ESA Sen4Stat

Sentinels for Agricultural Statistics

D7.0 – DDF - ATBD
"Algorithm Theoretical Basis Document: Cloud-free Composite Product"

Issue/Rev : 1.0

Date Issued : 25-08-21

Milestone: 2

Submitted to European Space Agency

Consortium Partners

Participant Organisation Name	Acronym	City, Country
Université catholique de Louvain	UCLouvain	Louvain-la-Neuve, Belgium
CS Romania	CS RO	Craiova, Romania
Systèmes d'Information à Référence Spatiale - Filiale CLS	SIRS-CLS	Villeneuve d'Ascq, France
Universidad Polytecnica de Madrid	UPM	Madrid, Spain

Contact

Université catholique de Louvain – Earth and Life Institute
Place de l'Université, 1 – B-1348 Louvain-la-Neuve – Belgium
Email : Sophie.Bontemps@uclouvain.be
Internet : https://uclouvain.be/en/research-institutes/eli/elie

Disclaimer

Document sheet

Authors and Distribution

Authors	Sophie BONTEMPS, Nicolas DEFFENSE, Cosmin UDROIU
Distribution	ESA - Benjamin KOETZ, Zoltan SZANTOI

Document Status

Issue/Rev.	Date	Reason
0.1	18/05/2021	Initial version of the deliverable
1.0	25/08/2021	1st version related to the Acceptance Review

Detailed record sheet

From version 1.0 to 1.x

Comment	Section	Problem description	Change

Table of contents

List of figures

List of tables

List of codes

Reference documents

ID	Title
RD.1	Weiss M. and Baret F.. 2016. S2 Toolbox Level 2 products: LAI, FAPAR, FCOVER, V1.1.
RD.2	Weiss, M., Baret, F., Leroy, M., Hautecoeur, O., Bacour, C., Prévot, L., and Bruguier, N. (2002). Validation of neural net techniques to estimate canopy biophysical variables from remote sensing data. Agronomie, 22, 547-554

Acronyms

Acronym	Definition
AD	Applicable Document
ANN	Artificial Neural Network
ATBD	Algorithm Theoretical Basis Document
DDF	Design Definition File
EO	Earth Observation
ESA	European Space Agency
fAPAR	fraction of Absorbed Photosynthetically Active Radiation
fCOVER	fraction of Vegetation Cover
GAI	Green Area Index
L1, L2, L3, L4	Level 1, Level 2, Level 3, Level 4
L8	Landsat 8
LAI	Leaf Area Index
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
NSO	National Statistical Office
S2	Sentinel-2
Sen4Stat	Sentinels for Agricultural Statistics

1 Logical model and processor overview

This Algorithm Theoretical Basis Document (ATBD) describes the algorithm developed to produce cloud-free composite products from Sentinel-2 (S2) - and possibly Landsat 8 (L8) - acquisitions.

The workflow is summarized in Figure 1-1 and is detailed in the following sections of this document.

For a given tile, the composite processor works in 2 main steps:

1) a pre-processing step, which for each date transforms the L2A in a L2A* product
    - to normalize directional effects for S2 (A & B)
    - in the case of L8, to resample it to the same resolution as the corresponding S2 band;
2) a processing step:
    - the composite product is a recurrent process that updates the product date after date;
    - the processor is run each time a new L2A* is available;
        - it uses as input a L2A* and the current composite product;
        - if there is still no composite product available, it is created.

In order to limit the amount of L2A products to keep on line in the disk system, the “Composite” processor has been defined as a recurrent operation. When the first L2A product necessary to produce the composite becomes available, the composite product is initialised. It is then updated each time a new L2A product belonging to the list of L2A products to be processed is obtained.

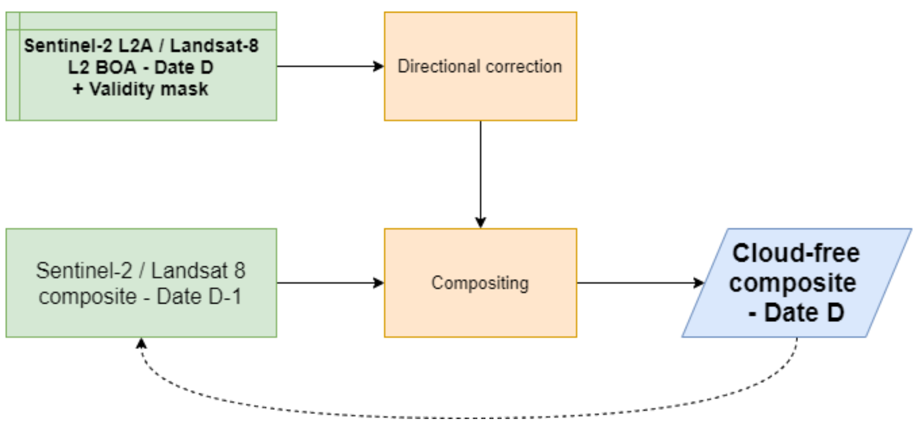

Figure 1-1. Workflow of the cloud-free composite products computation

2 Input data

2.1 Sentinel-2 L2A BOA reflectance

Individual daily observations are considered. All land bands are used except the blue one (B2): B3, B4, B5, B6, B7, B8a, B8, B11 and B12 (Table 2-1).

Table 2-1. S2 spectral characteristics

Acronym	Central wavelength (nm)	Width (nm)	Spatial resolution (m)	Band name
B3	560	35	10	Green
B4	665	30	10	Red
B5	705	15	20	Red edge 1
B6	740	15	20	Red edge 2
B7	783	20	20	Red edge 3
B8	842	115	10	NIR wide
B8a	865	20	20	NIR narrow
B11	1610	90	20	SWIR 1
B12	2190	180	20	SWIR 2

2.2 Landsat 8 L2A BOA reflectance

Like for S2, individual daily observations are considered and all land bands are used except the blue one (B2): B3, B4, B5, B6, B8 (Table 2-2).

Table 2-2. L8 spectral characteristics

Acronym	Central wavelength (nm)	Width (nm)	Spatial resolution (m)	Band name
B3	560	30	30	Green
B4	650	20	30	Red
B5	865	15	30	NIR narrow
B6	1610	40	30	SWIR 1
B8	2200	90	30	SWIR 2

2.3 Sentinel-2 and Landsat 8 validity mask

As detailed in the EO Data Pre-Processing ATBD, the cloud detection is realized using FMask. The FMask output is an image with the values from Table 2-3.

Table 2-3. Fmask classes

FMask class	Description
0	Clear land
1	Clear water
2	Cloud shadow
3	Snow
4	Cloud
255	No data

3 Detailed workflow

3.1 Pre-processing

This step is to be executed before the compositing processing in itself. It uses a Level 2A as input (see section 2) and delivers a pre-processed Level 2A referred hereafter as L2A*.

For S2, this L2A* provides the reflectance values obtained after a directional correction (see section 3.1.1).

For L8, as the viewing angles are not provided for each pixel in the Level 1T product, it is not possible to apply the directional correction. The L8 L2A* product thus correspond to the L2A resampled to the resolution of the corresponding S2 band (see section 3.1.2).

In both cases, the L2A* metadata and validity masks are identical to the L2A metadata and validity masks with only the resolution information changed in the case of L8.

The directional model used for the proposed correction derives from the following references:

- Maignan et al. 2004. “Bidirectional reflectance of Earth targets: Evaluation of analytical models using a large set of spaceborne measurements with emphasis on the Hot Spot.” Remote Sensing of Environment, vol. 90, no. 2, pp. 210-220.
- Bréon and Vermote, 2012. “Correction of MODIS surface reflectance time series for BRDF effects”. Remote Sensing of Environment, vol. 125, pp. 1-9.

The directional correction is only using the coefficients for the S2 red and NIR bands.

3.1.1 Directional correction

The directional correction consists in applying a model to each pixel, except of this pixel is classified as cloud, cloud shadow, water or snow (this status coming from the validity mask). The model depends on the sun and observation angles and on the NDVI of the pixel.

Pseudo-code of the directional model function computation is shown in Code 3-1 (with more details in Annex 2). Input and output variables of the directional model function are provided in Table 3-1 while the parameters are given in Table 3-2.

Table 3-1. Input and output variables of the directional model function

Input data	Description
b	Spectral band
NDVI	NDVI
(θ_s ; θ_v ; δ_ϕ)	Solar zenith Angle, Viewing zenith Angle, Relative Azimuth
Output data	**Description**
=mod-dir=	value of directional model for band b and for the input angles

Table 3-2. Parameters of the directional model function

Input data	Description
V0, V1	Coefficient for the volume scattering function, as a function of spectral band
R0, R1	Coefficient for the roughness scattering function, as a function of spectral band

For each spectral band b:
- \$mod-dir(θ_s, θ_v, δ_ϕ, b) = 1+ (V0(b) + V1(b)*NDVI) * F_V(θ_s, θ_v, δ_ϕ) + (R0(b) + R1(b)*NDVI) * F_R(θ_s, θ_v, δ_ϕ)

A python code to compute FV and FR is provided in Annex

Code 3-1. Computation of the directional model function

Code 3-2 shows the directional correction, with input variables provided in Table 3-3.

Table 3-3. Input variables of the directional correction

Input data	Description
ρ	Reflectance for each pixel and each spectral band
Angles from L2A images (θ_s ; θ_v ; δ_ϕ)	Solar zenith Angle, Viewing zenith Angle, Relative Azimuth
Cloud-Shadow Mask	Mask of cloud and cloud shadow pixels (coming from FMask - see Table 2-3)
Water Mask	Mask of water pixels (coming from FMask - see Table 2-3)
Snow Mask	Mask of snow pixels (coming from FMask - see Table 2-3)

Compute NDVI at 10m resolution => $NDVI_{10m}$

Resample it without aliasing to 20 m => $NDVI_{20m}$

Compute (θ_s ; θ_v ; δ_ϕ) for each pixel and each band according to S2 L1C product Users Manual.
 # The viewing angles are provided as a grid per S2 spectral band and per detector (there are 12 detectors per band). S2 metadata also provide a map of the detectors which were used to observe each pixel. The method consists in finding the detector corresponding to the pixel to process

Apply directional correction to each pixel except water and snow pixels, using directional model computed in Code 1

For each pixel and each band:
- if pixel is not cloud free, shadow free, water and snow free:

$$\rho(\theta s, 0, 0) = \rho(\theta s, \theta v, \delta\phi) * \frac{mod - dir(\theta s, 0,0)}{mod - dir(\theta s, \theta v, \delta\phi)}$$

Code 3-2. Directional correction

3.1.2 Resampling at S2 Resolution

Input and output variables of the resampling function are provided in Table 3-4, while the code is given in Code 3-3.

Table 3-4. Input and output variables of the resampling function

Input data	Description
S2-Bands (L8-bands)	S2-Band corresponding to every L8 band
ResS2(L8) bands	Resolution of each S2 (L8) band
ρ	Reflectance for each pixel and each L8 spectral band
Output data	**Description**
ρ*	Reflectance value for each pixel, for each L8 spectral band with a corresponding S2 spectral band
Cloud-Shadow Mask	Mask of cloud and cloud shadow pixels (coming from FMask - see Table 2-3)
Snow Mask	Mask of snow pixels (coming from FMask - see Table 2-3)

```
For each L8 spectral band bL8:
- resample the L8 band to the resolution Res(S2-bands(bL8))

Resample cloud-shadow mask and snow mask at 10 and 20m resolution
```

Code 3-3. Resampling L8 to S2 resolution

3.2 Composite update with a new product

The step is made of 3 successive modules:

1. Resample cloud and snow masks to 10m resolution;
2. Compute individual weights for each pixel in the image;
3. Update the composite product.

3.2.1 Composite product update

This module updates the composite product using the recurrent expression of a weighted average (Code 3-4). The weighted average is only computed for land and water pixels. Cloud, cloud shadow and snow pixels are discarded from the weighted average, unless no land or water pixel was observed so far in the composite window.

If it is the first iteration of the composite product update, an initial composite product with no-data pixels everywhere must be created. Land and water are considered the same class to avoid issues with rice, which is flooded at the early stages.

In the case of both S2 and L8 sensors are used, the weighted average is computed with a weight counter per each S2 band. This is not ideal as it adds a data volume equivalent to that of surface reflectance. If only S2 is used, it would be possible to use one counter for the 20m bands and a counter for the 10m bands (but that would not allow to account for possible failures in one of the bands of one of the satellites).

In order to reduce the data volume, the weighted average date and flag are provided for the red band only. It is assumed there will always be a red band and a blue band, which are required to choose the reflectance values.

Input and output variables of the composite update function are provided in Table 3-5.

Table 3-5. Input and output variables of the composite update function

Input data	Description
S2-Bands (L8-bands)	S2-Band corresponding to every L8 band L2A* product
$date_N$	Date from input product
ρ_{N*}	Reflectance value for each pixel, and each spectral band from input L2A* product
Cloud-Shadow Mask	Mask of cloud and cloud shadow pixels (coming from FMask - see Table 2-3)
Water Mask	Mask of water pixels (coming from FMask - see Table 2-3)
Snow Mask	Mask of snow pixels (coming from FMask - see Table 2-3)
L3A product	Composite product (if already created)
W_{N-1}	Weight for each pixel obtained so far
$date_{N-1}$	Weighted average date for L3A product so far
N-1	Weighted average reflectance value so far, for each pixel, and each spectral band
$flag_{N-1}$	Status of each L3A pixel: cloud, water , snow
Output data	**Description**
L3A product	Composite product
W_N	Weight counter for each pixel and for each band
$date_N$	Weighted average date for L3A product so far
N	Weighted average reflectance value so far, for each pixel, and each spectral band
$flag_N$	Status of each L3A pixel: cloud, water , snow

INITIALIZATION IS NECESSARY

1) If L3A product does not exist:

for each pixel:
- $flag_{N-1}$ = no data
- $date_{N-1}$ = no data
- for each S2 band:
  * $\bar{\rho}_{N-1} = no\ data$
  * $W_{N-1} = no\ data$

2) If L3A product exists, update with current product:

Compute weights for L2A product of date N => w_N per band and pixel

For resolutions res in (10m,20m):
- for each pixel:

  * if pixel not cloud/shadow, snow, water (nominal case):
    $flag_N$ = land
    for each S2 band with res resolution:

      if band is available: *# with L8, some bands are not available #*
        * $\bar{\rho}_N = \frac{W_{N-1} * \bar{\rho}_{N-1} + w_N * \rho_N}{W_{N-1} + w_N}$
        * $\overline{date}_N = \frac{W_{N-1} * date_{N-1} + w_N * date_N}{W_{N-1} + w_N}$
        * $W_N = W_{N-1} + w_N$

      else:
        * $\bar{\rho}_N = \rho_{N-1}$
        * $W_N = W_{N-1}$
        * if band is red: $\overline{date}_N = \overline{date}_{N-1}$

  * else (degraded case):
    if pixel is snow or water: *# replace the reflectance value #*
      if pixel is water: $flag_N$ = water
      if pixel is snow: $flag_N$ = snow
      for each S2 band with res resolution:

        if band is available:

          if W_{N-1} == 0 *# pixel never observed without cloud, water or now #*
            * $\bar{\rho}_N = \rho$
            * $W_N = 0$
            * if band is red: $\overline{date}_N = date_N$

          else *# pixel already observed cloud free, keep the previous weighted average #*
            * $\bar{\rho}_N = \bar{\rho}_{N-1}$
            * $W_N = W_{N-1}$
            * if band is red:
              $\overline{date}_N = \overline{date}_{N-1}$
              $flag_N$ = land

else: *# band not available, keep previous values #*
* $\bar{\rho}_N = \bar{\rho}_{N-1}$
* $W_N = W_{N-1}$

if composite pixel is cloud or cloud shadow: *# pixel never observed cloud snow or water free #*

if $flag_{N-1}$ is no-data *# replace no data with cloud #*
for each S2 band with res resolution:

if band is available:

* $\bar{\rho}_N = \rho$
* $W_N = 0$
* if band is red:
  $\overline{date}_N = date_N$
  $flag_N$ = cloud

else:

* $\bar{\rho}_N = no\ data$
* $W_N = 0$

if $flag_{N-1}$ is cloud or cloud shadow *# replace value only if new reflectance in the blue is smaller #*
if $\rho(blue) < {}_{N-1}(blue)$
for each S2 band with res resolution:

if band is available:

* $\bar{\rho}_N = \rho$
* $W_N = 0$
* if band is red:
  $\overline{date}_N = date_N$
  $flag_N$ = cloud

else:

* $\bar{\rho}_N = \bar{\rho}_{N-1}$
* $W_N = 0$

else: *# blue value too high #*
$flag_N = flag_{N-1}$
for each S2 band with res resolution:
* $\bar{\rho}_N = \bar{\rho}_{N-1}$
* $\overline{date}_N = \overline{date}_{N-1}$
* $W_N = 0$

Code 3-4. Composite product update

3.2.2 Weights computation

3.2.2.1 Weights on sensor

The idea here is to be able to merge different sensors (for instance, L8 and S2) in the same composite product (Code 3-5). As the information brought by each sensor is not equivalent, different weights may be used depending on the sensor.

Input and output variables related to the sensors weighting are provided in Table 3-6 while the parameters are given in Table 3-7.

Table 3-6. Input and output of the sensors weighting

Input data	Description
Current date sensor	S2 or L8
Output data	**Description**
W_{sensor}	Weight on sensor (one value per product)

Table 3-7. Parameters of the sensors weighting

Input data	Description	Default value
$W_{sensor\ -\ S2A}$	Weight for Sensor S2A	1
$W_{sensor\ -\ S2B}$	Weight for Sensor S2B	1
$W_{sensor\ -\ L8}$	Weight for Sensor L8	0.33

If current date sensor is S2A:
$W_{sensor} = W_{sensor\ -\ S2A}$

If current date sensor is S2B:
$W_{sensor} = W_{sensor\ -\ S2B}$

If current date sensor is L8:
$W_{sensor} = W_{sensor\ -\ L8}$

Code 3-5. Sensors weights

3.2.2.2 Weights on clouds

Input and output variables related to the sensors weighting are provided in Table 3-8 while the parameters are given in Table 3-9. Weights computation is shown in Code 3-6.

Table 3-8. Input and output of the cloud weighting

Input data	Description
Cloud-Shadow Mask	Mask of cloud and cloud shadow pixels from L2A product (coming from FMask - see Table 2-3) at 20m resolution
Output data	**Description**
W_{cloud}	Weight on distance to cloud (one value per pixel and per resolution, 10m, 20m)

Table 3-9. Parameters of the cloud weighting

Input data	Description	Default value
Coarse resolution	coarse resolution for quicker convolution	240m
Sigma-large-cloud-dist	standard deviation of gaussian filter for distance to large clouds	10
Sigma-small-cloud-dist	standard deviation of gaussian filter for distance to small clouds	2

```
Binarize cloud mask (0 if no cloud no shadow, 1 if either cloud or shadow)

Under-sample the binary cloud mask to lower resolution (240m), with a bicubic resampling, without aliasing
(radius must be 2 times the resampling factor)

Binarize resampled mask (1 if above 0.5, 0 if under 0.5)

Compute distances at low resolution:

   Dist_Large Cloud, Low Res = gaussian-filter(binarized low res cloud mask, standard deviation = sigma-large-
   cloud-dist)

   Dist_Small Cloud, Low Res = gaussian-filter(binarized low res cloud mask, standard deviation = sigma-small-
   cloud-dist)

   For each resolution (10m, 20m):

      * Resample at full resolution (10 and 20m)
      * Dist_Large Cloud, Full Res = bilinear-oversampling(Dist_Large Cloud, Low Res)
      * Dist_Small Cloud, Full Res = bilinear-oversampling(Dist_Small Cloud, Low Res)
      * For each pixel, compute weight:
         W_Cloud= (1 - Dist_Large Cloud, Full Res) * (1 - Dist_Small Cloud, Full Res)
```

Code 3-6. Cloud weights

3.2.2.3 Weights on dates

The idea here is to give more weight to the dates close to the central date of the compositing time window, and less weight to the edges of this window. Too large difference in weights would result in not considering at all the dates at the edges of the synthesis window, unless they are the only one available in the synthesis window - which is not desired. As a result, it was decided to weight the images by 1 in the middle of the synthesis window, and by 0.5 at the edges (Code 3-7).

Input and output variables related to the dates weighting are provided in Table 3-10 while the parameters are given in Table 3-11.

Table 3-10. Input and output of the dates weighting

Input data	Description
$date_{L2A}$	L2A date, expressed in days
$date_{L3A}$	L3A date, expressed in days
Δ_{Max}	Half Synthesis period (Days)
Output data	**Description**
W_{date}	Weight on date (one value per product)

Table 3-11. Parameters of the dates weighting

Input data	Description	Default value
W_{Min}	Minimum weight at edge of synthesis time window	0.5

 # abs( ) means absolute value #

$$W_{date} = 1 - \frac{abs(date_{L2A} - date_{L3A})}{\Delta_{Max}} * (1 - W_{Min})$$

Code 3-7. Dates weights

3.2.2.4 Weights on Aerosol Optical Thickness (AOT)

Input and output variables related to the AOT weighting are provided in Table 3-12while the parameters are given in Table 3-13. Weights computation is explained in Code 3-8.

Table 3-12. Input and output of the AOT weighting

Input data	Description
AOT	AOT values from L2A product at 20m resolution
Output data	**Description**
W_{AOT}	Weight on AOT (one value per pixel and per resolution, 10m, 20m). The higher the AOT, the lower the weight

Table 3-13. Parameters of the AOT weighting

Input data	Description	Default value
W_{AOTMin}	Min weight, depending on AOT	
W_{AOTMax}	Max weight, depending on AOT	
AOT_{Max}	Maximum value of the linear range for weights w.r.t. AOT	

```
Resample AOT at 10m and 20m resolution

For each resolution (10m, 20m):

    For each pixel:

        * If (AOT(pix) <= AOT_Max):

            W_AOT = W_AOTMin + (W_AOTMax - W_AOTMin) * (1 - AOT(l,p)/AOT_Max)

        * else:

            W_AOT = W_AOTMin
```

$$W_{AOT} = W_{AOTMin} + (W_{AOTMax} - W_{AOTMin}) * (1 - \frac{AOT(l,p)}{AOT_{Max}})$$

$$W_{AOT} = W_{AOTMin}$$

Code 3-8. AOT weights

3.2.2.5 Computation of total weight (product of individual weights)

The computation of the total weight is shown in Code 3-9, with input and output variables provided in Table 3-14.

Table 3-14. Input and output of the total weight computation

Input data	Description
L2A	L2A product from date N
Output data	**Description**
W_N	Weight on date N (one value per pixel and per resolution, 10m, 20m)

```
Compute weight on sensor (W_sensor)
Compute weight on date (W_date)
Compute weight on AOT (W_AOT)
Compute weight on cloud (W_cloud)

For each resolution (10m, 20m):

    W_N = W_sensor * W_date * W_AOT * W_cloud
```

Code 3-9. Total weight computation

4 Output

4.1 Composite product timeseries

The composite product contains the weighted average surface reflectance for all S2 (L8) bands available in the L2A products, at their original resolution (10m or 20m for S2, 30m for L8). The S2 bands at 60m resolution are not provided as these bands are useful for atmospheric correction but not aimed at observing the surface.

4.2 Metadata and quality flags

- Flags:
    - one byte per pixel, 10m resolution
    - values:
        - no-data: pixel was never observed during compositing period;
        - cloud: pixel is always cloudy or within cloud shadow;
        - snow: pixel is always covered by snow for the available cloud free observations;
        - water: pixel is always covered by water for the available cloud free observations;
        - land: pixel was free from cloud, snow and water at least once during composite period;
- Dates:
    - a weighted average of dates used in the composite is provided for each pixel;
    - to be provided only at 20m resolution;
- Number of valid observations over the period.

A table of the available dates is provided in the Composite metadata file.

5 Annex 1: Table of correspondence between Landsat 8 and Sentinel-2 bands

S2 Band	L8 band	wavelength (nm)	S2 Resol. (m)	provided in L3A
1	1	450	60	No
2	2	490	10	Yes
3	3	560	10	Yes
4	4	670	10	Yes
5	-	705	20	Yes
6	-	740	20	Yes
7	-	780	20	Yes
8	-	820	10	Yes
8a	5	865	20	Yes
9	-	940	60	No
10	6	1340	60	No
11	7	1650	20	Yes
12	8	2200	20	Yes

6 Annex 2: directional model

```
#This python_code provides the Ross_Thick- Li Sparse directional model
import math as m
import numpy as np
import pylab as p

class angles:
    #constructor
    def __init__(self,theta_s,phi_s,theta_v,phi_v):
self.theta_s=theta_s*m.pi/180
self.theta_v=theta_v*m.pi/180
self.phi=(phi_s - phi_v)*m.pi/180
#if self.phi < 0 :
    #self.phi=self.phi + 2*m.pi

    #function delta
    def delta(self):
delta=m.sqrt(m.tan(self.theta_s)*m.tan(self.theta_s) + m.tan(self.theta_v)*m.tan(self.theta_v) - 2*m.ta
return delta

    #Air Mass
    def masse(self):
masse=1/m.cos(self.theta_s)+1/m.cos(self.theta_v)
#print masse
return masse

    #Function xsi
    def cos_xsi(self):
cos_xsi=m.cos(self.theta_s)*m.cos(self.theta_v) + m.sin(self.theta_s)*m.sin(self.theta_v)*m.cos(self.ph
return cos_xsi

    def sin_xsi(self):
x=self.cos_xsi()
sin_xsi=m.sqrt(1 - x*x)
return sin_xsi

    def xsi(self):
xsi=m.acos(self.cos_xsi())
return xsi

    xsi_0=1.5*m.pi/180.
```

```
    #Function t
    def cos_t(self):
trig=m.tan(self.theta_s)*m.tan(self.theta_v)*m.sin(self.phi)
d=self.delta()
coef=1  #Coef=1 looks good, but Bréon et Vermote use Coef=2
cos_t=min(max(coef/self.masse()*m.sqrt(d*d + trig*trig),-1),1)
return cos_t

    def sin_t(self):
x=self.cos_t()
sin_t=m.sqrt(1 - x*x)
return sin_t

    def t(self):
#print 'theta_v %f cos_t %F'%(self.theta_v*180/m.pi,self.cos_t())
t=m.acos(self.cos_t())
return t

   #function FV Ross_Thick,  V stands for Volume
    def FV(self):

FV=self.masse()/m.pi*(self.t() - self.sin_t()*self.cos_t() - m.pi) + (1+self.cos_xsi())/2/m.cos(self.th
return FV

    #function FR  Li-Sparse, R stands for Roughness
    def FR(self):
A=1/(m.cos(self.theta_s)+m.cos(self.theta_v))

FR=4/3./m.pi*A*((m.pi/2-self.xsi())*self.cos_xsi()+self.sin_xsi())*(1+1/(1+self.xsi()/self.xsi_0))- 1./
return FR

    def dir_mod(self,kV,kR) :
rho=1 +kV*self.FV() + kR*self.FR()
return rho
```