

An Effects Table documents:

- The uncertainty associated with the given input quantity ('effect' according to 'FIDUCEO Vocabulary V2', 2018).
- The sensitivity coefficients required to propagate uncertainties associated with that effect to uncertainty associated with the measurand.
- The correlation structure over spatial, temporal, and spectral scales for errors from this effect.

Table 2. FIDUCEO "Effects Table" with descriptions of how each part is populated.

Table descriptor		**How this is codified**
Name of effect		A unique name for each source of uncertainty in a term of the measurement function
Affected term in measurement function		Name and standard symbol of affected term
Instruments in the series affected		Identifier of the specific instrument where this effect matters
Correlation type and form		Specific instance of the three general categories described in Section 4.4. Described in more detail Woolliams et al. 2017.
Correlation scale		In units of pixels/time – what is the scale of the correlation shape?
Channels/bands		Channel names in standard form
		Cross-channel correlation matrix
Uncertainty	PDF shape	Functional form of estimated error distribution for the term
	units	Units in which PDF shape is expressed (units of term, or can be as percentage)
	magnitude	Value(s) or parameterisation estimating width of PDF
Sensitivity coefficient		Value, equation or parameterisation of sensitivity of measurand to term

In certain cases, the errors which affect the input parameters of the measurement function cannot be assumed mutually independent. In this case the correlation between the various errors must be considered. The Horizon 2020 FIDUCEO (Fidelity and uncertainty in climate data records from Earth observations) project, in their vocabulary defined three broad categories of error correlation effects ('FIDUCEO Vocabulary V2', 2018).

- Random effects – "those causing errors that cannot be corrected for in a single measured value, even in principle, because the effect is stochastic. Random effects for a particular measurement process vary unpredictably from (one set of) measurement(s) to (another set of) measurement(s). These produce random errors which are entirely uncorrelated between measurements (or sets of measurements) and generally are reduced by averaging."
- Structured random effects – "means those that across many observations there is a deterministic pattern of errors whose amplitude is stochastically drawn from an underlying probability distribution; 'structured random' therefore implies 'unpredictable' and 'correlated across measurements'..."
- Systematic (or common) effects – "those for a particular measurement process that do not vary (or vary coherently) from (one set of) measurement(s) to (another set of) measurement(s) and therefore produce systematic errors that cannot be reduced by averaging.

4.4 Practical Implementation of LPU/MC

A Scheme based on MC is proposed to evaluate uncertainty in the frame of FRM4SOC-2 as it was done in FRM4SOC phase-1. The LPU based approach, under the condition that it is correctly applied, is methodologically valid, and the user may choose to combine measurement uncertainty in this way.

To evaluate the measurement uncertainty, regardless of the chosen method, several steps need to be taken:

1. A measurement function must be defined,
2. All sources of uncertainty must be identified.
3. Then according to the Law of Propagation of Uncertainty (see Eq. 2), the sensitivity coefficients must be calculated. Nominally, sensitivity coefficients are a partial derivative of a measurement function and a given contributor. For complex functions, it might be impossible to calculate these coefficients analytically. This step is not needed for Monte Carlo approach.
4. Using the analytical method, all inputs with non-Gaussian PDFs must be converted to that shape. To do this, divisors are specified for typical shapes. For example, to convert a rectangular PDF to its equivalent in Normal distribution, the uncertainty value is divided by $\sqrt{3}$. This step is not needed when the MC is used, as the original PDF can be propagated through the model.
5. If any of the input quantities are correlated along any of their dimensions or with each other, then an error covariance matrix is necessary (see Eq. 3)
6. To combine uncertainties, Eq. 2 is used for the analytical approach. In the MC approach, one first needs to generate joint PDFs of the input quantities, taking into account their error covariance matrix. The measurement function is then run many times, each time using randomly selected inputs from joint PDFs of the input quantities.
7. The results of the analytical approach by default have a Gaussian distribution and when quoted as the output of Eq. 2 are called the standard uncertainty, which means it has a coverage factor k = 1, or 1σ defined as one standard deviation from the mean assuming a normal distribution function. This information expresses the confidence level at around 67% that the estimated value is within its quoted uncertainty. The result in the MC is a PDF of the output quantity and depending on its shape the uncertainty is defined as the standard deviation of this distribution (Gaussian PDF) or by the upper and lower limits that define 67% coverage of that distribution.
8. For some applications higher confidence is necessary, and then the standard uncertainty is expanded to k = 2 for 95% coverage or k = 3 for 99% coverage. The expanded uncertainty is calculated by multiplication of a standard uncertainty value by the coverage factor, k. This is called the expanded uncertainty. The expanded uncertainties are often used by calibration laboratories to quate absolute calibration uncertainty of radiometric standard. This is important to convert k=2 uncertainty into k=1 level (simply by dividing tit by the coverage factor value, 2 in this case) before they are combined with other uncertainty sources.

5 Above water radiometry processing chain

The HyperInSPACE processor, developed by NASA for above water ocean colour radiometry for HyperOCR instrument class was chosen to be the basis for the new HyperInSPACE Community Processor (HyperCP). This endeavour proved to be collaborative, with the HyperInSPACE processor adapting to accommodate changes based on the experience of FRM4SOC project partners, introducing compatibility with TriOS and HyperOCR instruments as well as the uncertainty budget calculation detailed above in the Methodology section. The processing chain developed for the calculation of uncertainty within HyperCP is detailed here in the order in which it occurs within the processing chain. More information on the general function of the HyperCP can be found at (url: https://github.com/nasa/HyperInSPACE).

Above water OC surface measurements utilise three instruments: A cosine plane irradiance sensor which measures downwelling irradiance (E_s). Two radiance sensors which measure sky radiance (L_i)and total water radiance (L_t)respectively.

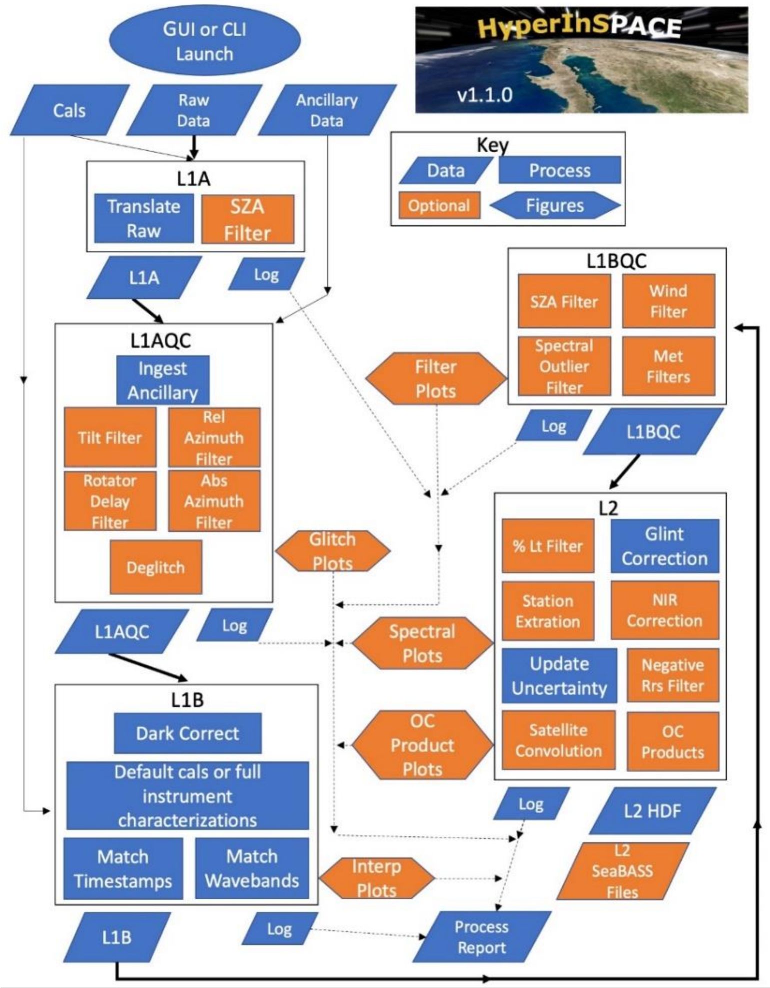


Figure 3 - Flow diagram of HyperInSPACE processing steps and functioning v1.1.0, used as the basis of the HyperCP.

When ingesting raw data, for each instrument, the data is in the form of counts per spectral pixels with some pixels occluded to give dark readings, depending on instrument class as discussed in-part in section 5.2.1.1. Currently, for the processing HyperOCR instrument class, HyperInSPACE will expect at least 7 calibration files; two per instrument, for light and dark readings, and one to read GPS and location data. Knowledge of these files is provided by the user via a Graphical User Interface of GUI which builds a configuration file. The configuration file also contains many other options for all levels of processing. The HyperInSPACE processor provides noise reducing 'deglitching' as well as threshold-based filters to the data to ensure basic data quality for further processing steps. The L1B stage (see Figure 3) applies the dark correction, where the dark counts are subtracted from the non-occluded instrument counts. After dark substitution, absolute radiometric calibration coefficients are applied. Uncertainties are provided by the labs which carry out this work as part of data quality assurances and are ingested via separate calibration files. Applying absolute calibration converts Digital Numbers (DN) or 'counts' to radiance/irradiance units depending on the instrument.

Other contributing factors: temperature dependence, spectral straylight, and deviations from the assumption that the radiance response of the detector is linear. The downwelling irradiance sensor also requires correction to any imperfections in the cosine response, while the radiance sensors for polarisation effects. All these factors are included in uncertainty evaluation. The correction coefficients, which are ascertained experimentally for each instrument used, are applied sequentially to the radiance/irradiance data where necessary, in the FRM branch only. Each correction includes a corresponding uncertainty which must be totalled in the budget and propagated to R_{rs} uncertainty. Since MC is used, as described in 4.2, it is simply enough to budget the contributing uncertainties at this point, saving the means and PDFs for application further into the process. Data files for each correction are ingested into the processor at this stage with their associated, lab derived, uncertainties. The resulting information is saved in a hierarchical data format (HDF) with the associated instrument radiance/irradiance data for further processing.

Level 2 processing occurs last in the chain, after a small step where the data is interpolated into common hyperspectral wavebands. In the L2 step the sea surface reflectance factor, ρ, is calculated and the products are evaluated using the interpolated, corrected, and processed data. This is, however, only sufficient to describe the generation of data products; uncertainties must be evaluated and stored throughout the processing chain.

The uncertainty evaluation will be applied within the HyperCP at two different levels, denoted by the branches of the schematic in Figure 4. The default processing branch is the one now used in HyperInSPACE for HyperOCR class instruments, being expanded for TriOS instruments by the current project. This uses standard processing and manufacturer type calibration files, with an additional absolute radiometric calibration uncertainty requirement. This is, currently, the most common way to process in situ data, where calibration files are provided without additional instrument characterisation information. To evaluate the measurement uncertainty in this branch, class-based uncertainty estimates for all instrument related uncertainty components are defined based on the laboratory characterisation tests of several radiometers from each class described in [AD-1] the input file format for these additional instrument characterisation data is described in [AD-3]. None of the instrument characteristics, like detector nonlinearity, will be corrected in the default processing branch, resulting in higher uncertainties when utilising this branch of the processing chain.

For a case when the absolute radiometric uncertainty is not available, the default branch processing can still be applied, and the class specific absolute radiometric uncertainty will be assigned within the HyperCP, however the results on that processing will be marked with non FRM compliant flag to highlight the requirement for the absolute radiometric uncertainty input.

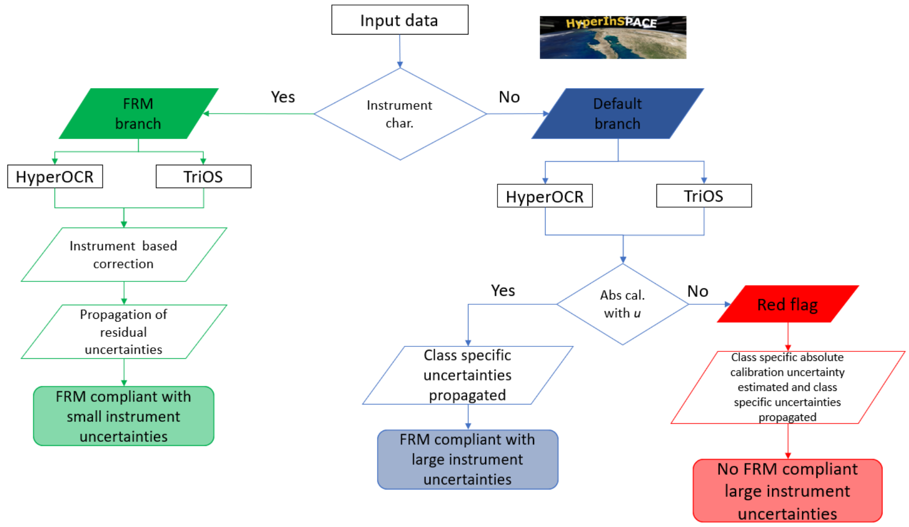

Input data
HyperInSPACE
Instrument char.
Yes
No
FRM branch
Default branch
HyperOCR
TriOS
HyperOCR
TriOS
Instrument based correction
Propagation of residual uncertainties
FRM compliant with small instrument uncertainties
Abs cal. with u
Yes
No
Red flag
Class specific uncertainties propagated
FRM compliant with large instrument uncertainties
Class specific absolute calibration uncertainty estimated and class specific uncertainties propagated
No FRM compliant large instrument uncertainties

Figure 4. Schematic of uncertainty processing in HyperCP.

The second “FRM” branch will contain an adapted processing chain for integrating instrument characteristics and corrections into the processing. The default calibration files will not be applicable here as a new radiometric calibration file format was defined by Tartu University, where the information about the radiometric standards and radiometric calibration uncertainty is included in the file. In addition, the new file format contains absolute radiometric calibration performed at two different integration times allowing the derivation of detector nonlinearity correction. This branch will apply all available instrument characteristics into the data processing chain including corrections for various instrument effects like nonlinearity of the detector, spectral stray light, temperature dependence and cosine response for the irradiance sensor.

Note here that the difference in processing between the branches is related only to instrument related components. All other processing steps such as sky glint and Bidirectional Reflectance Distribution Function (BRDF) correction can be done in the same manner for both branches.

5.1 Practical guide to Punpy

Propagating Uncertainty with Python or ‘punpy’ is a python tool designed to propagate uncertainties numerically, which is part of the CoMet Toolkit (https://comet-toolkit.org/). Punpy includes two methods for propagating uncertainty, the LPU and MC method, described in more detail in section 4.2. For further details on these methods, we refer to the punpy ATBD (https://punpy.readthedocs.io/en/latest/content/atbd.html). The MC approach is used in the processor, and we here summarize how the required inputs are passed to punpy.

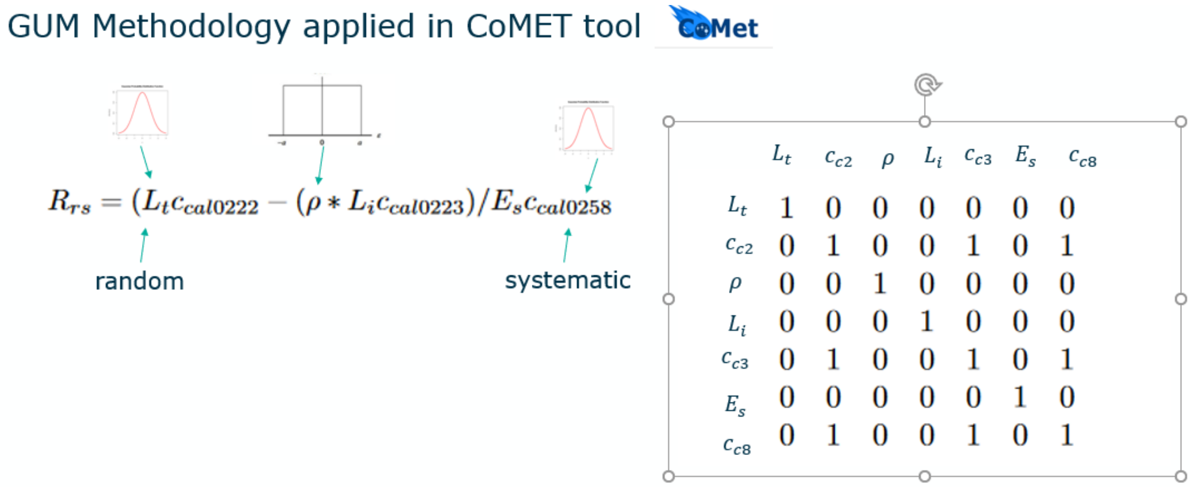
The MC method generates distributions for the input quantities based on the mean input values with their associated uncertainties and error-correlation. Therefore, the required inputs to punpy are a list of mean values, input uncertainties, error-correlations, and a measurement function. The measurement function must be defined in python, and must take as arguments the input quantities provided as single numbers or numpy arrays, and must return the measurand as a single number or numpy array. A pointer this function can then be given to punpy as an input argument when propagating the uncertainties. Stated explicitly, function pointers provide an ‘address’ so that the measurement function can be run multiple times by punpy using distributions generated from the mean values and uncertainties (in python, this is very easy as the function name itself is simply passed to punpy).

These generated distributions must further account for the difference between random and systematic errors. Punpy has two solutions for this issue. Firstly, the MC approach for punpy can be called using two functions, propagate_random and propagate_systematic. Each account for all input uncertainties being random and systematic respectively. However, should the input components consist of a mix of random and systematic components, then another argument ‘corr_x’ can be defined in order to provide the error-correlation for each component. corr_x can also be used to provide other (i.e. not random or systematic) error-correlation forms (provided as an error-correlation matrix), but this is not used within the processor. Within the processor, corr_x is a list of strings, of the same length as the other input arguments, that denotes ‘rand’ for random and ‘syst’ for systematic for each input respectively. With this information punpy can build distributions that account for the nature of the inputs, ensuring an accurate final uncertainty. The corr_x keyword accounts for the error-correlation along the wavelength-dimension for each input quantity, but not for the error correlation between the various input quantities. Punpy allows to provide this error correlation between the input quantities using the 'corr_between' keyword using an error-correlation matrix.

In addition to considering the error-correlation in the input data, we also need to calculate the error-correlation in the output data. The error-correlation matrix for the different radiometers, L_t, L_i, E_s is a finding from this report and as such is hard coded into the processor. However, proceeding steps utilise the calculated radiometric uncertainties. Punpy includes an ability to output the correlation between outputs, provided multiple outputs are processed simultaneously (which is the case in the processor). This output correlation matrix is then used to provide the input error-correlation information for the propagating of $R_{rs}$and L_w uncertainties.

When using punpy as a standalone tool (which is the case within the processor), all uncertainties inputted need to be absolute and not relative. It also outputs absolute uncertainties, meaning the exact amount the measurement may deviate from the measured value and not the percentage. For clarity it may be necessary to convert these uncertainties into relative values by dividing them by the mean value for their given input.

Figure 5 shows a graphical representation of a simplified R_{rs} measurements function, where each of the inputs will have defined a mean value and an associated PDF, which can be Gaussian or rectangular for example. Each of the input components in the equation has a defined corr_x as well where some of them will be random and some are systematic. In this example the inputs L_t, L_i, E_s, represent each instrument dark corrected averaged readings, thus the uncertainty associated with each term has normal distribution and is random. The following inputs c_{calXXX} represent the calibration coefficients for each of the instruments, they too have a PDF with Normal distribution, but the calibration errors are systematic, thus the same value will be present in all measurements, corr_x for them is defined as systematic. In the following example in ρ has assigned PDF shape that is rectangular. The matrix on the right side in Figure 5 shows correlation_between term, where we can define which input into the measurement equation are correlated. Whenever the value of 1 is present off the diagonal elements of that matrix this indicate that both inputs, one defined as column name and the second as the row names are correlated. This is the case for calibration coefficients assuming the instrument triplet was calibrated in one laboratory using the same irradiance standards and the same reflectance standards for radiance case (see the c_{cX} elements in the matrix). We simplified here the correlation between radiance and irradiance calibration and assume it is one, but this is more the case for both radiance calibrations. The FRM branch processing will evaluate the exact correlation value between radiance and irradiance calibration coefficients, whereas for the default branch an approximation of 1 is used.


GUM Methodology applied in CoMET tool

$$R_{rs} = (L_t c_{cal0222} - (\rho * L_i c_{cal0223})/E_s c_{cal0258}$$

random

systematic

$$\begin{array}{c|ccccccc} & L_t & c_{c2} & \rho & L_i & c_{c3} & E_s & c_{c8} \\ \hline L_t & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_{c2} & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ \rho & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ L_i & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ c_{c3} & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ E_s & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ c_{c8} & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Figure 5. Graphical representation of simplified R_{rs} measurement function and punpy inputs.

The real measurement functions processed in HyperCP have more components, but the same principle applies, each of them will have defined a mean value and uncertainty estimate, the nature of the error (random or systematic) and correlation with other elements of equation. Any method in the code related to propagating uncertainty can be found in “Source/Uncertainty_Analysis.py”. The uncertainties are generated throughout the HyperCP, however code relevant to uncertainty analysis are contained here wherever possible.

5.2 Level 1B processing

This is the level where uncertainty evaluation starts to be executed in HyperCP. In L1B each instrument is treated individually in its nominal wavelength scale, the uncertainties are provided for radiometric values of L_t, L_i, E_s per pixel. The correlation between is not needed as this stage as the uncertainties are calculated separately for each radiometric quantity, thus for example we have only one calibration coefficient in the measurement equation to calculate downwelling irradiance.

In the next subsection we present the default branch uncertainty evaluation. This is followed by the subsection 5.2.2 FRM branch processing , where in this version of the report is a placeholder only, and it will be completed for v3 and describe all processing steps in the FRM branch including the evaluation of correction coefficients for detector non linearity, spectral stay light , temperature sensitivity and cosine response.

5.2.1 Default branch

The processing steps in the default branch stay as they are already implemented in HyperInSPACE, with the HyperCPs new additions that brings the TriOS instrument class processing. The uncertainties in the default branch are calculated using the punpy package incorporated into HyperInSPACE and called in several places to evaluate uncertainty at various processing levels. This enables a user to see the evolution in uncertainty contributors, for example one might want to see what the uncertainty in downwelling irradiance measurement is, not only final products like remote sensing reflectance. Figure 6 and Figure 7 present the uncertainty tree diagrams for the default branch processing.

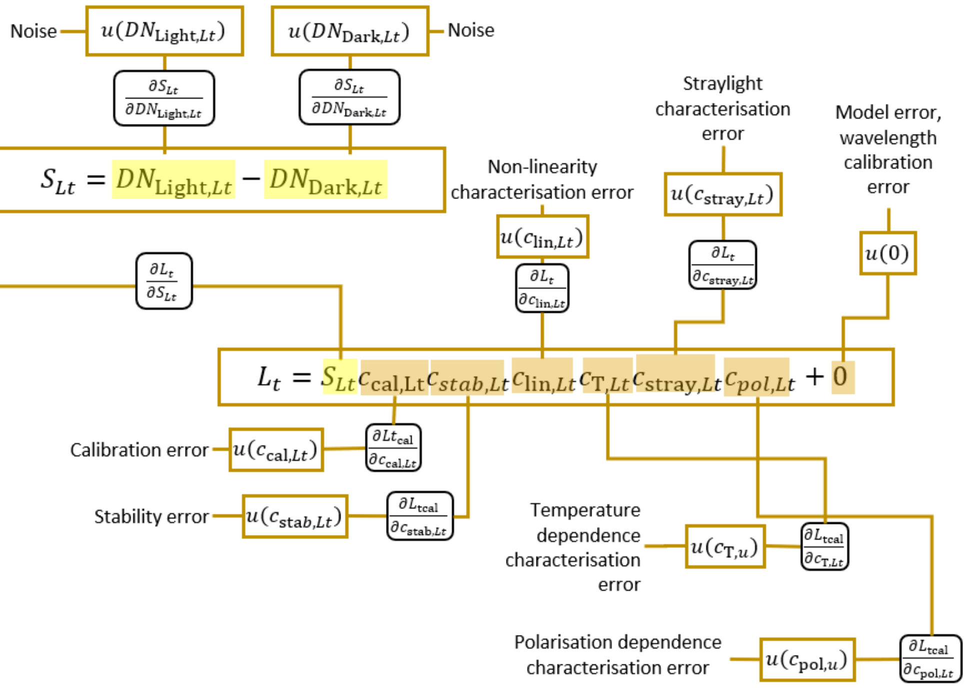

Noise
$u(DN_{\text{Light},Lt})$
$u(DN_{\text{Dark},Lt})$
Noise
$\frac{\partial S_{Lt}}{\partial DN_{\text{Light},Lt}}$
$\frac{\partial S_{Lt}}{\partial DN_{\text{Dark},Lt}}$
$S_{Lt} = DN_{\text{Light},Lt} - DN_{\text{Dark},Lt}$
Straylight characterisation error
Model error, wavelength calibration error
Non-linearity characterisation error
$u(c_{\text{stray},Lt})$
$u(c_{\text{lin},Lt})$
$u(0)$
$\frac{\partial L_t}{\partial S_{Lt}}$
$\frac{\partial L_t}{\partial c_{\text{lin},Lt}}$
$\frac{\partial L_t}{\partial c_{\text{stray},Lt}}$
$L_t = S_{Lt} c_{\text{cal},\text{Lt}} c_{stab,Lt} c_{\text{lin},Lt} c_{\text{T},Lt} c_{stray,Lt} c_{pol,Lt} + 0$
Calibration error
$u(c_{\text{cal},Lt})$
$\frac{\partial Lt_{\text{cal}}}{\partial c_{\text{cal},Lt}}$
Stability error
$u(c_{\text{stab},Lt})$
$\frac{\partial L_{\text{tcal}}}{\partial c_{\text{stab},Lt}}$
Temperature dependence characterisation error
$u(c_{\text{T},u})$
$\frac{\partial L_{\text{tcal}}}{\partial c_{\text{T},Lt}}$
Polarisation dependence characterisation error
$u(c_{\text{pol},u})$
$\frac{\partial L_{\text{tcal}}}{\partial c_{\text{pol},Lt}}$

Figure 6. Uncertainty tree diagram for total water radiance (L_t), the tree for sky radiance (L_i) would be the same with only one difference in subscripts change from $_t$ to $_i$, as it is presented in water leaving (L_w) uncertainty tree diagram in Figure 10.

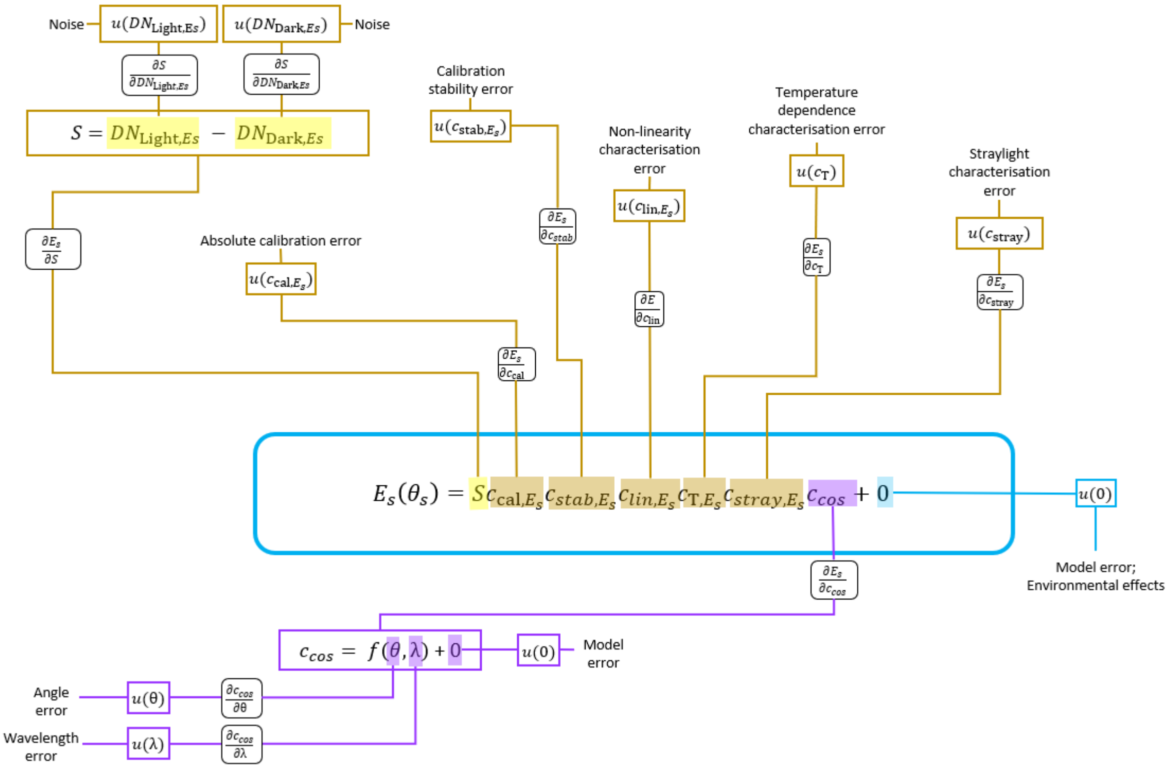

Noise
$u(DN_{\text{Light},Es})$
$u(DN_{\text{Dark},Es})$
Noise
$\frac{\partial S}{\partial DN_{\text{Light},Es}}$
$\frac{\partial S}{\partial DN_{\text{Dark},Es}}$
$S = DN_{\text{Light},Es} - DN_{\text{Dark},Es}$
Calibration stability error
$u(c_{\text{stab},E_s})$
Non-linearity characterisation error
Temperature dependence characterisation error
Straylight characterisation error
$u(c_{\text{T}})$
$u(c_{\text{lin},E_s})$
$u(c_{\text{stray}})$
$\frac{\partial E_s}{\partial S}$
Absolute calibration error
$u(c_{\text{cal},E_s})$
$\frac{\partial E_s}{\partial c_{stab}}$
$\frac{\partial E_s}{\partial c_{\text{T}}}$
$\frac{\partial E}{\partial c_{\text{lin}}}$
$\frac{\partial E_s}{\partial c_{\text{stray}}}$
$\frac{\partial E_s}{\partial c_{\text{cal}}}$
$E_s(\theta_s) = S c_{\text{cal},E_s} c_{stab,E_s} c_{lin,E_s} c_{\text{T},E_s} c_{stray,E_s} c_{cos} + 0$
$u(0)$
Model error; Environmental effects
$\frac{\partial E_s}{\partial c_{cos}}$
$c_{cos} = f(\theta, \lambda) + 0$
$u(0)$
Model error
Angle error
$u(\theta)$
$\frac{\partial c_{cos}}{\partial \theta}$
Wavelength error
$u(\lambda)$
$\frac{\partial c_{cos}}{\partial \lambda}$

Figure 7. Uncertainty tree diagram for downwelling irradiance (E_s).

All instrument related uncertainty are combined and propagated according to Eq. 4 for both radiance instruments (L_t and L_i) and Eq. 5 irradiance, executed using the punpy tool.

$$L_x(\lambda) = \left(DN_{\mathrm{light},L_x} - DN_{dark,L_x}\right) \cdot c_{\mathrm{cal},L_x}(\lambda) c_{\mathrm{stab},L_x}(\lambda) c_{\mathrm{lin},L_x}(\lambda) c_{\mathrm{stray},L_x}(\lambda) c_{T,L_x}(\lambda) c_{\mathrm{pol},L_x}(\lambda), \qquad \text{Eq. 4}$$

where $L_x(\lambda)$ is the radiance measured by one of the radiometers (the equations are the same for L_t and L_i), $DN_{\mathrm{light},L_x} - DN_{dark,L_x}$ is a mean of Digital Numbers for the ensemble of recordings composing a given "station", c_{cal,L_x} in the calibration coefficient for each instrument, $c_{\mathrm{stab},L_x}(\lambda)$ is the individual instrument calibration stability during the deployment, $c_{\mathrm{lin},L_x}(\lambda)$ is the detector non-linearity term, $c_{\mathrm{stray},L_x}(\lambda)$ is the spectral straylight term, $c_{T,L_x}(\lambda)$ is the temperature sensitivity and $c_{\mathrm{pol},L_x}(\lambda)$ is the polarisation sensitivity.

All components apart from radiometric calibration are set equal to one and hold information about various instrument related characteristics errors.

Similar to the radiance instrument, the equation 5 lists all components that shall be included for the irradiance:

$$E_s(\lambda) = \left(DN_{\mathrm{light},E_S} - DN_{dark,E_S}\right) \cdot c_{\mathrm{cal},E_S}(\lambda) c_{\mathrm{stab},E_S}(\lambda) c_{\mathrm{lin},E_S}(\lambda) c_{\mathrm{stray},E_S}(\lambda) c_{T,E_S}(\lambda) c_{\mathrm{cos},E_S}(\lambda), \qquad \text{Eq. 5}$$

where $E_s(\lambda)$ in the downwelling irradiance, and all the equation components are the same as already defined for Eq. 4 with a difference of subscripts, thus now they refer to irradiance instrument, and instead of polarisation sensitivity in irradiance case there is a term $c_{\mathrm{cos},E_S}(\lambda)$ that is cosine response of the diffuser.

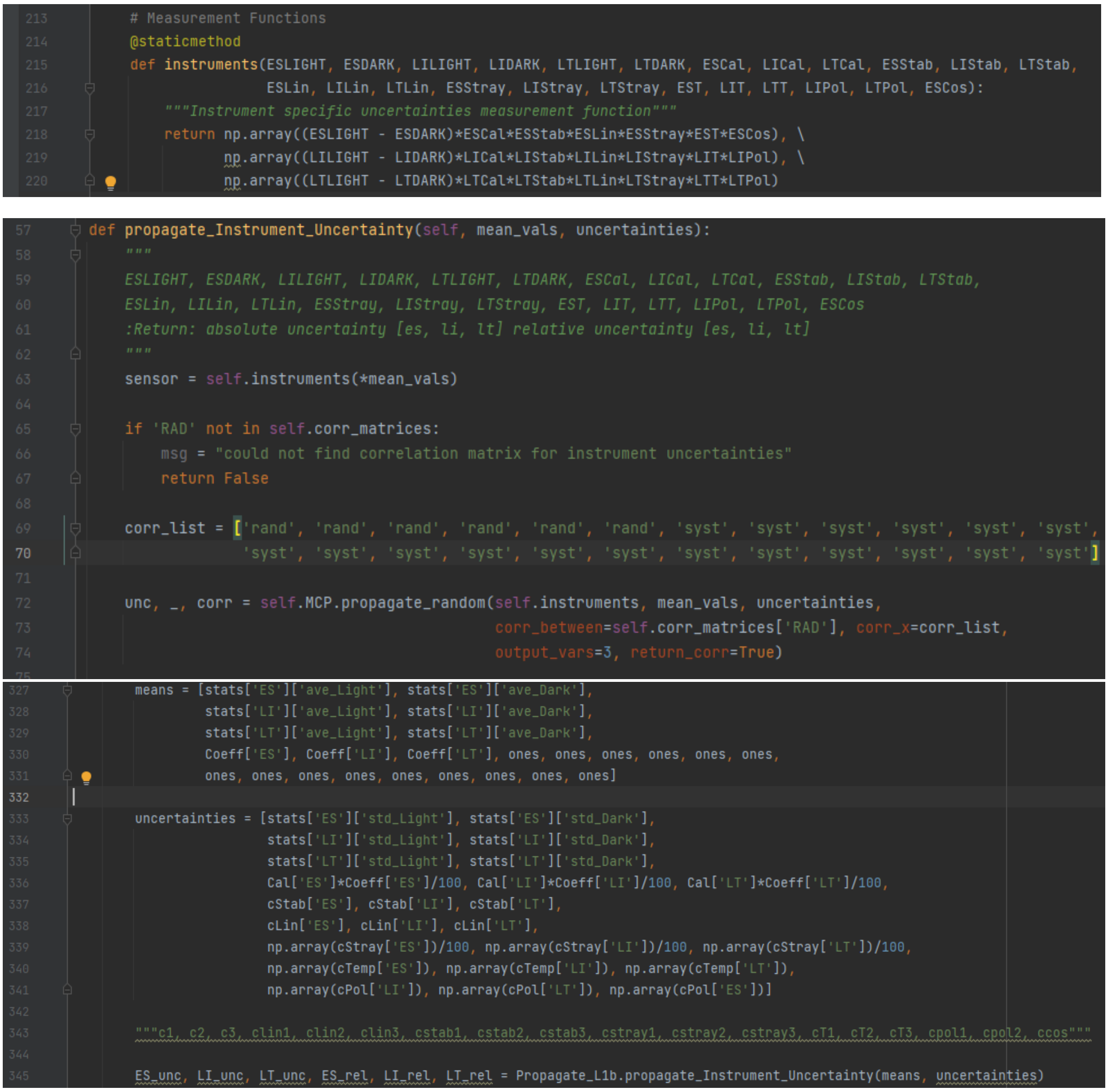
Figure 8 shows the source code used to define measurement function in HyperCP for each instrument and how a function to propagate uncertainty is defined. Where code lines 218-220 represent Eq. 4 and Eq. 5 for each instrument as measurement function in the `instruments' function (the @staticMethod is to indicate this is a static method as opposed to a class method). This is defined in "*Source/Uncertainty_Anylaysis.py*" as well as definition Propagate_Instument_Uncertainty, then the function itself is called in *"Source/ProcessL1b."* and "*Source/TriosL1b.py*" as Propagate_L1b.propagate_Instrument_Uncertainty(means, uncertainties), where the means and uncertainties of all inputs are defined. As is shown Figure 8, code lines 327-331[1] the value of each input is one apart from instrument noise reading and calibration coefficients. Lines 333-341 defined where to find uncertainty information. This is important to note that as all uncertainties inputs into punpy need to be absolute thus calibration uncertainty from the radcal.txt file needs to be changed from relative to absolute by multiplication by calibration coefficient and division by 100 (see code line 336 in Figure 8). Straylight information in the default branch now is kept as percentage values too, thus they need to be divided by 100 (code line 339) to make them absolute for the $c_{\mathrm{stray}} = 1$.

The information about the magnitude of instrument characteristics can be find in files imbedded into HyperCP, as presented in Figure 9. Each instruments characteristic has a dedicated input text file with the information used for uncertainty propagation. The files format is described in detail in [AD-3].

In Table 3 we recapture all inputs to the equations with information like those that in Effect Table (Table 2) but simplified for the needs of this approach. We induced as well, where possible, values of uncertainty associated with instrument related effect for each instrument class. When that uncertainty varies with wavelength and instrument type this was not possible to input that value to the table.

All instruments' characteristics are just read from the files stored in HyperCP and their content is use directly as input to uncertainty values except for temperature sensitivity error, that is calculated for each cast taking into consideration the actual temperature during the measurement.

[1] Please note that HyperCP is being continuously developed, thus the code line numbers might have changed if looked at in the future, nevertheless the function definitions will remain the same.



```
    # Measurement Functions
    @staticmethod
    def instruments(ESLIGHT, ESDARK, LILIGHT, LIDARK, LTLIGHT, LTDARK, ESCal, LICal, LTCal, ESStab, LIStab, LTStab,
                    ESLin, LILin, LTLin, ESStray, LIStray, LTStray, EST, LIT, LTT, LIPol, LTPol, ESCos):
        """Instrument specific uncertainties measurement function"""
        return np.array((ESLIGHT - ESDARK)*ESCal*ESStab*ESLin*ESStray*EST*ESCos), \
               np.array((LILIGHT - LIDARK)*LICal*LIStab*LILin*LIStray*LIT*LIPol), \
               np.array((LTLIGHT - LTDARK)*LTCal*LTStab*LTLin*LTStray*LTT*LTPol)
```

```
def propagate_Instrument_Uncertainty(self, mean_vals, uncertainties):
    """
    ESLIGHT, ESDARK, LILIGHT, LIDARK, LTLIGHT, LTDARK, ESCal, LICal, LTCal, ESStab, LIStab, LTStab,
    ESLin, LILin, LTLin, ESStray, LIStray, LTStray, EST, LIT, LTT, LIPol, LTPol, ESCos
    :Return: absolute uncertainty [es, li, lt] relative uncertainty [es, li, lt]
    """
    sensor = self.instruments(*mean_vals)

    if 'RAD' not in self.corr_matrices:
        msg = "could not find correlation matrix for instrument uncertainties"
        return False

    corr_list = ['rand', 'rand', 'rand', 'rand', 'rand', 'rand', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst',
                 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst']

    unc, _, corr = self.MCP.propagate_random(self.instruments, mean_vals, uncertainties,
                                             corr_between=self.corr_matrices['RAD'], corr_x=corr_list,
                                             output_vars=3, return_corr=True)
```

```
means = [stats['ES']['ave_Light'], stats['ES']['ave_Dark'],
         stats['LI']['ave_Light'], stats['LI']['ave_Dark'],
         stats['LT']['ave_Light'], stats['LT']['ave_Dark'],
         Coeff['ES'], Coeff['LI'], Coeff['LT'], ones, ones, ones, ones, ones, ones,
         ones, ones, ones, ones, ones, ones, ones, ones, ones]

uncertainties = [stats['ES']['std_Light'], stats['ES']['std_Dark'],
                 stats['LI']['std_Light'], stats['LI']['std_Dark'],
                 stats['LT']['std_Light'], stats['LT']['std_Dark'],
                 Cal['ES']*Coeff['ES']/100, Cal['LI']*Coeff['LI']/100, Cal['LT']*Coeff['LT']/100,
                 cStab['ES'], cStab['LI'], cStab['LT'],
                 cLin['ES'], cLin['LI'], cLin['LT'],
                 np.array(cStray['ES'])/100, np.array(cStray['LI'])/100, np.array(cStray['LT'])/100,
                 np.array(cTemp['ES']), np.array(cTemp['LI']), np.array(cTemp['LT']),
                 np.array(cPol['LI']), np.array(cPol['LT']), np.array(cPol['ES'])]

"""c1, c2, c3, clin1, clin2, clin3, cstab1, cstab2, cstab3, cstray1, cstray2, cstray3, cT1, cT2, cT3, cpol1, cpol2, ccos"""

ES_unc, LI_unc, LT_unc, ES_rel, LI_rel, LT_rel = Propagate_L1b.propagate_Instrument_Uncertainty(means, uncertainties)
```

Figure 8 – Representation of measurement functions in L1b processing, as it exists in the HyperCP.

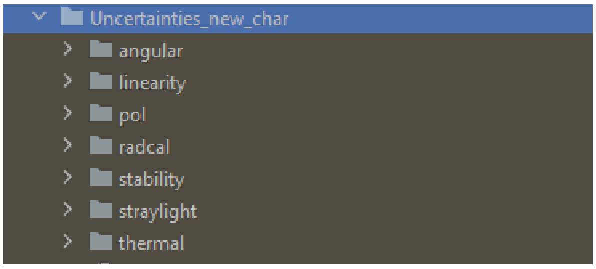


Figure 9 – Print screen from HPC showing that all the files with instrument characterisation information are stored in HPC.

5.2.1.1 Thermal sensitivity

The information required to estimate thermal sensitivity is provided within the characterisation files proposed in this study for a nominal temperature of 20°C. To correct for the thermal effects, we need to know the actual temperature of the instrument. The two instrument classes have different solutions for temperature readings:

i) HyperOCR instruments have a built-in thermistor, so information about the internal instrument temperature is always available and is used.

ii) The TriOS instrument class does not have a built-in temperature sensor, nor the internal shutter. The dark reading is estimated from several blacked pixels rather than by cutting the light throughput into the detector. Thus, the estimation of the actual detector temperature *in situ* is more challenging for this instrument class. Temperatures can be estimated from the ambient temperature, although it is not a very accurate method. As an alternative option for TriOS, it might be possible to include additional dark frame reading with set integration time during the *in situ* measurements to monitor the internal detector temperature. Currently in HyperCP we used the ambient temperature reading +5° as indication of TriOS internal temperature.

To correct for the temperature sensitivity of the detector the correction coefficient is calculated according to equation proposed by (Zibordi, Talone and Jankowski, 2017)

$$c_T(\lambda) = 1 + c(\lambda) \times \Delta T, \qquad \text{Eq. 6}$$

where $c_T(\lambda)$ is the temperature correction per pixel, $c(\lambda)$ in the instrument thermal sensitivity and ΔT is the difference in the temperature of the measurements and the nominal temperature of 20°C.

But as in default branch we do not correct for the thermal sensitivity effects we use this correction as an estimate of uncertainty due to temperature sensitivity.

Table 3. Summary information about each uncertainty component values for class-based approach (blue branch, Fig. 5)

Variable symbol	**Variable name/description**	**Exemplary uncertainty magnitude for class-based characterisation**		**PDF shape**	**Correlation** 'corr_x'	**Correlation between** 'corr_between'
		TRIOS	HyperOCR			
$(DN_{\mathrm{light},L_X} - DN_{dark,L_X})$ $(DN_{\mathrm{light},E_S} - DN_{dark,E_S})$	Mean value of DNs measured by a single instrument at a "station"	Standard deviation calculated per measurement from data statistics		Normal	Random	N/A
c_{cal}	Absolute radiometric calibration	Uncertainty values from calibration certificate divided by 2 to convert them back into standard uncertainty, k=1		Normal	Systematic	Between all three instruments
c_{stab}	Absolute calibration stability	1%		Rectangular	Systematic	N/A
c_{lin}	Detector non-linearity	2%		Normal	Systematic	Between all three instruments
c_{stray}	Spectral stray light	Vary spectrally and per instrument due to difference in spectral shape of the signal, should come from the class-based stray light file		Normal	Systematic	Between all three instruments
c_T	Temperature sensitivity	Vary spectrally come from the class-based temperature sensitivity file		Normal	Systematic	Between all three instruments
c_{pol}	Polarisation sensitivity (Radiance only)	Vary spectrally and per instrument to use published data from (Talone and Zibordi, 2016)	Vary spectrally and per instrument triple values for TRIOS, as shown in [AD-1]	Normal	Systematic	Between two radiance instruments
c_{cos}	Cosine response (Irradiance only)	Directional 3.5%	Directional 2%	Normal	Systematic	N/A

5.2.2 FRM branch processing

FRM branch is currently under development and verification testing. In this branch individual instrument characterisation files are used to derive correction coefficients for all instrument related effects apart from polarisation, that could be address in the future study. The FRM processing and uncertainty evaluation will become available in v3 of this report.

5.3 Level 2 processing

At L2 level the individual radiometric quantities are combined further into water leaving radiance (L_w) and remote sensing reflectance (R_{rs}) the products of interest. They are now in common wavelength scale. The measurement function and uncertainty tree diagram for water leaving radiance are shown in Figure 10. At this stage measurements from both radiance instruments are used to derive the measurand a water leaving radiance and the correlation_between term (see Figure 11) is used in the punpy call to correctly evaluate uncertainty in water leaving radiance including correlation between two instruments errors. At L2 level the processor has no more access to the raw instrument data, the individual instrument averaged readings are stored in .HDF files at level L1BCQ for each cast as radiometric quantities, thus in L2 the inputs into the measurement equations are slightly different to those used in L1B, nevertheless they hold the same information about each uncertainty component. That change is represent in Figure 10 mainly in the two top boxes, where the dark corrected signal is multiplied by calibration coefficient ($f_{cal,Lt}$ *and* $f_{cal,Li}$) and stored as cast averaged radiometric quantity $\bar{L_t}$ *and* $\bar{L_i}$ respectively. The standard deviation of that quantity still contains the information about noise during the measurements only and uncertainty in absolute radiometric calibration in not included. This is done where all other instrument related uncertainties are applied thus in the boxes below where L_t *and* L_i,are calculated. This is important to note that now in L2 c_{cal} is equal to 1 and holds only information about the radiometric absolute uncertainty.

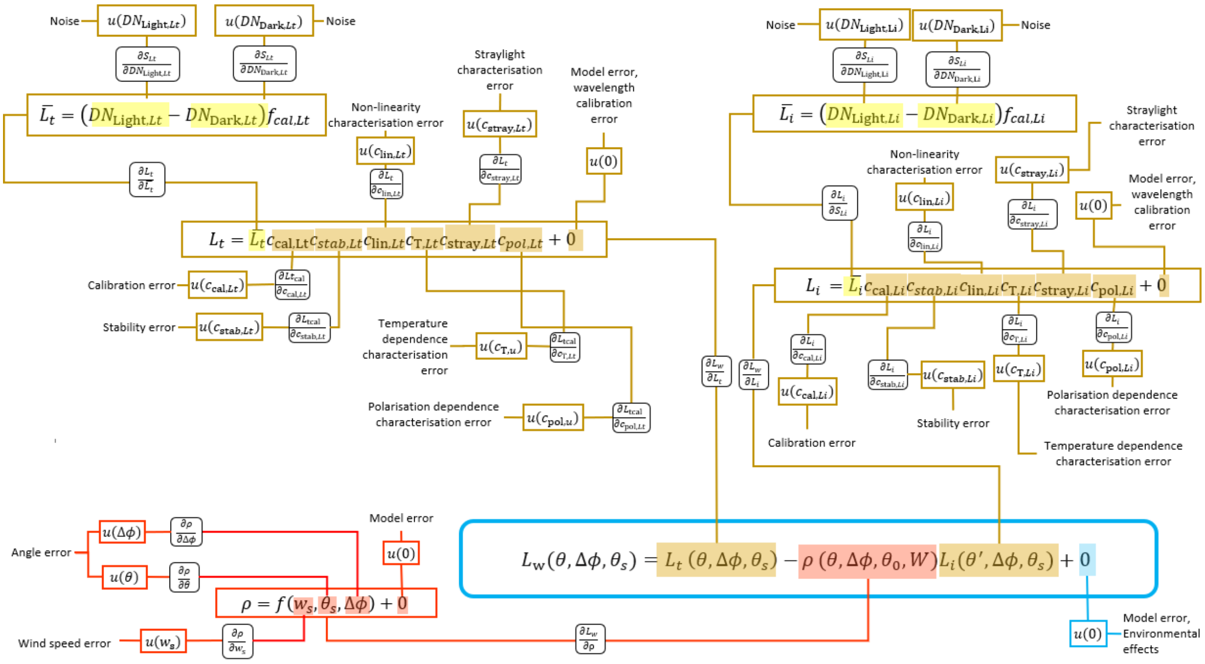

Noise
$u(DN_{Light,Lt})$
$u(DN_{Dark,Lt})$
Noise
$\frac{\partial S_{Lt}}{\partial DN_{Light,Lt}}$
$\frac{\partial S_{Lt}}{\partial DN_{Dark,Lt}}$
$\bar{L_t} = (DN_{Light,Lt} - DN_{Dark,Lt})f_{cal,Lt}$
$\frac{\partial L_t}{\partial \bar{L_t}}$
Non-linearity characterisation error
$u(c_{lin,Lt})$
$\frac{\partial L_t}{\partial c_{lin,Lt}}$
Straylight characterisation error
$u(c_{stray,Lt})$
$\frac{\partial L_t}{\partial c_{stray,Lt}}$
Model error, wavelength calibration error
$u(0)$
$L_t = \bar{L_t} c_{cal,Lt} c_{stab,Lt} c_{lin,Lt} c_{T,Lt} c_{stray,Lt} c_{pol,Lt} + 0$
Calibration error
$u(c_{cal,Lt})$
$\frac{\partial Lt_{cal}}{\partial c_{cal,Lt}}$
Stability error
$u(c_{stab,Lt})$
$\frac{\partial L_{tcal}}{\partial c_{stab,Lt}}$
Temperature dependence characterisation error
$u(c_{T,u})$
$\frac{\partial L_{tcal}}{\partial c_{T,Lt}}$
Polarisation dependence characterisation error
$u(c_{pol,u})$
$\frac{\partial L_{tcal}}{\partial c_{pol,Lt}}$
Noise
$u(DN_{Light,Li})$
$u(DN_{Dark,Li})$
Noise
$\frac{\partial S_{Li}}{\partial DN_{Light,Li}}$
$\frac{\partial S_{Li}}{\partial DN_{Dark,Li}}$
$\bar{L_i} = (DN_{Light,Li} - DN_{Dark,Li})f_{cal,Li}$
Straylight characterisation error
Non-linearity characterisation error
$u(c_{stray,Li})$
$\frac{\partial L_i}{\partial S_{Li}}$
$u(c_{lin,Li})$
$\frac{\partial L_i}{\partial c_{lin,Li}}$
$\frac{\partial L_i}{\partial c_{stray,Li}}$
$u(0)$
Model error, wavelength calibration error
$L_i = \bar{L_i} c_{cal,Li} c_{stab,Li} c_{lin,Li} c_{T,Li} c_{stray,Li} c_{pol,Li} + 0$
$\frac{\partial L_i}{\partial c_{T,Li}}$
$\frac{\partial L_i}{\partial c_{pol,Li}}$
$\frac{\partial L_w}{\partial L_t}$
$\frac{\partial L_w}{\partial L_i}$
$\frac{\partial L_i}{\partial c_{cal,Li}}$
$\frac{\partial L_i}{\partial c_{stab,Li}}$
$u(c_{stab,Li})$
$u(c_{T,Li})$
$u(c_{pol,Li})$
$u(c_{cal,Li})$
Polarisation dependence characterisation error
Stability error
Calibration error
Temperature dependence characterisation error
Model error
$u(\Delta\phi)$
$\frac{\partial \rho}{\partial \Delta\phi}$
Angle error
$u(0)$
$u(\theta)$
$\frac{\partial \rho}{\partial \theta}$
$\rho = f(w_s, \theta_s, \Delta\phi) + 0$
$L_w(\theta, \Delta\phi, \theta_s) = L_t(\theta, \Delta\phi, \theta_s) - \rho(\theta, \Delta\phi, \theta_0, W) L_i(\theta', \Delta\phi, \theta_s) + 0$
Wind speed error
$u(w_s)$
$\frac{\partial \rho}{\partial w_s}$
$\frac{\partial L_w}{\partial \rho}$
$u(0)$
Model error, Environmental effects

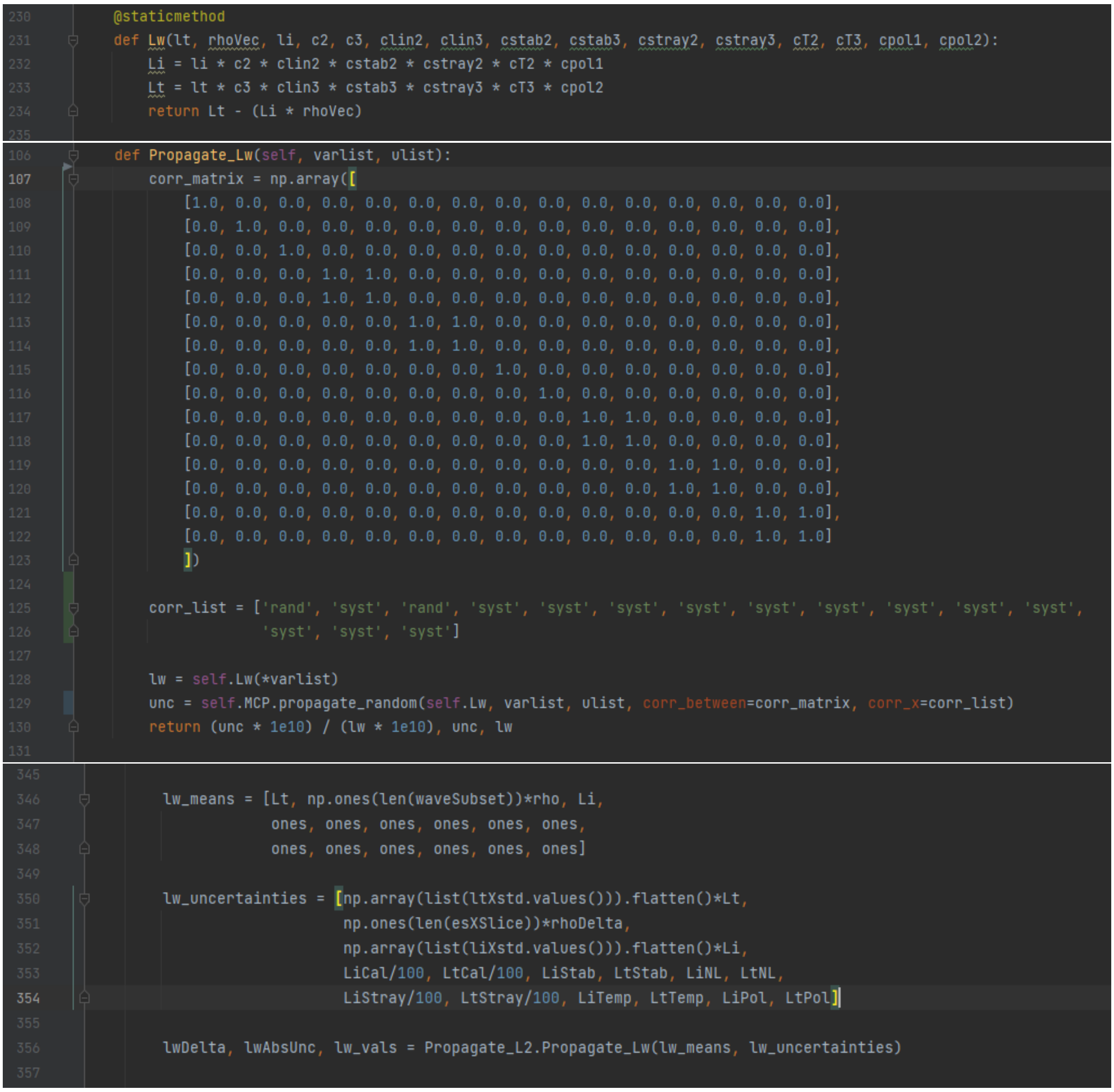
Figure 10 – Uncertainty tree for water leaving radiance (L_w).

$$L_w(\theta,\Delta\phi,\theta_s) = L_t(\theta,\Delta\phi,\theta_s) \; - \; \rho L_i(\theta',\Delta\phi,\theta_s), \qquad \text{Eq. 7}$$

where $L_w(\theta,\Delta\phi,\theta_s)$ is water leaving radiance calculated for a given viewing geometry $(\theta,\Delta\phi)$ and Sun zenith angle (θ_s), $L_t(\theta,\Delta\phi,\theta_s)$ is the total water radiance at given viewing geometry at given Sun zenith angle, ρ is the sea-surface reflectance factor.

The sea-surface reflectance factor (ρ)can be estimated using various models and generally there is still ongoing research on how to best estimated it. More detailed description of the methods inplemented in HPC, their estimated uncertainty and comparison between them can be found in section 5.3.1 Sky and Sun glint removal.

Currently in HPC for default branch processing and for FICE 2 exercise results Mobley (Mobley, 1999) method was used. Sea surface factor uncertainy is evaluated in a sepatere punpy call for the given method. The input components uncertainty for the Mobley method are listed in Table 4. The term +0 is listed in the table as a placeholder not used in the default branch, but could be enabled in a near future, when impovements in HyperCP allow for a quicker calulation of ρ factor using other methods are implement.

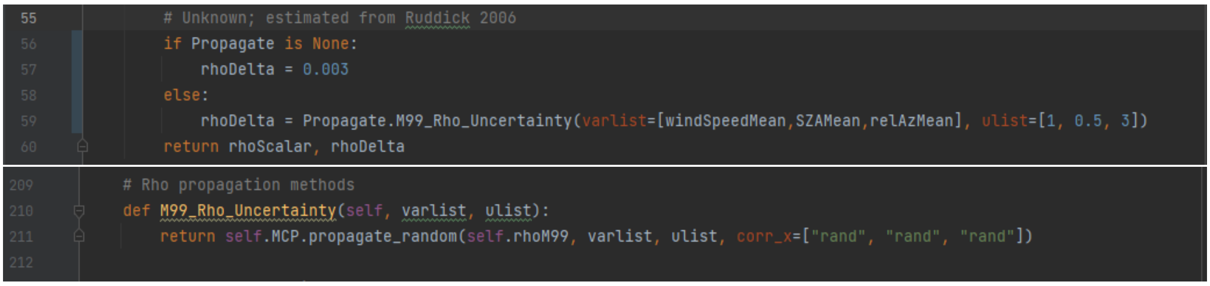


```
    @staticmethod
    def Lw(lt, rhoVec, li, c2, c3, clin2, clin3, cstab2, cstab3, cstray2, cstray3, cT2, cT3, cpol1, cpol2):
        Li = li * c2 * clin2 * cstab2 * cstray2 * cT2 * cpol1
        Lt = lt * c3 * clin3 * cstab3 * cstray3 * cT3 * cpol2
        return Lt - (Li * rhoVec)
```

```
    def Propagate_Lw(self, varlist, ulist):
        corr_matrix = np.array([
            [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0]
            ])

        corr_list = ['rand', 'syst', 'rand', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst',
                     'syst', 'syst', 'syst']

        lw = self.Lw(*varlist)
        unc = self.MCP.propagate_random(self.Lw, varlist, ulist, corr_between=corr_matrix, corr_x=corr_list)
        return (unc * 1e10) / (lw * 1e10), unc, lw
```

```
        lw_means = [Lt, np.ones(len(waveSubset))*rho, Li,
                    ones, ones, ones, ones, ones, ones,
                    ones, ones, ones, ones, ones, ones]

        lw_uncertainties = [np.array(list(ltXstd.values())).flatten()*Lt,
                            np.ones(len(esXSlice))*rhoDelta,
                            np.array(list(liXstd.values())).flatten()*Li,
                            LiCal/100, LtCal/100, LiStab, LtStab, LiNL, LtNL,
                            LiStray/100, LtStray/100, LiTemp, LtTemp, LiPol, LtPol]

        lwDelta, lwAbsUnc, lw_vals = Propagate_L2.Propagate_Lw(lw_means, lw_uncertainties)
```

Figure 11 – Representation of water leaving radiance (L_w) measurement function in L2 processing, as it exists in the HyperCP.



```
        # Unknown; estimated from Ruddick 2006
        if Propagate is None:
            rhoDelta = 0.003
        else:
            rhoDelta = Propagate.M99_Rho_Uncertainty(varlist=[windSpeedMean,SZAMean,relAzMean], ulist=[1, 0.5, 3])
        return rhoScalar, rhoDelta
```

```
    # Rho propagation methods
    def M99_Rho_Uncertainty(self, varlist, ulist):
        return self.MCP.propagate_random(self.rhoM99, varlist, ulist, corr_x=["rand", "rand", "rand"])
```

Figure 12 – Representation of sea surface reflectance (ρ) measurements function in L2 processing, as it exists on HPC

Table 4 – Summary information about each uncertainty component for sea surface reflectance factor (ρ) estimation using Mobley method

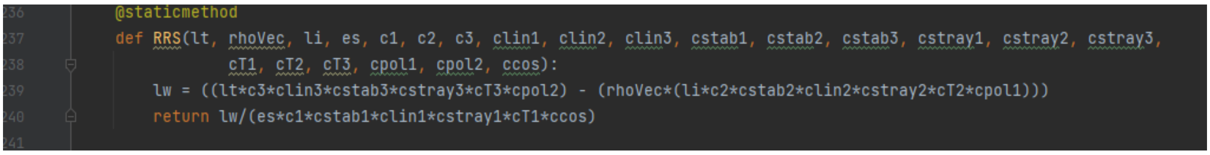
Variable symbol	**Variable name/description**	**Exemplary uncertainty magnitude**	**PDF shape**	**Correlation** 'corr_x'	**Correlation between** 'corr_between'
ρ	Sea surface reflectance	Calculated for each cast depends on all input components, especially wind speed	Normal	Random	N/A
w_s	Wind speed	1ms^{-1}	Normal	Random	N/A
$\Delta\phi$	Relative azimuth	3°	Normal	Random[2]	N/A
θ_s	Solar zenith angle	0.5°	Normal	Random	N/A
+0	Model error	Difference between Mobley and Zhang method	Rectangular	Systematic	N/A

The remote sensing reflectance is then calculated, the tree diagram can be found in Figure 2, and is expressed by Eq. 8. Please note that the same data for downwelling irradiance are stored in L1BCQ .HDF file like for both radiance instruments, thus signal from irradiance sensor is too represented as radiometric quantity in L2, and the same logic in combining uncertainty for downwelling irradiance at L2 applies like for both radiances.

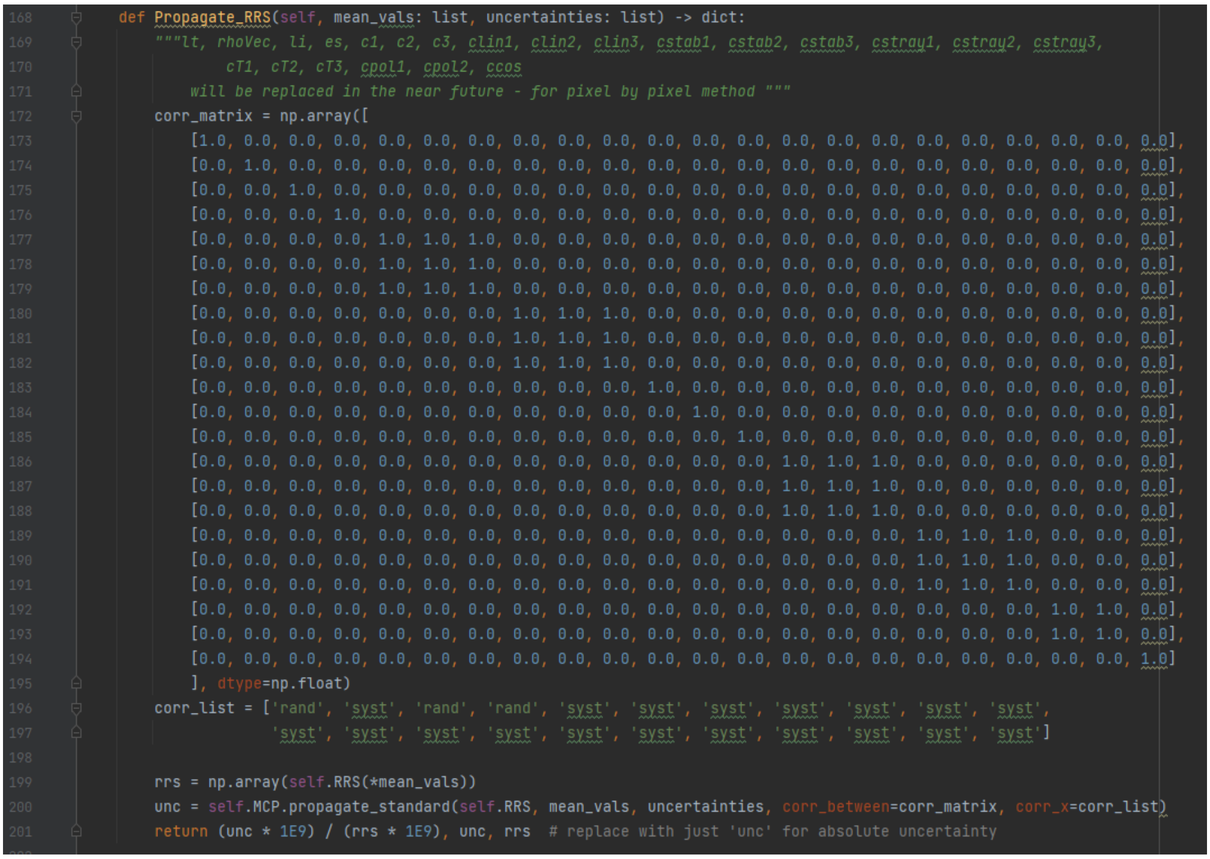
$$R_{rs}(\theta,\Delta\phi,\theta_s)=\frac{L_w(\theta,\Delta\phi,\theta_s)}{E_s(\theta_s)}, \qquad \text{Eq. 8}$$

where the water leaving radiance (L_w) is divided by downwelling irradiance (E_s). At this stage all the components and input uncertainties were already defined, the important aspect when remote sensing reflectance in called is to correctly define the correlation_between term as now all three instruments and theirs associated uncertainty inputs are used and the correlation between matrix has 22 inputs, as it is shown in Figure 13.

[2] This can be systematic for one deployment when the instrument is installed once and used, but random from cruise to cruise or campaign to campaign as the set up will be new every time.



```
    @staticmethod
    def RRS(lt, rhoVec, li, es, c1, c2, c3, clin1, clin2, clin3, cstab1, cstab2, cstab3, cstray1, cstray2, cstray3,
              cT1, cT2, cT3, cpol1, cpol2, ccos):
        lw = ((lt*c3*clin3*cstab3*cstray3*cT3*cpol2) - (rhoVec*(li*c2*cstab2*clin2*cstray2*cT2*cpol1)))
        return lw/(es*c1*cstab1*clin1*cstray1*cT1*ccos)
```



```
    def Propagate_RRS(self, mean_vals: list, uncertainties: list) -> dict:
        """lt, rhoVec, li, es, c1, c2, c3, clin1, clin2, clin3, cstab1, cstab2, cstab3, cstray1, cstray2, cstray3,
              cT1, cT2, cT3, cpol1, cpol2, ccos
           will be replaced in the near future - for pixel by pixel method """
        corr_matrix = np.array([
            [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0],
            [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
            ], dtype=np.float)
        corr_list = ['rand', 'syst', 'rand', 'rand', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst',
                     'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst', 'syst']

        rrs = np.array(self.RRS(*mean_vals))
        unc = self.MCP.propagate_standard(self.RRS, mean_vals, uncertainties, corr_between=corr_matrix, corr_x=corr_list)
        return (unc * 1E9) / (rrs * 1E9), unc, rrs  # replace with just 'unc' for absolute uncertainty
```

Figure 13 – Representation of remote sensing reflectance (R_{rs}) call.

5.3.1 Sky and Sun glint removal

Above water surface measurements, like those incorporated into the HyperCP, derive water leaving radiance (L_w) by subtracting a reflected radiance (L_r) from the total water radiance (L_t) of a sea-viewing sensor. L_r may not be directly measured however, it must be derived using ancillary data, Inherent Optical Properties (IOPs), statistical modelling, and simulation efforts. The estimated radiance reflected for a given direction (θ, ϕ) is:

$$L_r(\theta, \phi, \lambda) = \rho L_i(\theta', \phi, \lambda), \qquad \text{Eq. 9}$$

Where θ' is the zenith angle of the radiometer pointing upwards to the sky in above water measurements nominally ($\theta' = 140°$), and the θ is zenith angle of the other downwards towards the water ($\theta = 40°$). They are both at the same azimuth angle i.e. the difference between the sun and the sensor ($\phi = 90°$or $\phi = 135°$), ρ is the sea-surface reflectance factor, given by:

$$\rho = \frac{\int_0^{2\pi} \int_0^{\frac{\pi}{2}} r(\theta^*, \phi^* \rightarrow \theta, \varphi) L_{sky}(\theta^*, \phi^*) sin\theta^* d\theta^* d\varphi^*}{L_i(\theta', \phi, \lambda)}. \qquad \text{Eq. 10.}$$

Here $r(\theta^*, \phi^* \rightarrow \theta, \phi)$ is the Fresnel reflectance for a light ray traveling from θ^*, ϕ^* and reflecting up into the instrument field of view (FOV), $L_{sky}(\theta^*, \phi^*)$ denotes a skylight distribution for the angle θ^*, ϕ^*.

The OC community employs different methods to calculate ρ, making use of different skylight distribution models, assumptions, and wave statistics. D'Alimonte, *et al.* compared estimations of ρ using different wave statistics and skylight distribution models for 550 nm, where they found ρ to restrict to within 2% boundaries for low windspeeds (up to 4 ms^{-1}) and solar zenith angles SZA above 20⁰ (D'Alimonte *et al.*, 2021). As noted by others, these approaches do not factor in polarisation or the downwelling irradiance which has been shown to effect surface reflectance, dependent on sea conditions (Harmel *et al.*, 2012).

The HyperCP, at the time of this report, contains two fully implemented methods of calculating ρ: the Mobley method (Mobley, 1999) and Zhang method (Zhang *et al.*, 2017).

The Mobley method is perhaps the most used within the OC community. The surface reflectance factor was defined by lookup tables based on Cox-Munk wave-statistics (Mobley, 1999). The inputs to this model are the relative azimuth of the measurements, the windspeed and a clear-sky condition. This method does not account for many variables which are known to effect ρ, therefore, a model uncertainty must be assumed. Figure 14 present uncertainty tree diagram for Mobley method to estimate sea surface reflectance.

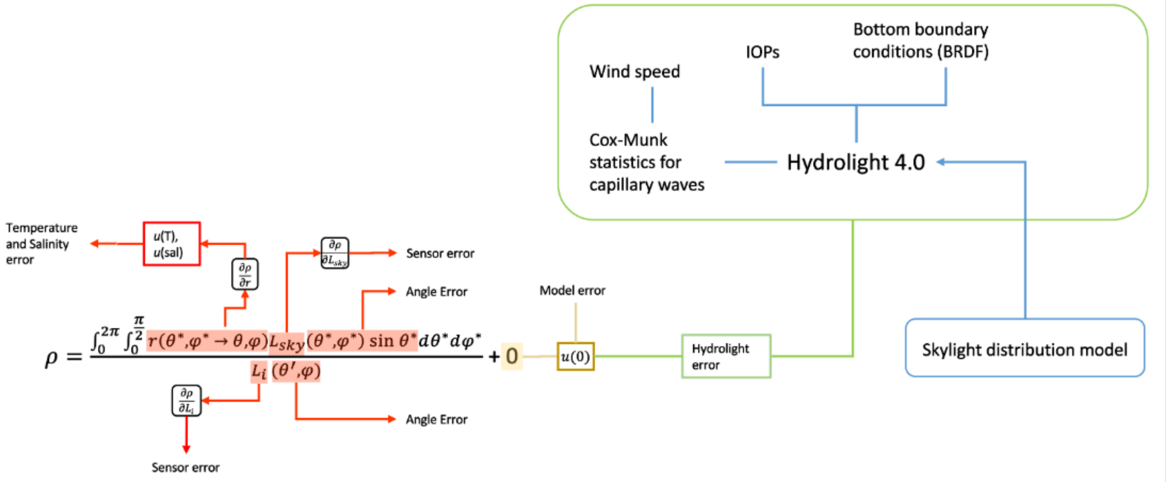

Bottom boundary conditions (BRDF)
IOPs
Wind speed
Cox-Munk statistics for capillary waves
Hydrolight 4.0
Temperature and Salinity error
$u(\mathrm{T})$, $u(\mathrm{sal})$
$\frac{\partial\rho}{\partial r}$
$\frac{\partial\rho}{\partial L_{sky}}$
Sensor error
Angle Error
Model error
$\rho = \frac{\int_0^{2\pi}\int_0^{\frac{\pi}{2}} r(\theta^*,\varphi^* \rightarrow \theta,\varphi) L_{sky}(\theta^*,\varphi^*) \sin\theta^* d\theta^* d\varphi^*}{L_i\,(\theta',\varphi)} + 0$
$u(0)$
Hydrolight error
Skylight distribution model
$\frac{\partial\rho}{\partial L_i}$
Angle Error
Sensor error

Figure 14. Uncertainty Tree Diagram for the Mobley 1999 method of calculating ρ.

The previous methods have the great advantage of simplicity (for the user) however, they do not account for spectral effects or account for the different spectral profiles of sun glint and sky glint. Where the methods discussed previously will generate a value for ρ which covers all pixels of a multi/hyperspectral instrument, Zhang calculates ρ per pixel. The uncertainties associated with the assumptions made are smaller, however there are greater input uncertainties due to the method requiring more input parameters. Additionally, the Zhang model accommodates for Sun glint, with a separate spectral response function and brightness. The uncertainty tree diagram for Zhang method is shown in Figure 15. The practice of taking *in situ* measurements above oceans has long included specific placement of the instruments, at 90⁰ or 135⁰ relative azimuth angle from the sun to minimise the effect of Sun glint.

The Zhang method, although implemented in the HyperCP, is currently being improved. The method reflects a computationally intensive calculation which precludes quick data analysis for big data series. In addition, adding MC uncertainty calculation, which requires many iterations of HyperCP code, to an already intensive model makes the whole process even longer. In the development stage we calculated uncertainty in Zhang model for few *in situ* data examples and to compare Mobley and Zang method, but in the default branch at present, uncertainty propagation for the Zhang model is not activated. Nevertheless, the work is ongoing, and all relevant code is already defined. There is a potential for this option to become available shortly, or to be reactivated from the source code if the user wishes to do so.

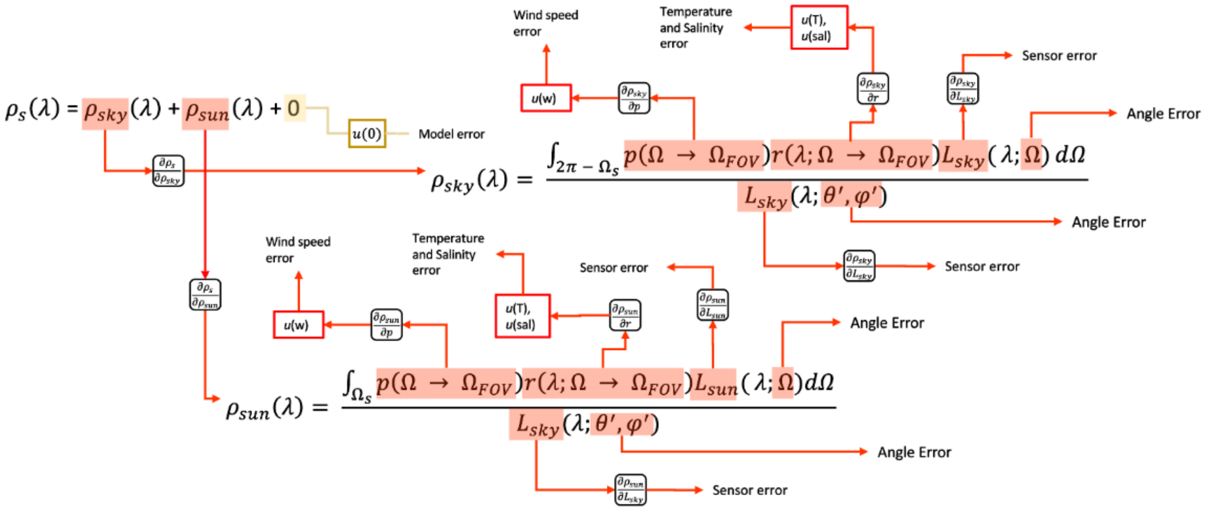

$\rho_s(\lambda) = \rho_{sky}(\lambda) + \rho_{sun}(\lambda) + 0$
$u(0)$
Model error
$\frac{\partial\rho_s}{\partial\rho_{sky}}$
$\frac{\partial\rho_s}{\partial\rho_{sun}}$
$\rho_{sky}(\lambda) = \frac{\int_{2\pi-\Omega_s} p(\Omega \to \Omega_{FOV}) r(\lambda;\Omega \to \Omega_{FOV}) L_{sky}(\lambda;\Omega)\, d\Omega}{L_{sky}(\lambda;\theta',\varphi')}$
Wind speed error
u(w)
$\frac{\partial\rho_{sky}}{\partial p}$
Temperature and Salinity error
u(T), u(sal)
$\frac{\partial\rho_{sky}}{\partial r}$
$\frac{\partial\rho_{sky}}{\partial L_{sky}}$
Sensor error
Angle Error
Angle Error
Sensor error
$\rho_{sun}(\lambda) = \frac{\int_{\Omega_s} p(\Omega \to \Omega_{FOV}) r(\lambda;\Omega \to \Omega_{FOV}) L_{sun}(\lambda;\Omega) d\Omega}{L_{sky}(\lambda;\theta',\varphi')}$
Wind speed error
u(w)
$\frac{\partial\rho_{sun}}{\partial p}$
Temperature and Salinity error
u(T), u(sal)
$\frac{\partial\rho_{sun}}{\partial r}$
Sensor error
$\frac{\partial\rho_{sun}}{\partial L_{sun}}$
Angle Error
Angle Error
$\frac{\partial\rho_{sun}}{\partial L_{sky}}$
Sensor error

Figure 15. Uncertainty Tree Diagram for the Zhang method.

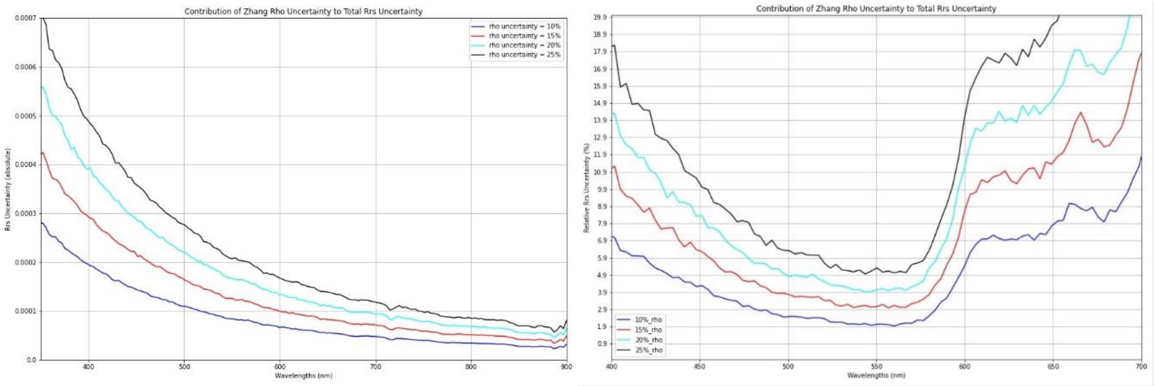


Figure 16. Sensitivity of R_{rs} uncertainty to ρ. R_{rs} uncertainty propagated with only ρ uncertainty contributions, set to 10%, 15%, 20%, & 25% of the ρ value as calculated by Zhang method. Presented as an absolute uncertainty (left) and relative (right)

A simple test was performed to study the sensitivity of ρ uncertainty on R_{rs}. Using data from FICE-1 HyperOCR instrument that belong to PLM.

To study the effect of ρ uncertainty on R_{rs} a simple test was performed; the uncertainty in ρ was taken to be a flat percentage of the calculated value per wavelength for the Zhang method. Then this was propagated using MC with differing values for ρ uncertainty and applying no other uncertainty contributions. The resulting graph can be seen in Figure 16, for $u(\rho) = 0.1 \cdot \rho, 0.15 \cdot \rho, 0.2 \cdot \rho, 0.25 \cdot \rho$, showing the absolute and relative uncertainty in R_{rs} per wavelength in these test cases. The blue data series are for ρ set to 10% and we can see the significant effect of this uncertainty contributor 2% uncertainty in R_{rs} for 550 nm increasing to 7% for both shorter and longer wavelengths.

5.3.1.1 Rho Uncertainty Study

Finding a full characterisation of uncertainty requires sea surface reflectance factor (ρ) uncertainty to be propagated for the various methods native to the processor. Two methods are included: Mobley and Zhang, as well as a third reference from FRM4SOC- 1. The Ancillary uncertainties are a point of contention within the entire OC field. Real uncertainties are used where possible, but some ancillary values and uncertainties have been estimated for this study. Since the area of interest here is the difference between the methods and their poorly defined model errors, it is considered acceptable in this context.

Table 5. Ancillary data and Ancillary uncertainties inputs for Rho calculation from FICE.

	Cast 1	Cast 2	Uncertainty
Windspeed	0.50	3.10	1 (0.3 for cast 2)
AOD	0.112	0.297	0.01
SZA	35.91	30.38	0.5
SST	20	20	2
Salinity	35	35	0.5
Relative Azimuth	90	90	3

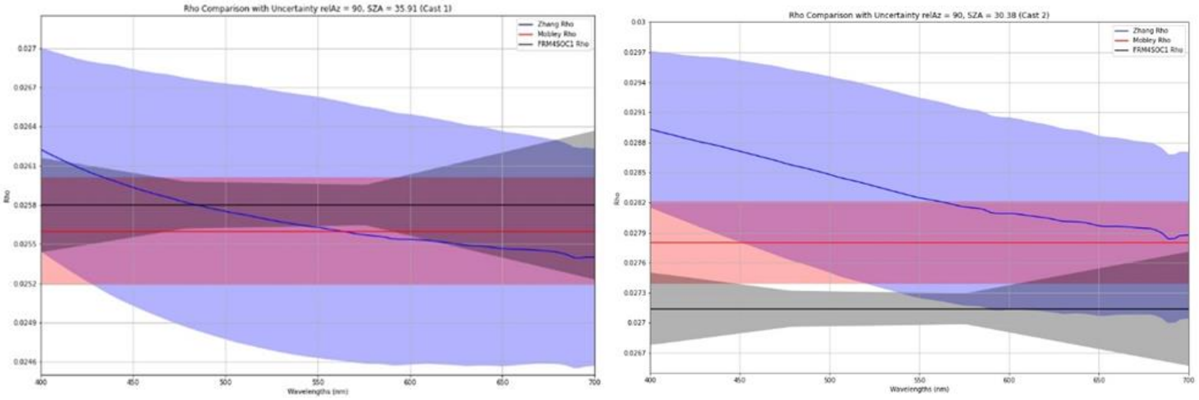


Figure 17. Rho as calculated with Zhang and Mobley methods relative Azimuth 90 degrees with associated input uncertainties for cast 1(left) and cast 2 (right). Additionally, Rho calculated in FRM4SOC1 has been added for both casts, taken from (Bialek and Douglas, 2019)

Figure 17 shows the result of the comparison of three ρ evaluation methods with their uncertainties indicated as shaded areas. We can see that for the example in the left panel the three methods agree within their estimated uncertainties, this is the case for a very low wind speed of 0.5 ms^{-1}, the increase of wind speed to 3.1 ms^{-1} the case from right panel leads to significant discrepancy between the models. The FRM4SOC1 method used Ruddick model that as the author says should not be used for a general case as was derived a function of wind speed for a specific water case.

5.4 Band convolution

The HyperCP includes, by default, options to convolve the L2 output into several commonly used satellite bands. Convolution uncertainties are provided for Sentinel 3 OLCI band convolution. Uncertainties in the Spectral Response Function (SRF) are not digested into the processor for 'default' branch processing. The band convolution uncertainties for the default branch are derived using the MC, from knowledge of the signal mean and associated uncertainty. The L2 products, L_w and R_{rs}, are convolved with the HyperCP's usual processing, however convolution uncertainties are provided by a Python package called matheo. This is accomplished by passing the signal means and averages through punpy, using the matheo band convolution function for Sentinel3 – OLCI as a measurement function. Matheo will check for the most up to date SRF available to be applied, making use of the RelativeSpectralResponse Class from the pyspectral rsr_reader.

In the current default processing both L1B and L2 products consider error-correlations between input uncertainties and broadly apply random or systematic correlations between pixels where appropriate. The band convolution applies to L2 products which exist as a mix of inputs, some of which have random pixel-pixel correlation, and others systematic. Therefore, the correlation of the convolved products, L_w and R_{rs}, cannot be fully random or systematic. To get a sufficient uncertainty output it is necessary to split R_{rs} into its random and systematic components and propagate their uncertainties separately. The band convolution uncertainty is therefore calculated twice; once for the random uncertainty components (Noise) and again for the rest. The resulting uncertainty values representing R_{rs} can then be added in quadrature once they have been passed through band convolution. The default branch band convolution uncertainty in R_{rs} can be seen in the 6 Results section, Figure 26.

6 Results

In this section we present some examples with uncertainty evaluation for *in situ* data. First, we show a preliminary analysis done using FICE -1 data from HyperOCR instrument as inputs to verify the pupny package usage, where a simplified evaluation with few uncertainty contributors is run to ensure that analytical method and MC method give us consistent results.

Section 6.2 Default branch uncertainties presents the results of the uncertainty evaluation implemented in HPC for the default branch and is run on *in situ* data from FICE-2 experiment. The details about FICE-2 can be found in [AD4].

6.1 Validating Punpy against LPU

The software base of the HyperCP, HyperInSPACE, provides time-averaged data as well as the associated standard deviations to the later stages of the processing chain. Before the addition of all uncertainty contributors, such as absolute calibration uncertainties, these standards deviations were used to validate punpy by comparing its output directly to the LPU and HyperCP implemented sum of squares uncertainty estimation. Firstly, the correct implementation of the LPU in this case must be found using Eq. 2. For that exercise we use Eq. 11 as a measurements function. We are in L2 processing level and can assume the red branch from Figure 4 thus no other uncertainty inputs available. The L_t, L_i, E_s values are averaged for each cast and standard deviation from that cast is an estimate of noise related uncertainty, in addition a value of $u(\rho) = 0.003$ was originally assigned in HyperCP to sea surface reflectance factor, thus we use it. None of the input uncertainties are correlated in this simplistic case.

The LPU simplifies to summation by quadrature when the measurements equation has only multiplication/division operation, or addition/subtraction but they cannot be both present in one equation. However, the measurement function Eq. 11 contains both additive and multiplicative components meaning that this simplification may not be made.

$$R_{rs} = \frac{L_t - \rho L_i}{E_s} \qquad \text{Eq. 11}$$

Splitting the uncertainty propagation is often an ideal solution for calculating the LPU in such cases, as propagated uncertainties may be used as inputs to the LPU in lieu of more complicated equations. Propagated R_{rs} uncertainty may then be used to verify punpy in this simple case. Where Eq. 12 is used to evaluate absolute uncertainty in reflected radiance ($L_r = \rho L_i$) term first applying correct sensitivity coefficients. Then to estimate relative water leaving reflectance $u(L_w)$ uncertainty, due to the minus sign in the equation the uncertainties need to be combined in absolute terms and then divided by the value of water leaving radiance to convert it to relative uncertainty expressed in % (Eq. 13), only then the sum of square solution can be applied to evaluate remote sensing reflectance uncertainty as it is showed in Eq. 14, where both inputs component are relative uncertainties.

$$u_{abs}(L_r) = \sqrt{L_i{}^2 * u(\rho)^2 + \rho^2 * u(L_i)^2} \qquad \text{Eq. 12}$$

$$u(L_w) = \sqrt{\frac{u_{abs}(L_t)^2 + u_{abs}(L_r)^2}{L_w^2}} \qquad \text{Eq. 13}$$

$$u(R_{rs}) = \sqrt{u(L_w)^2 + u(E_d)^2} \qquad \text{Eq. 14}$$

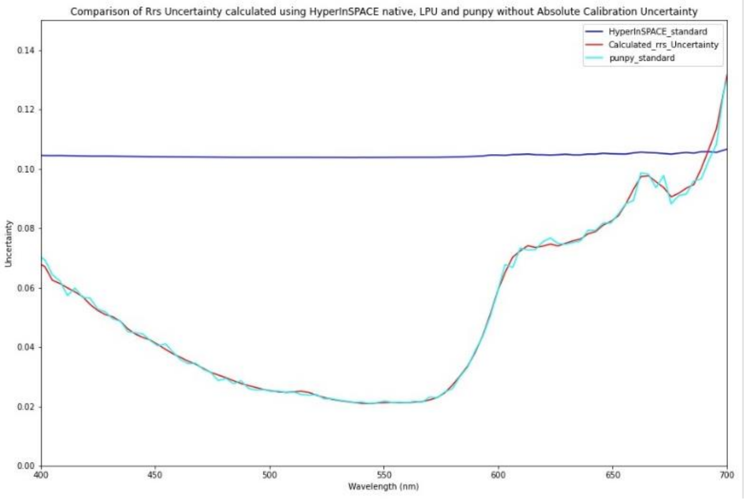


Figure 18. Remote sensing reflectance uncertainty propagated for noise and ρ uncertainty, using HyperInSPACE's native uncertainty method, LPU and Punpy.

The result (Figure 18) compares all three methods for a simple case where only the instrument standard deviations and $u(\rho) = 0.003$ are propagated. As can be clearly seen, LPU and punpy are in close agreement, any difference being associated with the stochastic nature of the MC method. Likewise, the disagreement in HyperInSPACE shown in the graph can be clearly attributed to the inadequacy of summation by quadrature to propagate uncertainty for this measurement function.

6.2 Default branch uncertainties

We present here one example of the default HPC output plots for each instrument class using the data from a selected cast from FICE2 experiment. For the full range of the results please see [AD-4]. The plots in Figure 19 and Figure 20, show the HyperOCR instrument, whereas Figure 21 and Figure 22 shows the same data from TRIOS instrument example. The output .HDF file contains information about uncertainties and user can just decide to run a default processor without any interaction with uncertainty evaluation part and obtain a default uncertainty estimation.

Figure 19 and Figure 21 are the outputs of L1B processing step, where still each instrument measurements is treated separately, thus they show three radiometric quantities L_t, L_i, E_s for a selected cast with associated uncertainty values marked as shaded areas. Figure 20 and Figure 22 show L2 processing outputs with L_w and R_rs results displayed with associated uncertainties.

In Figure 23 and Figure 24 relative uncertainties are plotted for L1B products for both instruments and L_w and R_rs from L2. Generally, uncertainties for the good casts for all instruments are below 5%, k=1, for wavelengths in the range 450 nm-700 nm except for a peak around 590 nm due to straylight uncertainty in total water radiance measurement. The uncertainties increase in the short and long wavelengths range as expected.

We can see in the right panel of Figure 23 that water leaving radiance uncertainties for that example of *in situ* data from FICE2 experiment are very similar, however, the uncertainty in remote sensing reflectance is significantly higher for the TriOS instrument. This increase can be explained by a higher cosine diffuser error for TriOS instrument class that is affecting downwelling irradiance measurements and is higher than for HyperOCR with values 3.5% and 2% respectively (as indicated in Table 3), E_s relative uncertainties are presented on its own in the top panel of Figure 24.

For sky radiance HyperOCR shows higher uncertainty especially for longer wavelength (the bottom left panel of Figure 24), which is due to higher uncertainty of polarisation sensitivity for this instrument class as indicated in Table 3. Uncertainties in the total water radiance are similar as the polarisation effects mainly affect sky radiance measurements and have insignificant contribution to water measurements (Talone and Zibordi, 2016).

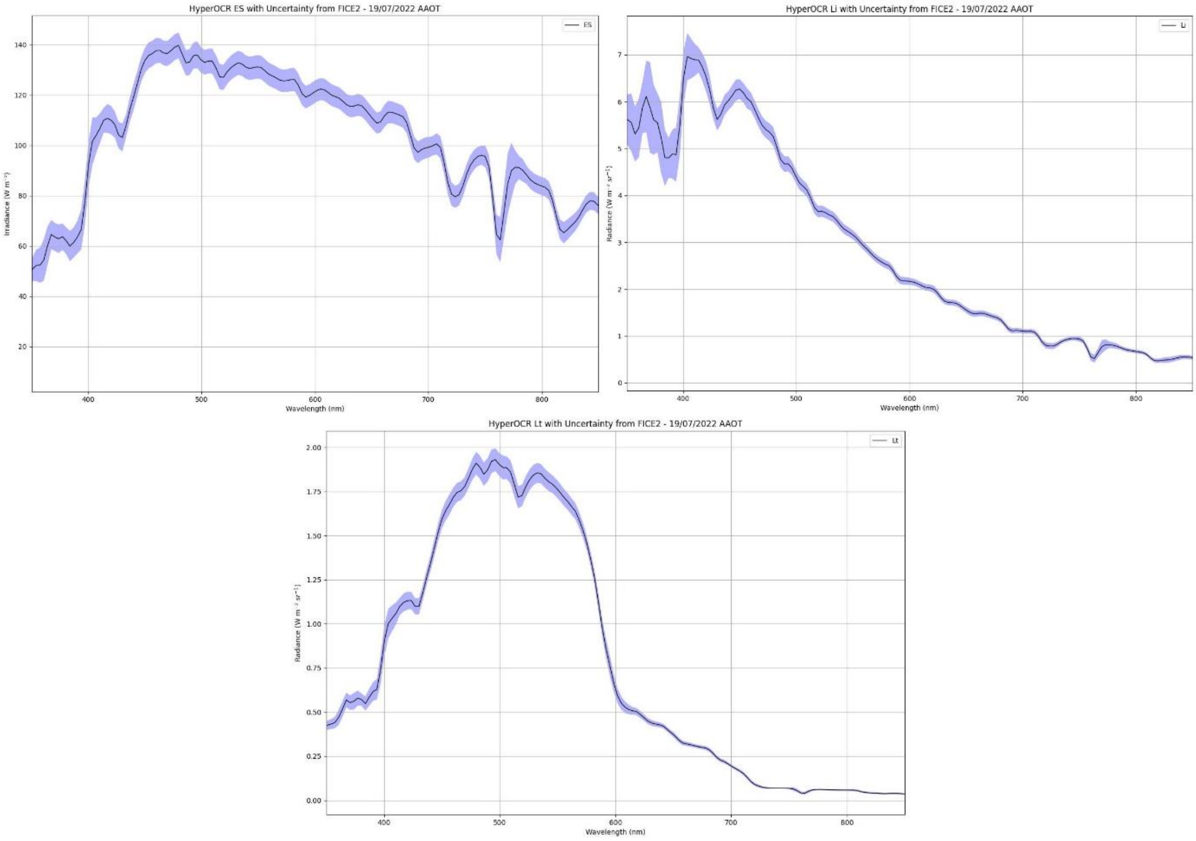


Figure 19. Plots to show HyperCP L1B output for HyperOCR data collected from the AAOT during FICE2. Relative uncertainties are displayed as a shaded blue area with: Top left – E_s, top right –L_i, and bottom – L_t.

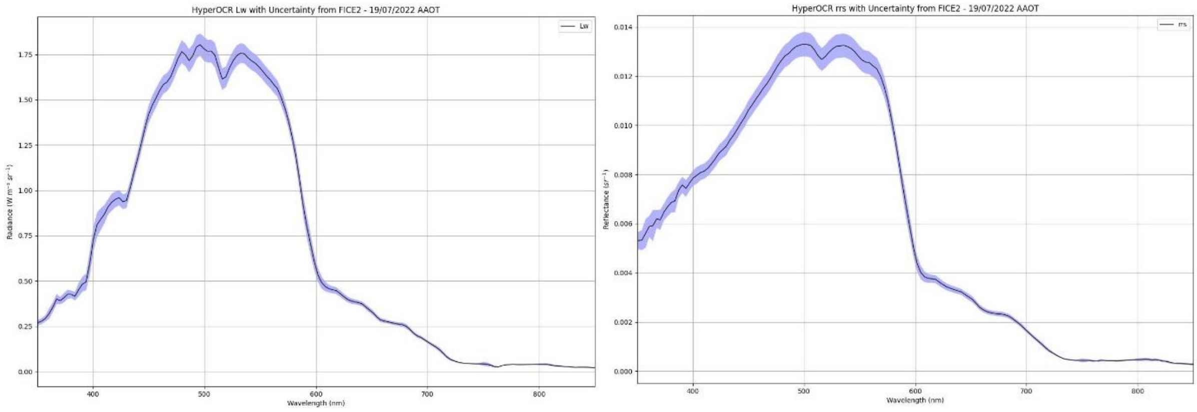


Figure 20. HyperCP output for HyperOCR, collected from AAOT during FICE2 with relative uncertainties. Left L_w, Right R_rs.

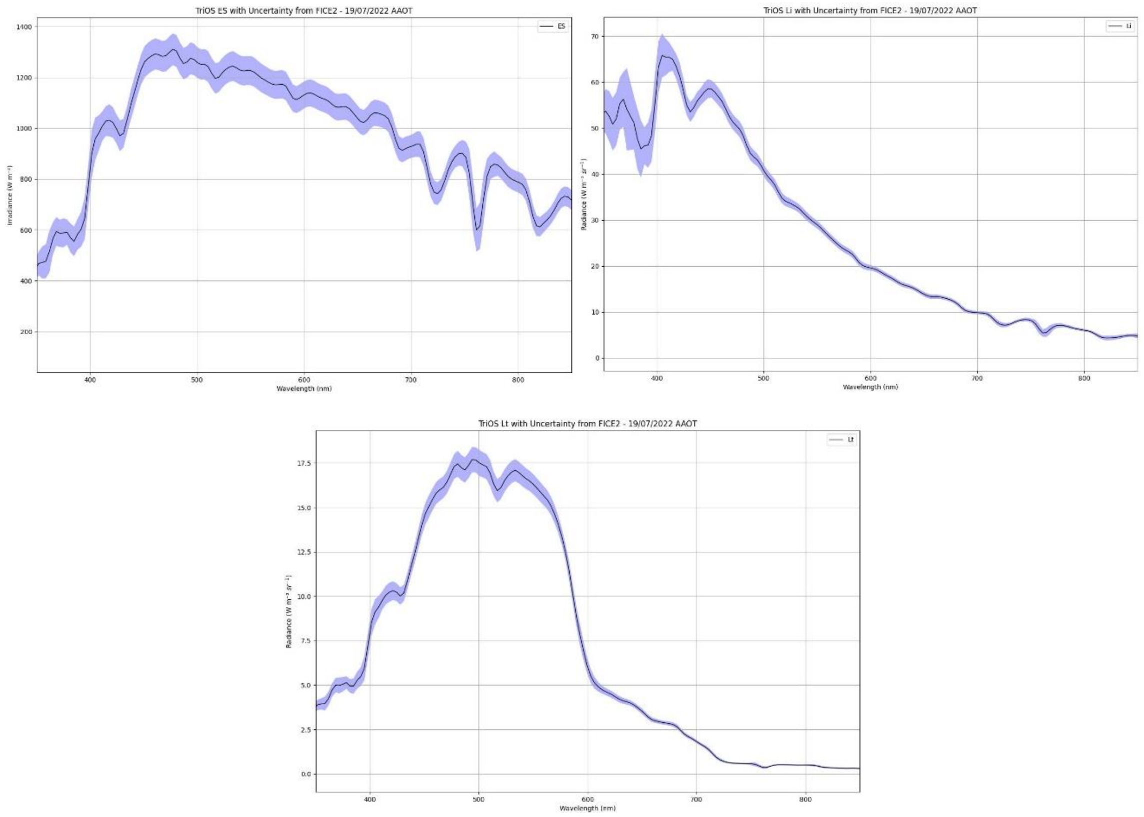


Figure 21. HyperCP L1B output for TriOS sensor, taken from AAOT data collected during FICE2. Uncertainties displayed in the shaded blue area with: Top left – E_s, top right –L_i, and bottom – L_t.

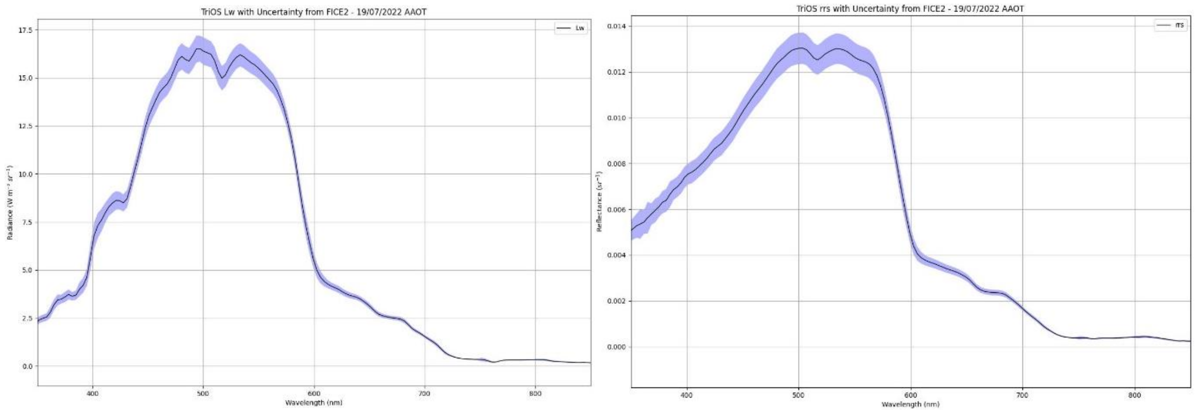


Figure 22. Plots to show HyperCP L2 output for TriOS data collected from the AAOT during FICE2. Relative uncertainties are displayed as a shaded blue area with: Left – Lw, Right - $\mathrm{R_{rs}}$.

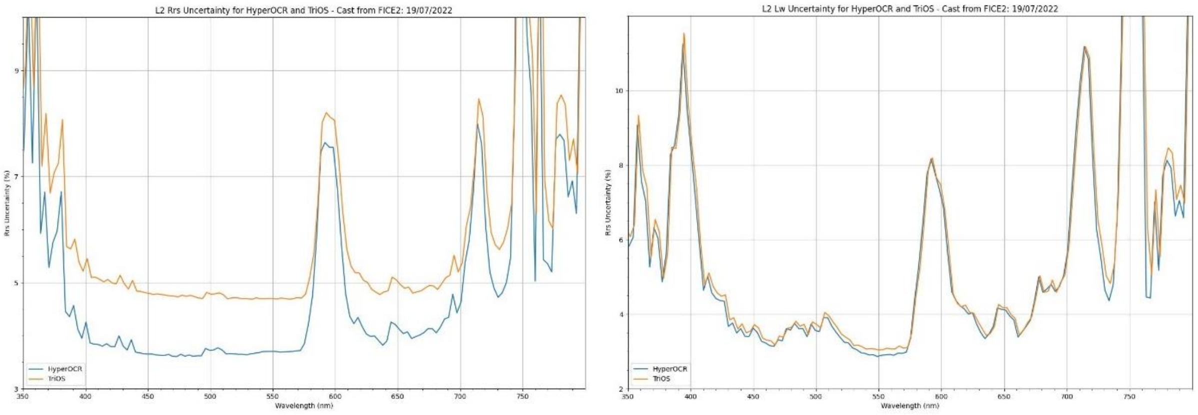


Figure 23. Default branch uncertainty in L2 products for TriOS (orange line) and HyperOCR (blue line). Casts from FICE-2 19-07-2022 are used as an example: 8:00 for TriOS, 10:40 HyperOCR. The products shown are: R_{rs} (left) and L_{w} (right)

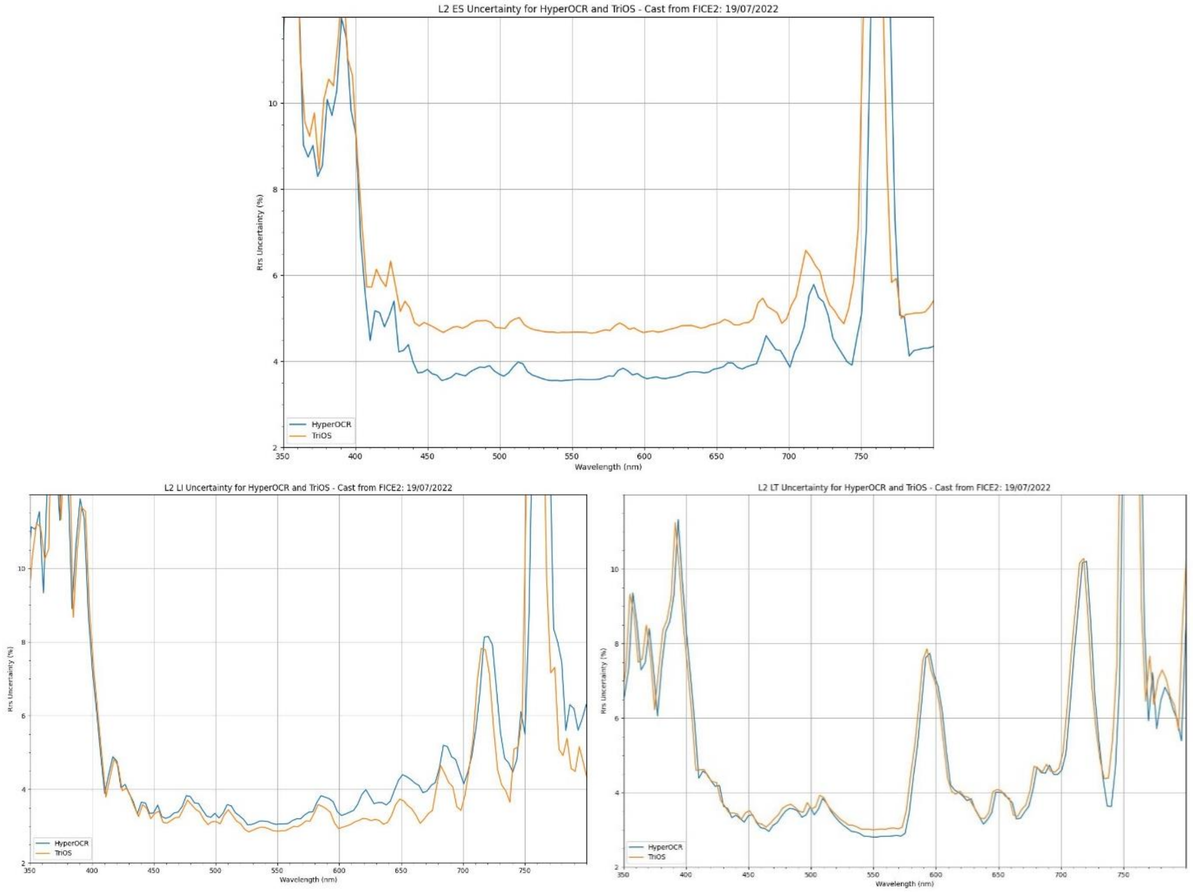


Figure 24. Default branch uncertainties in L1B products for TriOS (orange line) and HyperOCR (blue line). Casts from FICE-2 19-07-2022 are used as an example: 8:00 for TriOS, 10:40 HyperOCR. The products shown are: E_{s} (top), L_{i} (bottom left), and L_{t} (bottom right).

To get more information about each uncertainty component contribution Figure 25 presents the uncertainty "evolution" as the contributors are added into the propagation function for HyperOCR instrument. The blue data series show the uncertainty in each L1B output due to the instrument noise only, which is literally negligible. We can see only a small increase for total water measurements for longer wavelength (bottom right panel) where the measured signal is very low. This uncertainty can be higher if the measurands are not taken at the optimal environmental conditions and are not filtered. The orange data series represents the effect of the instrument noise and absolute radiometric calibration only on the L1B products triplet. This is a typical absolute radiometric uncertainty curve that increase in the short wavelength range. Green data series included radiometric calibration stability on top of the noise and radiometric calibration uncertainty. For the red one the detector nonlinearity effects are added. The purple one includes thermal sensitivity, where we can see the increase in uncertainty values with the longer wavelengths as expected for the thermal sensitivity of silicon sensors. The brown data series introduce some uncertainty peaks due to spectral straylight uncertainty and finally the pink data series shows all uncertainty contributors, thus for the top panel there is an addition of cosine diffuser error and for both bottom panels instrument polarisation sensitivity.

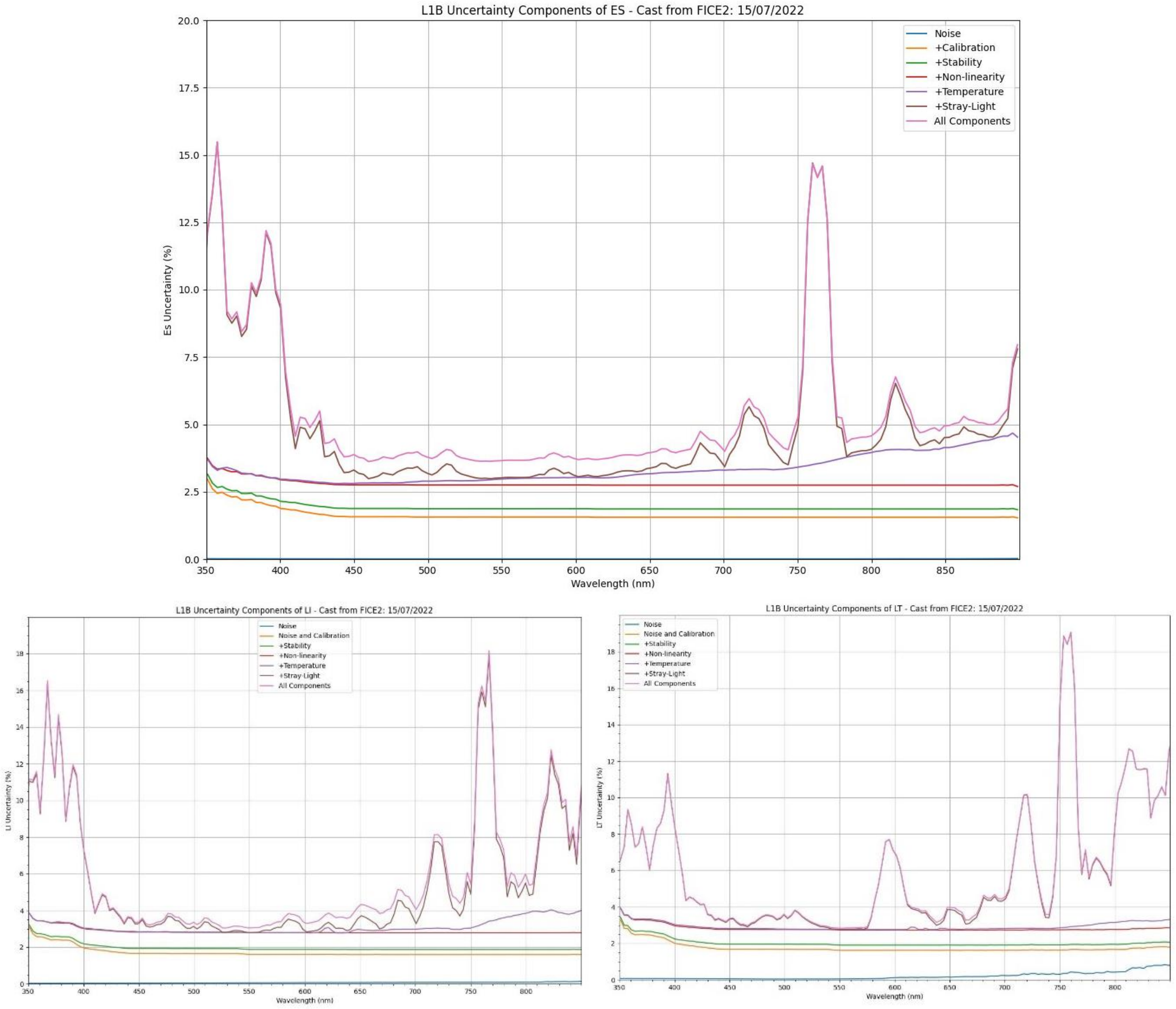


Figure 25. Graphical breakdown of the uncertainty components of the default branch for E_s (top), L_i (bottom left), & L_t (bottom right) for HyperOCR instrument. Each line represents the uncertainty at the L1B processing step, with each of the listed components factored into the HyperCP's uncertainty propagation and the rest set to zero.

The last step where uncertainties are currently implemented in HyperCP are the satellite band integrated values for Sentinel 3 OLCI bands. The difference between hyperspectral and S3 band integrated uncertainties is presented in Figure 26. The band integrated uncertainty is almost the same as the hyperspectral uncertainty. The uncertainties from few wavelengths that are used to derive band integrated values are dominated by systematic contributors, and do not reduce with square root on the number of input wavelengths as this is the case for random components, thus we do not observe reduction in band integrated uncertainty value

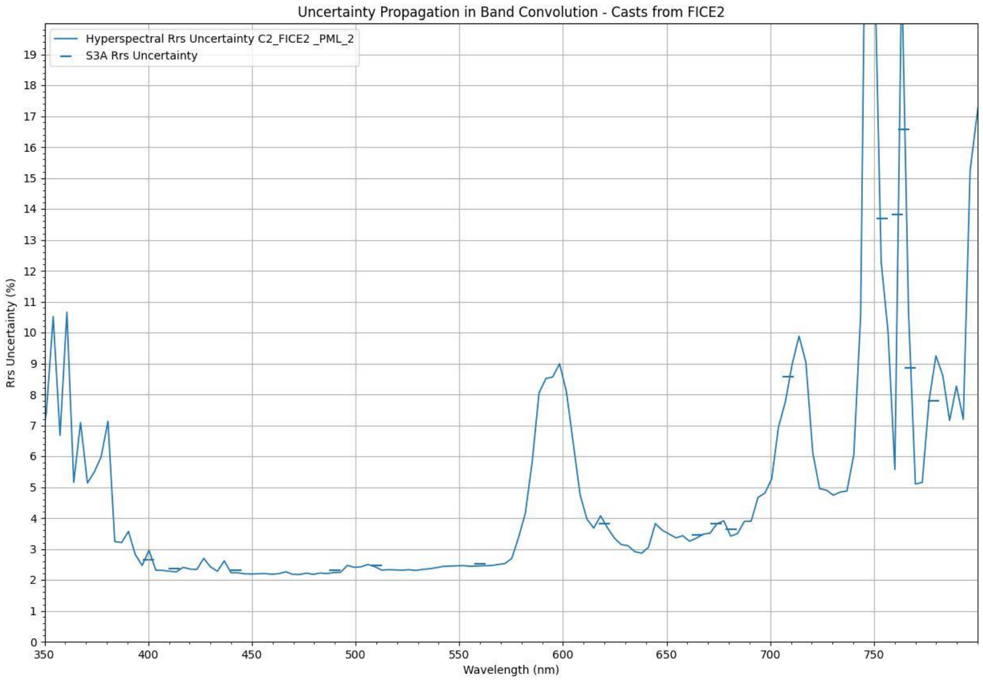


Figure 26. R_{rs} uncertainty for FICE2 19/07/2022 data example from, using PLM HyperOCR data and default branch uncertainty evaluation. The black line indicates uncertainty per instrument pixel, the blue bars indicate S3 band integration.

7 Summary

This report is a second version of D-10 Protocols for uncertainty budget calculation of FRMOCnet OCR and practical guide for OCR measurement end-to-end uncertainty budget calculation. We outlined here the methodology that is used for uncertainty propagation in the HyperCP that is being built on the existing software HyperInSPACE.

We collaborated with our NASA colleagues on HyperInSPACE to enable uncertainty propagation there too, thus some of the structural changes were already applied by NASA in HyperInSPACE v1.1.0 allowing to incorporate uncertainty propagation related parts into existing software.

We proposed to split HyperCP into two separate branches to allow for FRM treatment of instruments which will have full characterisation data, thus the instrument related correction can be applied to reduce instrument related uncertainty contributors. The second so-called default branch uses existing processing but has higher uncertainty contributors included in the uncertainty budget. These are applied on class-based basis and the main knowledge about the uncertainty values come from [AD-1] or other already published research. The uncertainty in the default branch have been implemented, tested on FICE-2 data and few examples were presented in section 6.2 Default branch uncertainties in this report. Generally, the above water radiometry uncertainty from measurements with the two instrument classes under study with uncorrected instruments characteristic during optimal environmental conditions can be considered below the 5% level k=1 for the wavelengths 450 nm-600 nm.

Sea surface reflectance component sensitivity on uncertainty in the various models' inputs components is presented which shows that we can estimate uncertainty associated with ρ component, based on the uncertainty propagation though the ρ evaluation models. We have shown that the two models used currently in HyperInSPACE agree within the model uncertainties for low wind speed conditions. Please note that this sensitivity study does not verify the correctness of the models itself. Currently HyperCP uses Mobley method to evaluate uncertainty in ρ estimation, as although it is possible to do so using Zhang approach too, this is computational expensive to run during the routine operation. The improvement on Zhang model computational efficiently is conducted in the frame of the further HyperCP development, thus when this is implemented the pupny uncertainty evaluation could be reconsidered to evaluate uncertainty for ρ using Zhang method in the operational mode.

The next version of this report v3 will contain the details FRM branch processing and uncertainty evaluation that are currently still being developed and tested.

8 Bibliography

Banks, A. C. *et al.* (2020) 'Fiducial Reference Measurements for Satellite Ocean Colour (FRM4SOC)', *Remote Sensing*, 12(8). doi: 10.3390/rs12081322.

Białek, A. *et al.* (2020) 'Example of Monte Carlo Method Uncertainty Evaluation for Above-Water Ocean Colour Radiometry', *Remote Sensing*, 12(5). doi: 10.3390/rs12050780.

Bialek, A. and Douglas, S. (no date) 'D-180 : Uncertainty Budgets of FRM4SOC Fiducial Reference Measurement (FRM) Ocean Colour Radiometer (OCR) systems used to Validate Satellite OCR products D-180 : Uncertainty Budgets of FRM4SOC Fiducial Reference Measurement (FRM) Ocean Colour Radiom'.

D'Alimonte, D. *et al.* (2021) 'Sea-surface reflectance factor: replicability of computed values', *Optics Express*, 29(16), p. 25217. doi: 10.1364/OE.424768.

'FIDUCEO Vocabulary V2' (2018), p. 13. Available at: https://research.reading.ac.uk/fiduceo/wp-content/uploads/sites/129/2020/12/Fiduceo-Vocabulary-V2.pdf.

Harmel, T. *et al.* (2012) 'Polarization impacts on the water-leaving radiance retrieval from above-water radiometric measurements', *Applied Optics*, 51(35), p. 8324. doi: 10.1364/ao.51.008324.

International vocabulary of metrology – Basic and general concepts and associated terms (VIM). 3rd edn (2008) *Tetrahedron Letters*. doi: 10.1016/j.tetlet.2017.07.069.

JCGM100:2008 (2008) *Evaluation of measurement data - {G}uide to the expression of uncertainty in measurement*.

JCGM101:2008 (2008) *Evaluation of measurement data - Supplement 1 to the 'Guide to the expression of uncertainty in measurement'" - Propagation of distributions using a Monte Carlo method'*.

Mobley, C. D. (1999) 'Estimation of the remote-sensing reflectance from above-surface measurements', *Appl. Opt.*, 38(36), pp. 7442–7455. doi: 10.1364/AO.38.007442.

Talone, M. and Zibordi, G. (2016) 'Polarimetric characteristics of a class of hyperspectral radiometers', *Applied Optics*, 55(35), pp. 10092–10104. doi: 10.1364/AO.55.010092.

Zhang, X. *et al.* (2017) 'Spectral sea surface reflectance of skylight', *Optics Express*, 25(4), p. A1. doi: 10.1364/oe.25.0000a1.

Zibordi, G., Talone, M. and Jankowski, L. (2017) 'Response to Temperature of a Class of In Situ Hyperspectral Radiometers', *Journal of Atmospheric and Oceanic Technology*, 34(8), pp. 1795–1805. doi: 10.1175/JTECH-D-17-0048.1.

Appendix

Uncertainty tree diagrams from FRM4SOC phase-1 study.

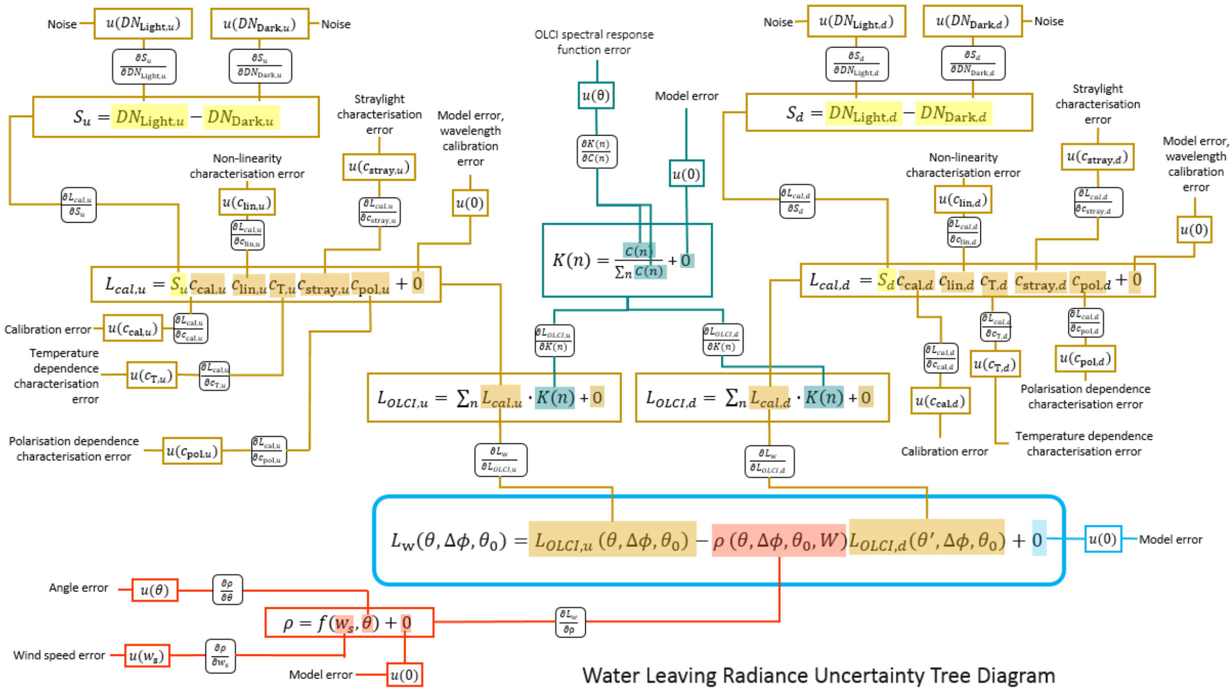

Noise
$u(DN_{\mathrm{Light},u})$
$u(DN_{\mathrm{Dark},u})$
Noise
$\frac{\partial S_u}{\partial DN_{\mathrm{Light},u}}$
$\frac{\partial S_u}{\partial DN_{\mathrm{Dark},u}}$
$S_u = DN_{\mathrm{Light},u} - DN_{\mathrm{Dark},u}$
Straylight characterisation error
$u(c_{\mathrm{stray},u})$
Model error, wavelength calibration error
$u(0)$
Non-linearity characterisation error
$u(c_{\mathrm{lin},u})$
$\frac{\partial L_{\mathrm{cal},u}}{\partial S_u}$
$\frac{\partial L_{\mathrm{cal},u}}{\partial c_{\mathrm{lin},u}}$
$\frac{\partial L_{\mathrm{cal},u}}{\partial c_{\mathrm{stray},u}}$
$L_{\mathrm{cal},u} = S_u c_{\mathrm{cal},u} c_{\mathrm{lin},u} c_{\mathrm{T},u} c_{\mathrm{stray},u} c_{\mathrm{pol},u} + 0$
Calibration error
$u(c_{\mathrm{cal},u})$
$\frac{\partial L_{\mathrm{cal},u}}{\partial c_{\mathrm{cal},u}}$
Temperature dependence characterisation error
$u(c_{\mathrm{T},u})$
$\frac{\partial L_{\mathrm{cal},u}}{\partial c_{\mathrm{T},u}}$
Polarisation dependence characterisation error
$u(c_{\mathrm{pol},u})$
$\frac{\partial L_{\mathrm{cal},u}}{\partial c_{\mathrm{pol},u}}$
OLCI spectral response function error
$u(\theta)$
$\frac{\partial K(n)}{\partial C(n)}$
Model error
$u(0)$
$K(n) = \frac{C(n)}{\sum_n C(n)} + 0$
$\frac{\partial L_{OLCI,u}}{\partial K(n)}$
$\frac{\partial L_{OLCI,d}}{\partial K(n)}$
$L_{OLCI,u} = \sum_n L_{\mathrm{cal},u} \cdot K(n) + 0$
$L_{OLCI,d} = \sum_n L_{\mathrm{cal},d} \cdot K(n) + 0$
Noise
$u(DN_{\mathrm{Light},d})$
$u(DN_{\mathrm{Dark},d})$
Noise
$\frac{\partial S_d}{\partial DN_{\mathrm{Light},d}}$
$\frac{\partial S_d}{\partial DN_{\mathrm{Dark},d}}$
$S_d = DN_{\mathrm{Light},d} - DN_{\mathrm{Dark},d}$
Straylight characterisation error
$u(c_{\mathrm{stray},d})$
Model error, wavelength calibration error
$u(0)$
Non-linearity characterisation error
$u(c_{\mathrm{lin},d})$
$\frac{\partial L_{\mathrm{cal},d}}{\partial S_d}$
$\frac{\partial L_{\mathrm{cal},d}}{\partial c_{\mathrm{lin},d}}$
$\frac{\partial L_{\mathrm{cal},d}}{\partial c_{\mathrm{stray},d}}$
$L_{\mathrm{cal},d} = S_d c_{\mathrm{cal},d} c_{\mathrm{lin},d} c_{\mathrm{T},d} c_{\mathrm{stray},d} c_{\mathrm{pol},d} + 0$
$\frac{\partial L_{\mathrm{cal},d}}{\partial c_{\mathrm{T},d}}$
$\frac{\partial L_{\mathrm{cal},d}}{\partial c_{\mathrm{pol},d}}$
$\frac{\partial L_{\mathrm{cal},d}}{\partial c_{\mathrm{cal},d}}$
$u(c_{\mathrm{T},d})$
$u(c_{\mathrm{pol},d})$
$u(c_{\mathrm{cal},d})$
Polarisation dependence characterisation error
Temperature dependence characterisation error
Calibration error
$\frac{\partial L_w}{\partial L_{OLCI,u}}$
$\frac{\partial L_w}{\partial L_{OLCI,d}}$
$L_w(\theta,\Delta\phi,\theta_0) = L_{OLCI,u}(\theta,\Delta\phi,\theta_0) - \rho(\theta,\Delta\phi,\theta_0,W) L_{OLCI,d}(\theta',\Delta\phi,\theta_0) + 0$
$u(0)$
Model error
Angle error
$u(\theta)$
$\frac{\partial\rho}{\partial\theta}$
$\rho = f(w_s,\theta) + 0$
$\frac{\partial L_w}{\partial\rho}$
Wind speed error
$u(w_s)$
$\frac{\partial\rho}{\partial w_s}$
Model error
$u(0)$
Water Leaving Radiance Uncertainty Tree Diagram

Figure 27. FRM4SOC phase 1 Uncertainty tree diagram for water leaving radiance

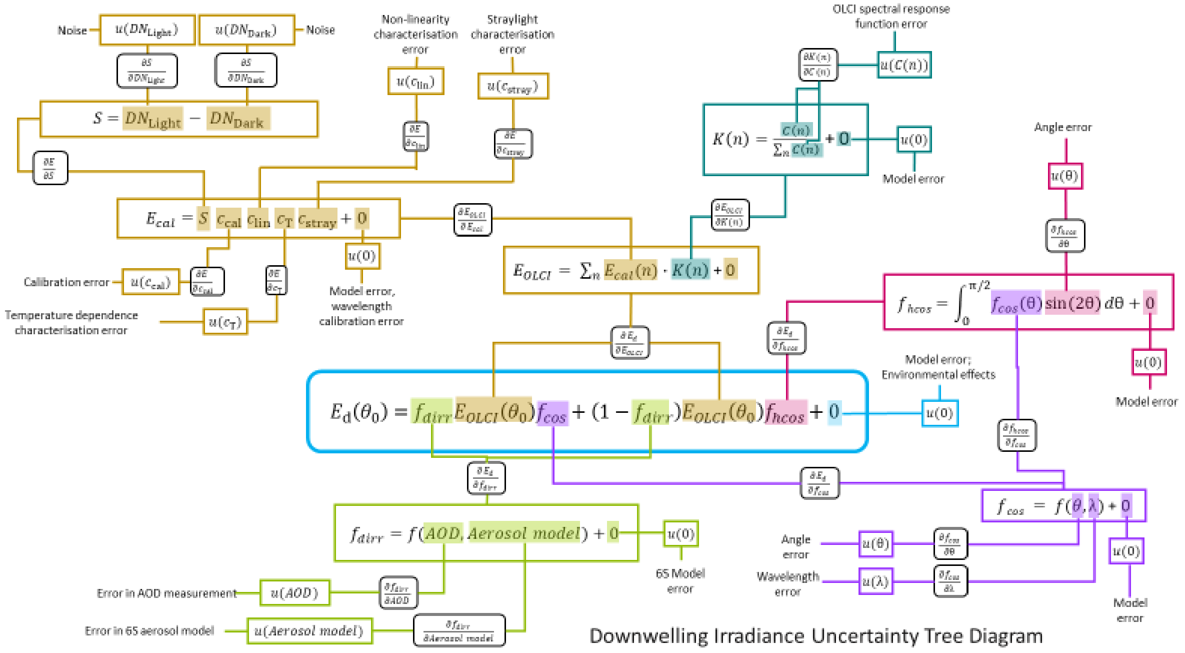

Noise
$u(DN_{\mathrm{Light}})$
$u(DN_{\mathrm{Dark}})$
Noise
$\frac{\partial S}{\partial DN_{\mathrm{Light}}}$
$\frac{\partial S}{\partial DN_{\mathrm{Dark}}}$
$S = DN_{\mathrm{Light}} - DN_{\mathrm{Dark}}$
Non-linearity characterisation error
$u(c_{\mathrm{lin}})$
Straylight characterisation error
$u(c_{\mathrm{stray}})$
$\frac{\partial E}{\partial c_{\mathrm{lin}}}$
$\frac{\partial E}{\partial c_{\mathrm{stray}}}$
$\frac{\partial E}{\partial S}$
$E_{cal} = S\, c_{\mathrm{cal}}\, c_{\mathrm{lin}}\, c_{\mathrm{T}}\, c_{\mathrm{stray}} + 0$
$u(0)$
Model error, wavelength calibration error
Calibration error
$u(c_{\mathrm{cal}})$
$\frac{\partial E}{\partial c_{\mathrm{cal}}}$
$\frac{\partial E}{\partial c_{\mathrm{T}}}$
Temperature dependence characterisation error
$u(c_{\mathrm{T}})$
$\frac{\partial E_{OLCI}}{\partial E_{cal}}$
OLCI spectral response function error
$\frac{\partial K(n)}{\partial C(n)}$
$u(C(n))$
$K(n) = \frac{C(n)}{\sum_n C(n)} + 0$
$u(0)$
Model error
$\frac{\partial E_{OLCI}}{\partial K(n)}$
$E_{OLCI} = \sum_n E_{cal}(n) \cdot K(n) + 0$
$\frac{\partial E_d}{\partial E_{OLCI}}$
Angle error
$u(\theta)$
$\frac{\partial f_{hcos}}{\partial\theta}$
$f_{hcos} = \int_0^{\pi/2} f_{cos}(\theta)\sin(2\theta)\,d\theta + 0$
$u(0)$
Model error
$\frac{\partial E_d}{\partial f_{hcos}}$
Model error; Environmental effects
$u(0)$
$E_d(\theta_0) = f_{dirr} E_{OLCI}(\theta_0) f_{cos} + (1 - f_{dirr}) E_{OLCI}(\theta_0) f_{hcos} + 0$
$\frac{\partial f_{hcos}}{\partial f_{cos}}$
$\frac{\partial E_d}{\partial f_{dirr}}$
$\frac{\partial E_d}{\partial f_{cos}}$
$f_{cos} = f(\theta,\lambda) + 0$
$f_{dirr} = f(AOD, Aerosol\ model) + 0$
$u(0)$
65 Model error
Angle error
$u(\theta)$
$\frac{\partial f_{cos}}{\partial\theta}$
Wavelength error
$u(\lambda)$
$\frac{\partial f_{cos}}{\partial\lambda}$
$u(0)$
Model error
Error in AOD measurement
$u(AOD)$
$\frac{\partial f_{dirr}}{\partial AOD}$
Error in 65 aerosol model
$u(Aerosol\ model)$
$\frac{\partial f_{dirr}}{\partial Aerosol\ model}$
Downwelling Irradiance Uncertainty Tree Diagram

Figure 28. FRM4SOC phase 1 Uncertainty tree diagram for downwelling